

POLITECNICO DI TORINO

Collegio di Ingegneria Meccanica, Aerospaziale, dell'Autoveicolo e della  
Produzione

**Corso di Laurea Magistrale in Ingegneria Aerospaziale**

Tesi di Laurea Magistrale

**Model Predictive Control Techniques  
for fixed-wing UAV maneuvers**



**Relatori**

Dr. Elisa Capello

Dr. Martina Mammarella

**Candidato**

Federico Giuffrida Trampetta

Dicembre 2018



# Contents

Chapter 1	Introduction.....	1
1.1	Abstract .....	1
1.2	Motivation and Objectives .....	2
1.3	Overview .....	5
Chapter 2	Unmanned Aerial Vehicle Dynamic Model .....	7
2.1	Reference frames.....	9
2.1.1	The Earth reference frame .....	9
2.1.2	North-East-Down reference frame.....	10
2.1.3	Air Trajectory reference frame .....	11
2.1.4	Body reference frame.....	11
2.2	Nonlinear Model .....	13
2.3	Equations of Motion Linearization .....	14
2.3.1	Longitudinal plane .....	15
2.3.2	Lateral-directional plane .....	16
2.4	Stability problem.....	16
2.4.1	Equilibrium solution and stability.....	17
2.4.2	Lyapunov stability direct method .....	18
2.4.3	Stability of LTI system .....	21
2.4.4	Control of LTI system.....	21
Chapter 3	Model Predictive Control.....	24
3.1	Linear MPC without constraints .....	26
3.2	Linear MPC with constraints .....	28
3.2.1	Input saturation constraints.....	28
3.2.2	State constraints .....	29
3.2.3	Quadratic Programming Solution .....	30
3.3	Stability analysis for linear MPC .....	30

Chapter 4	Tube-Based Robust Model Predictive Control.....	34
4.1	Overview .....	34
4.2	Problem statement and control strategy .....	37
4.2.1	Optimal control problem statement .....	38
4.2.2	Stability analysis .....	39
4.2.3	Overall TRMPC algorithm .....	40
Chapter 5	Simulation Results .....	42
5.1	MATLAB/Simulink Model description.....	43
5.1.1	Initial Condition Evaluation.....	43
5.1.2	Simulink Model .....	44
5.2	Model Predictive Control simulation results.....	46
5.2.1	Diamond path.....	47
5.2.2	Octagonal path .....	52
5.2.3	“Snake” path .....	57
5.3	Tube-based Model Predictive Control simulation results.....	61
5.3.1	Diamond path.....	63
5.3.2	Octagonal path .....	68
5.3.3	“Snake” path .....	73
Chapter 6	Conclusions.....	78
6.1	Conclusion .....	78
6.2	Possible future works .....	79
Chapter 7	References.....	80

## List of Figures

Figure 2.1 - Mini-UAV MH850.....	7
Figure 2.2 - Autopilot board .....	8
Figure 2.3 - HIL connections .....	9
Figure 2.4 - Scheme of Earth reference frame .....	10
Figure 2.5 - Scheme of NED reference frame .....	10
Figure 2.6 - Trajectory representation and wind axes.....	11
Figure 2.7 - Angle definition in Body axes.....	12
Figure 2.8 - Asymptotic stability for a two degree of freedom pendulum.....	18
Figure 2.9 - Open-loop control system .....	22
Figure 2.10 - Closed-loop control system.....	22
Figure 3.1 - Predicted state and control throughout the prediction horizon [17]	24
Figure 4.1 - Outer-bounding tube representation at the k-th time step over a prediction horizon of N.....	35
Figure 5.1 - Scheme of the controller proposed in this work.....	42
Figure 5.2 - Flow chart of initial condition determination.....	44
Figure 5.3 - Simulink model of the system.....	45
Figure 5.4 - Ideal trajectory diamond path.....	47
Figure 5.5 - Longitudinal plane inputs time history.....	48
Figure 5.6 – Longitudinal airspeed and altitude time history .....	48
Figure 5.7 - Longitudinal state variables time history .....	49
Figure 5.8 - Aileron time history .....	50
Figure 5.9 - Heading and roll angle time history .....	50
Figure 5.10 - Lateral-directional plane variable states time history .....	51
Figure 5.11 - Diamond path .....	52
Figure 5.12 - UAV 3D trajectory .....	52
Figure 5.13 - Octagonal ideal path.....	53
Figure 5.14 - Longitudinal inputs .....	53
Figure 5.15 - Longitudinal airspeed and altitude response .....	54
Figure 5.16 - Longitudinal states time history .....	54
Figure 5.17 - Aileron time history .....	55

Figure 5.18 - heading and roll angle time history .....	55
Figure 5.19 - lateral-directional states time history .....	56
Figure 5.20 - Octagonal two-dimensional trajectory .....	56
Figure 5.21 - Octagonal 3D trajectory .....	57
Figure 5.22 - Ideal “snake path” .....	57
Figure 5.23 - Longitudinal inputs time evolution .....	58
Figure 5.24 - Longitudinal airspeed and altitude time history .....	58
Figure 5.25 - Longitudinal states time history .....	59
Figure 5.26 - Aileron time history .....	59
Figure 5.27 - Heading and yaw angle time history .....	60
Figure 5.28 - Lateral-directional states time history .....	60
Figure 5.29 - Two-dimensional “Snake” trajectory .....	61
Figure 5.30 - 3D “snake” trajectory .....	61
Figure 5.31 - Tube-based MPC scheme .....	62
Figure 5.32 - Longitudinal inputs .....	64
Figure 5.33 - Throttle evolution during the first turn .....	64
Figure 5.34 - Longitudinal airspeed and altitude .....	65
Figure 5.35 - Longitudinal states evolution .....	65
Figure 5.36 - Aileron time history .....	66
Figure 5.37 - Heading and yaw angle evolution .....	66
Figure 5.38: Lateral-directional states evolution .....	67
Figure 5.39 - Two-dimensional trajectory .....	67
Figure 5.40 - 3D trajectory .....	68
Figure 5.41 - Longitudinal inputs .....	68
Figure 5.42 - Throttle evolution detail during a maneuver .....	69
Figure 5.43 - Longitudinal airspeed and altitude time history .....	69
Figure 5.44 - Longitudinal states evolution .....	70
Figure 5.45 - Aileron time history .....	70
Figure 5.46 - Heading and yaw angle evolution .....	71
Figure 5.47 - Lateral-directional states evolution .....	71
Figure 5.48 - Two-dimensional disturbed octagonal trajectory .....	72
Figure 5.49 - Three-dimensional octagonal disturbed trajectory .....	72
Figure 5.50 - Longitudinal inputs .....	73
Figure 5.51 - Throttle evolution detail during a maneuver .....	73

Figure 5.52 - Longitudinal airspeed and altitude evolution .....	74
Figure 5.53 - Longitudinal states time history .....	74
Figure 5.54 - Aileron evolution.....	75
Figure 5.55 - Heading and yaw angle evolution .....	75
Figure 5.56 - Lateral-directional state evolution.....	76
Figure 5.57 - “Snake” disturbed two-dimensional trajectory .....	76
Figure 5.58 - Three-dimensional disturbed “snake” trajectory.....	77



# Chapter 1: Introduction

## 1.1 Abstract

Unmanned Aerial Vehicles (UAVs) can accomplish many tasks as surveillance or search and rescue tasks and have many operating advantages compared to manned aircraft. UAVs' flight performance is affected by exogenous disturbances and additive noise, existing in a real operative environment. Dealing with always more demanding requirements of flight maneuvers, a robust Model Predictive Control (MPC) approach is proposed, which is able to handle external disturbances (as gusts or wind disturbances) and parametric uncertainties (as variations in mass, flight conditions or payload).

In this work, it is first considered a classical MPC design [1] [2]. A cost function and state and control constraints are built for the inner loop dynamics (pitch, roll and airspeed) and for the altitude outer loop. A PID control regulates the heading variation (navigation outer loop). MPC law is based on an optimization problem, which at each sampling time aims to find the optimal control sequence that minimizes some function and satisfies some constraints; only the first element of the predicted optimal control sequence is applied to the plant. To obtain a feasible problem it is necessary to choose accurately tuning parameters.

An interesting variation of classical MPC is the Tube-based Robust MPC (TBMPC) [1], which lets to deal with external bounded disturbances and parametric uncertainties with the same computational efficiency of a classical MPC and guarantees to respect hard constraints. In the TBMPC, a linear nominal system is taken into account and it represents a nominal dynamic of the system with no disturbances and uncertainties. The discrepancy between nominal and actual system lets to define the error dynamics. The TBMPC algorithm consists of an offline part and an online part. In the offline part a feedback gain matrix is evaluated in order to stabilize error dynamics and to define tightener constraint sets for nominal states and inputs. The advantages of using this kind of controller are twofold: (i) low computational effort, so this controller can be implemented on an on-board controller, (ii) guarantee of

robustness of the control system, able to handle variations of the system and to represent a realistic environment.

The key feature of this proposed approach is the real-time implementability, with a time-varying control law and, as said before, a feedback gain evaluated offline. Moreover, tightened state and control constraints are computed. In the online part a classical MPC optimization problem is solved at each sampling time, a nominal input is derived and then corrected according to the gain feedback matrix and the actual error. A Linear Matrix Inequality (LMI) approach is applied to the state feedback stabilization, to reduce the computational effort, guaranteeing the stability and improving real-time implementability.

## **1.2 Motivation and Objectives**

With Unmanned Aerial Vehicle we refer to an aircraft, which can fly autonomously or be piloted remotely without a human pilot on board [3]. The most common configuration for an UAV is the aerodyne, which generates dynamically aerodynamic forces to provide lift through rotary wings, for example quadrotor configuration, or fixed wings.

UAV operates as flying platform in a bigger system, called Unmanned Aerial System (UAS), which consists mainly of Unmanned Aircraft (UA), which includes Guidance, Navigation and Control systems, take-off and landing device and payloads, Ground Control Station (GCS), which is the human-machine interface that allows to the operator to overview the UA behavior, and Data Link, which provides communications between UA and GCS.

In this work a focus on Unmanned Aircraft and especially on the UA Control System is given. The main advantage of UAVs is that they can perform all those missions, which could be dangerous for humans. They can accomplish the so-called “Dull, Dirty and Dangerous” missions, so those missions which require long time, such as surveillance or monitoring operations, or which take place in hazardous environment for the crew’s health or which are unsafe for the pilot’s life.

UAVs have a wide variety of possible applications, such as goods delivery [4], terrain mapping [5] and building inspection [6]. For these reasons, UAVs development is nowadays of interest. In order to accomplish the required task, it is necessary that UAV must track precisely the desired trajectory and be stable. These key features can

be achieved through a good control system design, which has to take into account that additive noise in the realistic environment may affect the flight performance.

Proportional Integrative Derivative controllers (PIDs) are adopted in the most commercial autopilots, as it is explained in [7]. PID controllers are computationally efficient, easy to understand, to tune and they provide good control, but because they are non-model based, they do not guarantee optimality and cannot handle system uncertainties, external noise and at the same time take constraints into account.

Indeed, if some changes in the UAV dynamics occurs or the UAV operates in presence of ground effect or aerodynamic perturbations, the PID parameters needs to be tuned again. In [8] a robust PID controller is proposed for uncertain systems, but it does not deal with state and control constraints.

In the recent past, model-based optimal methodology has found many practical applications for multivariable control problems. In practical applications there are many problematics related to a proper dynamic response and to the presence of disturbances in the operational environment. Moreover, it is often required that the system respects defined states and/or inputs constraints during its dynamic evolution. The computational complexity of solving an optimization problem online has always been the main problem, which has limited the application field only to slow and simple dynamic systems.

Nowadays, powerful and efficient processors are available in the market and let to use these methodologies also in real-time applications. In a theoretically way, two Robust Model Predictive Control approaches are presented in [9], where the importance of robustness is discussed, but the real-time application to an UAV system is only marginally considered.

In [10] a switching Model Predictive Control for an unmanned quadrotor helicopter flying in presence of bounded disturbances, but constraints are not taken into account.

In [11] different methods are proposed for MPC real-time implementation for UAVs, but only for trajectory tracking, which is usually characterized by a slow dynamic.

A possible MPC strategy, which lets to obtain a robust MPC, consists of computing a trajectory of an undisturbed system by a classical MPC and applying an additional control law to maintain the perturbed system trajectory within a “tube”,

whose center coincides with disturbance free system [12]. This technique requires a priori knowledge of the bounded disturbances.

Therefore, the aim of this work is to employ a Model Predictive Control law [1] [2] to a fixed-wing Unmanned Aerial Vehicle, focusing on the real-time implementability of MPC-based controllers for inner loop dynamics, which is usually fast, and then to upgrade this control with a Tube-Based Robust Model Predictive Control [1], which lets to handle uncertainties and external disturbances.

In this work it is designed a Tube-based MPC for the inner loop on roll and pitch attitude and for the outer loop on altitude for trajectory, including atmospheric disturbances (i.e. additive noise) and model uncertainties (variations on speed  $V$  and mass  $m$ ).

This approach maintains the same computational efficiency of a conventional Model Predictive Control and at the same time guarantees robustness to disturbances and respect of constraints. To this end, the Tube-based Model Predictive Control is based on two systems: the nominal one, which is a linear undisturbed system and whose behavior is the center of the “tube”, and the disturbed one, which is subjected to uncertainties and additive disturbance and whose behavior is ensured to be within the “tube”.

Its algorithm, as discussed in [1], is split in two parts: the offline part, in which a feedback gain matrix is evaluated, taking model uncertainties into account, and constraint sets are defined, in order to guarantee that uncertain system response will be within a “tube” around the nominal system, and the online algorithm, which consists of controlling the nominal system through a classical MPC scheme and correcting the disturbed system input in a proper way. The outer loop on heading for trajectory is controlled by a PID.

The mini-UAV MH850 [13] has been chosen for testing this advanced control technique and the developed Tube-based MPC should be implemented in a custom-made autopilot, produced in the Department of Mechanical and Aerospace Engineering of Politecnico di Torino [14]. The guidance algorithm adopted in this work is described in detail in [15]. In order to simulate the UAV dynamic behavior in Simulink, a mathematical model has been built, comprehensive of nine first order differential equations, as described in [16].

### 1.3 Overview

The objective of this work is to verify the real-time implementability of the proposed controller, to implement it onboard. In the remaining part of the first chapter an introduction in MPC and Tube-Based MPC will be done. An example of “Receding Control Horizon” procedure [17] will be presented.

Since the dynamic system in this work is simulated in a virtual world, an UAV [13] mathematical model is described in second chapter. Firstly, useful reference frames, such as North-East-Down, air trajectory and body reference frames, are defined and then relations among variables in different reference frames are presented.

Secondly, a classical aircraft nonlinear model is written through nine first order differential equations [16]. To deal with a linear control problem, the nine differential equations are linearized and two state-space models in continuous time [18] are derived. This procedure lets to decouple equations between longitudinal and lateral-directional planes. These two linear continuous time models are then discretized by a zero-order hold method [19].

The stability problem for a general dynamic system is discussed through the equilibrium point definition and the equilibrium stability [18]. The Lyapunov direct method is proposed to draw conclusions about stability of an equilibrium point [18]. This method conducts often to deal with Linear Matrix Inequality problem [20] and some “tricks” are suggested to rewrite an apparently nonlinear matrix inequality as a LMI [21]. Finally, these concepts are applied to a linear time independent state-space model.

In the third chapter a deeper description of Model Predictive Control is presented [2] and a general formulation of optimization problem is given [1]. To understand how to explicit the cost function dependency on the current state of the system and on the predicted optimal control sequence, a problem without constraints is encountered and a method to derive the unknown optimal control sequence is proposed.

States and inputs constraints are then introduced and a procedure to write them as inequalities referred to the unknown optimal control sequence is explained. Some considerations are taken into account building the optimization problem cost function, in order to obtain a convex quadratic problem [22].

For the stability analysis a Lyapunov direct method, after some important definitions about positively invariant sets are given [23], is considered. Choosing

different terminal weighting matrix and terminal constraint set, stability is guaranteed.

Three methods are proposed:

1. “Zero terminal constraints” [24], which imposes that terminal state constraints are null.
2. “Terminal weighting matrix” [25], which considers nonnull terminal constraint sets, but terminal weighting matrix is derived by discrete time Lyapunov equation.
3. “Invariant terminal set” [1], which considers a positive invariant set containing the origin and the terminal weighting matrix calculated as solution of the Algebraic Riccati equation.

In the fourth chapter the problem of handling uncertainties and external disturbances is addressed and an introduction in robust controller is provided. An extension of conventional Model-Predictive Control is proposed: Tube-Based Robust Model Predictive Control [1].

Firstly, it is presented the offline algorithm, which ensures stability controlling the dynamic system. To do that some important definitions about robust positively invariant set [23], minimal robust positively invariant set [26] and about set operations, such as Minkovski set addition [27] and the Pontragyn set difference [28]. To evaluate the gain feedback matrix, an LMI problem is solved using `MATLAB` in combination with the optimization toolbox `YALMIP` [29].

In the fifth chapter the `MATLAB/Simulink` model is described with more focus. A general description of initial condition evaluation is given and the main boxes in `Simulink` model are presented. This model has been then tested through different waypoints and paths. Three possible paths are considered and each can be considered a possible flight mission.

## Chapter 2: Unmanned Aerial Vehicle Dynamic Model

In this chapter, as already explained in Introduction, the mathematical model of the considered system and the reference frames are described. The mini-UAV considered in this study is the MH850 (see Figure 2.1). It is a fixed-wing aircraft with tailless configuration, electric propulsion and without rudder.

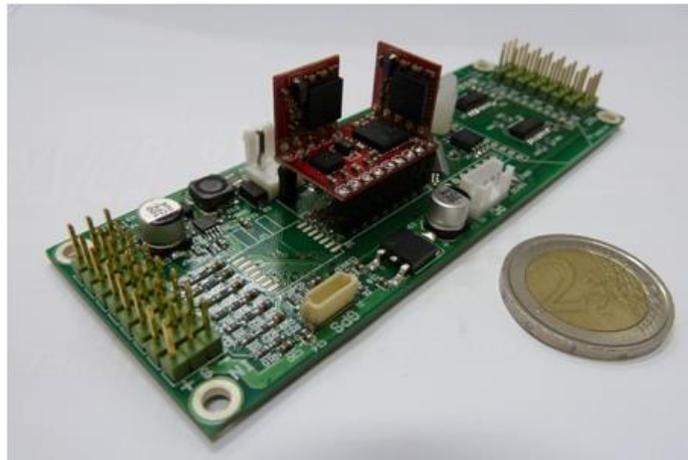


Figure 2.1 - Mini-UAV MH850

Technical features can be seen in the following list collected by [13]:

- Wingspan: 850 *mm*;
- Weight: 1000 *g*;
- Payload (excl. batteries): 100*g*;
- Airspeed: 7.5 ÷ 20 *m/s*;
- Range: 250 ÷ 5000 *m*;
- Endurance: 45 *min* @ 13.5 *m/s* .

The Model Predictive Control design is based on a mathematical model of the plant. A complete nonlinear model, as defined in [16], is a set of nine equations describing the forces, moments, angles and angular speeds which characterize the flight condition of the aircraft. The MH850 aircraft is able to perform autonomous flight thanks to the on-board installation of an autopilot. In this design, a custom-made autopilot, produced in the Department of Mechanical and Aerospace Engineering of Politecnico di Torino [14] (see Figure 2.2), is considered.



**Figure 2.2** - Autopilot board

The key aspects of this autopilot are an open architecture and the possibility to be reprogrammed in flight and real time telemetry. Sensors include GPS, barometric sensor, differential pressure sensor and three-axis gyros and accelerometers. The CPU is the ATXMEGA256A3U-3U model with 256Kb flash memory and 16Kb of RAM. A Radiomodem Xbee Pro S1 is used for the communication link between the Ground Control Station (GCS) and the autopilot. To validate the real-time effectiveness of the proposed controller in HIL simulations, hardware constraints are included. The hardware constraints are related to a commercial board (the XMOS XK-1A board) because of its characteristics and potentialities (i.e. flash memory of 128 Kb and a CPU clock of 20 MHz) similar the microcontroller, that is installed on the MH850. The XMOS XK-1A is a low-cost development board produced by XMOS Ltd ([www.xmos.com](http://www.xmos.com)), and it is characterized by the multi-core multi-thread processor XS1-L1 which is able to perform several real-time tasks. Its parallel computing ability is essential for unmanned applications where high-level tasks (for instance the control logic) have to be combined with low level assignments (such as I/O). A detail of the HIL cables connection and of the board is represented in Figure 2.3.

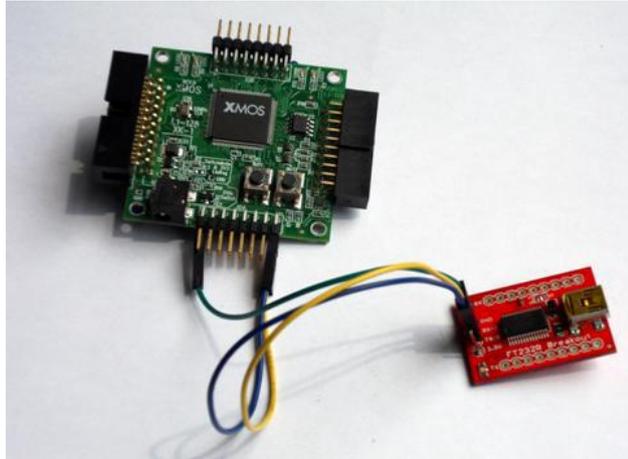


Figure 2.3 - HIL connections

In this chapter assumptions and reference frames to build the model are described.

## 2.1 Reference frames

The purpose of this section is to define the typical coordinate frames used to describe UAV model motion. Every reference frame has its origin and a set of three orthogonal axis which create a right-hand system. In this section four reference frames are described:

- Earth reference frame.
- North-East-Down reference frame.
- Air Trajectory reference frame.
- Body reference frame.

### 2.1.1 The Earth reference frame

The origin of this reference frame is at the center of the Earth and the axes can be defined in the following way:

- $Z_E$  axis has South-North direction.
- $X_E$  and  $Z_E$  lie on the equatorial plane.
- Greenwich Meridian belongs to the plane defined by  $X_E$  and  $Z_E$ .

This reference frame is not fixed but rotates with an angular speed equal to  $\Omega_E$ . Since Earth spins around its rotation axis and rotate around the Sun, it follows that Earth reference frame is not inertial. In this case, the assumption of flat and non-rotating earth is made. This assumption is realistic for UAV applications. A representation of this reference frame is shown in Figure 2.4.

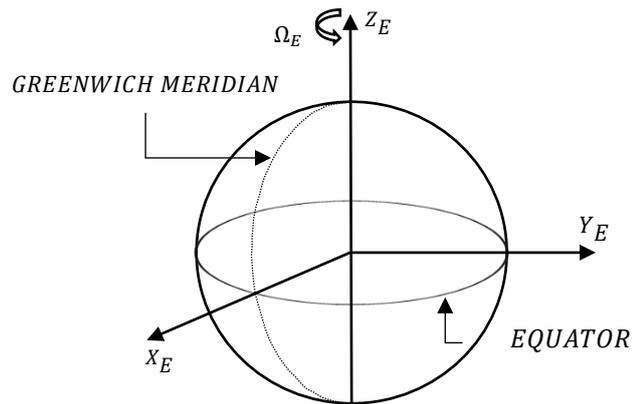


Figure 2.4 - Scheme of Earth reference frame

### 2.1.2 North-East-Down reference frame

It is also called NED reference frame and its origin coincide with the aircraft center of mass. The axes have the following properties:

- $Z_V$  axis direction is along the local gravity accelerator vector;
- $X_V$  and  $Y_V$  axes lie in a plane parallel to the Earth tangent plane and the distance between the two planes is equal to the flight altitude.

This reference frame is adopted in guidance and navigation algorithm. The vehicle trajectory  $[x \ y \ h]^T$  is derived by the integration of each component of the total airspeed  $V$  along NED axes ( $[V_N \ V_E \ V_D]^T$ ). A representation of this reference frame is shown in Figure 2.5.

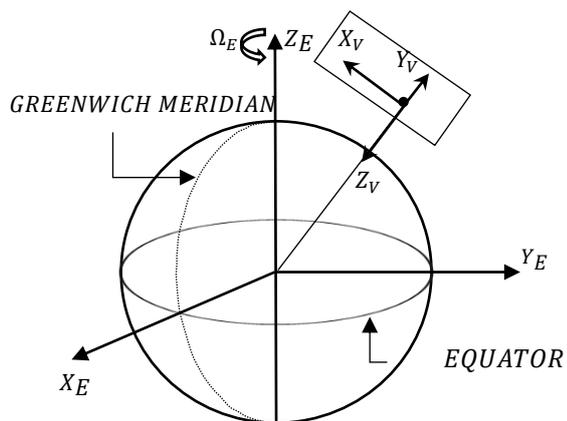


Figure 2.5 - Scheme of NED reference frame

### 2.1.3 Air Trajectory reference frame

The Air Trajectory axes are also said wind axes and their origin is the aircraft center of mass. The main properties are:

- $X_W$  axis has the same direction of the airspeed  $V$
- $Z_W$  axis lies in the vehicle plane of symmetry and its direction is from upper to lower surface of wing airfoil.

This reference frame is not used in this study (see Figure 2.6).

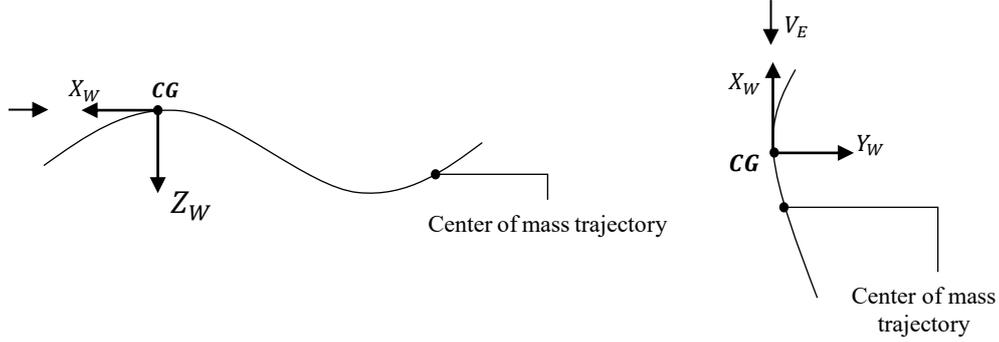


Figure 2.6 - Trajectory representation and wind axes

### 2.1.4 Body reference frame

Any set of axes, whose origin is fixed in the rigid body, are named Body Axes. The origin is usually located at the aircraft center of mass. Generally,  $X_B$  and  $Z_B$  axis lie in the aircraft plane of symmetry,  $Z_B$  is directed from upper to lower surface of wing airfoil. We can also describe the following:

- Principal axes of inertia: body axes are directed along principal axes of inertia. The main advantage frame is that the “mixed” inertial terms  $J_{xz}$ ,  $J_{xy}$  and  $J_{yz}$  are null with respect to principal axes of inertia ( $J_{xz} = J_{xy} = J_{yz} = 0$ ).
- Stability axes:  $X_B$  axes direction coincides with the projection of  $V$  in the plane symmetry at the starting condition.

The vector  $[u \ v \ w]^T$  has as element the components of total airspeed  $V$  along respectively  $X_B$ ,  $Y_B$  and  $Z_B$ . The relation between velocity components in NED reference frame and velocity components in body reference frame is the following:

$$\begin{aligned}
 V_N &= u \cos \theta \sin \phi + v(\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) \\
 &\quad + w(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\
 V_E &= u \cos \theta \sin \psi + v(\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) \\
 &\quad + w(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)
 \end{aligned} \tag{1}$$

$$V_D = u \sin \theta + v \cos \theta \sin \phi + w \cos \phi \cos \theta$$

where  $[\phi \ \theta \ \psi]^T$  are the Euler angles, which represent the vehicle orientation, i.e. the body axes orientation, with respect to the NED reference system, and they are respectively roll angle, pitch angle and yaw angle.

In this section it is also important to define aerodynamic angles such as angle of attack  $\alpha$  in longitudinal plane and sideslip angle  $\beta$  in lateral-directional plane. The angle of attack  $\alpha$  is the angle between null lift axis ( $c_L = 0$ ) and airspeed direction. If  $X_B$  axis is parallel to the null lift axis, then the angle of attack  $\alpha$  is also the angle between  $X_B$  axis and airspeed direction. Otherwise a constant angle is present between  $X_B$  axis and null lift axis and the angle between airspeed direction and  $X_B$  axis will be equal to the sum of angle of attack and the constant angle. Sideslip angle  $\beta$  represent the angular deviation of airspeed with respect to  $X_B$  axis in lateral-directional plane. The angle between total airspeed  $V$  and horizontal line is the slope angle and its name is  $\gamma$ . In Figure 2.7 aerodynamic angles are shown.

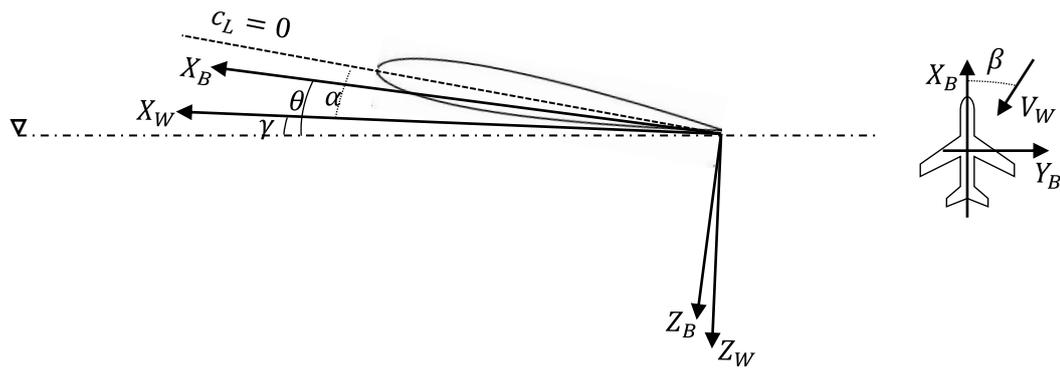


Figure 2.7 - Angle definition in Body axes

Assuming that  $X_B$  axis has the same direction of null lift axis, it is possible to decompose total airspeed  $V$  along the three body axes as follows:

$$\begin{aligned} u &= V \cos \alpha \cos \beta \\ v &= V \sin \beta \\ w &= V \sin \alpha \cos \beta \end{aligned} \tag{2}$$

## 2.2 Nonlinear Model

The model, which describes the aircraft's dynamic, consists of nine non-linear equations [16]. These equations are formulated in Body Reference Frame, whose origin is the aircraft center of mass.

