Politecnico di Torino

Facoltà di Ingegneria
Master degree in Mechatronic Engineering

Guidance and Control Optimization for U.A.V.

Candidate:
Laviola Domenico

Supervisors:
Prof. Novara Carlo
Ing. Perez N. Carlos

A.A. 2017 - 2018

Contents

List of Figures

Chapter 1

Introduction

The aim of this thesis is to design a discrete time optimized Guidance and Control for an already present Unmanned Aerial Vehicle (U.A.V.) realized for the Borea project [35] [14], [33], [30] by the Space and Precision Automatics (SPA) group from Politecnico di Torino. The main purpose is to replace the current not-optimal polynomial guidance and control block with an optimal one exploiting advanced control technique properties. The method identified to fulfill this aim is the so-called Model Predictive Control (MPC) that allows to solve trajectory optimization problem through the discrete-time constrained optimal control. Thanks to the extension of U.A.V. in the civil sphere and its analogy with aircraft in descending propulsive phase, research to increase performances have increased all over the world. These concerns the creation of an optimized guidance that returns feasible optimal trajectory profiles which bring the UAV to a desired point by minimizing a performance index. About the project, the main aim is to test Guidance, Navigation and Control (GNC) algorithms based on Embedded Model Control (EMC) theory [6], [8], [9], [10], [18]. Since the project is based on these techniques, a small description of how they interact with each other is given in Chapter 3. While in such a way to relate all the measures returned by different sensors, in Chapter 4 are provided descriptions and technical reasons about the assumpion of several reference frames. Subsequently in Chapter 5 are described the complete UAV's geometrical configuration, physical model and analysis of the attitude behaviour to understand how the quadcopter reacts to input commands. A physical model description is necessary inasmuch the guidance will be subject to dynamic equa-

tions, while kinematic and finally the dispatching technique are necessary to obtain the values of the forces that must be provided by each propellers-motor. Moreover, embedded models are breafly described since its definition is usefull for the implementation of the discrete time guidance and control optimization.
While the most important contribution is described in Chapter 6 where it is provided to replace the continuos time polynomial guidance with discrete time optimized one that furthermore takes into account the physical saturation of the actuators. Since the polynomial guidance is based only on the placement of polynomial coefficients that satisfy boundary condition and not considers dynamics the physical saturation of propellers-motors is not optimized and for this reason this limit has led to the use of new control techniques that are increasingly used in recent years. However, before proceeding with the complete replacement of the polynomial guide, is prosed to optimized it, leaving the algorithm free to choose the flight time. Guidance optimization issues are often solved as discrete time optimal control, wich the aim is to minimize cost function subject to equality and inequality linear constraints. The technique that best suits this purpo se is Model Predictive Control (MPC) technique. As the name itself indicates, this technique is based on predictive models that will be extensively explained in this chapter. In addition, for the fulfillment of the requirements a change has been made in such a way to avoid non-linearities, thus leading us back to a linear optimization problem. This modification is broadly described in the paragraph 6.2.1 and is the key point for the resolution and creation of a guidance/control optimization. From this assumption two techniques based on the same philosophy of optimization are designed using different approaches to achieve the same requirements. The two approaches correspond to the name of targetting (or two boundary value problem TBVP) and tracking, explained respectively in paragraphs 6.2.1 and 6.2.2. Both these techniques consider the bounds on output and input that for this reason they are called constrained. The results obtained using the above techniques are shown in Chapter 7 in the same order in which they have been described. In this chapter will be shown how the constraints of equality and inequality influence the feasibility of the considered optimization problem. While the conclusions are reported in Chapter 8 where are analyzed the improvement and the differences between not-optimal polynomial guidance and the constrained discrete time optimal showing theirs advantages and disadvantages.

Chapter 2

State of art

The guidance systems make their first appearance with the advent of rockets. This merit is attributed to Dr. Goddard who used a rudimentary gyroscope system. These systems are widely used and had important applications with the birth of spacecraft, guided missiles, etc. Following the Second World War the guidance systems had a notable evolution, in chronological time we remember the guidance navigation and control for V2 which was a very sophisticated system in 1942 thanks to Von Braun.

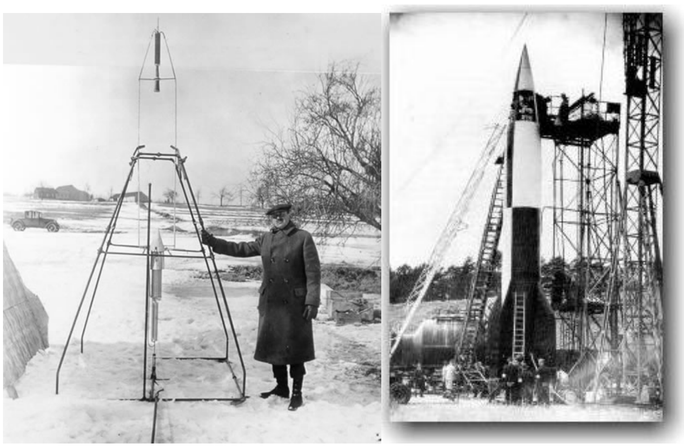

Figure 2.1: Goddard (left) and V2(right)

As written in [15] "Early v2 leveraged self -contained closed loop with 2 gyroscopes and lateral accelerometer and simple analog computer to adjust the azimuth for the rocket in flight. Analog computer signals were used to drive 4 external rudders on the tail fins for flight control". Subsequently, the Americans responded to the German technology of the V2, through the Jet Propulsion Laboratory founded by the Army Ordnance 1942 creating the MGM-5 Corporal. Two months after the advent of NASA (National Aeronautics and Space Administration), the JPL went under the Army jurisdiction. The guidance systems from that time on, had further applications and at the beginning of 1958, Nasa JPL and Caltech developed other types of guidance to be used primarily for unmanned flight. In the early 1950s MIT was chosen by the Air Force Western Development Division to create a self contained guidance system backup for ATLAS intercontinental ballistic missiles. Has written in [15] "The Atlas guidance system was a combination of an on-board autonomous system, and a ground-based tracking and command system. The self-contained system prevailed in ballistic missile applications" thanks to Jim Fletcher. New computational based solutions for guidance systems were introduced in 1952 by Dr J. Halcombe Laning Jr. and Dr. Richard Battin to achieve lower processing speeds. This led them to the realization of analytical solutions on the atlas inertial guidance used for the mission Apollo. Charlie Bossart and Walter Schweidetzky, head of this group, also contributed to these projects.

Figure 2.2: Shuttle, Apollo 11 `https://grahamedwardsonline.files.wordpress.com`

The initial guide resulting from it was called DELTA which took into consideration the distance of the aircraft with respect to the reference trajectory and controlled the speed through the VGO (velocity to be gained). Due to the limitations in accuracy of the IMU and calculation power, the guide described above was discarded. These problems were overcome with the introduction of the Q system that revolutionized the era of missiles that still uses variants of this method. Moreover, hybrid guidance types were created for the missions of the Apollo program starting from August 1961 and taking the name of Powered Explicit Guidance PEG composed by Delta and Q system guidance. Recently, the military force has developed vehicles completly autonomous, able to carry out reconnaissance on high risk areas avoiding the loss of soldiers and therefore saving lives. These vehicles are known under the name Unmanned aerial vehicles (UAV) and with them the guidance systems have evolved greatly. With the advent of UAVs, a myriad of research has been carried out in the scientific field involving and influencing the civil and industrial sectors. In fact, as often happens, the innovations in the military are exploited for the creation of civil and industrial applications. The UAVs that have had the greatest success in the civil field are those that are derived from the extension of the helicopter operating principle.

Figure 2.3: UAV (left) `https://grahamedwardsonline.files.wordpress.com` and JRC quadrocopter (right)

They are equipped with n propellers arranged radially and spaced with an angle that depends on the number of propellers present. The creation of the latter arises from the need to create a vehicle that has a high maneuverability necessary in impervious areas. The agility of the aircraft just described is due to the presence of the four propellers that bring about a change of quick set-up. Thanks to its extension in the civil sphere and its analogy with the lander for the inter-plenary landing, the research on the development and the increase of performances have increased all over the world. Improvements are made using advanced control techniques or better known as optimization methods. They concern the creation of an optimized guide that returns feasible values of the inputs and that brings the system to the desired final state by minimizing a performance index. This technique is known under various names but the most used and widespread in this field is "trajectory planning" and allows to generate the profiles of the state components as a reference for the control. This technique should not be confused with path planning through which the threedimensional (3D) configuration is located from the initial position to the final one. This corresponds to finding a static geometry path and does not include time evolution. Generally speaking, the model and solution algorithm of the trajectory planning problem are more complicated than the ones of the path planning problem.

In fact, in this regard, various optimization methods have been introduced. The most intuitive method is to adapt the theory of optimal control to the trajectory planning problem to demonstrate this. Yao and Zhao [40] show how it is possible to use a discrete-time technique of optimal control that works in a predictive form, hence the name Model Predictive Control (MPC). Other references show other types of guidance based on various optimization algorithms such as Duan et al that introduces many optimization algorithms. It solves the optimization problem with no consideration of dynamic and kinematic constraints including the chaotic artificial bee colony (ABC) approach, the chaotic predator–prey biogeography-based optimization (CPPBBO) algorithm, the improved gravitational search algorithm, and reducing the infinite dimensional problem into the finite dimensional one [3]. Although these method might be useful, the dynamic and kinematic model of the UAVs is completely ignored and this type of result is unacceptable and as was written in [1] "because the models of the trajectory planning problem need to be closer to the real aircraft model". The assumption of a model similar to the real one is strictly necessary to obtain real feasible trajectories. This however leads to the use of more complicated models which entails increasing the magnitude of the problem of calculation and therefore greater difficulty in planning trajectories. The problems that are concerned with the optimization of trajectories make use of constraints on inputs and outputs that make the feasibility difficult.

Other approaches have been made using the finite reciding horizon optimal control problem to implement the approach and land-based phase of a reusable launch vehicle (RLV). There are other algorithms which can solve the trajectory planning problem as used to solve the problem of the unmanned combat aerial vehicle (UCAV) [41] to solve the ground attack trajectory planning problem. Kamyar and Taheri [28] also provide a solution to solve the trajectory planning problem of 6-DOF dynamic model as differential evolution-sequential quadratic programming (DE-SQP) approach and the particle swarm optimization-sequential quadratic programming (PSO-SQP) approach .

In literature, it is possible to find other papers that describe in the most desperate ways how to solve this optimization problem, so the solution is not found with a single method.

Another example is found by [12] , where it uses the differential flatness and polynomial function method. The last frontier reached in the guidance systems for autonomous aircraft is that of cooperation where a set of drones must carry out a particular objective that a single drone cannot reach. References to this argument are [1][12][13][25] and represent the latest improvements in this area.

Chapter 3

Borea Project

3.1 Guidance Navigation and Control G.N.C and Embedded Model Control E.M.C.

The main aim of this project is to test Guidance, Navigation and Control (GNC) algorithms based on Embedded Model Control (EMC) theory [35],[14],[33],[30],[6]. Since the quadrotor has significant equality with aircraft space vehicles in the landing propulsive phase, allows to extend GNC[8],[9],[10],[18], algorithms not only for Earth flight applications but also for space applications [5], [7]. The guidance of a vehicle have as aim to describe an appropriate trajectory which bring the UAV to a desired point respecting all the requirements. Due to the fact that, in real applications, there are various type limits present, the guidance is not only constrained by the configuration of the vehicle, but also by the environment, obstacles and target position [38][39]. While inertial navigation is used to estimate the state of the vehicle system (position, velocity and attitude) through the inertial measurement unit (IMU) formed by accelerometer and Gyroscope. The IMUs are the most used device to drive UAVs [29], spacecraft and landing [23], intelligent missiles [24]. The control defines the input command value that the actuators will be given in such a way to track reference signal given by the guidance algorithm. Since the project is based on Embedded Model Control (EMC) a little description on this argument is necessary. A good explantion about this argument can be found in [11], [31], [27]. From this, it is clear that Embedded Model

Control consider the presence of unstructed parametric uncertainty in the development of the model based control law, this technique is introduced to guarantee closed-loop stability[22][20] and to consider a linear system. This aim is obtained by assigning ignorance coefficients that interact decreasing the feedback control effort with respect to the model-based design. Moreover EMC shows that a model-based control law is mantained stable under uncertainty if the uncertainties that affect the Embedded Model (EM) are considered, calculated in real-time as disturbance dynamics. The disturbance vector is estimated as the difference between plant and model output and updated in real-time in the model error, so for this reason is mandatory to design a noise estimator, moreover as written from previous work on this project in [11]" appropriate separation of the components into low and high frequency domains by the noise estimator itself allows to recover and guarantee stability, and to reject the low frequency uncertainty " [8][19]. The complete model is composed by:

1) The controllable dynamics

2) The disturbance dynamics

3) The neglected dynamics

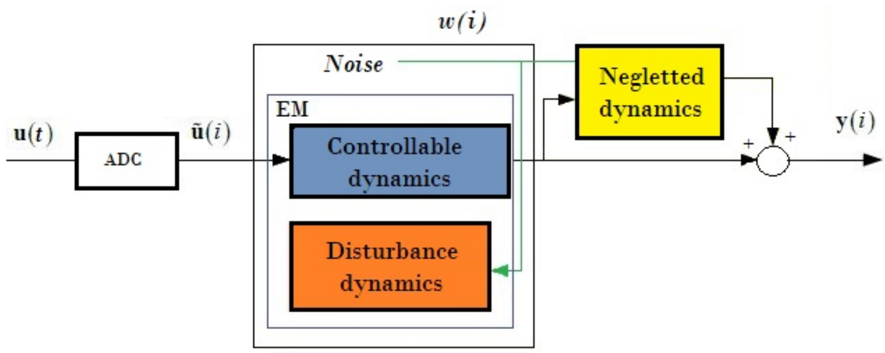

Figure 3.1: Extended plant [11]

As can be seen in the figure 3.1 that show that controllable and disturbance dynamics are observable and measured, together are called Embedded Model (EM) and has previously written all the control algorithm is designed taking into account these two parts. The EM is written under the consideration of discrete time and forms the control unit Heart.

3.1.1 Model Principles

As described by [38], the EM work sinchronusly and in a parallel way with respect to plant driven by the computed command u_i that have digital value and where i is the $i-th$ sample, the command comes from to the continuous quantized input driven by a polynomial guidance (that will be substituted with this thesis). The main aim is the formulation for guarantee a bounded model error where past uncertainty is modelized and deleted. The discrete value of the error variable e_i come from to the model error that is computed comparing plant and EM, as mentioned before. It is the only available measure of the past uncertainty and added in the disturbance state d_i , driven by an arbitrary input noise w_i.

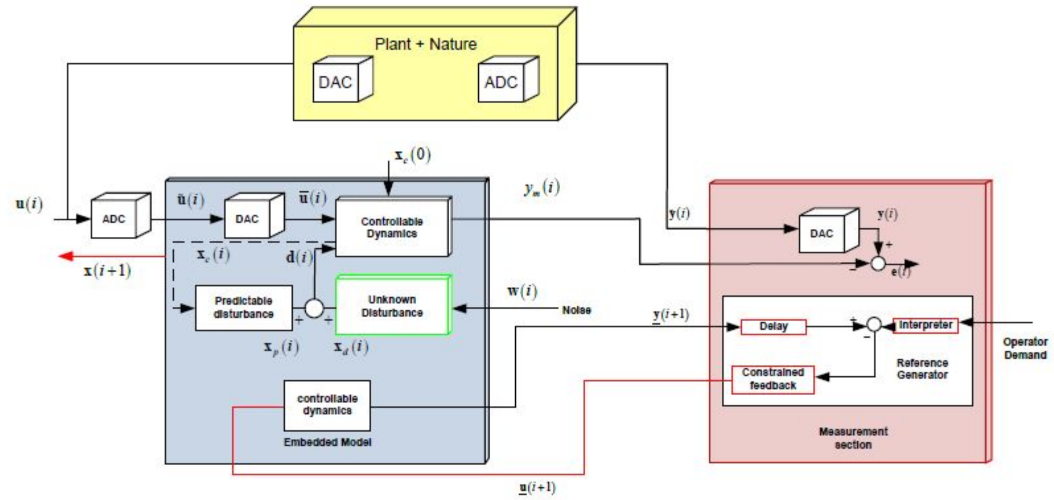

Figure 3.2: Plant, parallel embedded model and measurement [11]

The w_i taking correlation with e_i (noise estimator, NE) revealing the residual uncertainty and bringing the embedded model to align with the plant bounding e_i and to mantain the asymptotic stability of closed-loop system (EM+NE), a Noise estimator is designed. For the application that regards GNC (Guidance, Navigation and control), EMC is mandatory. The trajectory reference generator is realized using polynomial strategy while the Noise estimator that predicts the system states is implicitly developed in the navigation algorithm.

