

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

DOTTORATO DI RICERCA
IN DISEGNO E METODI DELL'INGEGNERIA INDUSTRIALE

CICLO XXI

Settore/i scientifico disciplinari di afferenza: ING-IND/03 Meccanica del volo

**Design, implementation and tests of Real-time
Feed-forward controller and navigation system
for a small scale UAV helicopter.**

Presentata da: Ing. Filippo Zanetti

Coordinatore Dottorato
Chiar.mo Prof. Ing. Franco Persiani

Relatore
Prof. Ing. GianMarco Saggiani

Esame Finale Anno 2009

Tables of contents

| | |
|---|-----------|
| Tables of contents | 3 |
| Nomenclature | 5 |
| Acronyms | 5 |
| 1 Motivation and Background | 7 |
| 1.1 UNIBO RUAV Project Overview | 8 |
| 2 UNIBO RUAV System description | 11 |
| 1.1 Flight Computer..... | 12 |
| 2.1 CRIO Real Time Application Design..... | 14 |
| 2.2 Sensors..... | 15 |
| 2.3 Attitude Heading and Reference System (AHRS) | 15 |
| 3 Development of Dynamic Model Structure | 17 |
| 3.1 Equation of Motion..... | 17 |
| 3.2 Rotor/Stabilizer-Bar Dynamics | 18 |
| 3.3 Heave Dynamics..... | 20 |
| 3.4 Yaw Dynamics | 21 |
| 3.5 Unibo RUAW dynamics linear model | 22 |
| 4 Baseline and FF controllers architectures | 25 |
| 4.1 Baseline controller description | 25 |
| 4.2 Feed-Forward controller description | 26 |
| 4.3 Transfer Functions..... | 27 |
| 4.4 Vertical Down Velocity Controller | 29 |
| 4.5 Heading Controller | 30 |
| 4.6 Engine Governor | 30 |
| 4.7 Discrete-time implementation of Baseline controller..... | 32 |
| 4.8 Discrete-time implementation of FF+PI control model | 34 |
| 5 Guidance..... | 37 |
| 5.1 Altitude Controller..... | 37 |
| 5.2 Lateral Track Controller | 38 |
| 5.3 Fixed-Heading Navigation | 43 |
| 5.4 Baseline controller In-Flight Autopilot Tests..... | 45 |
| 6 RUAV Dynamic model Identification..... | 49 |
| 6.1 On-Axis Identification..... | 49 |
| 6.2 Pitch and roll rate..... | 51 |
| 6.3 Vertical speed dynamic | 54 |
| 6.4 OFF-Axis Identification in open loop | 55 |

| | | |
|------|---|-----------|
| 6.5 | OFF-Axis Identification in Closed Loop | 59 |
| 6.6 | Flight Validation of the Baseline Control System Model..... | 62 |
| 6.7 | Disturbances and sensor noise model | 64 |
| 7 | Comparison of controllers tuned with the basic gains..... | 69 |
| 7.1 | Stability margins | 69 |
| 7.2 | Three-axis control and cross effects disturbances..... | 70 |
| 7.3 | Time domain comparison with basic gains set | 74 |
| 8 | Hardware in the Loop (HIL) Test bench..... | 77 |
| 9 | Baseline and FF controllers parameters automatic tuning..... | 79 |
| 10 | Comparison of equal-performances controllers | 81 |
| 10.1 | Stability analysis | 81 |
| 10.2 | HIL tests | 83 |
| 11 | Sensitivity analysis of Stability upon plant identified parameters | 87 |
| 12 | Performances comparisons in Navigation..... | 91 |
| 13 | Conclusions and Outlook..... | 95 |
| 14 | List of Figures | 97 |
| 15 | Bibiliography..... | 99 |

Nomenclature

| | |
|---|--|
| a, b | longitudinal and lateral rotor flapping |
| e | command-attitude transfer function gain coefficient |
| $e(t)$ | white noise |
| g | (9.81 m/s ²) acceleration of gravity |
| p, q | roll, pitch rates |
| $y(t)$ | noise in velocity signals |
| u, v, w | longitudinal, lateral and vertical speed |
| $A_{lon}, B_{lat}, X_u, Y_v, X_a$ | on-axis derivatives |
| Y_b, L_b, M_a, Z_{coll} | on-axis derivatives |
| M_u, M_v, L_u, L_v | speed derivatives |
| $A_{lat}, B_{lon}, M_b, L_a, M_{coll}$ | off-axis derivatives |
| $F(q)$ | auto regressive polynomial filtering function |
| K_p, K_i, K_d | baseline inner loop PID parameters – Longitudinal |
| K_{pv}, K_{iv}, K_{dv} | baseline outer loop PID parameters – Longitudinal |
| $K_{pLat}, K_{iLat}, K_{dLat}$ | baseline inner loop PID parameters – Lateral |
| $K_{pvLat}, K_{ivLat}, K_{dvLat}$ | baseline outer loop PID parameters – Lateral |
| K_{pm}, K_{im} | feedforward inner loop PI parameters – Longitudinal |
| K_{pvm}, K_{ivm} | feedforward outer loop PI parameters – Longitudinal |
| K_{pmLat}, K_{imLat} | feedforward inner loop PI parameters – Lateral |
| K_{pvmLat}, K_{ivmLat} | feedforward outer loop PI parameters – Lateral |
| $T_{filt}, T_{filtPhi}$ | feedforward filter constants – Longitudinal, Lateral |
| θ, ϕ | longitudinal and lateral attitude angles |
| δ | command-attitude transfer function damping coefficient |
| $\delta_{lon}, \delta_{lat}, \delta_{coll}$ | cyclic longitudinal, lateral, collective control inputs |
| τ_e | main rotor time constant |
| ω_{np}, ω_{nq} | lateral, longitudinal fuselage-rotor-bar natural frequencies |

Acronyms

| | |
|---------|--|
| AHRS | Attitude and Heading Reference System |
| AR | Auto Regressive |
| AV | Air Vehicle |
| CAPECON | Civil uav APplications & Economic effectivity of potential CONfiguration solutions |
| COTS | Commercial Off-The-Shelf |
| CRIO | CompactRIO |
| DL | Data Link |
| EU | European Union |
| FCS | Flight Control System |
| FF | Feedforward |
| FFA | Feedforward Action |
| FMS | Flight Management 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 |
| MGCS | Mobile Ground Control Station |
| MGS | Mobile Ground Segment |
| MMP | Mission Modular Payload |
| NGCS | Navigation Guidance & Control System |
| NI | National Instruments |
| PID | Proportional Integral Derivative |
| PI | Proportional Integral |
| PWM | Pulse Width Modulation |
| R/C | Radio Controlled |
| RF | Radio Frequency |
| RUAV | Rotary Wing Unmanned Aerial Vehicle |
| SISO | Single Input / Single Output |
| SS | States space |
| 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 |

1 Motivation and Background

It is well known that Unmanned Air Vehicles (UAVs) may represent a promising and cost-effective alternative to manned aircraft for a large number of civil applications. Compared to traditional air vehicles, UAVs may, in fact, offer significant advantages in terms of human safety (especially in dull, dirty and dangerous missions), operational cost reduction and work rate efficiency. In particular Rotorcraft UAV (RUAV) systems, due to their versatile flight modes, maneuverability and vertical take-off and landing capabilities, represent even a more promising solution than fixed wing UAVs.

In the last years UNIBO has developed an unmanned small scale helicopter that is now capable of autonomous flight and that can be used inside the Universities as a platform for researches in control and navigation laws, meanwhile it could be proposed as a technological prototype for industries interested in UAV development and manufacturing. In order to take advantage of existing and cost effective technology, UNIBO has used Commercial Off The Shelf (COTS) sensors and electronics for its RUAV avionics package.

The analysis and design of a good flight control system requires the knowledge of an accurate model of vehicle dynamics [1]: such model can be obtained using the known System identification techniques used for bigger machines, with some simplifications.

In this thesis a simple System Identification Procedure for Control Design is presented. In particular the identified system shall be used, in the following, to compare performances of a traditional PID controller that will be referred as Baseline controller [2,3], versus a Feedforward control algorithm based on dynamic model inversion of longitudinal and lateral dynamics. The proposed time-domain identification procedure is entirely developed in Matlab-Simulink environment, and requires no other external software applications.

The control of small scale helicopter (Rotorcraft UAV, RUAV), in order to maintain a stable attitude and to follow a desired trajectory, is particularly critical since it is well known that helicopters are inherently unstable systems. A good number of papers have been written in the last years about methodologies for increasing the limited performances of this class of small scale helicopters, but only few of them use the identified dynamic models to support advanced control design [1].

The validation of this innovative model-based feedforward (FF) controller for the UNIBO RUAV, as well as comparison in terms of performances with a Baseline controller

will be performed in a Hardware In the Loop (HIL) test bench.

In a first step the two control systems will be tuned with the same basic gains in order to assess the effect of the FF term, then in a second step, two sets of automatically tuned parameters (one for the Baseline and another one for the FF) will be used. These two sets of gains will be automatically tuned in order to make an impartial analysis by superimposing that the two controllers have similar performances (rise time, settling time).

Hardware In the Loop tests, as well as stability margins analysis, will demonstrate how, tuning the two systems to achieve the same performances, the feedforward controller allows to work with greater stability and, hence, with less oscillating attitudes. Finally, in the last section, it will be shown how the FF control architecture improves the tracking performances of a given guidance logic.

1.1 UNIBO RUAV Project Overview

The increasing interest in military Unmanned Air Vehicles (UAVs) 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 [4]. 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, to attempt to kick-start a civil UAV industry in Europe and try to fill the gap with the United States. In the last years, the University of Bologna (UNIBO) has carried out several research projects concerning the development and manufacturing of fixed wing UAV systems for the civil aviation market.

The goal to be achieved with the UNIBO RUAV research program was to develop a helicopter capable of autonomous flight which could be used inside the university as platform for researches in control and navigation laws; meanwhile it should be proposed as technological prototype to industry interested in UAV development and manufacturing.

An UAV system is generally constituted by at least four main integrated sub-systems (see figure1): the air vehicle (AV), the ground support system, the data link (DL) and the data distribution (DD) [5].

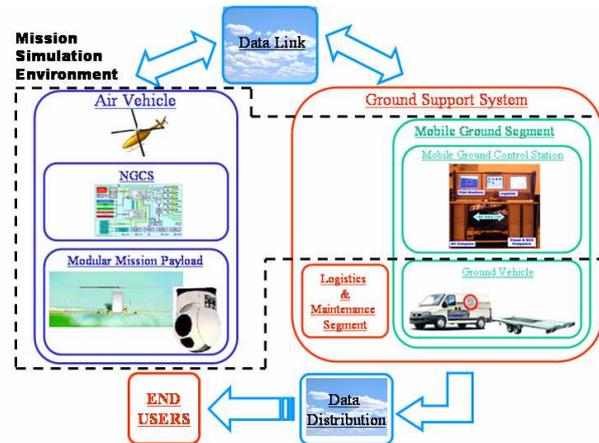


Figure 1: UAV System

- The AV includes all the airborne systems: the basic helicopter platform, the onboard computer and sensors, the mission payload and all the software necessary to guide, navigate and control the helicopter.
- The ground support system includes all the ground infrastructures and equipments to enable the AV operations, such as a mobile ground control station (GCS), a logistic and maintenance segment and a Ground Vehicle.
- The Data Link supports video, data and telemetry communications between the AV and the Ground Support Systems, while the Data Distribution is able to transmit annotated significant data, collected at the GCS, to potential users at remote locations.

The subsystem hardware and software equipments can be much or less sophisticated, depending on the RUAV system size and complexity.

For the purpose of the RUAV program, a small scale hobby model helicopter was used as flying platform, which was certainly a significant physical constraint for the RUAV subsystem equipment choice and development.

The work performed to develop the RUAV platform was carried out following a series of subsequent logical steps:

- first the RUAV hardware (including the onboard avionics, the air vehicle and the data link system) was selected and interfaced, placing attention to vibration isolation, electromagnetic interference and accessibility
- following the hardware set-up, sensor data acquisition software was developed and tested in flight in order to ensure sensor measurement reliability. This step plays a

crucial role in a RUAV development because, if the helicopter has to fly autonomously, reliable information about its states is needed by the onboard control and navigation system.

- parallel to the hardware set-up, simulation plays an important role in the development of an autonomous helicopter. A simulation model was developed, based on helicopter dynamics identification flight tests, to be used for the design of the onboard control and navigation algorithm
- once the previous task were completed, the onboard hardware and software were integrated into the simulation loop using a Hardware In the Loop (HIL) simulator. In this scenario, performance and possible errors of the onboard software can be detected during intensive ground safe and risk free tests
- in the end, autopilot flight test were performed for final verification and tuning of the control and navigation system.

One important aspect to be taken into account in the development of a RUAV system is that it is, actually, an aerial robot. The set-up of a capable task-worthy aerial robots is essentially an integration effort and, always, requires knowledge of several different disciplines and experimentation on new system development. In the past years most of the research efforts in miniature autonomous helicopter were lost for hardware integration and for obtaining reliable sensor measurement. For that reasons, taken also into account the outcomes of the CAPECON program, it was decided to evaluate the feasibility of using COTS sensors and electronics for the RUAV avionics package. Both the hardware and the software were integrated placing attention to modularity, growth potential, versatility and possibility for ease reconfiguration and software implementation. Results achieved in this work showed that the selected hardware and the onboard software were able to provide accurate flight data measurement and good helicopter control capabilities. Thanks to its modular architecture and accurate flight data measurement capabilities, the RUAV system may become a useful research test bench in several different field such as:

- aircraft /rotorcraft dynamic model identification
- researches in control and navigation laws (fast and ease software implementation could results also in a speed up of the research time)
- researches in man machine interface and air system integration which is addressed as one of the most critical technology aspect for the future development of the civil UAVs and their integration into the airspace [6,7].

2 UNIBO RUAV System description

The UNIBO RUAV, shown in Figure 2, is built around a modified Hirobo Eagle II 60 hobby helicopter which was modified to accommodate the avionics hardware, equipped with a more powerful engine, longer fiberglass blades, both for the main and the tail rotor, and longer tail boom. The new main rotor is a 2 blades see-saw type rotor with 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, and a 1.84 m diameter; the helicopter total mass is about 11.2 kg. A National Instruments CompactRIO system has been selected as flight computer and performs both the task of Autopilot and Flight Management System (FMS). For flight data acquisition a Crossbow NAV420 GPS-aided Attitude and Heading Reference System (AHRS) and ultrasonic sensors have been installed to provide accurate signals in velocity, altitude and helicopter attitude. Details of airframe set-up and performance calculation can be found in [8,9].

The on-board computer NI compactRio is programmed using Labview coding language that allows quick set up of different control logic algorithms. In a first step a traditional PID controller [2,3,10] has been implemented, while recently the proposed advanced controller based on feed-forward compensation has been coded.

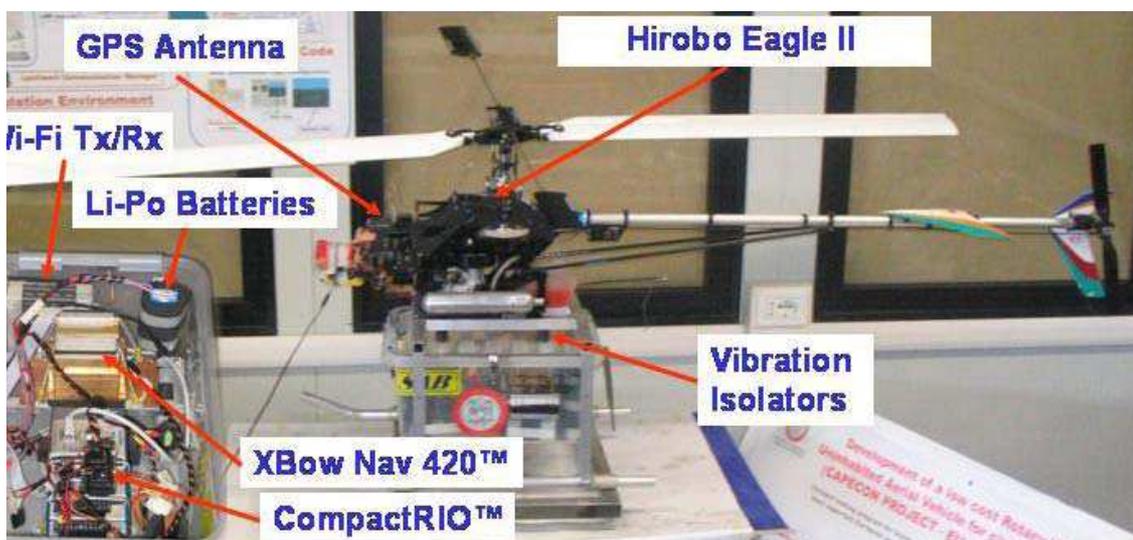


Figure 2: UNIBO Rotary wing UAV

The main helicopter characteristics are:

1. main rotor diameter: 1840 mm
- tail rotor diameter: 330 mm
2. total helicopter mass: 11.2 kg
3. engine: OS 91 Engine 15 cc; power 2.9 CV
4. main rotor rpm: 1200-1300
5. tail rotor rpm: 5000 -6000
6. payload carrying capabilities: 5-6 kg

1.1 Flight Computer

The CRIO system from NI was selected as flight computer due to its ability to fulfill many among the stated design requirements. Particularly, the most important CRIO features that encouraged its choice as onboard computer for the UNIBO RUAV system were:

1. modular and versatile architecture
2. easily reconfigurable with minimal time investment
3. ultrahigh performance and low power consumption
4. relatively low cost system
5. ease and open access to low level hardware resources
6. rapid embedded control and acquisition system development that rival the performance and optimization of custom-designed circuitry
7. possibility to use LabView graphical programming tool to develop a variety of different applications
8. relatively small size and weight compared to its control and data acquisition capabilities

The CRIO platform includes the CRIO-9004 real time controller endowed with an industrial Pentium 200 MHz floating –point processor, a four slot reconfigurable chassis featuring three million gate FPGAs chipset and a wide variety of analog\digital I\O module types.

Figure 5 shows the CRIO configuration actually mounted on the UNIBO RUAV system.

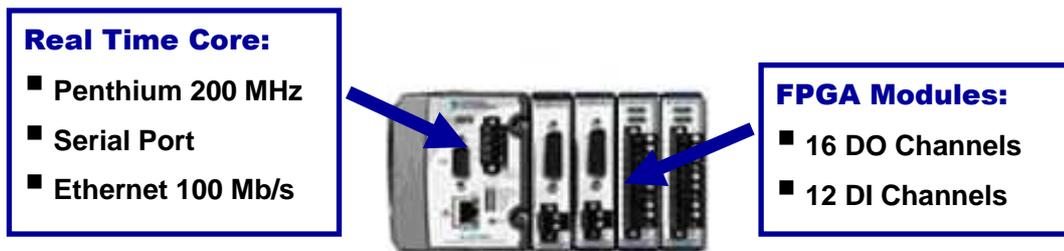


Figure 3: National Instruments CRIO Onboard Computer

The real time controller also features a 100 Mb/s Ethernet port for network communication with an host computer and a 9 PIN serial port.

The FPGA module currently used are:

9. CRIO 9411 mounted in slot 1 having 6 digital input channels
10. another CRIO 9411 mounted in slot 2 having 6 digital input channels
11. CRIO 9474 mounted in slot 3 having 8 digital output channels
12. another CRIO 9474 mounted in slot 4 having 8 digital output channels

Each CRIO module contains already build in signal conditioning.

FPGA devices are very useful and powerful since they combine the versatility of a reconfigurable digital architecture with a matrix of configurable-logic blocks surrounded by a periphery of I/O channels. This way, signal can be routed within the FPGA matrix in any arbitrary manner by programmable interconnected switches and wire routes (figure 4).

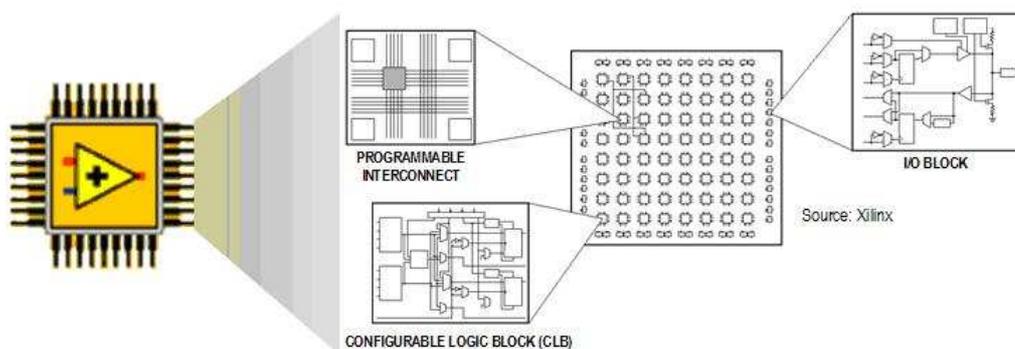


Figure 4: CRIO Field Programmable Gate Array (FPGA) Structure

Control loops can be also implemented inside the FPGA environment using “while loops” up to 40 MHz (25 ns). Moreover, these FPGA modules are ease programmable with NI LabView without need to know specialized hardware design languages such as VHDL (the LabView code is directly compiled in VHDL before being downloaded on the FPGA devices).

2.1 CRIO Real Time Application Design

The real time control and acquisition system which is possible to develop with the CRIO system typically contains four main components (see figure 5):

1. RIO FPGA core application for input, output, inter-thread communication and control
2. Time critical loop for floating point control, signal processing, analysis and point-by-point decision making
3. Normal priority loop for embedded data logging, remote panel interfaces and Ethernet/serial communication
4. Networked host PC for remote graphical user interface, historical data logging and postprocessing

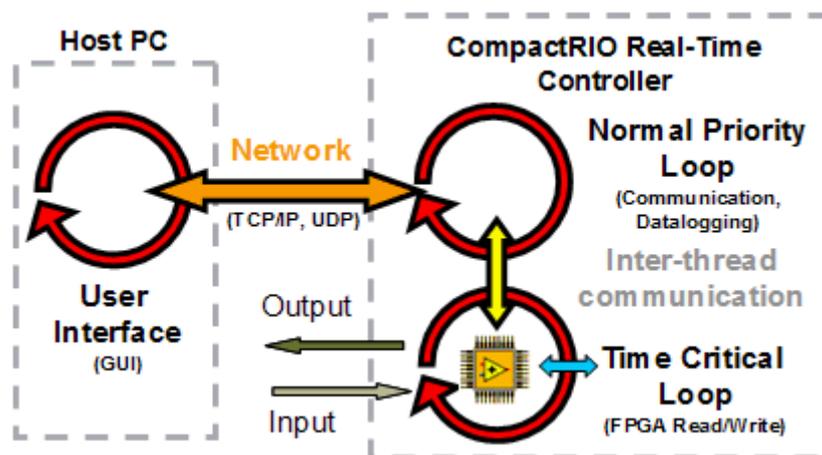


Figure 5: CRIO Programming Structure

Depending on the application requirements, it's possible to implement one or all of these application components.

The onboard software currently implemented on the flight computer follows this standard approach.

2.2 Sensors

If a UAV has to fly autonomously or needs stability augmentation in remote controlled flight, its flight control algorithms need information about its state, which can be obtained by means of onboard sensors. Depending on the vehicle type and its mission, sensors can be different. For the purpose of this work, sensor types have been split into Attitude Heading and Reference System (AHRS) and altitude sensors.

2.3 Attitude Heading and Reference System (AHRS)

Most common attitude sensors are based on gyros that can be either mechanical, piezoelectric or optical. A three axis gyro platform measures angular rates along all axes of the vehicle and is usually contained in an Inertial Measurement Unit (IMU) which also provides data from accelerometers. Magnetometers are also used to determine heading of the air vehicle by measuring the Earth magnetic field. Attitude and position can be then calculated in a state estimator by integrating IMU measurements. However the high accuracy, simplicity and availability of the Global Positioning System (GPS) makes it the emerging standard positioning system for UAVs as well as for general and commercial aviation. Depending on the quality of the GPS receiver, the achievable accuracy and the GPS update rate varies. Since common GPS update rate is usually once a second, this can result in a limited bandwidth of the UAV controller. A common way of solving that problem is to fuse data from all the flight sensors into a navigation filter in a state estimator. In addition altitude data (coming from a radar or sonar altimeter) and magnetometers measurements can be also used to improve the navigation filter. Usually an extended Kalman filter approach is used to integrate data from all the navigation sensors.

Unibo, using rapid prototyping approach, has adopted a complete AHRS like the CrossBow NAV 420, which was chosen as navigation platform for the purpose of this work. This kind of unit is able to directly deliver vehicle attitude, GPS velocity and position data, acceleration and rates at a rate up to 100 Hz, thanks to a high performance Kalman filter algorithm implemented on an internal digital signal processing module. Velocity data includes aiding from the inertial instruments such reducing the latency associated with stand-alone GPS measurements.

Particularly, the NAV 420 uses the latest in solid-states sensor technology and consists of the following subsystems (see figure 6):

- 1) Inertial Sensor Array: This is an assembly of three accelerometers, three gyros (rate sensors) and four temperature sensors.
- 2) A three axis fluxgate magnetometer board used to compute heading.
- 3) A WAAS capable GPS receiver for position and velocity measurement.
- 4) A digital signal processing (DSP) module, which receives the signals from the inertial sensors and magnetometers. This unit converts the signals to digital data, filters the data, computes the attitude solution, monitors and processes all BIT data, and transmits the results to the user.

The NAV420 analog sensor signals are sampled and converted to digital data at 1 kHz. The sensor data is filtered and down-sampled by a DSP.

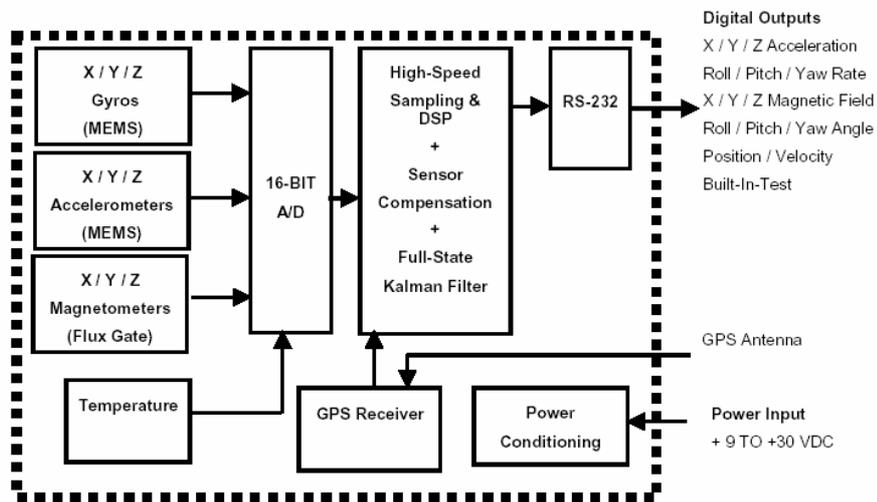


Figure 6: NAV420CA System Architecture

The choice of this kind of platform significantly reduced development time in signal processing and sensor fusion, greatly improves measurement reliability and guarantees sensor stability and performance in a high vibration operating environment, like the one of a small rotary wing platform.

Preliminary flight tests have showed it adequate controllability and robustness for the maneuvers required in hover-like flight. [8, 9]

3 Development of Dynamic Model Structure

The foundations for the development of the linear parameterized model are equations of motion derived from the first principle [1]

Using system identification, we want to achieve the best possible fit of the flight-data with a model that is consistent with the physical knowledge and intuition. The first part of the problem consists of the derivation of the dynamic equations that will define the state-space model with the unknown parameters. Once accomplished, the parameters of the model can be identified. Based on the results obtained, the model structure will be refined until satisfactory results are achieved

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 and longitudinal flapping dynamics and the rotor speed dynamics [1,11].

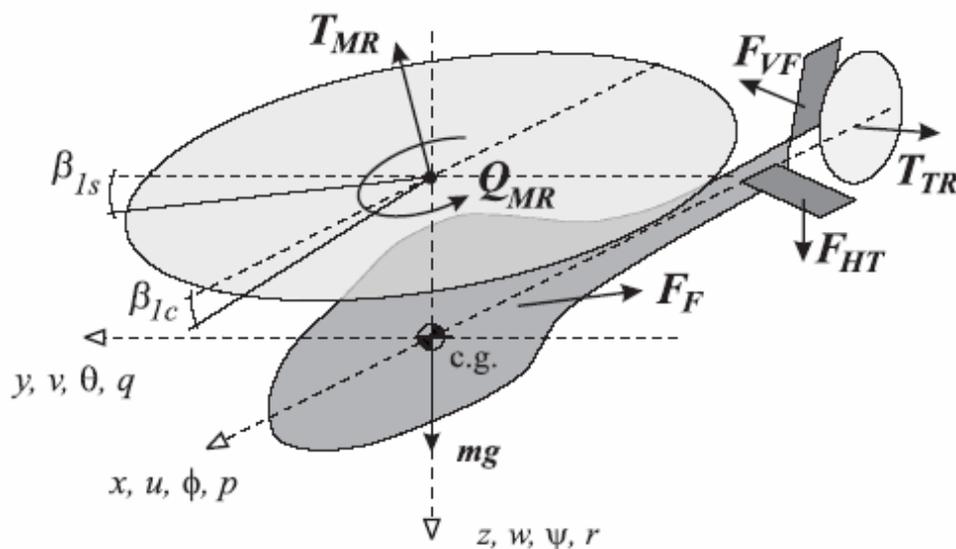


Figure 7: AV reference frame, forces & moments

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.

$$\begin{aligned}
\dot{u} &= vr - wq - g \sin \theta + (X_{mr} + X_{fus})/m \\
\dot{v} &= wp - ur + g \sin \phi \cos \theta + (Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf})/m \\
\dot{w} &= uq - vp + g \cos \phi \cos \theta + (Z_{mr} + Z_{fus} + Z_{ht})/m \\
\dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + (L_{mr} + L_{vf} + L_{tr})/I_{xx} \\
\dot{q} &= pr(I_{zz} - I_{xx})/I_{yy} + (M_{mr} + M_{ht})/I_{yy} \\
\dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + (-Q_e + N_{vf} + N_{tr})/I_{zz}
\end{aligned} \tag{1}$$

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 8 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 Rotor/Stabilizer-Bar Dynamics

The simplest way to represent the rotor dynamics is as a rigid disc which can tilt about the longitudinal and lateral axis. The resulting rotor equations of motions are two first order differential equations, for the lateral and longitudinal flapping:

$$\begin{aligned}
\tau_f \dot{b} &= -b - \tau_f p + B_a a + B_{lat} \delta_{lat} + B_{lon} \delta_{lon} \\
\tau_f \dot{a} &= -a - \tau_f q + A_b b + A_{lat} \delta_{lat} + A_{lon} \delta_{lon}
\end{aligned} \tag{2}$$

(3)

Mettler and Al, in the initial application of system identification to the modelling of the R-50 [12], were treating the rotor/stabilizer bar as a lumped system. The resulting model was accurate. However because the stabilizer bar has a major influence on the helicopter's flight-dynamic characteristics, they decided to explicitly model the stabilizer bar system. This

allowed better study of the effects of the stabilizer bar during flight control design or handling quality evaluations.