The body is supposed to be rigid and the Earth flat and non-rotating. These assumptions meet Mini-UAV's operating conditions. The UAV's non-linear mathematical model is made of 3 force equations, 3 moment equations and 3 kinematic equations.

The force equations are:

$$\begin{aligned}\dot{u} &= \frac{F_X}{m} + qw - rv + g \cdot \sin \theta \\ \dot{v} &= \frac{F_Y}{m} - pw + ru - g \cdot \cos \theta \cdot \sin \phi, \\ \dot{w} &= \frac{F_Z}{m} + pv - qu - g \cdot \cos \theta \cdot \sin \phi\end{aligned}\quad (3)$$

where  $m$  is the aircraft mass, considered constant,  $[u \ v \ w]^T$  are the components of the speed  $V$  along respectively the axes  $X_B$ ,  $Y_B$  and  $Z_B$ ,  $[p \ q \ r]^T$  are the angular speeds and  $[F_X \ F_Y \ F_Z]^T$  the components of the total force acting on the aircraft. These forces are evaluated at each sampling time. The force components along  $X_B$  and along  $Z_B$  axis depends on attack angle  $\alpha$ , nondimensional pitch rate  $\hat{q}$ , elevon deflection  $\delta_e$  and throttle position  $\delta_{th}$  ( $F_X = f_x(\alpha, \hat{q}, \delta_e, \delta_{th})$ ,  $F_Z = f_z(\alpha, \hat{q}, \delta_e, \delta_{th})$ ). The force component along  $Y_B$  is related with sideslip angle  $\beta$ , nondimensional roll rate  $\hat{p}$  and nondimensional yaw rate  $\hat{r}$  ( $F_Y = f_y(\beta, \hat{p}, \hat{r})$ ). The moment equations are:

$$\begin{aligned}\dot{p} &= \frac{L}{J_x} + \frac{[J_{xz}(\dot{r} + pq) + qr(J_y - J_z)]}{J_x} \\ \dot{q} &= \frac{M}{J_y} + \frac{[J_{xz}(r^2 + p^2) + pr(J_z - J_x)]}{J_y}, \\ \dot{r} &= \frac{N}{J_z} + \frac{[J_{xz}(\dot{p} - pq) + pq(J_x - J_y)]}{J_z}\end{aligned}\quad (4)$$

where  $J_i$  is the moment of inertia about  $i$ -axis with  $i = x, y, z, xz$ . The inertia terms  $J_{xy}$  and  $J_{yz}$  are null because, vehicle mass distribution is symmetrical about  $X_B Z_B$  plane. The vector  $[L M N]^T$  represents the roll, pitch and yaw moment and is built at each sampling time. The roll and yaw moments depend on sideslip angle  $\beta$ , nondimensional roll rate  $\hat{p}$ , nondimensional yaw rate  $\hat{r}$  and aileron deflection  $\delta_a$  ( $L = f_l(\beta, \hat{p}, \hat{r}, \delta_a)$ ,  $N = f_n(\beta, \hat{p}, \hat{r}, \delta_a)$ ). The pitch moment is function of attack angle  $\alpha$ , nondimensional pitch rate  $\hat{q}$  and elevon deflection  $\delta_e$  ( $M = f_m(\alpha, \hat{q}, \delta_e)$ ). The forces and the moments in the 6 degree of freedom equation of motions are derived through aerodynamic derivatives included in a database. The aircraft attitude is described by the Euler angles  $[\phi \theta \psi]^T$ , whose variation is defined by the kinematic equations:

$$\begin{aligned}\dot{\phi} &= p + q \cdot \sin \phi \cdot \tan \theta + r \cdot \cos \phi \cdot \tan \theta \\ \dot{\theta} &= q \cdot \cos \phi - r \cdot \sin \phi \\ \dot{\psi} &= q \cdot \frac{\sin \phi}{\cos \theta} + r \cdot \frac{\cos \phi}{\cos \theta}\end{aligned}\tag{5}$$

These 9 equations are coupled and nonlinear and that makes difficult to treat the control topic. To make the problem more manageable a linearization under certain assumption is done.

### 2.3 Equations of Motion Linearization

In order to decouple longitudinal and lateral-directional planes, the equations of motion linearization is applied. The reference flight condition is an airspeed of  $13.5 \text{ m/s}$  and at an altitude of  $100 \text{ m}$ . In these conditions the trim angle of attack is  $\alpha_0 = 6.12^\circ$  and, assuming a horizontal flight, the trim pitch angle  $\theta_0 = 6.12^\circ$ . From linearization is obtained a state-space model for longitudinal and lateral-directional planes as the following:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}\tag{6}$$

This is a classical linear state-space model in continuous time [18], where the system matrix  $A \in \mathbb{R}^{n,n}$ , the input matrix  $B \in \mathbb{R}^{n,m}$ , the output matrix  $C \in \mathbb{R}^{p,n}$  and

matrix  $D \in \mathbb{R}^{p,m}$ , with  $n$  the number of state variables,  $m$  the number of inputs and  $p$  the number of outputs. In this case the  $C$  matrix is always an identity, the matrix  $D$  is equal to zero, because it is assumed that there is no relation between output and input, and the number of inputs is the same of the number of outputs ( $n = m$ ). This state-space model is said time independent, because the matrices  $A$ ,  $B$ ,  $C$  and  $D$  do not vary in the time. Indeed, in this study only one equilibrium condition is evaluated. If more equilibrium conditions would be considered, new matrices  $A$ ,  $B$ ,  $C$  and  $D$  should be calculated. In some cases, state variables and control variables can be subjected to constraints, so we have  $x \in \mathbb{X}$  and  $u \in \mathbb{U}$ , with the state constraint set  $\mathbb{X} \subset \mathbb{R}^n$  and the control constraint set  $\mathbb{U} \subset \mathbb{R}^m$ .

The discrete-time state-space model is:

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) \\ y(k) = C_d x(k) + D_d u(k) \end{cases}, \quad (7)$$

where  $k \in \mathbb{Z}^+$ . In the following part the matrix for longitudinal and lateral-directional planes are reported. These matrices are built in continuous time and then they are converted to discrete time assuming a zero order hold [19] on the inputs.

### 2.3.1 Longitudinal plane

For longitudinal plane it is added a fifth equation, which is related to the altitude. In this case there are five state variables and two control variables:

$$x = \begin{pmatrix} u \\ \alpha \\ \theta \\ q \\ h \end{pmatrix} \quad u = \begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}$$

where  $\delta_{th}$  is the throttle and  $\delta_e$  is the elevon deflection angle. The state matrix  $A$  is:

$$A = \begin{bmatrix} X_u & X_\alpha & -g \cos(\theta_0) & 0 & 0 \\ \frac{Z_u}{U_0 - Z_\dot{\alpha}} & \frac{Z_\alpha}{U_0 - Z_\dot{\alpha}} & -\frac{g \sin(\theta_0)}{U_0 - Z_\dot{\alpha}} & \frac{Z_q + U_0}{U_0 - Z_\dot{\alpha}} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ M_u + \frac{M_\dot{\alpha} Z_u}{U_0 - Z_\dot{\alpha}} & M_\alpha + \frac{M_\dot{\alpha} Z_\alpha}{U_0 - Z_\dot{\alpha}} & -\frac{g \sin(\theta_0) M_\dot{\alpha}}{U_0 - Z_\dot{\alpha}} & M_q + \frac{M_\dot{\alpha} (Z_q + U_0)}{U_0 - Z_\dot{\alpha}} & 0 \\ 0 & -U_0 & U_0 & 0 & 0 \end{bmatrix}$$

The control matrix  $B$  is:

$$B = \begin{bmatrix} 1 & 0 \\ 0 & \frac{Z_{\delta_e}}{U_0 - Z_{\dot{\alpha}}} \\ 0 & 0 \\ 0 & M_{\delta_e} + \frac{M_{\dot{\alpha}} Z_{\delta_e}}{U_0 - Z_{\dot{\alpha}}} \\ 0 & 0 \end{bmatrix}$$

### 2.3.2 Lateral-directional plane

For lateral-directional plane it is not considered the yaw angle  $\psi$ , because its control is separated by the rest of dynamics and it is accomplished by a PID controller. State and control vectors are:

$$x = \begin{Bmatrix} v \\ p \\ r \\ \phi \end{Bmatrix} \quad u = \{\delta_a\}$$

Where  $\delta_a$  is the aileron deflection angle and it is the unique input in lateral-directional dynamic. The state matrix  $A$  is:

$$A = \begin{bmatrix} Y_v & Y_p & Y_r - U_0 & g \\ L_{v_d} & L_{p_d} & L_{r_d} & 0 \\ N_{v_d} & N_{p_d} & N_{r_d} & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The control matrix  $B$  is:

$$B = \begin{bmatrix} Y_{d_a} \\ L_{d_{ad}} \\ N_{d_{ad}} \\ 0 \end{bmatrix}$$

As can be seen in lateral-directional plane the only control surface is the aileron, because of the tailless configuration.

## 2.4 Stability problem

In this section an overview about stability problem is proposed. First a definition of equilibrium point is given, then the different kinds of stability for equilibrium points are illustrated and finally Lyapunov direct method is described. The reference used in this section is [18].

It is considered the following LTI system described by state space representation in continuous time  $\dot{x}(t) = Ax(t) + Bu(t)$ ,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ .

### 2.4.1 Equilibrium solution and stability

A solution  $x$  of the dynamic system is defined as equilibrium solution  $x_e$ , if in presence of the constant input  $u(t) = u_e$  and the initial condition  $x(0) = x_e$ , it results that the solutions for all following time are equilibrium solutions:  $x(t) = x_e \forall t \geq 0$ .

In this case the input  $u_e$  is said equilibrium input, the output  $y_e$  equilibrium output and the couple  $(x_e, u_e)$  equilibrium point. An equilibrium point  $(x_e, u_e)$  for a generic dynamic system such as  $\dot{x}(t) = f(x(t), u(t))$  satisfies the condition  $f(x_e, u_e) = 0$ . For a linear dynamic system with an invertible state matrix  $A$  the following relation between equilibrium state and equilibrium output is valid:

$$x_e = -A^{-1}Bu_e$$

Once an equilibrium solution  $x_e$  is found, it must be understood if it is stable or not. To do that, it is necessary to introduce a small initial perturbation near the equilibrium solution  $x_e$  and to observe if the system evolves near the equilibrium solution  $x_e$  or not. Supposing that an equilibrium solution of a generic dynamic system  $x_e = \phi(t, t_0, x_e, u_e)$  and the perturbed solution is  $x_p(t) = \phi(t, t_0, x_0, u_e)$ , where  $x_0$  is the perturbed initial condition in some neighborhood of  $x_e$ , and studying the perturbed evolution, it is possible to understand if the equilibrium point is:

- Stable if the perturbed solution evolves in a bounded neighborhood of  $x_e$ :

$$\forall \varepsilon > 0, \quad \exists \delta = \delta(\varepsilon) > 0 \mid \forall x_0: \|x_0 - x_e\| \leq \delta \Rightarrow \|x_p(t) - x_e\| \leq \varepsilon, \\ \forall t \geq 0$$

- Asymptotically stable if the perturbed solution tends to  $x_e$  for  $t \rightarrow \infty$ :

$$\lim_{t \rightarrow \infty} \|x_p(t) - x_e\| = 0$$

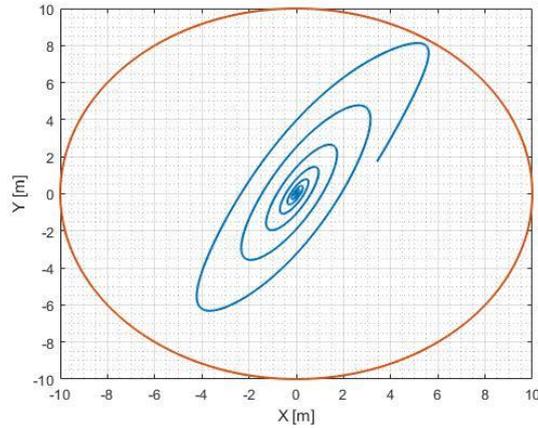
- Unstable if the perturbed solution evolves far away from  $x_e$ .

Considering the perturbed solution  $x_p = \phi(t, t_0, x_0, u_e)$  of a generic dynamic with constant input  $u_e$  and perturbed initial condition  $x_e$ , it is possible to define for asymptotically stable equilibrium point a domain of attraction  $S(x_e, u_e) = \{x_0: \lim_{t \rightarrow \infty} \|\phi(t, t_0, x_0, u_e) - x_e\| = 0\}$ .

Roughly speaking, the domain of attraction of the equilibrium point  $(x_e, u_e)$  is a set of initial condition  $x_0$  in the equilibrium state neighborhood  $x_e$ , for which the

definition of asymptotical stability is valid. Furthermore, if  $S(x_e, u_e) = \mathbb{R}^n$ , the equilibrium point  $(x_e, u_e)$  is globally asymptotically stable.

As example of asymptotical stability in real world, it is likely a two degrees of freedom pendulum. Its dynamic is described by two second order differential equations. If a reference system is centered in the pendulum quiet position, the origin  $(0,0)$  is an equilibrium point. The equilibrium point is asymptotically stable. If any initial position and initial speed is applied to the system within the circle centered in the origin with radius equal to the limb length, the perturbed solution will tend to the equilibrium point  $(0,0)$  for  $t \rightarrow \infty$ . The trajectory of a perturbed solution is shown in Figure 2.8.



**Figure 2.8** - Asymptotic stability for a two degree of freedom pendulum

#### 2.4.2 Lyapunov stability direct method

The Lyapunov direct method provides a sufficient condition for stability. If a function  $V(x)$ , which satisfies some requirements, is found, some conclusions about stability can be done. The main advantage of this method is that conclusion about stability can be drawn without solving the differential state equation.

Lyapunov function should be defined as a once differentiable function  $C^1$  in a domain  $\mathcal{X} \subset \mathbb{R}^n$ :  $V(x): \mathcal{X} \rightarrow \mathbb{R}$  and  $V(x) \in C^1$ .

Something about the stability of the equilibrium point  $x_e$  can be said, if a function  $V(x)$  positive definite or negative definite in  $x_e$  is found.  $V(x)$  is said:

- Positive definite in  $x_e \in \mathcal{X}$  if the following conditions are satisfied:
  - $V(x_e) = 0$
  - It exists a neighborhood  $I_\delta = \{x: \|x - x_e\| \leq \delta\}$  of  $x_e$  such that:

$$V(x) > 0 \forall x \in I_\delta, x \neq x_e$$

- Positive semi-definite in  $x_e \in \mathcal{X}$  if the following conditions are satisfied:
  - $V(x_e) = 0$
  - It exists a neighborhood  $I_\delta = \{x: \|x - x_e\| \leq \delta\}$  of  $x_e$  such that:

$$V(x) \geq 0 \forall x \in I_\delta, x \neq x_e$$

- Negative definite in  $x_e \in \mathcal{X}$  if the following conditions are satisfied:
  - $V(x_e) = 0$
  - It exists a neighborhood  $I_\delta = \{x: \|x - x_e\| \leq \delta\}$  of  $x_e$  such that:

$$V(x) < 0 \forall x \in I_\delta, x \neq x_e$$

- Negative semi-definite in  $x_e \in \mathcal{X}$  if the following conditions are satisfied:
  - $V(x_e) = 0$
  - It exists a neighborhood  $I_\delta = \{x: \|x - x_e\| \leq \delta\}$  of  $x_e$  such that:

$$V(x) \leq 0 \forall x \in I_\delta, x \neq x_e$$

Furthermore, if  $V(x)$  is positive definite function in  $x_e$ , then  $V(x)$  admits local minimum. Similarly, if  $V(x)$  is negative definite function in  $x_e$ , then  $V(x)$  admits local maximum.

Let now consider a generic dynamic system  $\dot{x}(t) = f(x(t), u(t))$  with an equilibrium point in  $x_e$ , which is contained in the domain  $\mathcal{X} \subseteq \mathbb{R}^n$ . If there exists a class  $C^1$  function  $V(x)$ , which is positive definite in  $x_e$  and such that the first derivative  $\dot{V}(x)$  is:

- semi-negative definite in  $x_e \in \mathcal{X}$ , then the equilibrium point  $x_e$  is stable.
- negative definite in  $x_e \in \mathcal{X}$ , then the equilibrium point  $x_e$  is asymptotically stable.
- positive definite in  $x_e \in \mathcal{X}$ , then the equilibrium point  $x_e$  is unstable.

The first derivative  $\dot{V}(x)$  can be calculated as:

$$\dot{V}(x) = \frac{dV(x)}{dt} = \frac{dV(x)}{dx} \frac{dx(t)}{dt} = \frac{dV(x)}{dx} f(x, u_e)$$

As example of this method application, let consider the linear system  $\dot{x} = Ax$ . As Lyapunov function attempt it is used the following one:

$$V(x) = \frac{1}{2} x^T P x$$

Where  $P$  is supposed to be a symmetrical positive definite matrix, so  $V(x)$  is positive definite. It must be studied the first derivative of this function:

$$\dot{V}(x) = \frac{dV(x)}{dx} \frac{dx(t)}{dt} = x^T P \dot{x} = x^T P A x = \frac{1}{2} (x^T P A x + x^T P A x)$$

Since  $x^T P A x = (x^T P A x)^T = x^T A^T P x$ , it follows:

$$\dot{V}(x) = \frac{1}{2} [x^T (P A + A^T P) x]$$

In order to have asymptotic stability this function must be negative ( $\dot{V}(x) < 0$ ). This condition is satisfied if and only if the matrix  $(P A + A^T P)$  is negative definite:

$$P A + A^T P < 0$$

The unknown term is the matrix  $P$ , which has to be determined with respect to the hypotheses. Let now consider a linear system with some inputs  $\dot{x} = A x + B u$ . If a feedback control law such as  $u = K x$  is introduced, then the dynamic system can be rewritten as:

$$\dot{x} = (A + B K) x$$

The feedback matrix  $K$  must be chosen in order to stabilize the system. To do a suitable choice of it, a linear matrix inequality problem [20] can be built as follows:

$$P(A + BK) + (A + BK)^T P < 0$$

Where the unknown matrices are  $P$ , which is symmetric and positive definite, and  $K$ . It is now demonstrated how to make this inequality explicitly bilinear [21]. Because the inverse of  $P$  matrix is also positive definite, it is a valid procedure to multiply from right and from left the inequality with matrix  $P^{-1}$ :

$$\begin{aligned} P^{-1}[(A + BK)^T P + P(A + BK)]P^{-1} &< 0 \\ P^{-1}A^T + P^{-1}K^T B^T + AP^{-1} + BKP^{-1} &< 0 \end{aligned}$$

It is possible now to rename the following matrices as:

$$X = P^{-1}$$

$$Y = KP^{-1} = KX$$

Lyapunov LMI can be rewritten as  $AX + XA^T + BY + Y^T B^T < 0$ , where  $X > 0$ . This inequality is now linear and can be solved with standard methods for LMIs.

For a discrete time, linear time invariant system  $x(k+1) = f(x(k), u(k))$  if there exists an equilibrium point  $x_e$ , contained in a domain  $\mathcal{X} \subseteq \mathbb{R}^n$ , and if there exists a class  $C^1$  function  $V(x(k))$ , which is positive definite in  $x_e \in \mathcal{X}$  such that

$$\Delta V(x) = V(x(k+1)) - V(x(k)) = V(f(x(k), u_e)) - V(x(k)) < 0 \quad (8)$$

in the equilibrium point  $x_e$ , then the equilibrium is asymptotically stable and  $V(x)$  is said Lyapunov function. If the closed loop discrete time, linear time independent

system  $x(k+1) = (A+BK)x(k)$  is considered and the function  $V(x) = \frac{1}{2}x^T Px$  is tried as Lyapunov function, then inequality (8) can be rewritten as:

$$\begin{aligned}\Delta V(x) &= \frac{1}{2}[x(k+1)^T Px(k+1) - x(k)^T Px(k)] \\ &= \frac{1}{2}x(k)^T [(A+BK)^T P(A+BK) - P]x(k) < 0\end{aligned}$$

Therefore, if a matrix  $K$ , which makes the equilibrium point asymptotically stable, is searched, the following matrix inequality has to be satisfied:

$$(A+BK)^T P(A+BK) - P < 0$$

### 2.4.3 Stability of LTI system

It is now considered a linear time independent system such as (6). The eigenvalues of state matrix  $A$  are denoted with  $\lambda_i(A)$ ,  $i = 1, \dots, n$ . The LTI system is:

- *Asymptotically stable* if and only if:
  - $\text{Re}(\lambda_i(A)) < 0, i = 1, \dots, n$
- *Internally stable* if and only if:
  - $\text{Re}(\lambda_i(A)) \leq 0, i = 1, \dots, n$  and  $\mu'(\lambda_j(A)) = 1$  with  $\text{Re}(\lambda_j(A)) = 0$ <sup>1</sup>
- *Unstable* if and only if:
  - $\exists i: \text{Re}(\lambda_i(A)) > 0$  or
  - $\text{Re}(\lambda_i(A)) \leq 0, i = 1, \dots, n$  and  $\mu'(\lambda_j(A)) > 1$  with  $\text{Re}(\lambda_j(A)) = 0$

A discrete time linear system, such as (7), is said *asymptotically stable* if and only if all the eigenvalues of the state matrix  $A_d$  lie strictly inside the unit circle in the discrete complex plane.

### 2.4.4 Control of LTI system

Control systems can be divided in two categories: open-loop and closed-loop systems [30]. In the first one the control action does not depend on the system output. This kind of control is applied to those systems whose response is well known and is

---

<sup>1</sup>  $\mu'(\cdot)$  is the geometric multiplicity.

not subjected to stability problems. In Figure 2.9 an example of this configuration is given.

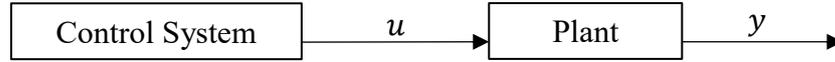


Figure 2.9 - Open-loop control system

In a closed-loop control system, or also called feedback control system, the control action depends on the system output. A feedback is introduced, in order to permit the output to be compared with the input to the system. This kind of control is used when the controlled system has a response, which is not well known, is unstable and/or operates in presence of disturbance or uncertainties. In Figure 2.10 the scheme of a closed-loop control is represented.

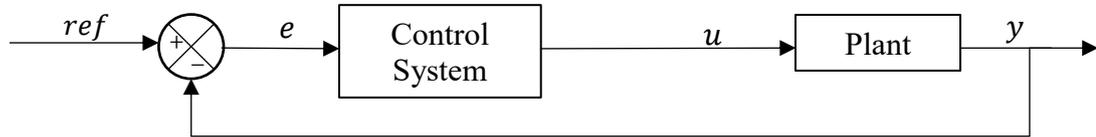


Figure 2.10 - Closed-loop control system

Let now consider a system written in state space formulation such as (6). The eigenvalues of the state matrix  $A$  determine the dynamic behavior of the system. If a stable system has to be controlled, an open-loop configuration can be adopted and the response would be acceptable. Otherwise, if an unstable system has to be controlled, it is possible to modify the eigenvalues of matrix  $A$  by considering an input which depends on the state variables:

$$u(t) = -Kx(t)$$

This means to introduce a feedback in the control and to choose a closed-loop control. The LTI system can be written as:

$$\dot{x}(t) = Ax(t) + Bu(t) = (A - BK)x(t)$$

The stability of LTI system depends now on the eigenvalues of the matrix  $(A - BK)$ . The wanted answer of the system can be achieved hereby by a suitable choice of the matrix  $K$ . An arbitrary assignment of the eigenvalues of the matrix  $(A - BK)$  through the choice of the matrix  $K \in \mathbb{R}^{m,n}$  is possible if and only if:

$$\rho(M_R) = \rho([B \ AB \ \dots \ A^{n-1}B]) = n$$

Where  $\rho(\cdot)$  returns the rank of argument matrix and  $M_R$  is the controllability matrix. If  $\rho(M_R) = n$  the couple  $(A, B)$  is said controllable.

In the further part of this work, it will be always considered a closed-loop configuration, because the UAV system, as it has been described previously, operates in presence of disturbance and uncertainties and it is unstable.

## Chapter 3: Model Predictive Control

Model Predictive Control is basically a method to control dynamic system solving an optimization problem. At each sampling time a finite horizon optimal control problem, where the actual state is the initial state, is solved on-line and an optimal control sequence is derived. Only the first element of this latter sequence is applied to the plant. The prediction horizon is then shifted and the problem is again solved with the new state of the system as initial state. Hence the optimization problem solution is a function of the current state of the system. This procedure is also known as “Receding Control Horizon”.

In Figure 3.1 it is drawn an example of this method inspired by [17]. At starting instant  $t$  the optimization problem is solved with initial state  $x(t)$  until the final instant  $t + N_p$  (blue curve); in this horizon it is applied to the system the optimal control sequence  $\mathbf{u}^*(t) = \{u^*(t), u^*(t + 1), \dots, u^*(t + N_p)\}$ , drawn in orange. The first element of this latter sequence  $u^*(t)$  is applied to the plant until the next sampling time and a new state is reached by the system. At the next sampling time, the new state become the current one, the prediction horizon is shifted and the starting instant is updated to be  $t + 1$ . Now the last instant in prediction horizon is  $t + N_p + 1$ . The procedure is repeated.

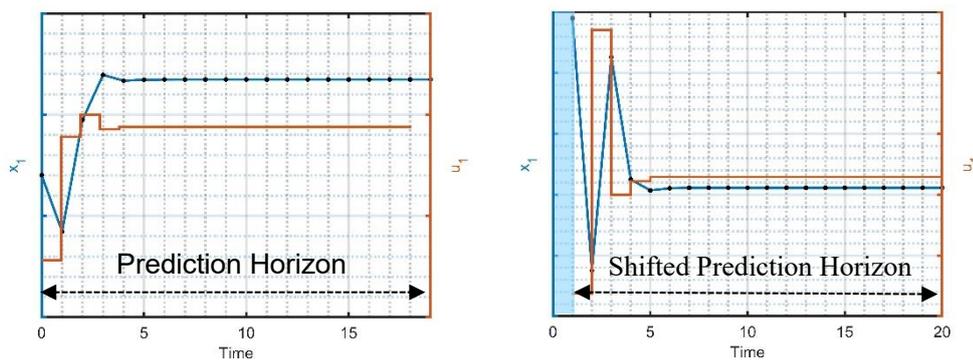


Figure 3.1 - Predicted state and control throughout the prediction horizon [17]

At each sampling time a mathematical optimization problem must be solved; this computational complexity represents the main limitation of MPC compared to PID controller. For this reason, in the past the most successful field for MPC was in the process industry, where the dynamic is quite slow and ensuring the satisfaction of hard

constraint is essential. Nowadays, due to the development of increasingly fast and energy-efficient processors, application of MPC law can spread also for dynamic system with smaller time constants.

Model Predictive Control uses a dynamic model to forecast system behavior and to produce the best control action in order to optimize the forecast [1]. The main goal of a classical model predictive control is to find an optimal control sequence, which minimizes a cost function over a prediction horizon [2]. In this study a discrete time formulation will be considered for a linear time independent system:

$$M: \begin{cases} x(k+1|k) = Ax(k|k) + Bu(k|k) \\ y(k|k) = Cx(k|k) \end{cases}, \quad x(k) \in \mathbb{R}^n, \quad u(k) \in \mathbb{R}^m, \\ y(k) \in \mathbb{R}^p$$

where  $k \in \mathbb{Z}^+$  represents the actual instant and  $x(k+i|k)$  is the  $i^{th}$  ahead state prediction step. The horizon prediction  $N_p$  should be chosen in order to define a cost function  $J(U(k), x(k|k))$ :

$$J(U(k), x(k|k)) = \sum_{i=0}^{N_p-1} L(x(k+i|k), u(k+i|k)) + \Phi(x(k+N_p|k))$$

where  $L(\cdot)$  is a per-stage weighting function and  $\Phi(\cdot)$  is the terminal state weighting function. It has to be noted that the cost function  $J(\cdot)$  depends only on the current state  $x(k|k)$ , that is known, and on the control sequence  $U(k) = [u(k|k), u(k+1|k), \dots, u(k+N_c-1|k)]^T$ , that has to be derived via the optimization problem. A control horizon  $N_c$  smaller or equal than prediction horizon  $N_p$  can be chosen. If  $N_c < N_p$ :

$$u(k+i|k) = u(k+N_c|k), \quad N_c \leq i \leq N_p - 1.$$

The functions  $L(\cdot)$  and  $\Phi(\cdot)$  are quadratic:

$$L(x, u) = x^T Q x + u^T R u \quad \Phi(x) = x^T P x$$

Where  $Q$ ,  $R$  and  $P$  are symmetric and positive definite:  $Q = Q^T \succcurlyeq 0, R = R^T \succcurlyeq 0, P = P^T \succcurlyeq 0$ .