3.1.2 UAV's parameters and model architecture

The Borea project [35] [14], [33], [30] developed by the Space and precision automatics group had as aim to build a quadrocopter inasmuch it has significant equality with aircraft space vehicles in the landing propulsive phase. The quadrocopter is a very agile aircraft that can be controlled by man (manned) or programmed to perform the desired tasks autonomously (unmanned). For this reason the quadricopter is also included in the category called Unmanned Aerial Vehicle and can operate remotely thanks to integrated algorithms. These vehicles are used to fulfill several tasks thanks to their great flexibility. In fact, with them it is possible to recognize large dangerous areas in the military, to search for missing persons due to avalanches, watching woods or sensitive structures, or simply taking aerial images. In this subsection, the physical parameters (as mass, inertia and dimension) and project parameters (sample period, simulation time, constant variable) are given.

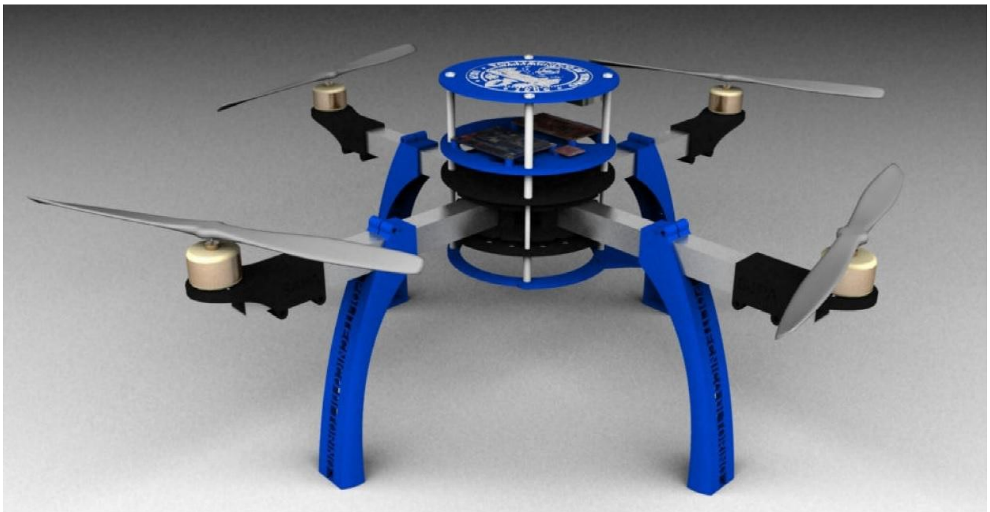

Figure 3.3: Borea Quadrotor

As can be seen from figure 3.3 , the quadrotor is equipped with 4 propellers, each driven by the respective motors allowing the aircraft to move in space. This aircraft has the ability to translate and rotate around its axis of reference thanks to its 6 degrees of freedom. The physical parameters that characterize this aircraft are summarize in the following table.

	Variable	Value	unit
1	mass	1.49	Kg
2	Inertia I_{xx}	$17.2e-3$	$Kg \cdot m^2$
3	Inertia I_{yy}	$18.5e-3$	$Kg \cdot m^2$
4	Inertia I_{zz}	$26.75e-3$	$Kg \cdot m^2$
5	Diameter Quadrotor	0.5	m
6	Quadrotor Height	0.05	m
7	Propeller Diameter	0.254	m

Table 3.1: Borea Quadrotor Parameters

Obviously, these data are not sufficient in the simulation environment, where to define how the system works it is necessary to define the sampling time or the simulation time itself. For this reason, the project parameters are also reported which are summarized in the table below as follows.

	Variable	Value	unit
1	Simulation Time	100	s
2	Gravity constant	9.81	$\frac{m}{s^2}$
3	sampling period	0.1	s

Table 3.2: Borea Quadrotor Parameters

To give more clarity and simplify at the same time how the simulator works, the complete architecture of the model is shown in the form of a block diagram .It can be divided into 4 main blocks that describe the main operations:

1. Operator
2. Simulator
3. Control unit
4. Sensors

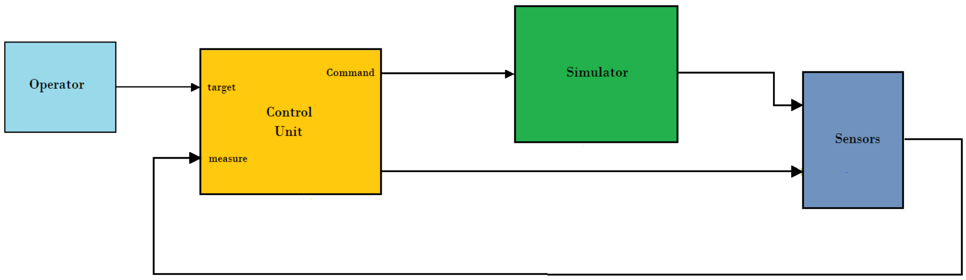

Figure 3.4: Model architecture

Where the operator block includes information about the target (position, velocity, acceleration), the control unit block takes into account the algorithms of guidance and navigation and control, the simulator takes in consideration the embedded model and the sensor block represents filter and algorithm to estimate attitude and position. In such a way to measure variables as acceleration linear and angular or position, a different sensor is present on board.

	Sensor	Value
1	Acceleration	3 axis linear acceleration
2	Gyroscope	3 axis angular acceleration
3	Magnetometer	Magnetic Field
4	Sonar	Vertical position
5	Gps	Horizontal position

Table 3.3: Quadrotor sensors

Chapter 4

Reference frame

The quadricopter, as was mentioned in the previous chapter, belongs to the category of unmanned aerial vehicle (UAV), which exploits the differential thrust in order to control the attitude (Dinamic,Kinematic) as is shown in Chapter 5, unlike conventional helicopters, which use complex mechanisms to control the coefficient of lift by modifying the propellers configuration.
In this chapter, the used reference frames are described, that are useful and required notations in the trajectory generation for dynamic systems, inasmuch could be possible that in a specific reference frame, the equations of motion assums a simpler form and consequently it is easier to find a solution that will be transformed again to the starting reference frame, obviously, in the starting reference frame. The reason for the assumption [2] of several different coordinate systems are the following:

- Newton's equations, aerodynamics forces and torques are expressed in the body frame.
- Sensors like accelerometers and gyroscope measures values with respect to the body frame while GPS measures position, ground speed with respect to the inertial frame.
- Most mission requirements like target points and flight trajectories, are specified in the local frame. In addition, map information is also given in an inertial frame of reference.

Due to this assumption, three reference frames (2 fixed and 1 mobile) are mandatory:

the first constitutes what is commonly called inertial reference frame, with origin fixed in O_i located in Earth COM, the second is the local reference frame, with origin fixed in O_l located and rigidly connected with the initial point target (obviously on the Earth surface) and lastly the body reference frame, or rather a reference frame where the origin O_b is located and rigidly connected with the geometric center of the propellers. Fistly to describe the used reference frames, a definition of a cartesian coordinete system is required .

Definition . An orthogonal frame of reference R=$\{O, i, j, k\}$(or *cartesian coordinate system*) is formed by an origin O and a set of three unitary vectors $\{i, j, k\}$ whit origin in O, that are mutually orthogonal.
Now consider a reference frame R=$\{O, i, j, k\}$ and a vector $\vec{\mathbf{r}} \in \mathbb{R}^3$, the vector can be written as linear combination of unit vectors of R as

$$\vec{\mathbf{r}} = x\,\vec{\mathbf{i}} + y\,\vec{\mathbf{j}} + z\,\vec{\mathbf{k}} \leftarrow \textit{physical vector}$$

where *x,y,z* are the coordinates. The vector **r** can also be represented as:

$$\vec{\mathbf{r}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leftarrow \textit{column vector}$$

Therefore it is necessary to set a law to express the state variables from the fixed reference to the mobile one and viceversa and for this reason, the rotation of one frame relative to the other is required. It is possible to make rotation around the axis i, j, k using a rotation matrix. Several methods to describe orientations of a rigid body in three dimensions have been developed. The first attempt to represent an orientation is attributed to Leonhard Euler[16] [17]that immagined three reference frames, rotating one around the other, and realized starting from the fixed reference frame and doing 3 rotations, he could get any other reference frame in the space. The values of these three rotation angles are called Euler angles also known as pitch, roll and yaw (ϕ , θ , ψ). They describe the orientation of a rigid body with respect to a fixed coordinate system. Any orientation can be achieved by composing three elementary rotations about the axes of a coordinate system. The angles of Euler can be defined as:

- ϕ is the angle between the x axis and the node line (N). This precession angle is defined in $[0, 2\pi]$ or in $[-\pi, \pi]$

- θ is the angle between the z and Z axes. So-called angle of nutation is defined in $[0, \pi]$ or in$[-\frac{\pi}{2}, \frac{\pi}{2}]$
- ψ is the angle between the node line and the X axis. This angle of rotation is defined in $[0, 2\pi]$ or in $[-\pi, \pi]$

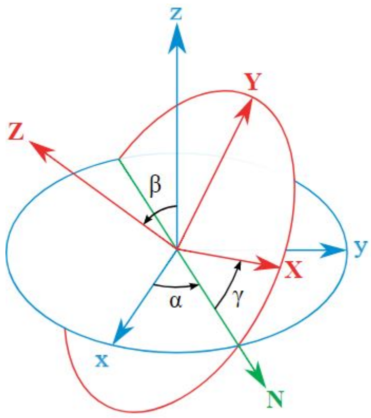

Figure 4.1: Tait-Bryan 123 Rotation

Where the line of nodes(N) are defined by the intersection of xy and XY planes. If the planes coincide, the line of the nodes N is defined as the X axis. Euler angles can be defined by three of these rotations.
Rotation about x through an angle ϕ:

$$T_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \quad where \quad \begin{cases} c_\phi = & cos(\phi) \\ s_\phi = & sin(\phi) \end{cases}$$

Rotation about y through an angle θ

$$T_2(\theta) = \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \quad where \quad \begin{cases} c_\theta = & cos(\theta) \\ s_\theta = & sin(\theta) \end{cases}$$

Rotation about z through an angle ψ

$$T_3(\psi) = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad where \quad \begin{cases} c_\psi = & cos(\psi) \\ s_\psi = & sin(\psi) \end{cases}$$

The total transformation can be seen as a sequence of three elementary rotations and can be extrinsic (rotations about the starting coordinate system, where it is assumed to remain fixed), or intrinsic (rotations about the rotating coordinate system x-y-z, rigidly connected with the body, that changes orientation after each rotation), there exist 12 possible sequences of rotation axes, divided in two groups [16]:

- Proper Euler angles
- Tait–Bryan angles

For Proper Euler angles the axes of the original frame are denoted as X,Y,Z and the axes of the rotated frame as x-y-z. The geometrical definition starts to define the line of nodes as the intersection of the planes XY and x-y.
There are 6 possibilities to choose the rotation axes for proper Euler angles. In all of them, the first and third rotation axes are the same:

- z-x'-z″ (intrinsic rotations) or z-x-z (extrinsic rotations)
- x-y'-x″ (intrinsic rotations) or x-y-x (extrinsic rotations)
- y-z'-y″ (intrinsic rotations) or y-z-y (extrinsic rotations)
- z-y'-z″ (intrinsic rotations) or z-y-z (extrinsic rotations)
- x-z'-x″ (intrinsic rotations) or x-z-x (extrinsic rotations)
- y-x'-y″ (intrinsic rotations) or y-x-y (extrinsic rotations)

While Tait–Bryan rotations represent rotations about three distinct axes (x-y-z or x-y'-z″). The three elementary rotations can be around the axes of the starting coordinate system, which remain fixed (extrinsic rotations), or around the axes of the mobile coordinate system or to change orientation after each elementary rotation (intrinsic rotations).
There are 6 possibilities to choose the rotation axes for Tait–Bryan rotation:

- x-y'-z″ (intrinsic rotations) or Z-Y-X (extrinsic rotations)

- y-z'-x″ (intrinsic rotations) or X-Z-Y (extrinsic rotations)
- z-x'-y″ (intrinsic rotations) or Y-X-Z (extrinsic rotations)
- x-z'-y″ (intrinsic rotations) or Y-Z-X (extrinsic rotations)
- z-y'-x″ (intrinsic rotations) or X-Y-Z (extrinsic rotations)
- y-x'-z″ (intrinsic rotations) or Z-X-Y (extrinsic rotations)

Sometimes, these sequences are called Euler angles, also called Cardan angles (or nautical angles, heading, elevation, and bank). The idea behind Euler rotations is to divide the overall rotation of the coordinate system into three simpler rotations[16], so-called precession, nutation, and intrinsic rotation, each one as an increment on each Euler angles. Notice that the first matrix will represent a rotation around one of the axes of the reference frame, and the final matrix represents a rotation around one of the moving frame axes. The second matrix represents a rotation around an intermediate axis called line of nodes.
However, the definition of Euler angles is not unique and in the literature many different conventions are used and conventions depend on the axes about the sequence rotations obtained.
The convention used is usually indicated by specifying the axes around consecutive rotations are done, marking them by index $(1, 2, 3)$ or letter $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$.
For this application, Tait-Bryan 123 is taken in consideration formed by the consecutive rotation around the Z, Y, X or x-y'-z":

$$T_{123} = T_1(\phi)T_2(\theta)T_3(\psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ s_\phi s_\theta c_\psi + c_\phi s_\psi & -s_\phi s_\theta s_\psi + c_\phi c_\psi & -s_\phi c_\theta \\ -c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi + s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \tag{4.1}$$

That correspond to :

Extrinsic	Intrinsic
a rotation by ψ about Z	a rotation by ϕ about x
a rotation by θ about Y	a rotation by θ about y'
a rotation by ϕ about X	a rotation by ψ about z"

Table 4.1: Extrinsic and Intrinsic rotation

4.1 Inertial reference frame

Frame of reference is called inertial if, of a isolated point mass, zero acceleration value is measured, whatever the instant in which the measurement is made and whatever the kinematic state it was of the point in the same instant t, in other words if a material point is in free motion, that is not undergoing a force or undergoing a null resultant force, therefore persist its rest state or in unvarying straight motion as long as no pertubation was actuated. The Earth is not a true system of this type, due to its revolute and rotational movement. In particular, the rotational motion undergoing the object on the surface away to the poles at a little centrifugal force. However this acceleration is negligible in some cases, for wich the Earth can be considered as an inertial frame of reference.
For this reason the ECEF is chosen as inertial frame of reference, acronym of Earth-Centered Earth-Fixed, rather than a cartesian geocentric coordinate system. The frame of reference taken in consideration can be defined as follow:
Definition. An orthogonal frame of reference $R_i = \left\{ O_i, \vec{\mathbf{i}}_i, \vec{\mathbf{j}}_i, \vec{\mathbf{k}}_i \right\}$ is is formed by an origin O_i and a set of three unitary vectors $\left\{ \vec{\mathbf{i}}_i, \vec{\mathbf{j}}_i, \vec{\mathbf{k}}_i \right\}$ with origin in O_i located in Earth COM, the axis x_i is located in the equatorial plane in direction of Greenwich meridian, axis y_i is located in the equatorial plane orthogonal respect to x_i axis., and z_i axis is defined orthogonal to the previous axes $\vec{\mathbf{k}}_i = \vec{\mathbf{i}}_i \times \vec{\mathbf{j}}_i$ in direction North-pole.

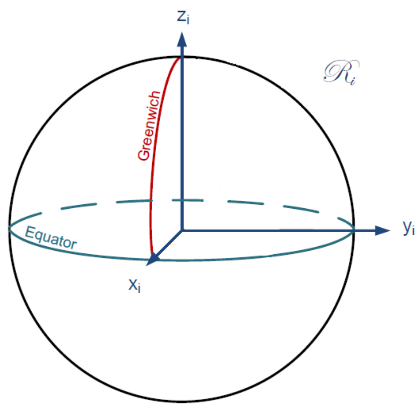

Figure 4.2: Inertial Reference frame

4.2 Local reference frame

As previously listed in the introduction of this chapter, all the requirements are expressed in the local frame of reference defined as follow:

Definition. An orthogonal frame of reference $R_l = \left\{ O_l, \vec{\mathbf{i}}_l, \vec{\mathbf{j}}_l, \vec{\mathbf{k}}_l \right\}$ is is formed by an origin O_l and a set of three unitary vectors $\left\{ \vec{\mathbf{i}}_l, \vec{\mathbf{j}}_l, \vec{\mathbf{k}}_l \right\}$ with origin in O_l located and rigidly connected with the initial point target, where x_i lies on the Earth tangent plane with direction to the Earth north, axis z_l is normal respect to the Earth tangent plane with direction opposite to the gravity $\vec{\mathbf{k}}_l = -\frac{\vec{\mathbf{g}}}{\mathbf{g}}$ in the same direction of the Zenith and axis y_l is defined orthogonal to the previous axes $\vec{\mathbf{j}}_l = \vec{\mathbf{k}}_l \times \vec{\mathbf{i}}_l$ with direction to the Earth West.
This reference frame is equal to so-called NEU (North, East, Up) rotated by π around z axis.

