

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN
Meccanica e scienze avanzate dell'ingegneria

Ciclo XIII

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

MODEL PREDICTIVE FLIGHT CONTROL SYSTEMS FOR ROTORCRAFT
UAS

Presentata da: Colautti Stefano

Coordinatore Dottorato

Prof Franco Persiani

Relatore

**Prof. Franco Persiani
Eng. Jan-Floris Boer**

Esame finale anno 2011

Table of contents

List of figures	3
List of tables.....	5
Abstract.....	6
1 Introduction.....	7
2 Helicopter models.....	11
2.1 Non-linear model	11
2.1.1 Main rotor	12
2.1.2 Hiller bar	12
2.1.3 Tail rotor	13
2.1.4 Control chain	13
2.1.5 Fuselage.....	15
2.1.6 Engine	15
2.1.7 Validation.....	15
2.2 Linearized models	16
2.2.1 Model structure.....	16
2.2.2 Model linearization procedure	17
2.2.3 Model validation	20
2.3 Simulation models	21
2.4 Prediction model.....	22
2.5 Model uncertainties	22
3 Model predictive control.....	25
3.1 MPC basics	27
3.2 Stability.....	29
3.3 Feasibility	29
4 Controller implementation.....	31
4.1 Open loop optimal problem solution	32
4.2 Reference tracking	35
4.3 Feasibility guarantee	37
4.3.1 Virtual reference	37
4.3.2 Infinite horizon	40

4.3.3	Soft constraints.....	43
4.4	Offset-free control.....	46
4.4.1	Integral action.....	46
4.4.2	State observer.....	50
5	Stability and Control Augmentation System.....	53
5.1	SCAS control structure.....	53
5.2	Simulation results.....	55
6	Trajectory Tracking System.....	59
6.1	System architecture.....	60
6.2	Simulation results.....	61
7	Path Following System.....	68
8	Conclusion and outlook.....	71
	Bibliography.....	73

List of figures

figure 1.1: UniBo RUAS	8
figure 2.1: Main rotor and Hiller bar hub.....	12
figure 2.2: Hiller bar	13
figure 2.3: PWM to pitch data.....	14
figure 2.4: fuselage aerodynamic data	15
figure 2.5: Typical power spectral density of the non linear model outputs	19
figure 2.6 Comparison between linear and nonlinear models	21
figure 2.7: Simulation model scheme	21
figure 2.8: rotor momentums acting on the fuselage	24
figure 3.1 MPC base structure	26
figure 3.2 MPC optimization.....	27
figure 3.3 terminal set and feasibility.....	30
figure 4.1: performances as function of prediction horizon.....	39
figure 4.2: virtual set-point as function of prediction horizon	40
figure 4.3: inner stabilizer loop	42
figure 4.4: infinite horizon performances.....	43
figure 4.5: comparison between soft and hard state constraints	45
figure 4.6: Integral action architecture scheme	47
figure 4.7: effects of the integral action	47

figure 4.8: effect of the anti wind-up system.....	49
figure 4.9: Monte Carlo simulation	50
figure 4.10: state observer control architecture	51
figure 4.11: Monte Carlo simulation	52
figure 5.1: longitudinal accelerations.....	56
figure 5.2: lateral accelerations	57
figure 5.3: Monte Carlo Simulation.....	58
figure 6.1: Trajectory tracking system architecture	61
figure 6.2: longitudinal reposition	62
figure 6.3: Lateral reposition maneuvers.....	63
figure 6.4: Constant vs. known reference, NED position.....	64
figure 6.5: Constant vs. known reference, body speeds	64
figure 6.6: constant vs. known reference, attitude angles	65
figure 6.7: constant vs. known reference, controls displacement.....	65
figure 6.8: Monte Carlo simulation	66
figure 6.9: Time to obtain the control action.....	67
figure 7.1: Path following principle	69
figure 7.2: path following vs. trajectory tracking	69
figure 7.3: speed time plot.....	70
figure 7.4: attitude time plot.....	70

List of tables

Table 2.1: model and real data comparison..... 16

Table 4.1: anti wind-up limit variables48

Table 6.1: SCAS constraints.....55

Abstract

Constraints are widely present in the flight control problems: actuators saturations or flight envelope limitations are only some examples of that. The ability of Model Predictive Control (MPC) of dealing with the constraints joined with the increased computational power of modern calculators makes this approach attractive also for fast dynamics systems such as agile air vehicles.

This PhD thesis presents the results, achieved at the Aerospace Engineering Department of the University of Bologna in collaboration with the Dutch National Aerospace Laboratories (NLR), concerning the development of a model predictive control system for small scale rotorcraft UAS.

Several different predictive architectures have been evaluated and tested by means of simulation, as a result of this analysis the most promising one has been used to implement three different control systems: a Stability and Control Augmentation System, a trajectory tracking and a path following system.

The systems have been compared with a corresponding baseline controller and showed several advantages in terms of performance, stability and robustness.

1 Introduction

Unmanned Aerial Systems (UAS) has been widely used in the military field in the past decade. Several mission profiles such as surveillance, reconnaissance and, more recently, attack have been committed to these systems obtaining numerous advantages in terms of human safety, cost reduction and work rate efficiency.

The same advantages could be exploited in the civil market where UAS can be employed into a even wider range of tasks such as:

- law enforcement
- traffic control
- weather monitoring
- aerial photography
-

The great potential of UAS for the civil market and the interest demonstrated by industry induced, in 2001, the European Community to sponsor the UAS development program CAPECON¹, to attempt to kick-start a civil UAS industry in Europe and try to fill the gap with the United States. Its main goal was to provide European industry with detailed design and manufacture know-how on safe cost effective and commercially viable civil UAS.

¹ Civil uav APplications & Economic effectivity of potential CONfiguration solutions

After a survey on industrial needs and the development of formal requirements five fixed wing and two rotary wing architectures were defined in order to cover all the possible mission requirements.

As part of this effort the University of Bologna focused its research on the conventional helicopter configuration and developed the UniBo Rotorcraft Unmanned Aerial System (RUAS). The aim was building a technological demonstrator for the national industries potentially interested in the unmanned systems and to be used inside the university as platform for research in innovative navigation and control laws or for Human Machine Interface studies.

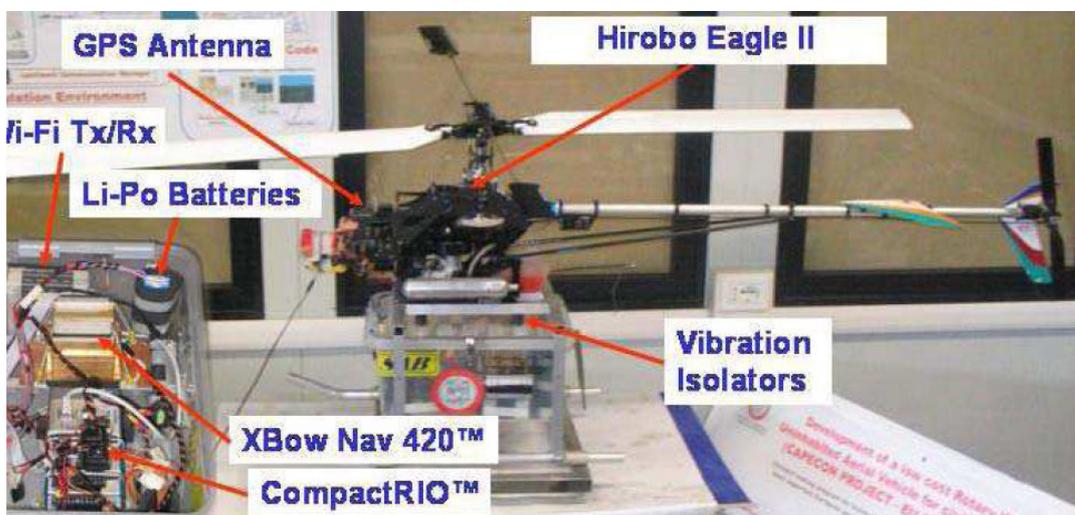


figure 1.1: UniBo RUAS

The system, shown in figure 1.1, 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 a longer tail boom. The new main rotor is a 2 blades see-saw type rotor with a diameter of 1.84 m. The rotorcraft is equipped 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. The helicopter total mass is about 11.2 kg and the engine has a maximum output power of 3hp.

The avionics box has been accommodated in the undercarriage and contains:

- a National Instruments CompactRIO system which performs both the task of Autopilot and Flight Management System (FMS).

- a crossbow NAV420 GPS-aided Attitude and Heading Reference System (AHRS) to provide the position, speed and attitude of the rotorcraft
- an ultrasonic sensor used to measure the altitude above the ground of the vehicle
- a data link system to transmit and receive information from the ground station.

The system architecture has been validated by means of hardware in the loop testing, flight testing and the development of a baseline nested loop PID automatic controller. For more details on the evolution of the project refer to [1], [2], [3].

In the frame of this project this thesis has the aim of presenting the results achieved by the author in the field of the Flight Control Systems (FCS) for small scale rotorcraft UAS. In particular the development of an innovative FCS based on the Model Predictive Control (MPC) theory is presented.

The development of a control algorithm for helicopters is a rather challenging task due to the instability of the system dynamics and the strong cross-coupling between all the axis. Moreover, as for the fixed wing aircrafts, several constraints have to be taken into account designing the control laws; actuator saturations, maximum attitude angles, limit load factors and speed constraints are only few examples of it.

In the last decade several different control approaches have been used to tackle the helicopter flight control problem, from simple PID nested loop controllers [4] to more complex and elaborate architectures based on optimal [5] robust [6] and non linear techniques [7]. As normally done in the majority of industrial applications the respect of constraints is addressed only during the calibration phase together with the other performance requirements introducing three main disadvantages:

- the result of the calibration is always a compromise between different requirements in particular when aggressive manoeuvring is required,
- the calibration process is complicated and time demanding,

- there is no guarantee that the constraints will be respected and that stability maintained.

Conversely model predictive control offers the possibility of explicitly introduce the presence of constraints into the controller formulation and guarantees that them will be respected for all the possible states configuration. This has the advantage of simplifying the calibration of the system since the respect of constraints has been already taken care of. Moreover stability of the constrained closed loop system can be mathematically demonstrated and, as will be shown in the thesis, this characteristic is maintained also in presence of significative uncertainties.

The thesis is organized as follows, in section 2 the models of the UniBO RUAS used in this thesis will be analyzed, in particular a fully non-linear model of the helicopter built into the FlightLab environment will be presented together with the continuous and discrete time models used for the control synthesis. Also the choice of model uncertainties use for robustness assessment will be shown in this section.

Section 3 will give a short introduction to the model predictive theory explaining the work principle of this class of controllers together with the stability and feasibility problems.

Section 4 shows different model predictive architectures giving the mathematical details of each formulation and analysing, with the help of simulations, the advantages and disadvantages of each of them. As result of this analysis the best control formulation has been chosen and adopted to build three different flight control systems presented in the following sections.

In section 5 a Stability and Control Augmentation System (SCAS) whose aim is to control the body frame speeds and the heading of the helicopter is presented. A comparison with a baseline LQR controller is shown by means of simulation, moreover the influence of model uncertainties is shown using a set of Monte Carlo simulations.

The same has been done for a trajectory tracking and a path following system respectively presented in sections 6 and 7. Finally, section 8 will draw the conclusions and offer further research recommendations.

2 Helicopter models

For the purposes of this work three different kind of models of the UniBo RUAS have been developed. Section 2.1 describes the assumptions made to build a fully non-linear model of the helicopter in the FlightLab environment; starting from this a series of linear models used for both simulation and control purposes have been derived by means of an identification procedure described in section 2.2. Finally, uncertainties have been added to the principal physical parameters of the model in order to evaluate the robustness characteristics of control system, this uncertain model is described in section 2.5.

2.1 Non-linear model

In order to better validate the control architectures developed in this thesis a non-linear model of the helicopter has been built into the FlightLab environment (a tool specifically intended to build rotorcraft models and provide analysis and simulation instruments).

The rotorcraft has been conceptually divided into its main physical components, namely:

- Main rotor
- Bell-Hiller bar
- Tail Rotor
- Control Chain (actuators, rods, etc.)

- Fuselage
- Engine

In the following each component model is described into the details justifying the choices made during the modelling phase

2.1.1 Main rotor

This component has been modelled as a 2-blade teetering rotor with a constant speed of 1100 rpm. The blades have been considered rigid with a uniform density distribution. The blade section profile has been kept constant along the entire span and has been approximated by the TsAgi 14% shape; the aerodynamic data has been estimated by using the CFD software package JavaFoil. The Peters-He three state dynamic inflow model has been adopted to calculate the inflow field and the aerodynamic interference between the main rotor wake and the fuselage has been considered.

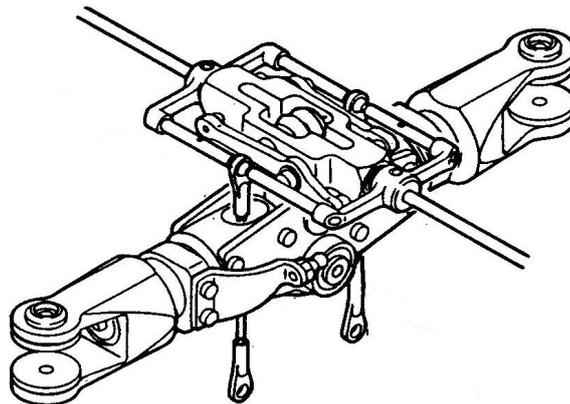


figure 2.1: Main rotor and Hiller bar hub

2.1.2 Hiller bar

The stabilizer bar (figure 2.2) has the function to provide a feedback action that improves the flying qualities of the rotorcraft, moreover the system provides a portion of the torque needed to control the main rotor blades pitch reducing the power needed by the actuators. This components models only the rotor dynamics while the connection between the Bell-Hiller stabilizer, the control servos and the main rotor will be described in the next sections. The stabilizer is located directly above the main rotor and has been modelled as a second teetering rotor. As the main rotor it has been considered rigid with the same constant

speed of 1100 rpm. The inertia distribution has been calculated summing the contributions of the plastic paddles, the supporting rod and the calibration brass masses.

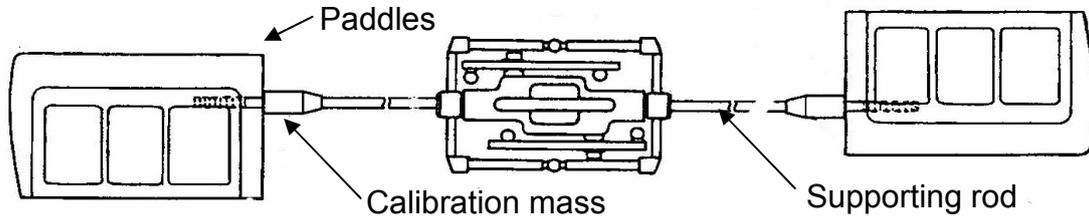


figure 2.2: Hiller bar

The paddles are built using a NACA0015 profile and no aerodynamic interference with the remaining components has been considered since the stabilizer paddles are considerably smaller than the main rotor blades.

2.1.3 Tail rotor

The tail rotor has been modelled as a simple Bailey² rotor since only the collective pitch control is applied. The blades have been considered rigid and uniform, a NACA 0010 airfoil has been used to model the aerodynamic characteristics.

2.1.4 Control chain

In the real helicopter the servo actuators are connected, via a complex mechanical gear system, to a couple of swashplates. One is located on the tail rotor and controls the corresponding blades, the second controls the pitch of the stabilizer paddles and, via a mechanical mixer, the pitch of the main rotor blades. This mixer sums the contribute given from the swashplate to another which is proportional to the stabilizer flapping angle. The servos are controlled using a Pulse Width Modulated (PWM) electric signal where the displacement of the actuator is proportional to the duty cycle of the input. This complex system can be modelled by the following relationships:

$$\theta_1(\psi) = K_{1c}X_c + (K_{1lat}X_a + K_{bh}\beta_{2s})\cos\psi + (K_{1long}X_b + K_{bh}\beta_{2c})\sin\psi \quad 2.1$$

² A Baley rotor is a rotor in which only the coning angle is considered while the longitudinal and lateral flapping are not considered; this is the typical configuration of tail rotors.

$$\theta_2(\psi) = K_{2lat} X_a \cos \psi + K_{2long} X_b \sin \psi \quad 2.2$$

$$\theta_3(\psi) = K_{ped} X_p \quad 2.3$$

Where $\theta_1, \theta_2, \theta_3$ are the blade pitch of the main rotor, the stabilizer and the tail rotor respectively, X_a, X_b, X_c, X_p are the servo control inputs and β_{2s}, β_{2c} are the lateral and longitudinal flapping angles of the Bell-Hiller rotor. The following observations can be done:

- the main rotor cyclic pitches are a linear function of the cyclic controls and the Bell-Hiller stabilizer flapping angles;
- the stabilizer paddles do not have any collective pitch;
- the tail rotor cyclic pitches are forced to be 0 according to the Bailey rotor model used

All the gains K present in the equations have been obtained measuring the pitch of each blade varying manually the control inputs on fixed azimuth positions (0 and 90 degrees), figure 2.3 shows the result of the experiment.