Functions  $L(\cdot)$  and  $\Phi(\cdot)$  can be written as follows:

$$L(x, u) = \|Qx\|_p + \|Ru\|_p \quad \Phi(x) = \|Px\|_p$$

with  $p = 1, \dots, \infty$ . Finally, the optimization problem can be formulated as follows:

$$\mathcal{P}_{N_p}(x): \begin{cases} \min_{U(k)} J(x(k|k), U(k)) \\ \text{s. t.} \\ x(k+1) = f(x(k), u(k)) \\ U(k) \in \mathbb{U} \\ x(k+i|k) \in \mathbb{X}, \quad i = 1, \dots, N_p - 1 \\ x(k+N_p|k) \in \mathbb{X}_f \subset \mathbb{X} \end{cases} \quad (9)$$

where  $\mathbb{U}$  is the input constraint set,  $\mathbb{X}$  is the state constraint set and  $\mathbb{X}_f$  is the terminal state constraint set. The problem  $\mathcal{P}_{N_p}(x)$  is solved by the minimizing control sequence  $U^0(k) = [u^0(k|k), u^0(k+1|k), \dots, u^0(k+N_c-1|k)]^T$ .

The cost function minimum is denoted by  $J^0(x(k|k)) = J^0(x(k|k), U^0(k))$ .

Summarizing the procedure:

1. Get the state  $x(k|k)$ ;
2. Solve the optimization problem  $\mathcal{P}_{N_p}(x)$  and get optimal control sequence  $U^0(k)$ ;
3. Apply to the plant only the first element of the optimal control sequence  $u^0(k|k)$ ;
4.  $k = k + 1$  and go to step 1.

### 3.1 Linear MPC without constraints

In this paragraph it is considered the same linear, discrete time, time invariant system  $M$  of the last section with  $n$  state variables,  $m$  inputs and  $p$  outputs. Furthermore, it is assumed that the prediction horizon is equal to the control horizon ( $N_p = N_c$ ) Considering a quadratic cost function:

$$\begin{aligned} J(U(k), x(k|k)) &= \sum_{i=0}^{N_p-1} L(x(k+i|k), u(k+i|k)) + \Phi(x(k+N_p|k)) = \\ &= \sum_{i=0}^{N_p-1} x(k+i|k)^T Q x(k+i|k) + u(k+i|k)^T R u(k+i|k) \\ &\quad + x(k+N_p|k)^T P x(k+N_p|k) \\ &= \sum_{i=0}^{N_p-1} \|x(k+i|k)\|_Q^2 + \|u(k+i|k)\|_R^2 + \|x(k+N_p|k)\|_P^2 \end{aligned}$$

This quadratic cost function will be reformulated in order to explicit its dependence on the control sequence  $U(k)$  and the actual state  $x(k|k)$ . It can be noticed

that the state  $x(k + i|k)$  at  $i^{th}$  step is only function of the control sequence and the actual state:

$$x(k + i|k) = A^i x(k|k) + \sum_{j=0}^{i-1} A^{i-j-1} B u(k + j|k), i = 1, \dots, N_p$$

The latter equation can be rewritten in vectorial form  $X(k) = \mathcal{A}x(k|k) + \mathcal{B}U(k)$ , where  $X(k)$  is the predicted state sequence  $X(k) = [x(k + 1|k), \dots, x(k + N_p|k)]^T \in \mathbb{R}^{nN_p}$  and the matrices  $\mathcal{A}$  and  $\mathcal{B}$  are constructed as follows:

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ A^{N_p-2}B & A^{N_p-3}B & \dots & B & 0 \\ A^{N_p-1}B & A^{N_p-2}B & \dots & AB & B \end{bmatrix}$$

where  $\mathcal{A} \in \mathbb{R}^{nN_p, n}$  and  $\mathcal{B} \in \mathbb{R}^{nN_p, mN_p}$ . Defining the matrix of the weight for states  $Q$  and control variables  $\mathcal{R}$ :

$$Q = \begin{bmatrix} Q & 0 & 0 & \dots & 0 \\ 0 & Q & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & Q & 0 \\ 0 & 0 & \dots & 0 & P \end{bmatrix} \in \mathbb{R}^{nN_p, nN_p} \quad \mathcal{R} = \begin{bmatrix} R & 0 & 0 & \dots & 0 \\ 0 & R & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & R & 0 \\ 0 & 0 & \dots & 0 & R \end{bmatrix} \in \mathbb{R}^{mN_p, mN_p}$$

The cost function can be now rewritten as  $J(U(k), x(k|k)) = X(k)^T Q X(k) + U(k)^T \mathcal{R} U(k)$ .

Writing the predicted state sequence as function of the actual state and of the control sequence:

$$\begin{aligned} J(U(k), x(k|k)) &= x(k|k)^T \mathcal{A}^T Q \mathcal{A} x(k|k) + 2x(k|k)^T \mathcal{A}^T Q \mathcal{B} U(k) + U(k)^T (\mathcal{B}^T Q \mathcal{B} \\ &+ \mathcal{R}) U(k) \end{aligned}$$

Assuming that:

$$H = 2(\mathcal{B}^T Q \mathcal{B} + \mathcal{R})$$

$$F = 2\mathcal{A}^T Q \mathcal{B}$$

$$\bar{J} = x(k|k)^T \mathcal{A}^T Q \mathcal{A} x(k|k)$$

where  $H$  is positive definite and  $\bar{J}$  is a known constant term, which depends only on the actual state, cost function can be finally be written as function of  $x(k|k)$  and  $U(k)$ :

$$J(U(k), x(k|k)) = \frac{1}{2} U(k)^T H U(k) + x(k|k)^T F U(k) + \bar{J}$$

To solve the optimization problem  $\mathcal{P}_{N_p}(x)$ , this cost function must be minimized. From the first derivative of the cost function:

$$\frac{\partial J}{\partial U} = HU(k) + x(k|k)^T F$$

The necessary minimum condition is  $\frac{\partial J}{\partial U} = 0$ . The optimal control sequence is:

$$U^0(k) = -H^{-1}F^T x(k|k) = -Kx(k|k)$$

where matrix  $K$  corresponds to a state feedback matrix. An appropriate choice of matrix  $Q$ ,  $R$  and  $P$  guarantees asymptotic stability.

### 3.2 Linear MPC with constraints

In real operative cases, most of the times optimization problem  $\mathcal{P}_{N_p}(x)$  is subjected to constraints on state variables and on control variables. In this section it is examined how to deal with constraints. It is considered always the same linear, discrete time, time invariant system  $M$  with  $n$  states,  $m$  inputs and  $p$  outputs. It is assumed that every input has upper and lower saturation constraints and that control horizon is equal to the prediction horizon  $N_p$ .

#### 3.2.1 Input saturation constraints

The input vector is a column vector ( $u(k|k) \in \mathbb{R}^m$ ) and the control sequence too ( $U(k) \in \mathbb{R}^{mN_p}$ ). In the following example the same constraints are imposed at every sample time.

$$\begin{aligned} u_{min} &\leq u(k|k) \leq u_{max} \\ u_{min} &\leq u(k+1|k) \leq u_{max} \\ &\vdots \\ u_{min} &\leq u(k+N_p-1|k) \leq u_{max} \end{aligned}$$

These inequalities can be rewritten as follows:

$$\begin{aligned} u(k|k) &\leq u_{max} \\ u(k+1|k) &\leq u_{max} \\ &\vdots \\ u(k+N_p-1|k) &\leq u_{max} \\ -u(k|k) &\leq -u_{min} \\ -u(k+1|k) &\leq -u_{min} \end{aligned}$$

$$\begin{aligned} & \vdots \\ & -u(k + N_p - 1|k) \leq -u_{min} \end{aligned}$$

Utilizing matrix formulation, these inequalities can be expressed as:

$$\begin{aligned} [I] \begin{bmatrix} u(k|k) \\ \vdots \\ u(k + N_p - 1|k) \end{bmatrix} &\leq W_{U_{max}} \\ -[I] \begin{bmatrix} u(k|k) \\ \vdots \\ u(k + N_p - 1|k) \end{bmatrix} &\leq W_{U_{min}} \end{aligned}$$

where  $I \in \mathbb{R}^{mN_p, mN_p}$  is an identity,  $W_{U_{max}} = [u_{max}, u_{max}, \dots, u_{max}]^T \in \mathbb{R}^{mN_p}$  and  $W_{U_{min}} = [-u_{min}, -u_{min}, \dots, -u_{min}]^T \in \mathbb{R}^{mN_p}$ . These inequalities can be finally written as linear inequalities:

$$L_U U(k) \leq W_U$$

where:

$$L_U = \begin{bmatrix} I \\ -I \end{bmatrix} \in \mathbb{R}^{2mN_p, mN_p} \quad W_U = \begin{bmatrix} W_{U_{max}} \\ W_{U_{min}} \end{bmatrix} \in \mathbb{R}^{2mN_p}$$

### 3.2.2 State constraints

In the following section it is considered any kind of linear state constraints, not only saturation constraints:

$$\begin{aligned} L_1 x(k + 1|k) &\leq W_1 \\ &\vdots \\ L_i x(k + i|k) &\leq W_i \\ &\vdots \\ L_{N_p} x(k + N_p|k) &\leq W_{N_p} \end{aligned}$$

where  $L_i \in \mathbb{R}^{n, n}$  and  $W_i \in \mathbb{R}^n$  with  $i = 1, \dots, N_p$ . Remembering that  $x(k + i|k) = A^i x(k|k) + \sum_{j=0}^{i-1} A^{i-j-1} B u(k + j|k)$ ,  $i = 1, \dots, N_p$ , it is possible to write state constraints as:

$$\begin{aligned} \begin{bmatrix} L_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & L_{N_p} \end{bmatrix} \begin{bmatrix} B & \dots & 0 \\ \vdots & \ddots & \vdots \\ A^{N_p-1} & \dots & B \end{bmatrix} \begin{bmatrix} u(k|k) \\ \vdots \\ u(k + N_p - 1|k) \end{bmatrix} \\ \leq - \begin{bmatrix} L_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & L_{N_p} \end{bmatrix} \begin{bmatrix} A \\ \vdots \\ A^{N_p} \end{bmatrix} x(k|k) + \begin{bmatrix} W_1 \\ \vdots \\ W_{N_p} \end{bmatrix}, \end{aligned}$$

which can be rewritten as  $L_x \mathcal{B}U(k) \leq -L_x \mathcal{A}x(k|k) + W_x$ , where  $L_x \in \mathbb{R}^{nN_p, nN_p}$  and  $W_x \in \mathbb{R}^{nN_p}$ . As it can be noticed, the state constraints depend on the actual state of the system; hereby at each step the right side of the last inequality changes. After posing  $L_X = L_x \mathcal{B} \in \mathbb{R}^{nN_p, mN_p}$  and  $W_X = -L_x \mathcal{A}x(k|k) + W_x \in \mathbb{R}^{nN_p}$ , this inequality is compacted as  $L_X U(k) \leq W_X$ . Rearranging both input and state constraints it is possible to write:

$$LU(k) \leq W$$

$$\text{where } L = [L_U, L_X]^T \in \mathbb{R}^{(2m+n)N_p, mN_p} \text{ and } W = [W_U, W_X]^T \in \mathbb{R}^{(2m+n)N_p}.$$

### 3.2.3 Quadratic Programming Solution

Optimization problem for Model Predictive Control is the following:

$$\begin{aligned} \min_{U(k)} J(U(k), x(k|k)) \\ &= \min_{U(k)} \frac{1}{2} U(k)^T H U(k) + x(k|k)^T F U(k) + \bar{J} \\ &\text{s. t.} \\ &LU(k) \leq W \end{aligned} \quad (10)$$

Matrix  $H$  is said the Hessian of quadratic programming. If  $H$  is positive definite, the quadratic problem is convex [22].

To solve this problem in MATLAB/Simulink, it is given the instruction `quadprog`, which solves the optimization constrained problem through an interior-point method [31].

### 3.3 Stability analysis for linear MPC

Stability must be ensured for a closed loop system based on receding horizon principle. In this chapter Lyapunov direct method is invoked to draw conclusions about stability. The optimal cost function  $J^0(x(k), U(k))$  is taken as candidate as Lyapunov function for asymptotic stability. Stability must be guaranteed over an infinite horizon. Because a finite future horizon is considered, some methods are used to choose appropriate weighting on the terminal state  $\Phi(\cdot)$  and an appropriate terminal constraint set  $\mathbb{X}_f$ . In this section the general formulation of the optimization problem  $\mathcal{P}_{N_p}(x)$ , written as in (9), is studied.

Some definitions are useful for the following discussion:

- Feasible initial set: the set  $\mathbb{S}_{N_p}$  is the set of initial states  $x_0 \in \mathbb{X}$  for which there exist feasible state and control sequences for  $\mathcal{P}_{N_p}(x)$ .
- Positively invariant set [23]: a set  $\mathbb{S} \in \mathbb{R}^n$  is said positively invariant for the closed loop system  $x(k+1) = f(x(k), u(k))$ , where  $u(k) = \mathcal{K}(x(k))$ , if  $f(x(k), \mathcal{K}(x(k))) \in \mathbb{S}$  for all  $x(k) \in \mathbb{S}$ .

In other words, if the initial state  $x_0$  belongs to the feasible initial set  $\mathbb{S}$ , which is also positively invariant, all the future state will be contained in the set  $\mathbb{S}$ , too. In the following methods a linear time independent, discrete time state-space, defined in (7), is adopted. Only in this section control horizon is not the same as prediction horizon.

The first method, proposed in [24], is the zero terminal constraints. The optimization problem  $\mathcal{P}_{N_p}(x)$  can be rewritten as:

$$\mathcal{P}_{N_p}(x): \begin{cases} \min_{U(k)} J(x(k|k), U(k)) = \sum_{i=0}^{N_p-1} \|x(k+i|k)\|_Q^2 + \|u(k+i|k)\|_R^2 \\ \text{s. t.} \\ x(k+1) = f(x(k), u(k)) \\ U(k) \in \mathbb{U} \\ x(k+i|k) \in \mathbb{X}, \quad i = 1, \dots, N_p - 1 \\ u(k+i|k) = 0, \quad i = N_c, \dots, N_p - 1 \\ x(k+N_p|k) = 0 \end{cases}$$

Supposing that the optimal control sequence, which makes the cost function minimum at time  $k$ , is  $U^0(k|k) = [u^0(k|k), u^0(k+1|k), \dots, u^0(k+N_c-1|k)]$  and the corresponding state sequence is  $X^0(k|k) = [x(k|k), x(k+1|k), \dots, x(k+N_p-1|k), 0]$ , then the Lyapunov function candidate is  $V(x(k|k)) = J(U^0(k|k), x(k|k))$ . The control law  $u^0(k|k)$  is applied and at the following sampling time  $k+1$  it is supposed to use the nonoptimal control sequence  $U_1(k+1|k+1) = [u^0(k+1|k), u^0(k+2|k), \dots, u^0(k+N_c-1|k), 0]$ , which has the same predicted control laws of the instant  $k$ ; the subsequent state sequence is  $X_1(k+1|k+1) = [x(k+1|k), x(k+2|k), \dots, x(k+N_p|k), 0]$ . The resulted cost function can be written as:

$$J(U_1, x(k+1|k+1)) = J(U^0, x(k|k)) - x^T(k|k)Qx(k|k) - u^T(k|k)Ru(k|k)$$

Since  $U_1$  is not the optimal control, it follows that  $J(U_1, x(k+1)) \geq J(U_1^0, x(k+1))$ , where  $U_1^0 = [u_1^0(k+1|k+1), u_1^0(k+2|k+1), \dots, u_1^0(k+N_c|k+1)]$  is the optimal control sequence. At the instant  $k+1$  the Lyapunov function candidate will be  $V(x(k+1|k+1)) = J(U_1^0(k+1|k+1), x(k+1|k+1))$ , if the optimal control sequence is introduced. Since the cost function is positive definite, it follows that the equilibrium is asymptotically stable if and only if the Lyapunov function increment  $\Delta V(k)$  is negative:

$$\begin{aligned}\Delta V(k) &= V(x(k+1|k+1)) - V(x(k|k)) \\ &= J(U_1^0(k+1|k+1), x(k+1|k+1)) - J(U^0(k|k), x(k|k)) \\ &\leq J(U_1, x(k+1|k+1)) - J(U^0(k|k), x(k|k)) \\ &= -x^T(k|k)Qx(k|k) - u^T(k|k)Ru(k|k) < 0\end{aligned}$$

This result shows that asymptotic stability is guaranteed with the zero terminal constraint.

The second method considers the terminal state nonnull and an appropriate terminal weighting matrix must be determined [25]. The optimization problem  $\mathcal{P}_{N_p}(x)$  can be rewritten as:

$$\mathcal{P}_{N_p}(x): \begin{cases} \min_{U(k)} J(x(k|k), U(k)) = \sum_{i=0}^{N_p-1} \|x(k+i|k)\|_Q^2 + \|u(k+i|k)\|_R^2 + \|x(k+N_p|k)\|_P^2 \\ \quad \text{s. t.} \\ \quad x(k+1) = f(x(k), u(k)) \\ \quad U(k) \in \mathbb{U} \\ \quad x(k+i|k) \in \mathbb{X}, \quad i = 1, \dots, N_p - 1 \\ \quad u(k+i|k) = 0, \quad i = N_c, \dots, N_p - 1 \end{cases}$$

The terminal weighting matrix  $P$  is chosen as solution of the discrete time Lyapunov equation:

$$A^T P A + Q = P$$

To derive the Lyapunov function increment  $\Delta V(k)$  a similar procedure as in the first method is followed; the only difference this time is that the state sequence at time  $k+1$  is  $X_1(k+1|k+1) = [x(k+1|k), x(k+2|k), \dots, x(k+N_p|k), x(k+N_p+1|k)]$ . The Lyapunov function increment can now be written as:

$$\begin{aligned}
\Delta V(k) &= V(x(k+1|k+1)) - V(x(k|k)) \\
&= J(U_1^0(k+1|k+1), x(k+1|k+1)) - J(U^0(k|k), x(k|k)) \\
&\leq J(U_1, x(k+1|k+1)) - J(U^0(k|k), x(k|k)) \\
&= -x^T(k|k)Qx(k|k) - u^T(k|k)Ru(k|k) \\
&\quad + x^T(k+N_p|k)(Q-P)x(k+N_p|k) \\
&\quad + x^T(k+N_p+1|k)Px(k+N_p+1|k) \\
&= -x^T(k|k)Qx(k|k) - u^T(k|k)Ru(k|k) + M_k
\end{aligned}$$

where  $M_k$  collects the state vectors in instants  $k+N_p$  and  $k+N_p+1$ . It is now necessary to demonstrate that the system is asymptotically stable if and only if  $\Delta V(k) < 0$ , hereby if  $M_k \leq 0$ . Since  $x(k+N_p+1|k) = Ax(k+N_p|k)$ , it follows:

$$\begin{aligned}
M_k &= x^T(k+N_p|k)(Q-P)x(k+N_p|k) + x^T(k+N_p+1|k)Px(k+N_p+1|k) \\
&= x^T(k+N_p|k)(Q-P)x(k+N_p|k) \\
&\quad + \left(Ax(k+N_p|k)\right)^T P \left(Ax(k+N_p|k)\right) \\
&= x^T(k+N_p|k)(A^T P A + Q - P)x(k+N_p|k) = 0
\end{aligned}$$

The third presented method is the one used in this thesis. The optimization problem  $\mathcal{P}_{N_p}(x)$  is modified as it follows:

$$\mathcal{P}_{N_p}(x): \left\{ \begin{array}{l} \min_{U(k)} J(x(k|k), U(k)) = \sum_{i=0}^{N_p-1} \|x(k+i|k)\|_Q^2 + \|u(k+i|k)\|_R^2 + \|x(k+N_p|k)\|_P^2 \\ \text{s. t.} \\ x(k+1) = f(x(k), u(k)) \\ U(k) \in \mathbb{U} \\ x(k+i|k) \in \mathbb{X}, \quad i = 1, \dots, N_p - 1 \\ u(k+i|k) = -K_{LQ}x(k+i|k), \quad i = N_c, \dots, N_p - 1 \\ x(k+N_p|k) \in \Omega_{LQ} \supset 0 \end{array} \right.$$

Where  $\Omega_{LQ}$  is the positive invariant terminal set and it contains the origin. The feedback matrix  $K_{LQ}$  and the terminal weighting matrix  $P$  are the solution of the algebraic Riccati equation (ARE):

$$P = Q + A^T(P - PB(R + B^T P B)^{-1}B^T P)A$$

Where the feedback matrix  $K_{LQ} = -(R + B^T P B)^{-1}B^T P$ .

## Chapter 4: Tube-Based Robust Model Predictive Control

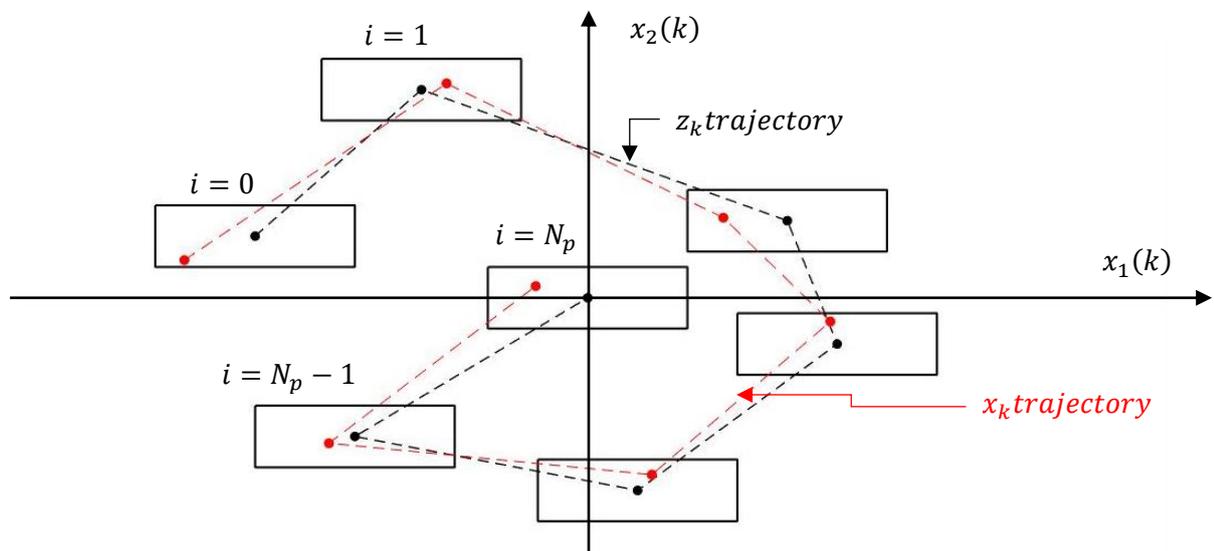
A classical Model Predictive Control approach is not always suitable for systems, such as UAV, which operates in an environment subjected to disturbances and noise. A different approach is necessary in order to meet hard constraint requirements and to guarantee robustness to exogenous disturbances. A Model Predictive Control variant is the Tube-based Model Predictive Control, which can control system that are uncertain. As uncertain system it is meant a system, whose actual behavior is not identical to the predicted behavior of the nominal system. Tube-based MPC [1] ensures that a dynamic system subjected to a bounded disturbance respects the imposed constraints. It is obvious that a disturbance is always assumed to be bounded, because it is impossible to guarantee stability with unbounded disturbances.

### 4.1 Overview

There are many MPC extensions, which let to obtain a Robust MPC, able to handle both uncertainties and hard constraints. One of the most attractive control methods for practical applications is Tube-Based Robust MPC [1], which is able to meet hard constraints in presence of random bounded noise and uncertainties. This approach provides robustness to disturbance with the same computational efficiency of a classical MPC. Indeed, the online procedure consists of solving a quadratic optimization problem in the same way as Model Predictive Control. The main strength of this procedure is that the problems of constraint requirements and external disturbances are treated separately.

In this case two systems are considered in parallel: the nominal system and the disturbed system. The first one is a system written as a linear discrete time state-space, which models the undisturbed dynamic of the system. The second one is the actual system, which is subjected to uncertainties and external disturbances and it is requested that it respects some state and control constraints. During the simulation these two systems run in parallel and provide respectively the system state without disturbances and the actual system state. A conventional Model Predictive Control operates on the nominal system and the nominal optimal control sequence is elaborated as a function

of the current nominal state. The inputs furnished to the nominal system are not the same of the actual system. The actual system inputs are derived as a function of the nominal control and the error, defined by the discrepancy between the actual and the nominal state. The error dynamic can be obtained by the actual and nominal dynamics. In order to stabilize error dynamic, it is necessary to make the state matrix eigenvalues real part negative through the introduction of a feedback gain matrix. This matrix is defined by an offline procedure, which consists of solving a Linear Matrix Inequality problem. Once the feedback gain matrix is derived, it is possible to define some tightened nominal state and control constraint sets, with which it is ensured that the original constraints for the actual system are respected also in presence of external noise. Since it is impossible to control a system affected by unlimited disturbances, it follows that disturbances must be bounded. Under these hypotheses a “tube” can be built. The center of the “tube” consists of the nominal state variables. Tube-based MPC ensures that the actual state variables never exceed the limits imposed by the “tube” constructed around nominal states. The section of the “tube” depends on the feedback gain matrix and on the disturbance boundaries. In Figure 4.1 it is shown an example of nominal and actual state sequence at the current time  $k$ .



**Figure 4.1** - Outer-bounding tube representation at the  $k$ -th time step over a prediction horizon of  $N$

The black points are the nominal states throughout the prediction horizon  $N_p$  and the black dashed line represents their trajectory. The tube is defined by the black

rectangles around the nominal states. Tube-based MPC ensures that the actual states never go out the boundaries specified by the rectangles.

In this section it is considered a system with parametric uncertainties and subject to an additive disturbance  $w$ , which satisfies constraint  $w \in \mathbb{W}$  where  $\mathbb{W} \subset \mathbb{R}^n$ .

$$x(k+1) = f(x(k), u(k)) + w,$$

where disturbed state  $x(k)$  is constrained to belong to the set  $\mathbb{X} \subset \mathbb{R}^n$  and the control must belong to the set  $\mathbb{U} \subset \mathbb{R}^m$ . The nominal system simulates undisturbed system dynamic, runs together actual system and lets to define the discrepancy between where the system should be if it would operate in a world without uncertainties and disturbances and where actually it is. Nominal state is described by:

$$z(k+1) = Az(k) + Bv(k),$$

where  $z(k)$  and  $v(k)$  are optimal solutions for zero disturbance and are respectively constrained to the sets  $\mathbb{Z} \subset \mathbb{X}$  and  $\mathbb{V} \subset \mathbb{U}$ . Hence, nominal states and controls are subjected to tighter constraints compared to the disturbed system. In this way the uncertain system trajectories can lie always within the “tube” around the predicted nominal trajectory and it is guaranteed that the original constraints are respected by uncertain system.

A conventional Model Predictive Control is applied to the nominal system and for each sampling time a nominal control  $v(k)$  is elaborated. Actual control can be derived at each step as a function of the nominal control  $v(k)$  and of the discrepancy between actual and nominal state:

$$u(k) = v(k) + K(x(k) - z(k))$$

This control feedback can be adequate at least when  $f(\cdot)$  is linear. The linear-time feedback controller  $K$  is chosen to ensure that the deviation of uncertain system from the nominal one is bounded. Since the matrix  $K$  can be determined offline, it follows that Tube-based MPC has the same order of complexity as the conventional one. The Tube-based MPC efficiency is due to the constrained control problem separation from uncertainties handling problem. A standard Quadratic Program can be fast solved with standard mathematical optimization algorithms. The resulted computational efficiency makes Tube-based MPC interesting also for dynamical system with high sampling frequencies.

This kind of control succeeds only when assumptions are satisfied and the disturbances not exceed the imposed bounds. Robustness is guaranteed only for the specified uncertainties.

## 4.2 Problem statement and control strategy

Let define a linear-time independent, discrete time state-space model subjected to a persistent disturbance:

$$x(k+1) = A_d x(k) + B_d u(k) + w(k) \quad (11)$$

Let refer to this system as uncertain system, where  $x(k)$  is the state vector,  $u(k)$  is the control vector and  $w(k)$  the additive disturbance. In this section  $k$  constitutes the actual time and it is a positive integer ( $k \in \mathbb{Z}^+$ ). The controlled system is required to satisfy the following constraints:

$$x(k) \in \mathbb{X}, u(k) \in \mathbb{U}$$

where  $\mathbb{X}$  is a subset of  $\mathbb{R}^n$  and  $\mathbb{U}$  is a subset of  $\mathbb{R}^m$ . Furthermore, to obtain a robust control with respect to additive noise  $w(k)$ , it is necessary that the disturbance is bounded. Indeed, the disturbance  $w(k)$  belongs to  $\mathbb{W}$ , which is a subset of  $\mathbb{R}^n$ . The central trajectory of the tube corresponds to the predicted behavior of the nominal system, which is defined as:

$$z(k+1) = A_d z(k) + B_d v(k)$$

As it can be seen, nominal system is the undisturbed system, for which mathematical optimization problem is solved. To ensure the respect of uncertain system's constraints, it is required that nominal state and control variables meet tightener constraints. Indeed, it is requested that the state vector  $z(k)$  belongs to a set  $\mathbb{Z}$ , which is a subset of previous defined set  $\mathbb{X}$ , and that the control vector  $v(k)$  belongs to a set  $\mathbb{V}$ , where  $\mathbb{V} \subset \mathbb{U}$ .