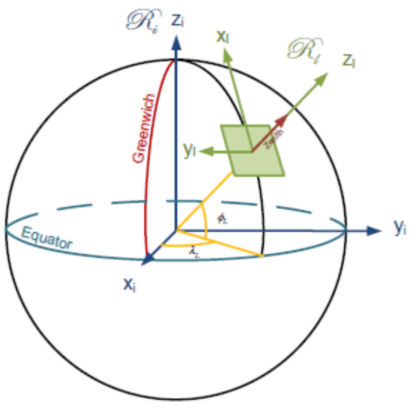

Figure 4.3: Local r. f.(green) ,Inertial r. f.(blu)

Any orientation can be achieved by composing three elemental rotations, using the following it is possible to obtain the relation between inertial frame and local frame:

$$\begin{bmatrix} x_{ECEF} \\ y_{ECEF} \\ z_{ECEF} \end{bmatrix} = \begin{bmatrix} c\lambda_l & s\lambda_l & 0 \\ -s\lambda_l & c\lambda_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi_l & 0 & s\phi_l \\ 0 & 1 & 0 \\ -s\phi_l & 0 & c\phi_l \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} \tag{4.2}$$

4.3 Body reference frame

Accelerometers and gyroscopes measure information with respect to body frame of reference, also called Aircraft Body Center (ABC). A definition of this reference frame is the following: **Definition**. An orthogonal frame of reference R_b=$\left\{O_b, \vec{\mathbf{i}}_b, \vec{\mathbf{j}}_b, \vec{\mathbf{k}}_b\right\}$ is is formed by an origin O_b and a set of three unitary vectors$\left\{\vec{\mathbf{i}}_b, \vec{\mathbf{j}}_b, \vec{\mathbf{k}}_b\right\}$with origin in O_b located and rigidly connected with the geometric center of the propeller, where axis x_b oriented as $\overrightarrow{A_1A_3}$, axis y_b is oriented as $\overrightarrow{A_2A_4}$and axis $\vec{\mathbf{k}}_b$ is defined orthogonal to the previous axes.

$$\vec{\mathbf{i}}_b = \frac{\overrightarrow{A_1A_3}}{|\overrightarrow{A_1A_3}|}$$

$$\vec{\mathbf{j}}_b = \frac{\overrightarrow{A_2A_4}}{|\overrightarrow{A_2A_4}|}$$

$$\vec{\mathbf{k}}_b = \vec{\mathbf{i}}_b \times \vec{\mathbf{j}}_b$$

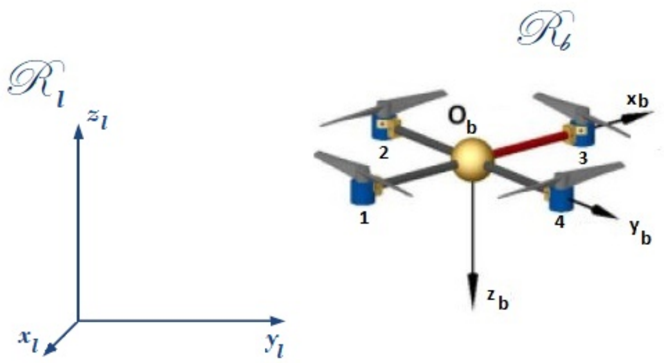

Figure 4.4: Local reference frame

Using the previously defined Tait-Bryan 1-2-3 rotation matrix as a consecutive multiplication of the 3 elementary rotation:

$$T_{123} = T_1(\phi)T_2(\theta)T_3(\psi) = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ s_\phi s_\theta c_\psi + c_\phi s_\psi & -s_\phi s_\theta s_\psi + c_\phi c_\psi & -s_\phi c_\theta \\ -c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi + s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \tag{4.3}$$

it is possible to express a vector defined in the body frame into the local reference frame that it is rotated troughout an extrinsic rotation equal to X-Y-Z .
Now, can be obtained the forces acting on the body's C.O.M that are expressed in the body frame, into the local frame by multiplying the rotation matrix to a vector w.r.t. body frame:

$$\vec{\mathbf{f}}_l = T_{123} \cdot \vec{\mathbf{f}}_b$$

Chapter 5

Quadrotor model

5.1 General description

For the description of the general case configuration with n-propellers, radially distributed on a circumference of radius r and rotated by an alpha angle with respect to the z axis, is initialy taken into consideration. The general configuration can be summarized in the Figure 5.1.

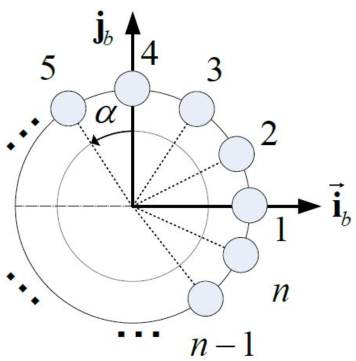

Figure 5.1: Propellers general configuration

The assembly is organized into $\alpha = 2\pi/n$ where n is the number of propellers.
Considering now a quadrotor with four propellers, the angle α must be:

$$\alpha = \frac{2\pi}{4} = \frac{\pi}{2}$$

Corresponding to propellers on the vertices of a square, whose diagonals form the interconnection between them, so defining in this way the geometric center called C_p as shown in the following Figure:

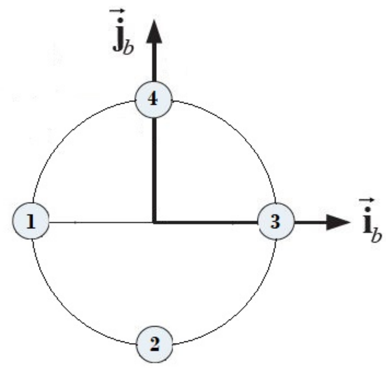

Figure 5.2: Proprellers configuration

Each propeller is composed by n_b blades with length R_b. The thrust is described through aereodynamic equation and was analyzed previously in [11] using momentum theory and blade element theory. In our case n_b is equal to 2.
In aerodynamics, the lift produces by the wing inside an air flow with velocity V is usually expressed by a non-dimensional coefficient:

$$L = \frac{1}{2}\rho\, C_L\, V^2\, S \tag{5.1}$$

where

- L is the lift force.
- ρ is the density of the air (1,225 kg/m^3 al livello del mare)
- V is the flight velocity;
- S is the wing surface;

However as known, propellers interact with air rotating, and part of the rotational kinetic energy given by the rotor speed is transformed into linear one, producing the i-th thrust command in the body frame identified by previous work [11], [31] and calculated as:

$$\overrightarrow{F}_{i\,body} = C_{wt}(R_b, n_b,)\vec{\mathbf{w}}_i^2 + C_{vt}(R_b, n_b)\vec{\mathbf{v}}_{pi} \tag{5.2}$$

where the dependence of the i-th thrust force by the square of i-th angular velocity $\mathbf{w}_i$ produced by the brushless motor and the i-th relative wind velocity $\mathbf{v}_{pi}$ is shown, while C_{wt} and C_{vt} are coefficients defined by the geometrical configuration and size of the propellers.
Since the body frame is rigidly connected to the quadrotor, and the motor direction is fixed and aligned with the z axis, the rotational velocity w_i is consider to be applied on the z axis defined in this way:

$$\vec{\mathbf{w}}_i = \begin{bmatrix} 0 \\ 0 \\ w_i \end{bmatrix} \tag{5.3}$$

and taking into account the equation (5.3) , also the vector force is written as:

$$\vec{\mathbf{F}}_{i-body} = \begin{bmatrix} 0 \\ 0 \\ f_{i-bz} \end{bmatrix} \tag{5.4}$$

With the force produced by each of the propellers, the question remains how this influences the displacement, velocity, acceleration, attitude and angular velocity of the quadrotor.
Six degrees of freedom (d.o.f.) are required to describe the dynamics of the quadrotor, three traslational and three rotational motions along the three axes. So for this reason it is necessary to take into account that the state of the quadrotor can be controlled by changing the angular velocity of the four motors. Acting on the w_i it is possible to obtain thrust and angular moment, where the roll and picth moments are caused by the difference in thrusts, yaw moment is caused by unbalancing angular velocities on the four rotors. They constitute the four input variable. Yaw moment is cancelled out when the 2 couple of motors (1-3 and 2-4) rotate in opposite direction:

- front and rear motors (i=3 and i=1) rotating counterclockwise;
- right and left motors (i=2 and i=4) rotating clockwise.

Considering the configuration and the change of rational speed of each rotor it is possible to identify four basic movements, as follows:

	HOVER / ALTITUDE CHANGE When all actuators are at equal thrust, the craft will either hold in steady hover (assuming no disturbance) or increase/decrease altitude depending the actual thrust value.

	ROLL MOMENT If one of the actuators is decreased or increased on the roll axis as compared to the other actuator on the same axis, a roll motion will occur. In this instance, the quadrotor would roll towards the right.

	PITCH MOMENT Similar to the roll axis, if either actuator is changed on the pitch axis, the axis will rotate in the direction of the smaller thrust. In this instance, the aircraft would pitch up towards the reader (out of the page) due to the differentice on the pitch axis.

	YAW MOMENT If the clockwise spinning actuators are decreased (or the counterclockwise actuators increased), a net torque will be induced on the craft resulting in a yaw angle change. In this instance, a yaw torque is induced.

Following what is listed in the table above and considering only the vertical movement, the force required to hold in steady hover or increase/decrease altitude is calculated as:

$$\vec{\mathbf{f}}_{ub} = \sum_{i=1}^{4} \vec{\mathbf{F}}_{i-body} = \begin{bmatrix} 0 \\ 0 \\ f_{ubz} \end{bmatrix}$$

while to describe lateral movement it is required to apply the classic equation to the equilibrium point, the torque due to the i-th thrust on the j-th axis in the body frame is defined as:

$$\tau_j = \sum (\mid \vec{\mathbf{r}}_i \mid \times \vec{\mathbf{F}}_{i-body})$$

that gives the possibility to calculate the torque on the y axis considering the presence of $\vec{\mathbf{F}}_1$ and $\vec{\mathbf{F}}_3$,so

$$\vec{\tau}_x = \mid \vec{\mathbf{r}}_1 \mid \times \vec{\mathbf{F}}_1 - \mid \vec{\mathbf{r}}_3 \mid \times \vec{\mathbf{F}}_3$$

where τ_x can be rewritten as:

$$\vec{\tau}_x = \vec{R} \times (\vec{\mathbf{F}}_1 - \vec{\mathbf{F}}_3) \tag{5.5}$$

Corrispondingly, the pitch torque is given by a similar expression:

$$\vec{\tau}_y = \vec{R} \times (\vec{\mathbf{F}}_2 - \vec{\mathbf{F}}_4) \tag{5.6}$$

Yaw torque is obtained considering the reactive torque produced by the motor, as:

$$\vec{\tau}_z = \sum_{i=1}^{4} (-1)^{i+1} \vec{w}_i^2$$

where the term $(-1)^{i+1}$ corresponds to a positive value for th i-th motor if it spins clockwise (i=2 and 4) and negative if it spins opposite (i=1 and 3). Thus, the total torque about the z axis is given by:

$$\vec{\tau}_z = b(+w_1^2 - w_2^2 + w_3^2 - w_4^2) = \begin{bmatrix} 0 \\ 0 \\ \tau_\psi \end{bmatrix} \tag{5.7}$$

Considering the equations (5.5)(5.6)(5.7), the total torque vector in the body frame acting on the quadcopter is expressed as:

$$\vec{\tau}_B = \begin{bmatrix} \vec{R} \times (\vec{\mathbf{F}}_1 - \vec{\mathbf{F}}_3) \\ \vec{R} \times (\vec{\mathbf{F}}_2 - \vec{\mathbf{F}}_4) \\ b(+w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{bmatrix} \tag{5.8}$$

Attitude control is done by varying the rotational velocity of the 4-propellers-motors. Considering the 6 outputs and the four inputs the quadrotor is considered as underactuated nonlinear system. So it is possible to define the inputs command.

5.1.1 Dispatching

Considering that all the inputs ($\vec{\mathbf{F}}_i, \vec{\tau}_x, \vec{\tau}_y, \vec{\tau}_z$) are expressed w.r.t. the body frame, where also each force given by i-th propeller-motor is known, a dispatch tecnique is needed to create a relationship. All the steps are solved in [36] and in previous work as[11].
Starting to define the propeller force set vectors as:

$$\vec{\mathbf{f}}_{p\,set} = \begin{bmatrix} f_{p1} \\ f_{p2} \\ f_{p3} \\ f_{p4} \end{bmatrix}$$

and considering that

$$\vec{\mathbf{f}}_{ub} = \sum_{i=1}^{4} \vec{\mathbf{F}}_{i\,body} = \begin{bmatrix} 0 \\ 0 \\ f_{1\,bz} + f_{2\,bz} + f_{3\,bz} + f_{4\,bz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{4\,bz} \end{bmatrix}$$

since each component of the propeller force set vector is expressed in the body frame and $\vec{\mathbf{f}}_{ub}$ is equal to the sum of this, the corresponding tranformation is the following:

$$\vec{\mathbf{f}}_{ub} = V \cdot \vec{\mathbf{f}}_{p\,set} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_{p1} \\ f_{p2} \\ f_{p3} \\ f_{p4} \end{bmatrix}$$

As known each propeller gives force generating torque:

$$\vec{\tau}_{Bi} = \begin{bmatrix} r \cdot sin(\alpha \cdot (i-1)) \\ -rcos(\alpha \cdot (i-1)) \\ b(-1)^i \end{bmatrix} \cdot f_{pi}$$

from this it is possible to dispatch the torque command as:

$$\vec{\tau}_B = M_t \cdot \vec{\mathbf{f}}_{pset} = \begin{bmatrix} 0 & r \cdot sin(\alpha) & r \cdot sin(2\alpha) & r \cdot sin(3\alpha) \\ -r & -rcos(\alpha) & -r \cdot cos(2\alpha) & -r \cdot cos(3\alpha) \\ -b & b & -b & b \end{bmatrix} \cdot \begin{bmatrix} f_{p1} \\ f_{p2} \\ f_{p3} \\ f_{p4} \end{bmatrix}$$

where r is the distance between the propeller and the center of mass and $\alpha = \frac{\pi}{2}$.
So considering the equation [36], to describe the overall transormation, a new vector is defined:

$$f_{zm} = \begin{bmatrix} f_{ubz} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = B_{zm} \cdot \begin{bmatrix} f_{p1} \\ f_{p2} \\ f_{p3} \\ f_{p4} \end{bmatrix}$$

where the matrix B_{zm} represents the relationship between x,y,z torques, z force and the i-th component of the propeller force set vector.
This matrix can be obtained from matrices V and M_t :

$$B_{zm} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & r \cdot sin(\alpha) & r \cdot sin(2\alpha) & r \cdot sin(3\alpha) \\ -r & -rcos(\alpha) & -r \cdot cos(2\alpha) & -r \cdot cos(3\alpha) \\ -b & b & -b & b \end{bmatrix}$$

to compute now the value of each component of f_{pset}, we define a new matrix G for fast computation :

$$G = B_{zm} \cdot B_{zm}^T = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & r(n/2) & 0 & 0 \\ 0 & 0 & r(n/2) & 0 \\ 0 & 0 & 0 & nK_m \end{bmatrix} \tag{5.9}$$

and finally inverting the equation (5.9), and considering G matrix, the vector f_{pset} is calculated as

$$\vec{\mathbf{f}}_{p\,set} = B_{zm} G^{-1} \begin{bmatrix} f_{ubz} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

That allows to obtain the following solution:

$$f_{p\,set} = \begin{bmatrix} \frac{1}{n} & -\frac{2sin(0)}{r} & -\frac{2cos(0)}{r} & -\frac{1}{n \cdot b} \\ \frac{1}{n} & -\frac{2sin(\frac{\pi}{2})}{r} & -\frac{2cos(\frac{\pi}{2})}{r} & \frac{1}{n \cdot b} \\ \frac{1}{n} & -\frac{2sin(\pi)}{r} & -\frac{2cos(\pi)}{r} & -\frac{1}{n \cdot b} \\ \frac{1}{n} & -\frac{2sin(\frac{3}{4}\pi)}{r} & -\frac{2cos(\frac{3}{4}\pi)}{r} & \frac{1}{n \cdot b} \end{bmatrix} \begin{bmatrix} f_{ubz} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{5.10}$$

5.2 Rigid body Dynamic

Rigid-body dynamics studies the movement of systems of interconnected bodies under the action of external forces. The assumption that the bodies are rigid, means that they do not deform under applied forces, simplifying the analysis by reducing the parameters that describe the system configuration [26],[34],[21]. This corresponds to considering bodies with low elasticity, and plastic deformation.

Through the laws of kinematics and Newton's second law rigid body dynamics can be desbribed. The solution that comes from these equations gives the possibility to compute position, velocity and the acceleration related to a time. The rigid body dynamics is necessary to implement simulation of mechanical systems.