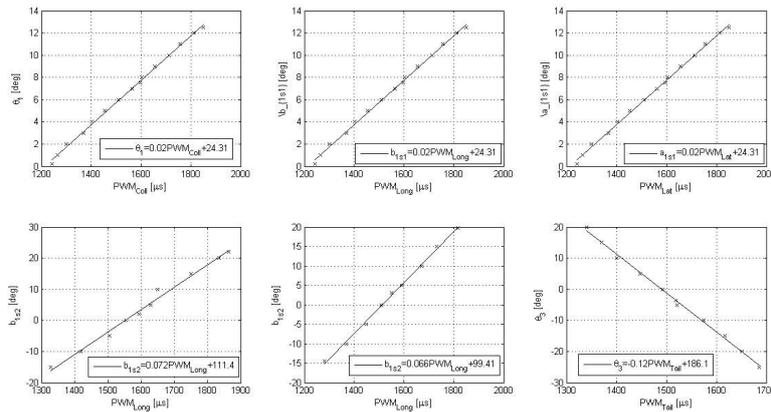


figure 2.3: PWM to pitch data

The equations 2.1, 2.2, 2.3 have been implemented using the Control System Graphic Editor (CSGE, a module of FlightLab) and connected to each single rotor.

2.1.5 Fuselage

This component models all the inertial and gravitational actions due to the entire UAS and the aerodynamic effects introduced by the fuselage body. In this model the fuselage has been considered as a rigid body; the mass and inertia variations due to the fuel consumption have been neglected. The centre of gravity position and the inertia tensor have been measured directly on the model through dedicated experiments.

Several experimental and CFD methods have been considered in order to calculate the aerodynamic coefficients of the fuselage. All the approaches revealed to be too complex and time demanding in relation with the accuracy needed in this application. In the model, data from full-scale helicopters given in [1] has been used (see figure 2.4)

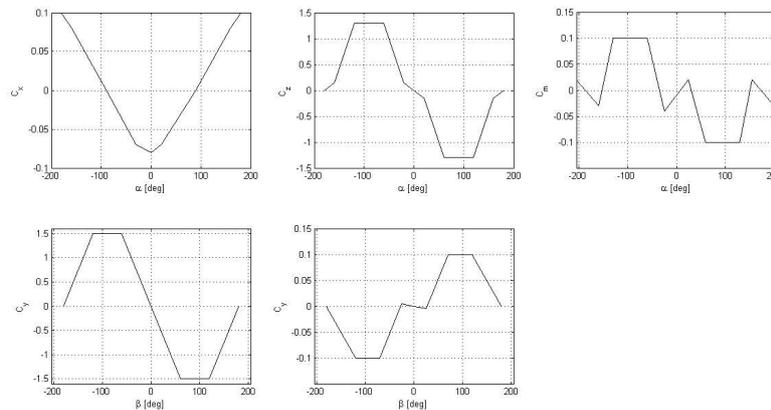


figure 2.4: fuselage aerodynamic data

2.1.6 Engine

Flight test data show that the rotor speed variation during typical manoeuvres is about 50 rpm that has a negligible effect on the dynamic behaviour. For this reason in this model no engine dynamics have been added, the rotor speed has been kept constant at a nominal value of 1100 rpm.

2.1.7 Validation

In order to have some indication about the quality of the model some comparison with the real data can be made. Table 2.1 resumes some comparisons made with real flight data

	Model	Flight Data
Collective pitch [deg]	7	6.5-7
Roll attitude [deg]	2.3	~3
Power required [hp]	0.97	

Table 2.1: model and real data comparison

Moreover the lock number of the main rotor and the Hiller bar have been calculated from the model linearization and resulted in $\gamma_f \cong 3.44$ and $\gamma_s \cong 0.59$ respectively, which corresponds to the literature data available for the same class of machines [9], [10].

A detailed validation with flight data has not been possible since, after a major accident, the rotorcraft has been repaired and modified. No flight data has been recorded yet in the new configuration.

2.2 Linearized models

In the formulation of the flight control systems used in this work a linear model of the rotorcraft dynamics is necessary, for this reason the FlightLab model described in the previous section has been linearized around several trim conditions. In the following the structure of the model and the linearization procedure used to obtain it will be described. Finally, the validity of such models will be demonstrated by comparing the response of the linear and the non-linear models subject to the same inputs.

2.2.1 Model structure

For the purpose of this work the classic state space linear model has been used

$$\dot{x} = Ax + Bu \tag{2.4}$$

where x and u are respectively the state and input vectors of the model and A and B the stability and control matrices. In order to obtain a good linear approximation of the plant is important not only to get the correct values for the stability and the control matrices but also to include all the significative states in the relative vector.

In full-scale helicopters the rotor flapping dynamics is significantly faster than the body motion and the related states can be neglected using a quasi-static flapping approximation. In small-scale model helicopters instead, the main rotor dynamics (slowed by the introduction of the stabilizer bar) and the body dynamics (very fast due to the small scale) have comparable time constants and the rotor flapping dynamics have to be included in the linear model.

The state and input vectors become then respectively:

$$x = [a, b, c, d, p, q, r, u, v, w, \varphi, \vartheta, \psi]^T$$

$$u = [x_a, x_b, x_c, x_p]^T$$
2.5

where:

- a, b, c, d are the longitudinal and lateral flapping angles of respectively the main and stabilizer rotors;
- p, q, r , the body frame angular rates;
- u, v, w the body frame linear speeds;
- φ, ϑ, ψ the helicopter Euler angles;
- x_a, x_b, x_c, x_p the lateral and longitudinal cyclic pitch, the collective pitch and the tail rotor pitch respectively.

2.2.2 Model linearization procedure

The standard linearization tool provided by FlightLab gives to the user a list of states that can be selected and included in the state space and input vectors; the states available on this list depend on the particular choices made by the user when building the model. The linearization algorithm runs the non-linear model twice for each state included in the linearized vector; in each simulation a small disturbance is added or subtracted from one of the states while the others are kept fixed to the trim value; moreover all the states not included in the vector are left free to evolve. The stability derivatives are obtained by calculating the time derivatives of the states in these conditions once the free states have reached a steady state condition.

Due to the structure of the model, the rotor flapping states are not available in the standard tool list; on the other hand, the time scale of the fuselage and the rotor dynamics are not

so much separated as happens for full scales rotorcrafts and the steady state assumption for the rotor dynamics can not be done. For this reason a different approach to obtain the linear models had to be taken. Since the model predictive control is fundamentally a time domain technique an identification procedure on the same domain has been adopted. The basic idea is to calculate the stability and control matrices by minimizing the difference between the response of the linear and the non-linear model subject to the same inputs. For each trim condition a set of simulations with the nonlinear model have been carried out recording the following data:

- 1) the system inputs u ;
- 2) all the model states x ;
- 3) the forces and moments introduced by each component

The data sets have been obtained forcing each control and state (except the flapping angles) to follow a 3-2-1-1 or sine-sweep profiles while the remaining states were frozen in the trim condition. In this way it has been possible to isolate the contribution of each control and state in the stability derivatives independently.

Analysing the frequency content of the obtained data makes it possible to identify two main contributions (see figure 2.5):

- 1) a low-frequency contribution due to the fuselage and flapping dynamics (from 0 to ~ 5 Hz);
- 2) a sequence of higher harmonics corresponding to multiples of the main rotor speed (36 and 73 Hz).

Since the latter contributions are not interesting for this application the data has been cut off above 10Hz.

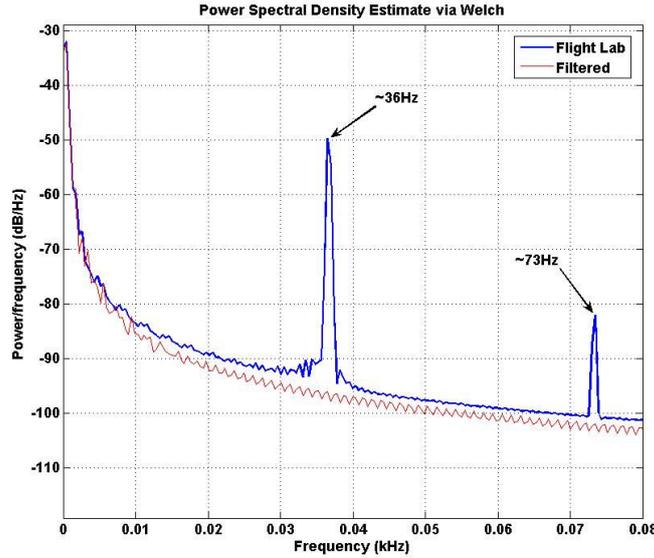


figure 2.5: Typical power spectral density of the non linear model outputs

Given the classic linear model formulation

$$\dot{x} = Ax + Bu \quad 2.6$$

The identification process has the objective to find the correct numerical values on the matrices A and B in order to keep the linear and the non-linear models response as similar as possible. In order to minimize the number of variables that have to be identified at the same time each row of the model 2.6 has been calculated separately. The identification algorithm calculates the terms in the i^{th} row of the matrices by minimizing the following cost function:

$$J = \int_0^T (\lambda_{nl} - \lambda_l)^2 dt \quad 2.7$$

Where λ is a variable associated to the row considered and the subscripts nl and l are referred to the non-linear and linear response respectively. The non-linear response is calculated directly using the FlightLab model while the linear approximation is calculated by:

$$\lambda_l = Mx_{nl} + Nu_{nl} \quad 2.8$$

The rotor dynamics stability and control coefficients have been calculated using as identification variable λ the time derivatives of the flapping angles ($\dot{a}, \dot{b}, \dot{c}, \dot{d}$).

The rigid body dynamics has been identified indirectly, the forces and moments of each model component have been used as identification variables. Once all the force and moments have been identified, the linear expression obtained have been substituted into the 6 d.o.f. rigid body equations 2.9 allowing to obtain the complete linear model by analytically linearizing the remaining parts. This choice has been made because facilitated the choice of the initial conditions for the identification algorithm and reduced again the number of variables to be identified.

$$\begin{aligned}
 \dot{u} &= vr - wq + \frac{F_x}{m} - g \sin \theta \\
 \dot{v} &= wp - ur + \frac{F_y}{m} + g \cos \theta \sin \phi \\
 \dot{w} &= uq - vp + \frac{F_z}{m} + g \cos \theta \cos \phi \\
 I_{xx} \dot{p} &= (I_{yy} - I_{zz})qr + I_{xz}(\dot{r} + pq) + M_x \\
 I_{yy} \dot{q} &= (I_{zz} - I_{xx})rp + I_{xz}(r^2 - p^2) + M_y \\
 I_{zz} \dot{r} &= (I_{xx} - I_{yy})pq + I_{xz}(\dot{p} - qr) + M_z
 \end{aligned} \tag{2.9}$$

2.2.3 Model validation

In order to validate the results obtained both the linear and the non-linear model have been excited by a common input sequence (different from the ones used in the identification process) and the responses obtained have been compared.

figure 2.6 shows an example of the comparisons, in this case a 3-2-1-1 input has been imposed in the collective control. As can be seen there is a good correspondence between the linear and the nonlinear model, especially to the direct contributions, the cross-coupling effects are captured with lower accuracy (in this example the lateral dynamics). The typical frequencies of both the flapping and the body dynamics are well captured giving an adequate correspondence between the linear and the fully nonlinear models, moreover those frequencies are in the same range of what can be expected of this class of vehicles and reported in the known literature [10]

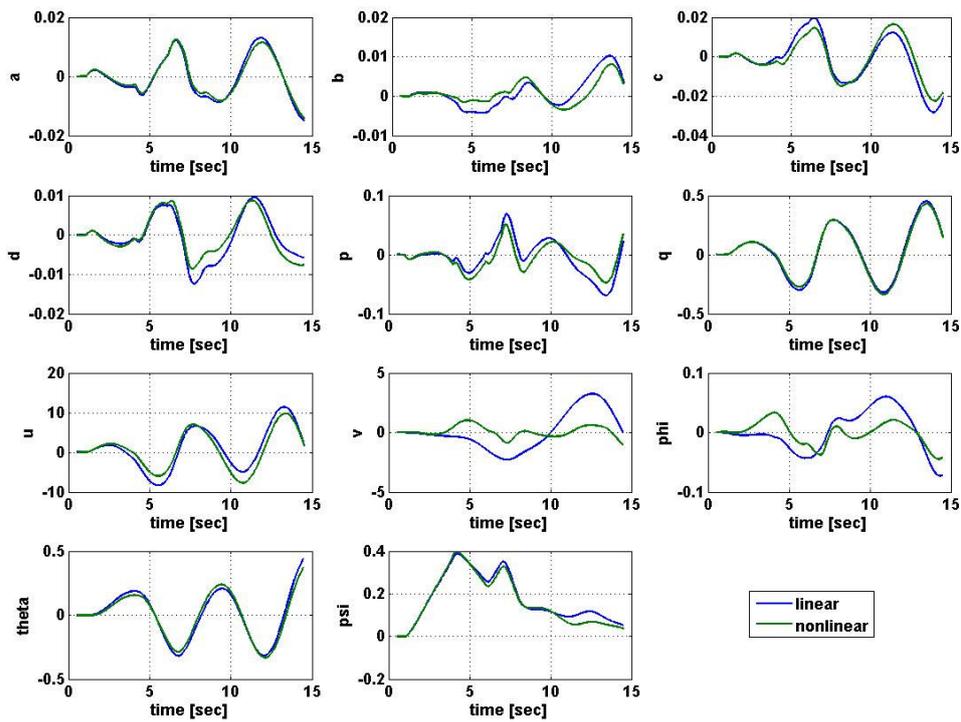


figure 2.6 Comparison between linear and nonlinear models

2.3 Simulation models

The direct implementation of the control systems in the FlightLab environment is too complicated and time demanding to be done in the first phase of the control architecture evaluation. For this reason the simulations needed to evaluate the characteristics of the control architectures have been carried in Simulink and a simplified model has been implemented as depicted in figure 2.7

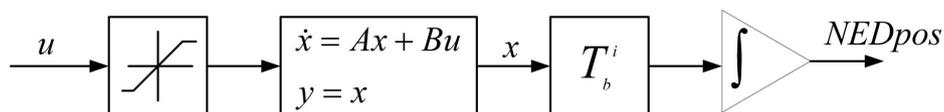


figure 2.7: Simulation model scheme

The body frame dynamics has been represented by the linear state space model developed in this section where all the states are supposed to be measured and available. The position of the helicopter has been represented in the North-East-Down (NED)

reference frame and has been calculated rotating the body speeds with the full non-linear direction cosine matrix T_b^i and then integrating the rotated speeds.

$$T_b^i = \begin{bmatrix} \cos \vartheta \cos \psi & \sin \varphi \sin \vartheta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \vartheta \cos \psi + \sin \varphi \sin \psi \\ \cos \vartheta \sin \psi & \sin \varphi \sin \vartheta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \vartheta \sin \psi - \sin \varphi \cos \psi \\ -\sin \vartheta & \cos \vartheta \sin \varphi & \cos \vartheta \cos \varphi \end{bmatrix} \quad 2.10$$

2.4 Prediction model

Model predictive control makes use of a plant of the model to calculate the control action. As will be explained later, in this work the prediction model is represented by a discrete time linear model, which has been obtained from the continuous time model as follows. Given a constant sampling time T_s the time derivative can be approximated by:

$$\dot{x} = \frac{x(k+1) - x(k)}{T_s} \quad 2.11$$

by substituting it into the continuous time formulation 2.6 we obtain:

$$x(k+1) = (T_s A + 1)x(k) + T_s B u(k) \quad 2.12$$

In order to maintain a simple notation the same symbols have been used for both discrete and the continuous time characteristic matrices, no confusion should be raised by this choice since the difference will be clear by the context.

$$x(k+1) = Ax(k) + Bu(k) \quad 2.13$$

2.5 Model uncertainties

In order to evaluate the sensitivity of the control systems to the possible mismatches between the prediction model and the controlled plant a series of uncertainties have been artificially added to the simulation model using the robust control toolbox of Matlab.