The stabilizer bar can be regarded as a secondary rotor, attached to the rotor shaft above the main rotor, through an unrestrained teetering hinge. The blades consist of two simple paddles. The stabilizer bar receives cyclic inputs from the swash-plate in a similar way as the main blades. Because of the teetering hinge and the absence of restraint, the stabilizer bar is virtually not subject to cross axis effects (the stabilizer bar restoring forces are entirely centrifugal, resulting in a resonant frequency for the flapping motion which is identical to the rotor rotation speed. Therefore, independently of the amount of damping in the system, the phase lag between the control input and the dynamic response is exactly 90°). In general it can be written the lateral (d) and longitudinal (c) stabilizer bar dynamic equations using the same equations as for the single rotor system but in an uncoupled form:

$$\begin{aligned}\tau_s \dot{d} &= -d - \tau_s p + D_{lat} \delta_{lat} \\ \tau_s \dot{c} &= -c - \tau_s q + C_{lon} \delta_{lon}\end{aligned}\tag{4}$$

Where D_{lat} and C_{lon} are the input derivatives, and τ_s is the stabilizer bar's time constant, which is a function of the paddle lock number γ_s and the rotor speed Ω .

The stabilizer bar does not exert any forces or moments on the shaft. The bar dynamics are coupled to the main rotor via the bell mixer. The bell mixer is a mechanical mixer, which superposes a cyclic command proportional to the amount of stabilizer bar flapping to the cyclic commands coming from the swash-plate. The resulting augmented lateral and longitudinal main rotor cyclic commands can be written as:

$$\bar{\delta}_{lat} = \delta_{lat} + K_d d \quad \text{and} \quad \bar{\delta}_{lon} = \delta_{lon} + K_c c\tag{5}$$

The gains K_d and K_c are the stabilizer bar gearing, which are functions of the geometry of the bell mixer. Applying the Laplace transformation to the stabilizer bar lateral flapping equations (Eq. 4) we obtain:

$$d = \frac{-\tau_s}{\tau_s s + 1} p + \frac{D_{lat}}{\tau_s s + 1} \delta_{lat}\tag{6}$$

which shows that the stabilizer bar does indeed act as a lagged rate feedback.

Using the same tip-path plane model formulation for the single rotor flapping equations, and introducing the augmented cyclic commands gives:

$$\begin{aligned}\tau_f \dot{b} &= -b - \tau_f p + B_a a + B_{lat}(\delta_{lat} + K_d d) + B_{lon} \delta_{lon} \\ \tau_f \dot{a} &= -a - \tau_f q + A_b b + A_{lat}(\delta_{lat} + K_c c) + A_{lon} \delta_{lon}\end{aligned}\quad (7)$$

where B_{lat} , B_{lon} and A_{lon} , A_{lat} are the input derivatives, τ_f is the main rotor time constant, which is a function of the main blade lock number γ and the rotor speed Ω . B_a and A_b account for the cross-coupling effects occurring at the level of the rotor itself.

In the final state-space model, the control augmentation is determined through the system's states. Therefore, we need to define the derivatives: $B_d = B_{lat}K_d$ and $A_c = A_{lon}K_c$. The relation between the derivatives and the gearing of the bell-mixer are:

$$K_d = \frac{B_d}{B_{lat}} \quad \text{and} \quad K_c = \frac{A_c}{A_{lon}} \quad (8)$$

In reality, since the bell-mixer operates the same way independently of the rotor azimuth, the gearing is the same for both axes. The gearing value was determined experimentally. This relation of Eq. 8 could be used as a constraint between the derivatives B_{lat} and B_d (A_{lon} and A_c) to reduce the number of unknown parameters. However, since we were not certain about our approach to the modelling of the stabilizer bar, we decided to leave them free (we will compare the identified value to the value obtained experimentally).

3.3 Heave Dynamics

The frequency response of the vertical acceleration to collective shows [12] that a first order system should adequately capture the heave dynamics. This agrees with the rigid body equations from the Newton-Euler equations:

$$\dot{w} = (-v_0 p + u_0 q) + Z_w w + Z_{col} \delta_{col} \quad (9)$$

The term in parenthesis corresponds to the centrifugal forces that are relevant exclusively for the cruise conditions.

3.4 Yaw Dynamics

The yaw dynamics of the bare helicopter airframe can usually be modelled as the simple first order system:

$$\frac{r}{\delta_{ped}} = \frac{N_{ped}}{s - N_r} \quad (10)$$

where N_r is the bare airframe yaw damping coefficient and N_{ped} is the sensitivity to the pedal control.

3.5 Unibo RUAW dynamics linear model

The system presented in this paragraph describes the model used for the identification tests that will be presented in chapter 6. The equations of this system present both the On-Axis parameters values and new Off-axis derivatives (see RUAV Dynamic model Identification chapter) and can be written as follows:

$$\begin{aligned}
 \dot{u} &= X_u \cdot u - g \cdot \theta + X_a \cdot a \\
 \dot{v} &= X_v \cdot v + g \cdot \phi + Y_b \cdot b \\
 \dot{p} &= L_b \cdot b + L_a \cdot a + L_u \cdot u + L_v \cdot v \\
 \dot{q} &= M_a \cdot a + M_b \cdot b + M_u \cdot u + M_v \cdot v + M_{coll} \cdot \delta coll \\
 \dot{\phi} &= p \\
 \dot{\theta} &= q \\
 \dot{a} &= -q - \frac{a}{\tau_f} + \frac{A_b}{\tau_f} \cdot b + \frac{A_{lon}}{\tau_f} \cdot \delta lon + \frac{A_{lat}}{\tau_f} \cdot \delta lat \\
 \dot{b} &= -p - \frac{b}{\tau_f} + \frac{B_a}{\tau_f} \cdot a + \frac{B_{lat}}{\tau_f} \cdot \delta lat + \frac{B_{lon}}{\tau_f} \cdot \delta lon \\
 \dot{w} &= Z_w \cdot w + Z_{coll} \cdot \delta coll
 \end{aligned} \tag{11}$$

This system is similar to the one used by Mettler in [1,11,12] and the hypothesis that lead to the linearization can be found in [1]. This model differs from the ones reported in literature for the absence of pedal input and yaw dynamic: in small scale helicopters, the cross effects due to yaw are close to zero and therefore often negligible (values equal to zero in [6]). Moreover here the stabilizer bar has not been explicitly modeled as the results of identification (see chapter 6) were found to be enough accurate. As in [12], M_{coll} derivative have been added in B matrix to account also cross-effect of collective input into helicopter longitudinal dynamic.

The system can be written in the State Space form as:

$$\dot{x} = A \cdot x + B \cdot u \tag{12}$$

$$y = C \cdot x + D \cdot u \tag{13}$$

where the input command vector is:
(14)

$$u = [\delta lon, \delta lat, \delta coll]^T$$

the state vector is:

$$x = [u, v, p, q, \phi, \theta, a, b, w]^T$$

the output vector is:
(15)

$$y = [u, v, p, q, \phi, \theta, w]^T$$

and the matrixes are:

$$A = \begin{bmatrix} Xu & 0 & 0 & 0 & 0 & -g & Xa & 0 & 0 \\ 0 & Yv & 0 & 0 & g & 0 & 0 & Yb & 0 \\ Lu & Lv & 0 & 0 & 0 & 0 & La & Lb & 0 \\ Mu & Mv & 0 & 0 & 0 & 0 & Ma & Mb & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1/\tau_f & Ab/\tau_f & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & Ba/\tau_f & -1/\tau_f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Zw \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & Mcoll \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ Alon/\tau_f & Alat/\tau_f & 0 \\ Blon/\tau_f & Blat/\tau_f & 0 \\ 0 & 0 & Zcoll \end{bmatrix} \quad (16)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4 Baseline and FF controllers architectures

In both the Baseline and in FF controllers, two independent SISO control systems are used for the lateral and longitudinal dynamics control. The adopted SISO controllers are based on PID regulators and have a nested structure: the outer control loop uses helicopter longitudinal and lateral velocities errors to produce the attitude reference $(\theta_{ref}, \phi_{ref})$ for the inner attitude loop and the inner attitude loop computes commands $(\delta_{lon}, \delta_{lat})$ using outer loop's attitude references $(\theta_{ref}, \phi_{ref})$ and measured helicopter attitude feedbacks (θ, ϕ) . The vertical position and the heading are controlled by other two separate PID single loop controllers.

4.1 Baseline controller description

Baseline controller for longitudinal and lateral dynamics consists in a SISO PID control with a two levels nested loop structure (see fig. 8). Lateral and longitudinal track velocities errors are used in the CV blocksets to generate respectively demands for the roll (ϕ) and the pitch (θ) attitude control module (CA blocksets),

A block diagram of a SISO controller for the Baseline control system is represented in Fig. 9.

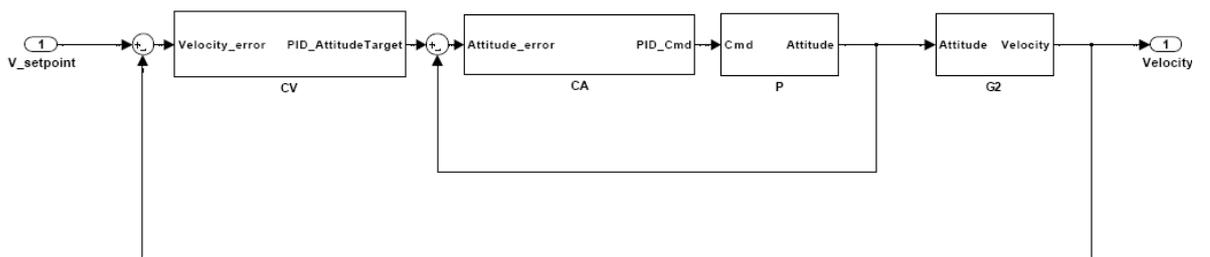


Figure 8: Block diagram of the Baseline controller for longitudinal or lateral dynamics

In previous schematic, CV is the outer Velocity loop PID blockset described by the following transfer function:

$$CV(s) = \frac{K_{dv} \cdot s^2 + K_{pv} \cdot s + K_{iv}}{s} \quad \text{-- PID Velocity Controller in Baseline control system (17)}$$

where K_{dv} is the derivative term gain, while K_{pv} and K_{iv} are respectively the proportional and the intergral terms gains. Again, in Fig.9, the CA blockset describes the inner Attitude loop PID implementing the following transfer function:

$$CA(s) = \frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \quad \text{– PID Attitude Controller in Baseline control system (18)}$$

where K_d is the derivative term gain, while K_p and K_i are again respectively the proportional and the intergral terms gains. The blockset P represents the transfer function of the command-to-attitude plant identified dynamic that is enclose in the State Space model (see cap.4 about Identification), whereas G_2 is the attitude-to-velocity transfer function:

$$P(s) = \frac{\theta(s)}{\delta_{lon}(s)}, \quad P(s) = \frac{\phi(s)}{\delta_{lat}(s)} \quad \text{– Longitudinal, Lateral command-attitude TF, from SS model (19)}$$

$$G_2(s) = \frac{\pi}{180} \cdot \frac{u(s)}{\theta(s)}, \quad G_2(s) = \frac{\pi}{180} \cdot \frac{v(s)}{\phi(s)} \quad \text{– Longitudinal, Lateral attitude-velocity TF, derived from SS model (20)}$$

4.2 Feed-Forward controller description

The proposed FF control architecture (see fig. 9) is very simple and it is based on SISO nested control loops.

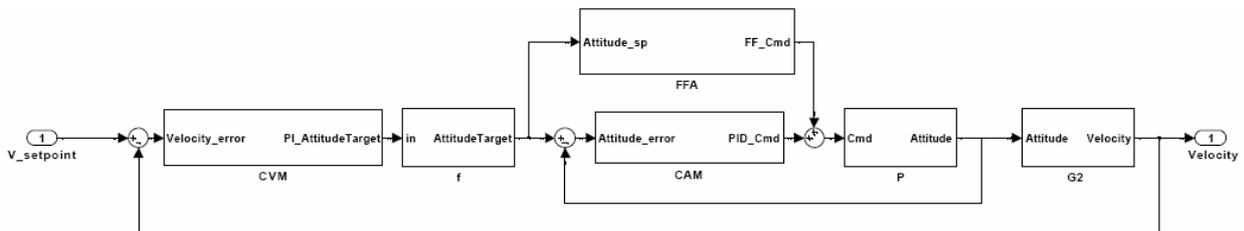


Figure 9: Block diagram of the FF controller for longitudinal or lateral dynamics

Once again it consists of a velocity outer loop (*CVM* blockset) based on a PI controller with no Derivative term:

$$CVM(s) = \frac{K_{pvm} \cdot s + K_{ivm}}{s} \quad \text{– PI Velocity controller in Feedforward control system (21)}$$

where K_{pvm} and K_{ivm} are again respectively the proportional and the intergral terms gains
(22)

The inner loop is, instead, composed by a FeedForward term + Proportional and Integral controller (*FFA+CAM* blockset): the attitude feedforward term (*FFA*) is obtained by the inversion of the attitude identified linear model, as described in the next paragraph:

$$FFA = \frac{\delta_{lon}}{\theta} = \frac{-\tau_e}{A_{lon}} \cdot \frac{s^3 + 1/\tau_e \cdot s^2 + \omega_{nq}^2 \cdot s}{\omega_{nq}^2} \quad \text{– Feedforward action, Longitudinal (23)}$$

$$CAM(s) = \frac{K_{pm} \cdot s + K_{im}}{s} \quad \text{– PI Attitude controller in Feedforward control system (24)}$$

The output of the PI velocity controller (*CMV* blockset) is filtered by means of a first order filter (*f* blockset) with dedicated time constants for longitudinal and lateral controllers (*Tfilt*, *TfiltPhi*).

$$f = \frac{1}{1 + T_{filt} \cdot s} \quad \text{– first order filter in Feedforward control system (25)}$$

In both cases, the discrete sampling time has been fixed in 0.020 seconds both for lateral and longitudinal dynamics.

4.3 Transfer Functions

The analysis of structures of Baseline and Feed-Forward controllers requires the computation of following transfer functions, that will be used in next chapters to compute Phase and Gain margins and to perform stability assessment.

$$BLG = CV(s) \cdot G_1(s) \cdot G_2(s) \quad \text{– Baseline Loop Gain (BLG)Transfer Function (TF) (26)}$$

$$CA(s) = \frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \quad \text{– PID Attitude Controller in Baseline control system (27)}$$

$$CAM(s) = \frac{K_{pm} \cdot s + K_{im}}{s} \quad \text{– PI Attitude controller in Feedforward control system (28)}$$

$$CV(s) = \frac{K_{dv} \cdot s^2 + K_{pv} \cdot s + K_{iv}}{s} \quad \text{– PID Velocity Controller in Baseline control system} \quad (29)$$

$$CVM(s) = \frac{K_{pvm} \cdot s + K_{ivm}}{s} \quad \text{– PI Velocity controller in Feedforward control system} \quad (30)$$

$$f = \frac{1}{1 + T_{fil} \cdot s} \quad \text{– first order filter in Feedforward control system} \quad (31)$$

$$FFA = \frac{\delta_{lon}}{\theta} = \frac{-\tau_e \cdot s^3 + 1/\tau_e \cdot s^2 + \omega_{nq}^2 \cdot s}{A_{lon} \omega_{nq}^2} \quad \text{– Feedforward action, Longitudinal} \quad (32)$$

$$FFA = \frac{\delta_{lat}}{\phi} = \frac{\tau_e \cdot s^3 + 1/\tau_e \cdot s^2 + \omega_{np}^2 \cdot s}{B_{lat} \omega_{np}^2} \quad \text{– Feedforward action, Lateral} \quad (33)$$

$$FLG(s) = G_{ff}(s) \cdot CVM(s) \quad \text{– FF Loop Gain} \quad (34)$$

$$FTF(s) = \frac{FLG(s)}{1 + FLG(s)} \quad \text{– FF Transfer Function} \quad (35)$$

$$G_1(s) = \frac{CA(s) \cdot P(s)}{CA(s) \cdot P(s) + 1} \quad (36)$$

$$G_2(s) = \frac{\pi}{180} \cdot \frac{u(s)}{\theta(s)}, \quad G_2(s) = \frac{\pi}{180} \cdot \frac{v(s)}{\phi(s)} \quad \text{– Longitudinal, Lateral attitude-velocity TF, derived from SS model} \quad (37)$$

$$G_{ff}(s) = G_{ffl}(s) \cdot G_2(s) \quad (38)$$

$$G_{ffl}(s) = f \cdot \frac{CAM(s) \cdot P(s) + FFA(s) \cdot P(s)}{CAM(s) \cdot P(s) + 1} \quad (39)$$

$$P(s) = \frac{\theta(s)}{\delta_{lon}(s)}, \quad P(s) = \frac{\phi(s)}{\delta_{lat}(s)} \quad \text{– Longitudinal, Lateral command-attitude TF, from SS model} \quad (40)$$

4.4 Vertical Down Velocity Controller

Both in FF and Baseline control architectures Vertical down velocity (Vd) control is implemented using a simple PI module [9]; its generic structure is shown in Figure 10.

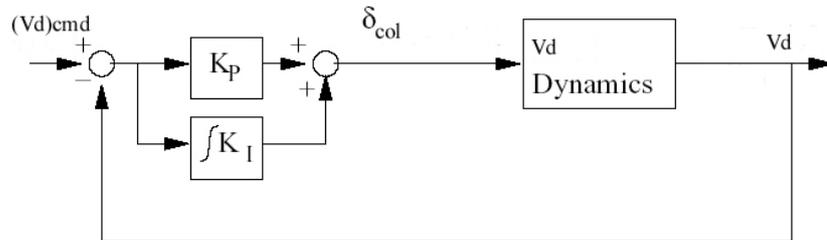


Figure 10: Vertical down velocity control module

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

We have found:

- $K_{pvd} = -1.1 \text{ rad}/(\text{m/s})$
- $K_{ivd} = -0.09 \text{ rad/m}$

The Vd module tracking performance are shown in Figure 11

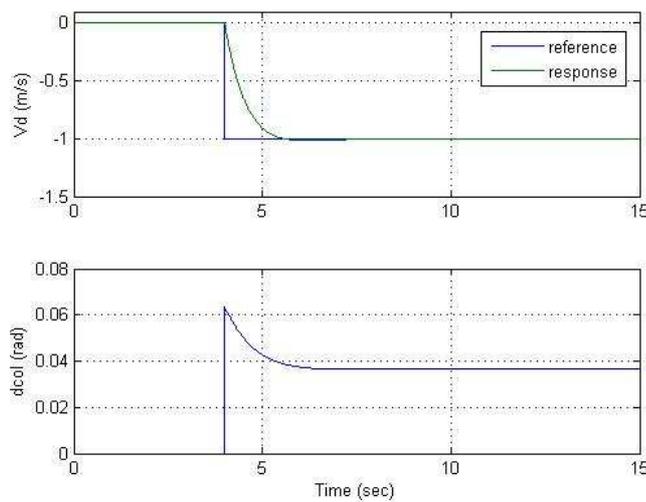


Figure 11: Vertical down velocity controller tracking performance

4.5 Heading Controller

Heading control is achieved using the onboard gyro AVCS (Angular Velocity Control System). Therefore, the implemented algorithm [8] 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 12

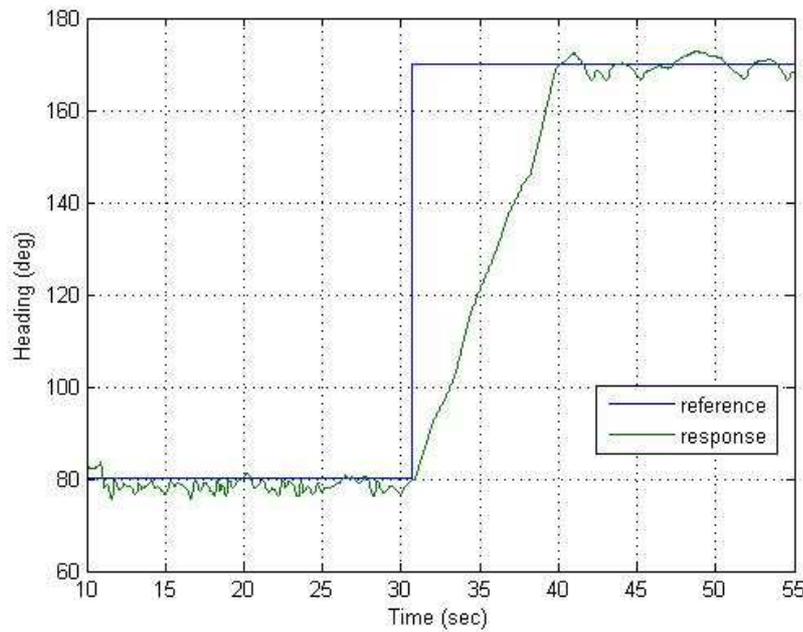


Figure 12: Heading controller tracking performance

4.6 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 [13], maintaining commanded rotorspeed by changing the throttle:

$$\delta_i = K_p \cdot \omega_i + K_i \cdot \dot{\omega}_i$$

$$\omega_i = \Omega_c - \Omega$$

(41)

where Ω_c is the rotorspeed command, 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 [14] and fine tuning we have found:

$$K_p = 0.1$$

$$K_i = 0.02$$

Figure 13 shows the throttle response and the rotorspeed tracking (96.3 rad/s) after a commanded climb at 2 m/s.

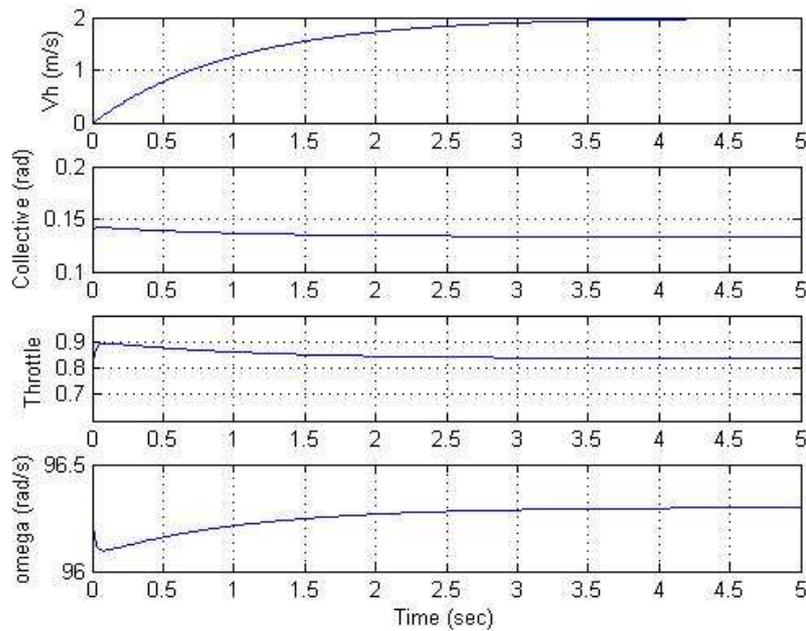


Figure 13: Engine governor tracking performance

4.7 Discrete-time implementation of Baseline controller

Regarding Baseline control system, in the case of the longitudinal, for example, the attitude PID controller, it can be written in continuous domain:

$$CA(s) = \frac{PID}{E} = K_d \cdot s + K_p + \frac{K_i}{s} = \frac{K_d \cdot s^2 + K_p \cdot s + K_i}{s} \quad (42)$$

where $E(s)$ is the transfer function of the error (difference between set-point and measured attitude) and where $PID(s)$ has been written using following constants: $K_d = K_p \cdot T_d$ and

$$K_i = \frac{K_p}{T_i}$$

that links the PID proposed representation to a more common one [15].

Discrete time domain implementation of PID controller has been done using Backward approximation through which, representation of first derivative is done using finite difference:

$$\frac{dx(t)}{dt} \cong \frac{x(t) - x(t - T_s)}{T_s} \quad (43)$$

Moreover, remembering that in digital controls domain operator z^{-1} is used to compute the 'old' value (value of variable computed or sampled at previous sampling task, $x(t - T_s) = x_{k-1} = x_k \cdot z^{-1}$), using z-transform representation, discrete PID can be written in the form:

$$CA(z) = \frac{PID}{E} = \frac{(K_p \cdot T_s + K_i \cdot T_s^2 + K_d) \cdot z^2 + (-K_p \cdot T_s - 2 \cdot K_d) \cdot z + K_d}{T_s \cdot z^2 - T_s \cdot z} \quad (44)$$

The same backward approximation has been used for the implementation of all the 5 PID controllers that are present in Baseline control architecture.

Following schematic shows the implementation in Simulink of the PID discrete controller:

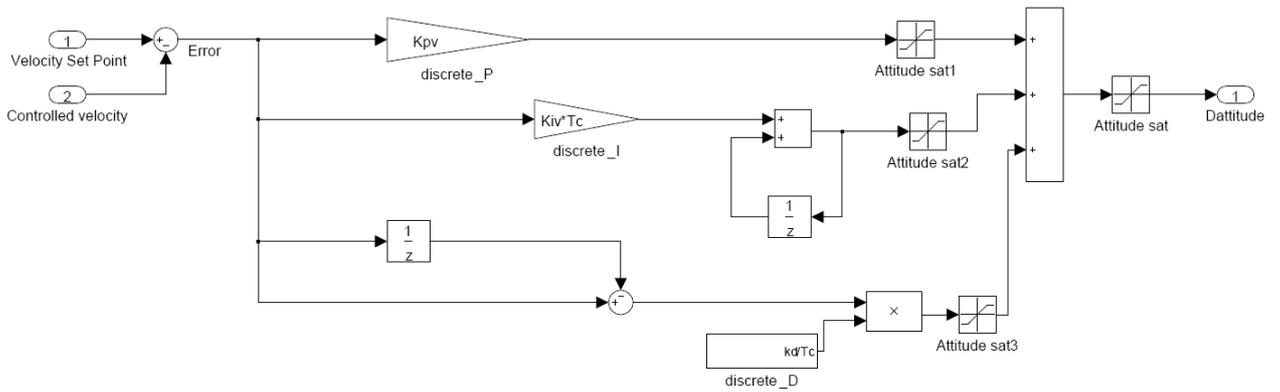


Figure 14: Schematic of the inner loop Attitude PID controller implementation

It can be noticed that each of the contributions of the PID, as well as the total sum of the three terms, have been saturated between a minimum and a maximum value that has been fixed in 40 deg.

As for the inner Attitude controller (CA), the Velocity controller CV can be written in continuous domain:

$$CV(s) = \frac{PID}{E} = K_{dv} \cdot s + K_{pv} + \frac{K_{vi}}{s} = \frac{K_{dv} \cdot s^2 + K_{pv} \cdot s + K_{iv}}{s} \quad (45)$$

whereas using z-transform representation can be written in the form:

$$CV(z) = \frac{PID}{E} = \frac{(K_{pv} \cdot T_s + K_{iv} \cdot T_s^2 + K_{dv}) \cdot z^2 + (-K_{pv} \cdot T_s - 2 \cdot K_{dv}) \cdot z + K_{dv}}{T_s \cdot z^2 - T_s \cdot z} \quad (46)$$

Also in this case each of the contributions of the PID, as well as the total sum of the three terms, have been saturated between a minimum and a maximum value that has been fixed in 27 deg that is the maximum allowable command..

4.8 Discrete-time implementation of FF+PI control model

In *FF+PI* controller, the feed forward attitude (FFA) term, as already mentioned, can be obtained by inverting the plant identified command-to-attitude transfer function. Neglecting cross-effects, this transfer function is a third order system formed by the product of a pure integrator term and a second order transfer function [1]:

$$\frac{\theta}{\delta_{lon}} = \frac{-A_{lon}}{\tau_e} \cdot \frac{1}{s} \cdot \frac{\omega_{nq}^2}{s^2 + s/\tau_e + \omega_{nq}^2}; \quad \frac{\phi}{\delta_{lat}} = \frac{B_{lat}}{\tau_e} \cdot \frac{1}{s} \cdot \frac{\omega_{np}^2}{s^2 + s/\tau_e + \omega_{np}^2} \quad (47)$$

A unique formulation, valid for longitudinal and lateral dynamics, can be expressed by:

$$P(s) = \frac{1}{s} \cdot \frac{e \cdot \omega_n^2}{s^2 + 2 \cdot \delta \cdot s + \omega_n^2} \quad (48)$$

where delta δ is the damping coefficient, ω_n is the natural frequency and e is the gain of the second order transfer function.

Comparing relations (1) and (2), for longitudinal dynamic following relations can be written:

$$e = -\frac{A_{lon}}{\tau_e} \quad \delta = \frac{1}{2 \cdot \tau_e \cdot \omega_{nq}} \quad \omega_n = \omega_{nq} \quad (49)$$

whereas for lateral dynamic:

$$e = \frac{B_{lat}}{\tau_e} \quad \delta = \frac{1}{2 \cdot \tau_e \cdot \omega_{np}} \quad \omega_n = \omega_{np} \quad (50)$$

Hence in the continuous time domain, the feed forward contribute (*FFA*) will be written as the inverse of $P(s)$:

$$FFA = P(s)^{-1} = \frac{s^3 + 2 \cdot \delta \cdot s^2 + \omega_n^2 \cdot s}{e \cdot \omega_n^2} \quad (51)$$

while turning into the discrete time domain, using Backward approximation [16], the command δ_k to be actuated can be computed (k index is related to the sample time instant):

$$\delta_k = \frac{1}{e \cdot \omega_n^2} \left[\frac{1}{T_s^3} (u_k - 3 \cdot u_{k-1} + 3 \cdot u_{k-2} - u_{k-3}) + \frac{2 \cdot \delta}{T_s^2} (u_{k-1} - 2 \cdot u_{k-2} + u_{k-3}) + \frac{\omega_n^2}{T_s} (u_{k-2} - u_{k-3}) \right] \quad (52)$$

In the longitudinal control, for example, if $u_k = \theta_{ref_k}$, or, in other words, if the input u_k is the filtered attitude set-point generated by the *CVM* blockset, then eq. 52 computes the command $\delta_{long_k} = \delta_k$ that would bring the modeled system to perfectly track its original set-point θ_{ref_k} , in total absence of disturbances.