The two defined systems will not evolve in the same way. The actual state  $x(k)$  will be deviated from the nominal state  $z(k)$ . To quantify this deviation, it is defined an error at the  $i^{th}$  steps ahead  $k$ :

$$e(i|k) = x(i|k) - z(i|k)$$

This error is necessary to apply a correction to the nominal optimal control  $v(k)$ , which it is calculated without considering disturbances and taking into account the nominal state of the system  $z(k)$  and not the actual one  $x(k)$ . The actual optimal control is derived by the following relation:

$$u(i|k) = v(i|k) + K(x(i|k) - z(i|k)) \quad (12)$$

where  $K$  is a feedback matrix chosen off-line in order to guarantee stability. How to choose it, it will be explained in the following part. Doing the difference between uncertain system and nominal system, it can be obtained the error dynamic:

$$\begin{aligned} x(k+1) - z(k+1) &= A_d(x(k) - z(k)) + B_d(u(k) - v(k)) + w(k) \\ e(k+1) &= (A_d + B_dK)e(k) + w(k) \end{aligned} \quad (13)$$

The error dynamic depends on the stability of the matrix  $A_K = A_d + B_dK$ . A correct choice of feedback matrix  $K$  makes the matrix  $A_K$  robustly stable. The stability analysis is treated in the next section.

#### 4.2.1 Optimal control problem statement

The classical optimization problem is solved for the nominal system through an optimization window  $N_p$  and a control horizon  $N_c = N_p$ :

$$\begin{aligned} J(V(k|k), Z(k|k)) \\ &= \sum_{i=0}^{N_p-1} (z(k+i|k)^T Q z(k+i|k) + v(k+i|k)^T R v(k+i|k)) \\ &\quad + z(k+N_p|k)^T P z(k+N_p|k) \end{aligned}$$

where  $V(k|k)$  is the nominal optimal control sequence over a  $N_p$  prediction horizon and  $Z(k|k)$  is the nominal predicted trajectory. The state and control weight matrices  $Q \in \mathbb{R}^{n,n}$  and  $R \in \mathbb{R}^{m,m}$  are both positive definite whereas the terminal state weight matrix  $P \in \mathbb{R}^{n,n}$  is the solution of the discrete Algebraic Riccati equation. Finally, optimization problem is defined as:

$$\begin{aligned} \min_{V(k)} J(V(k), Z(k)) \\ \text{s. t. } z(0|k) &= x(0|k) \\ z(i|k) &\in \mathbb{Z}, i \in [1, N-1] \\ v(i|k) &\in \mathbb{V}, i \in [0, N-1] \\ z(N|k) &\in \mathbb{Z}_f \end{aligned} \quad (14)$$

As it can be seen, it is imposed that nominal initial state is equal actual initial state and a terminal set  $\mathbb{Z}_f$  is defined as subset of  $\mathbb{Z}$ .

#### 4.2.2 Stability analysis

Stability analysis is performed to choose a suitable state feedback controller  $K$  which stabilizes error dynamic. The matrix  $K$  is chosen such that  $A_K = A + BK$  is Hurwitz. There exists a  $\tilde{P} \in \mathbb{R}^{n,n}$  such that

$$(A_d(q) + B_d(q)K)^T \tilde{P} (A_d(q) + B_d(q)K) - \tilde{P} \preceq 0, \quad \tilde{P} > 0$$

And the feedback gain matrix  $K$  stabilizes the system with respect to parametric uncertainty  $q$ . These uncertainties are due to differences between the mathematical model and the actual dynamic, neglected nonlinearities and reduced-order model approximations. The Edge Theorem, which is an extension of the Karitonov's theorem, is employed to affirm that the stability of a polytope of polynomials  $\hat{P}$  is ensured by the stability of its one-dimensional exposed edge polynomials. The family  $\hat{P}$  is defined as it follows:

$$\hat{P} = \left\{ p(s, q) = a_0(q) + a_1(q)s + \dots + a_{n-1}(q)s^{n-1} + s^n; a_i(q) \right. \\ \left. = a_{i_0} + \sum_{k=1}^l a_{i_k} q_k, q \in B_q, \quad i = 0, \dots, n-1 \right\}$$

where  $B_q$  contains affine functions of the uncertain vector  $q = [q_1, \dots, q_l]$ .

$$B_q = \{q \in \mathbb{R}^l | q_i \in [q_i^-, q_i^+], i = 1, \dots, l\} \quad (15)$$

The family  $\hat{P}$  is Hurwitz if and only if all edges of  $\hat{P}$  are Hurwitz. In the same way the family of  $\tilde{P}$  is Hurwitz if all the vertexes are stable. Therefore, the state and control matrices for the biggest parametric uncertainties are built as the following  $A_d^- = A(q^-)$ ,  $A_d^+ = A(q^+)$ ,  $B_d^- = B(q^-)$  and  $B_d^+ = B(q^+)$ . The following LMIs system is solved to obtain the feedback gain matrix  $K$  which stabilizes the system with respect to the uncertainties  $q \in B_q$ .

$$\begin{cases} (A_d^+ + B_d^+ K)^T \tilde{P} (A_d^+ + B_d^+ K) - \tilde{P} \preceq 0 \\ (A_d^+ + B_d^- K)^T \tilde{P} (A_d^+ + B_d^- K) - \tilde{P} \preceq 0 \\ (A_d^- + B_d^+ K)^T \tilde{P} (A_d^- + B_d^+ K) - \tilde{P} \preceq 0 \\ (A_d^- + B_d^- K)^T \tilde{P} (A_d^- + B_d^- K) - \tilde{P} \preceq 0 \end{cases}$$

Before to proceed with the calculation of the tightened constraint sets  $\mathbb{V}$  and  $\mathbb{Z}$ , it is necessary to introduce some important concepts:

- Robust positively invariant set [23]: a set  $\Omega$  is said to be robust positively invariant (RPI) for the system

$$x(k+1) = f(x(k), w(k))$$

If for all  $x(0) \in \Omega$  and  $w(k) \in \mathbb{W}$  the solution  $x(k) \in \Omega$  for all  $k \in \mathbb{Z}^+$ . The set  $\Omega$  is also minimal positively invariant set for the system if it is contained in every closed RPI set of the system [26].

- The Minkovski set addition [27] between given two sets  $\mathcal{A} \subseteq \mathbb{R}^n$  and  $\mathcal{B} \subseteq \mathbb{R}^n$  is defined by

$$\mathcal{A} \oplus \mathcal{B} = \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$$

- The Pontragyn set difference [28] between given two sets  $\mathcal{A} \subseteq \mathbb{R}^n$  and  $\mathcal{B} \subseteq \mathbb{R}^n$  is defined by

$$\mathcal{A} \ominus \mathcal{B} = \{c \in \mathbb{R}^n | a + b \in \mathcal{A}, \forall b \in \mathcal{B}\}$$

Once feedback gain  $K$  is found, it is possible to calculate the minimal robust positive invariant set  $S_k(\infty)$  for the disturbed system  $x(i+1|k) = A_d x(i|k) + w(i|k)$ ,  $w \in \mathbb{W}$ :

$$S_k(\infty) = \sum_{j=0}^{\infty} A_K^j \mathbb{W}$$

If the matrix  $A_K$  is stable, then  $S_k(\infty)$  exists and it is positive invariant for the disturbed system (13). Therefore if  $e(k) \in S_k(\infty)$ , it implies that  $e(k+1) \in S_k(\infty)$  for all  $w \in \mathbb{W}$  [28]. This operation is a summation of Minkovski sets addition. The tighten constraint sets are derived by:

$$\begin{aligned} \mathbb{Z} &\subseteq \mathbb{X} \ominus S_k(\infty) \\ \mathbb{V} &\subseteq \mathbb{U} \ominus K S_k(\infty) \end{aligned} \tag{16}$$

#### 4.2.3 Overall TRMPC algorithm

The algorithm to build a Tube-based Robust Model Predictive Control can be divided in two parts:

- Offline procedure, which consists of solving a Linear Matrix Inequality problem, in order to obtain a suitable feedback matrix  $K$  and then the tightened constraints for nominal states and inputs.
- Online procedure, which consists of solving at each sampling time an optimization problem for the nominal state-space model and then deriving the disturbed system input, correcting the nominal control input in function of the error between nominal and actual dynamic.

Offline procedure can be summarized in the following steps:

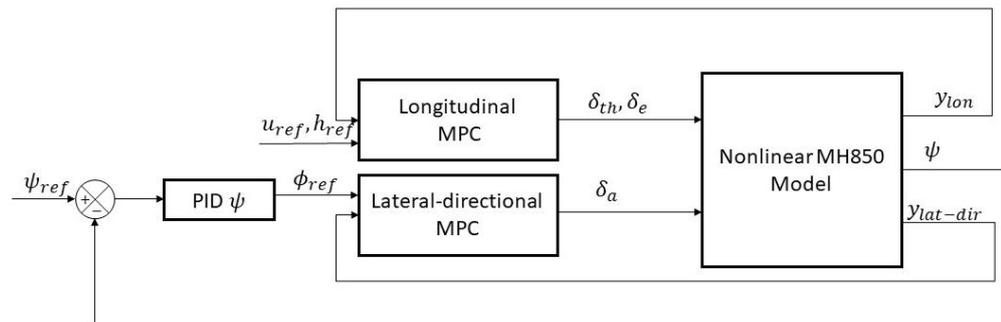
1. Define uncertainty set  $B_q$  as in (15)
2. Build the couple  $(A^i, B^i)$  for each  $i^{th}$  vertex with respect to uncertainties  $q$
3. Build  $sys_i = AX + XA^T + BY + Y^T B^T < 0$  for each  $i^{th}$  vertex, where  $X = \tilde{P}^{-1}$  and  $Y = K\tilde{P}^{-1} = KX$
4. Solve  $[X > 0, sys_i < 0]$  for each  $i$
5. Derive  $\tilde{P} = X^{-1}$  and  $K = YX^{-1}$
6. If  $A + BK$  is stable, go to the next step; otherwise go to step 1 and consider other uncertainties  $q$
7. Evaluate the tightened constraint sets  $\mathbb{Z}$  and  $\mathbb{V}$  for the nominal system as in (16)

In this study to solve the algorithm step 4, it is used MATLAB in combination with the optimization toolbox YALMIP [29]. Once offline procedure is completed and the feedback gain matrix  $K$  and the tightened constraint sets  $\mathbb{Z}$  and  $\mathbb{V}$  are evaluated, it is possible to apply the online procedure, which is computational efficient as conventional MPC algorithm. The Online procedure is defined by the following steps:

1. At current time  $k$   $x(0|k) = x(k|k)$
2. Set  $z(0|k) = x(k|k)$
3. Solve (14) and get the optimal control sequence for nominal system  $V^0(k|k)$
4. Apply to the nominal system only the first element of the optimal control sequence  $v^0(k|k)$
5. Evaluate  $x(k|k)$  and apply to the actual plant the corrected control  $u(k|k)$  according to (12)
6.  $k = k + 1$  and go to step 3.

## Chapter 5: Simulation Results

The design of a Model Predictive Controller is presented in this section. The UAV nonlinear dynamic is simulated through the implementation of nonlinear equations of motion (3), (4) and (5) presented in section Nonlinear Model. These equations are collected in a box which receive as input the control variables, derived in MPC box, and as output the system state. A guidance algorithm is implemented in order to guide the UAV towards predefined waypoints and to provide some references such as  $u_{ref}$ ,  $h_{ref}$  and  $\psi_{ref}$ , which must be followed by the system. This latter task is accomplished by the Model Predictive Controller, which has as inputs the references and as outputs the commands to be applied on the control surfaces and on the throttle. The only state variable controlled by a PID is the heading angle  $\psi$  for the outer navigation loop. Furthermore, a linear Model Predictive Controller is adopted; therefore, longitudinal and lateral-directional planes are studied independently and two Model Predictive Controller are designed. In Figure 5.1 it is shown a simplified scheme of the model.



**Figure 5.1** - Scheme of the controller proposed in this work

Several paths are considered in order to test the effectiveness of the tuning in both controllers. Since the MPCs are linear and so built with the linearization of the equation of motions around an equilibrium point defined by airspeed  $U_0 = 13.5 \text{ m/s}$  and altitude  $h = 100 \text{ m}$ , it follows only the waypoint position and initial UAV orientation are changed.

The Tube-based Model Predictive Control version is described by the almost same model. A disturbance is introduced in the nonlinear model and two linear state-space

models beside the disturbed nonlinear model are adopted to describe the nominal system dynamics in longitudinal and lateral-directional plane.

## 5.1 MATLAB/Simulink Model description

The dynamic model is divided in two parts:

1. MATLAB part: the initial conditions are elaborated and all constants, tuning parameters, constraints, waypoints position and many others are defined. A focus on initial condition determination will be given. Once the simulation has run, the postprocessing is done in MATLAB.
2. Simulink part: it is where the simulation takes place and it will be described more in details. The Simulink part is considered, since the connection with the hardware board will be done with a Simulink interface.

### 5.1.1 Initial Condition Evaluation

The initial conditions are obtained with a numerically derived database. Starting altitude and airspeed are given. Vehicle's aerodynamic characteristics are known for some altitudes; if the initial altitude is not one the known ones, a linear interpolation between two known conditions is done. Once the aerodynamic derivatives are calculated for the current altitude, initial attack angle  $\alpha$  is searched. Assuming horizontal flight as initial condition, lift coefficient can be evaluated as:

$$(C_L)_0 = \frac{mg}{\frac{1}{2}\rho U_0^2 S}$$

An iterative procedure is conducted, in order to derive the angle of attack  $\alpha$  and aerodynamic derivatives of that flight condition. The angle of attack  $\alpha$  is varied from its minimum value  $\alpha_{min}$  and its maximum value  $\alpha_{MAX}$  with a small increment  $\Delta\alpha$ . An attempt of lift coefficient  $C_{Ltry}$  is guessed as function of attack angle  $\alpha$ .

$$C_{Ltry} = C_X \sin \alpha - C_Z \cos \alpha$$

If the initial lift coefficient value is between two guessed lift coefficients, a linear interpolation is done, else the angle of attack is incremented of  $\Delta\alpha$  and the procedure is repeated. After the angle of attack is established, all the other aerodynamic characteristics are derived. A summary of this procedure is drawn in the following flow chart (see Figure 5.2).

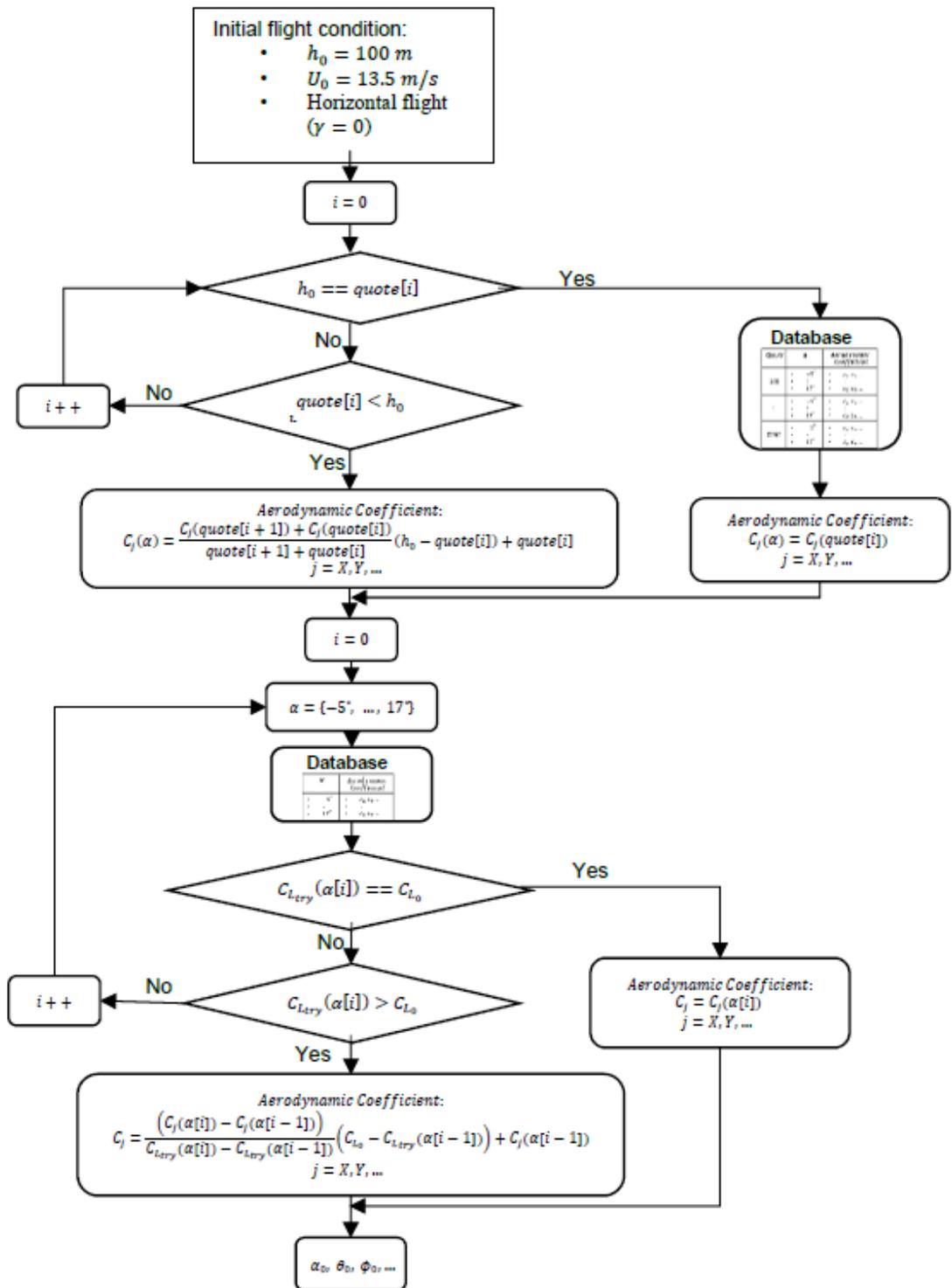


Figure 5.2 - Flow chart of initial condition determination

### 5.1.2 Simulink Model

Simulation takes place in this environment. Several blocks are defined as it can be seen in Figure 5.3. Even if it cannot be seen all these blocks are connected through labels. The solver adopted in this simulation is *ode3* which has a third order accuracy

and implements the Bogacki-Shampine integration technique. The time step is chosen to be fixed and equal to 0.01s.

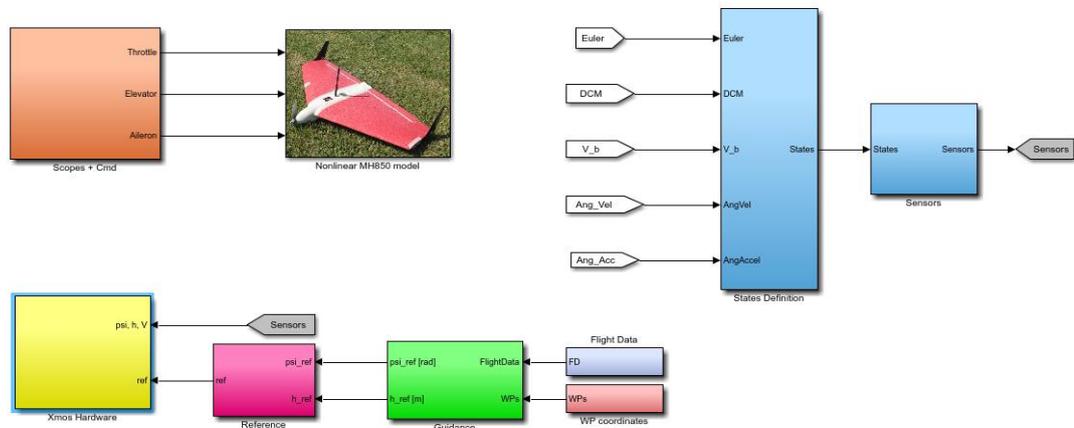


Figure 5.3 - Simulink model of the system

The most important blocks are:

1. The orange box in Figure 5.3 “Scopes+cmd”, in which the two MPC controllers are implemented. The inputs of this block are the outputs of “Guidance” block and represent the reference values of  $u_{ref}$ ,  $h_{ref}$  and  $\psi_{ref}$ , which are necessary to follow the right predefined path. The outputs are the commands, which are elaborated by the MPC controllers and must be employed to maintain a stable flight. Inputs and outputs are generated with a sample time of 0.01s. Only MPC blocks work with a ten times slower sample time (0.1s). This choice was made thinking about a possible future implementation in an embedded system. In this case the processor needs to have enough time to compute the optimization problem at each sampling time.
2. Within “Nonlinear MH850 model”, the masked box in Figure 5.3, are implemented the six degree of freedoms equation of motions. This block is a mathematical representation of the real system and it is solicited at every sampling time with new commands, coming from the “Scopes+cmd” block. The outputs are then reworked in “State definition” block.
3. The “States definition”, the light blue box in Figure 5.3, block computes relevant variables (such as angle of attack, indicated airspeed, etc.) from state data. Within this block equations (1) and (2) are written and conversion among reference frames is done. This is a useful task because in “Guidance” block NED-reference frame is adopted.

4. Within “Guidance” block, the green one in Figure 5.3, the guidance algorithm is implemented. This block ensures that all waypoints are reached and provides the reference signals to the control block. In this block some flag variables are defined such as flags which indicates toward which waypoint the UAV is aiming or which maneuver must be done in order to maintain the UAV in the ideal trajectory. The guidance algorithm adopted in this study is described in details in [15], in which some simplifying hypotheses are made to take into account the flash memory limitation of the on-board microcontroller. A given set of waypoints is considered, with assigned North, East and altitude coordinates. This set of waypoints includes the starting point, which is the point where the UAV finishes its climb and the autonomous flight starts. The starting point and all waypoints are assumed to be at the same altitude; thus, a 2D path is considered. A trajectory smoother, that renders the assigned trajectory kinematically feasible in terms of speed and turn rate constraints, is implemented.

## 5.2 Model Predictive Control simulation results

Three tests are conducted, to verify the flight performance. In each test a different path and a different initial position is given. Initial airspeed is  $U_0 = 13.5 \text{ m/s}$  and initial altitude is  $h_0 = 100 \text{ m}$ . Absence of wind or external disturbance is assumed. In the following tests the initial UAV orientation is the same of the arrow that connect the first waypoint with the second one. This means that  $X_B$  axis at the beginning of simulation has the same direction of the straight line which connects the first waypoint with the second one.

The following results are obtained through the simulation and tuning parameters, collected in Table 1.

**Table 1** - MPC simulation and tuning parameters

<b>Parameters</b>	<b>Value</b>
System sample time [s]	0.01
MPC sample time [s]	0.1
Prediction horizon ( $N_p$ )	10
$diag(Q_{lon})$	[2, 120, 10, 1, 0.01]
$diag(R_{lon})$	[0.07, 0.02 ]

$diag(Q_{lat})$	[1, 100, 10, 120]
$R_{lat}$	$10^4$

### 5.2.1 Diamond path

The first test is conducted with five waypoints. The first one coincides with initial position and the others are placed in four imaginary diamond vertices. The simulation is conducted for 200s, which is enough to cover more than one lap. The ideal path, which the UAV should follow is shown in Figure 5.4.

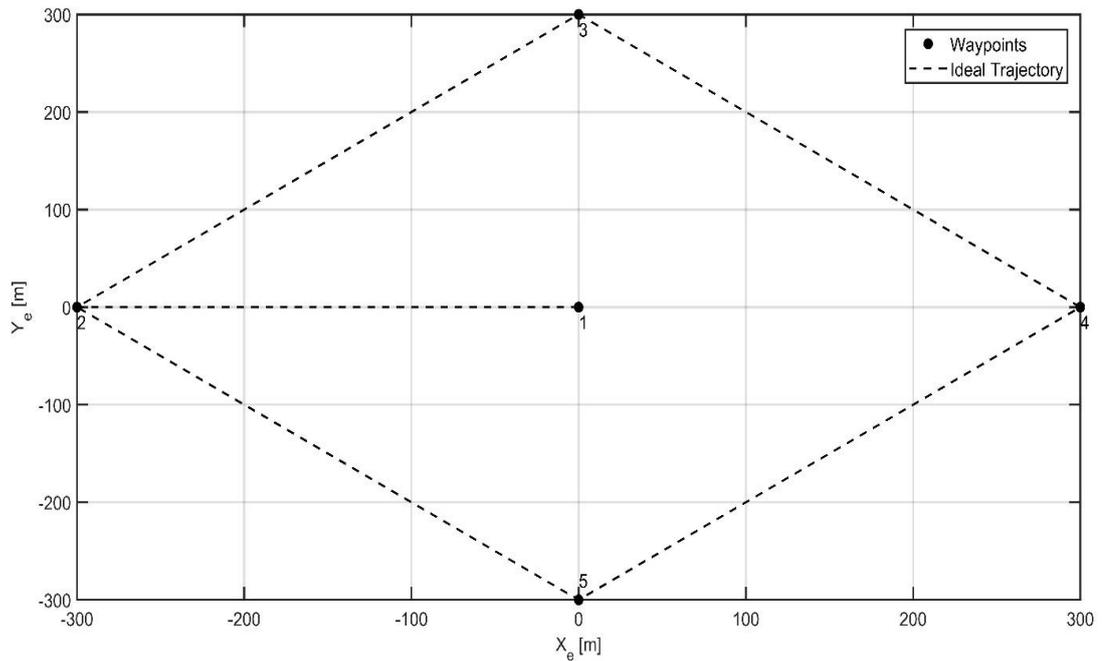
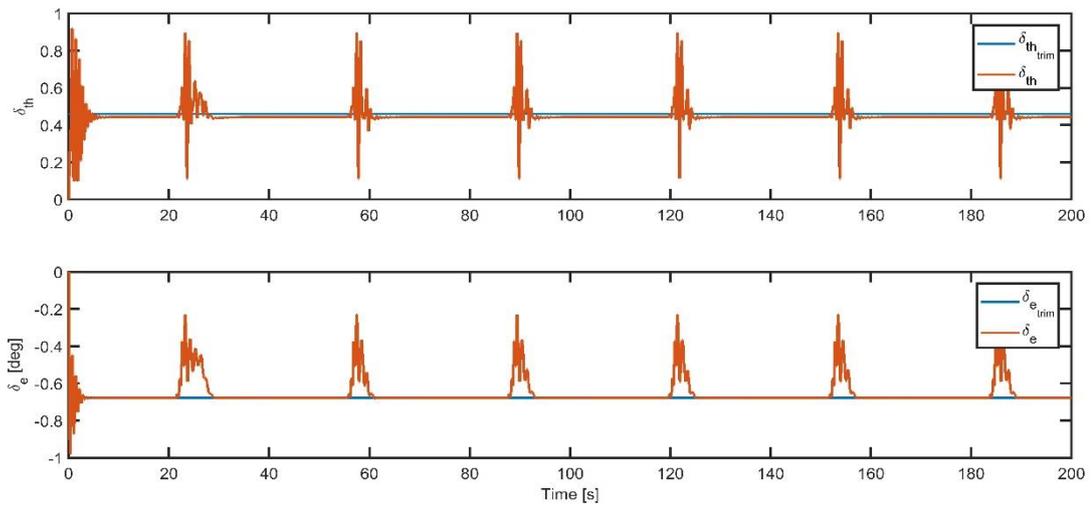


Figure 5.4 - Ideal trajectory diamond path

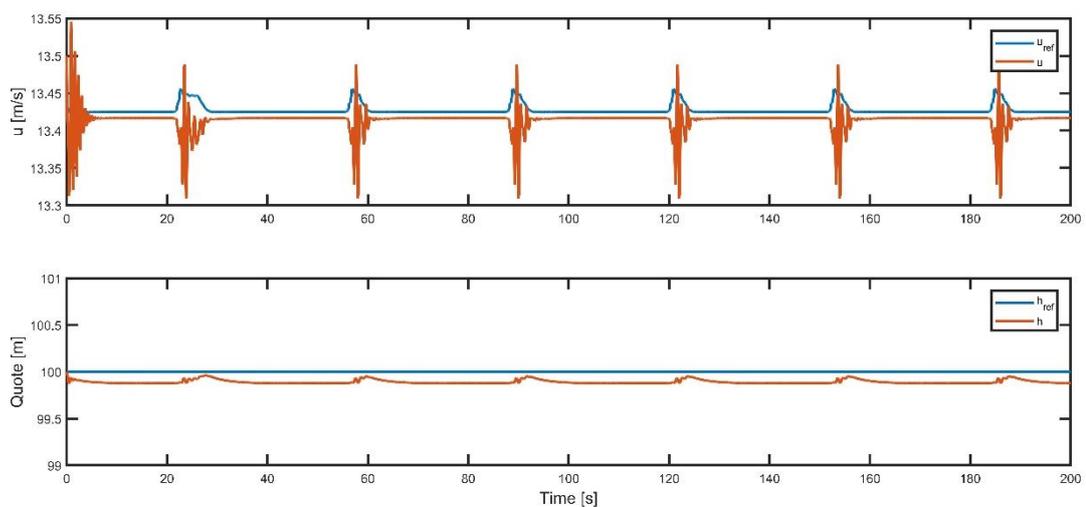
Firstly, it is considered the longitudinal plane, secondly the lateral-directional plane and finally the overall trajectory.

In longitudinal plane there are two inputs, throttle  $\delta_{th}$  and elevon deflection  $\delta_e$ , and five state variables, longitudinal velocity  $u$ , attack angle  $\alpha$ , pitch angle  $\theta$ , pitch rate  $q$  and altitude  $h$ , in the dynamic systems. Reference signals, generated directly by the guidance algorithm, are explicitly provided for longitudinal speed and for altitude. For the other variables the reference value is implicitly zero. The inputs assigned during the maneuver are in Figure 5.5. The blue curve represents the value of the input variable in trim condition, the orange one is the actual value.