Considering that the difference of the thrusts provided by each propeller, controls the attitude and therefore the state of the quadrocopter, it is also subject to the classical dynamic equations. To consider rigid body dynamics in three-dimensional space, Newton's second law must be extended defining the relationship between the movement of a rigid body and the system of forces and torques that act on it . Newton formulated his second law for a particle as written by [?],[4] The change of motion of an object is proportional to the force impressed. It is made in the direction of the straight line in which the force is impressed and to every action there is always opposed an equal reaction; or, the mutual actions of two bodies upon each other are always equal and directed to contrary parts. Because Newton generally expresses as the force equal to a mass times velocity over the time, the phrase ”change of motion” regards mass times body acceleration, and so this law is usually written as

$$\vec{\mathbf{F}} = m\vec{\mathbf{a}}$$

where $\vec{\mathbf{F}}$ is understood to be the only external force acting on the particle, m is the mass of the particle, and a is its acceleration vector. The extension of Newton’s second law to rigid bodies is achieved by considering a rigid system of particles.
As known, the dynamic equations have vector origins, therefore it is possible to express itheir values in different frames. In literature, for the drafting of the dynamic equations, local frame and the ABC (Aircraft Body Center or body) frame is used .The first is an orthogonal system with origin on the Earth surface with axes directed respectively towards North, West and the UP towards the Zenith . It is inertial and for this reason it is used to define the so-called navigation equations. The second is a reference frame rigdly connected to the rigid body with axes x,y directed along the connection beam and z perpendicularly to x,y with the same direction of the of gravity acceleration .
So throught the Newton’s second law expressed in the local reference frame , we can define the force f_l applied on the center of mass m_q.

$$\vec{\mathbf{f}}_l m_q = \vec{\mathbf{a}}_l \tag{5.11}$$

where $\vec{\mathbf{a}}_l$ is the body acceleration in the local frame:

$$\vec{\mathbf{a}}_l = \begin{bmatrix} a_{lx} \\ a_{ly} \\ a_{lz} \end{bmatrix}$$

and the force $\vec{\mathbf{f}}_l$ applied on the center of mass that is equal to:

$$\vec{\mathbf{f}}_l = \vec{\mathbf{f}}_{\mathbf{u}l} + \vec{\mathbf{f}}_{gl} + \vec{\mathbf{f}}_{dl}$$

where $\vec{\mathbf{f}}_{gl}$ is the force due to the gravity acceleration, $\vec{\mathbf{f}}_{dl}$ the disturbance force and $\vec{\mathbf{f}}_{\mathbf{u}l}$ the **input force** generated by the 4-motors-propellers set obtained from $\vec{\mathbf{f}}_{ub}$, being the input force expressed w.r.t. body:

$$\vec{\mathbf{f}}_{ub} = \sum_{i=1}^{4} \vec{\mathbf{F}}_{i\,body}$$

where $\vec{\mathbf{F}}_{i\,body}$ is the thrust force genereted by one motor throught propeller and i is the *i-th* propeller.
Due to the fact that the input force is expressed in the body reference frame, euler rotation T_{123} is used to obtain the corresponding value in the local frame:

$$\vec{\mathbf{f}}_{ul} = T_{123} \cdot \vec{\mathbf{f}}_{ub}$$

and considering that the body frame is rigly connected with the quadrotor, and the fixed direction of the propellers, only the k-th component of the input force is present:

$$\vec{\mathbf{f}}_{ub} = \begin{bmatrix} f_{ubx} \\ f_{uby} \\ f_{ubz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{ubz} \end{bmatrix}$$

where, by applying the rotation T_{123}, it is possible to find the value of input force expressed with respect to the local frame:

$$\vec{\mathbf{f}}_{ul} = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ s_\phi s_\theta c_\psi + c_\phi s_\psi & -s_\phi s_\theta s_\psi + c_\phi c_\psi & -s_\phi c_\theta \\ -c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi + s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ f_{ubz} \end{bmatrix}$$

that, once solved, becomes:

$$\vec{\mathbf{f}}_{ul} = \begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot f_{ubz}$$

Gravity force f_{gl} considered in the local reference frame, is defined as follows:

$$\vec{\mathbf{f}}_{gl} = m_q \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

About **disturbance force** f_{dl} considered in the local reference frame, is defined as follows:

$$f_{dl} = \begin{bmatrix} f_{dlx} \\ f_{dly} \\ f_{dlz} \end{bmatrix}$$

In conclusion the COM dynamics expressed in local reference frame is the following:

$$\underbrace{m_q \cdot \begin{bmatrix} a_{qlix} \\ a_{qliy} \\ a_{qliz} \end{bmatrix}}_{\vec{\mathbf{f}}_{ql}} = \underbrace{\begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot f_{ubz}}_{\vec{\mathbf{f}}_{ul}} + \underbrace{m_q \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}}_{\vec{\mathbf{f}}_{gl}} + \underbrace{\begin{bmatrix} f_{dlx} \\ f_{dly} \\ f_{dlz} \end{bmatrix}}_{\vec{\mathbf{f}}_{dl}}$$

However, due to the fact that the command force is expressed in the intermediate frame of reference, the COM dynamics is rotated through a rotation R_l^{li} as follows:

$$\underbrace{m_q \cdot \begin{bmatrix} a_{qlix} \\ a_{qliy} \\ a_{qliz} \end{bmatrix}}_{\vec{\mathbf{f}}_{ql}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot f_{ubz}}_{\vec{\mathbf{f}}_{\mathbf{u}l}} + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{m_q \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}}_{\vec{\mathbf{f}}_{gl}} + f_{dli}$$

that becomes:

$$m_q \cdot \begin{bmatrix} a_{qlix} \\ a_{qliy} \\ a_{qliz} \end{bmatrix} = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot f_{ubz} + m_q \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + f_{dli}$$

and by dividing the right-hand side by the mass m_q, it is possible to obtain the final form. This equation is explicit in acceleration and particularly useful from the control point of view:

$$\begin{bmatrix} a_{qlix} \\ a_{qliy} \\ a_{qliz} \end{bmatrix} = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot \frac{f_{ubz}}{m_q} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{f_{dli}}{m_q} \tag{5.12}$$

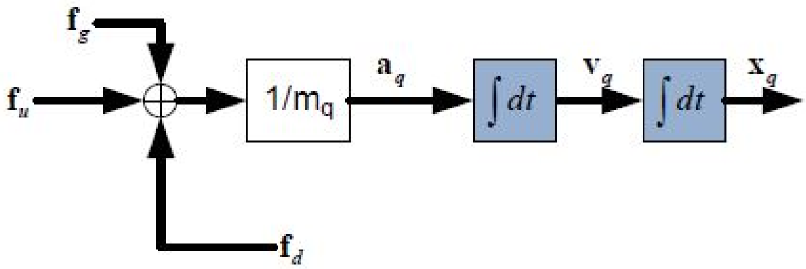

Figure 5.3: Dynamic model

5.2.1 Attitude dynamics - Euler moment equation

Using the center of mass and inertia matrix I, for a single rigid body, the force and torque equations takes the following form:

$$\vec{\mathbf{F}} = m\vec{\mathbf{a}} \tag{5.13}$$

$$\vec{M} = I\dot{\vec{\boldsymbol{\omega}}} + \vec{\boldsymbol{\omega}} \times I\vec{\boldsymbol{\omega}} \tag{5.14}$$

and are known as Newton's second law of motion for a rigid body, where when a rotating object is under the influcnce of torques, it exhibits the behaviours of precession and nutation. The fundamental equation describing the behavior of a rotating solid body gives the possibility to express the angular acceleration with respect to the torque command as:

$$\dot{\vec{\boldsymbol{\omega}}} = I^{-1} \cdot (\vec{M} - \vec{\boldsymbol{\omega}} \times I \cdot \vec{\boldsymbol{\omega}})$$

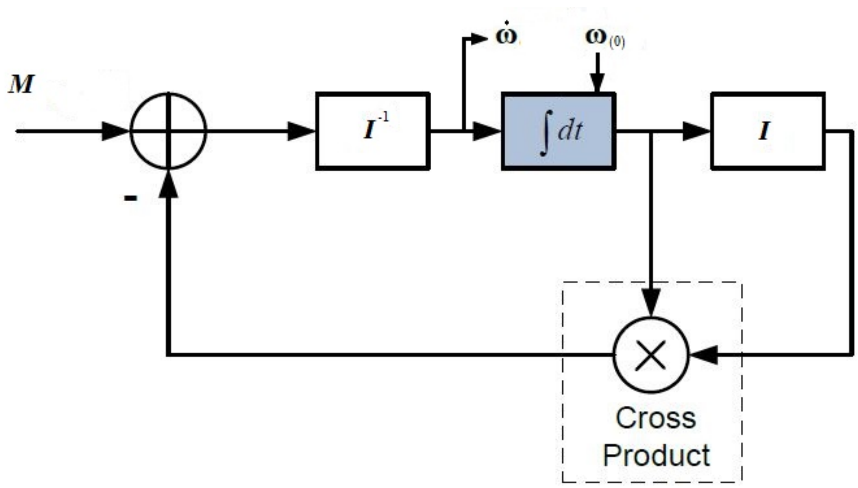

Figure 5.4: Attitude Dynamic

where $\vec{\boldsymbol{\omega}}\times$ is rewritten using the skew simmetrix matrix S

$$S = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

while the

$$\vec{M} = \vec{\tau_b} + \vec{\tau_d}$$

where $\vec{\tau_b}$ is the command torque vector that is derived from the difference in thrust in the body frame and $\vec{\tau_d}$ is the disturbance torque vector.

5.2.2 Attitude kinematics - Euler angle kinematics

To describe the evolution of the angles derivative $\dot{\phi}, \dot{\theta}, \dot{\psi}$ over the time with respect to the body angular velocity, it is necessary to see the overall rotation from the local to body as the sequence of three intrinsic elementary rotations (x-y'-z") around the mobile frame.
F_1 is rotated w.r.t. the local frame through the rotation $T_1(\phi)$ and considering $\vec{\boldsymbol{\omega}}_{\phi l} =$

$$\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{\boldsymbol{\omega}}_{\phi l} = T_1(\phi) w_{\phi 1} \longrightarrow \qquad \vec{\boldsymbol{\omega}}_{\phi 1} = T_1(\phi)^T \vec{\boldsymbol{\omega}}_{\phi l} = T_1(-\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

F_2 is rotated w.r.t. F_1 through the rotation $T_2(\theta)$ and considering $\vec{\boldsymbol{\omega}}_{\theta 1} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$

$$\vec{\boldsymbol{\omega}}_{\theta 1} = T_2(\theta) \vec{\boldsymbol{\omega}}_{\theta 2} \longrightarrow \qquad \vec{\boldsymbol{\omega}}_{\theta 2} = T_2(\theta)^T \vec{\boldsymbol{\omega}}_{\theta 1} = T_2(-\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

$$\vec{\boldsymbol{\omega}}_{\phi 2} = T_2(\theta) \vec{\boldsymbol{\omega}}_{\phi 1} \longrightarrow \quad \vec{\boldsymbol{\omega}}_{\phi 2} = T_2(\theta)^T T_1(\phi)^T \vec{\boldsymbol{\omega}}_{\phi l} = T_2(-\theta) T_1(-\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{\boldsymbol{\omega}}_{\phi l} = T_1(\phi) \vec{\boldsymbol{\omega}}_{\phi 1} \longrightarrow \qquad \vec{\boldsymbol{\omega}}_{\phi 1} = T_1(\phi)^T \vec{\boldsymbol{\omega}}_{\phi l} = T_1(-\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

F_B is rotated w.r.t. F_2 through the rotation $T_3(\psi)$ and considering $\vec{\boldsymbol{\omega}}_{\psi 2} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$

$$\vec{\boldsymbol{\omega}}_{\psi 2} = T_3(\psi) \vec{\boldsymbol{\omega}}_{zB} \longrightarrow \vec{\boldsymbol{\omega}}_{zB} = T_3(\psi)^T \vec{\boldsymbol{\omega}}_{\psi 2} = T_3(-\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\vec{\boldsymbol{\omega}}_{\phi 2} = T_3(\psi) \vec{\boldsymbol{\omega}}_{xB} \longrightarrow \vec{\boldsymbol{\omega}}_{xB} = T_3(\psi)^T T_2(\theta)^T T_1(\phi)^T \vec{\boldsymbol{\omega}}_{\phi 1} = T_3(-\psi) T_2(-\theta) T_1(-\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{\boldsymbol{\omega}}_{\theta 2} = T_3(\psi) \vec{\boldsymbol{\omega}}_{yB} \longrightarrow \vec{\boldsymbol{\omega}}_{yB} = T_3(\psi)^T T_2(\theta)^T \vec{\boldsymbol{\omega}}_{\phi 1} = T_3(-\psi) T_2(-\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

From these results, it is possible to write the angular velocity vector in the body frame knowing the derivative of the Euler angles as:

$$\vec{\boldsymbol{\omega}} = \omega_{xB}\mathbf{i} + \omega_{yB}\mathbf{j} + \omega_{zB}\mathbf{k} = \begin{bmatrix} \omega_{xB} \\ \omega_{yB} \\ \omega_{zB} \end{bmatrix}$$

That, taking the same steps into consideration, corresponds to:

$$\begin{bmatrix} \omega_{xB} \\ \omega_{yB} \\ \omega_{zB} \end{bmatrix} = T_3(-\psi)T_2(-\theta) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + T_3(-\psi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{0} \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} \omega_{xB} \\ \omega_{yB} \\ \omega_{zB} \end{bmatrix} = \begin{bmatrix} c_{(-\psi)} & s_{(-\psi)} & 0 \\ -s_{(-\psi)} & c_{(-\psi)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{(-\theta)} & 0 & -s_{(-\theta)} \\ 0 & 1 & 0 \\ s_{(-\theta)} & 0 & c_{(-\theta)} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \dots$$

$$\dots + \begin{bmatrix} c_{(-\psi)} & s_{(-\psi)} & 0 \\ -s_{(-\psi)} & c_{(-\psi)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{0} \\ 0 \\ \dot{\psi} \end{bmatrix}$$

rewriting everything in matrix form, the above equation becomes:

$$\begin{bmatrix} \omega_{xB} \\ \omega_{yB} \\ \omega_{zB} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\psi & 0 \\ -s_\psi c_\theta & c_\psi & 0 \\ s_\theta & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5.15}$$

Inverting the equation and integrating it is possible to calculate the Euler angles (ϕ, θ, ψ) from ω_{xB}, ω_{yB}, ω_{zB}.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi c_\theta & c_\theta c_\psi & 0 \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} \begin{bmatrix} \omega_{xB} \\ \omega_{yB} \\ \omega_{zB} \end{bmatrix} \tag{5.16}$$

Inverting the matrix as shown above, a singularity occurs when the denominator is 0, that corresponds to a rotation of $\theta = \pm\frac{\pi}{2}$. This phenomenon is known as Gimbal lock.

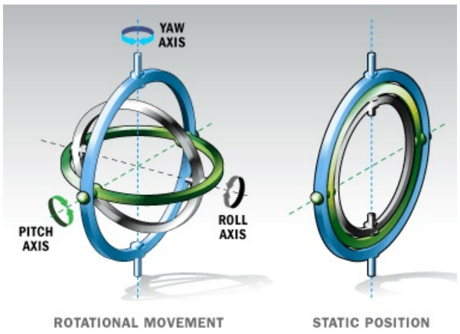

Figure 5.5: Gimbal Lock

For this reason it is useful to use quaternions to avoid this phenomenon.