Note that, the sampling time T_s for feedforward term has been selected in 0.040 s, whereas all the other variables continue to be updated every 0.020 s. This choice has been driven by the necessity of bounding the derivatives in eq. 52 as a consequence of the relatively poor resolution of the velocity signal computed by the inertial unit Crossbow NAV42

In FF+PI controller a first order filter has been implemented in discrete time domain using following representation:

$$y_k = y_{k-1} + K_{filt} \cdot (u_k - y_{k-1}) \quad (53)$$

where u_k is a generic input signal and y_k is the filter output and K_{filt} is the discrete first order filter constant that is related to the continuous time constant T_{filt} from the following relation:

$$K_{filt} = \frac{T_s}{T_s + T_{filt}} \quad (54)$$

As already mentioned, in longitudinal FF architecture a simple PI (without derivative term) was considered instead of a complete PID controller for longitudinal and lateral controls.

This Attitude PI controller (CAM, M stands for Model-based in FF architecture) can be deduced from PID discrete form the by imposing $K_d=0$,:

$$CAM(s) = \frac{PI}{E} = K_p + \frac{K_i}{s} = \frac{K_p \cdot s + K_i}{s} \quad (55)$$

Equation 55 reports, for example, the continuous time TF of PI controller for longitudinal dynamic.

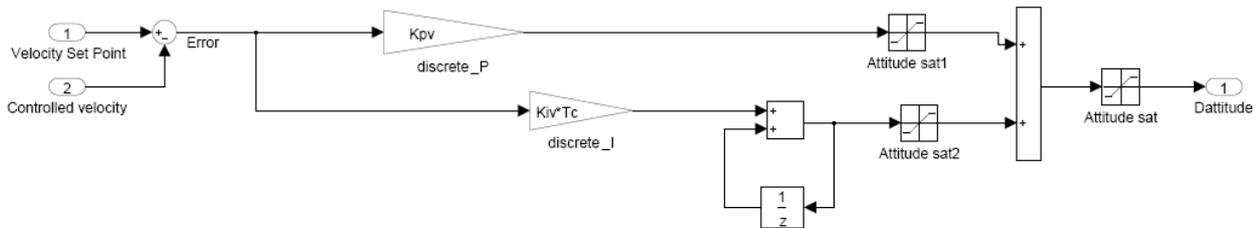


Figure 15: Schematic of the inner loop Attitude PI controller implementation

Once more, in discrete time domain, Backward approximation was used to implement the 4 different PI controllers, whose z-trasform representation is given by:

$$CAM(z) = \frac{PI}{E} = \frac{(K_{pm} \cdot T_s + K_{im} \cdot T_s^2) \cdot z + (-K_{pm} \cdot T_s)}{T_s \cdot z - T_s} \quad (56)$$

The CVM transfer function can be deduced, as in the case of the Attitude controller, from PID discrete form the by imposing $K_{dvm}=0$:

$$CVM(z) = \frac{PID}{E} = \frac{(K_{pvm} \cdot T_s + K_{ivm} \cdot T_s^2) \cdot z^2 + (-K_{pvm} \cdot T_s) \cdot z}{T_s \cdot z^2 - T_s \cdot z} \quad (57)$$

The vertical velocity PID implemented in FF+PI controller is exactly the same of the vertical velocity PID implemented in the Baseline controller.

5 Guidance

The guidance system is composed by two main parts: the lateral track control and the altitude controller. In this case the same logics have been implemented for the Baseline and the FeedForward controllers. In the following sections same

5.1 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 [14] and fine tuning we have found the values of the proportional and integral gains in:

- $K_p = 1.2$
- $K_i = 0.05$

Figure 16 shows an example of altitude tracking.

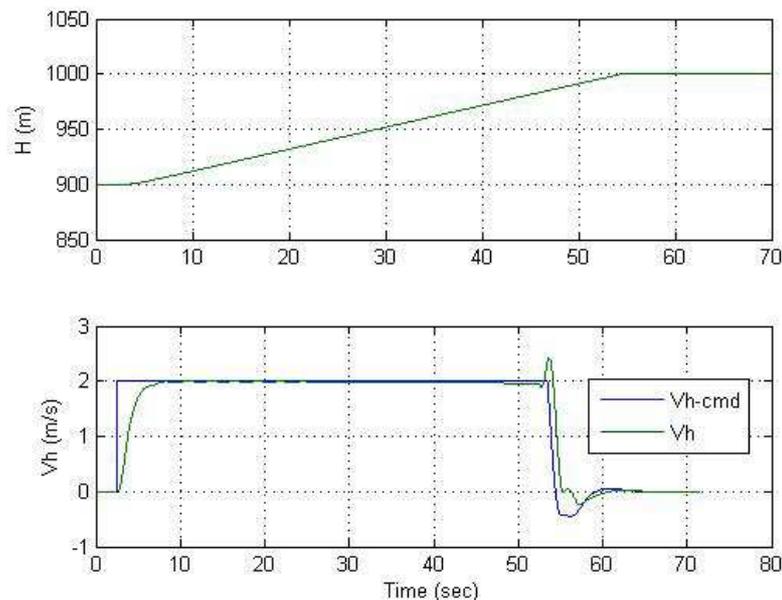


Figure 16: Altitude controller tracking performance

In this simulation, the initial helicopter altitude has been fixed in 900 m while the commanded altitude has been chosen equal to 1000 m. Therefore the altitude controller commands a vertical velocity (saturated at 2 m/s) until the reference altitude is captured. The commanded vertical velocity will be actuated as reported in paragraph 3.4.

5.2 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 [17].

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 17). The transformation will be a rotation of an angle $(\Psi_{12} - \pi/2)$ and the associated rotation matrix is given by:

$$T_{\psi} = \begin{bmatrix} \cos(\psi_{12} - \pi/2) & -\sin(\psi_{12} - \pi/2) \\ \sin(\psi_{12} - \pi/2) & \cos(\psi_{12} - \pi/2) \end{bmatrix} \quad (58)$$

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

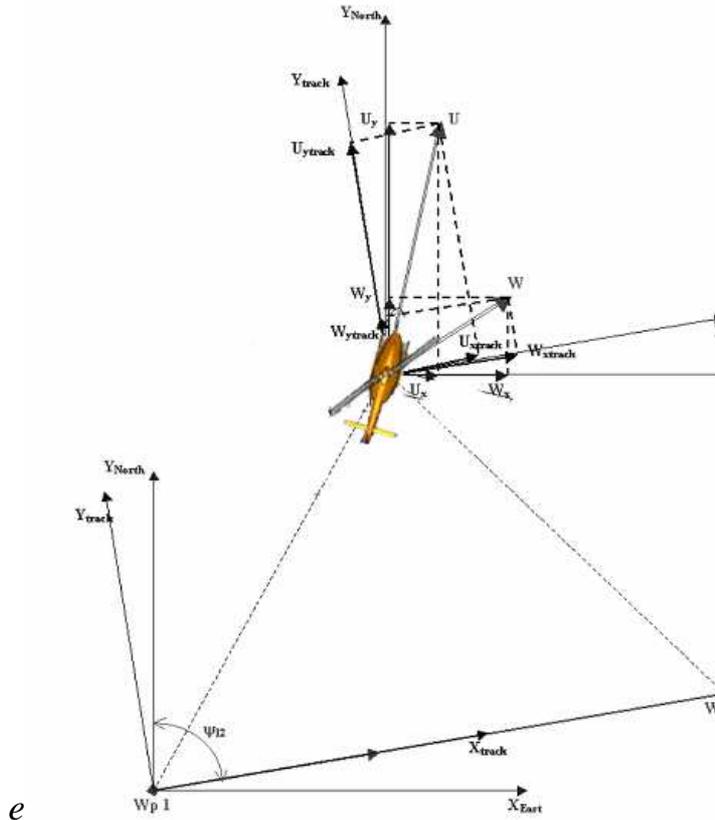


Figure 17: Track reference frame

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

$$\begin{bmatrix} \dot{X}_{track} \\ \dot{Y}_{track} \end{bmatrix} = T_\psi \begin{bmatrix} V_N \\ V_E \end{bmatrix} \quad (59)$$

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

$$\begin{bmatrix} X_{track0} \\ Y_{track0} \end{bmatrix} \quad (60)$$

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 $(1-k) \cdot X_{track}$. From the geometry of the similar triangles OAB and OCD (Figure 18), a new control strategy is proposed based on establishing the helicopter position and velocity according the following relationship:

$$\frac{\dot{X}_{track}}{kX_{track}} = \frac{\dot{Y}_{track}}{Y_{track}} \quad (61)$$

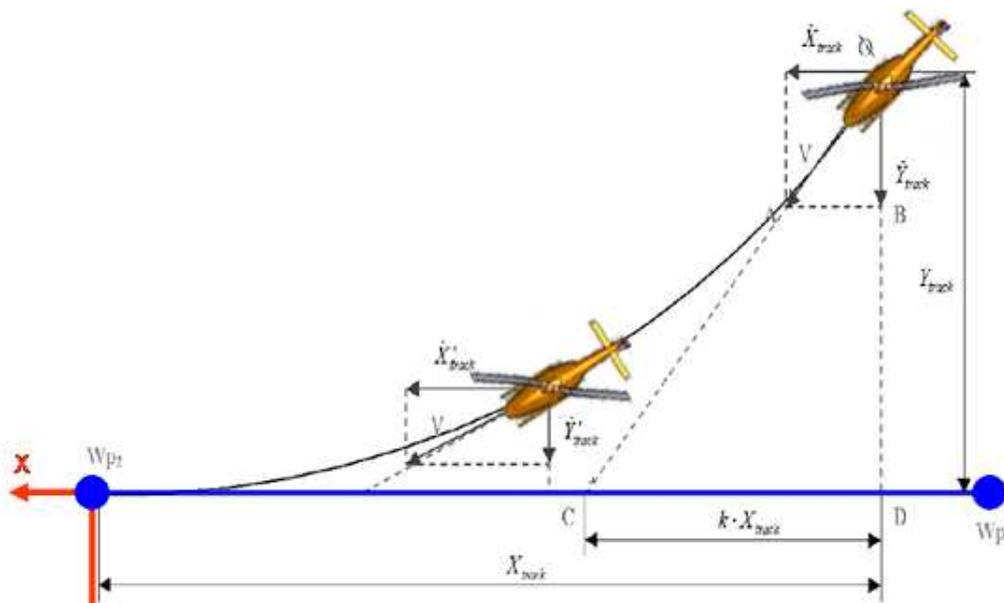


Figure 18: Guide control strategy

To achieve this objective, the error E given by:

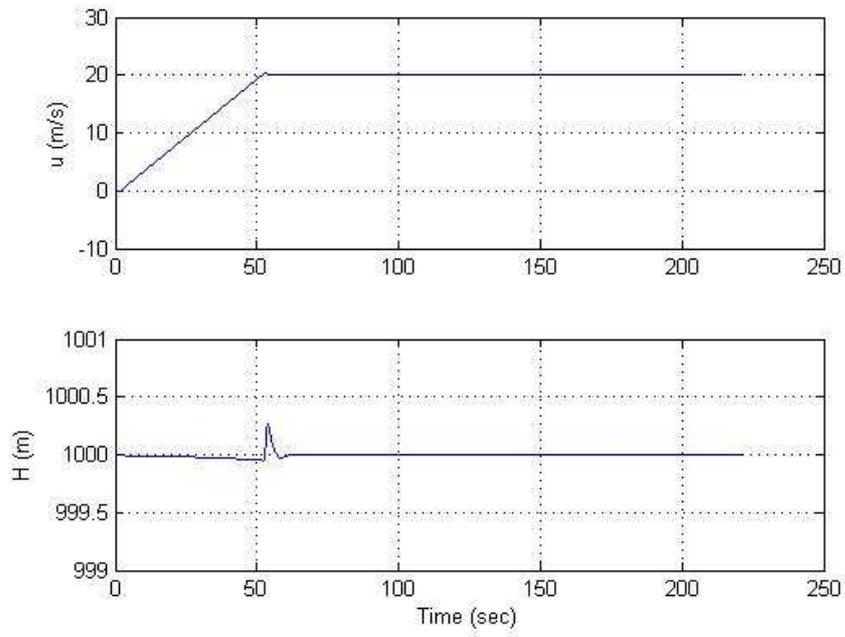


Figure 20: Guide tracking performance – Speed,Altitude

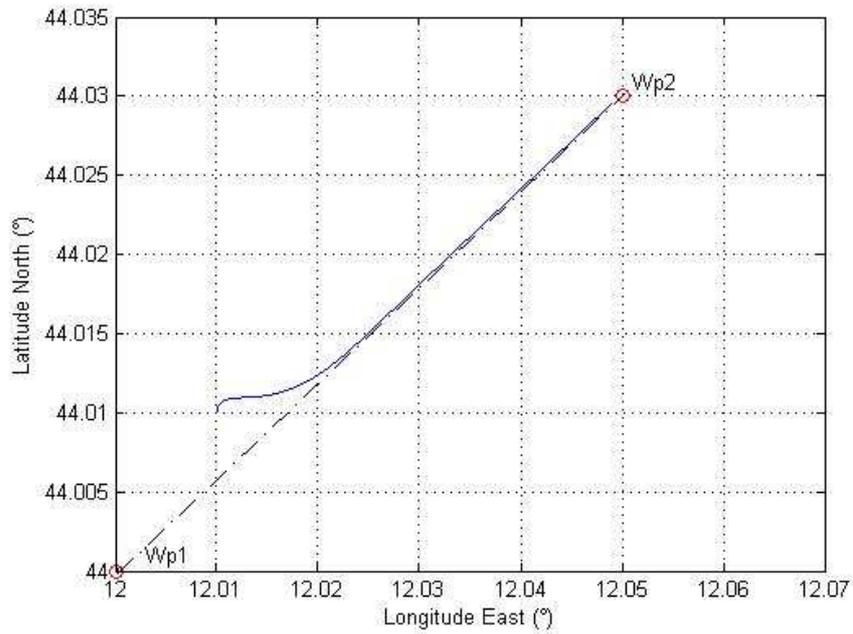


Figure 21: Guide tracking performance – Latitude,Longitude

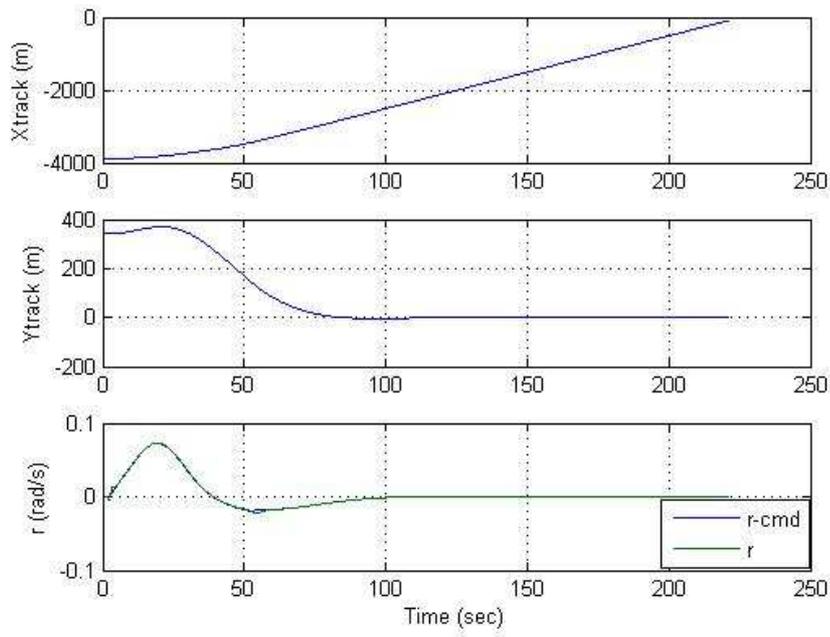


Figure 22: Guide tracking performance – X_{track} , Y_{track} , Yaw Rate

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 [17].

5.3 Fixed-Heading Navigation

The guidance system that has been implemented in the onboard computer includes also a fixed yaw navigation modality. This special navigation modality allows to set the yaw target parameter to a constant value for each track of the flight plan. For each segment of the flight plan connecting two consequent waypoints, together with the usual parameters of the waypoint (N, E, H), the yaw attitude can be set to a desired value as well.

Next table reports an example of flight plan parameters for a ‘8-like’ circuit. This particular flight plan will be used in following section to perform other comparisons between the Baseline and the FF controllers performances.

| Way point n. | Delta N | Delta E | H | Velocity |
|--------------|---------|---------|----|----------|
| 1 | 0 | 0 | 50 | 0 |
| 2 | 20 | -20 | 50 | 2 |
| 3 | 60 | 20 | 50 | 2 |
| 4 | 80 | 0 | 50 | 2 |
| 5 | 60 | -20 | 50 | 2 |
| 6 | 20 | 20 | 50 | 2 |
| 7 | 0 | 0 | 50 | 2 |

Table 1: Waypoints parameters input

The target yaw is updated each time a new waypoint is reached and it is kept constant till the next one. For example in the ‘8-like’ circuit we will have the following Target Yaws:

| Way points track | | Target Yaw |
|------------------|---|------------|
| 1 | 2 | -45 deg |
| 2 | 3 | 45 deg |
| 3 | 4 | -45 deg |
| 4 | 5 | -135 deg |
| 5 | 6 | -225 deg |
| 6 | 7 | -135 deg |
| 7 | 8 | -45 deg |

Table 2: Waypoints yaw

Next figure reports the logic implemented for the fixed yaw navigation.

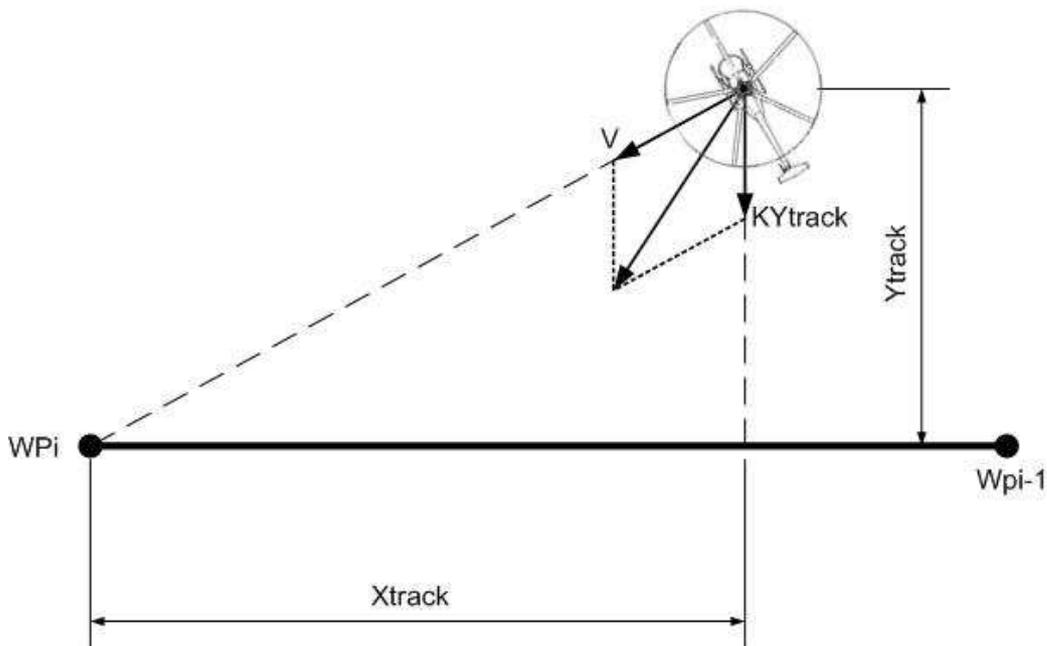


Figure 23: Target velocity computation

In this case the total velocity setpoint of the helicopter is formed by the velocity V towards the waypoint, that is taken from Table 1, together with the other flight information (Waypoints positions), and by the real-time computed Y_{track} velocity whose effect is to bring the rotorcraft trajectory as near as possible the segment connecting W_{pi} and W_{pi-1} .

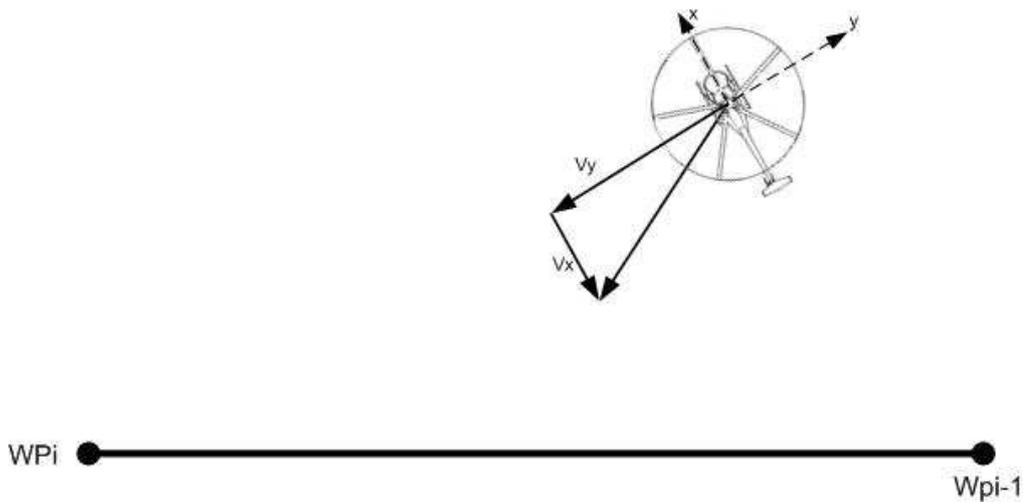


Figure 24: Target velocity computation

In a second step, as depicted in figure 24, the total velocity setpoint is decomposed in its components in the Body frame V_x and V_y . These latter velocities will be used as input target velocities by the control systems (see par 4.1 and 4.2).

5.4 Baseline controller In-Flight Autopilot Tests

In this paragraph it is shown how the onboard Baseline control software has been tested in flight trials starting from basic maneuvers towards more complicated flight plans..

The complete flight campaign for the Baseline control system has been 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 3, 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.

| 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 | -0.77366 | -0.22078 | -1.0418 | -0.11346 |
| Experimental | -1 | -1 | -1 | -1 |

Table 3: 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 | -12.89 | -4.03 | -11.43 | -3.55 | -3.622 | -4.96 |
| Experimental | -10 | -1 | -10 | -1 | -10 | -10 |

Table 4: Calculated vs Experimental velocity PI gains

More details on the parameters computation can be found to R.Pretolani's Phd Thesis [9]

- Once the attitude controllers were with fist-attempt values calibrated, the nested PI Velocity – Attitude controllers were tested. During these tests, collective and tail commands were still left to the R/C pilot for safety reasons. As shown in Table 4, 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 (or varied) 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 3 and 4, 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 25 and 26, the helicopter was able to perform autonomously and successfully the preprogrammed pattern.

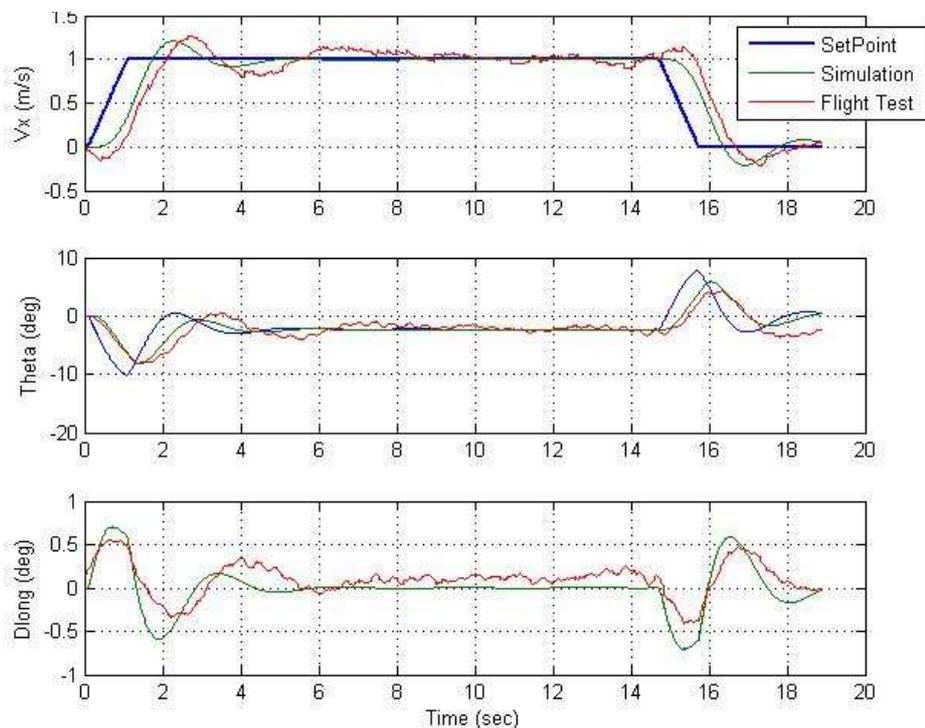


Figure 25: Simulate vs Experimental longitudinal controller tracking performance

As an example Figure 26 shows good V_x and θ tracking performance and good agreement between simulation and real flight tests.

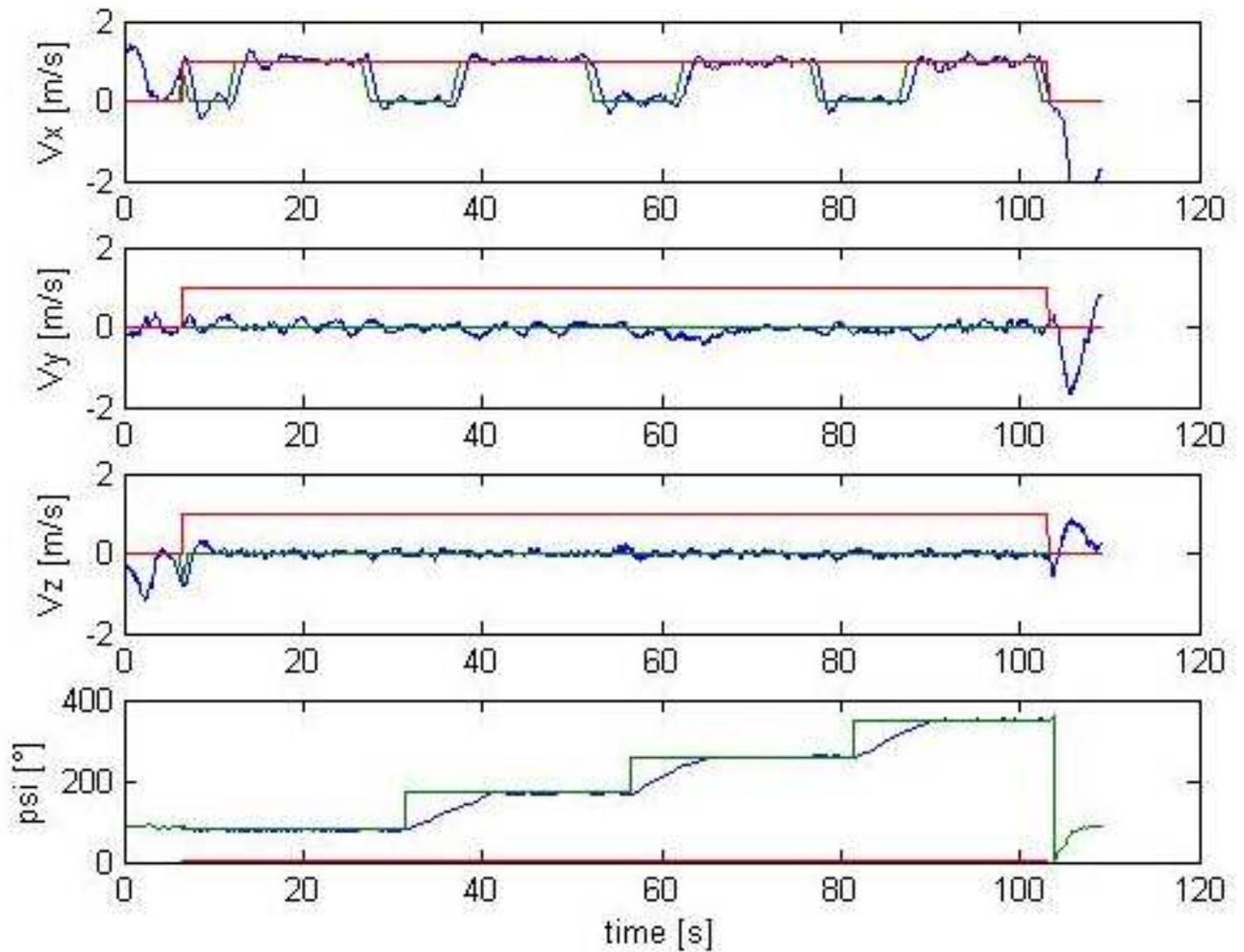


Figure 26: Recorded data during autonomous square pattern

In Figure 27:

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

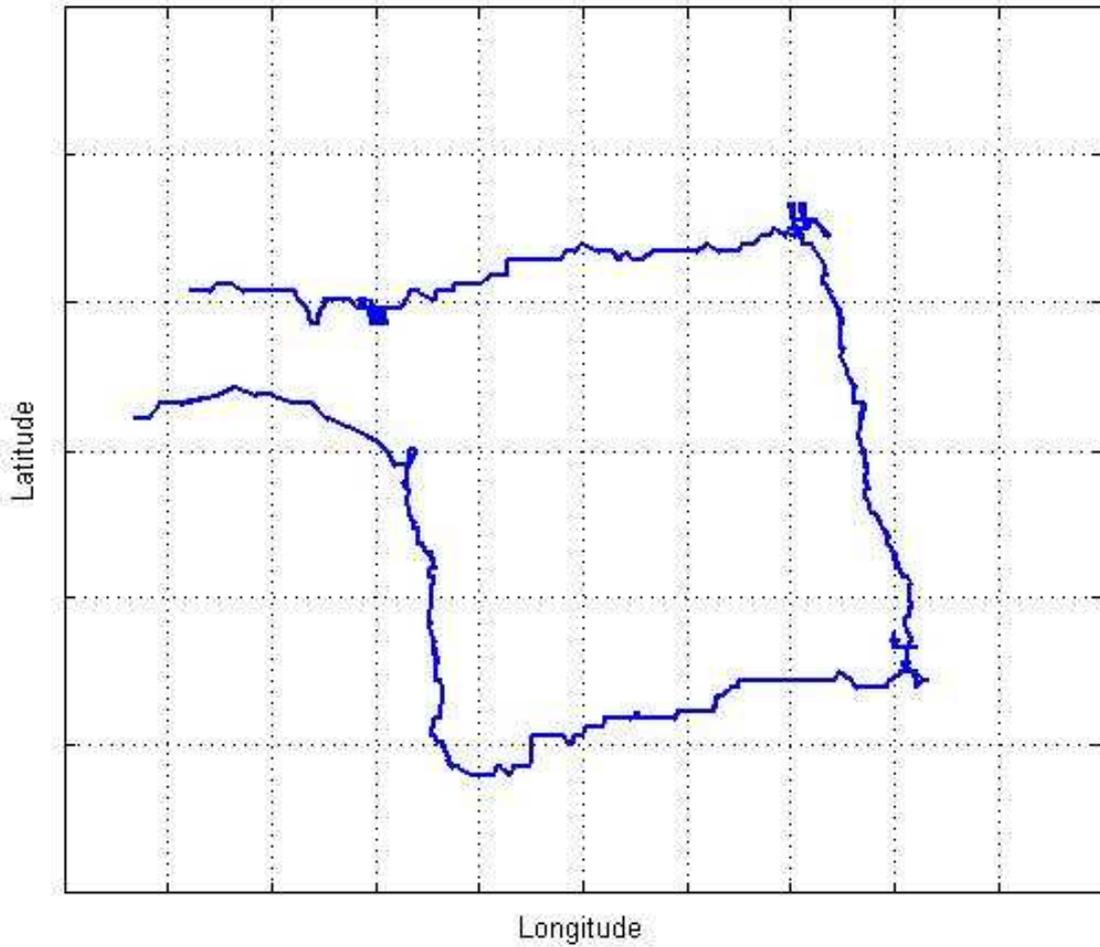


Figure 27: Autonomous square pattern UAV ground track

The flight data that have been collected during these flight tests will be used in next chapter to perform the rotorcraft dynamics identification. This identification is fundamental to determine the parameters that will be used to compute the FF term (cap. 4), that for the longitudinal dynamic are:

$$e = -\frac{A_{lon}}{\tau_e} \quad \delta = \frac{1}{2 \cdot \tau_e \cdot \omega_{nq}} \quad \omega_n = \omega_{nq} \quad (64)$$

Whereas for lateral:

$$e = \frac{B_{lat}}{\tau_e} \quad \delta = \frac{1}{2 \cdot \tau_e \cdot \omega_{np}} \quad \omega_n = \omega_{np} \quad (65)$$

6 UAV Dynamic model Identification

Aim of this chapter is to present a simple System Identification Procedure for Control Design. In particular the identified system will be used, in next chapters, as plant to compare performances of a traditional PID controller versus a feed-forward control algorithm based on dynamic model inversion (both for longitudinal and lateral dynamics). The proposed time-domain identification procedure is entirely developed in Matlab-Simulink environment, and requires no other external software applications [25].