**Figure 5.5** - Longitudinal plane inputs time history

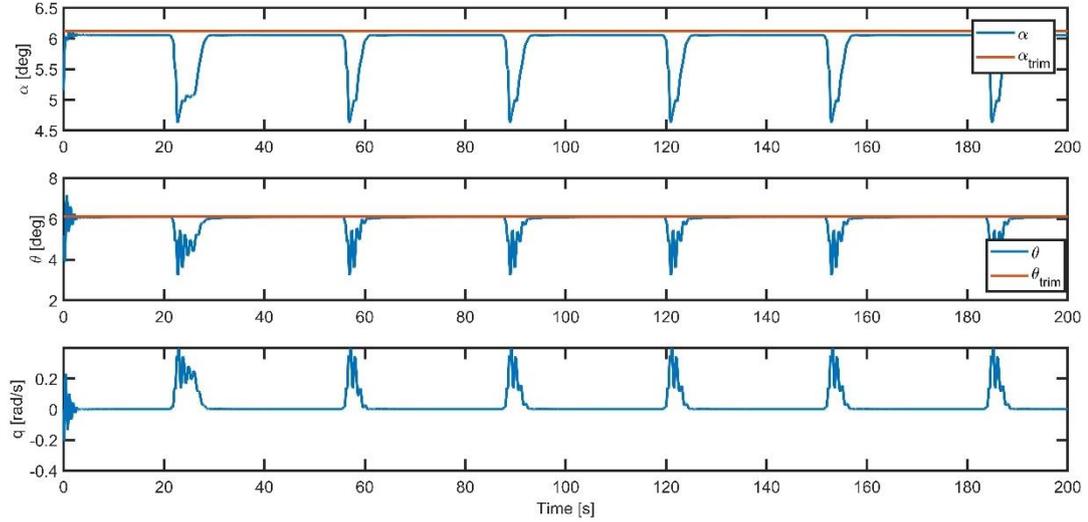
As it can be seen, there are fluctuations in the first seconds of simulation, because the Model Predictive Controller is looking for the equilibrium condition. In less than ten seconds the trim condition is reached. These oscillations could cause many problems in a real system and usually inputs are forced to maintain the trim values for the first instants. In this case they are due to the nonlinearities contained in “Nonlinear MH850 Model”. The elevator oscillation amplitude is limited to a maximum value of about  $-0.2^\circ$ , whereas the throttle one is quite large and reaches almost the 100% of the throttle range. For the remaining simulation time there are some periodic amplitude variations, which correspond to right turning, once a waypoint is reached. The most important state variables in longitudinal plane are longitudinal velocity and altitude, for which a reference tracking is done, as in Figure 5.6.



**Figure 5.6** – Longitudinal airspeed and altitude time history

It can be noticed that these variables are subjected to some small oscillations as the inputs. The reference tracking is acceptable for the altitude, whereas it is a little bit

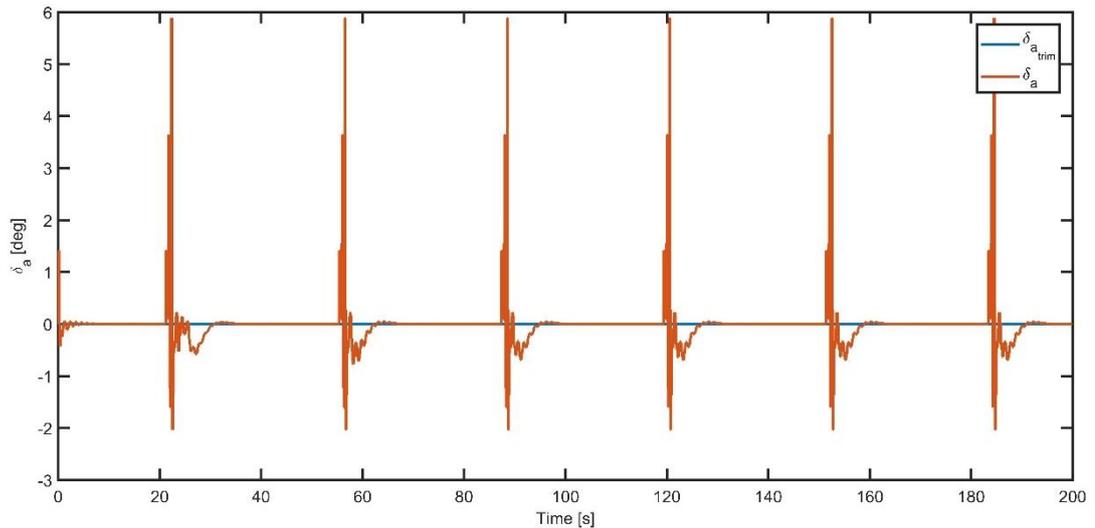
inaccurate for the longitudinal velocity. During the maneuvers a higher airspeed is demanded, but the actual longitudinal airspeed is subjected to some oscillations. Anyway, after every turning the values come back to the original ones. The remaining longitudinal state variables are presented in Figure 5.7.



**Figure 5.7** - Longitudinal state variables time history

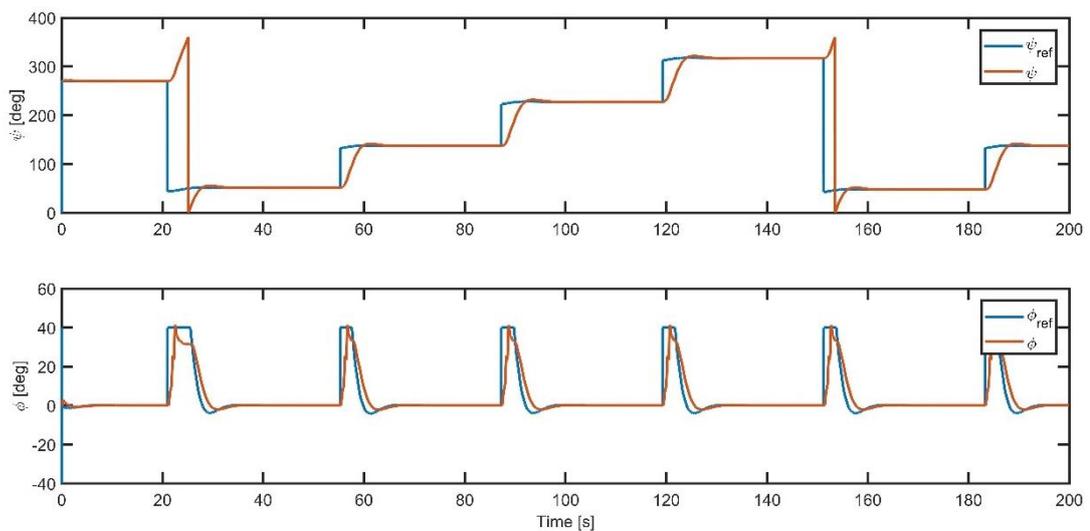
Angle of attack  $\alpha$  and pitch angle  $\theta$  run quite quickly after the respective trim values, whereas pitch rate  $q$  is almost always null. The fact that angle of attack  $\alpha$  decreases and the height increases during every turning can be surprising. If a positive pitch rate is imposed during this maneuvers, a vertical airspeed is generated and this induces a decreasing on the angle of attack from the point of view of the wing leading edge.

For lateral-directional plane only the inner loop is controlled by a MPC law. The outer navigation loop is regulated by a PID controller, which has as input the heading orientation error and produces as output the reference roll angle  $\phi_{ref}$ . The control in this plane is not a trivial problem for this UAV, because there does not exist a rudder or a vertical surface to control the yaw angle  $\psi$  and rate  $r$ . Indeed, the aileron surfaces have the task to control also the rotation and rotational speed around  $Z_B$  axis. The aileron deflection is shown in Figure 5.8.



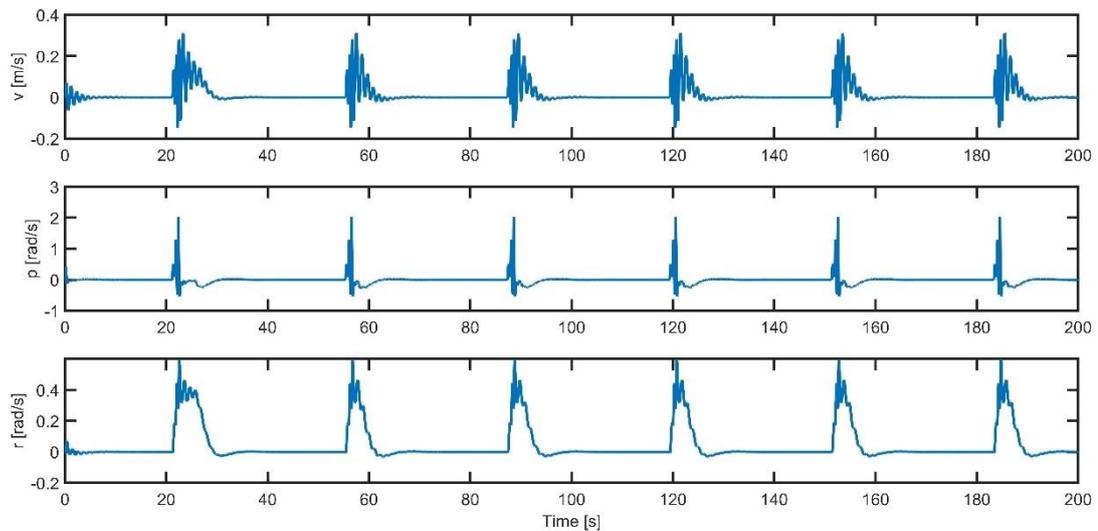
**Figure 5.8** - Aileron time history

Because the initial condition is a horizontal straight flight, it follows that the aileron trim value  $\delta_{a_{trim}}$  is null. In this graph it can be seen quite well that the commands are constants for 0.1s. The ailerons are deflected every time the UAV needs to turn. Checked maneuvers are performed, so after the first positive deflection other corrective maneuvers are necessary. Ailerons are almost always activated because they have to accomplish also rudder tasks. The most important state variables controlled in this plane are roll angle  $\phi$  and yaw angle  $\psi$ , whose dynamic response to this input is shown in Figure 5.9. The reference value  $\psi_{ref}$  for yaw angle is provided directly by guidance algorithm and the reference signal  $\phi_{ref}$  is derived by a PID controller from the discrepancy between the actual value  $\psi$  and the reference one.



**Figure 5.9** - Heading and roll angle time history

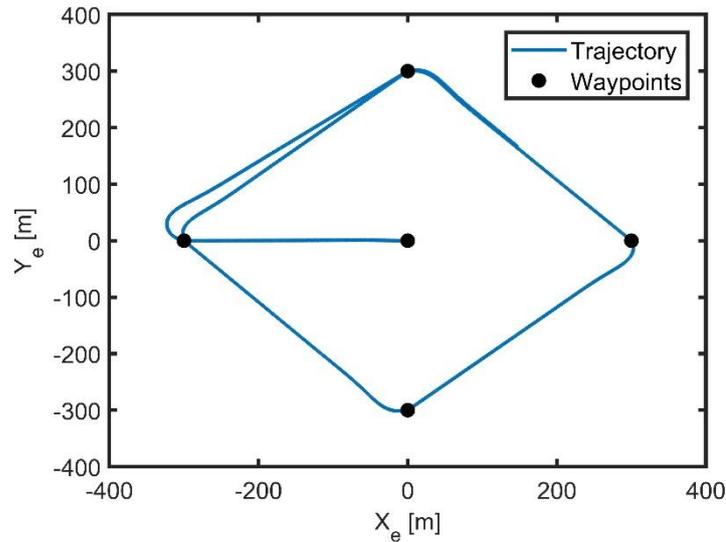
The yaw angle  $\psi$  runs after the reference value  $\psi_{ref}$  quite slowly, but after a small overshoot it settles precisely to the reference value. When the vehicle must turn, a very high roll angle reference  $\phi_{ref}$  is given and the saturation constraint is activated. During the maneuvers, the actual roll angle  $\phi$  has a small overshoot, with which it reaches the saturated reference value, and then settles on a smaller value than the reference one. However, when the maneuvers are completed, the roll angle follows precisely the reference. In Figure 5.10 the other state variables of lateral-directional plane are included.



**Figure 5.10** - Lateral-directional plane variable states time history

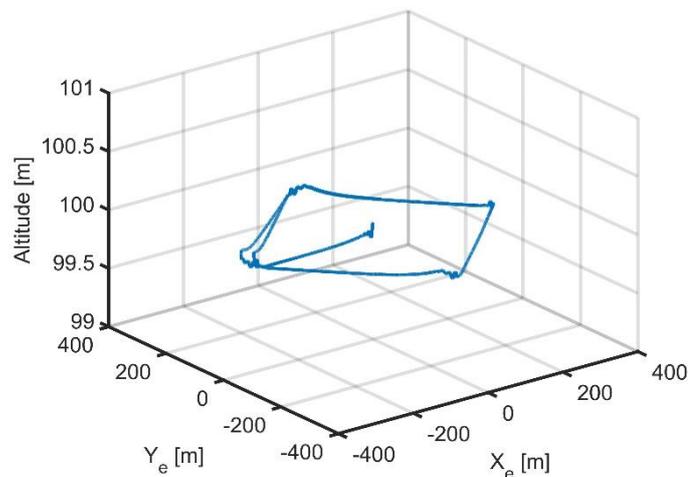
In every turning the UAV rotates in positive sense around axis  $X_B$  and  $Z_B$ . The aileron deflection affects firstly roll angular speed  $p$ , which has almost the same evolution of the aileron. The first command has a strong effect on roll angular speed  $p$  and yaw angular speed  $r$ . The corrective control restores the condition of horizontal straight flight.

The trajectory on the North-East plane is shown in Figure 5.11. Black dots identify the five waypoints.



**Figure 5.11** - Diamond path

In the first lap the first turning is accomplished in a larger space compared to the other maneuvers because the heading variation in the first maneuver is bigger than in the others. Once the UAV has completed a lap, the trajectories of the following laps coincide. A three-dimensional trajectory representation is made in Figure 5.12.



**Figure 5.12** - UAV 3D trajectory

Small altitude oscillations can be seen at the beginning of the simulation and every time a maneuver is undertaken.

### 5.2.2 Octagonal path

The second path is chosen to be octagonal. The UAV departs from the first waypoint and then completes a lap following the remaining waypoints. Once it comes to the ninth waypoint, the guidance algorithm provides the reference signals to reach

the second waypoint and to begin a new lap. This path and the corresponding waypoints are shown in Figure 5.13.

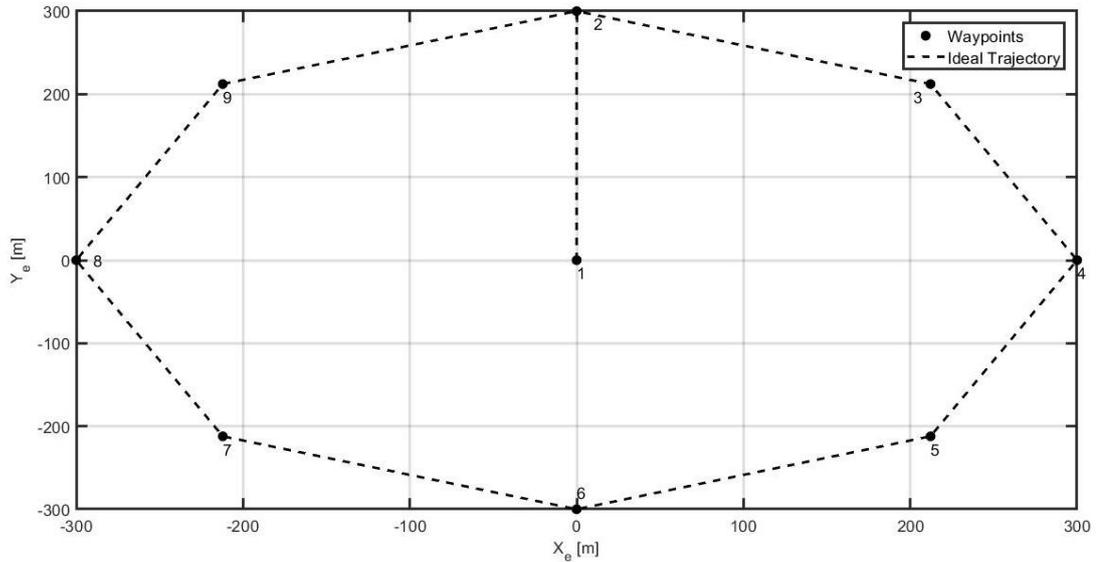


Figure 5.13 - Octagonal ideal path

The simulation is stopped after 200s, which is enough time to complete one lap. To cover this path, the longitudinal inputs are shown in Figure 5.14.

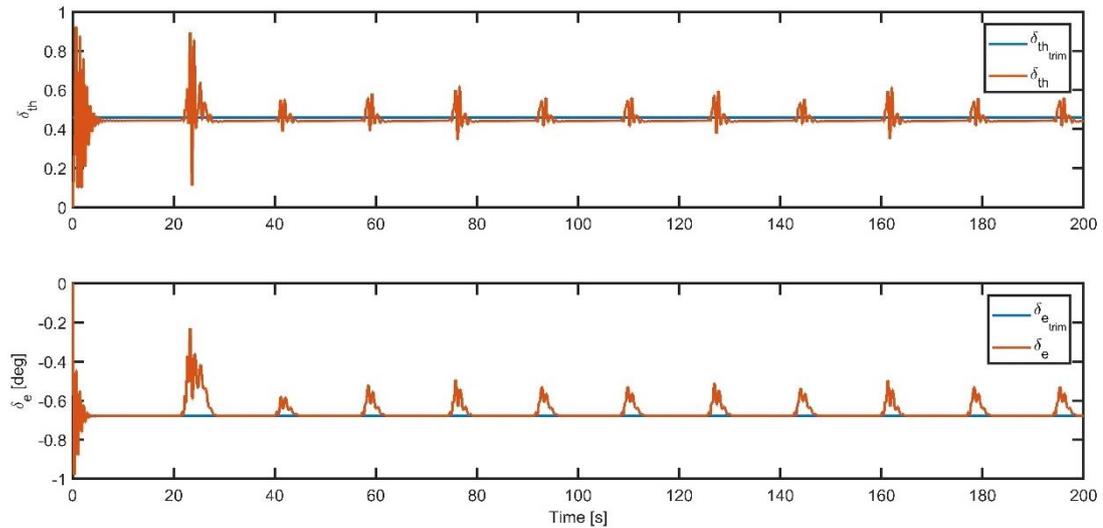
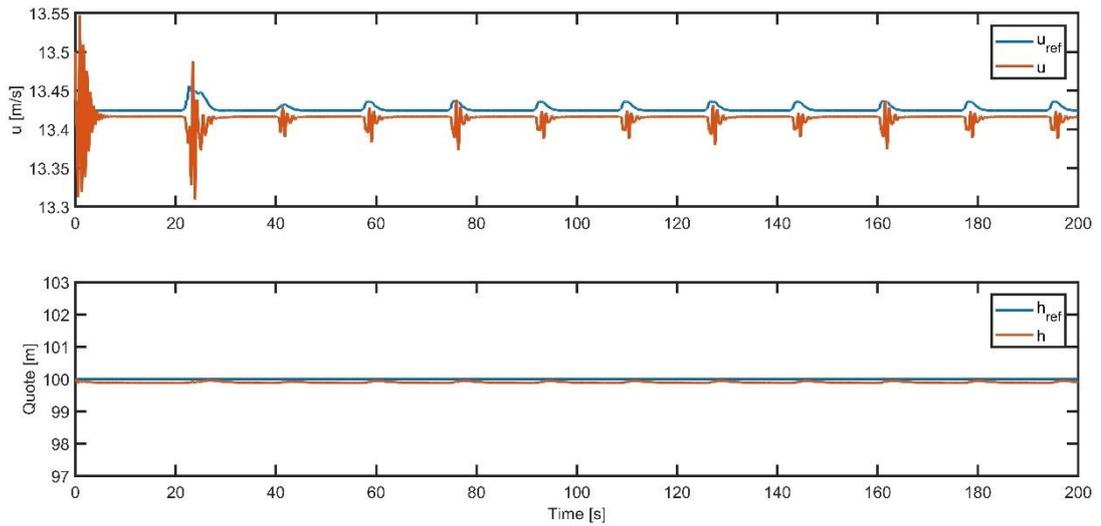


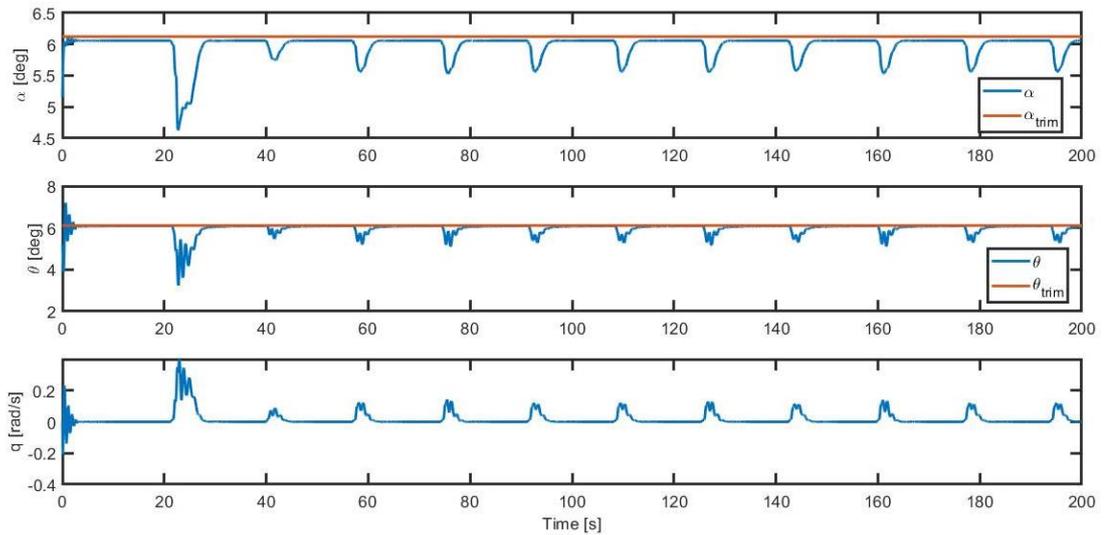
Figure 5.14 - Longitudinal inputs

As it can be seen, the constraints are respected with a certain margin for both throttle  $\delta_{th}$  and elevon  $\delta_e$ . Smaller oscillations than the previous case occur in both control variables during the maneuvers, because a smaller variation of heading is requested and smoother turn are undertaken. The longitudinal airspeed  $u$  and the altitude response are plotted in Figure 5.15.



**Figure 5.15** - Longitudinal airspeed and altitude response

In this case the reference tracking is precise and the variables do not deviate never significantly to the reference values. The other longitudinal states are reported in Figure 5.16.



**Figure 5.16** - Longitudinal states time history

As it can be seen, the maneuvers cause small variation in longitudinal states, whose evolution is widely within the constraints. The aileron evolves as it is shown in Figure 5.17.

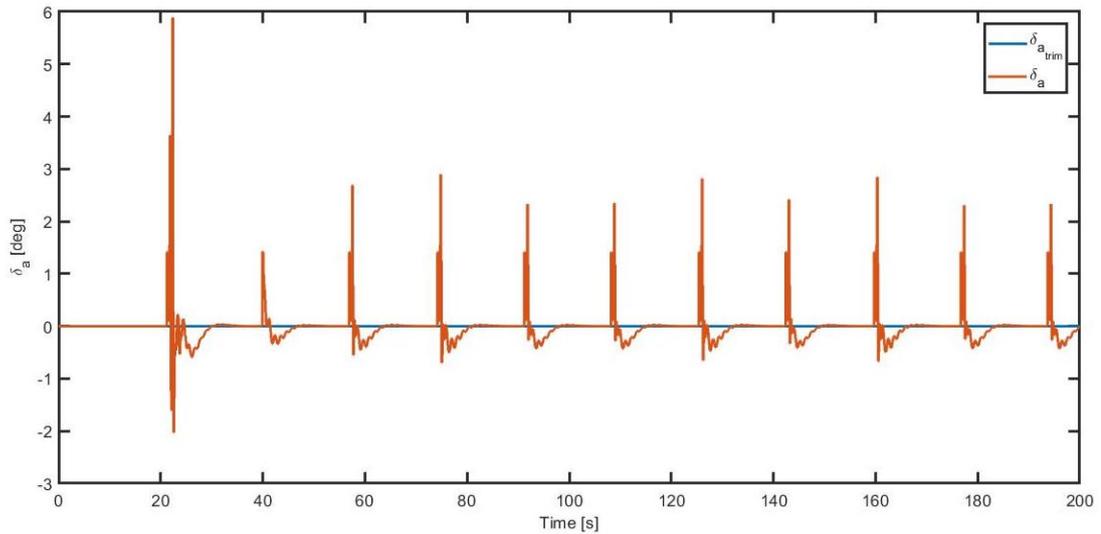


Figure 5.17 - Aileron time history

As it can be noticed, the first turn requires a heading variation bigger than the others, hence the aileron deflection is larger for the first maneuver. In Figure 5.18 the heading angle  $\psi$  and the roll angle  $\phi$  evolution as consequence of the aileron deflection is shown.

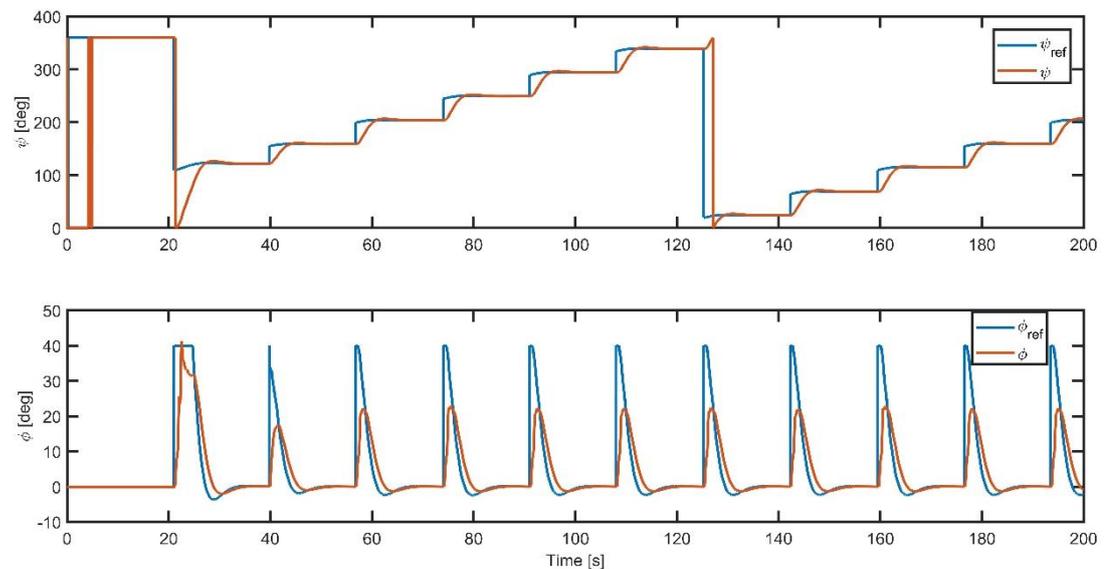
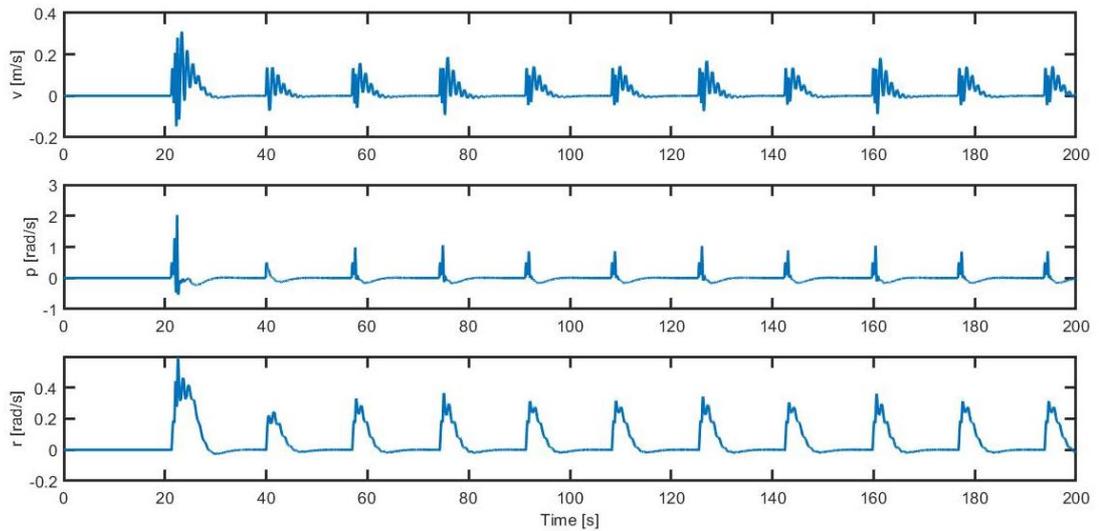


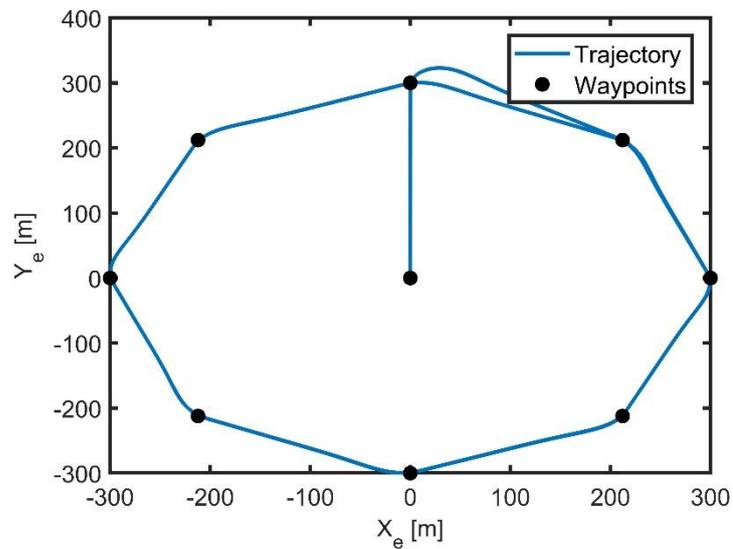
Figure 5.18 - heading and roll angle time history

The reference tracking is precise for the heading angle, whereas for the roll angle a small delay in running after the reference value is there. In this latter case the saturation constraint is activated during every turning. In the first maneuver the roll angle has the time to reach saturated reference value, whereas in the following turning the reference is changed rapidly and cannot be reached by the actual roll angle peak. For the straight flight between two turns the reference tracking is satisfactory and precise. In Figure 5.19 the other lateral-directional states are reported.