5.2.3 Attitude kinematics - Quaternion kinematics

The Aim is to describe the time evolution of the rotation quaternion q in function of w_1, w_2, w_3. In order to describe the rotation and the attitude kinematics, the quaternion-kinematics representation is used.
A necessary condition is that a quaternion has a unit norm $||q|| = 1$ that is called unit quaternion. Consider now the unit quaternion:

$$\vec{\mathbf{u}} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = (u0; \boldsymbol{u}) = \begin{bmatrix} cos(\theta) \\ \boldsymbol{u}sin(\theta) \end{bmatrix}$$

that represents the rotation $Rot(u; 2\theta)$ around the axis specified by the unit vector $\boldsymbol{u} = (u_1, u_2, u_3)^T$.
The opposit is also true, i.e., given a rotation $Rot(u; \theta)$ of an angle θ around the axis specified by the unit vector $\boldsymbol{u} = (u_1, u_2, u_3)^T$, the unit quaternion:

$$\vec{\mathbf{q}} = \vec{\mathbf{u}} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} cos(\frac{\theta}{2}) \\ \boldsymbol{u}sin(\frac{\theta}{2}) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_4 \end{bmatrix}$$

To compute the rotation matrix $R(\boldsymbol{q})$ given a unit quaternion $\boldsymbol{q} = (q_0; \mathbf{q})$, we use the following relation:

$$R(\boldsymbol{q}) = \left(q_0^2 - \mathbf{q}^T \boldsymbol{\cdot} \boldsymbol{q}\right) \boldsymbol{\cdot I} + 2\boldsymbol{q} \boldsymbol{\cdot} \boldsymbol{q}^{\mathrm{T}} - 2 \cdot q_0 \cdot S(\mathbf{q})$$

where $S(\mathbf{q})$ is the skew-symmetric matrix of the unit vector **q:**

$$S(\boldsymbol{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

Quaternion representation has the advantage when it is used in kinematic differential equation, because it is possible to know the derivative of quaternion $\dot{\boldsymbol{q}}$ from the current quaternion $\boldsymbol{q}$ and the angular velocity vector $\boldsymbol{w}^q = (0, \boldsymbol{w})$ as follow:

$$\dot{\boldsymbol{q}} = \frac{1}{2} \cdot \boldsymbol{q} \otimes \boldsymbol{w}^q$$

where $\otimes$is the Hamilton product that is computed as follow:

$$\boldsymbol{q} \otimes \boldsymbol{p} = (q_0 p_0 - \mathbf{q} \boldsymbol{\cdot} \mathbf{p}) + (q_0 \boldsymbol{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p}) = \begin{bmatrix} q_o & -\mathbf{q}^T \\ \mathbf{q} & q_0\, \boldsymbol{I} + S(\mathbf{q}) \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ \mathbf{p} \end{bmatrix}$$

That could be resolved in an equivalent way as:

$$\dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{\Omega} \cdot \boldsymbol{q}$$

where Ω is a skew matrix defined as :

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -w_1 & -w_2 & -w_3 \\ w_1 & 0 & w_3 & -w_2 \\ w_2 & q_1 & 0 & w_1 \\ w_3 & w_2 & -w_1 & \end{bmatrix}$$

or similarly like

$$\dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{Q} \cdot \boldsymbol{w}$$

where $\boldsymbol{Q}$ is a matrix that takes the following form:

$$\boldsymbol{Q} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

In conclusion the dynamic and kinematic equations shown above can be seen a series connection of two nonlinear system:

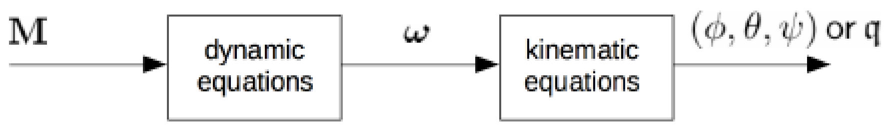

Figure 5.6: Series connection of dynamic and kinematic equation

5.3 Quadrotor embedded models

The model is obtained by approximations to the nonlinear quadrotor model, simplified using the feedback linearization method. This method corresponds to connecting the horizontal model and the attitude model through a feedback linearization trasforming the non linear problem into a linear one [11]. This connection allows to consider a unique problem avoiding the realization of two control levels. The embedded models are discrete time controlled by two inputs composed by the thrust command input calculated for each sample time k , and the noise vector, known only for the k-th instant (it is unpredictable). Since the initial model is described in continuous time, the backward euler method has been used to bring it back to its corresponding discrete form. With this method it is possible to express the variation of any variable in the time unit as the difference between the k-th sample and its previous one in the unit time which in this case corresponds exactly to the sampling time T_s. In continuous time the law that describes this variation is expressed as:

$$\dot{x} = \frac{x(t + \triangle T) - x(t)}{\triangle T} \tag{5.17}$$

In order to clarify these models, the method previously described is rewritten in the form of a diagram where digital integrators have been used for the realization:

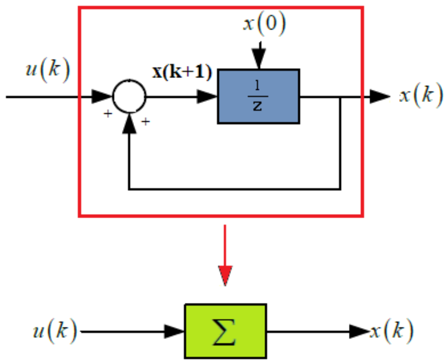

Figure 5.7: Embedded scheme

This scheme will be used for the description of embedded models that are divided into vertical, horizontal and spin dynamics. For the creation of the vertical and horizontal model, reference is made to what has been written in Ch. 5 where the dynamic of the rigid body has been described through equation (5.12) which is shown below:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}(t) = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot \frac{f_{bz}(t)}{m_q} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{f_d}{m_q}$$

Attitude control is done by varying the rotational velocity of the 4-propellers-motors. Considering the 6 outputs and the 4 inputs the quadrotor is a hightly underactuated nonlinear MIMO time-invariant system.

5.3.1 Vertical embedded model

For the realization of these models, the dynamics are divided in vertical and the horizontal so this implies that the previous equation is divided into two parts : the first one that considers the first two components of acceleration and defines the horizontal dynamic and the second that considers the third component of acceleration and defines the vertical dynamic. So, dividing this equation and using only the second part we obtain the vertical model that is expressed in continuous time form:

$$a_z(t) = \ c_\phi c_\theta \ \cdot \frac{f_{bz}(t)}{m_q} - g + \frac{f_d}{m_q} \tag{5.18}$$

Now considering the model obtained by backward euler as:

$$x(k+1) = x(k) + \triangle T \dot{x}(k) \tag{5.19}$$

and considering the continuous-time state-space vertical model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_z = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u_z(t) - \begin{bmatrix} 0 \\ 1 \end{bmatrix} g + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d \tag{5.20}$$

where the input is linearized through the association $u_z(t) = c_\phi c_\theta \cdot \frac{f_{bz}(t)}{m_q}$ and the source of disturbance is modeled by the component d acting as an input for acceleration as described in Ch (3).

Using equation (5.20) the continuous model is led back to the underlying discrete form:

$$\begin{bmatrix} x_z \\ v_z \\ D_1 \end{bmatrix}(k+1) = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x_z \\ v_z \\ D_1 \end{bmatrix}(k) + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{B} u_z(k) - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} g + \dots$$

$$\dots + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} w_{1z} \\ w_{2z} \end{bmatrix}$$

which can be rewritten in a schematic form using the embedded diagram of figure (5.8).

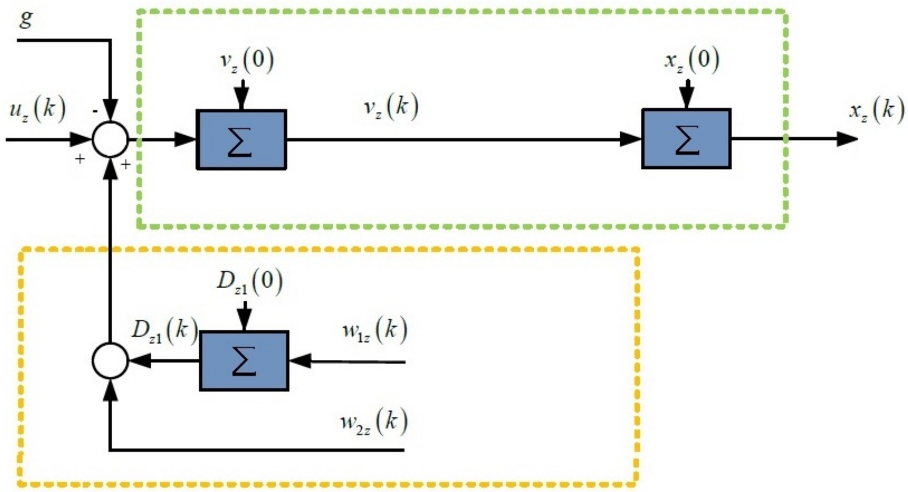

Figure 5.8: Vertical Embedded model [11]

5.3.2 Horizontal Embedded Model

The horizontal model is obtained similarly to the vertical model described in the previous paragraph. In this case the dynamics concerning the XY plane will be modeled, therefore the x and y components of the acceleration vector will be considered only:

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix}(t) = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \end{bmatrix} \cdot \frac{f_{bz}(t)}{m_q} + \frac{f_d}{m_q}$$

Since the discrete time embedded model to be created is linear, it is linearized using a state transformation.

$$\mathbf{q} = \begin{bmatrix} q_{lx} \\ q_{ly} \end{bmatrix} = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \end{bmatrix}$$

Since the attitude is controlled by the application of the torque vector, but the new state can not be controlled by this, the introduction of a new state variable is required.

$$\Omega_l = \dot{\mathbf{q}}$$

$$\Omega_l = \begin{bmatrix} \Omega_{lx} \\ \Omega_{ly} \end{bmatrix} = \begin{bmatrix} c_\phi c_\theta \dot{\phi} - s_\phi s_\theta \dot{\theta} \\ c_\theta \dot{\theta} \end{bmatrix} = \begin{bmatrix} c_\phi c_\theta & -s_\phi s_\theta \\ 0 & c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}$$

In the horizontal dynamics, the disturbance signals are present not only as acceleration but also as torque. The disturbance dynamics is set to three, considering what we have found now, the complete horizontal embedded model takes the following form:

$$\begin{bmatrix} x \\ v_x \\ q_x \\ \Omega_l \\ D_1 \\ D_2 \\ D_3 \end{bmatrix}(k+1) = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ v_x \\ q_x \\ \Omega_l \\ D_1 \\ D_2 \\ D_3 \end{bmatrix}(k) + \ldots$$

$$\dots + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_x} u_x(k) + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_{1x} \\ w_{2x} \\ w_{3x} \\ w_{4x} \\ w_{5x} \end{bmatrix}$$

where $\alpha = f_{bz}(k)$ is a time varying parameter. The system is MIMO (Multiple Input Multiple Output) considering its 6 inputs and 7 outputs. This model is implemented in the simulator using the Embedded diagrams as shown in the following figure:

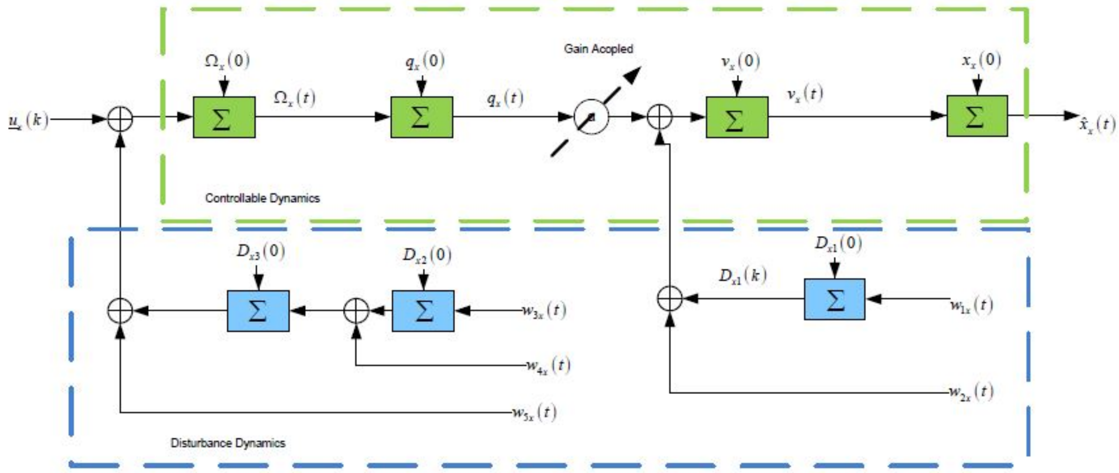

Figure 5.9: Horizontal EM scheme [11]

5.3.3 Spin embedded model

The embedded model for spin is described separately as:

$$\begin{bmatrix} \psi \\ \Gamma \\ D_{\phi 1} \end{bmatrix} (k+1) = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} \psi \\ \Gamma \\ D_{\phi 1} \end{bmatrix} (k) + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{B_x} u_x(k) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{1x} \\ w_{2x} \end{bmatrix}$$

Chapter 6

Trajectory planning strategies

In this chapter will be described trajectory generation methodologies, based on two different tecnique formulated as optimization problem. The purpose of this development is to provide the necessary references that allows to achieve our quadrocopter in any target position starting to initial one [32]. The first method shown is related to the pre-existing polynomial guidance algorithm with fixed time of fly and defined in continuos time. It is based on a reconstruction of the polynomial functions that respectively describe position velocity and acceleration. This reconstruction is carried out considering boundary condition (initial and final) on all the states, optimizing the polynomial functions coefficients . To optimize this guidance will be consider a time interval that could have the final value of time and therefore no longer a fixed one. The method used to optimize the problem is based on the minimization of an **objective function** also known under the names of **performance index** or **cost function** using the weighed standard method. Polynomial Guidance allows our system to evolve until reaching the target, in very short calculation times, but using this approach it is not possible to take into account other important things such as dynamics respecting input saturation or keeping in mind the presence of obstacles. To achieve this aim is necessary to develop a new guidance algorithm completely described in discrete time and which respects not only the dynamics of the system but also the limits that characterize the outputs and saturation for inputs . The main requirements on which our approach is based could be represented sequentially in 4 steps as follow :

- The generated trajectory must satisfy the dynamic and input constraints finding feasible solution.

- The generated trajectory bring the quadrotor to the target position such that, picth and roll angles are minimized over all the trajectory.
- The calculation must be fast enougth for online calculation.
- The optimization must be calculated for each sample time, considering an implicit control low, and applying only the first element of the sequence vector $u*$.

This optimization approach, implemented in the next paragraphs, directly incorporates output (forbidden fly zone, initial and final position), input and the dynamic constraints of the quadrotor controlled by four command inputs. As known, in the real case the inputs can not have an infinite value due to the fact that the components used have finite physical limits (motor speed) and for this reason the optimization problem will be subject to inpunt constraints that take into account inputs saturation. The presence of constraints are hard limits to be respected and that often make the problem difficult to solve. In fact, feasibility of the generated trajectory is guarateed by limiting trajectory acceleration such that the actual control inputs do not saturate and ouput do not exceed the desidered value.

6.1 Continuos time optimization problem

6.1.1 Polynomial Unconstrained Optimization Problem

To optimize the already present polynomial guidance we refers to what was written in [37][11] . The following optimization problem became to the improvement performance over the basic polynomial guidance that consider now, not a fixed value for time of flight but as a free-parameter $t_f \in [t_{min}\, , \, t_{max}]$. The problem could be posed describing the 5^{th} derivative of the position or in better way the second derivative of the acceleration so called **crackle** expressed in general polynomial form:

$$\boldsymbol{\alpha}(t) = C_0 + C_1 t + ... \ + C_N t^N \tag{6.1}$$

where $N \geq 3$.
The problem is posed by considering N=4 , so the cranckle profile is rewritten as:

$$\boldsymbol{\alpha}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = C_0 + C_1 t + C_2 t^2 + C_3 t^3 + C_4 t^4 \tag{6.2}$$

The states track a trajectory as is shown in the following equation

$$\underbrace{\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\Omega} \\ \boldsymbol{a} \\ \boldsymbol{v} \\ \boldsymbol{x} \end{bmatrix}(t)}_{\boldsymbol{x}(t)} = \underbrace{\begin{bmatrix} \boldsymbol{\alpha}(0) \\ \boldsymbol{\Omega}(0) + \boldsymbol{\alpha}(0)t \\ \boldsymbol{a}(0) + \boldsymbol{\Omega}(0)t + \boldsymbol{\alpha}(0)t^2/2 \\ \boldsymbol{v}(0) + \boldsymbol{a}(0)t + \boldsymbol{\Omega}(0)t^2/2 + \boldsymbol{\alpha}(0)t^3/6 \\ \boldsymbol{x}(0) + \boldsymbol{v}(0)t + \boldsymbol{a}(0)t^2/2 + \boldsymbol{\Omega}(0)t^3/6 + \boldsymbol{\alpha}(0)t^4/24 \end{bmatrix}}_{\boldsymbol{x_0}(t)} + \dots$$

$$\dots + \underbrace{\begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \\ t & \frac{t^2}{2} & \frac{t^3}{3} & \frac{t^4}{4} & \frac{t^5}{5} \\ \frac{t^2}{2} & \frac{t^3}{6} & \frac{t^4}{12} & \frac{t^5}{20} & \frac{t^6}{30} \\ \frac{t^3}{6} & \frac{t^4}{24} & \frac{t^5}{60} & \frac{t^6}{120} & \frac{t^7}{210} \\ \frac{t^4}{24} & \frac{t^5}{120} & \frac{t^6}{360} & \frac{t^7}{840} & \frac{t^8}{1680} \end{bmatrix}}_{\boldsymbol{A}(t)} \cdot \underbrace{\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}}_{C}$$

Where we have defined with $\boldsymbol{\Omega}$ the jerk state, $\boldsymbol{a}$ the acceleration,$\boldsymbol{v} the velocity$,$\boldsymbol{x}$ the position.
The equation above can be rewritten in the compact form:

$$\boldsymbol{x}(t) = \boldsymbol{x_0}(t) + \boldsymbol{A}(t) \cdot \boldsymbol{C} \tag{6.3}$$

This show that the profile of the state depend by the initial value of the sistem and its dynamic.
The aim of the problem is to find the coefficients that describe in exact way all the state profile in order to satisfy the terminal boundary conditions. The method used to

optimize the problem is based on the minimization of an objective function J_E where in our case could be choosen to minimize the following cost function:

$$J_E = \int_0^{t_f} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \, dt \tag{6.4}$$

That is subject to an equality constranit given by the equation (6.3) where implies that the coefficients $(C_0, C_1, C_2, C_3, C_4)$ must satisfy the following equation:

$$\boldsymbol{A}(t) \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \boldsymbol{x}(t) - \boldsymbol{x_0}(t) \tag{6.5}$$

That obviously correspond to the following general form for the equality constraint:

$$\boldsymbol{A}x = \mathbf{b}$$

Now to take some trasformation usefull to split and reformulate the problem, the crankle profile is rewrittes as the multiplication of:

$$\boldsymbol{\alpha}(t) = \boldsymbol{\Phi}^T \boldsymbol{C}$$

where $\boldsymbol{\Phi}^T = \begin{bmatrix} 1 & t & ... & t^N \end{bmatrix}$ that can be substituted in the quadratic objective function and it returns:

$$J_E = C^T S(t_f) C \tag{6.6}$$

with $S(t_f) = \int_0^{t_f} \boldsymbol{\Phi}^T \boldsymbol{\Phi} \, dt$.
The optimization problem became, considering linear quadratic function e linear equality constraint:

$$\underset{(C_0, C_1, C_2, C_3, C_4)}{min} J_E = \int_0^{t_f} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \, dt$$

$$subject\ to \quad \boldsymbol{A}(t_f)\boldsymbol{C} = \boldsymbol{b}(t_f)$$

where

$$\boldsymbol{A}(t_f) = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \\ t & \frac{t^2}{2} & \frac{t^3}{3} & \frac{t^4}{4} & \frac{t^5}{5} \\ \frac{t^2}{2} & \frac{t^3}{6} & \frac{t^4}{12} & \frac{t^5}{20} & \frac{t^6}{30} \\ \frac{t^3}{6} & \frac{t^4}{24} & \frac{t^5}{60} & \frac{t^6}{120} & \frac{t^7}{210} \\ \frac{t^4}{24} & \frac{t^5}{120} & \frac{t^6}{360} & \frac{t^7}{840} & \frac{t^8}{1680} \end{bmatrix}$$

and $\boldsymbol{b}(t_f) = \boldsymbol{x}(t) - \boldsymbol{x_0}(t)$.
The optimization problem exposed correspond to the so-called standard minimum weighted norm and can be solved as:

$$C(t_f) = S^{-1}A^T(AS^{-1}A^T)^{-1}b \tag{6.7}$$

Thanks to this solution is possible to reconstruct the polinomial form of each state and the respective command input. In such a way to use the the Polynomial guidance is needed to discretize the polynomial inputs previously to apply it inasmuch the overall system is described in discrete time. Following this procedure can be express the overall algorithm as:

Algorithm 1 Polynomial algorithm

$k \leftarrow 0$
for $k \leftarrow 1, N$ **do**
Require: $\mathbf{x}_0$ *and* $\mathbf{x}_f$ ▷ Initial and target Condition
  minimize J_E ▷ Standard minimum weighted norm
Ensure: $\mathbf{A}x_c = \mathbf{b}$
  Update Profile ▷ Polynomial Reconstruction
  Calculate $\tau(t)$ ▷ Torque command
  Discretize ▷ Profile and Torque command
end for

So the optimization problem is runned for all the value of $t_f \in [t_{min}\,,\, t_{max}]$ choosing the coefficient that correspond to a minimun value of the objective function.
However, the higher order polynomial guidance laws are only guaranteed to satisfy the terminal boundary condition *a priori*, and obvioustly don't take into saturation of input. For this reason a constrained guidance are developed in the following section.