This chapter is composed of three parts; in the first Open-Loop (OL) identification without cross-effects is presented, the second introduces cross-effects while in the last section parameters refinement is performed using a Closed Loop (CL) identification technique. Results will demonstrate how the proposed identification procedure provides a model showing good agreement with the recorded flight data, especially in closed loop validation where cross-axis effects are accounted.

6.1 On-Axis Identification

At the beginning, the helicopter longitudinal and lateral dynamics were considered as totally separated without any off-axis effect. The adopted Time-domain identification procedure is based on the comparison between the (real) measured signal and the simulated one. Measured angular rates signal have been previously filtered using a low pass filter with a cut-off frequency of 10 Hz.

Figure 28 shows a schematic representation of the procedure for longitudinal dynamics on-axis identification in Open loop chain.

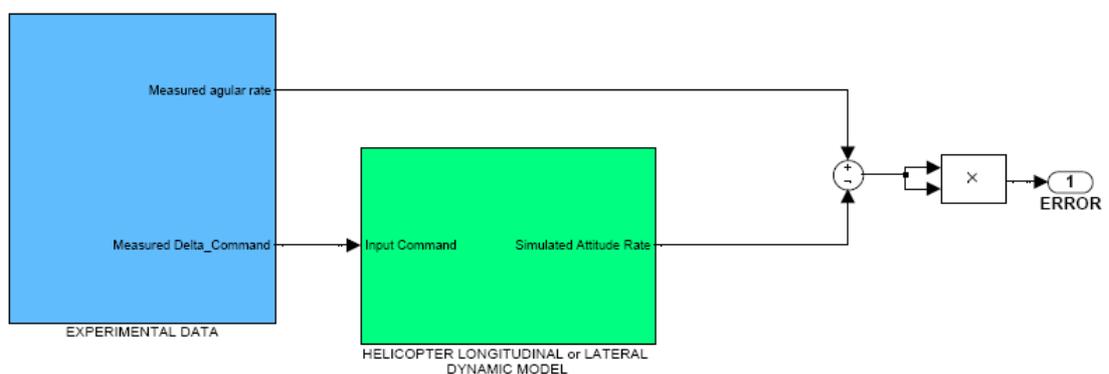


Figure 28: Open Loop On-axis Identification

The cost function (1) is the sum of the errors between the measured and simulated signals and is minimized in order to find the optimum transfer function parameters.

$$CostFunction = \sum (Y_{meas} - Y_{sim})^2 \quad (66)$$

where Y_{meas} and Y_{sim} are, respectively, the measured and simulated data.

The sum is performed every computing task, with a simulation step time equal to 0.01 s, both for longitudinal and lateral dynamics.

Proprietary scripts have been used in order to find the unknown model parameter values using MATLAB@ function [18]:

FMINSEARCH: Multidimensional unconstrained nonlinear minimization

Once these parameters have been computed, by minimizing $CostFunction$ over a training data set, a cross-validation test is then performed using totally new a data set (unused data, [19]).

In order to test fitting performances, the following *goodness of fit* index [19] has been computed:

$$R^2 = 1 - \frac{\sum (Y_{meas} - Y_{sim})^2}{\sum (Y_{meas} - \text{mean}(Y_{meas}))^2} \quad (67)$$

It must be noted that R^2 index value is strictly related with the data set used (Y_{meas}) for testing the model performance, and that its value can significantly change using different data sets. For this reason, it's more correct to use R^2 index to compare the performance of different models, using the same input, rather than to assess the performance of the same model using different inputs.

Moreover R^2 can also assume negative values (typically when the error is, on average, greater than the amplitude of signal) but, in any case, the greater is R^2 the better is the fitting goodness of the model.

6.2 Pitch and roll rate

Following a classical approach [1], second order transfer functions for the pitch and roll rate responses to pilot inputs have been considered (2, 3), and the relative parameters have been identified for several frequency sweeping commands [20].

$$\frac{q}{\delta lon} = \frac{-A lon}{\tau_e} \frac{\omega_{nq}^2}{s^2 + 1/\tau_e s + \omega_{nq}^2} \quad (68)$$

$$\frac{p}{\delta lat} = \frac{B lat}{\tau_e} \frac{\omega_{np}^2}{s^2 + 1/\tau_e s + \omega_{np}^2} \quad (69)$$

Using the cost function defined in (1), for longitudinal dynamic the following values were identified in:

$$\omega_{nq} = 12.1 \text{ (rad / s)} \quad A lon = 0.2488 \text{ (rad / rad)} \quad \tau_e = 0.132 \text{ (s)}$$

while for lateral dynamic:

$$\omega_{np} = 18.1 \text{ (rad / s)} \quad B lat = 0.22 \text{ (rad / rad)} \quad \tau_e = 0.132 \text{ (s)}$$

More details and initial values computation are reported in [20].

Figure 29 gives evidence of training data and Figure 30 of cross-validation data. The figures on the left column show the RUAV longitudinal dynamics behavior, while on the right show the lateral one. In both figures, the commands histories are reported in the bottom boxes while the helicopter responses (angular rates) are in the upper ones.

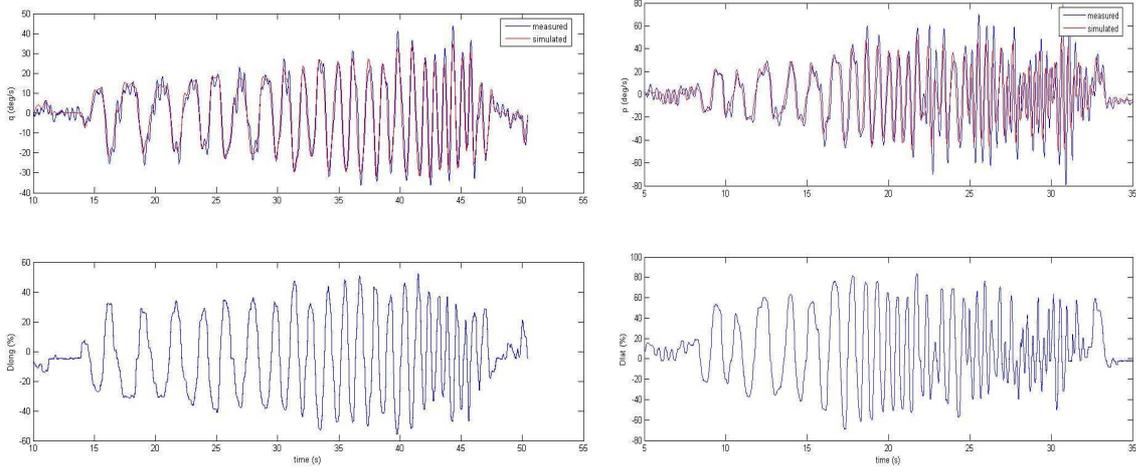


Figure 29: Longitudinal (left) and Lateral (right) sweeping input commands (below) and relative angular rates (upper) used for system training

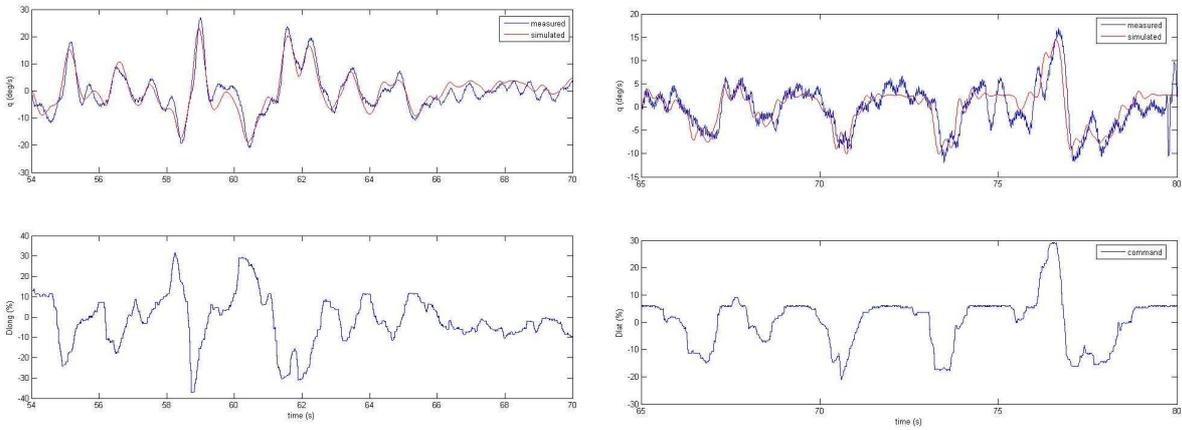


Figure 30: Longitudinal (left) and Lateral (right) validation input commands (below) and relative angular rates (upper) used for system validation

For cross-validation goodness of fit, the longitudinal index have been computed in $R^2=0.8586$ while for lateral in $R^2=0.6085$. A better agreement of the longitudinal model validation can effectively be seen by a simple look at the figures above.

Then, in body-frame reference, first order attitude-velocity transfer functions (4) have been chosen, and relative parameters have been identified using different recorded flight maneuvers (near hovering conditions [3]).

$$\frac{u}{\theta} = \frac{-g}{s - X_u} \qquad \frac{v}{\phi} = \frac{g}{s - Y_v} \qquad (70)$$

In both cases g parameter was assumed to be equal to 9.81 m/s^2 (gravity acceleration) and the remaining parameters have been identified.

As above, figures 31 and 32 show *training* and *cross-validation data* sets for attitude-velocity transfer functions (left column: longitudinal dynamic; right column: lateral dynamic; upper figures; body frame velocity response; below figures: input command history).

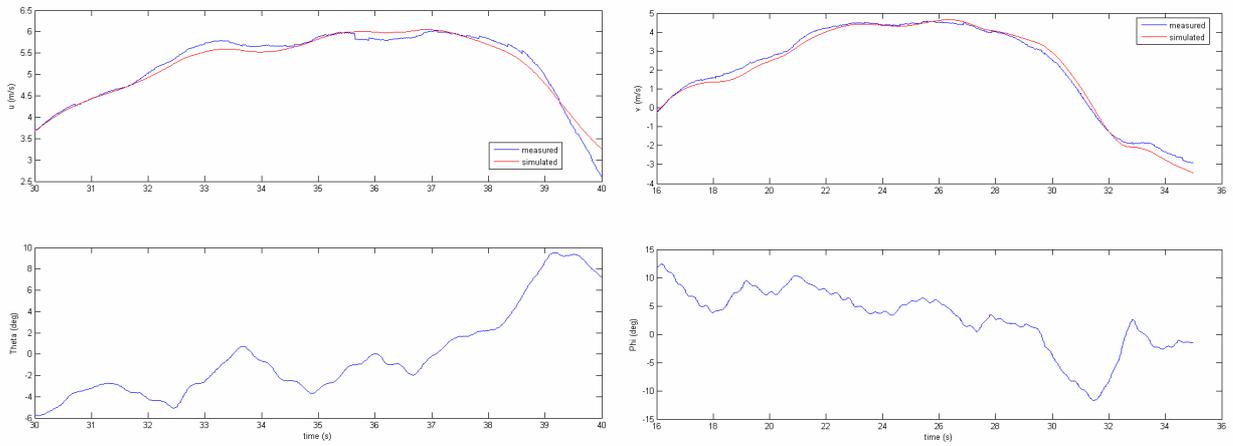


Figure 31: Input commands (below) and relative speeds (upper) for Longitudinal (left) and Lateral (right) velocity model Identification

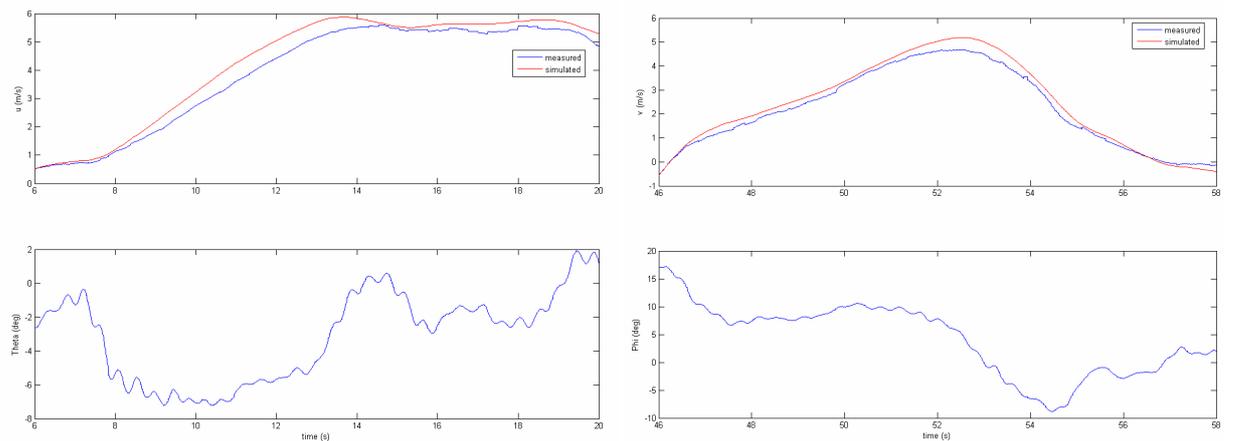


Figure 32: input commands (below) and relative speeds (upper) for Longitudinal (left) and Lateral (right) velocity model Validation

Again, using the cost function defined in (1), the following identified parameters were computed:

$$Xu = -0.052 \text{ (1/s)}$$

$$Yv = -0.046 \text{ (1/s)}$$

with respectively validated goodness of fit index of $R^2=0.9366$ for longitudinal dynamic and $R^2=0.9586$ for lateral.

6.3 Vertical speed dynamic

Vertical speed dynamic have been modeled using a first order transfer function from command to velocity [1]:

$$\frac{w}{\delta_{coll}} = \frac{Z_{coll}}{s - Z_w} \quad (71)$$

Figure 33 shows *training* and *cross-validation* data sets used for heave dynamic transfer functions identification.

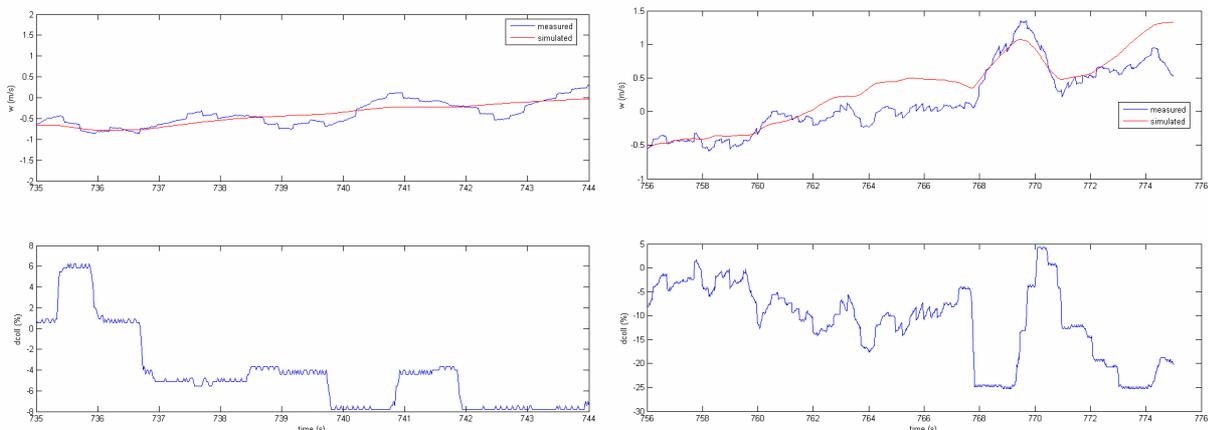


Figure 33: Training (left) and validation (right) data sets for heave dynamic transfer functions (upper: vertical velocity, below: collective input command)

The relative derivatives have been identified in:

$$Z_{coll} = -7.7330 \quad (m/(s^2 \cdot rad))$$

$$Z_w = -0.3567 \quad (s^{-1})$$

with a validated goodness of fit index of $R^2=0.7127$.

Table n.1 summarizes the values found for the On-Axis parameters using Open Loop identification only:

| X_u | Y_v | $Alon$ | $Blat$ | tf | Xa | Yb | g | Lb | Ma | Z_{coll} | Z_w |
|-------|-------|---------|---------|-------|------------------------|------------------------|------------------|------------------|------------------|------------------------|---------|
| 0.052 | 0.046 | 0.2488 | 0.22 | 0.132 | 9.81 | 9.81 | 9.81 | 327.6 | 146.4 | -7.733 | -0.3567 |
| 1/s | 1/s | rad/rad | rad/rad | s | m/(s ² rad) | m/(s ² rad) | m/s ² | 1/s ² | 1/s ² | m/(s ² rad) | 1/s |

Table 5: On-axis Open Loop Identified Values using a pure longitudinal or lateral maneuver

6.4 OFF-Axis Identification in open loop

As already shown in chapter 3, the system below (6) describes the model used for Off-axis Open Loop (OL) identification tests. The equations of this system present both the known On-Axis parameters values (reported in Table 5) and new 11 Off-axis derivatives (reported in right column of eq.72 with a question mark) not yet identified

$$\begin{aligned}
 \dot{u} &= X_u \cdot u - g \cdot \theta + X_a \cdot a & L_a \text{ ?} \\
 \dot{v} &= X_v \cdot v + g \cdot \phi + Y_b \cdot b & M_b \text{ ?} \\
 \dot{p} &= L_b \cdot b + L_a \cdot a + L_u \cdot u + L_v \cdot v & L_u \text{ ?} \\
 \dot{q} &= M_a \cdot a + M_b \cdot b + M_u \cdot u + M_v \cdot v + M_{col} \cdot \delta coll & L_v \text{ ?} \\
 \dot{\phi} &= p & M_u \text{ ?} \\
 \dot{\theta} &= q & M_v \text{ ?} \\
 \dot{a} &= -q - \frac{a}{\tau_f} + \frac{A_b}{\tau_f} \cdot b + \frac{A_{lon}}{\tau_f} \cdot \delta lon + \frac{A_{lat}}{\tau_f} \cdot \delta lat & A_b \text{ ?} \\
 & & B_a \text{ ?} \\
 \dot{b} &= -p - \frac{b}{\tau_f} + \frac{B_a}{\tau_f} \cdot a + \frac{B_{lat}}{\tau_f} \cdot \delta lat + \frac{B_{lon}}{\tau_f} \cdot \delta lon & A_{lat} \text{ ?} \\
 & & B_{lon} \text{ ?} \\
 \dot{w} &= Z_w \cdot w + Z_{coll} \cdot \delta coll & M_{col} \text{ ?}
 \end{aligned}
 \tag{72}$$

The first 7 parameters $Ab, Ba, Alat, Blon, Ma, Lb, Mcol$ have been identified using again several frequency sweeping inputs near hovering condition [1].

After many tests, it was found that, as suggested by Mettler [21], it can be set $Ab=Ba=0$. Furthermore, since MATLAB@ *fminsearch* function seems to work better with a maximum of 5 parameters, reducing the number of parameters to be identified it is also desirable.

For off-axis derivatives identification procedure, all unknown parameters were set to ‘zero’ at first. In all test cases, in order to verify that a true global minimum of the cost function was found, all the runs were repeated with different initial conditions.

A new cost function, reported in Fig. 34, that takes into account both errors in longitudinal and in lateral attitude, was then adopted:

$$CostFunction = \sum ((Phi_meas - Phi_sim)^2 + (Theta_meas - Theta_sim)^2) \tag{73}$$

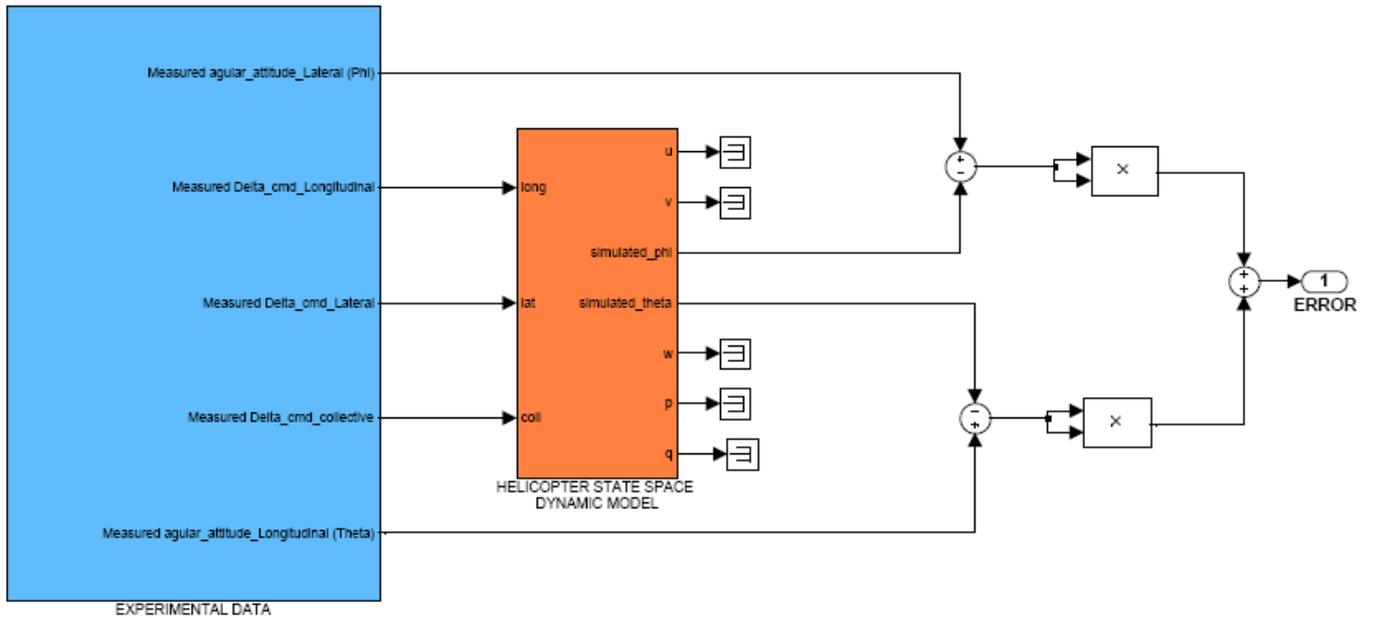


Figure 34: Off-axis Derivatives Open Loop Identification

The same recorded frequency sweeping input commands have been again used in input to the model for both longitudinal and lateral dynamics analysis and, as expected, a great improvement has been noticed in reducing attitude drifts compared to the No-Cross-effect-Model (NCM).

Figures 35 and 36 report training experiments with longitudinal (Fig.36) and lateral (Fig.37) excitements; blue lines depict the experimental recorded data (command history) used in input, while red ones belong to the cross-effect model response and green ones to the NCM. It can be noticed how the cross identification reduce errors in helicopter attitude estimation for both the dynamics.

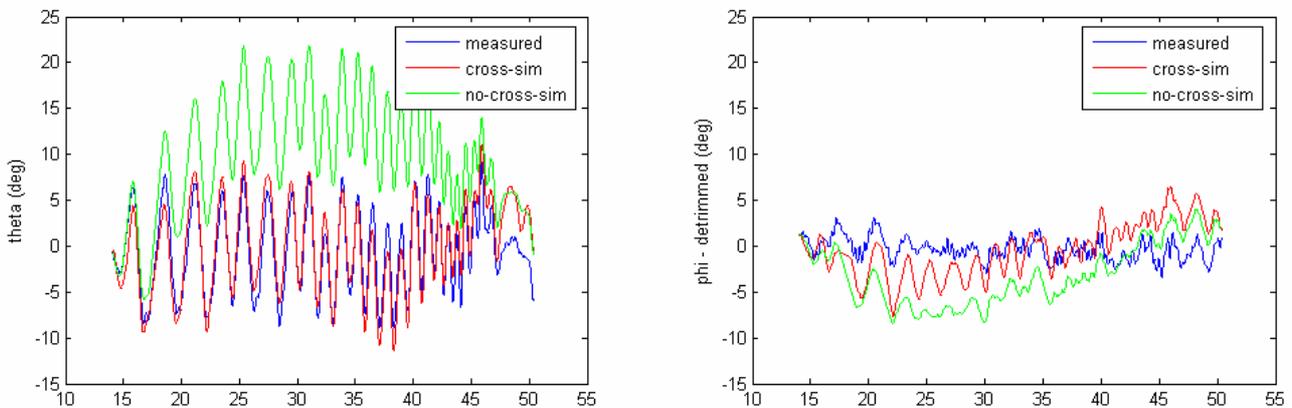


Figure 35: Off-axis Derivatives Open Loop Identification - Training data set: longitudinal inputs (Left: pitch angle, Right: roll angle.)

For the longitudinal test, the goodness of fit gives $R^2= 0.7353$ vs $R^2= -5.5632$ for NCM in pitch and for roll $R^2= -1.5551$ vs $R^2= -12.5321$ for NCM: the fit function show a clear increase in attitude estimation.

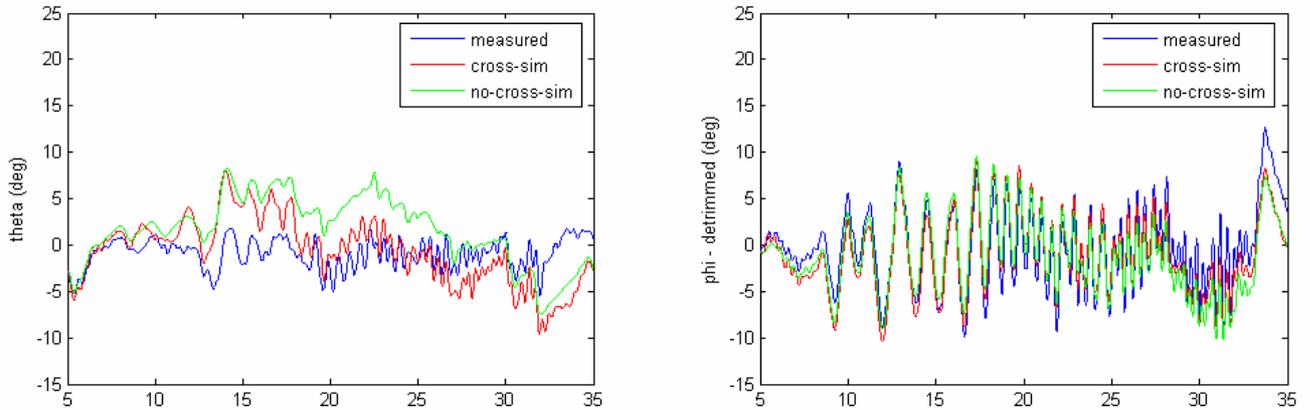


Figure 36: Off-axis Derivatives Open Loop Identification - Training data set: lateral inputs (Left: pitch angle, Right: roll angle.)

Also for the lateral sweeping test, a good improvement can be seen in signals fitting, and it can be noticed how certain helicopter dynamics are now present in the cross-effects model response (e.g. Fig. 36, left, red vs green signal for the roll angle, $t=[20\div 25s]$). In this case, for pitch tests, goodness of fit is $R^2=-3,4643$ whereas $R^2=-6.5254$ for NCM and, for roll tests, $R^2=0.7616$ whereas $R^2=0.6636$ if no cross effects area accounted.

All the identified parameters are reported in Table 6. Looking at the values, it can be seen that they assume different values if identified with longitudinal or lateral test case and, sometimes, present also a change in sign.

| Derivative | Longitudinal Test Value | Lateral Test Value | OL Mean value |
|---------------------------------------|--------------------------------|---------------------------|----------------------|
| <i>Alat</i> (-) | 0.2023 | -0.0446 | 0.0789 |
| <i>Blon</i> (-) | 0.0655 | -0.0648 | 3.5000e-004 |
| <i>La</i> (-) | -0.5404 | 173.4853 | 86.4724 |
| <i>Mb</i> (-) | -37.4823 | -69.9203 | -53.7013 |
| <i>Mcoll</i> ($1/s^2$) | -21.8085 | -12.3811 | -17.0949 |
| <i>Mu, Mv, Lu, Lv</i> ($rad/(m s)$) | Set to 0 | Set to 0 | Set to 0 |

Table 6: Off-axis Open Loop Identified Values using a pure longitudinal or lateral maneuver

As to proceed, it was decided to assume mean values for these parameters obtaining a sort of ‘mean system’ and to measure goodness of fit of this new model only with cross-validation tests.