The set of parameters to be modelled as uncertain has been chosen analysing the basic principles of the helicopter flight dynamics. As well known in these kinds of vehicles the main source of control forces and momentums is the main rotor which can be represented by a simple disk flapping around the hub. The flapping dynamics can be represented by the simplified equations:

$$\begin{aligned}
 \dot{b} &= -p - \frac{b}{\tau_{mr}} - \frac{1}{\tau_{mr}} \frac{\partial b}{\partial \mu_v} \frac{v}{\Omega R} + \frac{B_{xa}}{\tau_{mr}} (x_a + B_d d) \\
 \dot{a} &= -q - \frac{a}{\tau_{mr}} + \frac{1}{\tau_{mr}} \left(\frac{\partial a}{\partial \mu} \frac{u}{\Omega R} + \frac{\partial a}{\partial \mu_w} \frac{w}{\Omega R} \right) + \frac{A_{xb}}{\tau_{mr}} (x_b + A_c c) \\
 \dot{c} &= -p - \frac{c}{\tau_{sr}} - \frac{1}{\tau_{sr}} \frac{\partial c}{\partial \mu_v} \frac{v}{\Omega R} + \frac{C_{xa}}{\tau_{sr}} x_a \\
 \dot{d} &= -q - \frac{d}{\tau_{sr}} + \frac{1}{\tau_{sr}} \left(\frac{\partial d}{\partial \mu} \frac{u}{\Omega R} + \frac{\partial d}{\partial \mu_w} \frac{w}{\Omega R} \right) + \frac{D_{xb}}{\tau_{sr}} x_b
 \end{aligned} \tag{2.14}$$

and the following parameters have been chosen as uncertain:

- the main and stabilizer rotor time constants: τ_{mr}, τ_{sr} ,
- the control gains: $A_{xb}, B_{xa}, C_{xa}, D_{xb}$
- the Bell-Hiller mixer gains A_c, B_d

With reference to figure 2.8 the roll and pitch dynamics can be expressed as:

$$\begin{aligned}
 I_{xx} \dot{p} &= (K_\beta + Th_{mr}) b \\
 I_{yy} \dot{q} &= (K_\beta + Th_{mr}) a
 \end{aligned} \tag{2.15}$$

and only the inertia parameters have been considered uncertain.

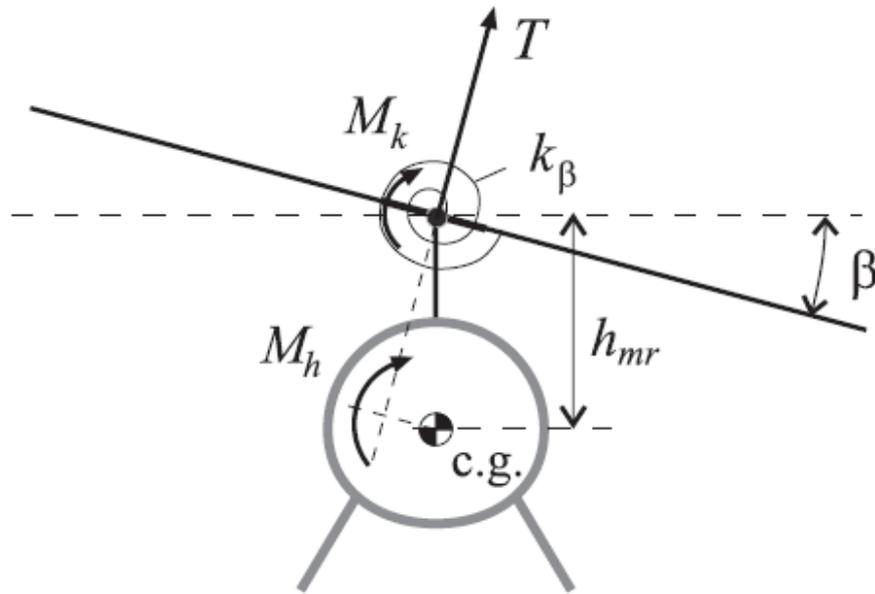


figure 2.8: rotor momentums acting on the fuselage

For what concerns the linear motion the longitudinal and lateral dynamics can be expressed as:

$$\begin{aligned} m\dot{u} &= -g\vartheta + Ta \\ m\dot{v} &= g\varphi + Tb \end{aligned} \tag{2.16}$$

again only the mass parameter is sufficient to represent uncertainty.

With the same considerations, also the following parameters have been considered uncertain:

- z-axis principal momentum of inertia I_{zz}
- the collective and tail control gains.

A detailed analysis on the uncertainty levels goes beyond the scope of this thesis; in order to have an indication of the robustness of the control system an error up to 25% on each uncertain parameter has been used.

3 Model predictive control

Mayne et al. in [11] define Model Predictive Control (MPC) or Receding Horizon Control (RHC) as is called sometimes as *a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.*

In other words, the predictive approach obtains the input to be applied to the process by minimizing the difference between the future reference and the predicted outputs of the system, the latter are calculated by means of a proper model of the plant. MPC differs from other conventional approaches since it solves the optimal control problem *on-line* for the current state of the plant, rather than determining *offline* a feedback policy (that provides the optimal control for all states).

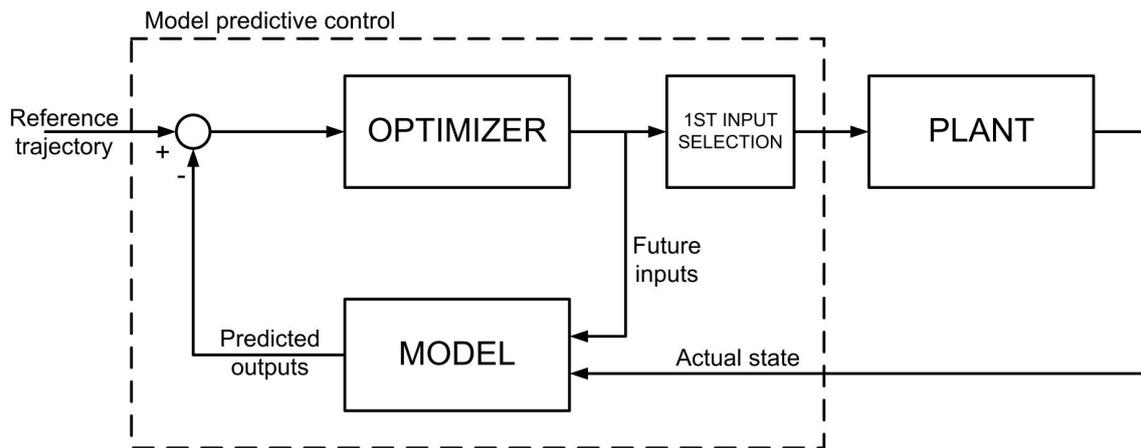


figure 3.1 MPC base structure

As will be explained later the MPC formulation is very general and includes a wide class of algorithms, each of them has its own particular characteristics but shares the same base structure (see figure 3.1) and the principal components, namely:

- the prediction model;
- the objective function;
- the optimization solver;
- the constraints expression.

The various MPC approaches owe their success, especially in the process industry, due to the ability of:

- handling multivariable processes naturally;
- taking into account for input and state constraints explicitly;
- handling non minimum phase and unstable plants;
- being easy to tune.

On the other hand, the use of MPC has been limited:

- due to the time domain nature of the approach which leads to partially lose the frequency domain information;

- to systems with relatively slow dynamics due to the heavy computational burden associated with the solution of the constrained optimal problem.

The first applications of MPC can be found in the petro-chemical and process industry where economical considerations required operating points on the boundary of the admissible sets defined by the constraints. Examples of such early predictive algorithms are given by IDCOM (identification and command) proposed by Richalet et al in 1978 which employed a finite horizon pulse response linear model, a quadratic cost function, and input and output constraints. Or by program quadratic dynamic matrix control (QDMC; Garcia& Morshedi, 1986) where quadratic programming is employed to solve exactly the constrained open-loop optimal control problem that results when the system is linear, the cost quadratic, and the control and state constraints are defined by linear inequalities.

Similarly to other technical inventions MPC was implemented by industries long before the development of a well established theory, nevertheless a great research effort has been made by the academic community since the mid eighties and nowadays a strong conceptual and practical framework for both practitioners and theoreticians has been built. The basic concepts of both linear and non-linear MPC can be found respectively in [11] and [14].

In the following, the mathematical foundations of the predictive approach will be given together with the more important theoretical results regarding stability and feasibility problems.

3.1 MPC basics

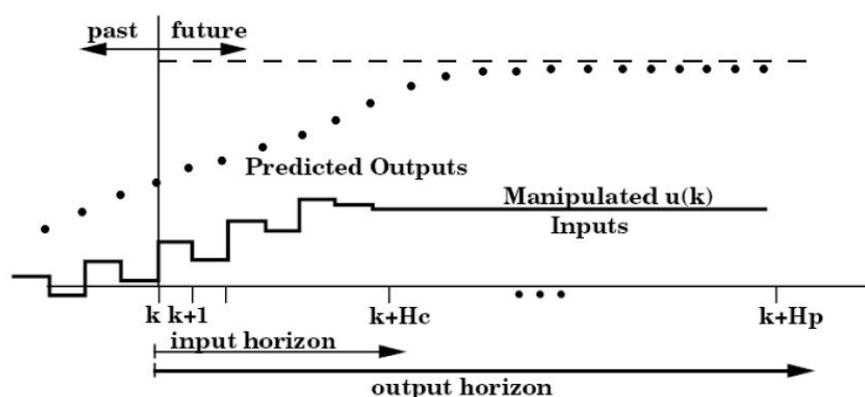


figure 3.2 MPC optimization

In order to obtain a rigorous formulation of the model predictive approach we assume that the plant can be modelled, as usually happens in the MPC literature, by the difference equation:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k), u(k)) \end{aligned} \tag{3.1}$$

where x is the state vector with n elements, u the input vector of dimension m , y the z -dimensional output vector, f and h generic functions used to model the process.

The control action is obtained by solving, at each sampling instant k , an open loop optimal problem defined by a cost index in the form 3.2 and the set of constraints 3.3. The result of the optimization is a sequence of future controls $\mathbf{u} = [u(k+1), \dots, u(k+H_c)]$ where only the first input $u(k+1)$ is actually applied to the plant.

$$J(x, \mathbf{u}) = \sum_{i=k+1}^{k+H_p-1} L(x(i), u(i)) + F(x(k+H_p)) \tag{3.2}$$

$$\begin{cases} u \in U \\ x \in X \\ x(k+H_p) \in X_f \end{cases} \tag{3.3}$$

The cost index J is defined over a future time window, usually called *prediction horizon*, which starts at the actual sampling instant k and ends H_p steps later. Each step is long one sampling interval. The cost is given by the sum of two different contributes: the stage cost L which is function of the predicted states and inputs and the terminal cost F which is function only of the state at the end of the prediction horizon.

The constraints are formulated by means of the sets U , X and X_f . The first is a subset of R^m and gives the input constraints; the second is a subset of R^n and limits the admissible states while the terminal set $X_f \subset R^n$ involves only the last state of the prediction and is used for stability reasons as will be explained later.

As depicted in figure 3.2 a control horizon is often introduced, in this case only the first H_c control moves are allowed while the remaining $H_p - H_c$ are obliged to assume a fixed

value; this is done in order to reduce the degrees of freedom of the optimization reducing its complexity; in the following when is not specified $H_c = H_p$.

3.2 Stability

The most important theoretical result in the MPC theory is given by the possibility of guaranteeing the stability of the closed loop system, by properly choosing the basic components of the optimization problem.

In order to demonstrate this property a local control law $u = k_f(x)$ has to be introduced, this control action is applied only when the state x is inside the terminal constraint set X_f . On this premise is possible to demonstrate that the constrained closed loop system is stable if:

- S1. $X_f \subset X, X_f$ is closed and $0 \in X_f$ (the state constraints are satisfied in the terminal set)
- S2. $k_f(x) \in U, \forall x \in X_f$ (the input constraints are respected by the local control law)
- S3. $f(x, k_f(x)) \in X_f \forall x \in X_f$ (X_f is positively invariant subject to k_f)
- S4. $F(f(x, k_f(x))) - F(x) + l(x, k_f(x)) \leq 0, \forall x \in X_f$

The detailed demonstration can be found in [1].

3.3 Feasibility

The predictive control approach is based on the ability of finding a solution of the open loop optimization problem given by equations 3.2 and 3.3 for each possible condition of the plant. Even before trying to find the solution is worth to understand if such a solution exists or not, in other words we need to ask: is the open loop problem feasible?

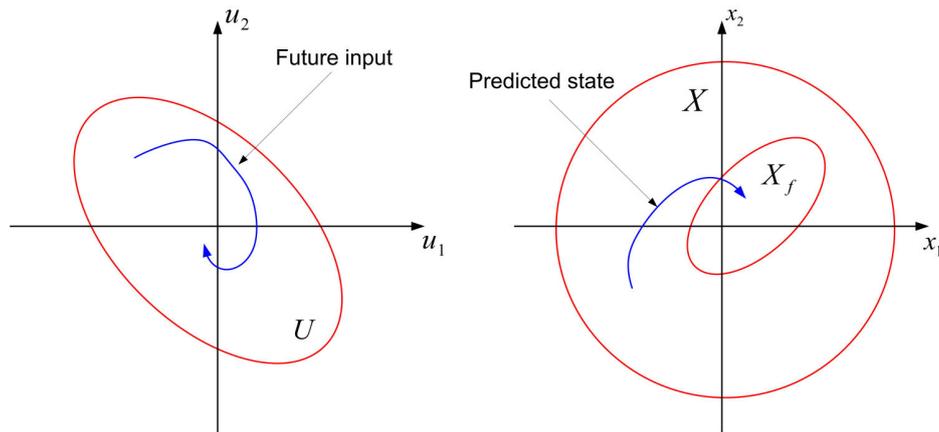


figure 3.3 terminal set and feasibility

Analyzing the constraints structure, see figure 3.3 which shows a bi-dimensional example, is possible to understand how the presence of the terminal constraint can be a source of unfeasibility. In fact, given an initial condition $x(k)$ the terminal constraint requires to find a future control sequence in U , and a corresponding state trajectory in X , which drives the last state of the prediction inside the terminal set X_f in less than one prediction horizon. Such a condition is quite restrictive and, in our knowledge, there are no general results giving conditions for the feasibility of the finite horizon problem.

The solution of the feasibility problem is normally addressed at the implementation level, in other words when the choices of the prediction model, the cost function and the constraints expressions are made.

The solutions investigated for the flight control system will be discussed in the next section together with the other implementation issues.

4 Controller implementation

In order to successfully apply MPC to practical control problems the base concept described in section 3 has to be specialized to meet the specific needs of the particular application under exam.

In this section the implementation issues addressed during the development of the flight control system will be described and the solutions obtained analyzed into the details, in particular:

1. a fast optimization algorithm based on the solution of a quadratic programming problem has been implemented;
2. the ability of following a given reference rather than regulate the plant towards the origin has been obtained by a proper coordinates transformation;
3. the feasibility problem has been addressed using two different techniques: a virtual set-point and a infinite prediction horizon;
4. zero steady state error has been obtained by using integral actions and state observer techniques.

4.1 Open loop optimal problem solution

The MPC theory does not provide a way to calculate the solution of the optimization problem and, therefore, a suitable optimization algorithm has to be found. In the literature different classes of solvers such as interior point, fast gradient or explicit methods have been proposed each one with different characteristics and computational burdens. For the purposes of this work an interior point method especially developed for real time applications have been used. This optimizer is able to solve quadratic programming optimization problems in the form of eq. 4.1

$$\begin{aligned} J &= v^T H v + v^T G \\ s.t & \\ v_{\min} &\leq M v \leq v_{\max} \end{aligned} \tag{4.1}$$

Where v is the vector of optimization variables and for MPC problems corresponds to the future control sequence.