**Figure 5.19** - lateral-directional states time history

Lateral-directional airspeed reveals some oscillations during the maneuvers, which are damped to a null value in the straight flight. All these control and state response are related to the trajectory shown in Figure 5.20.



**Figure 5.20** - Octagonal two-dimensional trajectory

Once the first lap is covered and the third waypoint is reached, the UAV follows almost the same trajectory of the first lap. In the trajectory is plotted in a three-dimensional space.

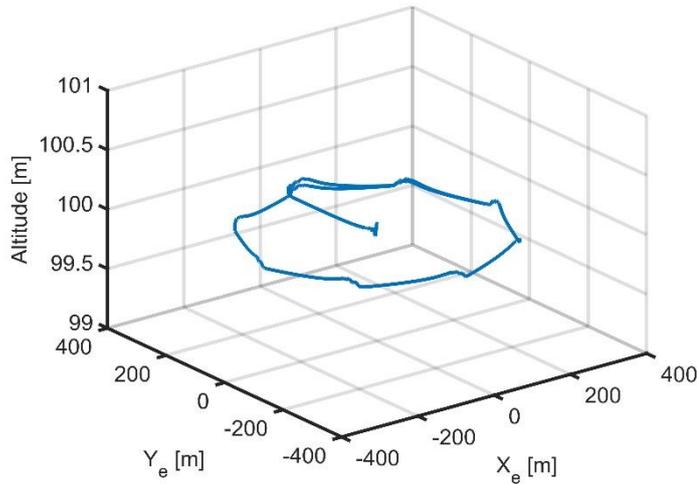


Figure 5.21 - Octagonal 3D trajectory

### 5.2.3 “Snake” path

The third chosen path could be considered the most realistic one, because it represents a possible flight mission to accomplish to surveillance or monitoring tasks.

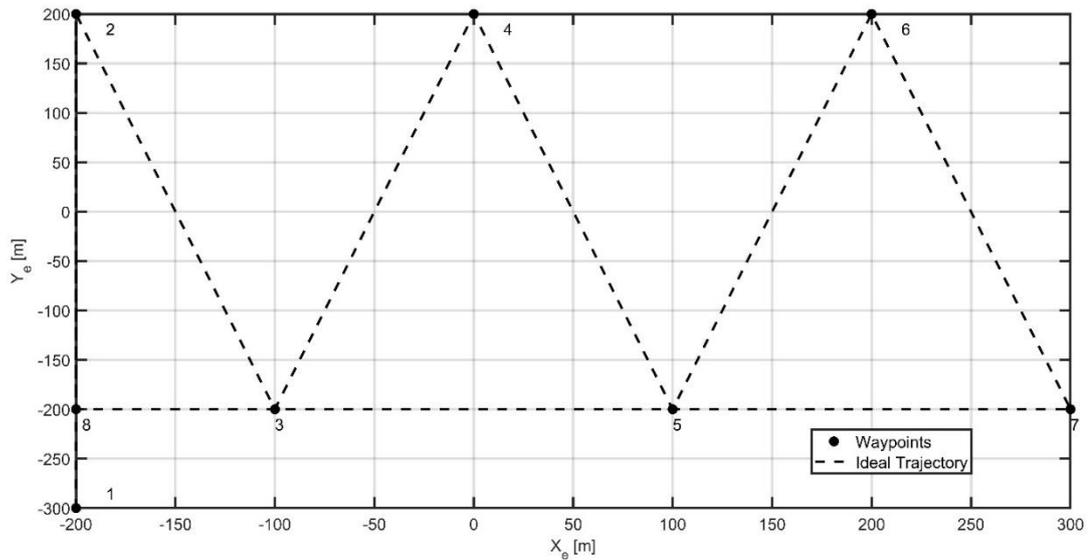
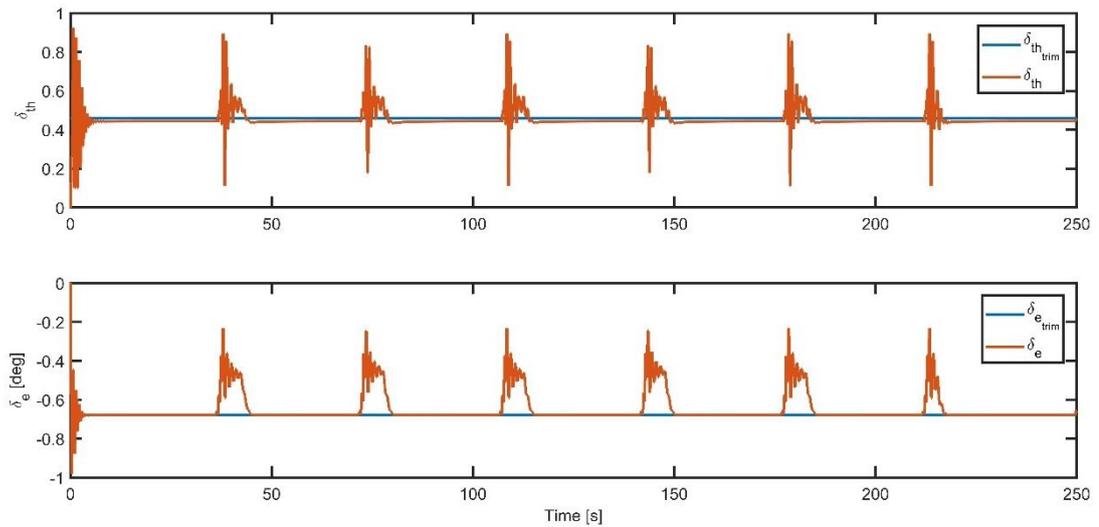


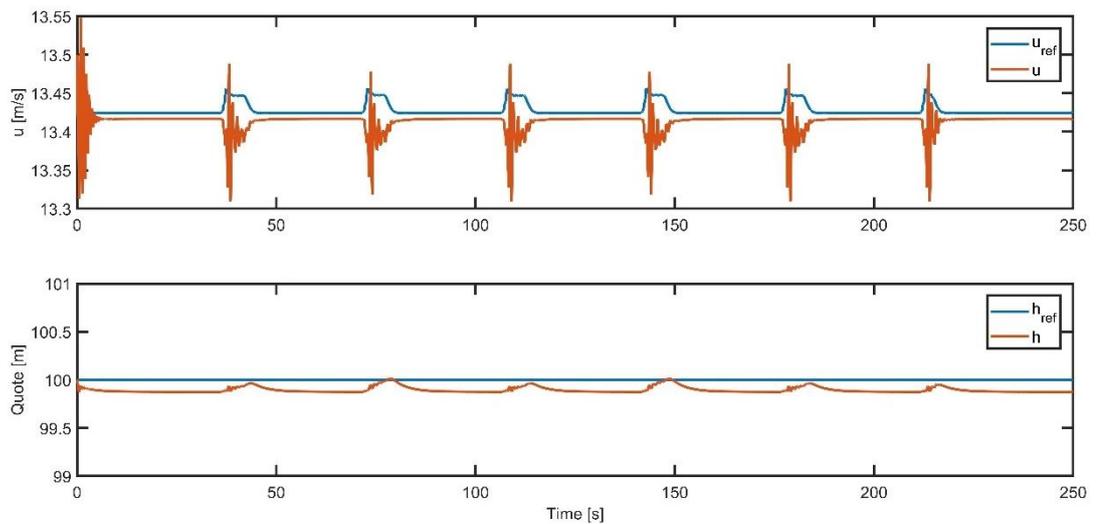
Figure 5.22 - Ideal “snake path”

Once the waypoint 8 is reached, the UAV aims again towards the waypoint 1. In this case the UAV initial position is not the origin of the reference system, but it coincides with point with coordinates  $(-300\text{ m}, -200\text{ m})$ . Simulation time is set to be  $250\text{ s}$ . It is interesting to notice that in this path some left turns are there. The longitudinal inputs are plotted in Figure 5.23.



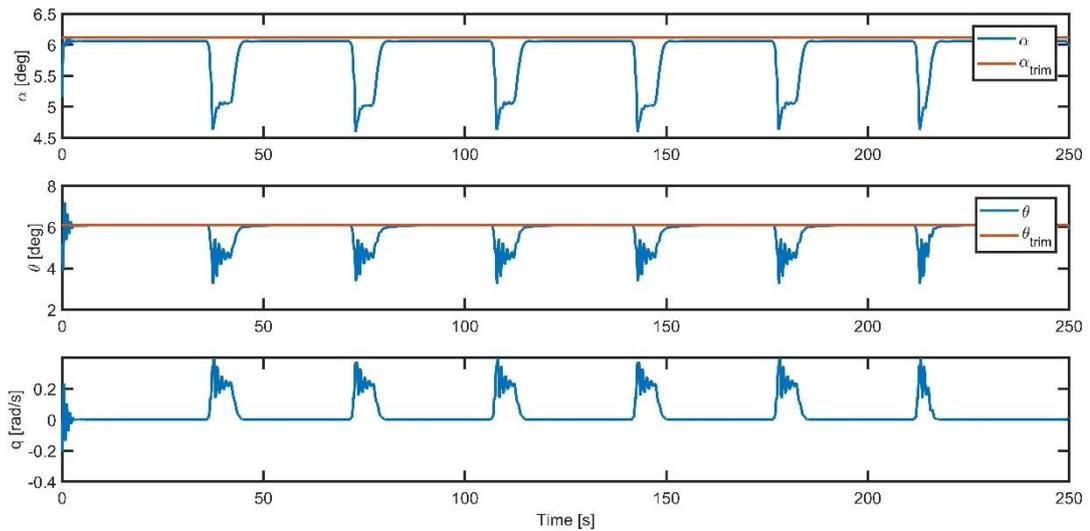
**Figure 5.23** - Longitudinal inputs time evolution

As it can be expected, even if the turns are not all in the same direction, the longitudinal input is the same. Because the turns cause a bigger variation in heading angle, the corrective actions of the longitudinal inputs are wider. The longitudinal system response to these control actions are plotted in Figure 5.24.



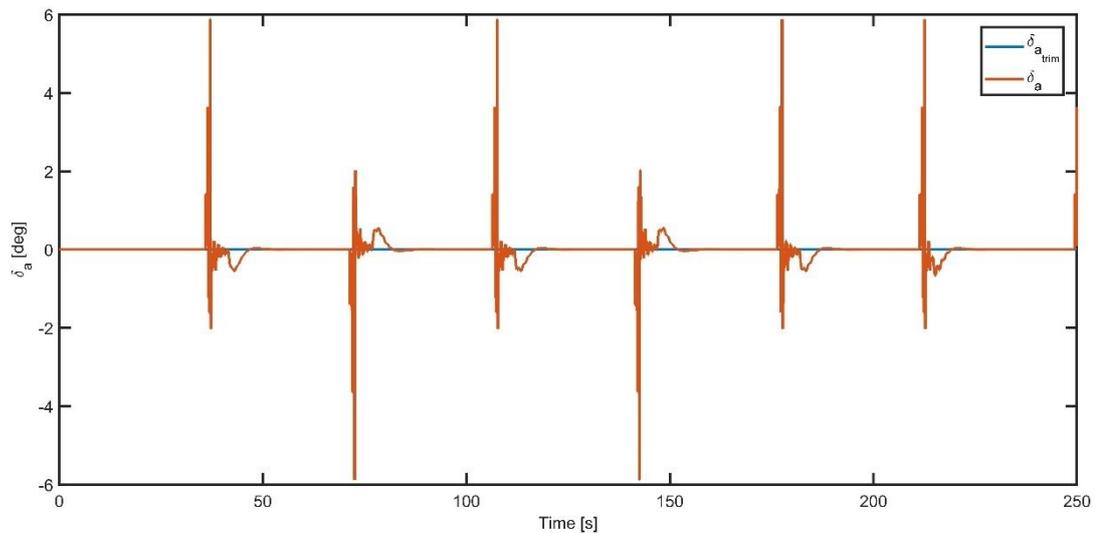
**Figure 5.24** - Longitudinal airspeed and altitude time history

The reference tracking is satisfactory for the straight flight, whereas the longitudinal airspeed averagely decreases during the maneuvers, because a small increasing in altitude occurs. In Figure 5.25 the other longitudinal states are shown.



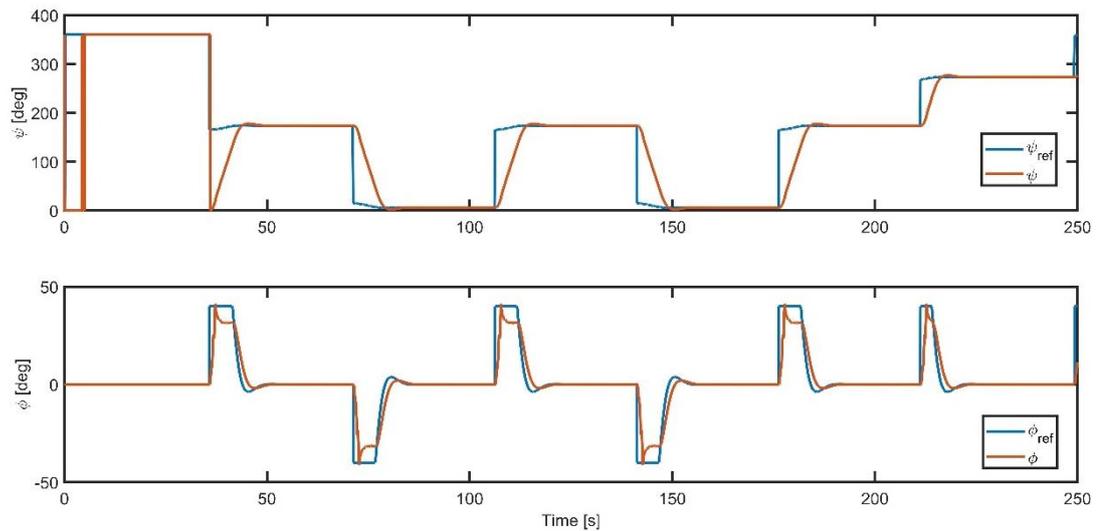
**Figure 5.25** - Longitudinal states time history

The variables follow well the respective reference values and, as it can be envisaged, their variations during the maneuvers is always in the same direction and with the same amplitude. The aileron deflection assigned for the lateral-directional plane is shown in Figure 5.26.



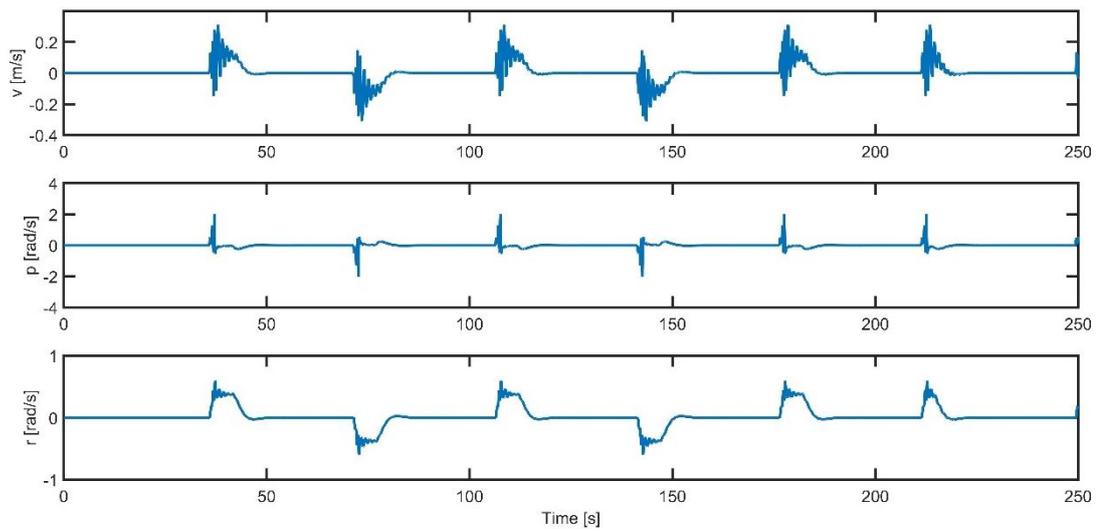
**Figure 5.26** - Aileron time history

In this case the first four maneuvers are applied in alternate direction. They have the same amplitude modulus and cause oscillations with the same amplitude modulus. In Figure 5.27 it is shown the effect that this input has to the heading angle and to the yaw angle.



**Figure 5.27** - Heading and yaw angle time history

The yaw reference tracking is precise for straight flight, whereas during the maneuvers it shows some overshoot, which almost reaches the saturation constraint. It can be also noticed a small delay, while the reference changes. In Figure 5.28 the other later-directional states are plotted.



**Figure 5.28** - Lateral-directional states time history

The two-dimensional trajectory is plotted in Figure 5.29.

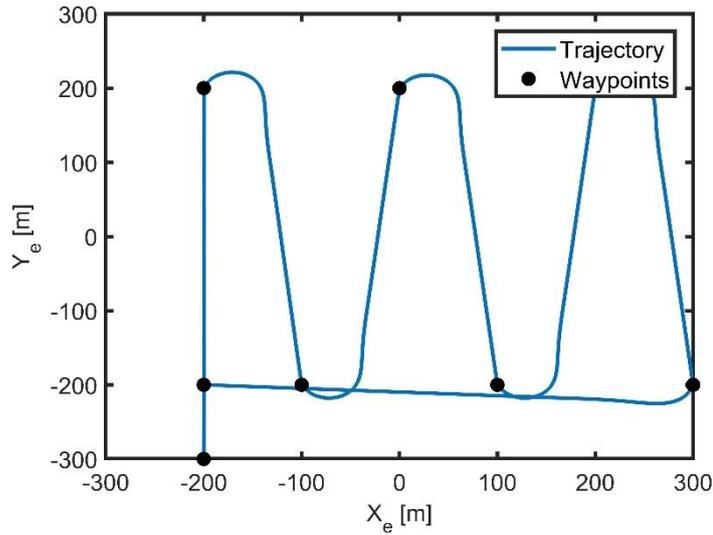


Figure 5.29 - Two-dimensional “Snake” trajectory

Finally, the three-dimensional “snake” trajectory is represented in Figure 5.30.

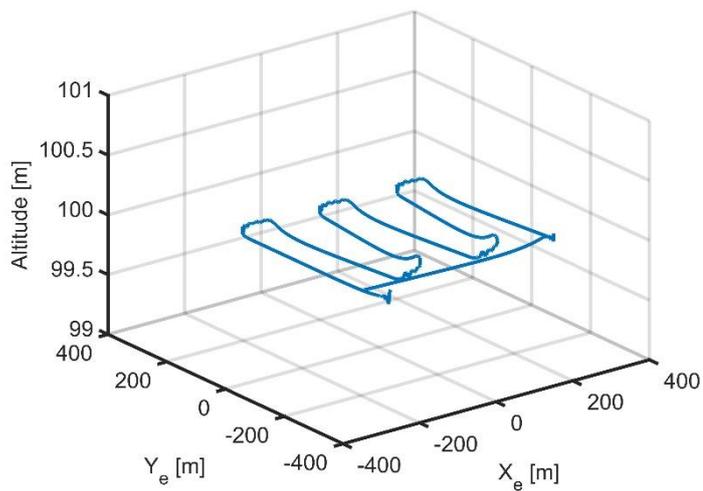


Figure 5.30 - 3D “snake” trajectory

### 5.3 Tube-based Model Predictive Control simulation results

Before to present the simulation results, it is important to understand which changes have been made in the model to extend the Model Predictive Control to a Tube-based Model Predictive Control. As described in Chapter Chapter 4:, a conventional Model Predictive Control is applied to the nominal linear system, which runs in parallel to the disturbed UAV nonlinear model. The MPC elaborates the nominal system inputs, from which the inputs for the disturbed system are derived in relation to the discrepancy between disturbed states and nominal states. A scheme of this algorithm is shown in Figure 5.31, where the disturbed state  $x$  is the Tube-based

MPC input, the nominal state  $z$  is the MPC input, the nominal control  $v$  is the MPC output and the uncertain system control  $u$  is the Tube-based MPC output.

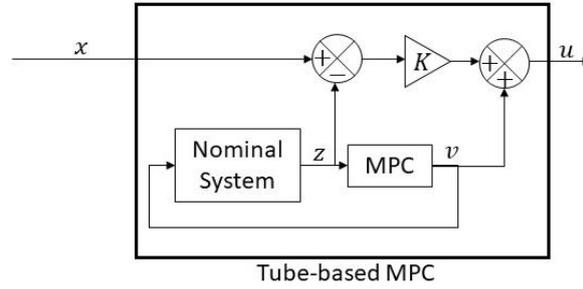


Figure 5.31 - Tube-based MPC scheme

Nominal system and nonlinear disturbed system are initialized with the same states, hence the initial error is imposed to be null and no corrections are applied to the nominal control  $v$ , which is also equal to the disturbed state control  $u$  at the beginning of simulation. Moreover, only the Model Predictive Control block work at a sample time of 0.1s, whereas the nominal system evolution and the correction of the control action are derived every 0.01s.

The feedback gain matrix  $K$  is evaluated through the procedure described in section 4.2.2, in order to make the matrix  $A_K = A + BK$  Hurwitz. Small uncertainties are introduced in the system, in order to build an LMI problem. 3% uncertainty for airspeed  $V$  and 1% for mass are adopted. After solving two LMIs system for longitudinal and for lateral-directional plane, two feedback gain matrices are derived for both planes and the resulting matrix  $A_K$  eigenvalues are reported in Table 2.

Table 2 - matrices  $A_{K_{lon}}$  and  $A_{K_{lat}}$  eigenvalues

Matrix	Eigenvalues	Eigenvalues modulus
$A_{K_{lon}} = A_{lon} + B_{lon}K_{TBMPC_{lon}}$	$\lambda_1 = 0.7478$ $\lambda_2 = 0.8805$ $\lambda_3 = 0.9331$ $\lambda_4 = 0.9790 + 0.0196i$ $\lambda_5 = 0.9790 - 0.0196i$	$ \lambda_1  = 0.7478$ $ \lambda_2  = 0.8805$ $ \lambda_3  = 0.9331$ $ \lambda_4  = 0.9792$ $ \lambda_5  = 0.9792$
$A_{K_{lat}} = A_{lat} + B_{lat}K_{TBMPC_{lat}}$	$\lambda_1 = 0.7754 + 0.5229i$ $\lambda_2 = 0.7754 - 0.5229i$ $\lambda_3 = 0.9502 + 0.0877i$ $\lambda_4 = 0.9502 - 0.0877i$	$ \lambda_1  = 0.9352$ $ \lambda_2  = 0.9352$ $ \lambda_3  = 0.9542$ $ \lambda_4  = 0.9542$

Since we are dealing with discrete time state space system, it follows that both longitudinal and lateral-directional dynamics are now asymptotically stable thanks to the gain feedback matrix  $K$ . Indeed, matrices  $A_{K_{lat}}$  and  $A_{K_{lon}}$  eigenvalues are strictly inside the unit circle in the discrete complex plane (see section 2.4.3).

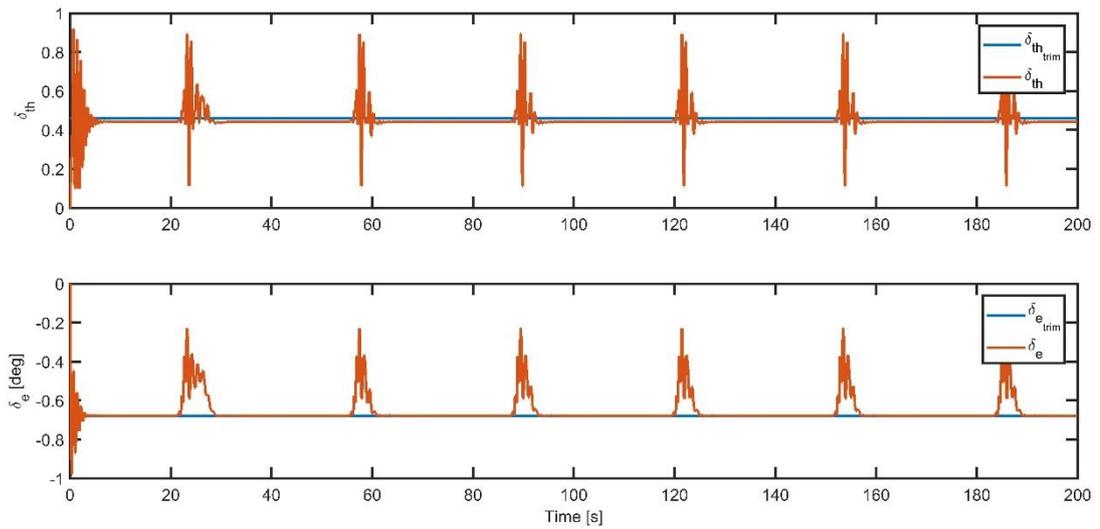
Two nominal systems and two Model Predictive Controller are defined for longitudinal plane and for lateral-directional plane. Atmospheric turbulence is modeled as additive disturbances. New tuning parameters have been chosen, in order to guarantee a good reference tracking also in presence of additive disturbance (see Table 3).

**Table 3** - TB MPC simulation and tuning parameters

<b>Parameters</b>	<b>Value</b>
System sample time [s]	0.01
MPC sample time [s]	0.1
Prediction horizon ( $N_p$ )	10
$diag(Q_{lon})$	[100, 2500, 2000, 50, 100]
$diag(R_{lon})$	[80, 200]
$diag(Q_{lat})$	[50, $10^3$ , 10, 4700]
$R_{lat}$	50
$[w_u w_\alpha w_\theta w_q w_h]$	$[10^{-2} 10^{-6} 10^{-6} 10^{-6} 10^{-3}]$
$[w_v w_p w_r w_\phi]$	$[10^{-2} 10^{-6} 10^{-6} 10^{-6}]$

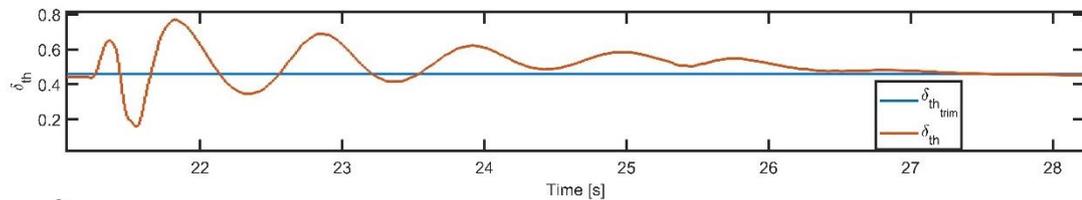
### 5.3.1 Diamond path

The first test in presence of disturbance is conducted on the same path and waypoints presented in Figure 5.4. As in the previous case, firstly longitudinal results are presented and then lateral-directional. Finally, the actual trajectory in two and three dimensions is shown. The longitudinal inputs are plotted in Figure 5.32.



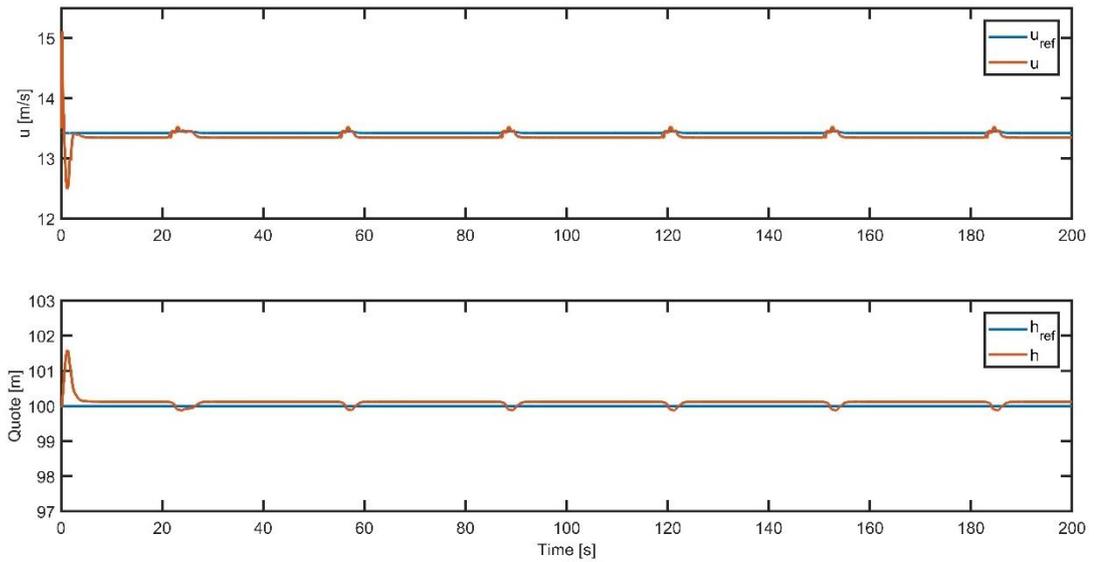
**Figure 5.32** - Longitudinal inputs

For the elevon response some oscillations occur in the first five seconds of simulation, but they are acceptably within the imposed constraints. As it can be seen, there are throttle oscillations in the first instants of simulation. After less 5 seconds, the controller has found the trim condition, which is followed well by the actual throttle. Moreover, the throttle constraints are respected during the rest of the simulation. In Figure 5.33 it is shown a detail of throttle evolution during the first turn, which is the most critical one.