Using mean values for *Alat*, *Blon*, *Ma*, *Lb* and *Mcoll* derivatives makes the model to assume a *mean behavior* between the one optimized with longitudinal test case only and the one with lateral one. This behavior has been noticed, by the authors, in all the validation tests they have performed.

Figure 37 shows the cross-validation of the final system with longitudinal and lateral real flight data.

The goodness of fit for longitudinal cross-validation test is $R^2=0.7549$ versus $R^2= -0.2332$ for NCM for pitch and for roll $R^2= -0.7291$ whereas $R^2= -0.7301$ for NCM.

The goodness of fit for lateral cross-validation test, instead, gives $R^2= -0.3219$ vs $R^2= -2.0041$ for NCM for pitch and for roll $R^2= 0.1903$ for roll whereas $R^2= 0.1058$ for NCM.

Again, cross-effect model shows a better agreement than NCM especially in longitudinal dynamic, and it can be seen (Fig 37, right column, phi signal e.g. $t=74s$) that the final model captures some off-axis helicopter dynamics.

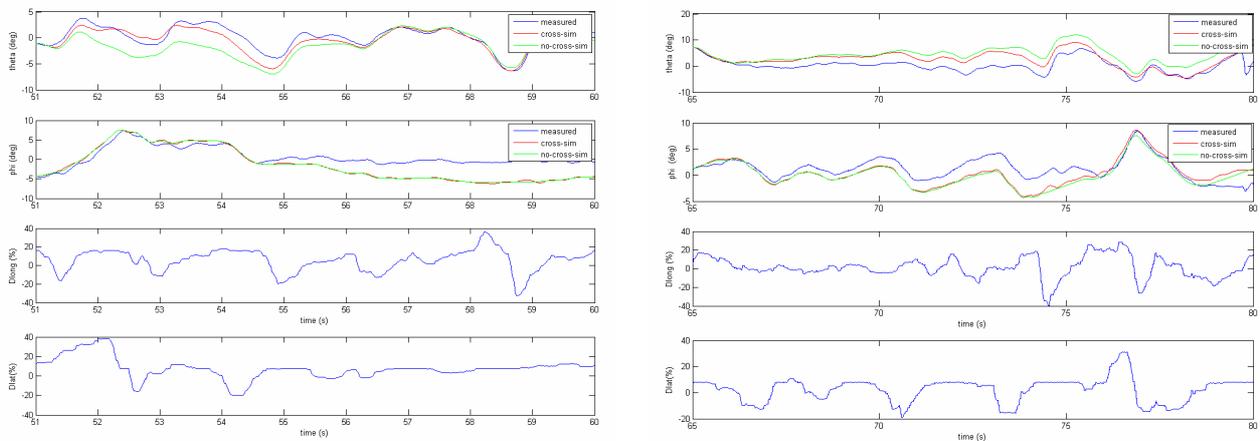


Figure 37: Open Loop Validation - Longitudinal (left column) and Lateral (right column)

The OL identification procedure for speed derivatives Mu , Mv , Lu , Lv , was performed using zero as starting values for the optimization algorithm, given that the data used were relative to flights near hovering condition.

However, probably due to too big drifts in simulated speed signals, it was not possible to find a set of values different from zero providing a better agreement in cross-validation tests.

Therefore, zero values for Mu , Mv , Lu , Lv , have been used as starting point for closed loop identification tests (next section).

6.5 OFF-Axis Identification in Closed Loop

During closed loop (CL) identification the above mentioned 9 unknown parameters $Alat$, $Blon$, La , Mb , Mu , Mv , Lu , Lv , $Mcol$ were initialized using the OL mean values (Table 6).

The proposed CL identification procedure is based on the control architecture depicted in Figure 38. In this procedure, real measured set-point values and the relative off-axis commands are the inputs to the model (e.g. $U_set_point+dlat+dcoll$ for longitudinal maneuvers, $V_set_point+dlong+dcoll$ for lateral ones), while the attitude error (difference between real and predicted output of the outer control loop) and the command error (difference between real and predicted output of the inner control loop) are used to compute the new cost function.

For longitudinal maneuvers the cost function is:

$$CostFunction = \sum ((Theta_meas - Theta_sim)^2 + (dlong_meas - dlong_sim)^2) \quad (74)$$

while for lateral maneuvers:

$$CostFunction = \sum ((Phi_meas - Phi_sim)^2 + (dlat_meas - dlat_sim)^2) \quad (75)$$

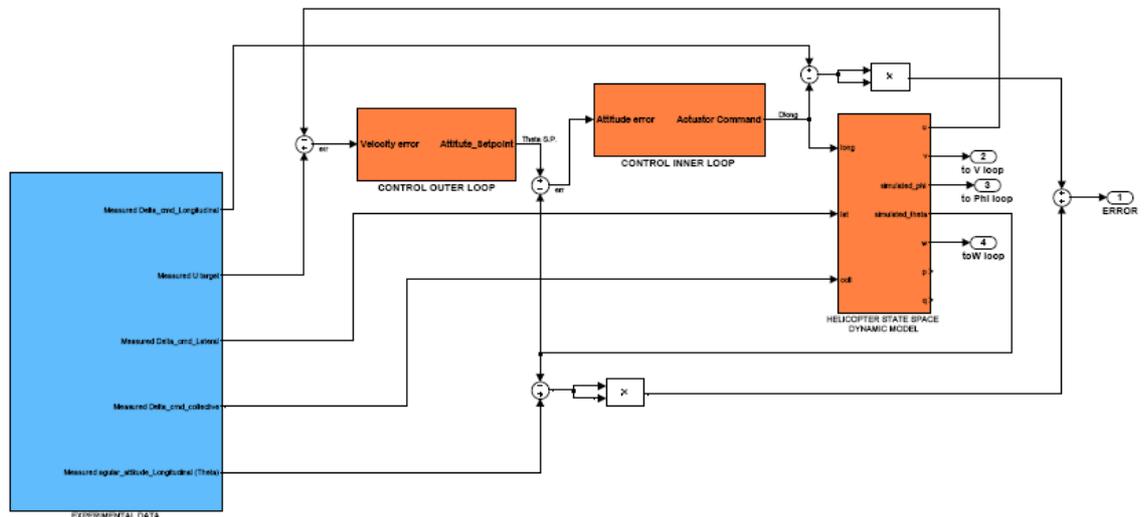


Figure 38: Closed Loop Identification logic

The table 7 reports the identified values during step-like cross-validation velocity maneuvers.

| Derivatives | OL (Mean) | CL longitudinal | CL lateral | CL Mean |
|-------------------------|------------------|------------------------|-------------------|---------------------|
| Alat (-) | 0.0789 | 0.1328 | 0.0777 | 0.1053 |
| Blon (-) | 3.5000e-004 | 2.9896e-004 | 3.4119e-004 | 3.2008e-004 |
| La (-) | 86.4724 | 157.4712 | 89.2649 | 123.3681 |
| Mb (-) | -53.7013 | -108.0648 | -55.6379 | -81.8513 |
| Mu (rad/(m s)) | 0 | -0.0053 | 6.4525e-005 | -0.0026 |
| Mv (rad/(m s)) | 0 | -0.0018 | -1.6989e-005 | -9.0849e-004 |
| Lu (rad/(m s)) | 0 | -8.9655e-004 | 1.2198e-004 | 3.8729e-004 |
| Lv (rad/(m s)) | 0 | -0.0026 | -8.7996e-006 | -0.0013 |
| Mcol (rad/(m s)) | -17.0949 | -17.0854 | -17.1030 | -17.0942 |

Table 7: Values Identified in Off-axis Closed Loop (CL)

It can be noticed that the final values are similar to those reported in [11] and [12] for the X-cell helicopter (a small scale model very similar to Unibo RUAV), except for La and Mb that, in authors case, are greater than expected but correctly smaller than the on-axis corresponding derivatives Lb, Ma (see Table 6 for comparison).

About speed derivatives Mu , Mv , Lu , Lv , it can be seen how the final values are much smaller than those found in literature for similar rotorcrafts; anyway in our opinion, this confirms that the influence of speed near hover condition can be neglected. In fact, $Mu = -0.0026$ means that a speed change of $u = 2$ m/s induces a really poor contribution (0.0052 rad/s^2) to pitch acceleration with respect to the contribution of Ma (e.g. 2.54 rad/s^2 with $a = 1 \text{ deg.}$). Mv and Lu (the off-axis speed derivatives) are even smaller and therefore negligible.

Validation tests of the identified parameters in different closed loop controls are shown in Fig.39 both for longitudinal (left column) and lateral (right column) inputs.

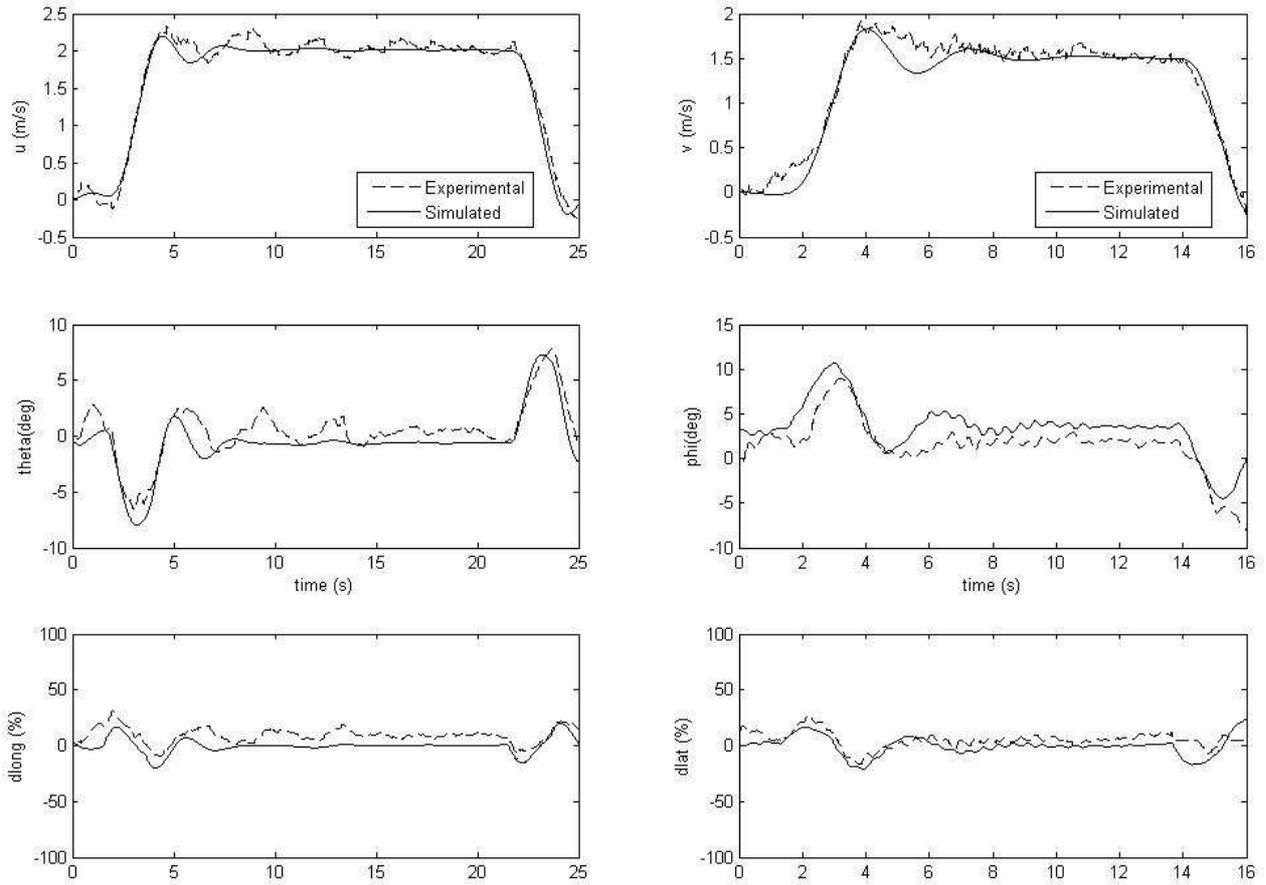


Figure 39: Comparison between predicted (solid) and recorded (dashed) closed loop Baseline Control Systems responses for longitudinal (column 1, left) and lateral (column 2, right) velocity steps.

| GIF index | Longitudinal | Longitudinal NCM | GIF index | Lateral | Lateral NCM |
|--------------|--------------|------------------|------------|---------|-------------|
| $R^2 u$ | 0.9525 | 0.9447 | $R^2 v$ | 0.9689 | 0.9662 |
| $R^2 \theta$ | 0.7886 | 0.7699 | $R^2 \phi$ | 0.8288 | 0.8250 |
| $R^2 dlong,$ | 0.6449 | 0.6229 | $R^2 dlat$ | -0.3617 | -0.4065 |

Table 8: Closed loop Goodness of Fit Indexes comparison

Last of all, Table 8 gives final evidence that accounting cross-effects into the dynamic model brings to a better agreement of the model itself regarding speeds, attitude and closed loop commands and that this improvement is shown both in longitudinal and lateral dynamics.

This simple and innovative identification procedure that has been fully developed in a Matlab-Simulink environment, has been presented in order to obtain a representative dynamic model of a small rotorcraft UAV near hovering flight condition. In following paragraphs, the identified model will be used for control design purposes, for example by using MATLAB@

signal constraint blockset, it will be possible to pre-tune controller gains, giving a set of control target performances as *raise time*, *maximum overshoot* and *final error*.

The complete derived models will be used also to perform comparison between advanced control architecture, based on feed-forward actions, with common control architecture (like PID).

6.6 Flight Validation of the Baseline Control System Model

In this paragraph, before the analysis and comparison between the performance of the feed-forward and the baseline control system, is reported the verification of how well the model of the baseline control predicts the closed-loop behaviour of UNIBO autonomous helicopter dynamics. As already mentioned Baseline control system has been implemented and tested in UNIBO RUAV in National Instruments C-Rio computer using NI Labview Programming Code and flight trials were used to identify dynamics. Since closed loop verification involves all components of the helicopter control system, from the flight-mechanics to the computer systems, it allows to detect possible anomalies or un-modeled dynamics.

Next table resumes the gains values adopted in flight experiments for UNIBO RUAV (baseline) controller.

| | Baseline Longitudinal | Baseline Lateral |
|-----------------------|-----------------------|------------------|
| Attitude Proportional | $Kp=-1$ | $KpLat=1$ |
| Attitude Integral | $Ki=-1$ | $KiLat= 1$ |
| Attitude Derivative | $Kd=0$ | $KdLat=0$ |
| Velocity Proportional | $Kpv=-10$ | $KpvLat=10$ |
| Velocity Integral | $Kiv=-1$ | $KivLat=1$ |
| Velocity Derivative | $Kdv =0$ | $KdvLat =0$ |
| Filter time constant | -- | -- |

Table 9:Basic Gains for Baseline Flight tests

This set of gains has been tested in a great number of flight trials and it has been showed that the system tuned with this set of parameters has sufficient controllability and robustness for the maneuvers required for slow hover-like flight. [8,9]

For closed loop model validation, the helicopter was given a step-like velocity reference command in the longitudinal ($V_{xref}=2$ m/s) and in lateral directions ($V_{yref}=1.5$ m/s). The actual helicopter responses were recorded during the flight-test; meanwhile the predicted helicopter responses were obtained from the model of the closed loop system. The comparisons between the real and predicted responses, for the lateral and longitudinal directions, have been shown in figure 39 in previous paragraph. All key variables, the control signals, the attitude angles and the longitudinal and lateral velocities, show a good agreement. Notice in particular how the model accurately follows real responses during velocity transients. This is a quite important requirement for the simulation model, since the feed-forward compensation will be very important, above all, during such transients.

6.7 Disturbances and sensor noise model

The dynamic model described in the previous section predicts only the low frequency response of the helicopter (approximately under 5Hz), while real signals acquired by onboard sensors present also higher frequency noise values. Analyzing the power spectral density (PSD) of speed signals acquired during several flight tests (fig. 40), it can be noticed the presence of well distinct peaks at 34 Hz, corresponding to the double of main rotor revolution frequency ($\omega_n \cong 17Hz$ corresponding to about 1000 rpm).

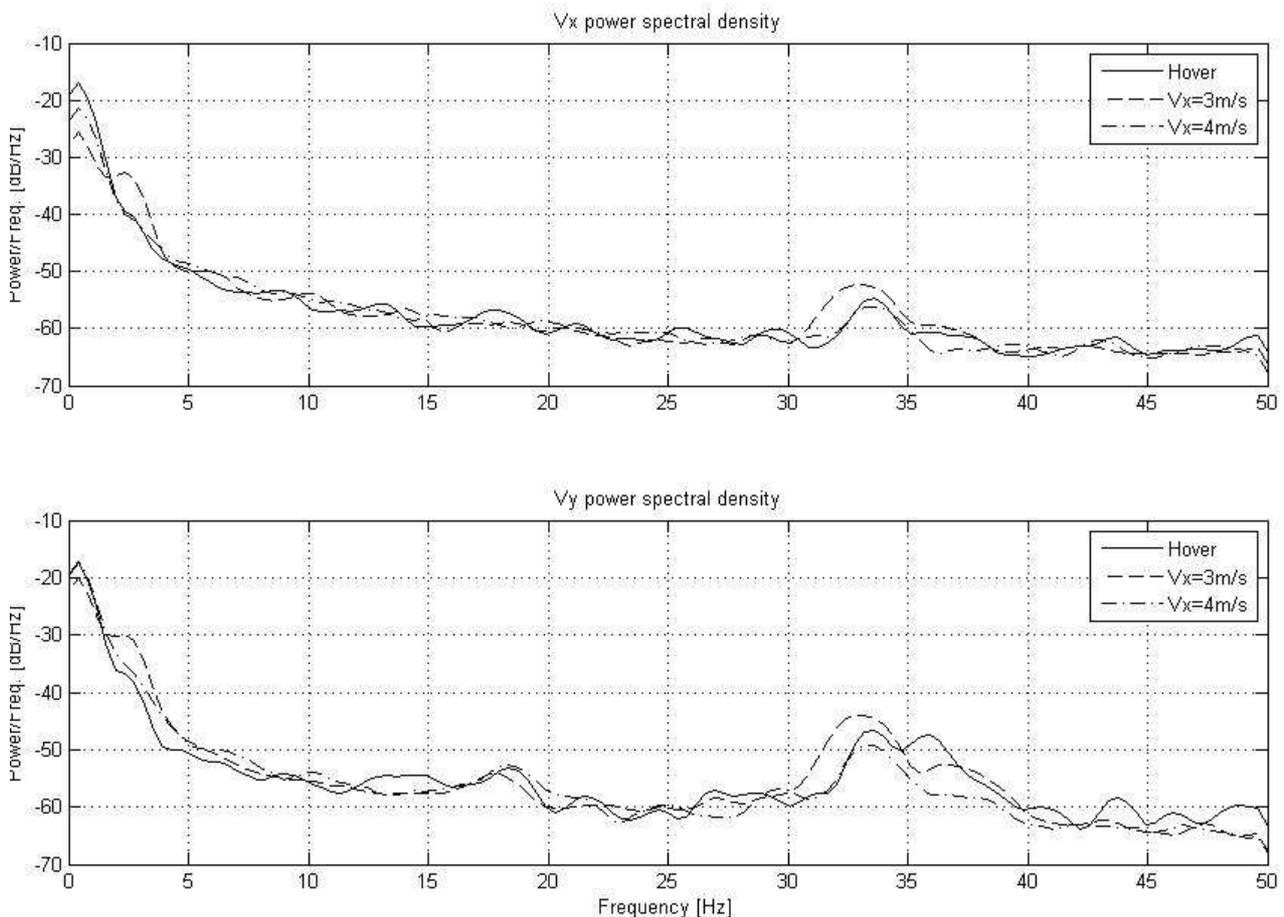


Figure 40: Power spectral density of longitudinal and lateral velocity signals

This high frequency noise in attitude and speed signals can downgrade the controller performances. In order to have a more realistic prediction of the controller behavior during simulation tests that will be performed in Hardware in the loop test bench (see HIL in Chapter 8), a model of these disturbances should be added to the state space model helicopter

dynamics. The effect of noise attitude and speed signals has been predicted by calculating the closed loop transfer functions between a injected disturbance on attitude ($na(s)$) or speed ($nu(s)$) and the consequent speed disturbance ($u(s)$).

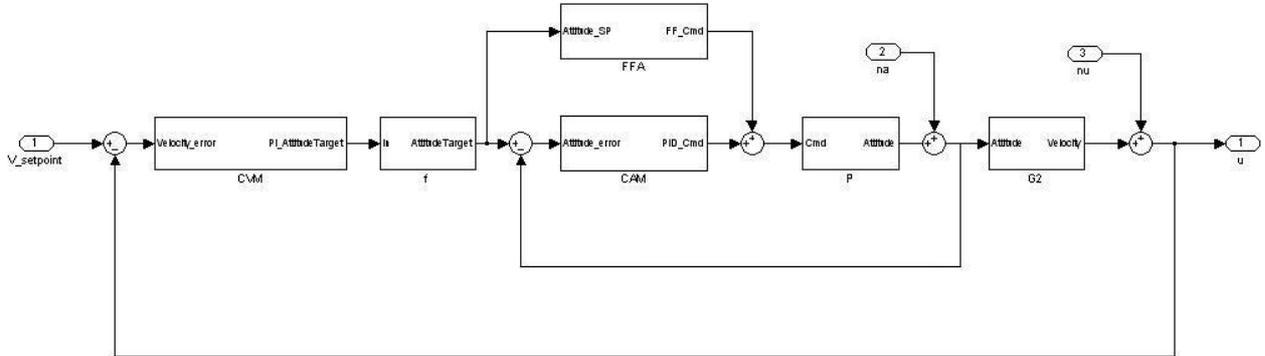


Figure 41: Block diagram of the FF controller with noise injection

Referring to figure 41, the transfer function between “noise on attitude” and speed is:

$$\frac{u(s)}{na(s)} = \frac{G_2(s)}{1 + P(s) \cdot (CAM(s) \cdot (1 + G_2(s) \cdot CVM(s) \cdot f(s)) + FFA(s) \cdot G_2(s) \cdot CVM(s) \cdot f(s))} \quad (76)$$

and the transfer function between “noise on speed” and speed output is:

$$\frac{u(s)}{nu(s)} = \frac{1 + CAM(s) \cdot P(s)}{1 + P(s) \cdot CAM(s) + G2(s) \cdot P(s) \cdot CVM(s) \cdot f(s) \cdot (FFA(s) + CAM(s))} \quad (77)$$

Same relations are valid both for Feedforward and for the Baseline controller setting, but, for the latter, it has to be set:

$$f(s) = 1, FFA(s) = 0 \quad (78)$$

Bode plots of Equations 76 and 76 have been reported in Fig.43 and Fig.44.

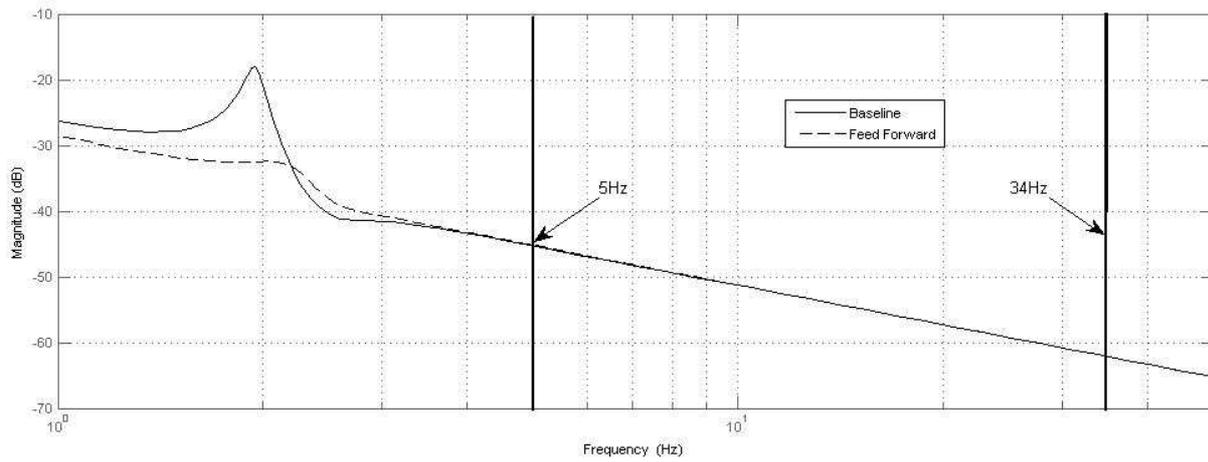


Figure 42: Bode plot of velocity 'noise on attitude' transfer function ($u(s)/na(s)$)

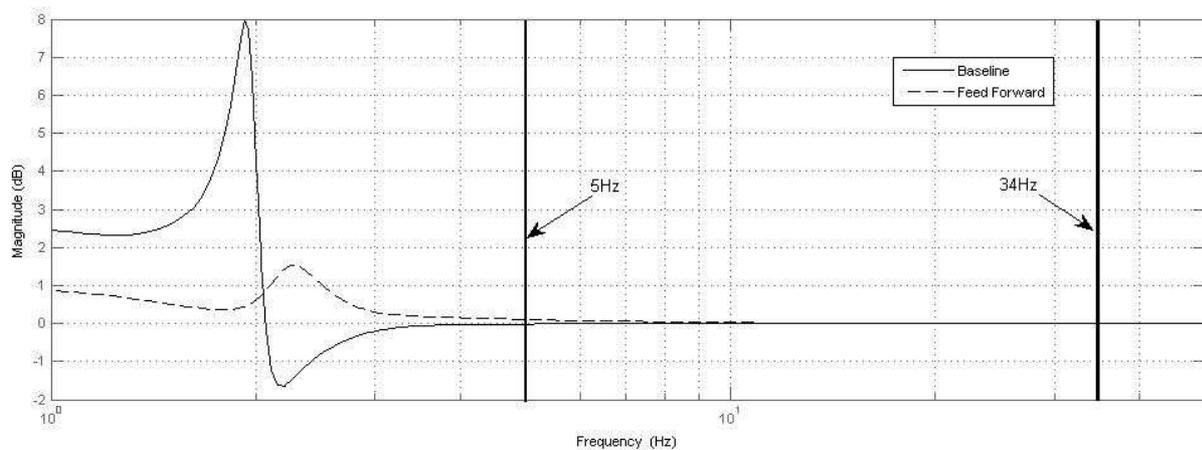


Figure 43: Bode plot of velocity 'noise on velocity' transfer function ($u(s)/nu(s)$)

As it can be seen in fig. 42, all the disturbance frequencies in the attitude signal are heavily damped by the system and will have only small influence on controller performances. In the case of noise injected in velocity, instead, figure 43 shows that the high frequencies remain unaltered and can still disturb the controller. This analysis leads to the conclusion that it is necessary to model the high frequencies of speed noise in order to have a more realistic prediction of the controller performances during the real flight tests, while noise on attitude is not necessary. Moreover modeling 'noise on velocity' appears to be critical especially for Feedforward controller which operates derivatives (Eq. 52) with velocity speed error signals without tunable gains.

The PSD of the speed signals shows that there is no correlation between the noise characteristics and the flight command inputs; for this reason Auto Regressive structure (AR) [19] has been chosen to model these disturbances.

In general, the AR model can be written as:

$$F(q) \cdot y(t) = e(t) \quad (79)$$

$$F(q) = 1 + f_1 q^{-1} + \dots + f_n q^{-n}$$

where $y(t)$ is the output signal at time t , $e(t)$ is a white noise and q is a delay operator.

Identification of f_1, \dots, f_n parameters has been performed using a least square method starting from flight data previously filtered in order to eliminate the low frequency values, which are still modeled by state space model. The polynomial order, which is the only free parameter in this approach, has been fixed equal to 30, because, previous tests have revealed that this value represents a good compromise between accuracy and computational effort.

Fig 44 shows the result of the identification process. Data used for the identification process were the one with the higher power at 33Hz (flight test with longitudinal speed at 3m/s). As it can be seen, all the main characteristics of the noise PSD have been predicted with high accuracy, bringing to a good statistical model of the noise.