6.1.2 Results

Implementing the knowledge described in the previous paragraph is possible to obtain the optimal solution t^*that give feaseble trajectory [37]. First of all is necessary to calculate the matrix of the coefficients C for each $t_f \in [t_{min}\ , \ t_{max}]$ using the equation (6.7) that is shown in the following figure:

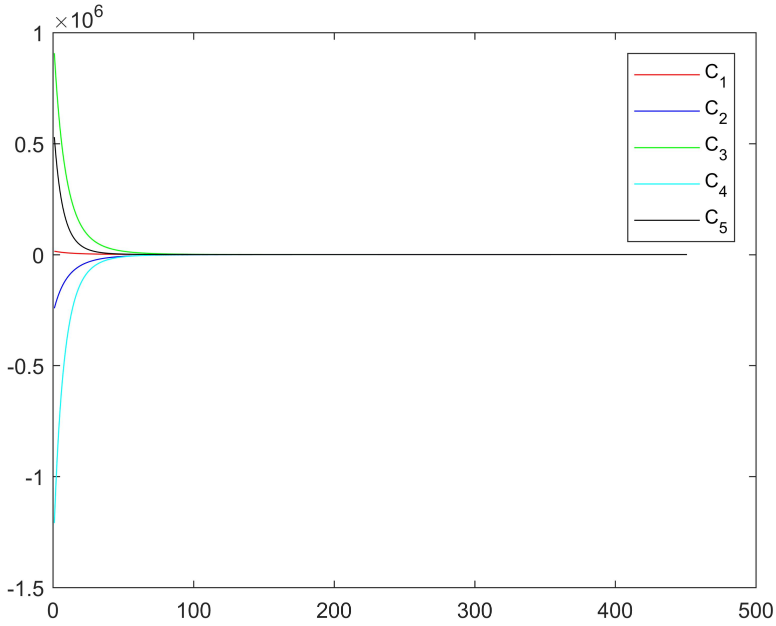

Figure 6.1: C(t_f) with $t_f \in [t_{min}\ , \ t_{max}]$

As can be seen the value of the coefficent converge to a 0 value for t that goes to t_{max}. From this result is possible compute the cost function trought the equation:

$$J_E = C^T S(t_f) C$$

that for all $C(t_f)$ with $t_f \in [t_{min}\ , \ t_{max}]$ where $t_{min} = 1$; $t_{max} = 10$; gives the following responce:

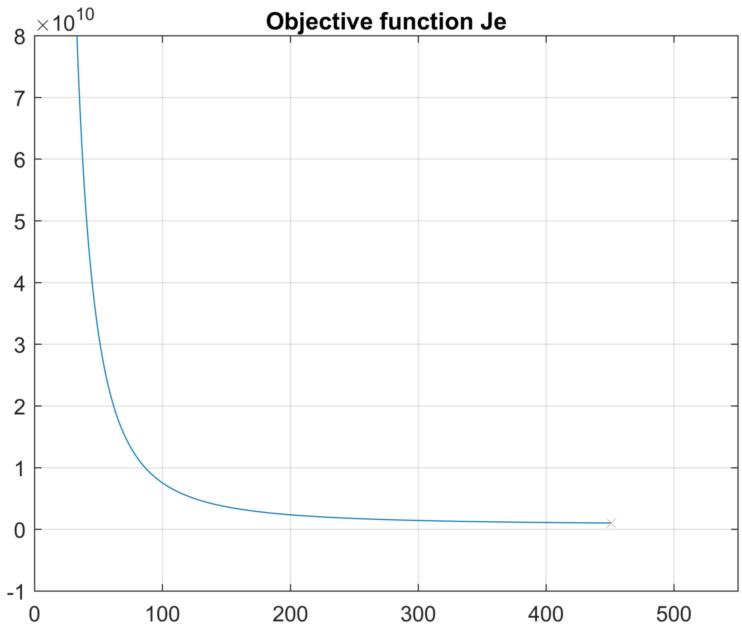

Figure 6.2: Cost function J_E

That show the minimum for $t_f^* = 10$ due to the fact that $tf_{min} = 1$; $tf_{max} = 10$; and the sampling time $Ts = 0.02$;
From the result obtained above is possible to obtain the optimal vector $C^*_{(t_f)}$:

$$C^*(10) = S^{-1}A^T(AS^{-1}A^T)^{-1}b = \begin{bmatrix} 1.6800 \\ -2.6880 \\ 1.0080 \\ -0.1344 \\ 0.0059 \end{bmatrix}$$

and using this optimal value for the computation of the acceleration vector is possible to obatin the feaseble trajectory as follow:

$$\boldsymbol{\alpha}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1.6800 \\ -2.6880 \\ 1.0080 \\ -0.1344 \\ 0.0059 \end{bmatrix} = 1.68 - 2.688 \cdot t + 1.008 \cdot t^2 - 0.1344 \cdot t^3 + 0.0059 \cdot t^4$$

The states track a trajectory as is show in the following equation:

$$\underbrace{\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\Omega} \\ \boldsymbol{a} \\ \boldsymbol{v} \\ \boldsymbol{x} \end{bmatrix}(t)}_{\boldsymbol{x}(t)} = \underbrace{\begin{bmatrix} \boldsymbol{\alpha}(0) \\ \boldsymbol{\Omega}(0) + \boldsymbol{\alpha}(0)t \\ \boldsymbol{a}(0) + \boldsymbol{\Omega}(0)t + \boldsymbol{\alpha}(0)t^2/2 \\ \boldsymbol{v}(0) + \boldsymbol{a}(0)t + \boldsymbol{\Omega}(0)t^2/2 + \boldsymbol{\alpha}(0)t^3/6 \\ \boldsymbol{x}(0) + \boldsymbol{v}(0)t + \boldsymbol{a}(0)t^2/2 + \boldsymbol{\Omega}(0)t^3/6 + \boldsymbol{\alpha}(0)t^4/24 \end{bmatrix}}_{\boldsymbol{x_0}(t)} + \dots$$

$$\dots + \underbrace{\begin{bmatrix} 1 & t & t^2 & t^3 & t^4 \\ t & \frac{t^2}{2} & \frac{t^3}{3} & \frac{t^4}{4} & \frac{t^5}{5} \\ \frac{t^2}{2} & \frac{t^3}{6} & \frac{t^4}{12} & \frac{t^5}{20} & \frac{t^6}{30} \\ \frac{t^3}{6} & \frac{t^4}{24} & \frac{t^5}{60} & \frac{t^6}{120} & \frac{t^7}{210} \\ \frac{t^4}{24} & \frac{t^5}{120} & \frac{t^6}{360} & \frac{t^7}{840} & \frac{t^8}{1680} \end{bmatrix}}_{\boldsymbol{A}(t)} \cdot \underbrace{\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}}_{C}$$

for each $t \in [0\,,\, t_f^*]$, all the profile are shown below.

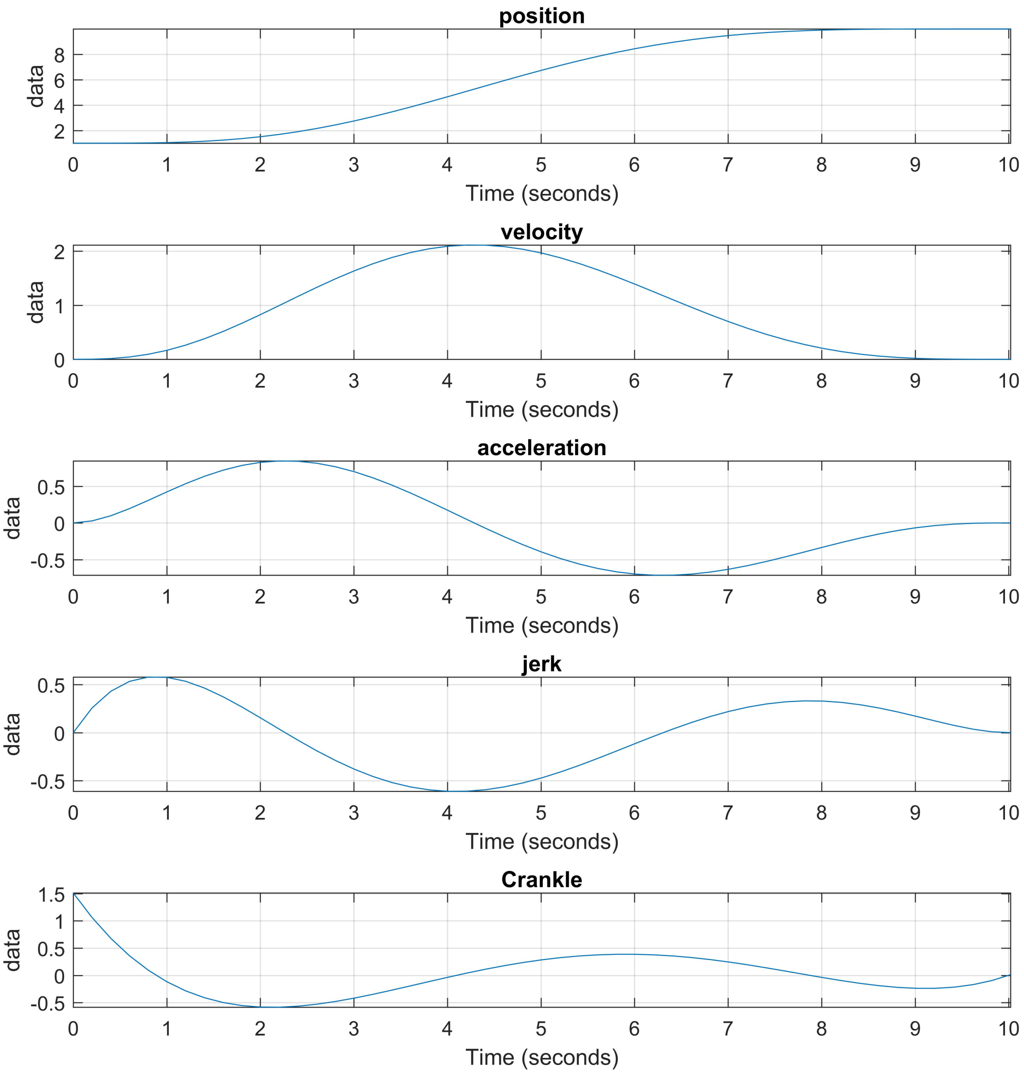

Figure 6.3: Profiles

And finally the command reference is reconstruct following the equation below :

$$\tau(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \gamma_1/t_f \\ \gamma_2/t_f^2 \\ \gamma_3/t_f^3 \\ \gamma_4/t_f^4 \end{bmatrix} 24\Delta x/t_f^4 \tag{6.8}$$

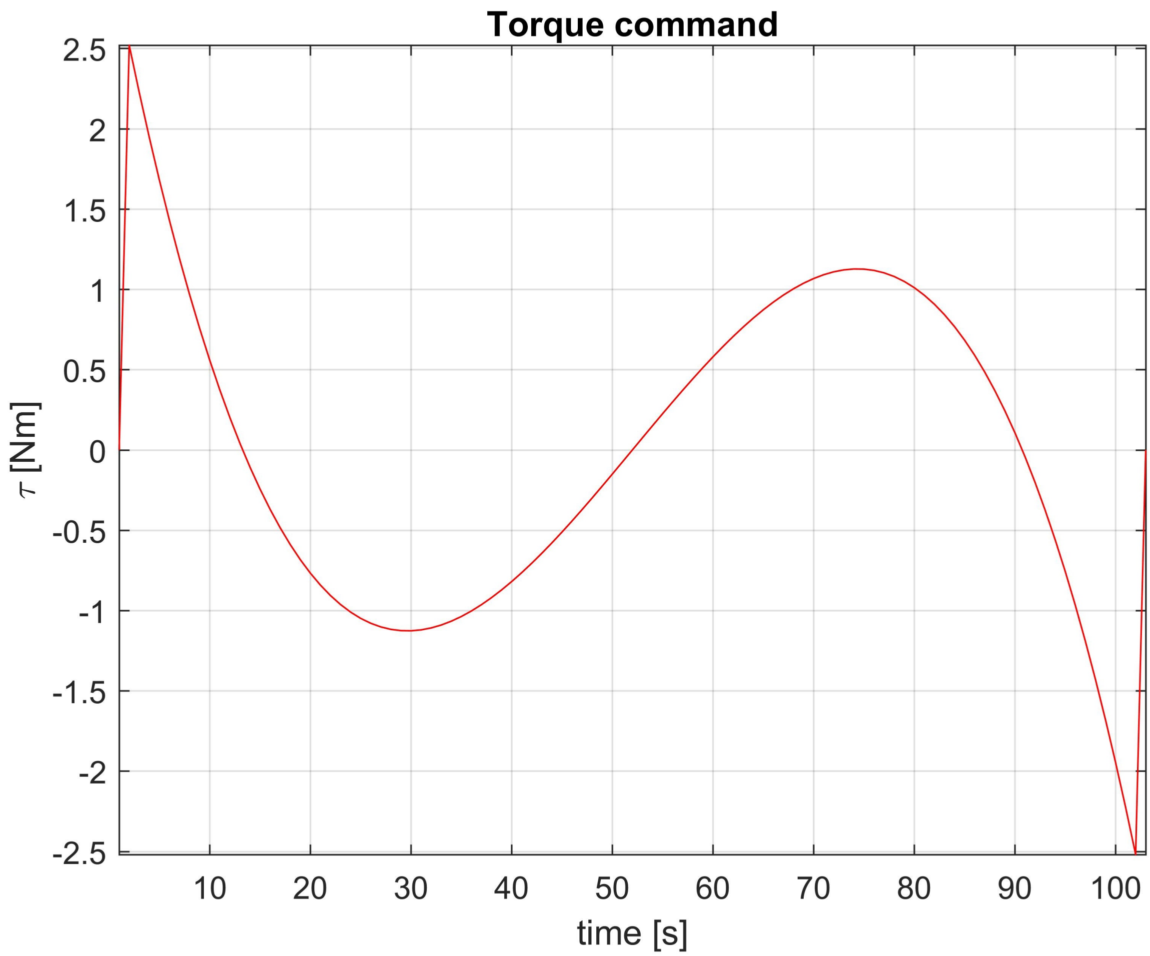

Figure 6.4: Torque command

where gives the possibility to calculate the angles.