In order to transform the optimization problem given by equations 3.2, 3.3 into a QP problem the stage and terminal cost functions have to be chosen in the quadratic form 4.2 and the model has to be in the linear time invariant form 4.3.

$$J = \sum_{i=k+1}^{k+H_p-1} (x(i)^T Q x(i) + u(i)^T R u(i)) + x(k+H_p)^T F x(k+H_p) \tag{4.2}$$

$$x(k+1) = Ax(k) + Bu(k) \tag{4.3}$$

the future state sequence $[x(k+1), \dots, x(k+H_p)]$ can be obtained by recursively applying equation 4.3 to the previous state of the sequence as follows:

$$\begin{aligned}
 x(k+1) &= Ax(k) + Bu(k) \\
 x(k+2) &= Ax(k+1) + Bu(k+1) = A^2x(k) + ABu(k) + Bu(k+1) \\
 x(k+3) &= Ax(k+2) + Bu(k+2) = A^3x(k) + A^2Bu(k) + ABu(k+1) + Bu(k+2) \\
 &\dots \\
 x(k+s) &= A^s x(k) + A^{s-1}Bu(k+1) + A^{s-2}Bu(k+2) \dots + Bu(k+s) \\
 &\dots
 \end{aligned} \tag{4.4}$$

or, in a more compact notation

$$\mathbf{x} = S_{x_0}x_0 + S_u \mathbf{u} \tag{4.5}$$

where the bold character denotes a sequence of vectors, e.g. $\mathbf{x}=[x(k+1), \dots, x(k+H_p)]$, and $x_0 = [x(k), u(k)]$.

On the other hand, the cost function can be written in the matrix form

$$\begin{aligned}
 J = & \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+H_p-1) \\ x(k+H_p) \end{bmatrix}^T \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & 0 \end{bmatrix} + \dots \\
 & \dots + \begin{bmatrix} u(k+1) \\ \vdots \\ u(k+H_c) \\ u(k+H_c) \\ \vdots \\ u(k+H_p) \end{bmatrix}^T \begin{bmatrix} R & & & \\ & \ddots & & \\ & & R & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+H_p-1) \\ x(k+H_p) \end{bmatrix} + \dots \\
 & \dots + \begin{bmatrix} u(k+1) \\ \vdots \\ u(k+H_c) \\ u(k+H_c) \\ \vdots \\ u(k+H_p) \end{bmatrix}^T \begin{bmatrix} R & & & \\ & \ddots & & \\ & & R & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} u(k+1) \\ \vdots \\ u(k+H_c) \\ u(k+H_c) \\ \vdots \\ u(k+H_p) \end{bmatrix} + \dots
 \end{aligned} \tag{4.6}$$

In short

$$J = \mathbf{x}^T \bar{Q} \mathbf{x} + \mathbf{u}^T \bar{R} \mathbf{u} \tag{4.7}$$

By substituting 4.5 in 4.7 we can obtain the following:

$$J = \mathbf{u}^T (S_u^T \bar{Q} S_u + \bar{R}) \mathbf{u} + \mathbf{u}^T (S_u^T \bar{Q} S_{x_0} x_0) + const \quad 4.8$$

For what concerns the constraints usually are expressed in the polyhedral form 4.9 over the entire prediction horizon.

$$\begin{aligned} x_{\min} &\leq x \leq x_{\max} \\ u_{\min} &\leq u \leq u_{\max} \end{aligned} \quad 4.9$$

In order to transform this expression in the desired form is necessary to explicitly write the constraints over the future states and input sequences:

$$\begin{aligned} \begin{bmatrix} x_{\min} \\ \vdots \\ x_{\min} \end{bmatrix} &\leq \mathbf{x} \leq \begin{bmatrix} x_{\max} \\ \vdots \\ x_{\max} \end{bmatrix} \\ \begin{bmatrix} u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix} &\leq \mathbf{u} \leq \begin{bmatrix} u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} \end{aligned} \quad 4.10$$

By substituting 4.5 in the future states expression, is possible to obtain the desired form.

$$\begin{aligned} \begin{bmatrix} x_{\min} \\ \vdots \\ x_{\min} \end{bmatrix} - S_{x_0} x_0 &\leq S_u \mathbf{u} \leq \begin{bmatrix} x_{\max} \\ \vdots \\ x_{\max} \end{bmatrix} - S_{x_0} x_0 \\ \begin{bmatrix} u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix} &\leq \mathbf{u} \leq \begin{bmatrix} u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} \end{aligned} \quad 4.11$$

Both the cost function expression in 4.8 and the constraints 4.11 are written in the desired QP form. It is important to notice that the linear term G and the constraints limits v_{\min} and v_{\max} depend on the initial condition x_0 which gives the closed loop behaviour to this control architecture.

Is also worth to point out that the conversion from the model predictive standard formulation to the quadratic programming one cannot be done completely offline, in fact both the cost function and the constraint formulation depend on the initial condition x_0 which changes at every sampling instant and, hence, the passage from 4.8 and 4.11 to 4.1 has to be done at each iteration.

4.2 Reference tracking

The open loop optimization problem described by equations 3.2 and 3.3 is written in the regulator form; in other words, the objective of the control system is to drive the state towards the origin. In this work, on the other hand, the objective is to drive the outputs of the system $y(k)$ to follow a given reference vector $r(k)$.

In order to use the results obtained for the regulator formulation also for the tracking is sufficient to perform an axis translation given by:

$$\begin{aligned} u &= u - u_{ss} \\ x &= x - x_{ss} \end{aligned} \tag{4.12}$$

where the steady state values x_{ss} and u_{ss} can be calculated by imposing:

$$\begin{aligned} f(x_{ss}, u_{ss}) &= 0 \\ y(x_{ss}, u_{ss}) &= r \end{aligned} \tag{4.13}$$

In particular using a linear prediction model the new equilibrium point can be easily calculated by:

$$\begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix} = \begin{bmatrix} W_x \\ W_u \end{bmatrix} r \tag{4.14}$$

the cost function 4.2 can be then transformed in:

$$J = \sum_{i=k+1}^{k+H_p-1} \left(\|x(i) - W_x r(i)\|_Q^2 + \|u(i) - W_u r(i)\|_R^2 \right) + \|x(k+H_p) - W_x r(k+H_p)\|_F^2 \quad 4.15$$

where the notation $\|x\|_Q^2$ denotes the quadratic form $x^T Q x$ and has been introduced in order to simplify notation.

In order to transform 4.15 in to the QP formulation a sequence of future references r and steady state values x_{ss} , u_{ss} are needed. There are two possible cases:

- the future set-point is known and, therefore, is possible to select the window corresponding to prediction horizon used in the optimization;
- only the actual reference vector $r(k)$ is known and is kept constant over the entire horizon.

The sequence of steady state values can be calculated from the reference in the two different cases respectively using equations 4.16 and 4.17

$$\begin{bmatrix} x_{ss}(k+1) \\ \vdots \\ x_{ss}(k+H_p) \end{bmatrix} = \begin{bmatrix} W_x & 0 \\ & \ddots \\ 0 & W_x \end{bmatrix} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+H_p) \end{bmatrix} \quad 4.16$$

$$\begin{bmatrix} x_{ss}(k+1) \\ \vdots \\ x_{ss}(k+H_p) \end{bmatrix} = \begin{bmatrix} W_x \\ \vdots \\ W_x \end{bmatrix} r(k+1) \quad 4.17$$

In the following equation 4.18 will be used for both cases

$$\begin{aligned} x_{ss} &= \overline{W}_x r \\ u_{ss} &= \overline{W}_u r \end{aligned} \quad 4.18$$

Again, the cost function can be written in the quadratic programming form obtaining:

$$J = \mathbf{u}^T (S_u^T \bar{Q} S_u + \bar{R}) \mathbf{u} + \mathbf{u}^T (2S_u^T \bar{Q} S_{x_0} x_0 - 2S_u^T \bar{Q} \bar{W}_x \mathbf{r} - 2\bar{R} \bar{W}_u \mathbf{r}) + const \quad 4.19$$

which, besides the already mentioned dependence from x_0 , depends from the future reference sequence allowing the system to take into account for the set-point.

4.3 Feasibility guarantee

In order to use the predictive approach for the flight control system the feasibility of the open loop optimal problem has to be guaranteed in any case. The main cause of unfeasibility is due to the presence of the terminal state combined with a finite prediction horizon. As explained in section 3.3 given an initial condition $x(k)$ can happen that is impossible to reach the terminal set within the prediction horizon. In order to overcome this problem three different approaches can be adopted:

- move the terminal set towards the initial condition shortening the path from the initial point to it;
- enlarge the terminal set obtaining the same effect as the previous method;
- increase the prediction horizon in order to give more time to the system to reach the terminal set.

4.3.1 Virtual reference

A terminal control law $u = k_f(x)$ has to be defined in order to guarantee the stability of the closed loop system. Since we are making use of linear state space models of the plant the more natural way to define it is by the classic linear feedback law

$$u = K(x - x_{ss}) + u_{ss} = Kx + Nr \quad 4.20$$

where K is obtained by the well-known linear quadratic regulator algorithm using the same weights used in the MPC cost function (Q and R) and $N = W_u - KW_x$. The stability conditions S1 and S2 are satisfied by defining the terminal set as:

$$X_f = \{x \in X \mid u_{\min} \leq Kx + Nr \leq u_{\max}\} \quad 4.21$$

which, among the rest, depends on the reference vector r .

This method makes use of this characteristic to move the terminal set towards the initial state $x(k)$ in order to guarantee feasibility. To do so the real reference r is substituted by a virtual reference z which is calculated together with the other optimization variables in order to minimize the cost function:

$$J = \sum_{i=k+1}^{k+H_p-1} \left(\|x(i) - W_x z\|_Q^2 + \|u(i) - W_u z\|_R^2 \right) + \|x(k+H_p) - W_x z\|_F^2 + \|z - r\|_T^2 \quad 4.22$$

is worth to notice that the first part is similar to the standard cost function 4.15 besides the fact that the system is driven to follow the virtual reference rather than the actual one; the additional term $\|z - r\|_T^2$ has the function to keep the virtual and the real set-points as near as possible in a quadratic sense.

The terminal cost can be calculated directly by imposing the stability condition S4 with strict equality:

$$(A - BK)^T F(A - BK) - F + (A - BK)^T Q(A - BK) + K^T B^T R B K = 0 \quad 4.23$$

As done before the optimization problem can be transformed into the quadratic programming formulation obtaining:

$$J = \begin{bmatrix} \mathbf{u} \\ z \end{bmatrix}^T \begin{bmatrix} S_u^T \bar{Q} S_u + \bar{R} & -2(S_u^T \bar{Q} \bar{W}_x + \bar{R} \bar{W}_u) \\ 0 & T + \bar{W}_x^T \bar{Q} \bar{W}_x + \bar{W}_u^T \bar{R} \bar{W}_u \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ z \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ z \end{bmatrix}^T \begin{bmatrix} 2S_u^T \bar{Q} S_{x0} & -2W_x^T \bar{Q} S_{x0} \\ 0 & -2T \end{bmatrix} \begin{bmatrix} x_0 \\ r \end{bmatrix} \quad 4.24$$

A final consideration has to be done on the computational burden associated to the solution of the optimal problem. Indicating with m the dimension of the input u and with v the dimension of r is possible to calculate the dimension of the optimization vector which

gives a measure of the complexity of the optimal problem. Considering the case where the reference is known over the entire prediction horizon and the virtual reference has to be calculated accordingly the dimension of the optimization vector becomes:

$$h = (m + v)H_p \tag{4.25}$$

while considering the reference constant over the entire horizon the dimension reduces to:

$$h = mH_p + v \tag{4.26}$$

posing $m=4$ and $v=4$, how is normal in the helicopter control problem, the computational burden in the second case is 45 to 50% lower than the first case depending on the prediction horizon chosen and, hence, only the second case has been implemented.

In order to evaluate the performances of this architecture the speed control of the helicopter has been implemented. To do so is sufficient to introduce the model of the helicopter body dynamics as the plant model and select the body frame speeds u , v , w and the heading angle ψ as outputs of the system.

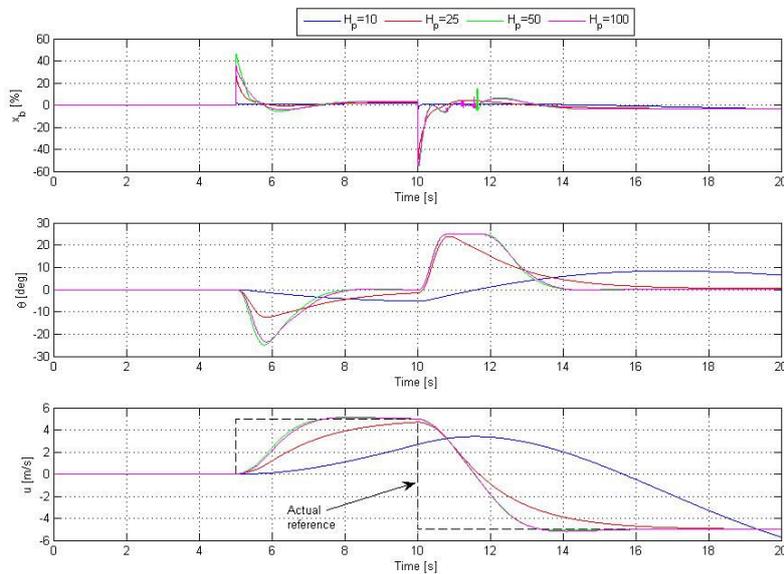


figure 4.1: performances as function of prediction horizon

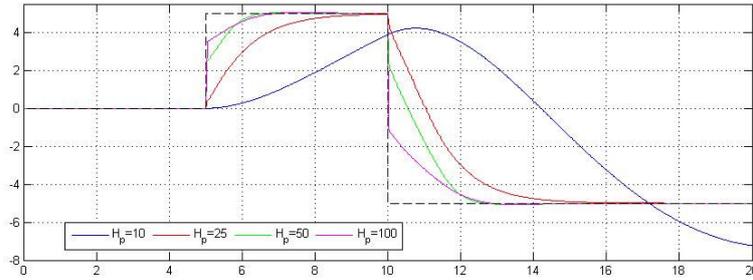


figure 4.2: virtual set-point as function of prediction horizon

Besides the obvious variations due to the calibration of the weight matrices Q and R is interesting to notice the influence of the prediction time on the performances, figure 4.1 shows the results obtained by changing the prediction horizon from 10 to 100 steps.

As can be seen going from 10 to 50 steps the performances progressively improve decreasing the settling and rise times. This changing can be explained by looking at the virtual reference plots reported in figure 4.2, with longer prediction horizons the system can reach states further away from the initial condition and this allows to choose a virtual reference nearer the actual one. Conversely a short prediction horizon forces the optimizer to choose a reference near the initial condition rather than near the reference obtaining poor performances.

By further increasing the prediction horizon, in the figure going from 50 to 100 steps, there is no gain in performances since the dynamics of the system are dominated by the constraints, in particular the constraint imposed on the attitude angles (± 25 degrees) limits the acceleration to a maximum which can not be violated even choosing higher values of reference.

It is clear that such behavior is function not only of the prediction horizon but of the reference itself, moreover the best performances are obtained for longer horizons and, hence, with higher computation burdens. Such behavior is an heavy drawback for this method since relates the computational capabilities of the FCC with the performances of the control algorithm.

4.3.2 *Infinite horizon*

In order to guarantee feasibility of the open loop optimal problem a second method is to enlarge the prediction horizon; taking this concept to the limit this control architecture is

implemented considering an infinite horizon, is trivial to understand that with this choice is always possible to reach the terminal set however it is defined from every possible state. The stability conditions can be met by using:

$$\begin{cases} k_f(x) = 0 \\ X_f = \{0\} \\ F = 0 \end{cases} \quad 4.27$$

on the other hand the standard formulation does not allow the system to be solved since an infinite number of unknown variables have to be found. In order to overcome a problem the control horizon is introduced, the first H_c control moves are allowed to change and are calculated by the optimization algorithm while the other inputs are obliged to be $u = u_{ss}$.