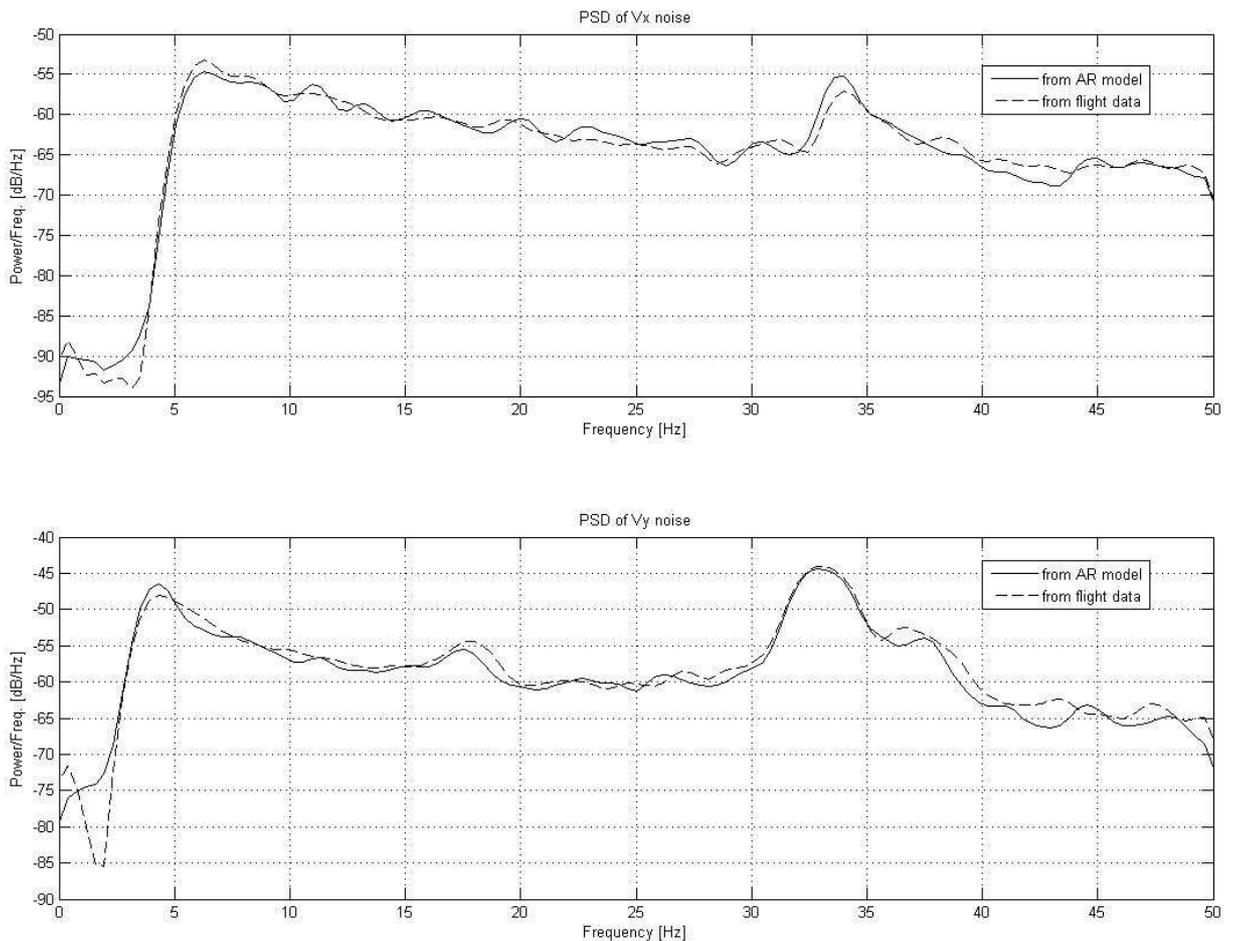


Figure 44: Comparison between modeled (dashed) and original (continuous) flight data PSD

7 Comparison of controllers tuned with the basic gains

In order to assess the different performances of the two proposed control architectures, a preliminary analysis on stability margins and on disturbances rejection has been conducted. Since the aim of this first comparison is to test the influence of the presence of the feed forward term, the same PID gains were adopted both for Baseline and for Feedforward controllers. The set of basic gains, whose values are derived from the ones that have been used in flight experiments of UNIBO RUAV Baseline controller, is reported in next table.

| | Baseline Longitudinal | Baseline Lateral | FF Longitudinal | FF Lateral |
|-----------------------|-----------------------|------------------|-----------------|-----------------|
| Attitude Proportional | $Kp=-1$ | $KpLat=1$ | $Kpm=-1$ | $KpmLat=1$ |
| Attitude integral | $Ki=-1$ | $KiLat=1$ | $Kim=-1$ | $KimLat=1$ |
| Attitude derivative | $Kd=0$ | $KdLat=0$ | $Kdm=0$ | $KdmLat=0$ |
| Velocity Proportional | $Kpv=-10$ | $KpvLat=10$ | $Kpvm=-10$ | $KpvmLat=10$ |
| Velocity integral | $Kiv=-1$ | $KivLat=1$ | $Kivm=-1$ | $KivmLat=1$ |
| Velocity derivative | $Kdv=0$ | $KdvLat=0$ | $Kdvm=0$ | $KdvmLat=0$ |
| Filter time constant | -- | -- | $Tfilt=0.15$ | $TfiltLat=0.15$ |

Table 10: Basic PID gains

Moreover, for Feedforward controller, a time constant of 0.15 s. has been adopted for the first order filter both for longitudinal and lateral dynamics.

7.1 Stability margins

The analysis of Baseline and FF control systems stability have been conducted using Gain and Phase margin analysis [22] The transfer functions BLG and FLG, reported in paragraph 4.3 for Baseline and FeedForward systems have been used to plot Bode gain and Phase Margins. Following table reports stability margins computed or baseline and FF control systems computed with this basic set of parameters.

| | | Gain margin (dB) | Phase margin (deg) |
|-------------|--------------|----------------------------|--------------------------|
| Baseline | Longitudinal | 14.15 dB (@ 4.55 rad/sec) | 33.5 deg (@1.9 rad/sec) |
| | Lateral | 24.53 dB (@ 13.57 rad/sec) | 30.9 deg (@1.79 rad/sec) |
| Feedforward | Longitudinal | 17.58 dB (@ 14.70 rad/sec) | 74.2 deg (@1.66 rad/sec) |
| | Lateral | 18.47 dB (@ 16.08 rad/sec) | 74.1 deg (@1.65 rad/sec) |

Table 11: Stability margins of the FF and Baseline controllers for longitudinal and lateral dynamics

Compared with the specifications used for flight control design (MIL-F-9490), which require a gain margin of 6 dB and a phase margin of 45 deg, it can be seen how the baseline system tuned with basic gains of table 10 lacks adequate Phase margins for longitudinal and lateral dynamics control [23].

On the contrary, it must be noticed that FF tuned with the same PID gains is already compliant to the stability margins required by the norms, and thus it means that just adding the feedforward term to the Baseline controller in the inner loop, can bring the system to a higher stability level.

7.2 Three-axis control and cross effects disturbances.

Another important aspect in controllers design is to achieve a good disturbances rejection. In present paragraph we study the relation between disturbances and their effects on controlled variables (longitudinal and lateral velocities) and we compared the behavior of the FF and the Baseline control systems in the frequency domain

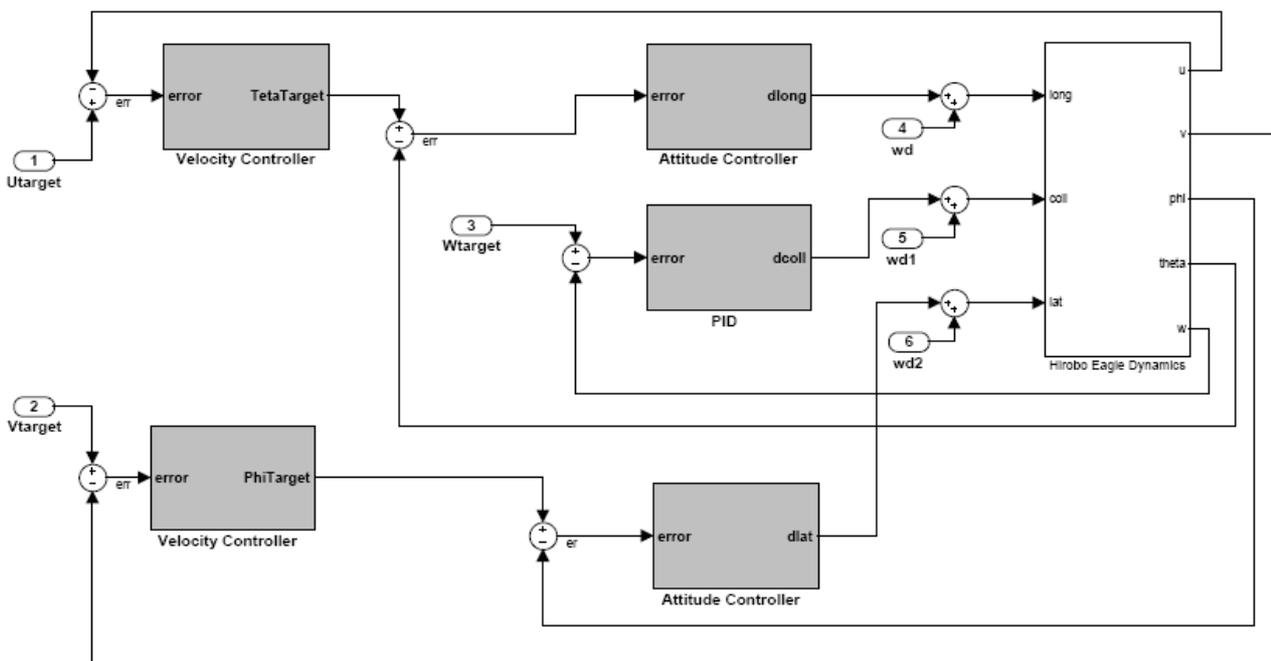


Figure 45: cross effects and external disturbance

In both baseline and proposed FF+PI control systems the control model is composed by three SISO controllers working at the same time for longitudinal (U_{target}), lateral (V_{target}), and vertical (W_{target}) dynamics control.

In this way it is possible to give three different set-points time-histories for longitudinal, lateral and vertical velocities and make them vary in a total independent way, for example it is possible to give a velocity step to longitudinal velocity and zero as set-point for lateral and vertical velocities.

This contemporaneous control over three axis, since the identified model takes into account cross-effects (long-lateral, vert-long), induces disturbances on each of the three controlled velocities. This kind of disturbances could be considered as internal in order to distinguish them from external disturbances as, for example, wing gusts.

Simulation model used in following simulations takes in to account also wind gust that are modeled as command input disturbances (vd Mettler p 152) and are represented as inputs wd ,wd1, wd2 in figure 45. As already mentioned and as suggested in ref. (libro Mettler) a wind gust disturbs the direction and the speed of the airflow encountered by the helicopter. This changes the aerodynamic load of the rotor, producing an uncommanded rotor flapping response which will, in turn, disturb the helicopter attitude. If we ignore the forces produced by the gust on the fuselage, a wind gust, since it changes the aerodynamic angle of attack at the blade (in a cyclic manner), has effect on the helicopter similar to that of a sudden change in the cyclic controls. Thus wind gust are effectively modeled as command input disturbances.

Wind gust model used is the Discrete Wind Gust Model in the form of the standard "1-cosine" shape in conformity with mathematical representation in the Military Specification MIL-F-8785C [24]. The gust is applied to each axis individually, or to all three axes at once. It can be specified the gust amplitude (the increase in wind speed generated by the gust), the gust length (length, in meters, over which the gust builds up) and the gust start time [18]

Figure 47 shows a control structure scheme valid both for FF lateral and longitudinal velocity control. We can see that the wind disturbance is modeled as an input and that its contribution is summed to FF action (FFA) term and to PI Attitude controller term (CAM).

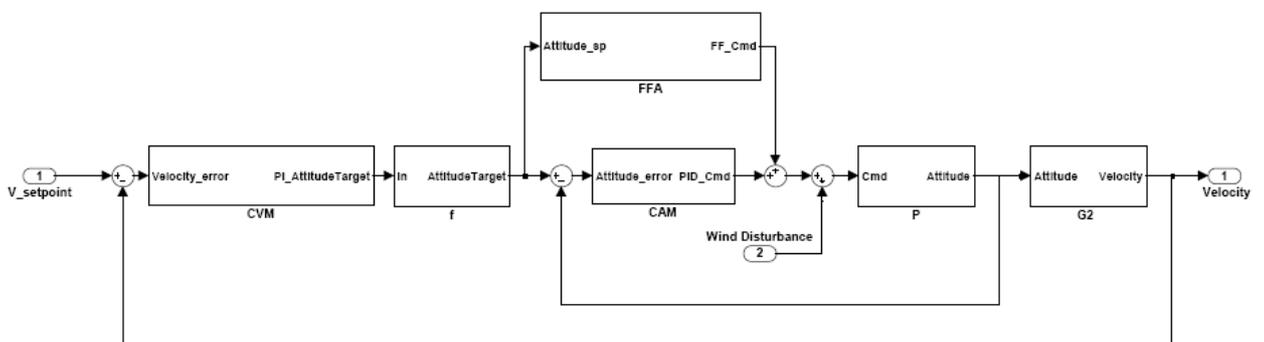


Figure 46: FF controller architecture with Wind Disturbance modeled as command input

It can be demonstrated that, referring for example to longitudinal control system

reported in Fig.46 the transfer function between wind disturbances D and controlled longitudinal velocity U for longitudinal FF control system is:

$$\frac{U}{D} = \frac{G_2 \cdot P}{1 + P \cdot (CAM \cdot (1 + G_2 \cdot CVM \cdot f) + FFA \cdot G_2 \cdot CVM \cdot f)} \quad (80)$$

while for longitudinal Baseline control system:

$$\frac{U}{D} = \frac{G_2 \cdot P}{1 + P \cdot CA \cdot (1 + G_2 \cdot CV)} \quad (81)$$

Where the transfer functions (CA , G_2 , CV , P , etc.) have been already showed in paragraph 4.3

Analogue relations can be found for lateral dynamic.

$$\frac{V}{D} = \frac{G_2 \cdot P}{1 + P \cdot (CAM \cdot (1 + G_2 \cdot CVM \cdot f) + FFA \cdot G_2 \cdot CVM \cdot f)} \quad (82)$$

$$\frac{V}{D} = \frac{G_2 \cdot P}{1 + P \cdot CA \cdot (1 + G_2 \cdot CV)} \quad (83)$$

Where this time the transfer functions (CA , G_2 , CV , P , etc.) refers to the lateral dynamic.

Bode gain diagrams of disturbance-velocity transfer function are reported in figure below for Baseline and FF control systems tuned with basic gains (table 10) both for longitudinal and for lateral case.

It must be noticed that the feedforward term improves the rejection to disturbances especially for those disturbances with frequencies between about 1 and 3 Hz, while for the other frequencies the behavior of the FF controller is pretty much the same of the Baseline one.

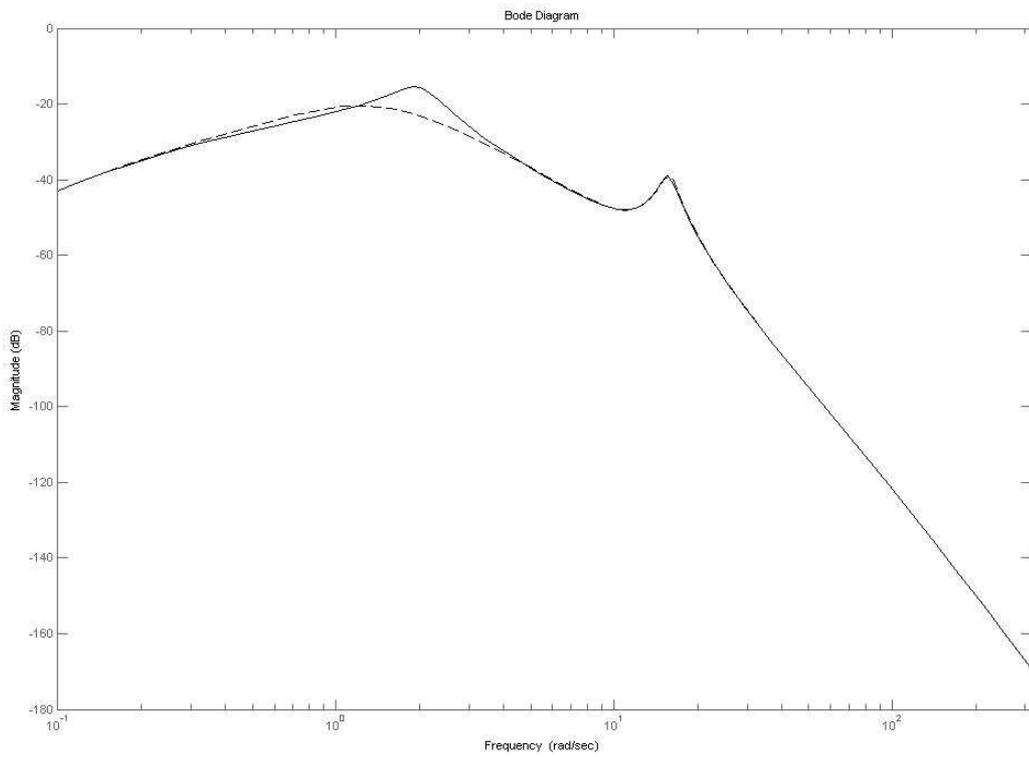
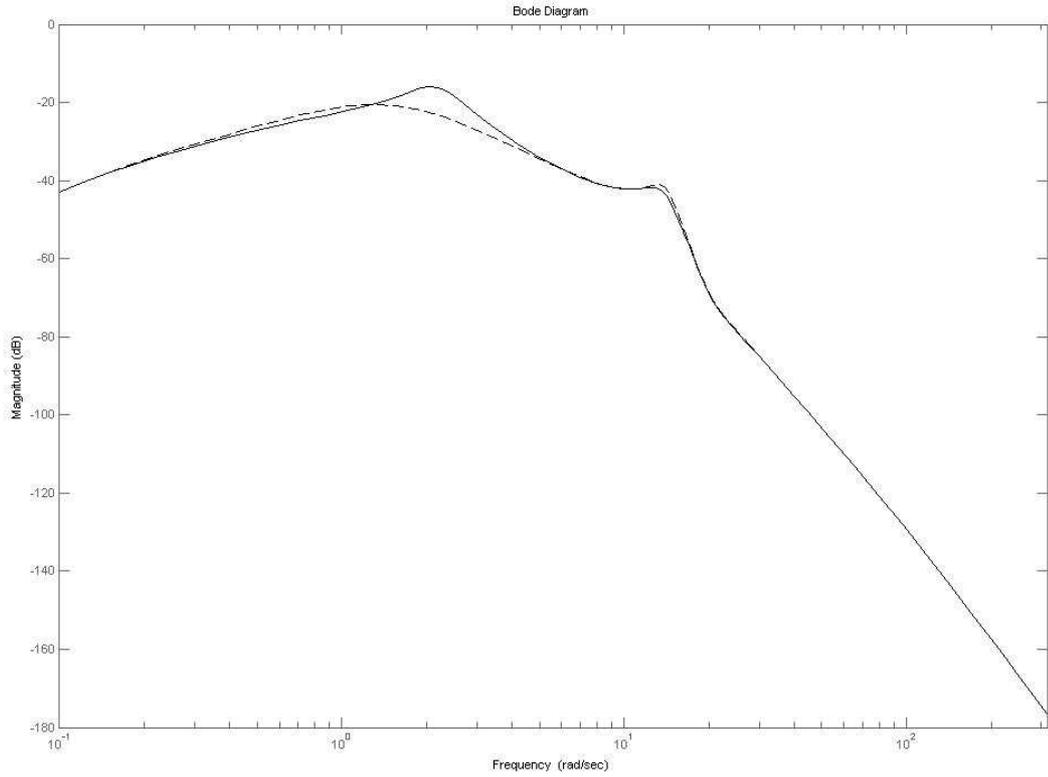


Figure 47: Bode diagrams of Baseline and Feedforward (dashed) disturbance-to-velocity transfer functions. Longitudinal (upper), Lateral (below)

7.3 Time domain comparison with basic gains set

Next figure represents a simulation in time domain of a unitary step response performed by the Baseline (dotted line) and by the FF (continuous line).

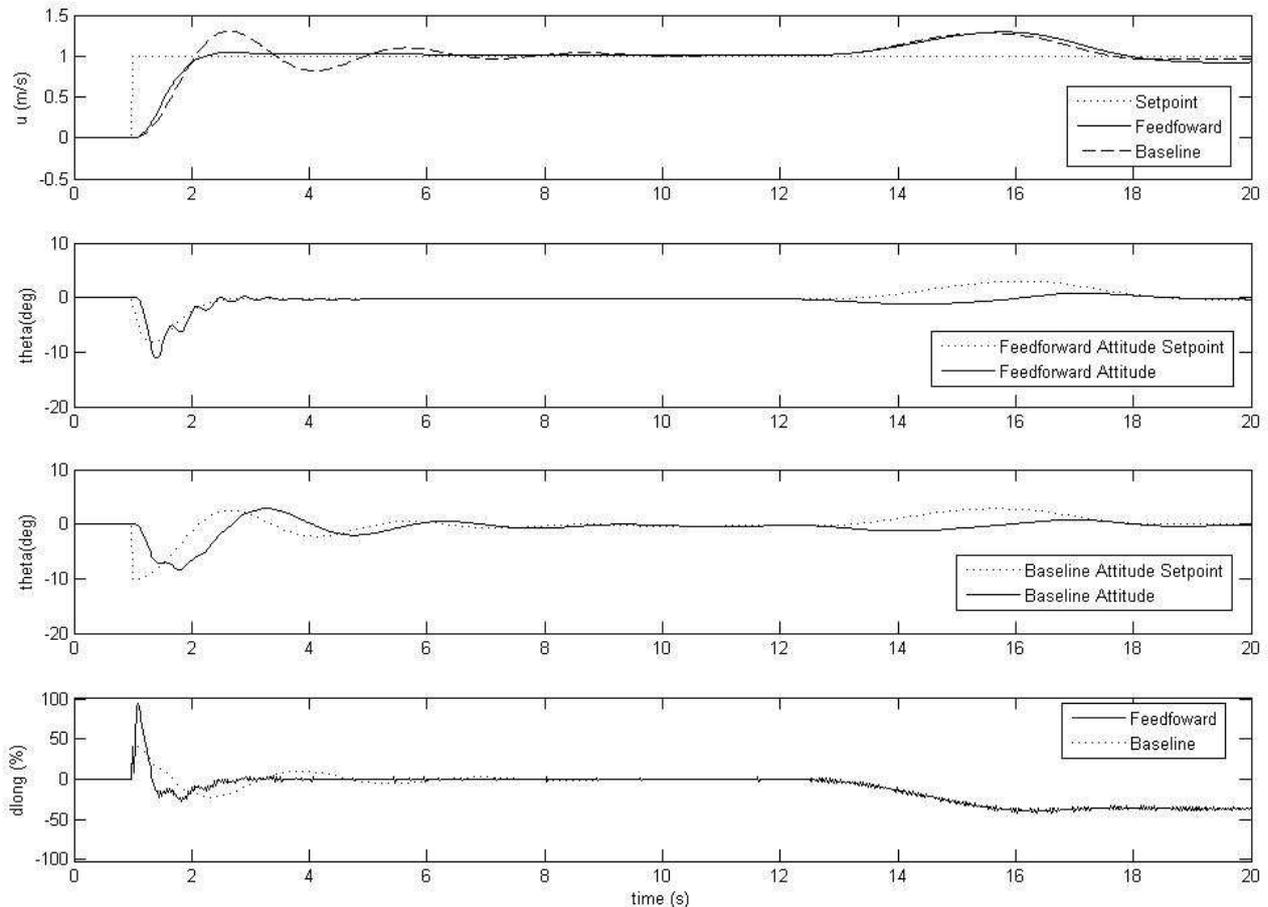


Figure 48: Simulated responses to a unitary longitudinal velocity set point of Baseline and Feedforward controllers tuned with the same gains values

It can be easily seen how the FF response presents a highly reduced overshoot. Remembering that the two controllers share the same gains for PID, it means again that they only differ for the presence of the FFA (FeedForward Action) in the FF inner loop. Hence, the presence of the FFA term makes the FF inner control loop to work with smaller errors in FF controller than in Baseline (see Fig. 39 , second , third row, attitudes) and this highly reduce overshoot and settling time. Moreover it can be noticed how the FFA terms reduce the oscillations due to the lightly damped rotor-fuselage coupled dynamic (mettler articolo optimization) Finally it can be observed that, anyway, the rise time is very similar for the two controllers if measured at the 90% of the step.

In the same simulation reported in figure 39 it was added a gust disturbance starting from simulation time $t=14$ s, it must be noticed how the behavior of the two controller is almost identical on the response to this disturbance. The disturbance added to longitudinal command was, in this case, modelled as a discrete 1-cosine gust in vertical direction with amplitude of 7 m/s that corresponds to a command disturbance of about 5 deg. computed at $0.75 R$ (Rotor radius $R=0.92$ m) and considering a angular rotor speed of about 1200 rpm.

8 Hardware in the Loop (HIL) Test bench

To assess the performance of the two different controllers, a Hardware In the Loop test bench was developed and its architecture is illustrated in figure 49 (right side).

It is composed by:

- A CRIO, equal to the onboard one, which runs the control software;
- an FPGA module which acquires PWM commands (PXI 7831);
- a computer which emulates the helicopter plant and the onboard sensor outputs.

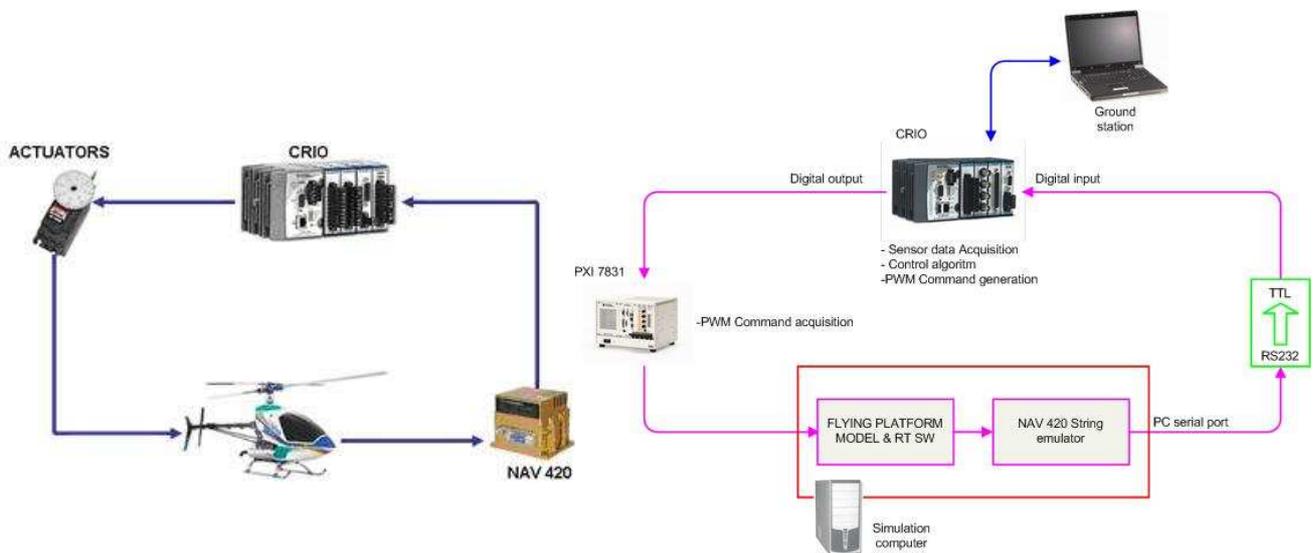


Figure 49 Real control loop (left) and Hardware in the loop test bench (right)

CompactRIO digital outputs, that usually drive the servo actuators, are acquired, in the HIL test bench, by the PXI FPGA module and converted in degrees of servo control actuation. On the simulation computer a NI Labview software implements the state space and noise models illustrated before. That module computes, in real time, the helicopter response due to control input. A NAV420 emulator simulates the original serial data packet format and is used to send information to the main controller.

Since PC serial port uses RS232 signals and CRIO digital inputs accept TTL voltage levels, an integrated circuit board has been placed between computer output and CRIO input.

The helicopter ground station can be connected to the HIL for sending to the controller the desired speed profile, or any kind of commands, and to save helicopter outputs: these outputs are then used for assessing the controller performances.

The HIL global model is composed of a state space model which predicts only the signal frequencies under 5Hz and of an Auto Regressive (AR) model which simulate the remaining frequencies between 5Hz and 50Hz.

9 Baseline and FF controllers parameters automatic tuning

In order to make an impartial analysis of controllers goodness, in terms of stability and performances, an automatic tuning strategy, in house developed and based on step response characteristics, has been adopted. In this way, Baseline and Feedforward controllers have been automatically tuned in order to have same performances in the response to a unitary (1 m/s) velocity step.

Table 2 reports the adopted constraints values .

| | Rise Time | % Rise | Settling Time | % Settling | % Overshoot | % Undershoot |
|-----------------------------|--------------|--------|---------------|------------|-------------|--------------|
| Longitudinal/Lateral | 1.0 /1.2 (s) | 90% | 2.5 | 2% | 2% | 2% |

Table 12: Response characteristics for longitudinal and lateral dynamics

The meaning of the constraints reported in table 12 is more clearly defined in figure 50.

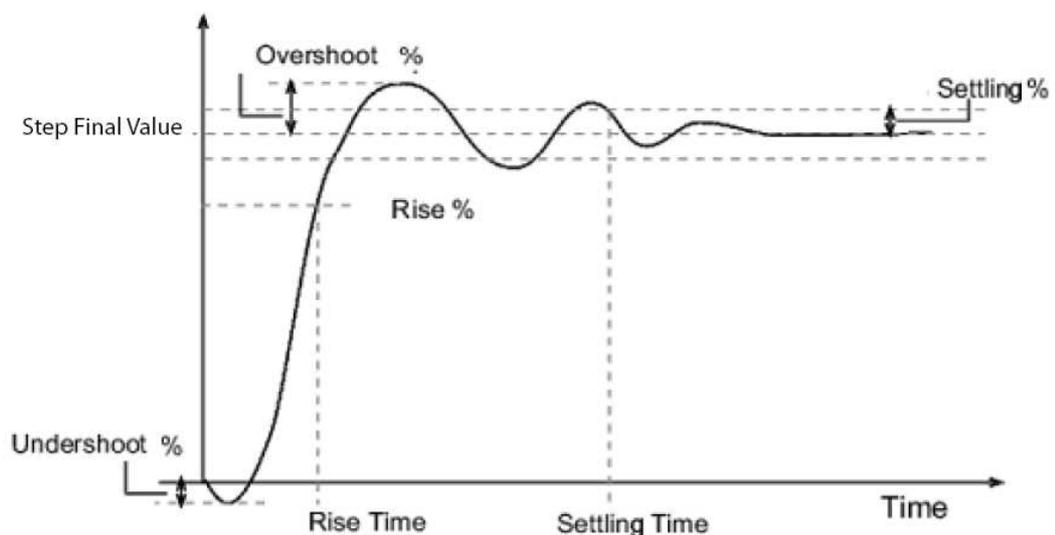


Figure 50: Unitary step response characteristics

Referring to table 12, it has to be noticed that a very small overshoot requirement has been chosen to fulfill the ADS33 [24] hover and low speed specification (cit. *“There shall be*

no noticeable overshoots in the response of translational rate to control”) and that settling percentage has been set to the same value of overshoot in order to obtain a first order response as prescribed by the norms.

With the adopted automatic tuning procedure and the above constraints it has been possible to find the controllers gains reported in table 13.

| | Baseline Longitudinal | Baseline Lateral | FF Longitudinal | FF Lateral |
|------------------------------|------------------------------|-------------------------|------------------------|------------------------|
| Attitude Proportional | $K_p = -2.0062$ | $K_{pLat} = 2.4$ | $K_{pm} = -1.0336$ | $K_{pmLat} = 1.9068$ |
| Attitude integral | $K_i = -4.5837$ | $K_{iLat} = 1.44$ | $K_{im} = -2.1015$ | $K_{imLat} = 1.2618$ |
| Attitude derivative | $K_d = 0$ | $K_{dLat} = 0.06$ | -- | -- |
| Velocity Proportional | $K_{pv} = -11.3730$ | $K_{pvLat} = 7.9685$ | $K_{pvm} = -9.5234$ | $K_{pvmLat} = 9.5498$ |
| Velocity integral | $K_{iv} = -0.6914$ | $K_{ivLat} = 0.410$ | $K_{ivm} = -0.3864$ | $K_{ivmLat} = 0.3442$ |
| Velocity derivative | $K_{dv} = -1.1017$ | $K_{dvLat} = 0.0077$ | -- | -- |
| Filter time constant | -- | -- | $T_{filt} = 0.1117$ | $T_{filtLat} = 0.2187$ |

Table 13: Controllers parameters computed under performances constraints

10 Comparison of equal-performances controllers

In this chapter the comparison between the Baseline and FF controllers will be performed by using the tuning parameters that have been computed in previous chapter. These parameters allow to achieve very similar performances and have been automatically tuned under the performances constraints that have been showed in table 12 of previous section. It appears obvious that comparing two systems with very similar performances, the best of the two controllers is the one that maintains the greatest stability margins.