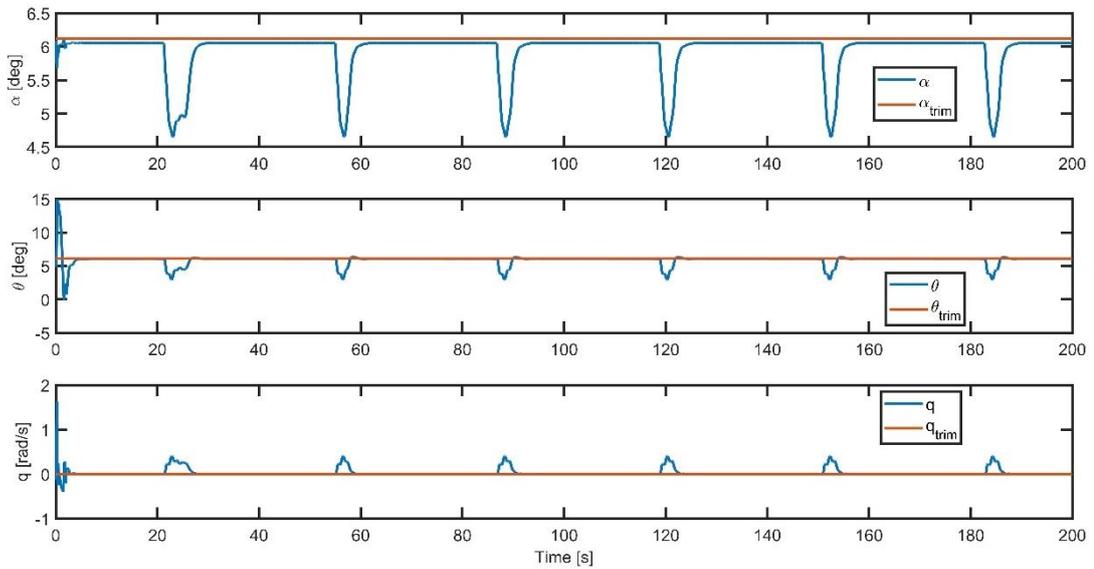
**Figure 5.33** - Throttle evolution during the first turn

The longitudinal airspeed and the altitude responses to these inputs are shown in Figure 5.34.



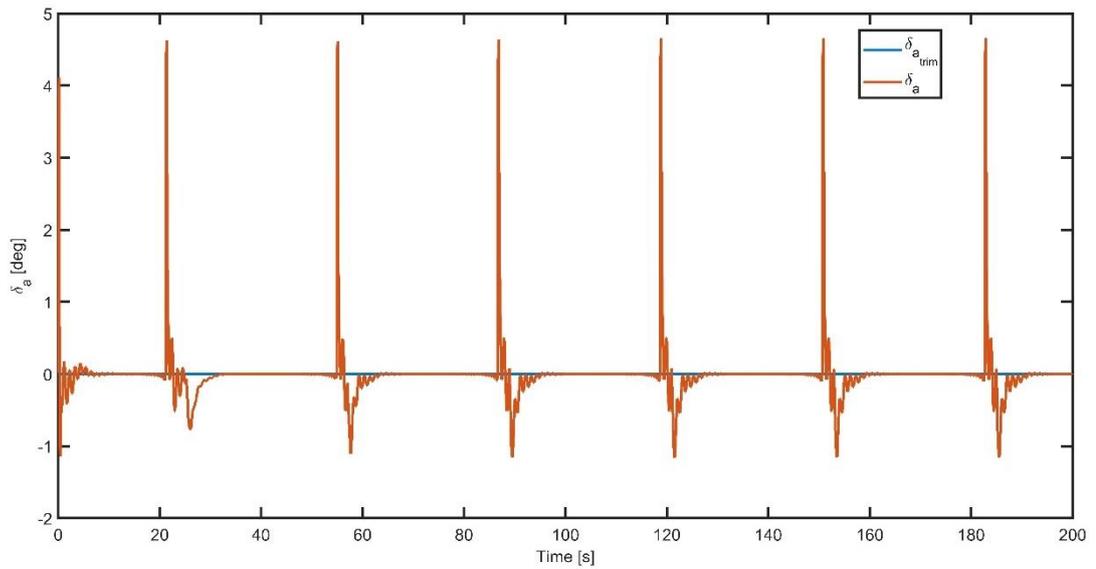
**Figure 5.34** - Longitudinal airspeed and altitude

The reference tracking is good for both longitudinal airspeed and altitude. In this case an altitude decreasing appears during every right turn. Consequently, the longitudinal airspeed increases during the maneuvers. The other state variables are shown in Figure 5.35.



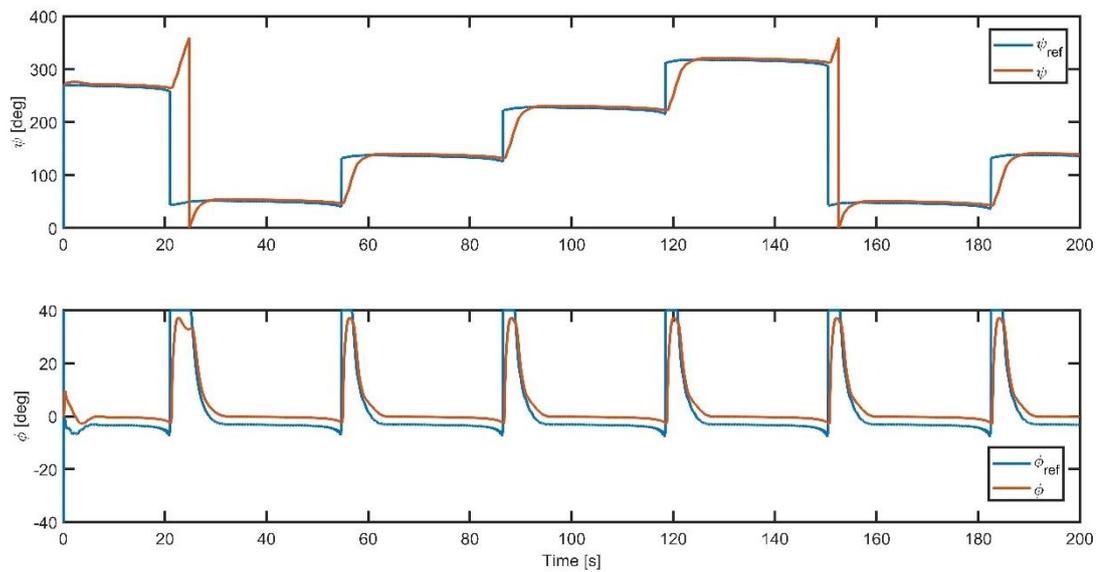
**Figure 5.35** - Longitudinal states evolution

As it can be noticed, the angle of attack  $\alpha$  and the pitch angle  $\theta$  run quickly after their respective trim value, whereas the pitch rate  $q$  is nonnull only during the maneuvers. The lateral-directional input is shown in Figure 5.36.



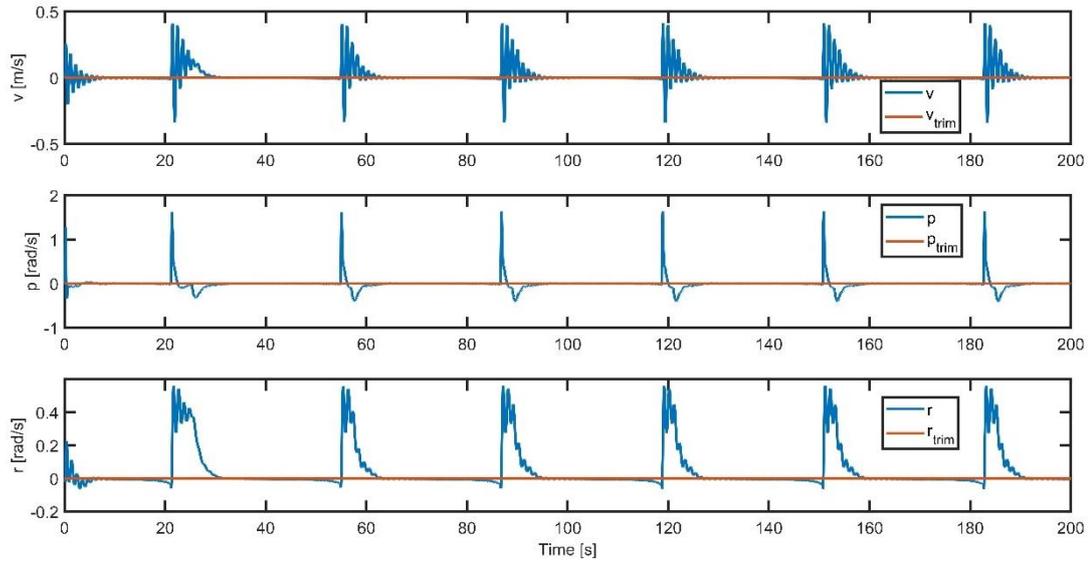
**Figure 5.36** - Aileron time history

The control in this plane is activated only during turning. After every turn small oscillation occur. The heading and yaw angle response is represented in Figure 5.37.



**Figure 5.37** - Heading and yaw angle evolution

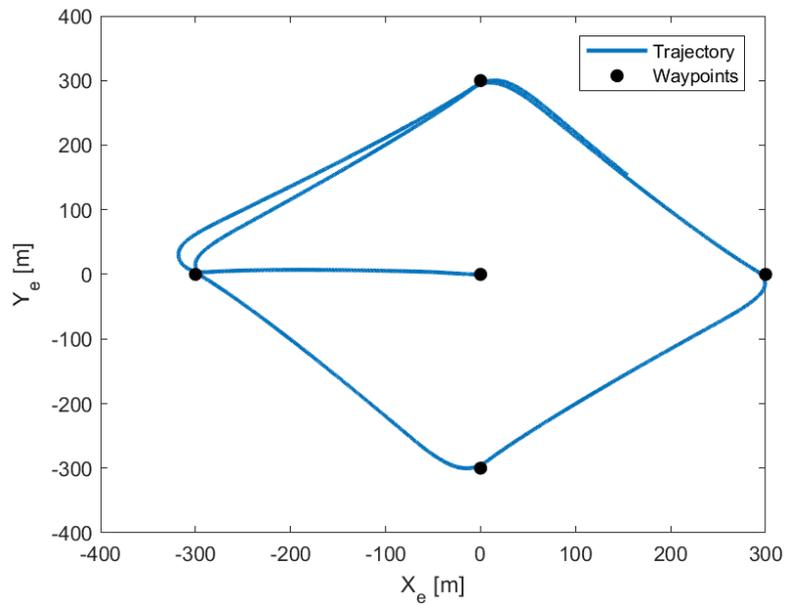
Even if some disturbances are present, the yaw angle  $\phi$  runs after its reference value  $\phi_{ref}$ , commanded by a PID controller, quite precisely and always respecting the imposed constraints. The other lateral-directional state variables are presented in Figure 5.38.



**Figure 5.38:** Lateral-directional states evolution

Even if some oscillation occurs, these variables become null in a quite short time.

The actual trajectory and the waypoints are shown in Figure 5.39.



**Figure 5.39 -** Two-dimensional trajectory

As it can be seen, the first turn requires more time to be completed, because the heading variation is bigger compared to the other turns. The trajectory in a three-dimensional space is presented in Figure 5.40.

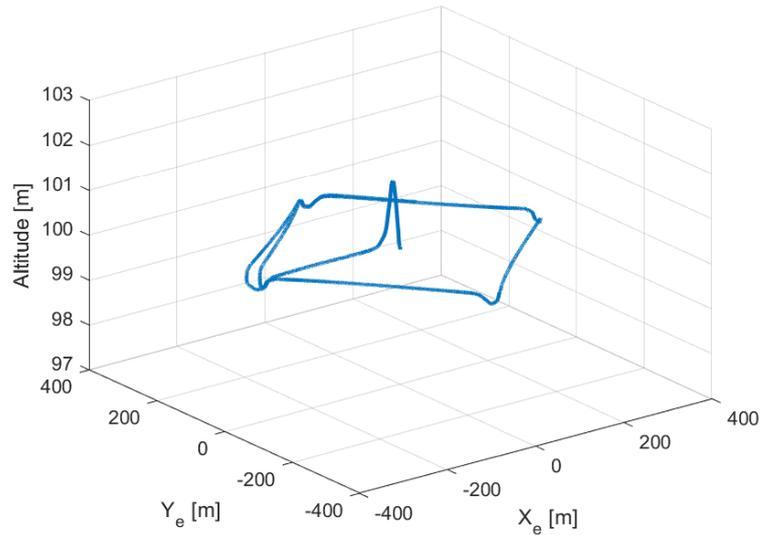


Figure 5.40 - 3D trajectory

A small altitude reduction is caused by the maneuvers. However, the control action rapidly corrects this reduction after the maneuvers. The presence of additive disturbances does not cause important variations from the undisturbed case. This result demonstrates the robustness of Tube-based MPC.

### 5.3.2 Octagonal path

The second simulation path chosen is an octagonal trajectory, as shown in Figure 5.13. The UAV has to have good robust control, in order to complete the frequent turns in presence of disturbance. The inputs in longitudinal plane are assigned as plotted in Figure 5.41.

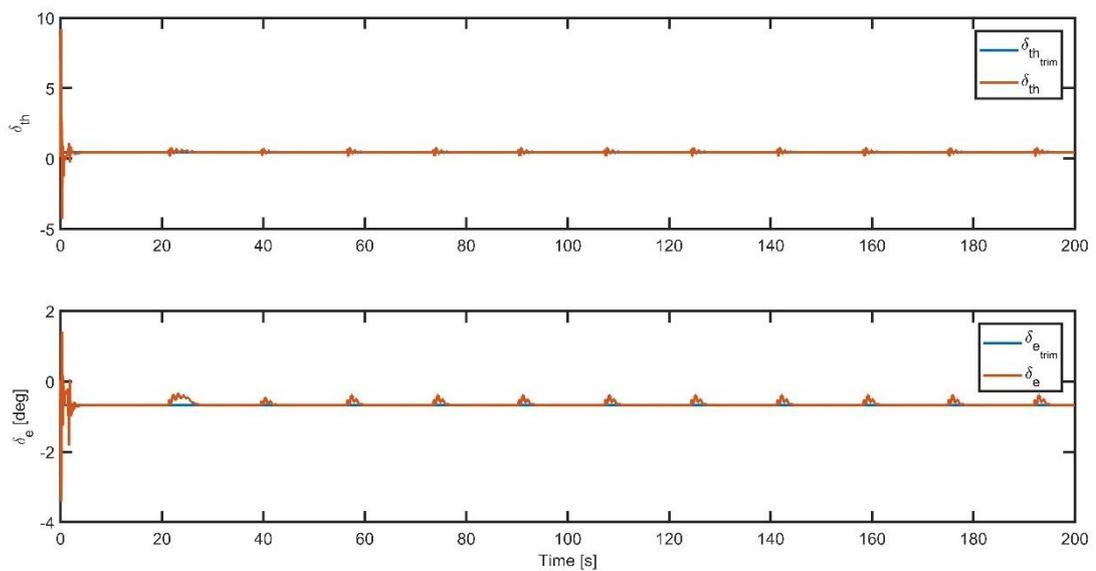


Figure 5.41 - Longitudinal inputs

Excluding the initial oscillations, which are within the imposed boundaries, the elevator presents a good dynamical behavior. It follows the reference value during straight flight and deviates slightly only during the maneuvers. In the first instants of simulation, a critical throttle behavior appears. Both upper and lower throttle bounds are widely exceeded in the first steps of simulation. This behavior is due to a very high initial throttle level in nominal system and consequently also in the nonlinear model. Indeed, the initial condition of null error between the two systems is imposed. The control action is hence the same for linear and nonlinear system in the first step. As the discrepancy between the two systems grows, the gain feedback stabilizing action has effect and the throttle runs correctly after the trim value. During the maneuvers, some oscillation occurs, but always widely within the imposed constraints, as it can be seen in Figure 5.42.

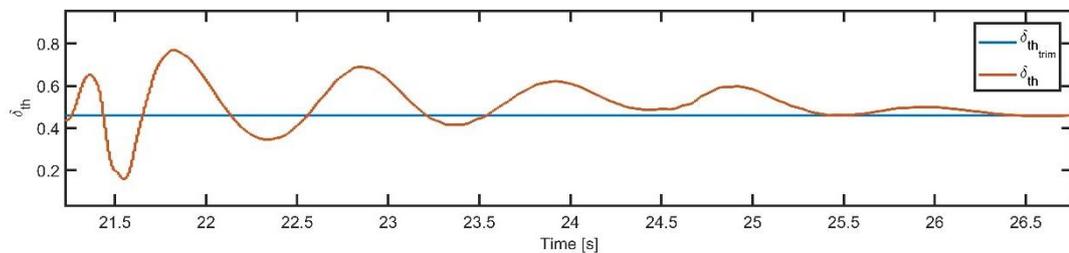


Figure 5.42 - Throttle evolution detail during a maneuver

It is important to notice that throttle constraints are respected for the rest of simulation, even in presence of disturbance. The longitudinal airspeed and the altitude respond to these control actions as shown in Figure 5.43.

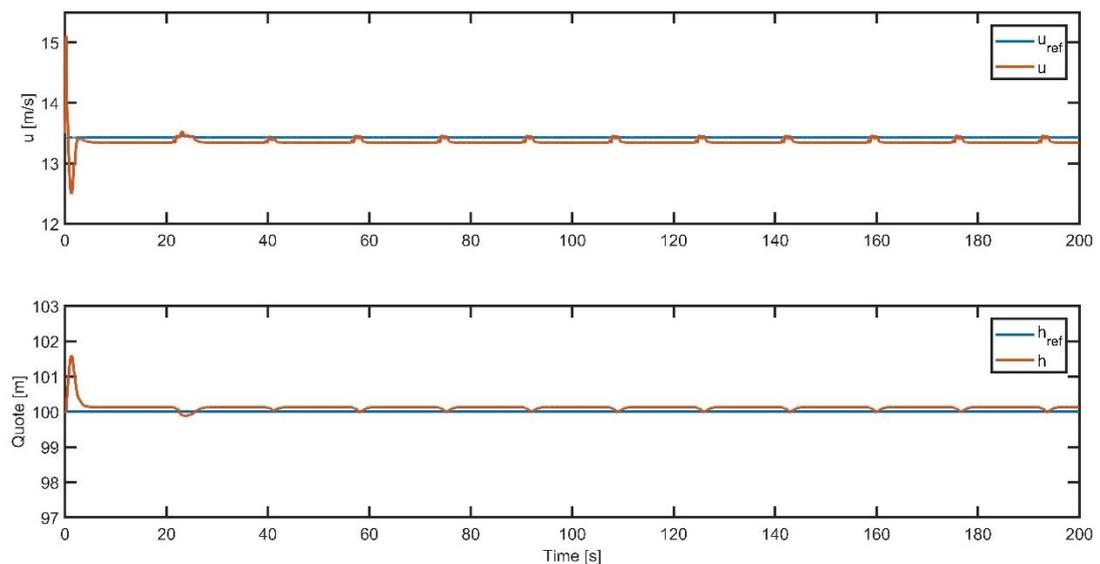


Figure 5.43 - Longitudinal airspeed and altitude time history

Throttle beginning oscillations reflect on longitudinal airspeed oscillations, but its short duration does not affect the stability of the system. Indeed, after few seconds the longitudinal airspeed tracks correctly its reference. The altitude evolution is precise and shows a small decrease only during the maneuvers. The other longitudinal states are reported in Figure 5.44.

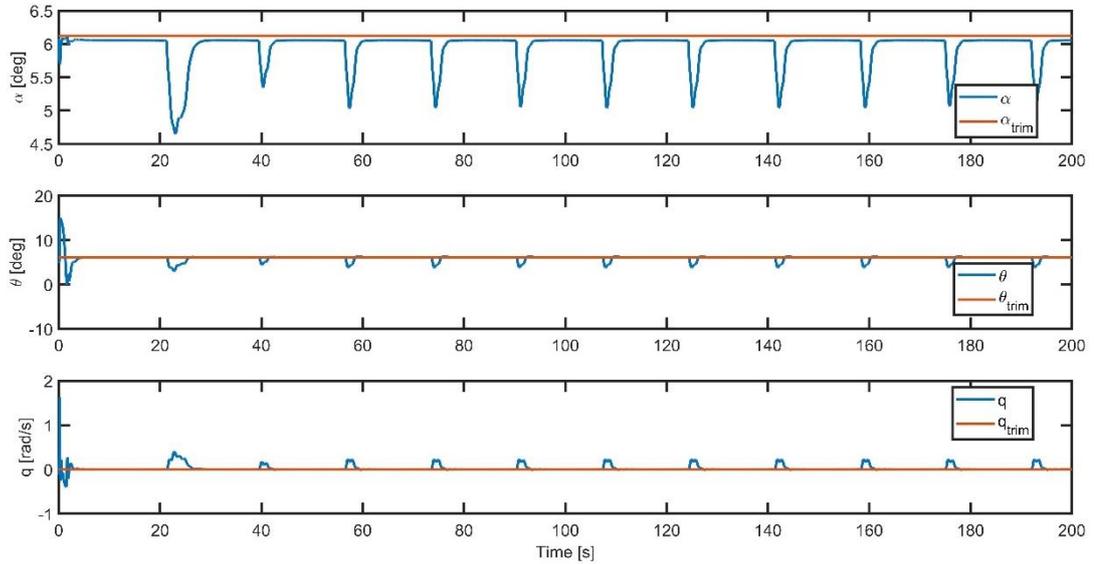


Figure 5.44 - Longitudinal states evolution

Angle of attack  $\alpha$  and pitch angle  $\theta$  run perfectly after their respective trim value and they show some limited small oscillations at the beginning of simulation. The pitch rate  $q$  is null in the straight horizontal flight and is positive during the maneuvers. Some fluctuation exists at the first instants of the simulation. In the lateral-directional plane the aileron deflection reported in Figure 5.45 is assigned.

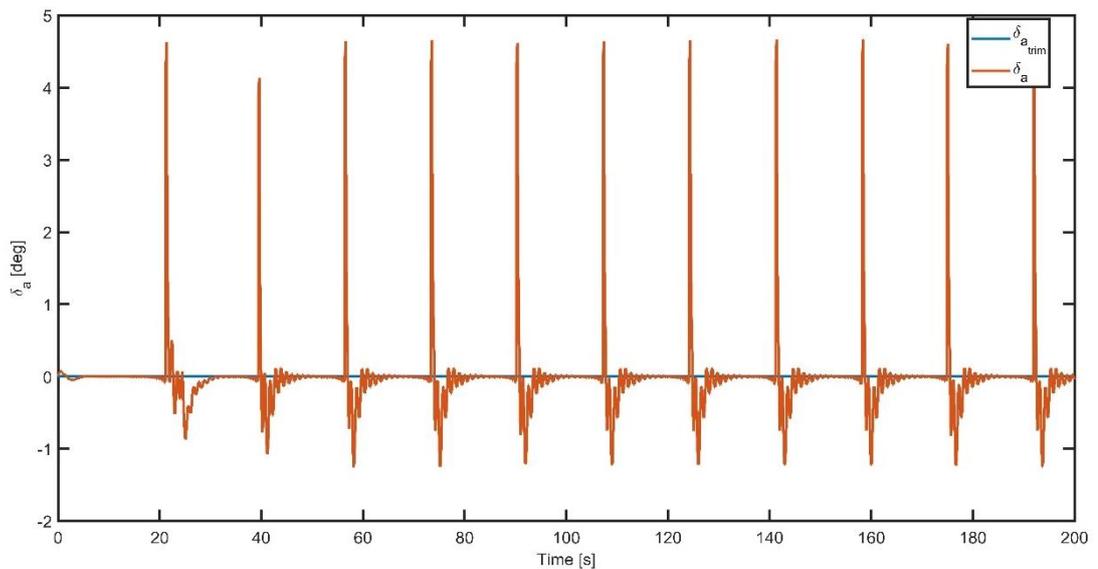
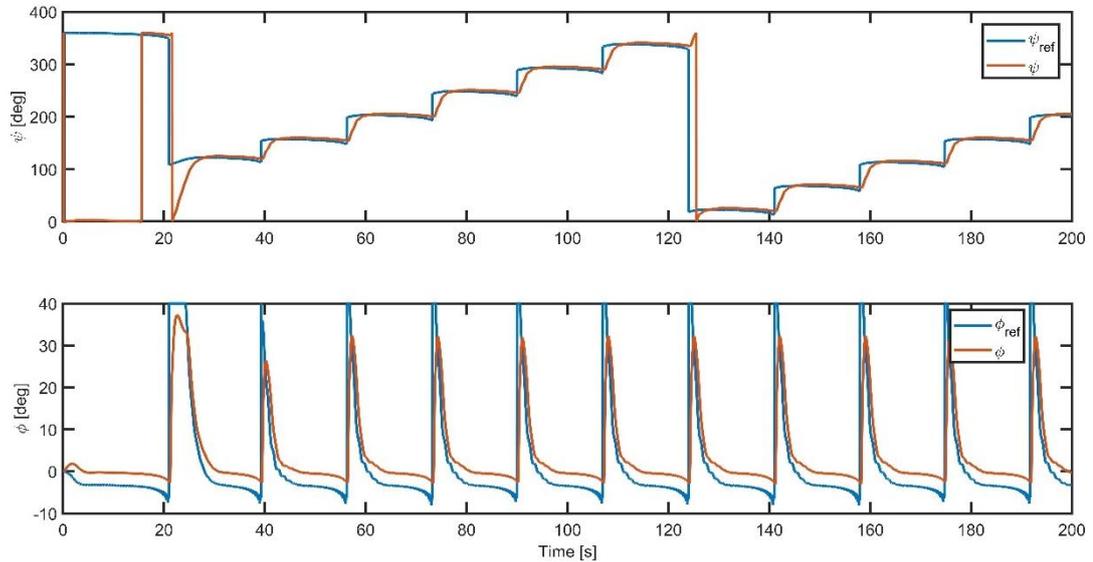


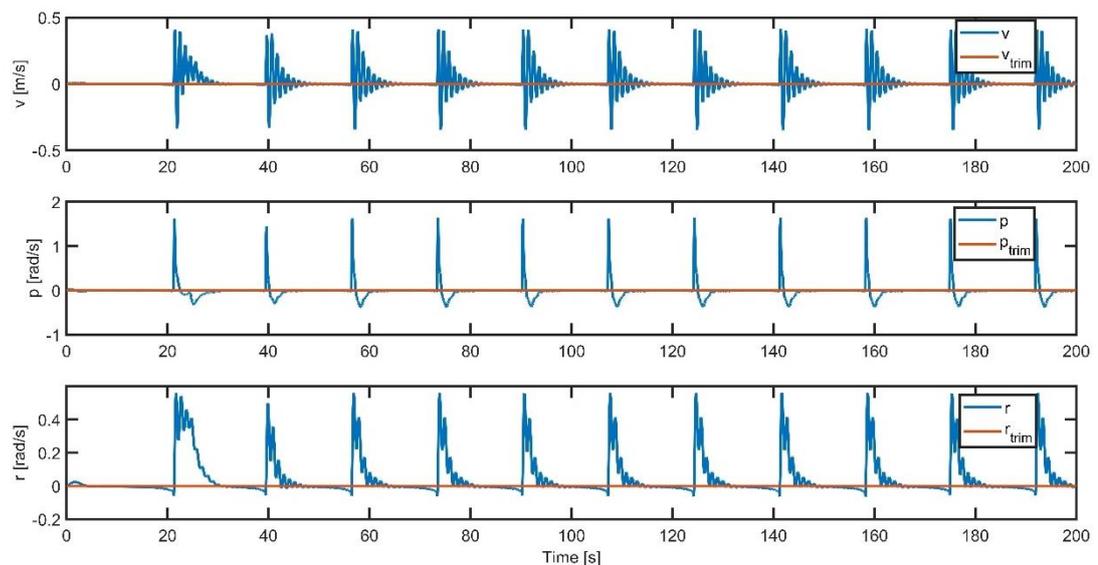
Figure 5.45 - Aileron time history

No critical fluctuations exist at the beginning of the simulation and the aileron is not deflected in the horizontal flight. A quite strong action, followed by some corrective action, is given during the maneuver, in order to reach the desired value of heading and roll angle. These variables are shown in Figure 5.46.



**Figure 5.46** - Heading and yaw angle evolution

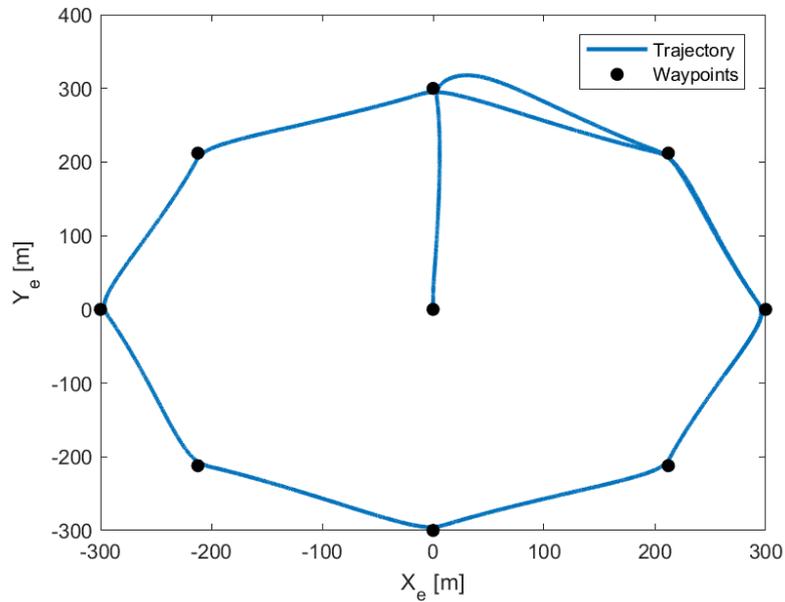
An overdamped behavior is shown in roll angle during the maneuvers. While the roll reference is saturated during the maneuvers, the actual value does not reach the desired value. Furthermore, a delay is shown in following the reference. The other lateral-directional variables are plotted in Figure 5.47.