The cost function then becomes:

$$J = \sum_{i=k+1}^{k+H_c-1} \left(\|x(i) - x_{ss}\|_Q^2 + \|u(i) - u_{ss}\|_R^2 \right) + \sum_{i=k+H_c}^{+\infty} \|x(i) - x_{ss}\|_Q^2 \quad 4.28$$

the second term is still impossible to be calculated numerically but, if the system is stable, converges to:

$$\sum_{i=k+H_c}^{+\infty} \|x(i) - x_{ss}\|_Q^2 = \|x(k + H_c) - x_{ss}\|_P^2 \quad 4.29$$

where P is the solution of the discrete Lyapunov matrix equation:

$$A^T P A - P + Q = 0 \quad 4.30$$

Since the helicopter is unstable relation 4.29 cannot be applied directly but the system has to be stabilized first. To do so an inner control loop has been added as depicted in figure 4.3

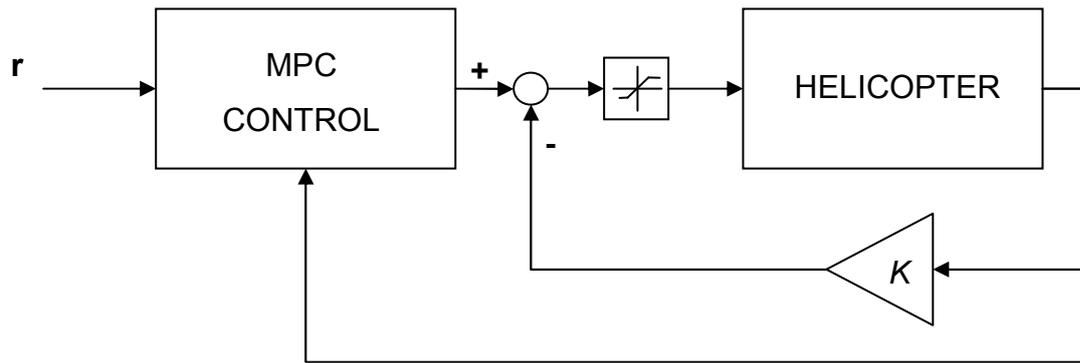


figure 4.3: inner stabilizer loop

again, a linear feedback law has been used but in this architecture the weight matrices used are not the same of the MPC cost function but are calibrated in order to satisfy this linearity condition:

$$u_{\min} \leq -Kx \leq u_{\max} \quad \forall x \in [x_{\min}, x_{\max}] \quad 4.31$$

In this way is possible to guarantee that the plant seen by the MPC system is stable for every possible state of the system.

At this point the model used to calculate the steady state condition and to predict the future response becomes:

$$x(k+1) = (A - BK)x(k) + Bu(k) \quad 4.32$$

where the inputs are the reference for the stabilizer rather than the helicopter controls. The input constraints have to be modified as follows:

$$u_{\min} \leq u - Kx \leq u_{\max} \quad 4.33$$

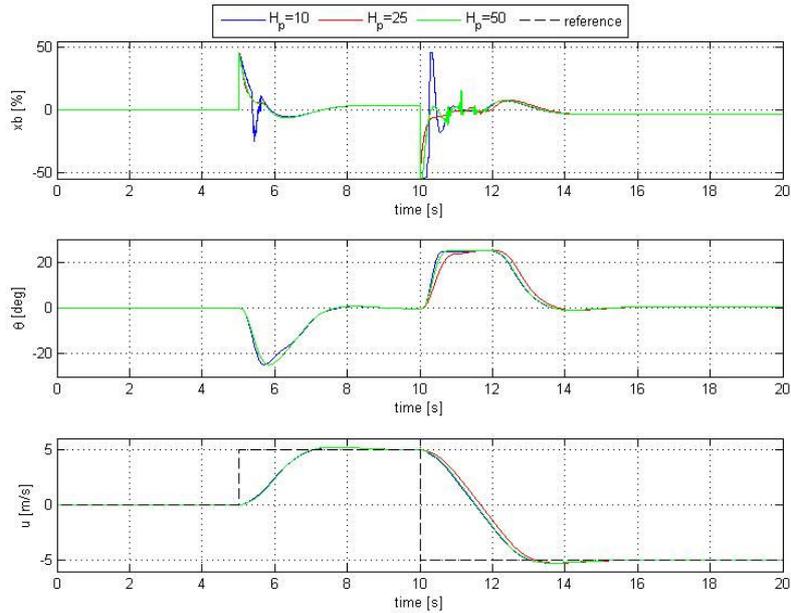


figure 4.4: infinite horizon performances

This architecture has been tested in the same framework used in the previous one and the results are shown in figure 4.4. As can be seen the prediction horizon has little influence on the performances, all the signals are similar going from 10 to 50 prediction steps. Moreover is possible to notice that the response is similar to the best performances obtained with the previous architecture.

Is worth to notice that the same architecture can be obtained enlarging the terminal set. If we choose $X_f = X$, which obviously guarantees feasibility for every $x \in X$, and a linear terminal control law $k_f = -Kx$ the stability condition S2 corresponds to 4.31 and the terminal cost calculated by 4.30 or by 4.23 are equivalent.

4.3.3 Soft constraints

The input constraints correspond to the physical limitation of the actuators and cannot be exceeded in any case. On the other hand the state constraints are imposed by the designer for several different reasons but do not correspond to any physical limitation of the helicopter. An example of this is the attitude angle, in the previous examples has been limited to be less than 25 degrees, but the helicopter can reach also higher values.

Modelling errors or disturbances, such as wind gusts, can drive the system outside the limits imposed by the state constraints, when this happens the optimization problem could become unfeasible. This happens when, given all the constraints the first states in the predicted sequence cannot be driven back into the admissible state subspace X .

Since it is very likely that this condition will occur during real flights a way to restore feasibility is needed, two approaches can be used in this case:

- a. detect the situation and use a different controller,
- b. formulate the MPC architecture in order to be robust to such a situation.

The first approach is more complicated since it requires a way to detect unfeasibility and needs a second control system capable of dealing with the situation, conversely the second approach is more attractive even though it implies a modification of the optimal control problem.

The solution adopted distinct two different classes of constraints:

- hard constraints which can not be exceeded under any circumstances (like the input constraints)
- soft constraints that can be exceeded only if feasibility can not be obtained otherwise.

The first class has been applied to the input constraints since those have to be respected anyway while the state constraints have been softened in order to enable the system to restore feasibility. To do so a set of slack variables ε has been introduced in the state constraints formulation:

$$x_{\min} \leq x - \varepsilon \leq x_{\max} \tag{4.34}$$

by means of these variables the admissible state subspace can be moved in every direction as necessary to restore feasibility. In order to use such possibility only when necessary the vector of slack variables is weighted in the cost function transforming it into:

$$J = \sum_{i=k+1}^{k+H_p-1} (\|x\|_Q^2 + \|u\|_R^2) + \|x(k+H_p)\|_F^2 + \|\varepsilon\|_S^2 \quad 4.35$$

figure 4.5 shows the comparison between two different controllers, both have been implemented using the infinite horizon architecture but one uses all hard constraints while the second implements the soft constraint architecture described above.

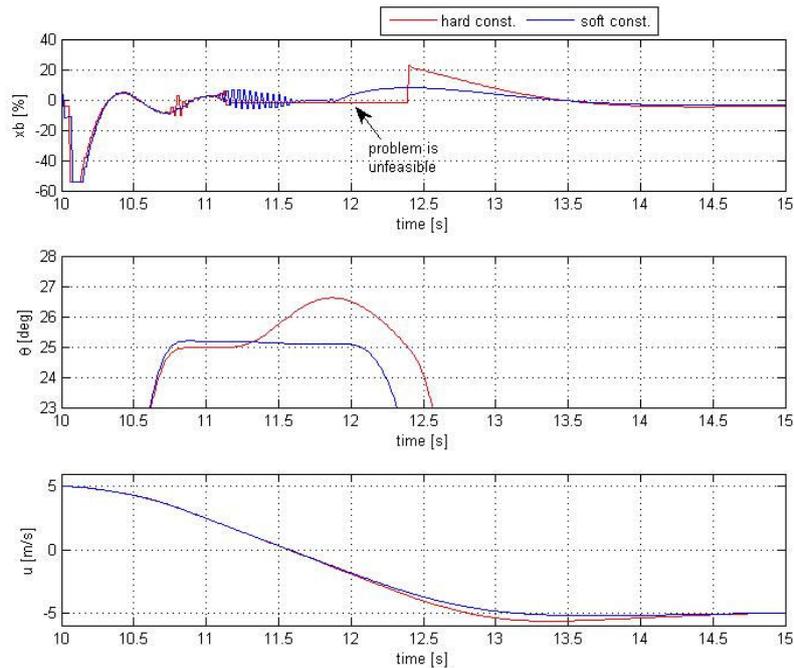


figure 4.5: comparison between soft and hard state constraints

In the first case the problem becomes unfeasible after 11.2 seconds from the beginning of the simulation, the input is set to the steady state value and the system is controlled only by the inner stabilization loop. As can be seen the unfeasibility starts as soon as the attitude exceeds the limit value and no control is given by the MPC algorithm until the inner stabilizer drives back the system into the feasible region; it is worth to point out that once the MPC stops working the stability of the closed loop is not guaranteed anymore especially when controlling the non linear plant.

Conversely when the state constraints are softened the unfeasibility does not occur even if the attitude reaches values higher than the 25 degrees limit. At this point the MPC keeps controlling the system and all the stability guarantees are valid.

4.4 Offset-free control

One of the more common requirements for a control system is to avoid errors at regime. The presence of mismatches between the real behavior and the predicted dynamics, due to modeling errors and external disturbances, is the cause of such errors in the predictive approach. Once again is possible to modify the control system architecture in order to avoid this behavior. One possible approach is quite classic and consists into adding the integral of the tracking error into the state vector and penalize it into the cost function. A less obvious approach makes use of a particular formulation of the plant model where the inputs are moved into the state vector; an observer estimates them by using only the information on the system outputs coming from the sensors.

4.4.1 Integral action

The idea of compensate for steady state errors using an integral action is based on the classic PID architecture. In order to implement this feature into the MPC control the first step is to obtain a prediction model where the integral of the tracking error e is calculated. To do so is sufficient to expand the linear model as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad 4.36$$

the future state becomes function of two different inputs: the actual inputs u which have to be optimized and the reference r that has to be treated as a parameter by the optimization algorithm.

In order to obtain the future control sequence is possible to use the same procedure seen before and obtain:

$$\mathbf{x} = S_{x_0} x_0 + S_u \mathbf{u} + S_r \mathbf{r} \quad 4.37$$

which can be included into the cost function in order to obtain the QP formulation:

$$J = \mathbf{u}^T (S_u^T \bar{Q} S_u + \bar{R}) \mathbf{u} + \mathbf{u}^T (2S_u^T \bar{Q} S_{x_0} x_0 - 2S_u^T \bar{Q} \bar{W}_x \mathbf{r} - 2\bar{R} \bar{W}_u \mathbf{r} + 2S_u \bar{Q} S_r \mathbf{r}) + const \quad 4.38$$

The control scheme is then modified in order to include the feedback of the integral error as depicted in figure 4.6

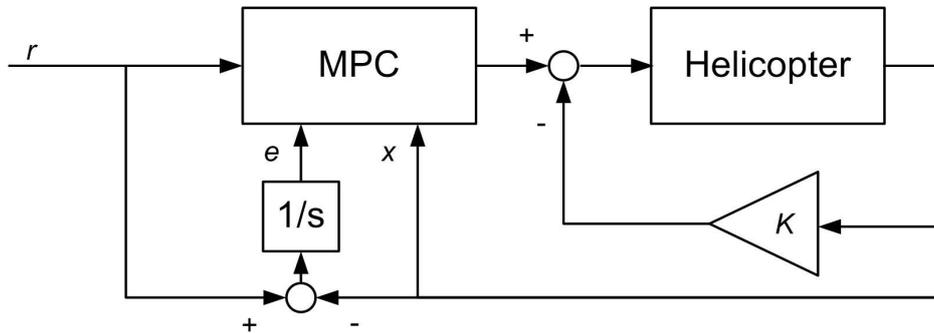


figure 4.6: Integral action architecture scheme

In figure 4.7 the comparison between two different controllers is shown. The infinite horizon control developed in section 4.3.2 and the same controller augmented with the integral action.

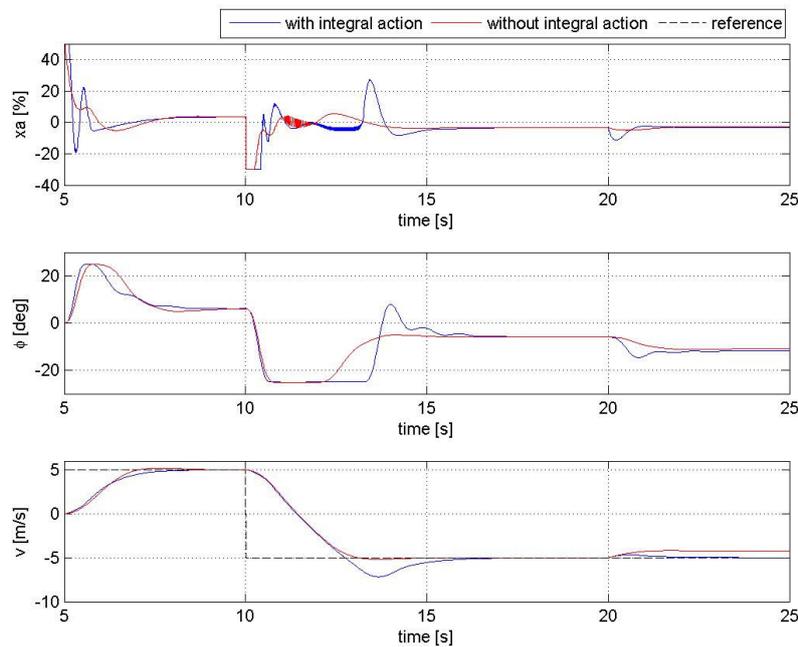


figure 4.7: effects of the integral action

A small disturbance has been introduced at the second 20, as can be seen the baseline controller, indicated in red, presents a constant tracking error. Conversely the introduction of the integral action allows the system to cancel this error.

On the other hand a significative overshoot can be noticed on the step response of the system when the integral action is introduced. This is caused by the interaction between the constraints and the integral called wind-up. As can be seen the lateral acceleration of the helicopter his limited by the attitude constraint, when this limit it reached the integral of the error keeps increasing without any further effect on the performances. When the set-point is reached the contribute of the integral is higher than the necessary and the system keeps accelerating. At this point the error is inverted and the integral error decreases reaching the correct value.

In the control literature several anti wind-up schemes can be found, all rely on the same basic concept: when the saturation is reached the integration action is freezed at the reached value. The same concept has been applied here by using the following relations:

$$\begin{cases} \dot{e}_i = r_i - y_i & x_{\min,i} \leq x_i \leq x_{\max,i} \\ \dot{e}_i = 0 & \textit{otherwise} \end{cases} \quad 4.39$$

where the subscript i identifies the single component of the error vector and are reported in Table 4.1.

i	r_i, y_i	x_i	$x_{\min,i}$	$x_{\max,i}$
1	u_{sp}, u	\mathcal{G}	\mathcal{G}_{\min}	\mathcal{G}_{\max}
2	v_{sp}, v	φ	φ_{\min}	φ_{\max}
3	w_{sp}, w	x_c	$x_{c\min}$	$x_{c\max}$
4	ψ_{sp}, ψ	r	r_{\min}	r_{\max}

Table 4.1: anti wind-up limit variables

The anti wind-up scheme adopted adds a non linearity on the controlled plant and, therefore, is not possible to include this feature on the prediction model. This leads to a mismatch between the predicted integral error and the one actually calculated but this error has no consequences on the practical application of this control architecture.

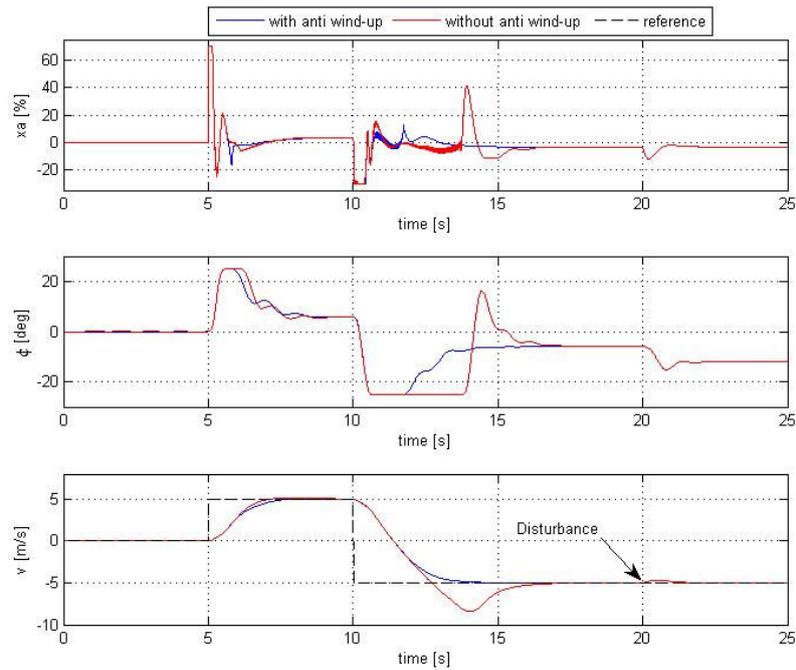


figure 4.8: effect of the anti wind-up system

In figure 4.8 the effect of the anti wind-up system is clearly shown: the ability of canceling the steady state error has been maintained without variations while the overshoot caused by the integral wind-up has been eliminated obtaining lot better tracking performances.