10.1 Stability analysis

Stability analysis has been performed by computing Gain and Phase stability margins [1,23] for the Baseline and the Feedforward control systems. The two Loop Gain transfer functions for longitudinal and lateral controllers have been derived from control schemes depicted in fig. 9 and fig. 10 and automatically computed tuned parameters have been used

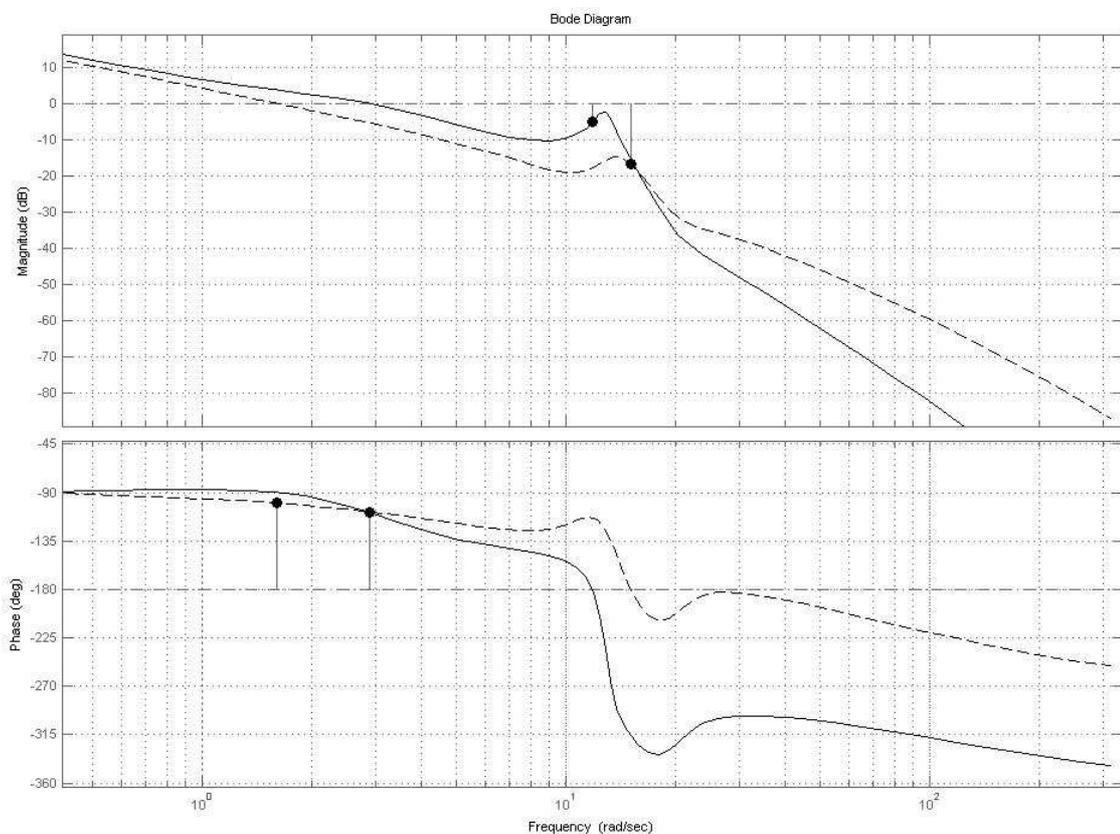


Figure 51: Bode Diagrams for Stability analysis, Longitudinal dynamic: Baseline (continuous line) and Feedforward (dashed line) controller

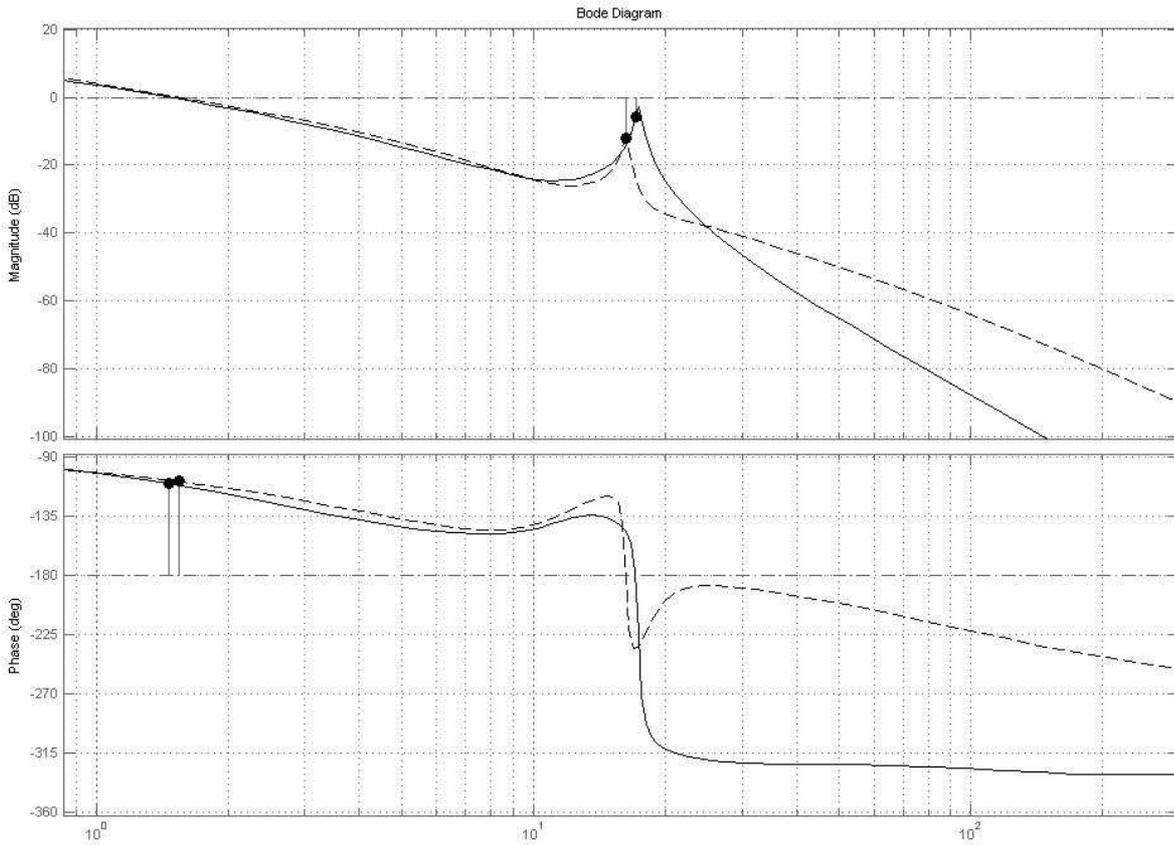


Figure 52: Bode Diagrams for Stability analysis, Lateral dynamic: Baseline (continuous line) and Feedforward (dashed line) controller

Bode magnitude and phase diagrams are reported in figure 52 for Baseline Loop Gain (BLG) and the FF Loop Gain (FLG) transfer functions both for longitudinal (left) and lateral (right) dynamics in which stability margins have been indicated by means of circular markers. As can be easily noticed, they appear to be very similar to those reported in literature [1].

Looking at these figures it can be seen that, in both cases, the feedforward architecture assures an improvement of phase and stability margins. In fact, for the Baseline controllers the critical frequencies for stability (11.8 rad/sec for the longitudinal, 17.1 rad/sec for the lateral dynamic) almost coincide with the natural frequency of the lightly damped coupled rotor/stabilizer/fuselage group caused by the stabilizer bar (12.1 rad/sec for the longitudinal and 18 rad/sec for the lateral dynamic [23]) and this brings to a great reduction in the gain margin.

The effect of the FF compensation brings to a reduction in the lightly damped coupled rotor/stabilizer/fuselage influence and, hence, to an improvement of gain margin.

Gain and Phase stability margins are finally reported in table 14.

| | | Gain margin (dB) | Phase margin (deg) |
|--------------------|---------------------|--------------------------|---------------------------|
| Baseline | Longitudinal | 5.24 dB (@ 11.8 rad/sec) | 71.6 deg (@ 2.88 rad/sec) |
| | Lateral | 5.54 dB (@ 17.1 rad/sec) | 69.8 deg (@ 1.47 rad/sec) |
| Feedforward | Longitudinal | 16.9 dB (@ 15 rad/sec) | 80.2 deg (@1.61 rad/sec) |
| | Lateral | 12.1 dB (@ 16.2 rad/sec) | 71.6 deg (@1.54 rad/sec) |

Table 14: Stability margins of the FF and Baseline controllers for longitudinal and lateral dynamics

It has to be noticed that only FF controller fulfils the specifications for flight control design [24], which require a gain margin of 6 dB and a phase margin of 45 deg, whereas the Baseline control system lacks adequate Gain margin.

This stability analysis, moreover, doesn't take into account noise effects and problems due to quantisation of the analogue signals that slightly affect final performances and that will be analyzed in next section.

10.2 HIL tests

The Baseline and Feedforward control systems tuned with the parameters reported in the previous section, have been tested in the Hardware in The Loop test bench described in fig. 49. The dynamic model has been used coupled with the velocity signal noise model (ref. Chapter 6).

HIL tests results are reported in next figures and confirm the stability analysis described in previous paragraphs. In figure 53 are depicted the axial velocity, pitch attitude (baseline and FF) and the longitudinal command, related to a unitary velocity step for the two controllers. Whereas the controllers have been automatically tuned in order to attain the same velocity performances, a small difference in performance can be anyway observed, like, for example, a smaller rise time for the Baseline velocity response. This is probably due to the automatic tuning procedure that has set up a slightly faster solution for Baseline controller. Another reason for this difference can be found in small differences between the Simulink dynamic and control models used during tuning sessions and in the HIL test bench that is entirely coded in Labview. Contrary result occurred in lateral controller tuning where Baseline controller resulted slightly lower than FF but, anyway, the differences between the velocity responses are small and they do not invalidate our comparison.

In the second and third strips of the same figure, it can see that the Baseline attitude and actuation command signals oscillate, whereas the feedforward ones are much more stable. This appears to be consistent with the consideration reported in previous paragraph, about the smaller gain margin of Baseline system.

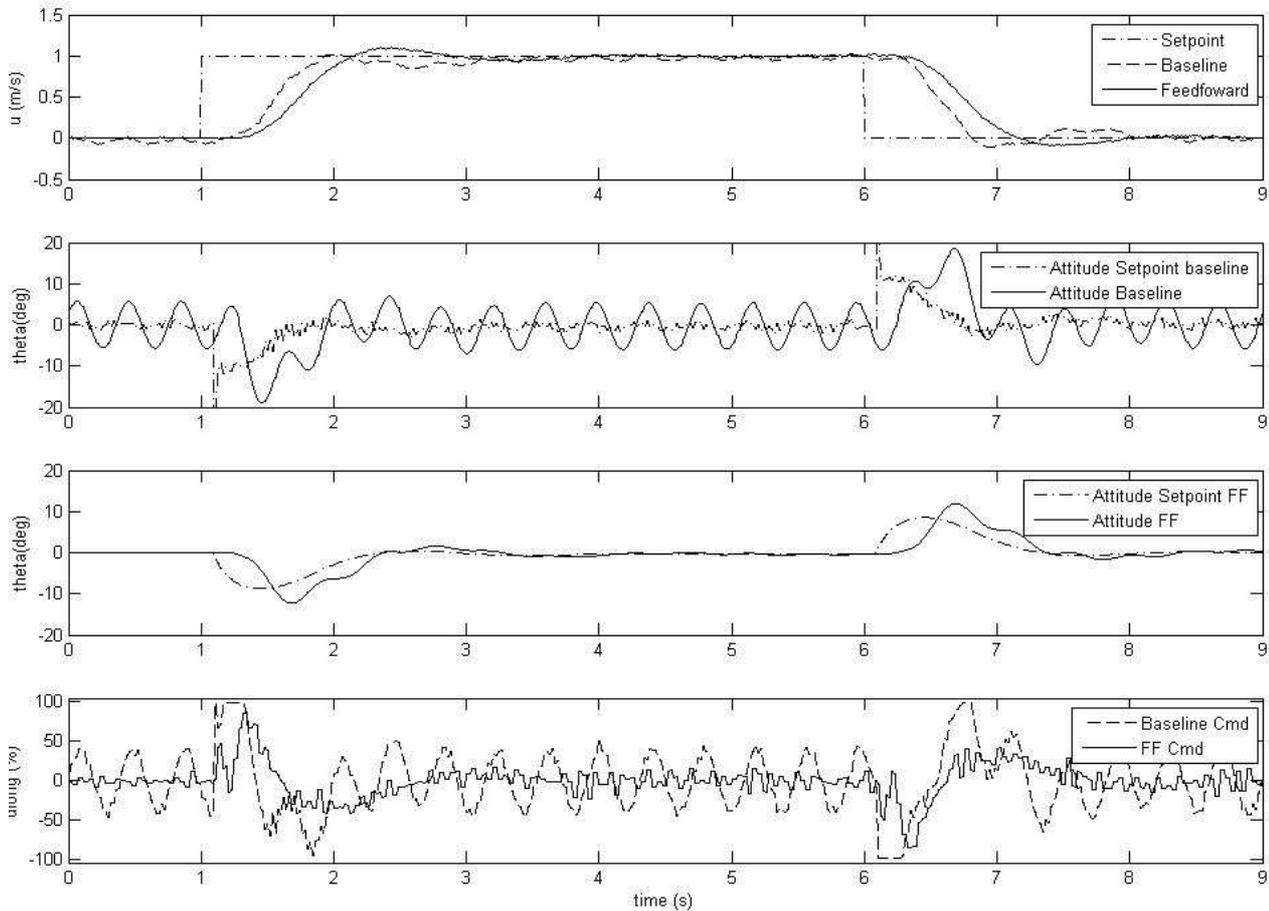


Figure 53: HIL simulation; Longitudinal velocity 1 m/s step

Same considerations can be applied to the 5 m/s forward and lateral velocity steps (fig 54 and 56) and to 1 m/s lateral velocity step (fig 55). In the 5 m/s step velocity cases (fig 54 and fig.56), it has to be noticed that the feedforward controller achieves a reduced overshoot amount even if the Baseline can count on the Derivative term of the PID. This can be explained considering that, during transients, the FF structure allows smaller errors in the inner attitude control loop.

Finally, in all performed tests, it has been observed a less oscillating behavior in the feedforward controller responses, justified by the greater stability margins that are, at least in this case, well better than the norm requirements [24].

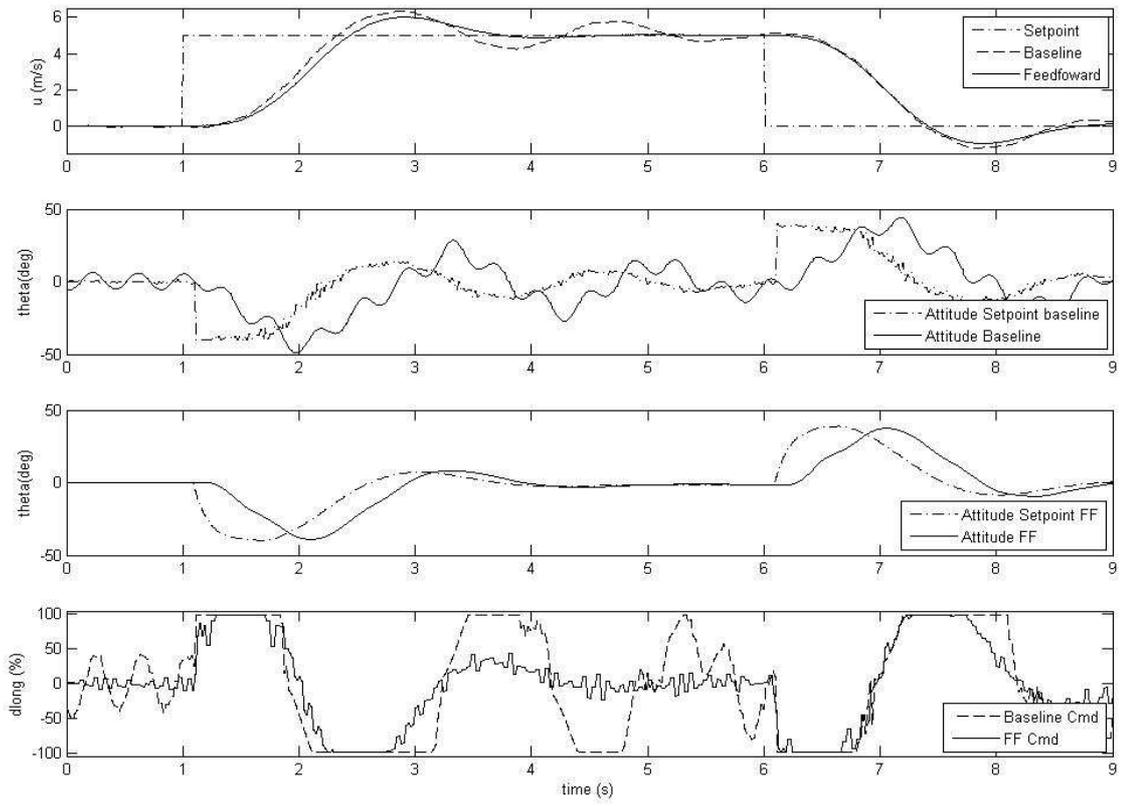


Figure 54: HIL simulation; Longitudinal velocity 5 m/s step

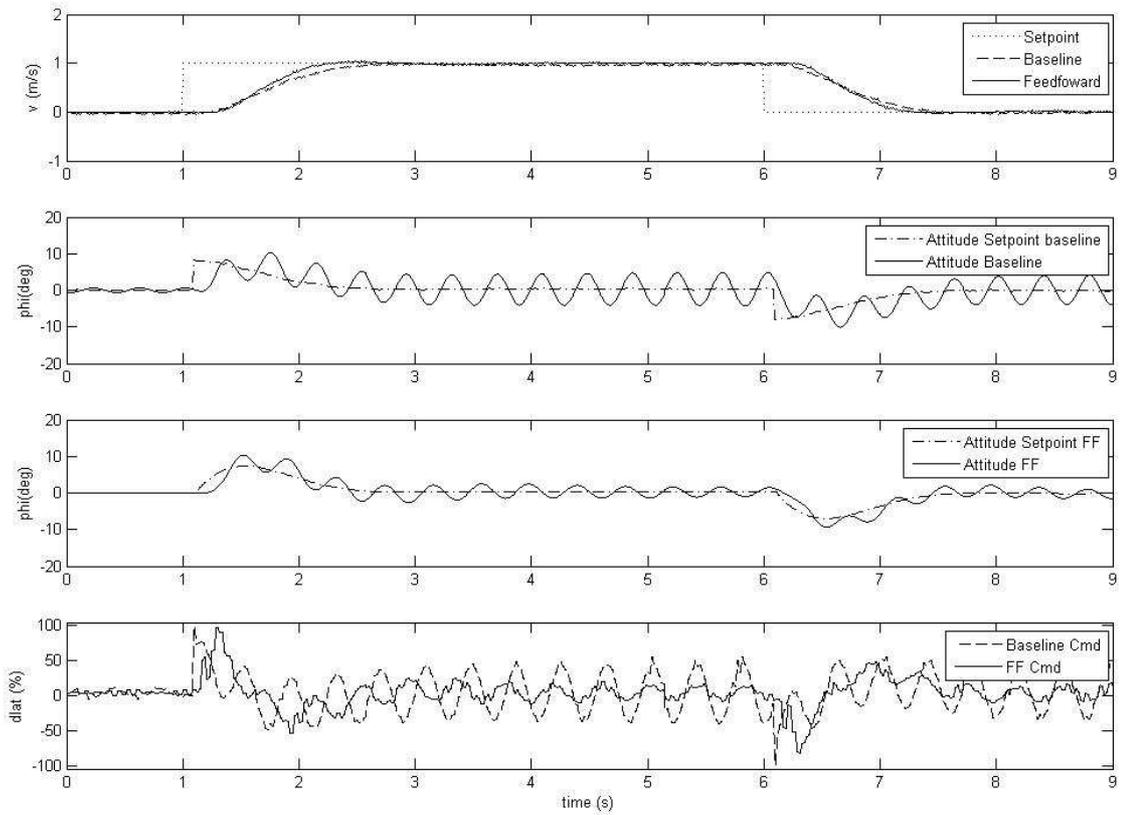


Figure 55: HIL simulation; Lateral velocity 1 m/s step

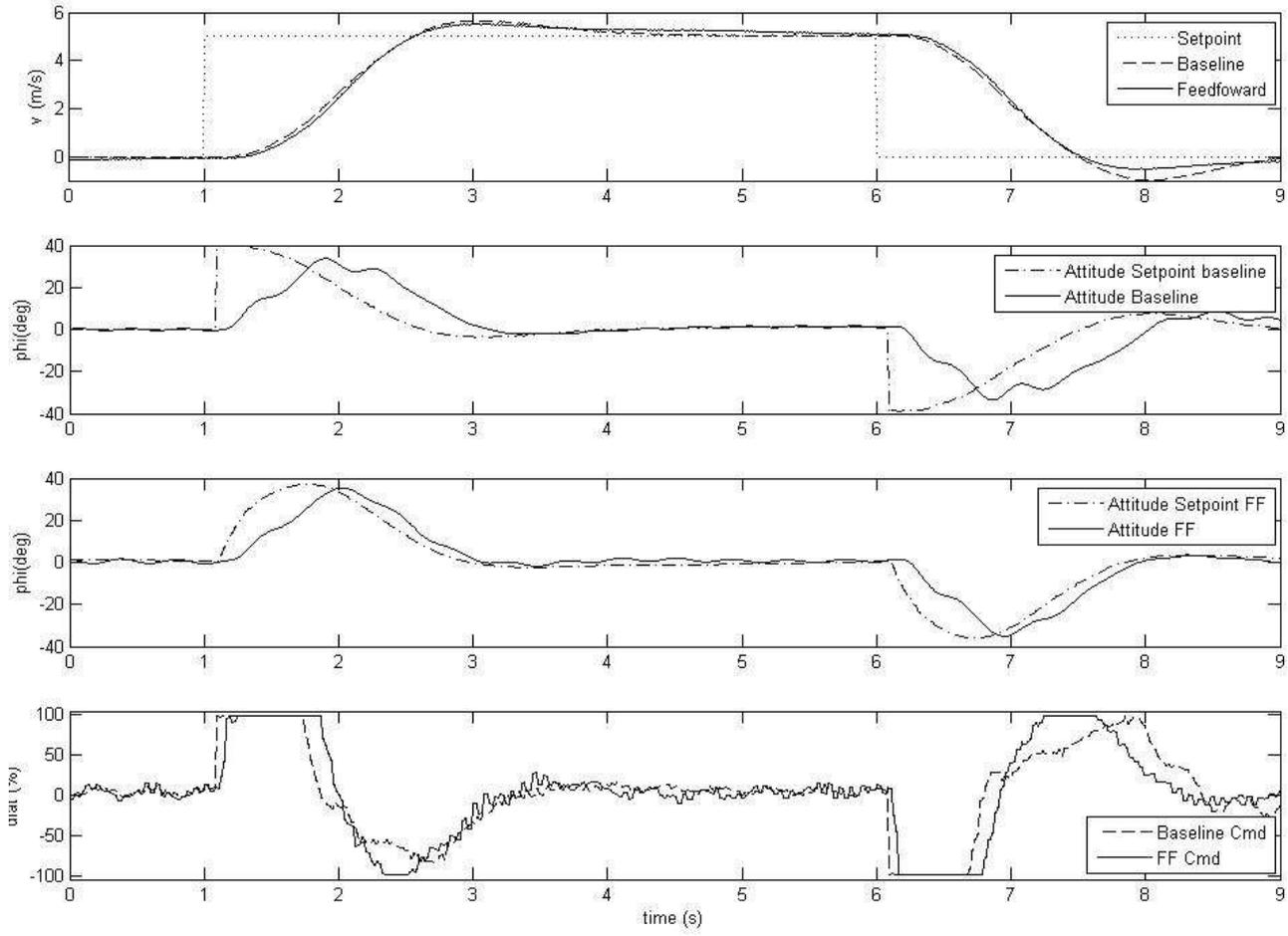


Figure 56:HIL simulation; Lateral velocity 5 m/s step

11 Sensitivity analysis of Stability upon plant identified parameters

The Feedforward compensators, both for longitudinal and lateral dynamics, have been designed starting from the knowledge of the identified command-attitude transfer functions. Thus, it is very important to rely on a very good identification model and to understand how much stability and performances of feedforward controllers are depending upon the goodness of identified parameters. In order to assess this relationship, we studied how much the stability margins change as a consequence of drifts of identified parameters from their nominal values (identification values).

Considering, for example, longitudinal dynamics, we have seen in paragraphs 4.2 and 4.8 that it necessary to know the values of following identified key parameters:

$$A_{lon}, \omega_{nq}, \tau_e$$

in order to compute the longitudinal feedforward term (FFA).

In this paragraph, we will check, using a proprietary Matlab script, what happens to stability margins and to other performances indexes when each of the three parameters change of +/-20% in total independent way. In this way we wish to simulate the case that a +/-20% error was done in the identification of each key parameter. Obviously, during each test, the plant identified model remains the same (identified parameters equal to nominal values) and only the controllers key parameters are increased or diminished of 20% of their nominal values.

Following conventional names will be adopted for key parameters degraded values, these names are obtained by adding a plus or minus symbol to key parameters name in order to refer to the increased or diminished value. For example:

$$A_{lon+} = A_{lon} + 20\% \cdot A_{lon} \quad (84)$$

$$A_{lon-} = A_{lon} - 20\% \cdot A_{lon} \quad (85)$$

Referring to Longitudinal dynamic, nominal values for key parameters are:

$$A_{lon}=0.2488$$

$$\omega_n=12.1$$

$$\tau_e=0.132$$

Following table has been computed considering all the possible combinations between increased and diminished key parameters values:

| Test N° | A_{lon} | ω_n | τ_e | GM | PM | Stability |
|---------|------------|---------------|-------------|------------------------|-----------------------------|-----------|
| Nominal | A_{lon} | ω_n | τ_e | 16.9 dB (@ 15 rad/sec) | 80.2 deg (@1.61 rad/sec) | Yes |
| 1 | A_{lon-} | ω_n- | τ_e- | 10.2177 dB (@15.2402) | 78.6373 deg.(@1.5372 rad/s) | Yes |
| 2 | A_{lon-} | ω_n- | τ_{e+} | 9.8140 dB (@15.7979) | 93.7258 deg.(@1.3224 rad/s) | Yes |
| 3 | A_{lon-} | ω_{n+} | τ_e- | 17.1225 dB (@13.5070) | 80.5648 deg.(@1.6152 rad/s) | Yes |
| 4 | A_{lon-} | ω_{n+} | τ_{e+} | 16.1279 dB (@13.0511) | 96.9602 deg.(@1.3732 rad/s) | Yes |
| 5 | A_{lon+} | ω_n- | τ_e- | 14.1145 dB (@15.3074) | 62.5827 deg.(@1.7712 rad/s) | Yes |
| 6 | A_{lon+} | ω_n- | τ_{e+} | 13.6220 dB (@15.8632) | 79.4297 deg (@1.5846 rad/s) | Yes |
| 7 | A_{lon+} | ω_{n+} | τ_e- | 22.0459 dB (@13.4606) | 62.9814 deg.(@1.8348 rad/s) | Yes |
| 8 | A_{lon+} | ω_{n+} | τ_{e+} | 20.5392 dB (@12.7534) | 80.8309 deg.(@1.6456 rad/s) | Yes |

Table 15: Stability margins of the FF (longitudinal) with deviation of feedforward key parameters of 20%

Stability of the feedforward controller is reported in the table above in the rightest column and it has been computed as a Boolean that is true if all real parts of close loop transfer function poles are all positive numbers. We can see that the FF system is still stable for a variation of +/-20% of key parameters from nominal values. Obviously, the same thing could have been derived also from the fact that Gain and Phase margins are still positive, but it has been done in this way to have a redundant check.

Even some combinations (e.g. the n.3) appears to be more stable (higher gain margin, same phase margin), it can be easily assessed that the time response of the FF controller with these key parameters set is less performing than the nominal FF controller. In fact, looking at fig.57 it can be seen that the degraded FF system (point-dash line) has higher rise time and higher assessment time. This is due to the fact that PI gains were not optimized in each every different test (1-8), but the same PI gains values that were optimized for the nominal case were adopted.

From figure 58 we can also see better how FF nominal controller presents no overshoot at all and, moreover, that differences in performances appear to be small and,

hence, that degraded FF controller seems to be still a good controller.