**Figure 5.47** - Lateral-directional states evolution

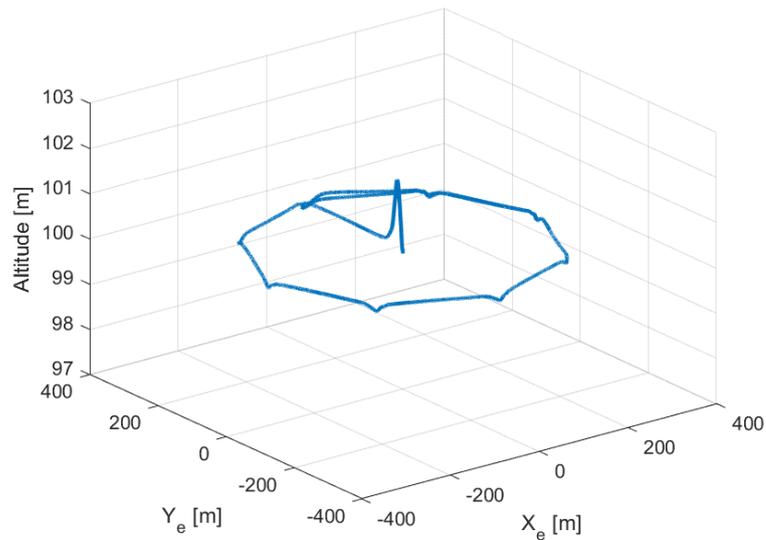
The lateral airspeed  $v$  goes never away from the null value during the flight between two waypoints, but it shows some damped oscillations during the turns. The

roll rate  $p$  response is precise, not fluctuating and fast. The yaw rate  $r$  becomes nonnull only during the maneuvers. When the reference yaw angle  $\psi_{ref}$  is reached, the yaw rate is again suppressed. The two-dimensional trajectory is represented in Figure 5.48.



**Figure 5.48** - Two-dimensional disturbed octagonal trajectory

Comparing Figure 5.20 with Figure 5.48, it is possible to notice the effect of the disturbance on the trajectory. As it can be seen, the trajectory between the first and the second waypoint is not perfectly straight, but a small correction is needed in order to maintain the UAV in the correct way. The resulting three-dimensional trajectory is reported in Figure 5.49.



**Figure 5.49** - Three-dimensional octagonal disturbed trajectory

### 5.3.3 “Snake” path

A “snake” path is now considered in presence of additive disturbance. The ideal path is the one reported in Figure 5.22. Throttle and elevon deflection are assigned as shown in Figure 5.50.

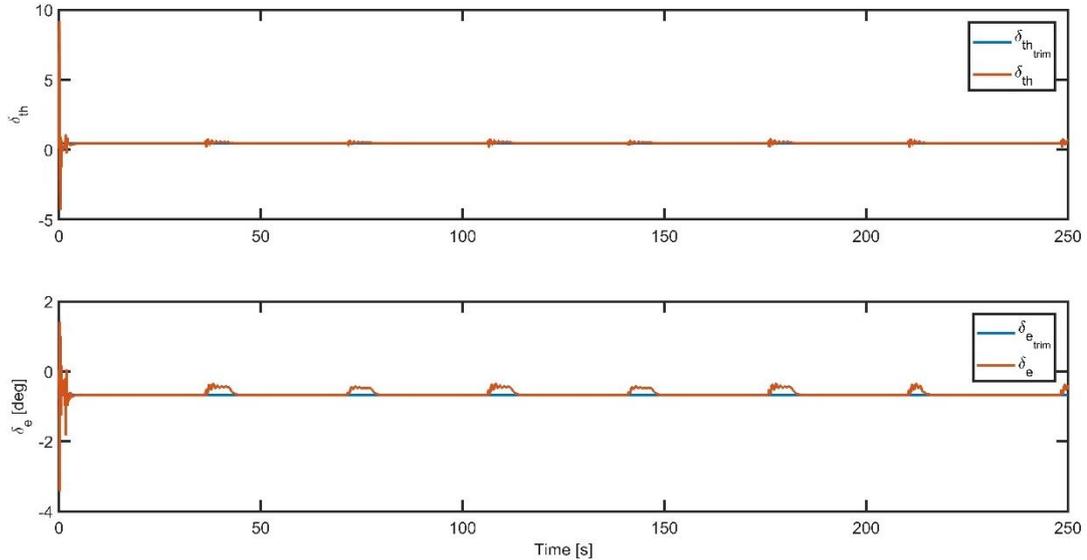


Figure 5.50 - Longitudinal inputs

Elevon deflection evolution is satisfactory and it never steps away from trim value, neither during the maneuvers. The control action is independent from the direction of the turn. Throttle presents a critical behavior in first steps of simulation, when it exceeds both upper and lower bounds. After about 5 s, the throttle settles on the trim value and its response become precise and satisfactory. The constraints are respected during every maneuver. As example it is taken the first turn (see Figure 5.51).

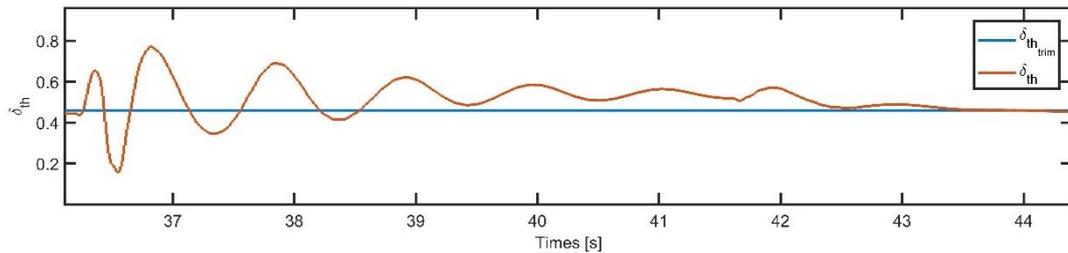
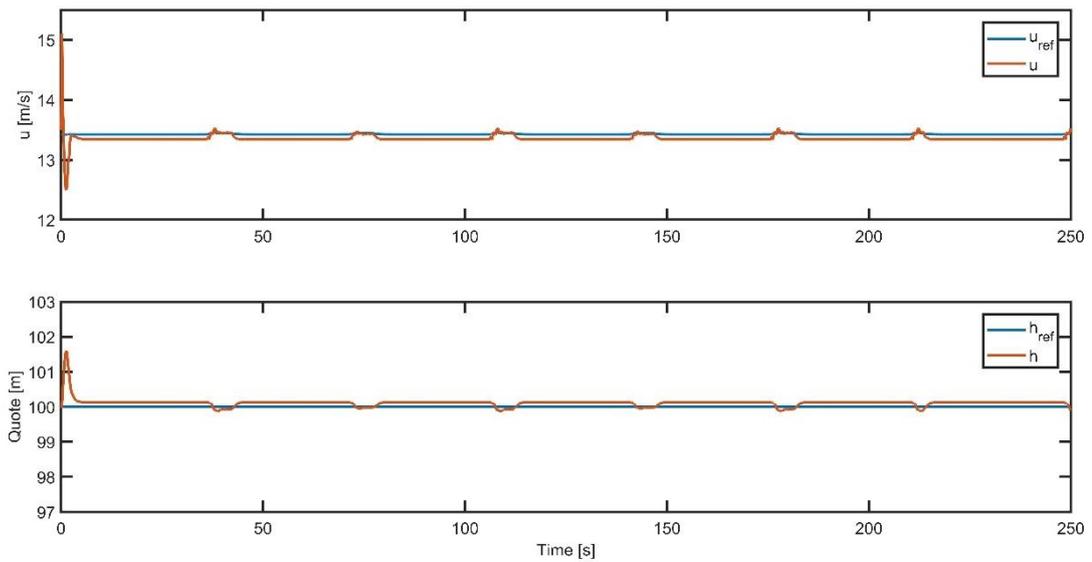


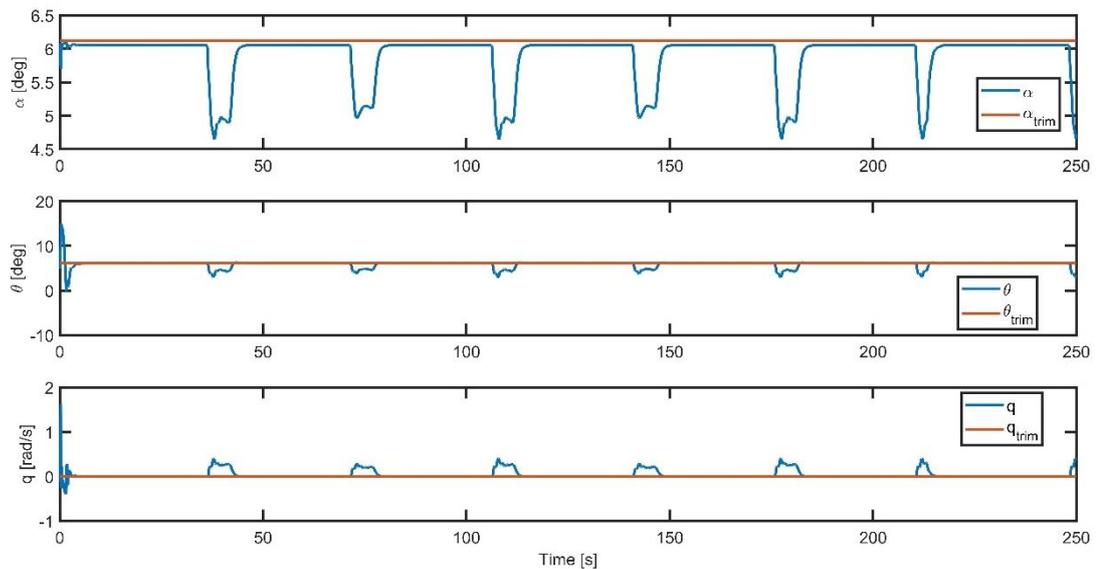
Figure 5.51 - Throttle evolution detail during a maneuver

As it can be seen, even if additive disturbance exists in this simulation, throttle never exceeds the lowest value of 0.15 and the biggest value of 0.8. Longitudinal airspeed and altitude response to these inputs is shown in Figure 5.52.



**Figure 5.52** - Longitudinal airspeed and altitude evolution

The same beginning oscillations as in the previous cases occur, but they are damped in few seconds and do not compromise the quality of the control. Airspeed increases slightly during the maneuvers, because of an altitude reduction. In Figure 5.53 the longitudinal states are plotted.



**Figure 5.53** - Longitudinal states time history

The angle of attack  $\alpha$  does not manifest any oscillation before settling to its trim value, whereas the pitch angle  $\theta$  fluctuates without exceeding the constraints before reaching its trim value. The pitch rate  $q$  presents some fluctuations, which are quickly damped, and settles to the null value. During the maneuvers pitch angle and angle of attack decrease and cause an altitude reduction. The lateral-directional input assigned in this path is reported in Figure 5.54.

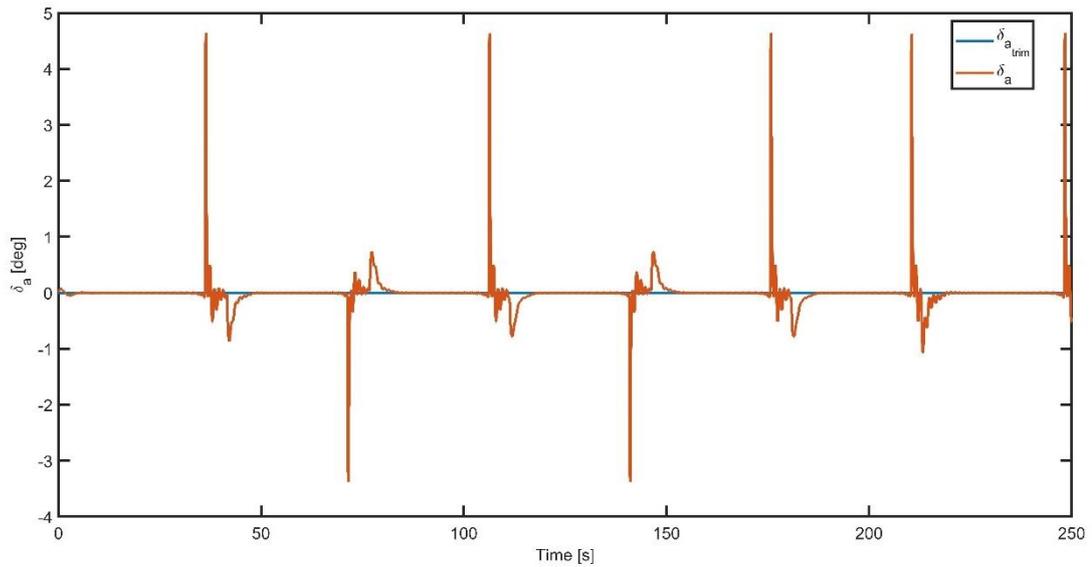


Figure 5.54 - Aileron evolution

In this case the first four turns are in alternate directions. Because of the lateral wind disturbance, it is interesting to notice that in this case the right turns require a bigger aileron deflection than the left turns. This different control action reflects on the roll angle  $\phi$  as it can be seen in Figure 5.55.

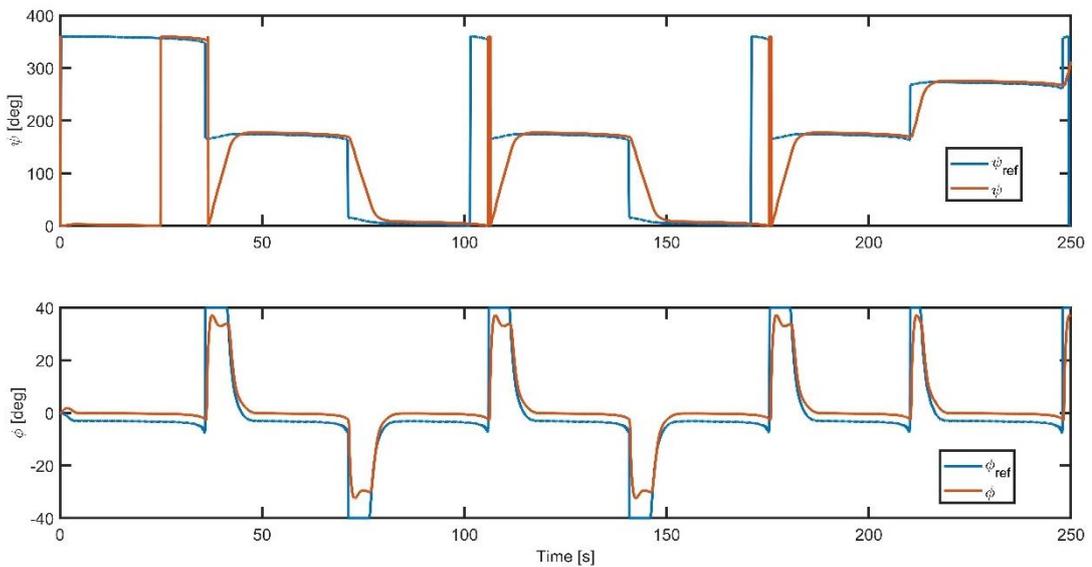


Figure 5.55 - Heading and yaw angle evolution

Even if the required heading variation  $\Delta\psi = \psi - \psi_{ref}$  is the same for every maneuver, the yaw angle response is not the same for the right and the left turn because of the presence of disturbance. In right turns the maximal heading angle is about  $37^\circ$ , whereas in the left turns the maximal heading angle is about  $33^\circ$ . This not symmetrical behavior in lateral-directional plane can be also seen in Figure 5.56.

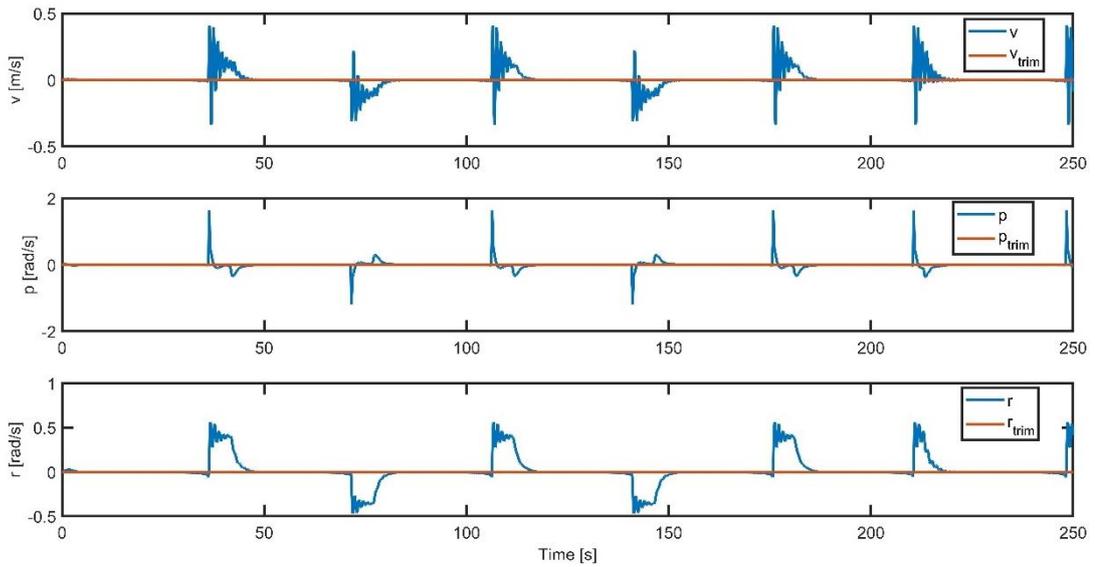


Figure 5.56 - Lateral-directional state evolution

The lateral airspeed  $v$  presents positive deviation bigger than the negative. Indeed, the maximal lateral airspeed is about  $0.4 \text{ m/s}$  and the minimal is about  $-0.33 \text{ m/s}$ . The roll rate  $p$  shows bigger positive variations than the negative as well as the yaw rate  $r$ . The two-dimensional trajectory is plotted in Figure 5.57.

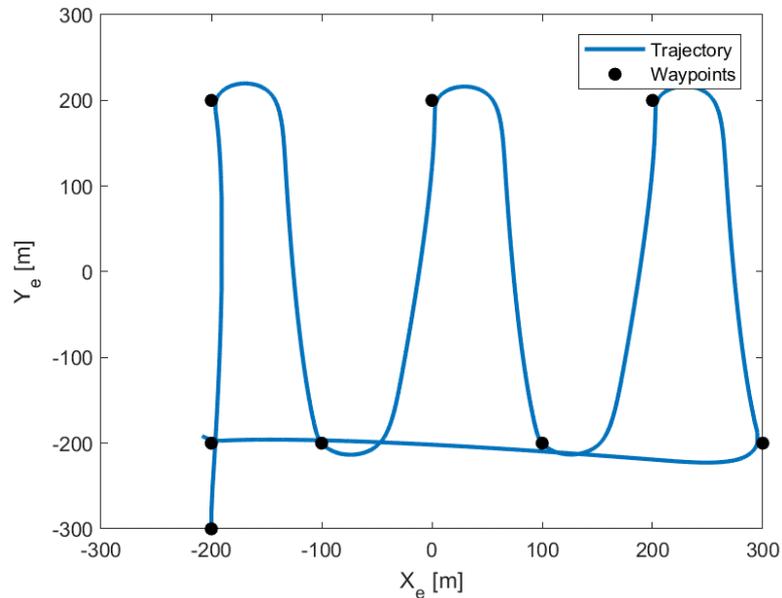
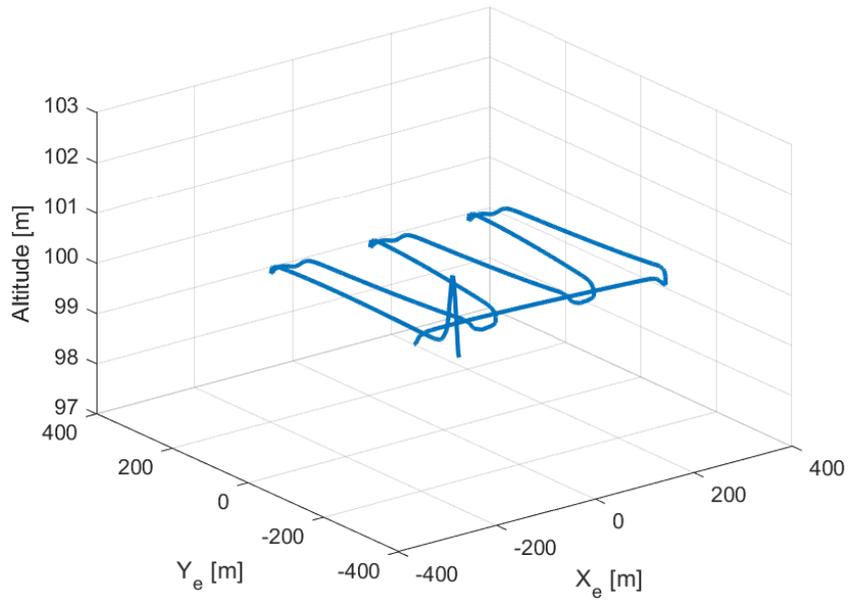


Figure 5.57 - "Snake" disturbed two-dimensional trajectory

The trajectory does not coincide to the undisturbed one, in Figure 5.29, because of the disturbance. The effect of a nonsymmetrical aileron control action can be seen in Figure 5.57, the right turns have a smaller turning radius than the left ones. Anyway, the Tube-based Model Predictive Control provides a robust control against the external noise and lets to the UAV to accomplish the assigned task. The three-dimensional disturbed "snake" trajectory is plotted in Figure 5.58.



**Figure 5.58** - Three-dimensional disturbed “snake” trajectory

## Chapter 6: Conclusions

In this chapter the main conclusions are drawn about this work and possible future works are suggested to improve model performance and confidence.

### 6.1 Conclusion

The aim of this work is to develop a model-based control, able to handle with uncertainties and exogenous disturbance and to be implementable in a real-time application, such as an UAV.

Starting from a database containing physic and aerodynamic properties of the mini-UAV MH850, a mathematical model is built to simulate the UAV dynamic behavior and to develop a suitable Tube-based Model Predictive Controller for the system. To simplify the problem, a linearization of the nonlinear equation of motions is done and decoupled linear equations of motion are derived.

Firstly, two Model Predictive Controllers are elaborated based on the two UAV linear models, one for the longitudinal plane dynamics and the other for the lateral-directional plane dynamics. In the longitudinal plane five state variables are considered, in order to control both airspeed and altitude. In the lateral-directional plane only the inner loop is controlled by a Model Predictive Controller, whereas the outer navigation loop is controlled by a PID.

In view of a future hardware real-time implementation, Model Predictive Controllers work with a slower sample time compared to the dynamic time constant. These controllers are validated through three tests conducted on the model in three different paths.

It is successfully demonstrated the MPC implementability in real-time applications with respect to constraint sets. Indeed, no critical responses have been revealed in the simulations and both state and control constraints are always respected.

Secondly, these controllers are extended to Tube-based Model Predictive Controllers, which ensure the hard constraints, the robustness and real-time implementability with almost the same computational effort than a conventional Model Predictive Control.

Three tests are conducted on three different paths in presence of disturbance. Only the throttle presents a critical behavior in the first instants of the simulation, but after about 5 s it settles on its trim value. The other variables evolve within the imposed

boundaries and in general have a satisfactory dynamic response. Thanks to these tests, also the robustness of this approach is successfully demonstrated.

Definitely, it is proven, how Tube-based Model Predictive Control is a good approach in real-time applications, such as UAV, and to those systems, which must evolve within hard constraints and are subjected to uncertainties and exogenous disturbance.

## **6.2 Possible future works**

During this work several areas, which need to be considered in further researches or studies, are emerged. Further studies can be conducted in order to:

1. Improve the tuning parameters for both MPC and Tube-based MPC, in order to delete the starting oscillations, which occur to many variables and are especially critical for the throttle, with the final goal to obtain a flyable controller.
2. Consider more than one flying condition.
3. Improve the disturbance modeling. Disturbance are now considered only as additive disturbance for each body axes, a more realistic disturbance model could be necessary.
4. Improve sensors modeling and simulate their disturbed dynamic, too.
5. Implement the flyable controller on autopilot board described at the beginning of Chapter Chapter 2: for the Hardware -In-the-Loop and experimental validation.

These further works could let to obtain a more realistic simulation environment, where it could be possible to experiment new control methodologies, as well as to have an experimental validation of this controller.

## Chapter 7: References

- [1] D. Q. Mayne e J. B. Rawlings, Model Predictive Control: Theory and Design., Nob Hill Publishing, 2009.
- [2] L. Wang, Model Predictive Control System Design and Implementation using MATLAB®, London: Springer, 2009.
- [3] J. Gertler, «U.S. Unmanned Aerial Systems,» *Congressional Research Service Report for Congress*, 2012.
- [4] R. Mo, Q. Geng e X. Lu, «Study on control method of a rotor UAV transportation with slung-load,» *2016 35th Chinese Control Conference (CCC)*, 2016.
- [5] K. N. Tahar, A. Ahmad e W. A. A. W. M. Akib, «UAV-based stereo vision for photogrammetric survey in aerial terrain mapping,» *2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, 2011.
- [6] S. Choi e E. K. Kim , «Building crack inspection using small UAV,» *2015 17th International Conference on Advanced Communication Technology (ICACT)*, 2015.
- [7] H. Chao, Y. Cao e Y. Chen, «Autopilots for small unmanned aerial vehicles: A survey,» *International Journal of Control, Automation and Systems*, 2010.
- [8] B. Khada e Y. Ghazzawi, «Robust PID Control Design for an UAV Flight Control System,» *World Congress on Engineering and Computer Science*, 2011.
- [9] G. V. S. O. N. Michel, S. Bertrand e D. Dumur, «Design and parameter tuning of a robust model predictive controller for uavs,» *20th IFAC World Congress*, 2017.

- [10] K. Alexis, G. Nikolakopoulos e A. Tzes, «Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances,» *Control Engineering Practice*, 2011.
- [11] M. Kamel, T. Stastny, K. Alexis e R. Siegwart, «Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System,» *Cham: Springer International Publishing*, pp. 3-39, 2017.
- [12] W. Langson, I. Chrysoschoos e S. Raković, «Robust model predictive control using tubes,» *Automatica*, 2004.
- [13] MAV Tech, 5 11 2018. [Online]. Available: <http://www.mavtech.eu/it/prodotti/discontinued-products/drone-mh850/>.
- [14] E. Capello, P. Marguerettaz e F. Quagliotti, «Preliminary assessment of flying and handling qualities for mini-UAVs,» *Journal of Intelligent & Robotic Systems*, vol. 65, n. 1, pp. 43-61, January 2012.
- [15] E. Capello, G. Guglieri e G. Ristorto, «Guidance and control algorithms for mini-UAV autopilots,» *Aircraft Engineering and Aerospace Technology*, vol. 89, n. 1, p. 133–144, 2017.
- [16] B. Etkin e L. Reid, *Dynamics of Flight: Stability and Control*, New York: John Wiley and Sons, 1996.
- [17] A. Bemporad e M. Morari, «Robust Model Predictive Control: A Survey,» in *vol. 245 of Lecture Notes in Control and Information Sciences*, London, Springer, 1999.
- [18] D. G. Luenberger, *Introduction to Dynamic Systems*, New York: John Wiley & Sons, 1979.
- [19] B. Stevens e F. Lewis, *Aircraft Control and Simulation*, New York: John Wiley and Sons, 2003.
- [20] S. Boyd, L. E. Ghaoui, E. Feron e V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Philadelphia, USA: Society for Industrial and Applied Mathematics (SIAM), 1994.
- [21] J. Zhang, A. Swain e S. K. Nguang, «Appendix A Solving Linear Matrix Inequality (LMI),» in *Robust Observer-Based Fault Diagnosis for*

*Nonlinear Systems using MATLAB®*, Springer International Publishing Switzerland, 2016.

- [22] G. C. Goodwin, M. M. Seron e J. A. de Doná, *Constrained Control and Estimation: An Optimisation Approach*, London: Springer-Verlag, 2005.
- [23] F. Blanchini, «Set invariance in control,» *Automatica*, 1999.
- [24] W. H. Kwon e A. E. Pearson, «A modified quadratic cost problem and feedback stabilization of a linear system,» *IEEE Transactions on Automatic Control*,, 1977.
- [25] W. H. Kwon, A. M. Bruckstein e T. Kailath, «Stabilizing State Feedback Design Via the Moving Horizon Method,» *INT J Control*, 1983.
- [26] S. V. Raković, E. Kerrigan, K. Kouramas e D. Mayne, «Invariant Approximations of the Minimal Robust Positively Invariant Set,» *IEEE Transactions on Automatic Control*, 2005.
- [27] R. Schneider, *Convex bodies: the Brunn-Minkowski theory*, Cambridge University Press, 2013.
- [28] I. Kolmanovsky e E. Gilbert, «Theory and Computation of Disturbance Invariant Sets for Discrete-Time Linear Systems,» *Mathematical Problems in Engineering*, 1998.
- [29] J. Löfberg, «YALMIP: A Toolbox for Modeling and Optimization in MATLAB,» in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [30] J. J. Di Stefano, A. R. Stubberud e I. J. Williams, *Feedback and Control Systems*, Schaum's outline series a cura di, McGraw-Hill, 1967.
- [31] C. V. Rao, S. J. Wright e J. B. Rawlings, «On the application of interior point methods to model predictive control,» *Journal of Optimization Theory and Applications*, 1998.