Finally a robustness analysis has been performed by means of Monte Carlo simulations, the controlled plant has been substituted by the uncertain model described in section 2.5 and the uncertainties have been randomly chosen at each simulation.

In figure 4.9 a representative set of the simulations has been reported, is worth to notice that all the constraints have been respected and the tracking error at regime has been cancelled by the presence of the integral action.

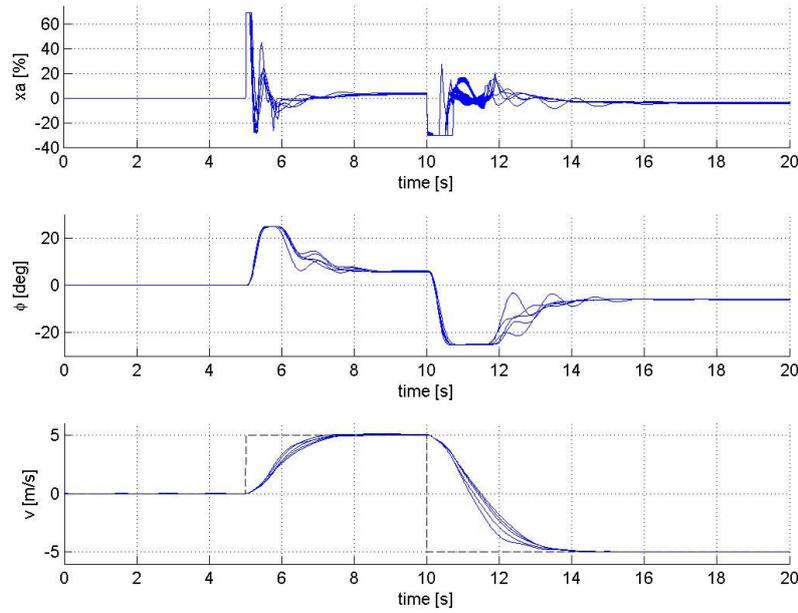


figure 4.9: Monte Carlo simulation

4.4.2 State observer

In order to obtain offset-free control with MPC a different approach has been proposed in [17], the prediction model is augmented by using the delta- u formulation reported in equation 4.40 and an observer estimates both the states x and inputs u in order to minimize the mismatches between the predicted and the real outputs.

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k) \quad 4.40$$

These estimates are calculated as follows:

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{u}(k+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{u}(k) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k) + K_{obs} (C\hat{x}(k) - y(k)) \quad 4.41$$

and are used by the MPC algorithm as initial states of the optimization. The overall control scheme is reported in figure 4.10.

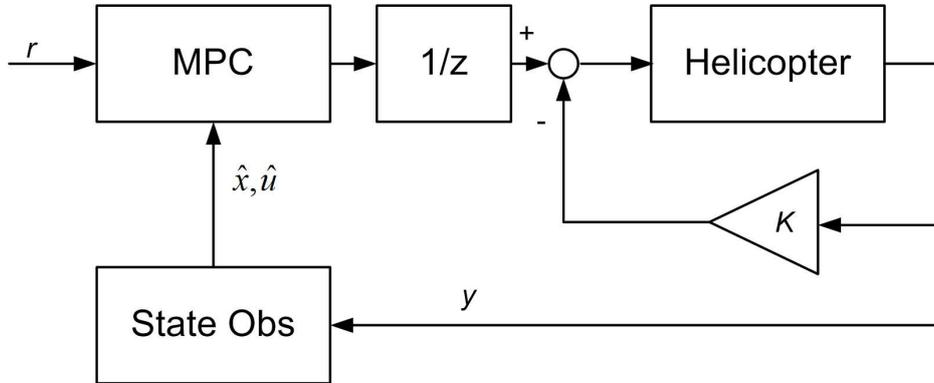


figure 4.10: state observer control architecture

In this scheme the offset free control is achieved since the mismatches between the predicted outputs and the actual ones are lumped into the estimates of the unmeasured variables and the inputs.

In figure 4.11 the results of a Monte Carlo simulation performed with the state observer control architecture are shown. Zero steady state error has been obtained but the attitude constraints has not been respected, this is caused by the fact that the real and the estimated values are different and the actions taken by the MPC control to respect the constraints do not obtain the desired result on the real plant.

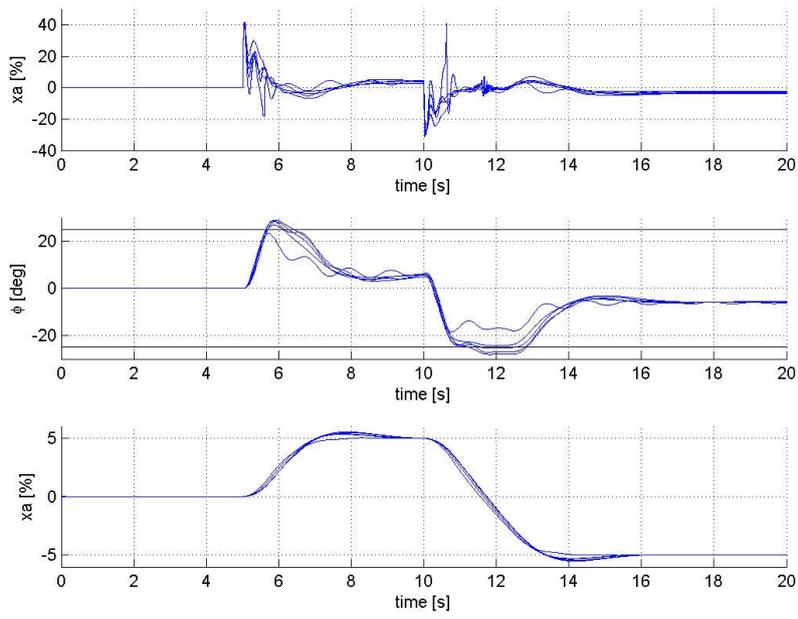


figure 4.11: Monte Carlo simulation

5 Stability and Control Augmentation System

One of the more diffused ways to control the motion of the UAS is through remote piloting; the operator directly controls the air vehicle from the ground station using an Human Machine Interface (HMI) really similar to those used in manned aircrafts.

As already mentioned helicopters are unstable and heavily coupled systems and are really difficult to pilot, the SCAS system has the aim of stabilize and decouple all the controlled axis of the helicopter in order to simplify the task of the operator; moreover the same system could be also used as an inner control loop of more complex Guidance, Navigation and Control (GNC) systems.

In the following sections the control system architecture will be described in depth and a set of simulations will be shown in order to demonstrate its performances.

5.1 SCAS control structure

The aim of the SCAS is to control the body speeds and the heading of the helicopter, since these variables are included in the standard linear approximation of the plant (see eq.2.5) this can be directly used into the control system definition. Given the analysis presented in section 4 the following base architectures have been joined together:

- the infinite prediction horizon control,
- the integral error action with the antiwind-up correction
- the soft constraints formulation.

For clarity in the following the complete controller formulation is reported. The inner stabilizer loop has been obtained by the classic LQR control procedure imposing the weight matrices as:

$$Q = I, \quad R = 1000I \quad 5.1$$

and the prediction model has been set as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad 5.2$$

with

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad 5.3$$

The cost function to be minimized is

$$J = \sum_{i=k+1}^{k+H_c-1} \left(\|x_a(i) - W_x r\|_Q^2 + \|u_a(i) - W_u r\|_R^2 \right) + \|x_a(k+H_c) - W_x r\|_P^2 + \|\mathcal{E}\|_S^2 \quad 5.4$$

where the subscript *a* indicates the state and inputs of the augmented model 5.2. The weight matrices *Q* and *R* are different from the ones used to calibrate the inner stabilizer and have been tuned in order to avoid response overshoot for low and medium aggressive manoeuvres while maximizing performances.

Finally, the input and state constraints are reported in Table 5.1

Variables	Symbol	Limits
Inputs	u	$-u_{tr} \div 100 - u_{tr}$
Flapping angles	a, b, c, d	$-15 \div 15 \text{ deg}$
Angular speeds	p, q, r	$-2 \div 2 \text{ rad} / \text{s}$
Long and lateral speeds	u, v	$-10 \div 10 \text{ m} / \text{s}$
Vertical speed	w	$-5 \div 5 \text{ m} / \text{s}$
Long and lat. attitude angles	ϑ, ϕ	$-35 \div 35 \text{ deg}$
Yaw angle	ψ	$-360 \div 360 \text{ deg}$

Table 5.1: SCAS constraints

In order to point out the advantages given by the use of the model predictive approach, in particular the ability of enforcing constraints and use of the information about future set points, an LQR control system has been developed and calibrated for comparison purposes. This control approach has been chosen since maintains the same basic structure of the MPC, in particular handles multiple input multiple output systems and makes use of a cost function similar to the one used in the predictive controllers.

Moreover in the SCAS also the integral action has been included by using the following augmented system as plant model

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad 5.5$$

The baseline controller has been calibrated in order to have similar performance with respect to the MPC system for low and medium aggressive manoeuvres.

5.2 Simulation results

Two different sets of simulations have been carried on in order to evaluate the controller characteristics. In the first set the SCAS has been applied to a linear model of the helicopter identical to the one used in the prediction model, in the second set model

uncertainties have been applied to simulation model and a Monte Carlo simulation have been carried on to evaluate the robustness characteristics of the system.

Figures 5.1 and 5.2 show the results obtained in two separate simulations where a series of acceleration manoeuvres were performed on respectively the longitudinal and lateral axis. The speed set-point has been designed in order to obtain manoeuvres with increasing aggressiveness in order to evaluate the response of both the systems.

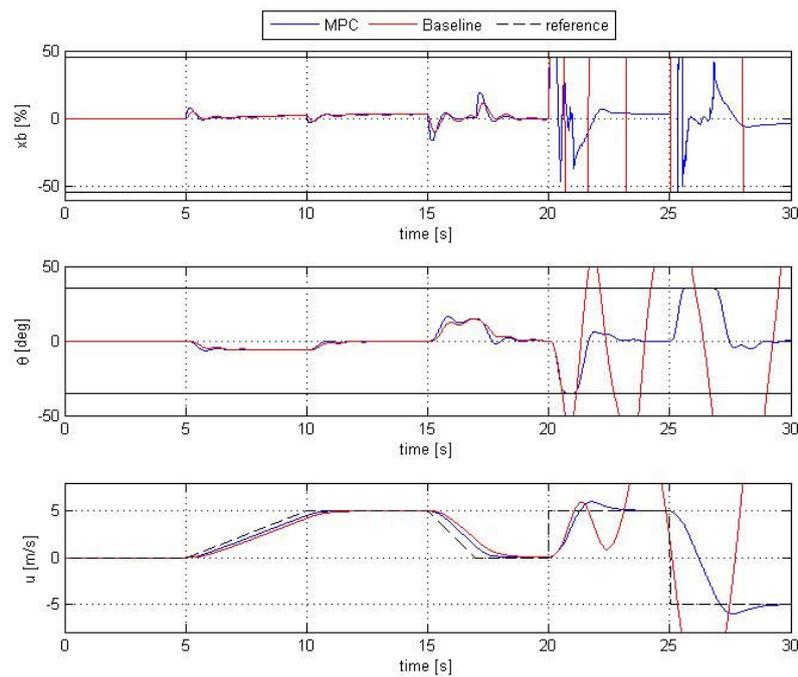


figure 5.1: longitudinal accelerations

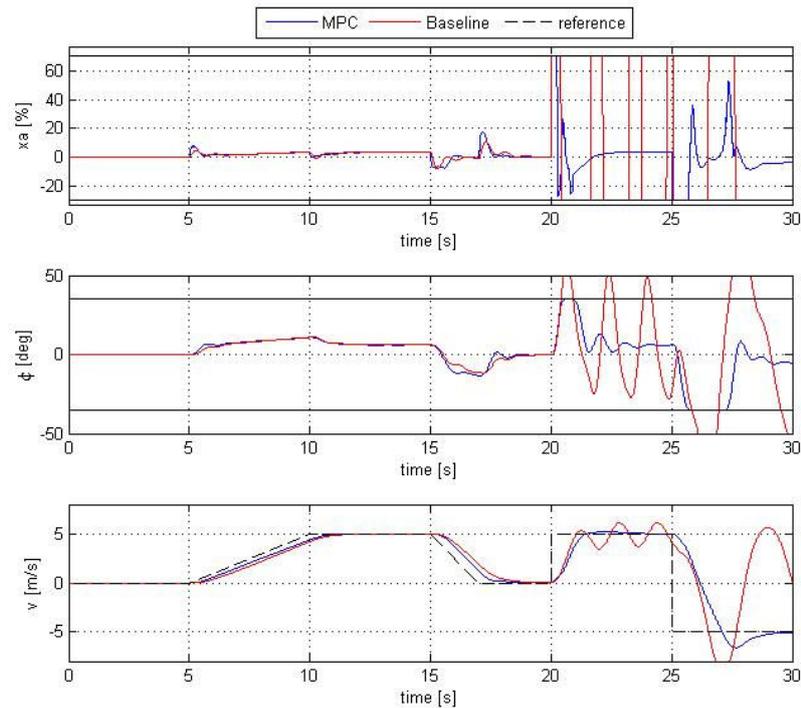


figure 5.2: lateral accelerations

As expected during the low and medium aggressive maneuvers the response of input, attitude and speed variables is quite similar comparing the MPC and the LQR controllers. A significative difference can be noticed when the system is asked to perform really aggressive maneuvering, in this case the MPC controller has the ability of stabilizing the system and perform the maneuvers with a smooth behavior. Moreover both the input and attitude constraints are respected.

On the contrary the baseline SCAS does not have the information about input constraints and, when the calculated input is significantly larger than the saturations, the system loses the ability of stabilizing and controlling the helicopter.

In order to evaluate the robustness characteristics of controller a Monte Carlo simulation has been performed using the uncertain model shown in section 2.5. In figure 5.3 a reduced number of simulations have been reported, as can be seen both the input and attitude constraints are still respected by the control system while offset free tracking is achieved.

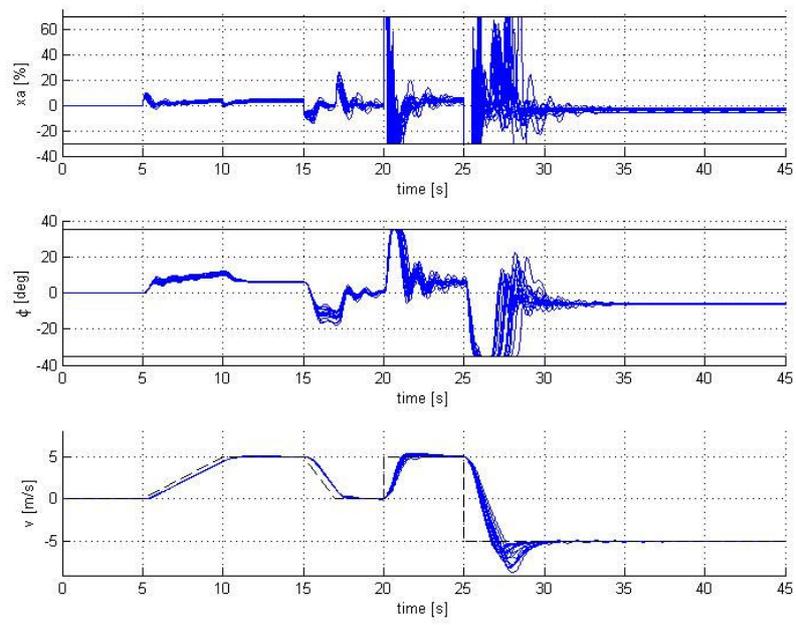


figure 5.3: Monte Carlo Simulation

6 Trajectory Tracking System

In order to increase the level of autonomy of the system and reduce the work load of the human operator, the position control of the rotorcraft can be automatically executed by the FCS. The motion control of air vehicles can be roughly divided in two different classes: trajectory tracking, where the UAS is required to track a time parameterized reference, and path following, where the vehicle is required to converge and follow a given path without any time specification.