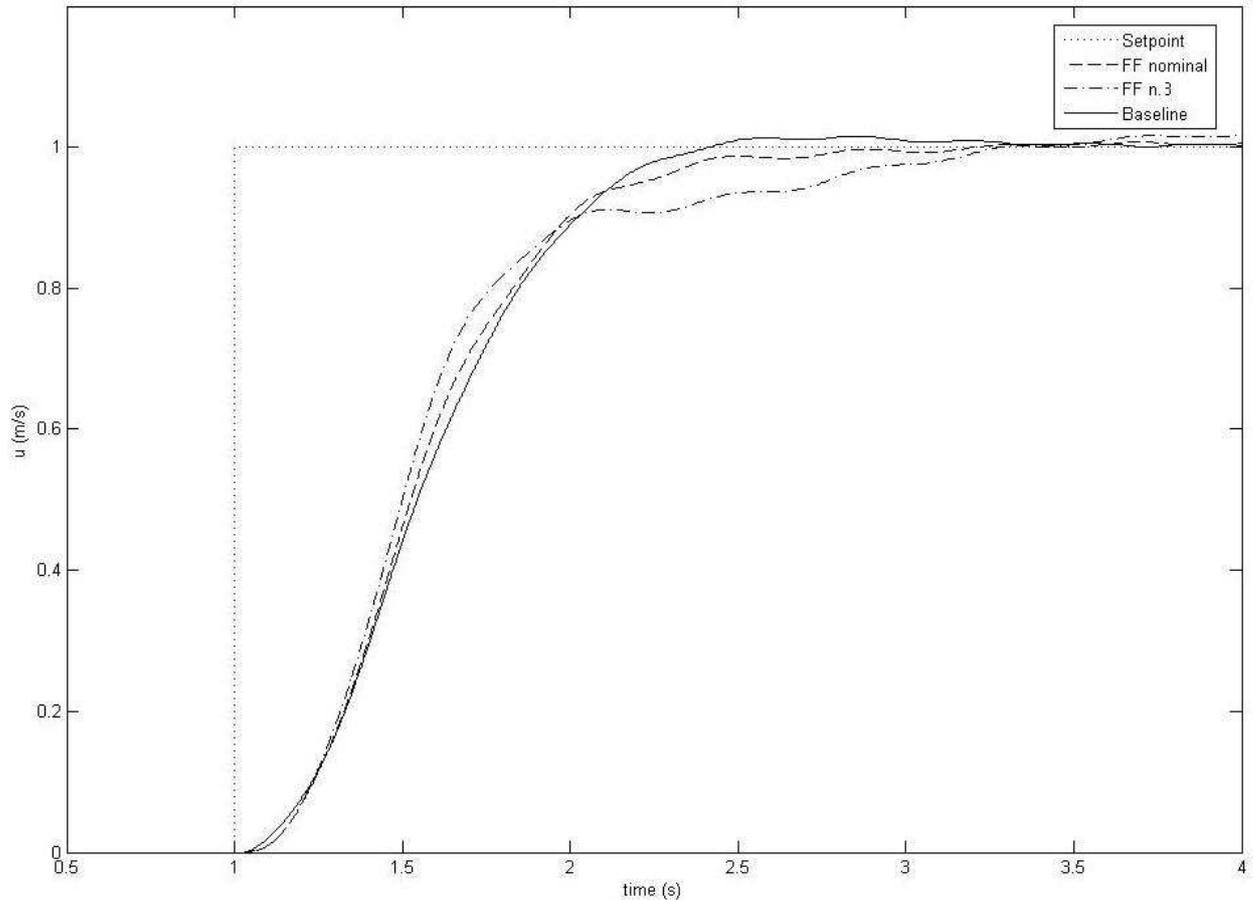


Figure 57: Baseline, FF (testcase n.3) responses to a unitary longitudinal velocity step

In the following table the same kind of analysis have been reported for lateral dynamic stability margins computation:

| Test N° | B_{lat} | ω_{np} | τ_e | GM | PM | Stability |
|---------|------------|----------------|-------------|--------------------------|--------------------------|-----------|
| Nominal | B_{lat} | ω_{np} | τ_e | 12.1 dB (@ 16.2 rad/sec) | 71.6 deg (@1.54 rad/sec) | Yes |
| 1 | B_{lat-} | ω_{np-} | τ_{e-} | 16.6759 dB (@17.4863) | 71.8338 deg (@1.5218) | Yes |
| 2 | B_{lat-} | ω_{np-} | τ_{e+} | 28.3864 dB (@25.1427) | 85.0821 deg (@1.5910) | Yes |
| 3 | B_{lat-} | ω_{np+} | τ_{e-} | 8.1766 dB (@16.0879) | 71.6584 deg (@1.5481) | Yes |
| 4 | B_{lat-} | ω_{np+} | τ_{e+} | 5.1997 dB (@16.0519) | 84.7408 deg (@1.6300) | Yes |
| 5 | B_{lat+} | ω_{np-} | τ_{e-} | 26.2649 dB (@18.9148) | 62.1976 deg (@1.5438) | Yes |
| 6 | B_{lat+} | ω_{np-} | τ_{e+} | 32.7217 dB (@26.0324) | 71.6292 deg (@1.5372) | Yes |
| 7 | B_{lat+} | ω_{np+} | τ_{e-} | 13.2472 dB (@15.9932) | 62.1120 deg (@1.5595) | Yes |
| 8 | B_{lat+} | ω_{np+} | τ_{e+} | 9.7695 dB (@15.9841) | 71.5565 deg (@1.5553) | Yes |

Table 16: Stability margins of the FF for lateral dynamics varying feedforward key parameters of 20%

Also for lateral dynamic, it can be observed that the system is still stable under all the possible combinations of 20% deviated key parameters.

Finally maximum allowable Deviations have been computed for the FF control system:

- Longitudinal Dynamic :Evaluation of max allowable Deviations for Norms and Stability:
 - 42% for Stability
 - 29% for ADS33 compliancy (GM>6dB, PM> 45 deg)

- Lateral Dynamic :Evaluation of max allowable Deviations for Norms and Stability:
 - 38% for Stability
 - 17% for ADS33 compliancy (GM>6dB, PM> 45 deg)

The FF controller appears to be stable till a deviation of key parameters of about the 42% while for lateral maximum parameters deviation without instability is 38%.

This is an important achievement as it means that even in presence of great drifts in system dynamics (e.g. due to payload variations, fuel consumption) the feed-forward controlled system remains stable.

Moreover, the maximum deviation of key parameters from their nominal values that assures that FF system has Phase Margins and Gain Margins greater than the minimum values specified by MIL-F-9490, have been valuated in the 29% for Longitudinal dynamic and in 17% for lateral dynamic. In presence of greater deviations, the critical Gain Margin limit of 6 dB was exceeded both for longitudinal and lateral dynamics.

This appears, again, as a good achievement because it means that even in presence of sensible drifts of the key parameters of the system and even if the controller gains are kept to their nominal values, the control performances will be still compliant to norms prescriptions in terms of stability margins.

12 Performances comparisons in Navigation

In this paragraph, HIL simulations of navigation tests are presented. The aim of these tests is to assess the influence of the control system architecture upon the navigation performances in terms of deviation of the rotorcraft position from the target trajectory.

The Baseline and FeedForward Controllers are tuned with the gains of table 13 in chapter 9 and, hence, the two controllers should have similar performances to unitary step, these performances are those reported in table 13 of chapter 9.

Next figure reports a comparison of tracking performances of the two controllers in a '8-like' circuit. In this test the longitudinal target velocity is kept at the constant value of 2 m/s.

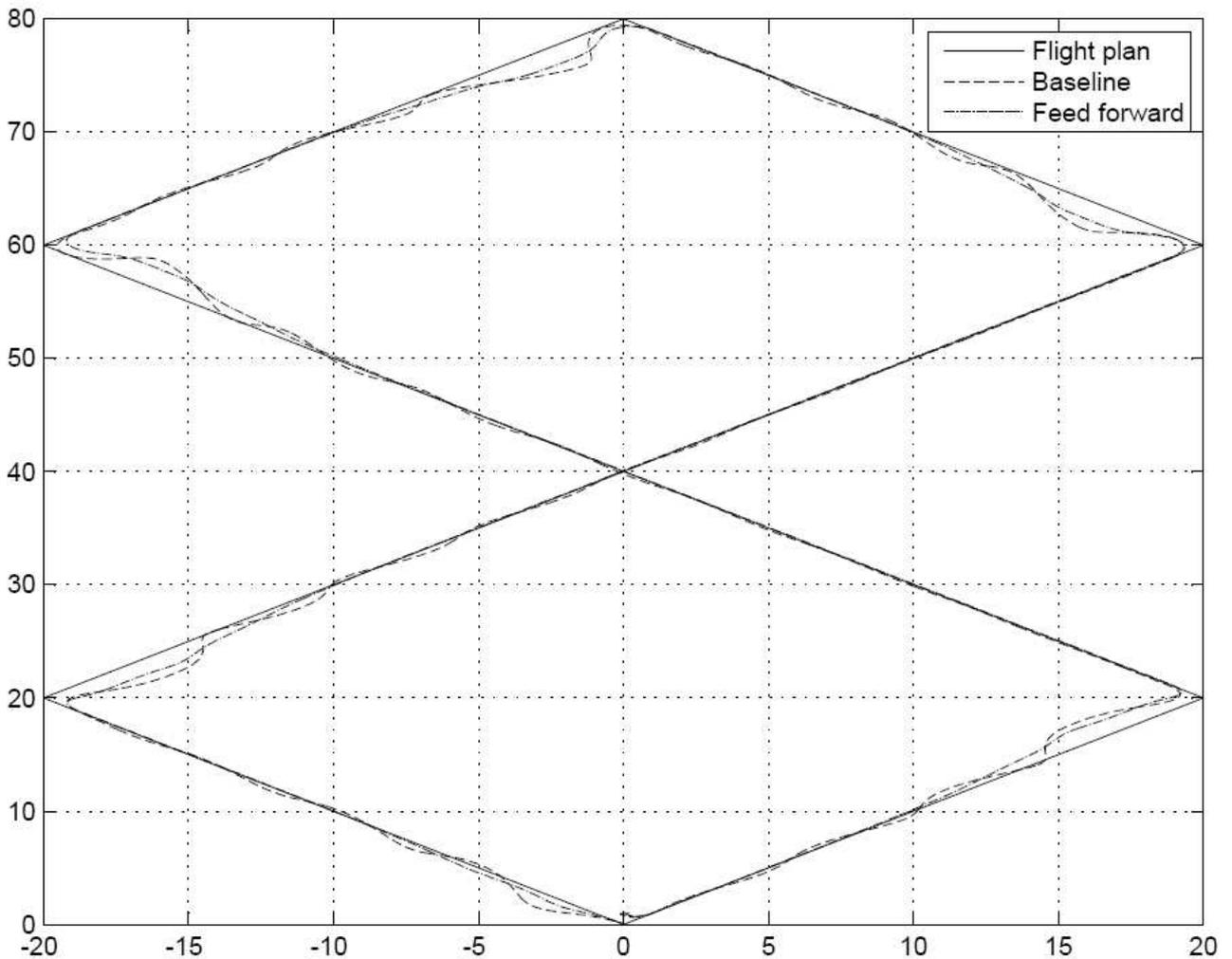


Figure 58: Comparison: HIL simulations of '8-like' circuit with velocity of 2 m/s

The flight path is the same already reported in table 2, and ,hence, the tracking velocity is equal to 2 m/s but in this test the yaw orientation has been kept fixed to 0 deg (North).

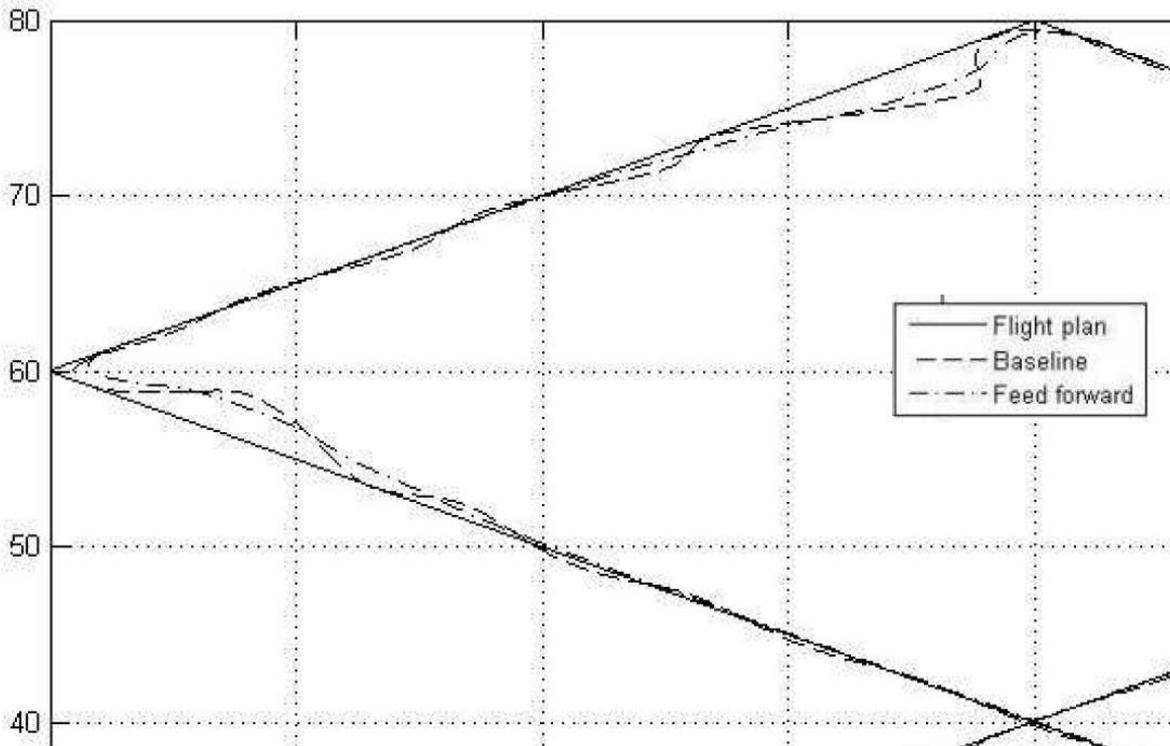


Figure 59: Zoom of figure 58

Figure 59 shows a zoomed view of figure 58. Here it can be easily noticed that the Feedforward controller improve the tracking performances of navigation, reducing the tracking oscillations that are much more present in Baseline system. Moreover here it can be observed that the feedforward controller almost never brings the rotorcraft beyond the subsequent trajectory line since it allows a faster change of trajectory. In fact, even if the change of setpoint trajectory occurs, for both the controllers, at the same distance from the subsequent waypoint, as the Baseline exhibit a slower and more oscillating response whereas the Feedforward change of trajectory appears much more rapid and hence seems to anticipate the target trajectory.

Last figure of this chapter reports a comparison test performed with a target velocity of 4 m/s in a '8-like' circuit. As in previous test the yaw orientation has been kept fixed to North direction for the entire path. In this case it becomes apparent how the Baseline Tracking control becomes almost unstable and, as a consequence, its performances are of course unacceptable.

Remembering that the rotorcraft starts form the point (0,0) toward the (20,-20) waypoint, it is possible to track the trajectory for the Baseline controller; it seems that the biggest issues occurs when the RUAV leave the waypoint (20,-20) towards the waypoint of

coordinates (40,0). The Baseline RUAV begins to track circles around the waypoints and gradually loose its path.

The feedforward controller, instead, attains performances very similar to those of the 2m/s '8-like' circuit reported in figure 58 and 59.

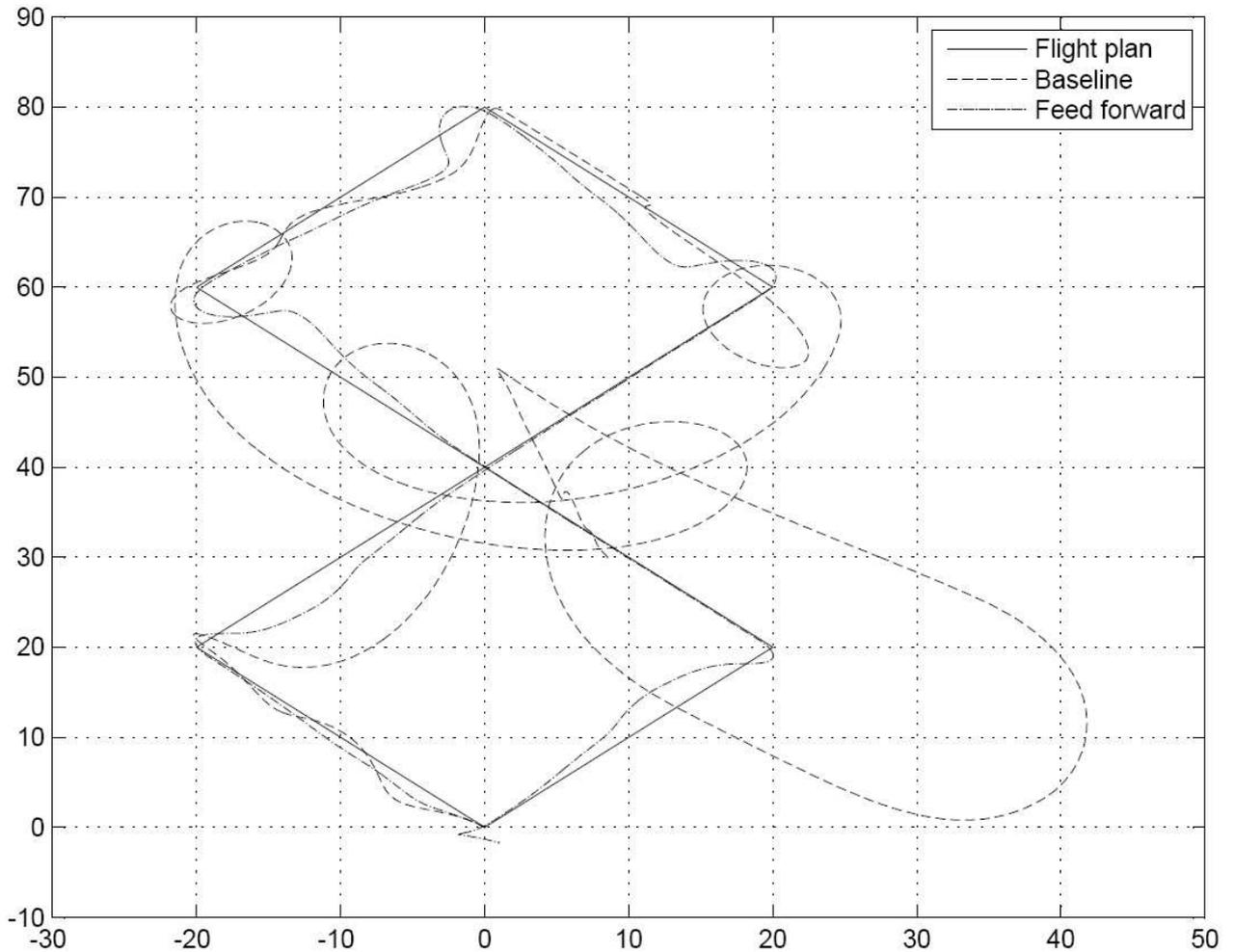


Figure 60: Comparison: HIL simulations of '8-like' circuit with velocity of 4 m/s

Of course the implemented fixed heading navigation is very simple and could be improve, and perhaps, navigation gains tuning could be improved as well, but, in our opinion, what should be focused is that the same navigation algorithm, if used in conjunction with two different attitude and velocity control systems, brings to completely different results.

Moreover it must be remarked that the Baseline and FF controller used in these navigation tests were automatically tuned in order to attain same performances in a 1 m/s velocity step. Hence from this latest simulation, it is evident how the Baseline controller presents performances that highly downgrade with the increasing speed of test, whereas the FF seems to achieve good performances also for higher velocities.

13 Conclusions and Outlook

In this thesis latest results of the RUAV activity achieved at Bologna University have been presented. In particular a model-based feed-forward controller designed for a small scale Helicopter has been illustrated and its performances have been assessed and compare to those of a traditional Baseline controller. This Feed-Forward controller has a very simple control architecture based on nested proportional-integral control loops with feedforward compensation in the inner loop. This feedforward term is obtained by the inversion of the command-attitude identified plant models and its smart discrete-time implementation has been shown as well.

The accuracy of the model has been initially verified by showing that it successfully predicts the behaviour of the baseline control system that is currently used for the UNIBO RUAV.

In subsequent chapters, it has been shown how this control combines benefits of feedforward and feedback controllers, where performances are achieved with feedforward action and robustness is achieved with feedback, and how feedforward action make the system to work with smaller errors and therefore with less saturation problems and so makes parameters easier to be tuned.

A good number of comparisons upon performance and stability has been illustrated between the proposed feedforward controller and the baseline controller. The comparisons have been done using, in a first step, the same basic tuning parameters for the controllers, then different sets of parameters has been adopted. These latter parameters have been derived, in order to make an impartial analysis, by mean of an automatic tuning strategy. Moreover, in dedicated section, the sensitivity analysis of the stability margins upon the identified plant parameters has been evaluated and it has been shown how the proposed model based controller remains stable even in presence of big uncertainties on plant identified parameters.

The presented analysis has been done by using a Hardware In The Loop systems with mathematical model of helicopter dynamics identified in low speed flight conditions. This identified dynamic has been derived by using a simple and innovative identification procedure, developed in a Matlab-Simulink environment, in order to obtain a representative dynamic model of a small rotorcraft UAV near hovering flight condition. Preliminary parameters identification based on a open-loop session have been shown both with and without cross-effects; then a parameters refinement through a closed loop identification

technique was performed. For each test, indexes of relative goodness of fit have been presented demonstrating the benefits of the improved model.

In the sequel, the identified model has been used for control design purposes, so it has been possible, using Matlab scripts, to pre-tune controller gains, giving a set of control target performances as raise time, maximum overshoot and final error.

As already mentioned the comparisons upon performance and stability between the proposed FF controller and the Baseline one have been done using set of calibrations that have been automatically tuned in order to make an impartial analysis. Results have shown that, tuning the two systems for achieving the same performances, the feedforward controller works with higher stability margins and, hence, with less oscillating attitudes.

Finally, after illustrating guidance algorithms, a comparison of the tracking abilities of the Baseline and FF controller is presented in a typical '8-like' circuit. In this test it has been demonstrated how the FF controller allows a smoother trajectory tracking as well.

As already mentioned, the presented analysis has been done by using a mathematical model of helicopter dynamics identified in low speed flight conditions. Anyway, since the dynamic model in forward flight conditions can be described by transfer functions of the same kind [1], it would be interesting in the future to asses if the benefits of this control technique are present also in forward flight.

Unfortunately in the last period of this activity the IMU has been found to be faulty and issues has been discovered in velocity signals and, as consequence, it hasn't been possible to perform a comparison of the two controllers in real flight test. A flight trials campaign and a deep investigation upon the performances of the Baseline and FF controller will be done once the IMU has been restored.

14 List of Figures

| | |
|---|----|
| Figure 1: UAV System | 9 |
| Figure 2: UNIBO Rotary wing UAV | 11 |
| Figure 3: National Instruments CRIO Onboard Computer | 13 |
| Figure 4: CRIO Field Programmable Gate Array (FPGA) Structure | 13 |
| Figure 5: CRIO Programming Structure | 14 |
| Figure 6: NAV420CA System Architecture..... | 16 |
| Figure 7: AV reference frame, forces & moments | 17 |
| Figure 8: Block diagram of the Baseline controller for longitudinal or lateral dynamics..... | 25 |
| Figure 9: Block diagram of the FF controller for longitudinal or lateral dynamics | 26 |
| Figure 10: Vertical down velocity control module..... | 29 |
| Figure 11: Vertical down velocity controller tracking performance | 29 |
| Figure 12: Heading controller tracking performance | 30 |
| Figure 13: Engine governor tracking performance..... | 31 |
| Figure 14: Schematic of the inner loop Attitude PID controller implementation | 33 |
| Figure 15: Schematic of the inner loop Attitude PI controller implementation | 36 |
| Figure 16: Altitude controller tracking performance..... | 37 |
| Figure 17: Track reference frame | 38 |
| Figure 18: Guide control strategy..... | 39 |
| Figure 19: Simulink guide implementation | 40 |
| Figure 20: Guide tracking performance – Speed,Altitude..... | 41 |
| Figure 21: Guide tracking performance – Latitude,Longitude..... | 41 |
| Figure 22: Guide tracking performance – X_{track} , Y_{track} , Yaw Rate | 42 |
| Figure 23: Target velocity computation | 44 |
| Figure 24: Target velocity computation | 44 |
| Figure 25: Simulate vs Experimental longitudinal controller tracking performance | 46 |
| Figure 26: Recorded data during autonomous square pattern | 47 |
| Figure 27: Autonomous square pattern RUAV ground track..... | 48 |
| Figure 28: Open Loop On-axis Identification | 50 |
| Figure 29: Longitudinal (left) and Lateral (right) sweeping input commands (below) | 52 |
| Figure 30: Longitudinal (left) and Lateral (right) validation input commands (below)..... | 52 |
| Figure 31: Input commands (below) and relative speeds (upper) for Longitudinal (left) and Lateral (right) velocity model Identification | 53 |
| Figure 32: input commands (below) and relative speeds (upper) for Longitudinal (left) and Lateral (right) velocity model Validation..... | 53 |
| Figure 33: Training (left) and validation (right) data sets for heave dynamic transfer functions | 54 |
| Figure 34: Off-axis Derivatives Open Loop Identification | 56 |
| Figure 35: Off-axis Derivatives Open Loop Identification - Training data set: longitudinal inputs | 56 |
| Figure 36: Off-axis Derivatives Open Loop Identification - Training data set: lateral inputs .. | 57 |
| Figure 37: Open Loop Validation - Longitudinal (left column) and Lateral (right column) ... | 58 |
| Figure 38: Closed Loop Identification logic..... | 59 |
| Figure 39: Comparison between predicted (solid) and recorded (dashed) closed loop Baseline Control Systems responses for longitudinal (column 1, left) and lateral (column 2, right) velocity steps. | 61 |

| | |
|--|----|
| Figure 40: Power spectral density of longitudinal and lateral velocity signals | 64 |
| Figure 41: Block diagram of the FF controller with noise injection..... | 65 |
| Figure 42: Bode plot of velocity - 'noise on attitude' transfer function ($u(s)/na(s)$) | 66 |
| Figure 43: Bode plot of velocity - 'noise on velocity' transfer function ($u(s)/nu(s)$) | 66 |
| Figure 44: Comparison between modeled (dashed) and original (continuous) flight data PSD | 67 |
| Figure 45: cross effects and external disturbance | 70 |
| Figure 46: FF controller architecture with Wind Disturbance modeled as command input.... | 71 |
| Figure 47: Bode diagrams of Baseline and Feedforward (dashed) disturbance-to-velocity transfer functions. Longitudinal (\upper), Lateral (below) | 73 |
| Figure 48: Simulated responses to a unitary longitudinal velocity set point of Baseline and Feedforward controllers tuned with the same gains values | 74 |
| Figure 49 Real control loop (left) and Hardware in the loop test bench (right) | 77 |
| Figure 50: Unitary step response characteristics..... | 79 |
| Figure 51: Bode Diagrams for Stability analysis, Longitudinal dynamic: | 81 |
| Figure 52: Bode Diagrams for Stability analysis, Lateral dynamic:..... | 82 |
| Figure 53: HIL simulation; Longitudinal velocity 1 m/s step..... | 84 |
| Figure 54: HIL simulation; Longitudinal velocity 5 m/s step..... | 85 |
| Figure 55:HIL simulation; Lateral velocity 1 m/s step | 85 |
| Figure 56:HIL simulation; Lateral velocity 5 m/s step | 86 |
| Figure 57:Baseline, FF (testcase n.3) responses to a unitary longitudinal velocity step | 89 |
| Figure 58: Comparison: HIL simulations of '8-like' circuit with velocity of 2 m/s..... | 91 |
| Figure 59: Zoom of figure 58..... | 92 |
| Figure 60: Comparison: HIL simulations of '8-like' circuit with velocity of 4 m/s..... | 93 |

15 Bibliography

- [1] B.Mettler “*Identification Modeling and Characteristics of Miniature Rotorcraft*”, Kluwer Academic Publishers, Boston M.A., USA, 2003
- [2] R. Miller, B.Mettler, O. Amidi, “*Carnegie Mellon University’s 1997 International Aerial Robotics Competition Entry*”, Orlando, FL, June 1997. AUVSI
- [3] C.Castillo, W.Alvis, M.Castillo-Effen, K.VAlavanis, W.Moreno, “*Small scale helicopter analysis and controller design for non-aggressive flights*”, Proceedings of IEEE SMC Conference, Hawaii, 2005
- [4] U.S. Air Force Scientific Advisory Board, “*Unmanned Aerial Vehicles in perspective: Effects, Capabilities and Technologies*”, SAB-TR-03-01, July 2003.
- [5] Sevin C., “*Internal Report on survey of potential applications for rotary-wing UAVs*”, ECDCAPECON Report ID 2.1/2, 25 September 2002
- [6] R. E. Weibel R. J. Hansman, MIT, “*Safety Considerations for Operation of Different Classes of UAVs in the NAS*”, AIAA-2004-6421, AIAA’s 4th Aviation Technology, Integration and Operations (ATIO) Forum, 20 - 22 September 2004, Chicago, Illinois. 138
- [7] R. E. Weibel R. J. Hansman, MIT, “*An Integrated Approach to Evaluating Risk Mitigation Measures for UAV Operational Concepts in the NAS*”, AIAA-2005- 6957, AIAA’s 5th Infoech@Aerospace Conference, 26-29 September 2005, Arlington, Virginia.
- [8] B.Teodorani, “*On the Development of a Rotary Wing UAV Platform: Avionics and Onboard Software Set-Up*”, PhD. Thesis, University of Bologna, 2007
- [9] R.Pretolani, “*Design and development of the navigation and control system for an autonomous helicopter*”, PhD. Thesis, University of Bologna, 2007
- [10] 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.
- [11] B.Mettler, M.Tischler, T.Kanade “*System identification of a small-sized unmanned Helicopter Dynamics*”, American helicopter society 55^o annual forum, 1999
- [12] B.Mettler, M.Tischler, T.Kanade, “*System identification modeling of a small-scaled unmanned Rotorcraft for flight control design*”, Journal of American Helicopter Society, 2001
- [13] V. Gavrillets, B. Mettler, E. Feron, “*Dynamic Model for a Miniature Aerobatic Helicopter*”, M.I.T., 2001

- [14] M.E.Penati, G.Bertoni, “*Automazione e Sistemi di Controllo*”, Vol. II, Società Editrice Esculapio, Bologna, 1997.
- [15] K.Astrom, B.Wittermark, “*Computer controlled systems*“, Prentice Hall, 1997
- [16] Houppis-Lamont, “*Digital control systems*“, Mc Grow Hill Publisher, Boston, M.A., USA, 2003
- [17] M. Niculescu, “*Lateral Track Control Law for Aerosonde UAV*”, AIAA 2001-0016, University of Washington, Seattle, January 2001
- [18] MATLAB@ help guide
- [19] L.Ljung: “*System Identification - Theory For the User*”, PTR Prentice Hall, Upper Saddle River, N.J., 1999
- [20] R.Pretolani, G.M.Saggiani, V.Rossi, B.Teodorani “*An “off the shelf” avionics system for Rotary Wing UAV rapid prototyping*”, Proceedings of the 32nd European Rotorcraft Forum, Maastrich (The Netherlands), Sept. 2006, paper ID UA03
- [21] B.Mettler, C.Dever, E.Feron, “*Scaling effects and dynamic Characteristics of Miniature Rotorcraft*”, AIAA, Journal of Guidance, Control, and Dynamics, Vol.27, No.3, 2004
- [22] S.Skogestad and I.Postlethwaite, “*Multivariable Feedback Control analysis and design*”, John Wiley and Son, NY,1997
- [23] B. Mettler, M.B. Tischler, and T. Kanade, “*Attitude Control Optimization for a Small-Scale Unmanned Helicopter*“, AIAA Guidance, Navigation and Control Conference, 2000, pp. 40-59.
- [24] *Handling qualities requirements for military rotorcrafts* .Techincal Report ADS-33D-PRF, United States Army, St-Louis Missouri, 1996
- [25] F.Zanetti , B. Teodorani , G.M.Saggiani, R.Pretolani, “*Rotary Wing UAV System Identification for flight control design*” European Rotorcraft Forum Conference 2007, 11-09-2007 Kazan, Russia