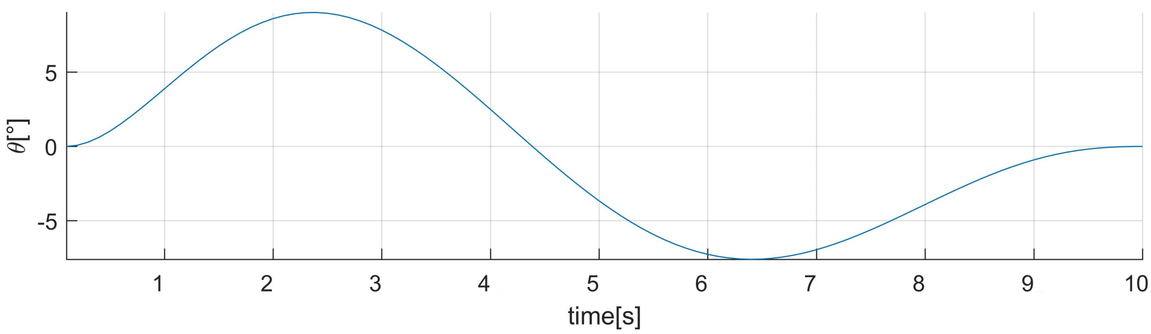

Figure 6.5: Angle θ

6.2 Discrete time Constrained optimization problem

In this subsection is described the mathematical formulation of a discrete time trajectory optimization for a UAV on the earth with uniform gravitational field. The problem formulation involves an optimal control (discrete time) problem with cost function[32], translational dynamics, state and control constraints, that allows to plan the trajectory from any initial state to a target. The main requirements to satisfy are:

- The trajectory must be feasible subject to dynamic, input and output constraints that takes into account inputs saturation and phisical bounds for the outputs.
- The trajectory calculation must be fast enough to be used on-board.
- The input controlled are the thrust and torque moments.
- It must be possible to generate an implicit feedback control law by replanning the trajectory at each time step, and applying only the first input element of the optimal sequence vector u^*.

The aim is to find, the optimal control sequence $u^* = \left\{ u_0^*, \ u_1^*, \ \ldots, \ u_N^* \right\}$ and the corrispondig optimal state sequence $x^* = \left\{ x_0^*, \ {x_1}^*, \ \ldots, \ x_N^* \right\}$. So considering what has just been written, the technique that best suits our purpose is the one based on linear quadratic discrete time constraind optimal control that is known under the name of Model Predictive Control (MPC). In constrained (and discrete time) optimal control problem with finite prediction horizon, there does not exist any simple closed-form expression for the solution. The main idea to build a closed loop is to apply only the first control move of optimal sequence u^* to the system and resolve a new finite horizon problem for each time step. In this chapter is shown how set fixed horizon optimal problem using quadratic cost function and linear constraints trought quadratic program, where is taken into accout the following linear, time-invariant system:

$$x_{k+1} = Ax_k + Bu_k$$

$$y_{k+1} = Cx_k + d_k$$

where $x_k \in \mathbb{R}^n$ define the state vector, $u_k \in \mathbb{R}^m$ define control input and $y_k \in \mathbb{R}^m$

define the output while $d_k \in \mathbb{R}^m$ define the time-varyng output disturbance.
Furthermore, are assumed that matrices A, B, C allows to stabilize e detectabilize the system. So the aim of the optimization problem is to minimize the cost function regulating the control sequence vector that brings the state vector to a desired value. For quadratic programming the performance index is generally defined as:

$$J(x(0), u) = x'(N)Q_N x(N) + \sum_{k=1}^{N}(\ x'(k)Qx(k) + u'(k)Ru(k))$$

- N= Prediction time horizon
- the first term mesure the deviation of the final state with respect to desired value x_{target}
- the second measure the deviation of the state vector x with respect to desired value x_{target}
- the third mesure the intensity of the control input (actuator authority)

while R, Q, Q_t are parameters to tune, with explict physical/economic meaning.
In conclusion the minimization problem is subject to a input and output constraints that sometimes if the prediction horizon is restricted, gives infesible solution. They are expressed as:

$$u_{min} \leq u(k) \leq u_{max} \ \ \forall k \in [1; N] \tag{6.9}$$

$$x_{min} \leq x(k) \leq x_{max} \ \ \forall k \in [1; N] \tag{6.10}$$

Moreover initial and final condition must be taken in consideration

$$x(1) = x_0 \tag{6.11}$$

$$x(N) = x_{target} \tag{6.12}$$

To satisfy the requirements to which the trajectory optimization problem will be subject, the use of advanced control techniques is strictly mandatory. The optimization of the trajectory is a topic widely dealt with in literature and that has undergone considerable changes over the years. As written in Chapter 4, the guidance is the component that deals with the generation of reference profiles through an optimization process. An indication of the technique to be used come from itself by as the optimization problem is posed. Therefore, making the some considerations, it was possible to identify two approaches for solving the same problem, both based on the MPC control technique, but using different approach: 1- Two Boundary Value Problem (T.B.V.P.) : fixed limit conditions (initial and final) that defines Two boundary condition for the problem. 2-Tracking: exploits the optimization properties, minimizing the difference between the state vector and its reference, thus leaving the system much freer to evolve.

Moreover, in such a way to avoid high non linearity of the system, and in such a way to link the linear quadratic formulation of the optimization problem with the aim of our problem will be exploits relation that link accelerations and angles through a tecnique so-called differential flatness.

6.2.1 Two Boundary Value Problem (T.B.V.P.) Constrained using modificated Model Predictive Control (M.P.C.)

The first technique that is introduced to solve the trajectory optimization problem is known under the name of Two Boundary Value Problem (T.B.V.P.) or simply targetting, where the problem is set imposing at a certain instant the achievement of a target. In the literature various examples shown how to satisfy all the requirements and then solve a problem of this type. Usually the consideration of boundary conditions as well as the inclusion of bounds on input and output may be causes infeasibility solution. However, the techniques described above solve linear problems and therefore it is not possible to use them to find the solution for non-linear problem as well as one of this type. So to arrive a solution, a strategy is requested to link the non linear system with linear optimization problem. The optimization method will be posed as describe above, however nonlinear dynamic are present in the model and this must be avoided to use linear optimization. This aim is the key point to design correctly a trajectory problem.
The standar form of optimization problem, formulated as TBVP is posed through discrete time optimal control problem, involving cost function, quadrotor dynamics, states and control constraints. It can be summarized using the classical notation of optimization problem as:

$$\min_{u(k)} \ ||\vec{\phi}||^2 + ||\vec{\theta}||^2 \tag{6.13}$$

$$subject\ to \quad \underbrace{x(k+1) = f(x(k), u(k))}_{non\ linear\ dynamic} \quad k = 1, 2, ..., N \tag{6.14}$$

$$u_{min} \leq u(k) \leq u_{max} \ \forall k \in [1; N] \tag{6.15}$$

$$x_{min} \leq x(k) \leq x_{max} \ \forall k \in [1; N] \tag{6.16}$$

$$x(1) = x_0 \tag{6.17}$$

$$x(N) = x_{target} \tag{6.18}$$

where "min" considers minimization problem of objective function formed by the square norm of sequence vector $\vec{\phi} = \left\{ \ \phi_0, \ \ \phi_1, \ \ ..., \ \ \phi_N \ \right\}$, $\vec{\theta} = \left\{ \ \theta_0, \ \ \theta_1, \ \ ..., \ \ \theta_N \ \right\}$.
The problem formulated as above correspond to a Two Boundary Value Problem (TBVP) limits by bounds on input and boundary condition and non linear dynamic constraints for outputs.
The precence of this latest one do not allows to use linear formulation and according to what written precedently, a strategy is requested to express a linear quadratic formulation involving linear constraints. To reach this purpose the current objective function that consider the minimization of angles is rewritten in such a way to avoid non linearity using **differential flatness** approach [41], where is shown that acceleration vector depend directly to ϕ and θ angles. To use this approach is mandatory to take into account the equation (5.12) that describe translation dynamics w.r.t. local reference:

$$\vec{\mathbf{a}}_l(t) = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot \frac{f_{ubz}(t)}{m_q} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

The solution can be obtained considering 2^3 combination of boundary conditions on the acceleration vector as:

1. if $a_x = 0$, $a_y \neq 0$, $a_z \neq 0$
   $a_x = sin(\phi)cos(\theta) \cdot f_{ubz} = 0$ gives that $sin(\phi) = 0$ implying $\phi = 0$
   $a_y = sin(\theta) \cdot f_{ubz}$
   $a_z = cos(\phi)cos(\theta) \cdot f_{ubz} - g = cos(\theta) \cdot f_{ubz} - g$ where $cos(\phi) = 1$
   gives the solution $\theta = tan^{-1}\left(\frac{a_y}{a_z+g}\right)$
2. if $a_x \neq 0$, $a_y = 0$, $a_z \neq 0$ that gives $sin(\theta) = 0$ that gives $\theta = 0$
   $\phi = tan^{-1}\left(\frac{a_x}{a_z+g}\right)$
3. if $a_x \neq 0$, $a_y \neq 0$, $a_z = 0$
   $\phi = tan^{-1}\left(\frac{a_x}{g}\right)$
   $\theta = tan^{-1}\left(\frac{a_y}{a_z+g} \cdot sin(\phi)\right)$
4. if $a_x \neq 0$, $a_y \neq 0$, $a_z \neq 0$
   $\phi = tan^{-1}(\frac{a_x}{a_z+g})$
   $\theta = tan^{-1}\left(\frac{a_y}{a_z+g} \cdot cos(\phi)\right)$
5. if $a_x = 0$, $a_y = 0$, $a_z = 0$
   $\phi = 0$
   $\theta = 0$
6. if $a_x \neq 0$, $a_y = 0$, $a_z = 0$ implying $\theta = 0$
   $\phi = tan^{-1}\left(\frac{a_x}{g}\right)$
7. if $a_x = 0$, $a_y \neq 0$, $a_z = 0$ implying $\phi = 0$
   $\theta = tan^{-1}\left(\frac{a_y}{g}\right)$
8. if $a_x = 0$, $a_y = 0$, $a_z \neq 0$
   $\phi = 0$
   $\theta = 0$

This solution allows to formulate a new cost funcion that involves accelleration e not the angles ϕ and θ. Inasmuch, this angles depends to the arcotangent and if the aim is to minimize it, is mandatory to minimize the argument, minimizing the numerator and maximizing the denominator, that correspond at same time to minimize the difference between the three component of acceleration vector. Then, this allows to formulate the new cost function as:

$$min \ \ ||(\overrightarrow{a_x} - \overrightarrow{a_z})||^2 + ||(\overrightarrow{a_y} - \overrightarrow{a_x})||^2 \tag{6.19}$$

Where $\overrightarrow{a_x}$, $\overrightarrow{a_y}$ and $\overrightarrow{a_z}$ $\in \mathbb{R}^{1\times N}$. Having obtained what needed, let's move on to the description of the dynamics to which optimization is subject. Now to describe the dynamics, the linear model are considered as previously done for Embedded Model (EM) which take into account a discrete-time formulation. So, to obtain translation dynamics is taken into account the equation () wrt local frame for each axes as:

$$\begin{bmatrix} x \\ v_x \\ a_x \end{bmatrix}(k+1) = \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A_x} \begin{bmatrix} x \\ v_x \end{bmatrix}(k) + \underbrace{\begin{bmatrix} 0 \\ T_s \end{bmatrix}}_{B_x} u_x(k) \tag{6.20}$$

$$\begin{bmatrix} y \\ v_y \end{bmatrix}(k+1) = \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A_y} \begin{bmatrix} y \\ v_y \end{bmatrix}(k) + \underbrace{\begin{bmatrix} 0 \\ T_s \end{bmatrix}}_{B_y} u_y(k) \tag{6.21}$$

$$\begin{bmatrix} z \\ v_z \end{bmatrix}(k+1) = \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A_z} \begin{bmatrix} z \\ v_z \end{bmatrix}(k) + \underbrace{\begin{bmatrix} 0 \\ T_s \end{bmatrix}}_{B_z} u_z(k) \tag{6.22}$$

That can be rewritten in compact form as:

$$\underbrace{\begin{bmatrix} x \\ v_x \\ y \\ v_y \\ z \\ v_z \end{bmatrix}}_{\mathbf{x}}(k+1) = \underbrace{\begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ v_x \\ y \\ v_y \\ z \\ v_z \end{bmatrix}(k) + \underbrace{\begin{bmatrix} 0 \\ T_s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_x} u_x(k) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ T_s \\ 0 \\ 0 \end{bmatrix}}_{B_y} u_y(k) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ T_s \end{bmatrix}}_{B_z} u_z(x) \tag{6.23}$$

Now, in such a way to describe dynamic constraint as equality constraint for each time step, the method described in [41] is taken into account to eliminate the dipendence of $x(k)$. This method is so-called **Batch approach**:

$x(1) = x0$

$x(2) = Ax(1) + B_x u_x(1) + B_y u_y(1) + B_z u_z(1)$

$x(3) = Ax(2) + B_x u_x(2) + B_y u_y(2) + B_z u_z(2)$

$x(4) = Ax(3) + B_x u_x(3) + B_y u_y(3) + B_y u_z(3)$

$\dots$

$x(N+1) = Ax(N) + B_x u(N) + B_y u_y(N) + +B_z u_z(N)$
Substituting the first eq. in the second, the second in the third one and so on:
$x(1) = x0$
$x(2) = Ax0 + B_x u_x(1) + B_y u_y(1) + B_z u_z(1)$
$x(3) = A(Ax0 + B_x u_x(1) + B_y u_y(1) + B_z u_z(1)) + B_x u_x(2) + B_y u_y(2) + B_z u_z(2)$
$x(4) = A(A(Ax0 + B_x u_x(1) + B_y u_y(1) + B_x u_x(2) + B_y u_y(2) + B_z u_z(2)) + Bu(3)$
$\dots$

$$x(N+1) = A^N x_0 + \begin{bmatrix} A^{N-1} \cdot B_x & A^{N-2} B_x & ... & AB_x & B_x \end{bmatrix} \underbrace{\begin{bmatrix} u_x(1) \\ u_x(2) \\ u_x(3) \\ ... \\ u_x(N) \end{bmatrix}}_{\mathbf{u_x}} + \dots$$

$$\dots + \begin{bmatrix} A^{N-1} \cdot B_y & A^{N-2} B_y & ... & AB_y & B_y \end{bmatrix} \underbrace{\begin{bmatrix} u_y(1) \\ u_y(2) \\ u_y(3) \\ ... \\ u_y(N) \end{bmatrix}}_{\mathbf{u_y}} + \dots$$

$$\dots + \begin{bmatrix} A^{N-1} \cdot B_z & A^{N-2} B_z & ... & AB_z & B_z \end{bmatrix} \underbrace{\begin{bmatrix} u_z(1) \\ u_z(2) \\ u_z(3) \\ ... \\ u_z(N) \end{bmatrix}}_{\mathbf{u_z}}$$

That in compact form could be rewritten as:

$$\underbrace{\begin{bmatrix} x(2) \\ x(3) \\ x(4) \\ ... \\ x(N) \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ A^3 \\ ... \\ A^N \end{bmatrix}}_{F} \mathbf{x_0} + \underbrace{\begin{bmatrix} B_x & 0 & 0 & 0 & 0 \\ AB_x & B_x & 0 & 0 & 0 \\ A^2 B_x & AB_x & B_x & 0 & 0 \\ ... & A^2 B_x & AB_x & B_x & 0 \\ A^{N-1} \cdot B_x & ... & A^2 B_x & AB_x & B_x \end{bmatrix}}_{H_1} \underbrace{\begin{bmatrix} u_x(1) \\ u_x(2) \\ u_x(3) \\ ... \\ u_x(N) \end{bmatrix}}_{\mathbf{u_x}} +$$

$$\ldots + \underbrace{\begin{bmatrix} B_y & 0 & 0 & 0 & 0 \\ AB_y & B_x & 0 & 0 & 0 \\ A^2B_y & AB_y & B_y & 0 & 0 \\ ... & A^2B_y & AB_y & B_y & 0 \\ A^{N-1}\cdot B_y & ... & A^2B_y & AB_y & B_y \end{bmatrix}}_{H_2} \underbrace{\begin{bmatrix} u_y(1) \\ u_y(2) \\ u_y(3) \\ ... \\ u_y(N) \end{bmatrix}}_{\mathbf{u_y}} + \ldots$$

$$\ldots + \underbrace{\begin{bmatrix} B_z & 0 & 0 & 0 & 0 \\ AB_z & B_z & 0 & 0 & 0 \\ A^2B_z & AB_z & B_z & 0 & 0 \\ ... & A^2B_z & AB_z & B_z & 0 \\ A^{N-1}\cdot B_z & ... & A^2B_z & AB_z & B_z \end{bmatrix}}_{H_3} \underbrace{\begin{bmatrix} u_z(1) \\ u_z(2) \\ u_z(3) \\ ... \\ u_z(N) \end{bmatrix}}_{\mathbf{u_z}}$$