In this section the first approach is presented, the trajectory is constituted by a sequence of three-dimensional positions, expressed in the North-East-Down (NED) reference frame, and a target heading angle to be reached at a specific time.

Differently from the classic approach, which divides the vehicle guidance and the control problem into an inner and an outer loop, the system presented here has been built using an integrated approach an both the navigation and control problems are solved simultaneously.

As done for the SCAS first the system architecture will be presented and explained into the details, afterwards a set of simulations will be shown to demonstrate the system capabilities.

6.1 System architecture

Normally the position of the helicopter is expressed into an earth fixed reference frame, typically the NED frame; on the other hand the helicopter dynamics is expressed into the body frame. In order to calculate the actual position of the helicopter is necessary to rotate the body speed from one frame to the other using the actual Euler angles; since this operation is non-linear it can not be directly included in the prediction model for the MPC architecture used here.

A linear approximation of the helicopter kinematics can be obtained by expressing the position into the body frame axis at the beginning of the horizon and use it trough all the prediction horizon. If the change of attitude over the horizon is small the position dynamics can be approximated as:

$$\dot{s} = [u, v, w] \tag{6.1}$$

where s is the position of the helicopter expressed in the new frame.

The full linear model used to predict the future state of the helicopter can be expressed as

$$\begin{bmatrix} \dot{x}_b \\ \dot{s} \end{bmatrix} = \begin{bmatrix} A & 0 \\ L & 0 \end{bmatrix} \begin{bmatrix} x_b \\ s \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_b \tag{6.2}$$

where L is a matrix, which selects the body speeds from the state vector x_b .

At this point is sufficient to rotate both the actual position and the future reference from the NED to the actual body frame to close the loop.

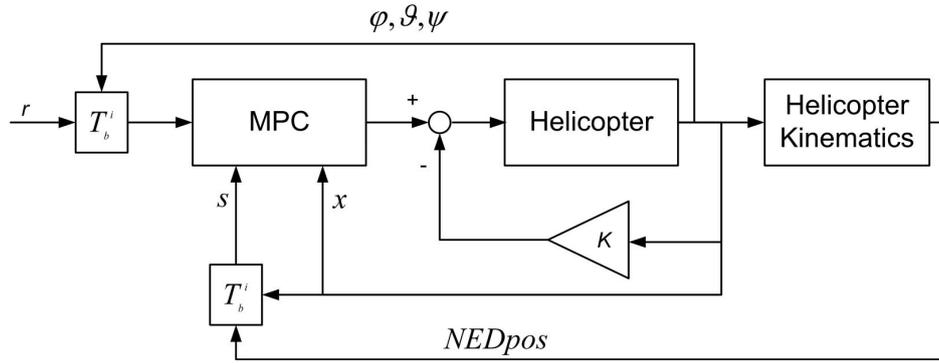


figure 6.1: Trajectory tracking system architecture

Finally, regarding the MPC architectures proposed in section 4 an infinite prediction horizon has been used together with a soft state constraints formulation, in this case the integral action is not needed as simulations will demonstrate.

The baseline controller has been built around the same linear approximation of the helicopter kinematics used for the MPC, the system 6.2 has been used as a plant and the following control law has been implemented.

$$u = Nr - Kx \quad 6.3$$

where

$$N = W_u + KW_x \quad 6.4$$

6.2 Simulation results

As done for the SCAS the first set of simulations presented shows the results obtained by the system when the helicopter dynamics and the model used in the prediction are identical. Is worth to notice that, in this case, a difference between the prediction model and the controlled plant is still present due to the linear approximation of the helicopter kinematics shown in the previous section.

Figures 6.2 and 6.3 show again two different simulation where a reposition manoeuvre has been performed on the longitudinal and lateral axis respectively. As for the SCAS, the

performances of the MPC and the baseline controller are the same for low and medium aggressive manoeuvres. It is important to notice that for the medium and high aggressive manoeuvres the LQR controller does not respect the attitude constraints, in particular for the aggressive repositions the attitude angles exceed the 90 degrees which clearly unfeasible on the real helicopter.

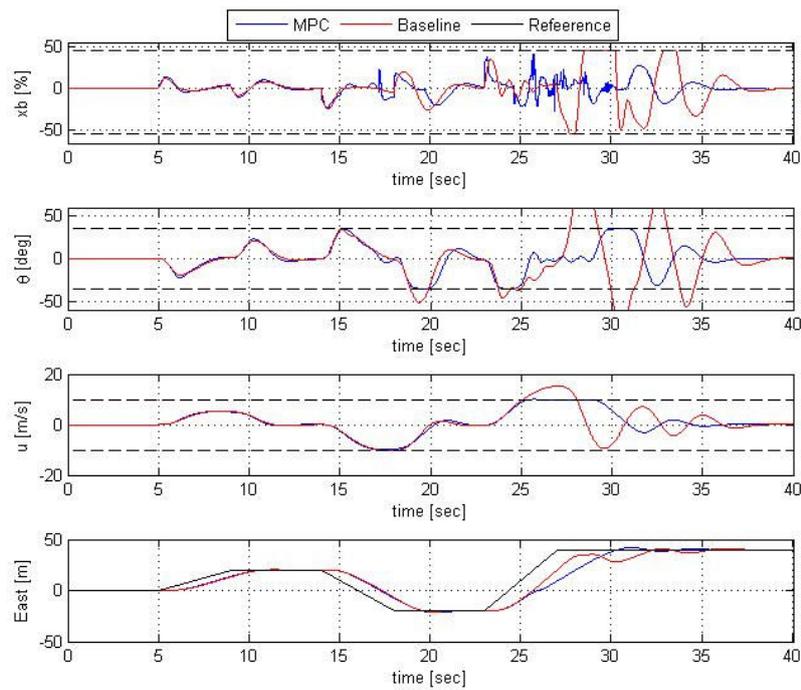


figure 6.2: longitudinal reposition

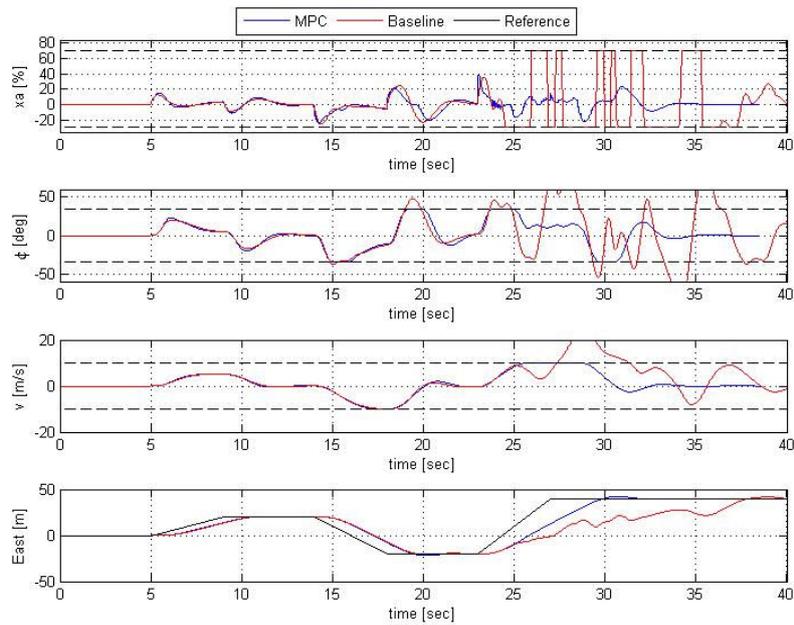


figure 6.3: Lateral reposition maneuvers

Another advantage given by the predictive control formulation is the ability of taking into account information about the future set-point when this is available. The control of the position of an UAS is the typical case where this happens since the desired flight plan is normally delivered before starting the mission.

In order to evaluate the change of performances given by such ability the previous control formulation has been modified in order to use a future sequence reference.

Figures 6.4, 6.5, 6.6, 6.7 show respectively the position, speed, attitude and controls of the helicopter performing the lateral reposition maneuvers under the control of the standard and modified MPC systems.

Observing the lateral axis variables, namely the East position, the lateral speed and the roll angle is possible to see how the information about the future set-point allows the system to anticipate the maneuver reducing significantly the tracking error.

Moreover the modified system starts the maneuvers in a smoother way which is reflected on the other axis response where the error is significantly reduced.

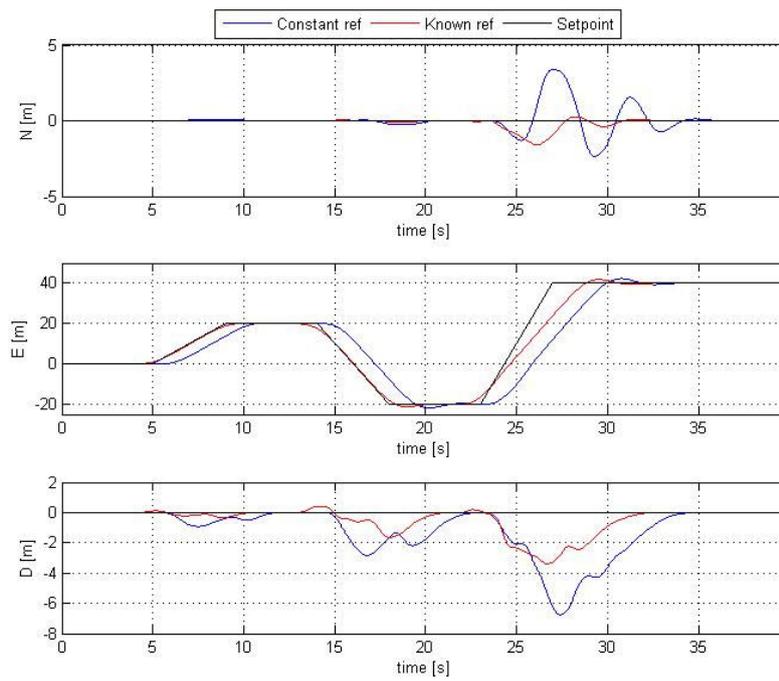


figure 6.4: Constant vs. known reference, NED position

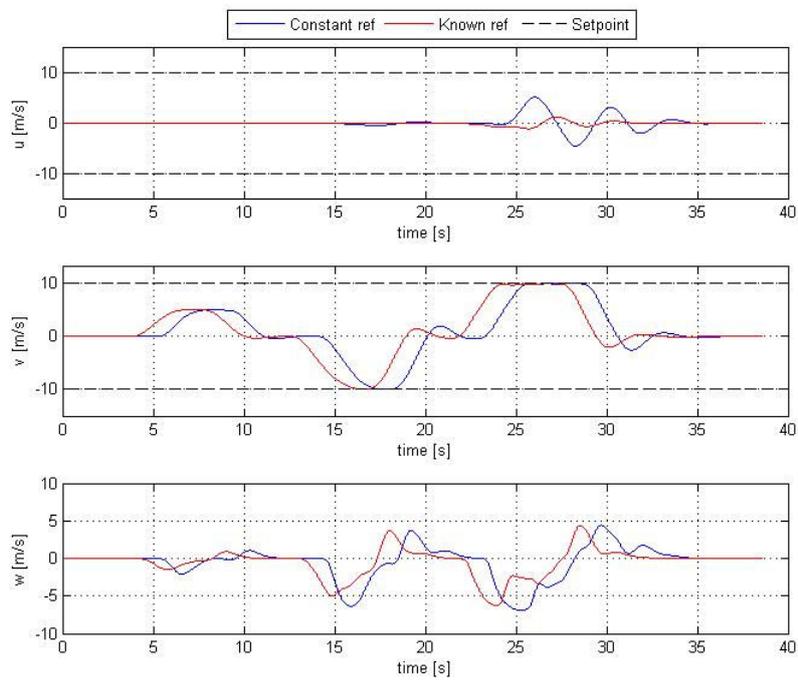


figure 6.5: Constant vs. known reference, body speeds

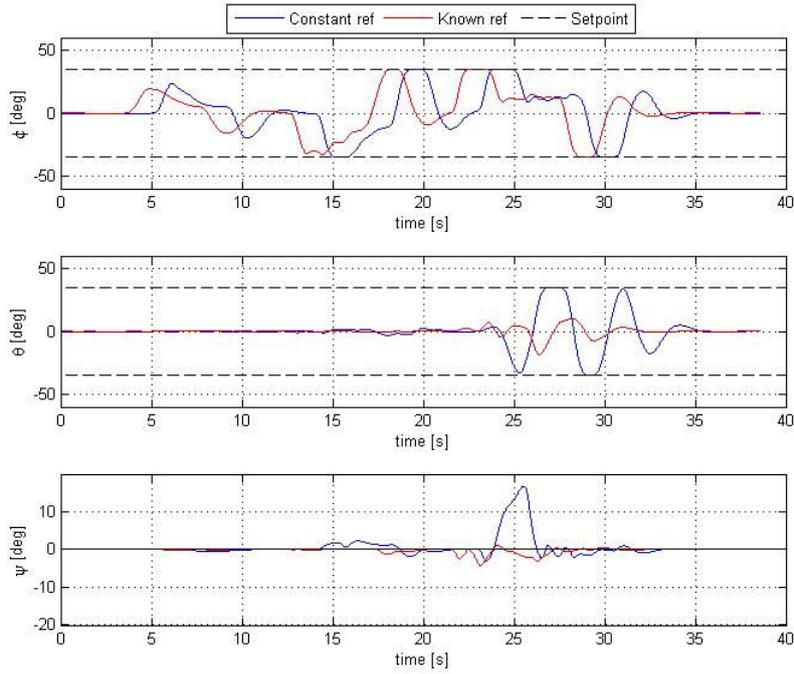


figure 6.6: constant vs. known reference, attitude angles

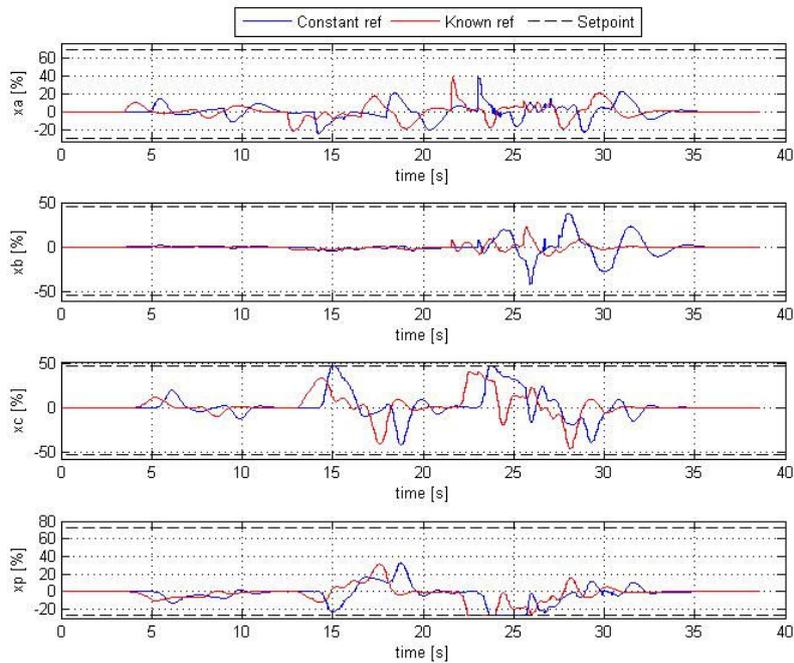


figure 6.7: constant vs. known reference, controls displacement

The robustness test has been carried on the predictive architecture, as observed for the SCAS controller the predictive architecture is able of maintaining stability and the ability of

enforcing all the constraints. Moreover, as anticipated, zero steady state error is achieved without the need of an integral action.

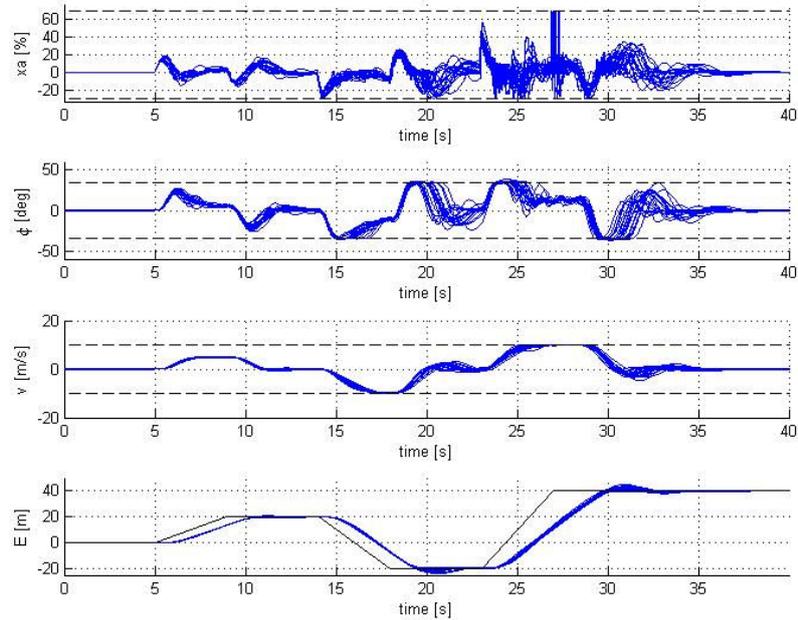


figure 6.8: Monte Carlo simulation

Finally a measure of the computational burden associated to the control has been evaluated measuring the time needed to solve the online part of the control algorithm, in order to obtain real time performances this time as to be less than one sample interval, in the specific case 50ms.

The tests have been carried on a laptop computer equipped with an Intel Centrino Duo processor whose computational capability has been measured using a benchmarking software and obtaining a capacity of 15 GFLOPS (Giga Floating point Operations per second). A standard embedded controller for aerospace applications has a more limited capability of about 2 GFLOPS [23], in order to understand if such a controller is capable of real time performances the times obtained in simulation have been increased proportionally to the computational capabilities and the results reported in

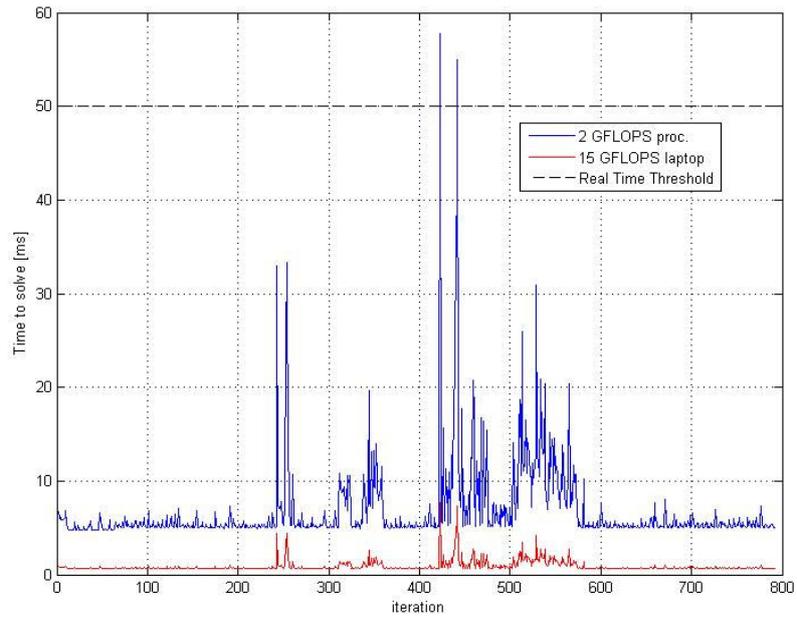


figure 6.9: Time to obtain the control action

As can be seen the control algorithm is fast enough to be implemented on a standard control system.

7 Path Following System

In this section the path following approach is used for controlling the helicopter's position. As explained previously the aim of such system is to converge on and follow a given path with no time specification. In other words the control action has to be function only on the actual position and not of the time at which the position is reached.

Conversely, the predictive approach is a time domain technique and cannot be directly applied to solve the path following problem but can be used transforming the latter in a sequence of trajectory tracking problems.

At each sampling instant the system calculates a future reference trajectory as function of the desired path and the current position then, by means of the system developed in the section 6, the control actions needed to follow the trajectory are calculated maintaining the MPC advantages.

To calculate the current reference trajectory the system simply finds the point on the path nearer the helicopter, this point becomes the starting point of the future trajectory. Then the desired position is propagated in the future by a simple linear interpolation between the waypoints.

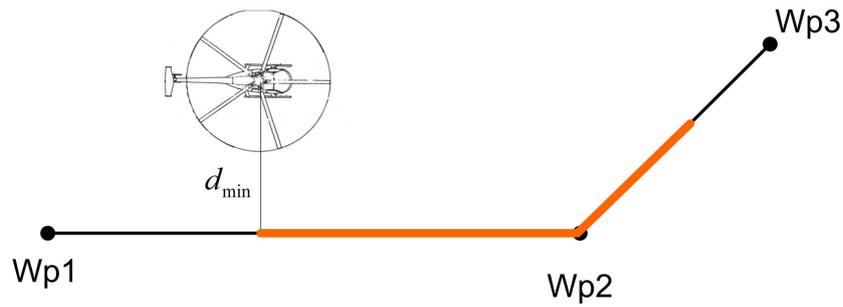


figure 7.1: Path following principle

As usual the performances of the path following system will be compared with the ones of the trajectory tracking system with known reference while executing a series of manoeuvres. The first test shown reports the results following a path with different angles turn that has to be performed keeping the heading fixed to the north.

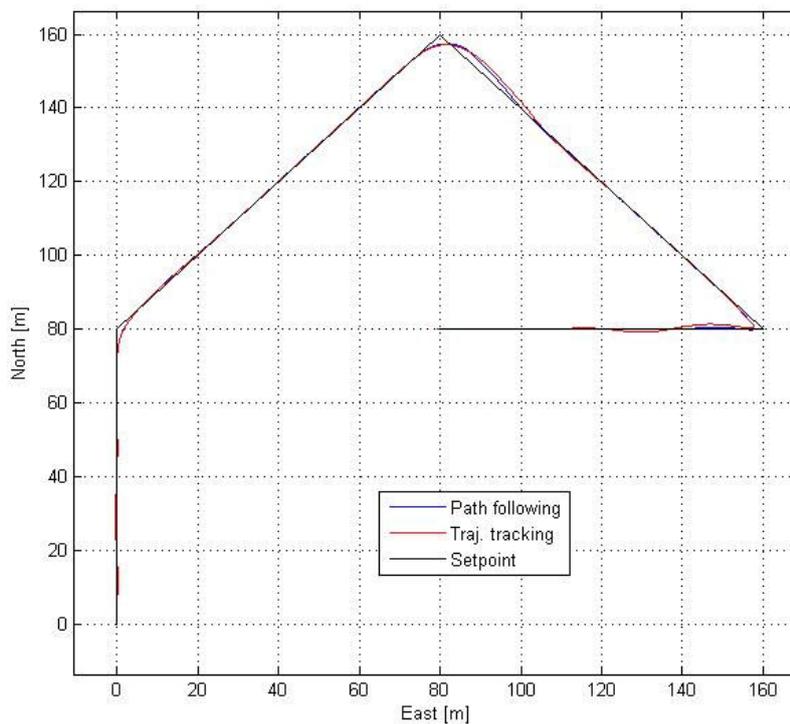


figure 7.2: path following vs. trajectory tracking

In figure 7.2 the trajectory followed by the two systems is shown, as can be seen the performances are quite similar. The difference can be noticed examining the time histories of the speeds, figure 7.3, and attitude, figure 7.4. While the trajectory tracking system needs to compensate for the non instantaneous accelerations with a speed overshoot the

path following system reaches the target speed without this problem, this is also reflected in a smoother maneuvering and a lower demand in attitude angles.

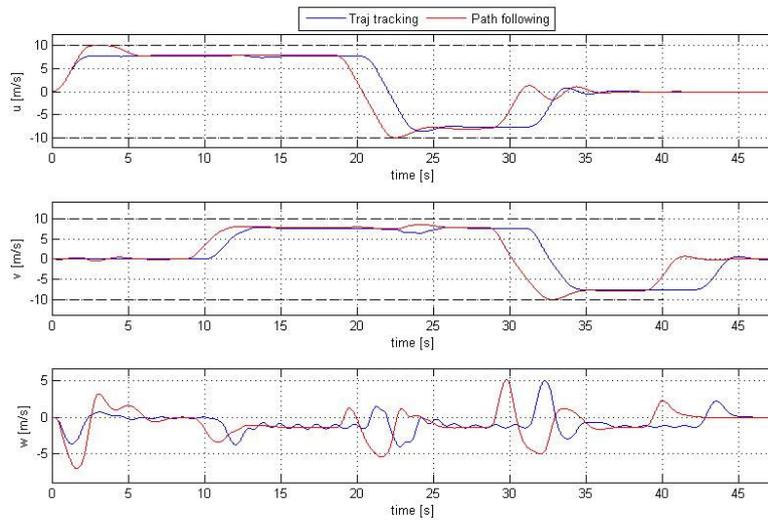


figure 7.3: speed time plot

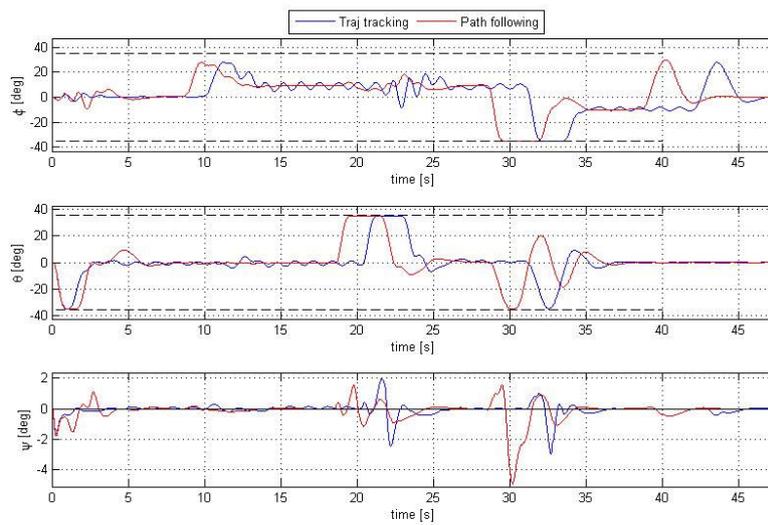


figure 7.4: attitude time plot

8 Conclusion and outlook

In this thesis the latest results achieved at the Bologna University regarding the UniBo RUAS project have been presented. In particular a series of flight control systems based on the model predictive control theory have been illustrated, its performances have been assessed and compared with a baseline linear quadratic regulator.

First, a set of base architectures has been shown and compared each other in order to gain a better understanding of the advantages and disadvantages introduced by each one. As result of this analysis the most promising architectures have been joined together and used to solve the problem of flight control of a small scale helicopter.

The MPC theory has been applied to three different control systems: a SCAS, a trajectory tracking and a path following system. A comparison with a baseline LQR controller has been carried on and, for all the three configurations, the predictive control showed several advantages:

- constraints on both the input and states of the controlled plant can be explicitly introduced in the flight control system formulation and have been used to take into account for control saturations and flight envelope protection,
- the presence of constraints ensures stability of the system also performing high aggressive manoeuvres where the baseline controller fails,

- the predictive approach has the ability of using all the information about the future reference, when this is available, and uses it to enhance the control performances.

On the other hand the computational burden associated with the MPC approach is a lot heavier than the classic control techniques, nevertheless thank to the capabilities of modern processors and new optimization algorithms applied in this thesis real time performances can be achieved by state of the art microcontrollers.

Further development of this work will include the implementation of the proposed architectures in the actual flight control computer and validation trough hardware in the loop simulation and actual flight testing.

Moreover, a further investigation on the coupling between model predictive control and state observers has to be done in order to better asses the effects on control and constraints introduced by state estimation errors.

Bibliography

- [1] R. Pretolani, "*Progetto e realizzazione del sistema di navigazione, guida e controllo per un elicottero con capacità di volo autonomo*", Il School of Engineering, University of Bologna, PhD Thesis, 2007

- [2] B. Teodorani "*Progetto e realizzazione del sistema di gestione autonoma del volo e controllo in remoto per un velivolo UAV ad ala rotante.*" Il School of Engineering, University of Bologna, PhD Thesis, 2007

- [3] F. Zanetti "*Design, implementation and tests of Real-time Feed-forward controller and navigation system for a small scale UAV helicopter*" Il School of Engineering, University of Bologna, PhD Thesis, 2009

- [4] F.Zanetti, S.Colautti, G.M. Saggiani, V.Rossi "*Rotary wing UAV: HIL tests for a model-based FeedForward controller*" 34th ERF meeting proceedings, Liverpool, UK, Sept 2008.

- [5] A. Budiyo, S. Wibowo "*Optimal Tracking Controller Design for a Small Scale Helicopter*", *Journal of Bionic Engineering*, Vol.4, No. 4, pp 271-280, 2007

- [6] L. Marconi, R. Naldi, “*Robust full degree of freedom tracking control of a helicopter*”, *Automatica*, Vol. 43, No. 11, pp 1909-1920, 2007
- [7] K. Peng, G. Cai, B. M. Chen, M Dongb, K. Y. Luma, T. H. Lee “*Design and implementation of an autonomous flight control law for a UAV helicopter*”, *Automatica*, Vol. 45, No. 10, pp 2333-2338, 2009
- [8] Gareth D. Padfield. “*Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling. Second Edition*”, Bramwell Publishing 2007
- [9] R. Cunha, C.Silvestre, “*SimModHeli: A Dynamics Simulator for Model-Scale Helicopters*” The 11th Mediterranean conference on control and automation proceedings, 2003
- [10] B.Mettler, C.Dever and E.Feron ”*Scaling effects and Dynamic Characteristics of Miniature Rotorcraft*” *Journal of Guidance, Control and Dynamics* Vol.27, No. 3, 2004
- [11] D.Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, “*Constrained model predictive control: Stability and optimality*”, *Automatica*, Vol 36, pp.789-814, 2000
- [12] T.Zou, S.Y.Li, B.C.Ding, “*A dual mode nonlinear MPC with enlarged terminal constraint sets*”, *Acta Automatica Sinica*, Vol 32, 2006
- [13] L. Imsland, J. A. Rossiter, B.Pluymsers , J. Suykens, “*Robust Triple Mode MPC*” *American Control Conference Proceedings*, 2006
- [14] Y.I. Lee, B. Kouvaritakisb, M. Cannon “*Constrained receding horizon predictive control for nonlinear systems*” *Automatica* Vol 38, pp. 2093 – 2102, 2002
- [15] D.Limon, I.Alvarardo,T.Alamo, E.F. Camacho “*MPC for tracking constant references for constrained linear systems*” *Automatica* Vol 44, pp. 2382-2387, 2008

- [16] V.Exadaktylos, C.J. Taylor, A. Chotai “*Non-minimal model predictive control with an integral-of-error state variable*” *UKACC International Conference, Glasgow, August 2006*
- [17] U. Maeder, F. Borrelli, M. Morari “*Linear offset-free Model Predictive Control*” *Automatica* Vol 45, pp 2214-2222, 2009
- [18] G. V. Raffo, M. G. Ortega, F. R. Rubio “*An integral predictive/nonlinear Hinf control structure for a quadrotor helicopter*” *Automatica* Vol 46, pp. 29-39, 2010
- [19] S. Oлару, D. Dumur Compact explicit “*MPC with guarantee of feasibility for tracking*” *Proceedings of the 44th IEEE Conference on Decision and Control, Seville 2005*
- [20] E.Frazzoli, M.A.Dahleh, E.Feron “*Trajectory tracking control design for autonomous helicopters using a backstepping algorithm*” *Proceedings of the American Control Conference, Chicago 2000*
- [21] Fabio A. Almeida, “*Waypoint Navigation Using Constrained Infinite Horizon Model Predictive Control*” *AIAA Guidance, Navigation and Control Conference and Exhibit, 2008*
- [22] *Handling qualities requirements for military rotorcrafts*. Technical Report ADS-33DPRF, United States Army, St-Louis Missouri, 1996
- [23] http://www.mpl.ch/DOCs/MPLdoc_00000410.pdf