Where $\overrightarrow{\mathbf{x}} \in \mathbb{R}^{m\cdot N\times 1}$ with m equal to the number of state corrisponding to $m = 6$, $F \in \mathbb{R}^{mN\times 1}$, $H_1 \in \mathbb{R}^{mN\times N}$,$H_2 \in \mathbb{R}^{mN\times N}$,$H_3 \in \mathbb{R}^{mN\times N}$, $u_x \in \mathbb{R}^{N\times 1}$, $u_y \in \mathbb{R}^{N\times 1}$, $u_z \in \mathbb{R}^{N\times 1}$. That can be rewritten in compact form as:

$$\mathbf{x} = F\mathbf{x_0} + \begin{bmatrix} H_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_3 \end{bmatrix} \begin{bmatrix} \mathbf{u_x} \\ \mathbf{u_y} \\ \mathbf{u_z} \end{bmatrix}$$

with $\mathbf{0} \in \mathbb{R}^{mN\times N}$. Instead for what concern state and control constraints, the problem can be posed as:

$$\mathbf{x}_{min} \le \mathbf{x}(k) \le \mathbf{x}_{\ max} \ \ \forall k \in [1; N]$$

$$\begin{bmatrix} u_{x\ min} \\ u_{y\ min} \\ u_{z\ min} \end{bmatrix} \le \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_z(k) \end{bmatrix} \le \begin{bmatrix} u_{x\ max} \\ u_{y\ max} \\ u_{z\ max} \end{bmatrix} \forall k \in [1; N]$$

$$\begin{bmatrix} -\triangle u_{x\ max} \\ -\triangle u_{y\ max} \\ -\triangle u_{z\ max} \end{bmatrix} \le \begin{bmatrix} \triangle u_x(k) \\ \triangle u_y(k) \\ \triangle u_z(k) \end{bmatrix} \le \begin{bmatrix} \triangle u_{x\ max} \\ \triangle u_{y\ max} \\ \triangle u_{z\ max} \end{bmatrix} \ \forall k \in [1; N]$$

While the initial and the final boundary condition are specified as:

$$\mathbf{x_0} = \begin{bmatrix} x_0 \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{x}_{target} = \begin{bmatrix} x_t \\ v_{x\,t} \\ y_t \\ v_{y\,t} \\ z_t \\ v_{z\,t} \end{bmatrix} = \begin{bmatrix} p_x \\ 0 \\ p_y \\ 0 \\ p_z \\ 0 \end{bmatrix}$$

And considering safety flight, is required that the trajectory does not go below the surface during the maneuver. This could be replaced with a constranits on z position corresponding to :

$$\mathbf{x}_5(k) \geq 0 \ \ \forall k \in [1; N] \tag{6.24}$$

For all the time of flight $\forall k \in [1, N]$.
Finally, we can summarized the overall trajectory optimization problem with the following statement:

$$min \ \ ||(\vec{a_x} - \vec{a_z})||^2 + ||(\vec{a_y} - \vec{a_x})||^2 \tag{6.25}$$

subject to

$$\mathbf{x} = F\mathbf{x_0} + \begin{bmatrix} H_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_3 \end{bmatrix} \begin{bmatrix} \mathbf{u_x} \\ \mathbf{u_y} \\ \mathbf{u_z} \end{bmatrix}$$

$$\mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{\,max} \ \ \forall k \in [1; N]$$

$$\begin{bmatrix} u_{x\ min} \\ u_{y\ min} \\ u_{z\ min} \end{bmatrix} \leq \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_z(k) \end{bmatrix} \leq \begin{bmatrix} u_{x\ max} \\ u_{y\ max} \\ u_{z\ max} \end{bmatrix} \ \forall k \in [1; N]$$

$$\begin{bmatrix} -\triangle u_{x\ max} \\ -\triangle u_{y\ max} \\ -\triangle u_{z\ max} \end{bmatrix} \leq \begin{bmatrix} \triangle u_x(k) \\ \triangle u_y(k) \\ \triangle u_z(k) \end{bmatrix} \leq \begin{bmatrix} \triangle u_{x\ max} \\ \triangle u_{y\ max} \\ \triangle u_{z\ max} \end{bmatrix} \ \forall k \in [1; N]$$

$$\mathbf{x}_5(k) \geq 0 \;\; \forall k \in [1; N]$$

$$\mathbf{x_0} = \begin{bmatrix} x_o \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \quad \mathbf{x}_{target} = \begin{bmatrix} x_t \\ v_{x\,t} \\ y_t \\ v_{y\,t} \\ z_t \\ v_{z\,t} \end{bmatrix} = \begin{bmatrix} px \\ 0 \\ py \\ 0 \\ pz \\ 0 \end{bmatrix}$$

Where p_x, p_y and p_z are the target position and the velocity and acceleration target are set to 0 value inasmuch the aim is to reach the target in hover condition. Bounds are set as equality and inequality constraints to respect not only a initial and final state but also input saturation and its increment. To set these value a consideration on minimum and maximum value for thrust force and Euler angles is mandatory.

$$T_{min} \leq f_{ubz}(k) \leq T_{max} \; \forall k \in [1; N]$$

$$\phi_{\,min} \leq \phi(k) \leq \phi_{\,max} \; \forall k \in [1; N]$$

$$\theta_{\,min} \leq \theta(k) \leq \theta_{\,max} \; \forall k \in [1; N]$$

that is bounded for all the time flight. Now, given the state and control constraints, the minimun trajectory optimization problem to be solved become much hard to respect feasebility. As known infeasible solutions appear when bounds are not respected in the given time horizon N and in such a way to avoid it, a condition that detect the violation is developed. The condition allows to increase the time horizon giving the possibility to find a feasible trajectory without violation. This can be summarized through the following scheme:

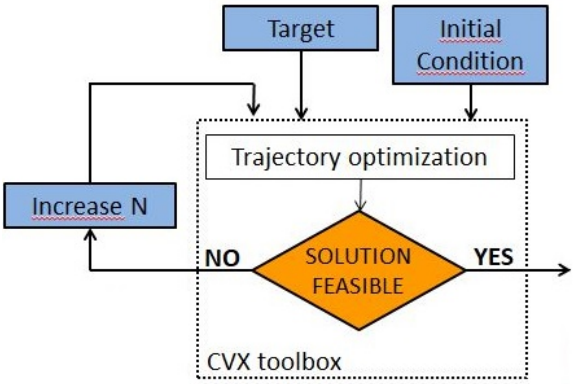

Figure 6.6: Flow chart

An optimization of this type involves an hight number constraints proportional the time horizon N, having negative effects on calculation time. For this reason, in such a way to reduce time calculation, this technique has also been further optimized reducing the problem by scaling the matrices involved in the optimization of a factor k for each step until the desired target is reached and restarting as soon as it is reached. While if the target is not achieved, the solution of the optimization problem would be infeaseable, that implies that optimization is restarted by increasing the prediction horizon. Since a disturbances affect the system and considering what written in the description of this chapther, an implicit feedback control law is required. Its is performed by replanning the trajectory at each time step, and applying only the first input element of optimal sequence vector u^*. This may to result strange or wrong but if there isn't noise that corrupt our output, the trajectory replanned remain optimal as described by Bellman principle.

Bellman principle: given the optimal sequence $u^* = \{u_0^*, u_1^*, \dots, u_N^*\}$ and the corresponding state sequence $x^* = \{x_0^*, {x_1}^*, \dots, x_N^*\}$, the subsequence $\{u_1^*, u_2^*, \dots, u_N^*\}$ remain optmil for the problem on horizon $[\,t_1,\, T\,]$ starting to the optimal state ${x_1}^*$.

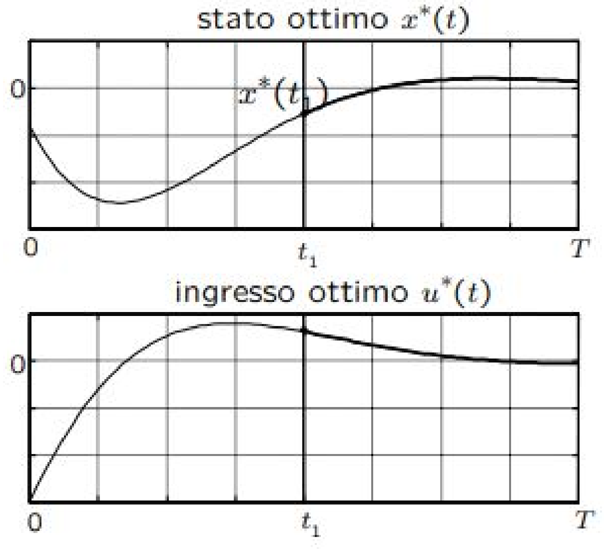

Figure 6.7: Bellman Principle

Moreover, the initial trajectory depends only on the initial state and the sequence of the inputs from 0 to T, so the optimal trajectory from t_1 to T depends $x^*(1)$.
So the requirements that would be respected are expressed as follow:

Algorithm 2 Closed loop

1: $k \leftarrow 0$
2: **for** $k \leftarrow 1, N$ **do**

Require: $\mathbf{x}_0$ *and* $\mathbf{x}_f$ ▷ Initial and target Condition
3: **minimize** J ▷ Constrained Optimization
4: Solve optimization problem
Ensure: optimal sequence u^*
5: Apply $\mathbf{u}_{x,y,z}(1)$
6: back to step 2
7: **end for**

That correspond to apply only the 1^{st} element of the optimal sequence control vector to the system. This state feedback form has the advantage of being more robust in the presence of perturbations on the output giving the possibility to replan the trajectory optimization at each k step.

If no disturbance are presents the 1^{st} trajectory found can be followed with no variation for all k steps. In conclusion is possible to summarize the overall TBVP algorithm developed as follow:

Algorithm 3 TBVP algorithm

$k \leftarrow 0$
for $k \leftarrow 1, N$ **do**

Require: $\mathbf{x}_0$ *and* $\mathbf{x}_f$ ▷ Initial and target Condition
    Calculate Matrices F, H_1, H_2, H_3 ▷ Scaled by k
    minimize J ▷ Constrained Optimization
Ensure: subject to $\forall k \in [1; N]$

$$\mathbf{x} = F\mathbf{x_0} + \begin{bmatrix} H_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_3 \end{bmatrix} \begin{bmatrix} \mathbf{u_{x,y,z}} \end{bmatrix}$$

▷ Dynamic constraint
    $u_{min} \leq \text{u(k)} \leq \text{u}_{max}$ ▷ input constraint
    $-\triangle u_{max} \leq \triangle u(k) \leq \triangle u_{max}$ ▷ input rate constraint
    $\mathbf{x}_{min} \leq \text{x(k)} \leq \text{x}_{max}$ ▷ Output constraint
    $\mathbf{x}_7(k) \geq 0$ ▷ z axis constraint
    Solve optimization problem
    if solution infeasible **then** increase N
    else
        Apply $\mathbf{u}_x(1)$, $\mathbf{u}_y(1)$, $\mathbf{u}_z(1)$ ▷ Apply 1^{st} element Control sequences
        Calculate F_{tot}, $\tau(1)$ ▷ Torques command
        $F_{tot}(1)$, $\tau_{x,y,z}(1) \rightarrow F_{1,2,3,4}(1)$ ▷ Dispatching
        Apply $F_{1,2,3,4}(1)$ to Quadrocopter
    end if
end for

6.2.2 Tracking approach using modificated Model Predictive Control

The approach just described allows the achievement of the aim with excellent results, the only problem that arises is the presence of boundary conditions on the output that cause infeasibility solution. The solution at this problem can be reached deciding to add a term inside the performance index, together to the terms that derive from differential flatness approach, that take into account the difference between the state vector and the reference vector. With this type of strategy, in such a way to reach the minimum value of the cost function the state will have to converge towards the reference. The strategy that best suits this type of solution is known under the name **tracking**. In addition, with this approach, is possible to avoid forcing the state vector to a precise fixed final time that correspond to a problem of optimization less constrained. The performance index to perform an optimization of this type can be chosen as follows:

$$min \;\; \rho(||(\vec{\mathbf{a}}_x - \vec{\mathbf{a}}_z)||^2 + ||(\vec{\mathbf{a}}_y - \vec{\mathbf{a}}_z||^2) + \gamma(||(\vec{\mathbf{x}} - \vec{\mathbf{x}}_{ref})||^2 + ||(\vec{\mathbf{a}} - \vec{\mathbf{a}}_{ref})||^2) \tag{6.26}$$

With the inclusion of the parameters ρ and γ is possible to weight terms involved in the perfomance index giving much importance to one term respect to the other. An example on how the system react changing these parameters is given in the following chapter. The problem formulation is subjet to a inequality and equality constraint as previously defined for TBVP: *subject to*

$$\mathbf{x} = F\mathbf{x}_0 + H_1\vec{\mathbf{a}}_x + H_2\vec{\mathbf{a}}_y + H_3\vec{\mathbf{a}}_z$$

$$\mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{\,max} \;\forall k \in [1; N]$$

$$\begin{bmatrix} u_{x\;min} \\ u_{y\;min} \\ u_{z\;min} \end{bmatrix} \leq \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_z(k) \end{bmatrix} \leq \begin{bmatrix} u_{x\;max} \\ u_{y\;max} \\ u_{z\;max} \end{bmatrix} \;\forall k \in [1; N]$$

$$\begin{bmatrix} -\triangle u_{x\;max} \\ -\triangle u_{y\;max} \\ -\triangle u_{z\;max} \end{bmatrix} \leq \begin{bmatrix} \triangle u_x(k) \\ \triangle u_y(k) \\ \triangle u_z(k) \end{bmatrix} \leq \begin{bmatrix} \triangle u_{x\;max} \\ \triangle u_{y\;max} \\ \triangle u_{z\;max} \end{bmatrix} \;\forall k \in [1; N]$$

$$\mathbf{x}_5(k) \geq 0 \;\forall k \in [1; N]$$

$$\mathbf{x_0} = \begin{bmatrix} x_o \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{x}_{ref} = \begin{bmatrix} x_{ref} \\ v_{x\,ref} \\ y_{ref} \\ v_{y\,ref} \\ z_{ref} \\ v_{z\,ref} \end{bmatrix} = \begin{bmatrix} \overrightarrow{\mathbf{p}}_x \\ \mathbf{0} \\ \overrightarrow{\mathbf{p}}_y \\ \mathbf{0} \\ \overrightarrow{\mathbf{p}}_z \\ \mathbf{0} \end{bmatrix}$$

Where $\overrightarrow{\mathbf{x}} \in \mathbb{R}^{m \cdot N \times 1}$ with m equal to the number of state corrisponding to $m = 6$, $F \in \mathbb{R}^{mN \times 1}$, $H_1 \in \mathbb{R}^{mN \times N}$,$H_2 \in \mathbb{R}^{mN \times N}$,$H_3 \in \mathbb{R}^{mN \times N}$, $u_x \in \mathbb{R}^{N \times 1}$, $u_y \in \mathbb{R}^{N \times 1}$, $u_z \in \mathbb{R}^{N \times 1}$ and $\overrightarrow{\mathbf{p}}_x$, $\overrightarrow{\mathbf{p}}_y$ and $\overrightarrow{\mathbf{p}}_z$ are the ref position and the velocity and acceleration target are set to 0 value for each k steps inasmuch the aim is to reach the target in hover condition. Bounds are set as equality and inequality constraints to respect not only a initial and final state but also input saturation and its increment. To set these value a consideration on minimum and maximum value for thrust force and Euler angles is mandatory.

$$T_{min} \leq f_{ubz}(k) \leq T_{max} \ \forall k \in [1; N]$$

$$\phi_{\ min} \leq \phi(k) \leq \phi_{\ max} \ \forall k \in [1; N]$$

$$\theta_{\ min} \leq \theta(k) \leq \theta_{\ max} \ \forall k \in [1; N]$$

that are bounded for all the time flight. Thanks to the properties of this strategy, the optimization involves an lower number of constraints then TBVP inasmuch the algorithm will work with restricted control horizon P. This consideration have positive effects on calculation time. Moreover, the problem formulated as now do not reach infeasible solution inasmuch the minimization problem take the sistem as much as possibly near the reference target. Since a disturbances affect the system and considering what written in the description of this chapther, an implicit feedback control law is required. Its is performed by replanning the trajectory at each time step, and applying only the first input element of optimal sequence vector u^*. The overall algorithm can be summarized using a pseudo code that take into accout the main steps:

Algorithm 4 Tracking algorithm

$k \leftarrow 0$

for $k \leftarrow 1, N$ **do**

Require: $\mathbf{x}_0$ *and* $\mathbf{x}_f$ ▷ Initial and target Condition

Calculate Matrices F, H_1, H_2, H_3 ▷ Scaled by k

minimize J ▷ Constrained Optimization

Ensure: subject to $\forall k \in [1; N]$

$$\mathbf{x} = F\mathbf{x_0} + \begin{bmatrix} H_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_3 \end{bmatrix} \begin{bmatrix} \mathbf{u_{x,y,z}} \end{bmatrix}$$

▷ Dynamic constraint

$u_{min} \leq \text{u(k)} \leq \text{u}_{max}$ ▷ input constraint

-$\triangle u_{max} \leq \triangle u(k) \leq \triangle u_{max}$ ▷ input rate constraint

$\mathbf{x}_{min} \leq \text{x(k)} \leq \text{x}_{max}$ ▷ Output constraint

$\mathbf{x}_7(k) \geq 0$ ▷ z axis constraint

Solve optimization problem

if solution infeasible **then** increase N

else

Apply $\mathbf{u}_x(1)$, $\mathbf{u}_y(1)$, $\mathbf{u}_z(1)$ ▷ Apply 1^{st} element Control sequences

Calculate F_{tot}, $\tau(1)$ ▷ Torques command

$F_{tot}(1)$, $\tau_{x,y,z}(1) \rightarrow F_{1,2,3,4}(1)$ ▷ Dispatching

Apply $F_{1,2,3,4}(1)$ to Quadrocopter

end if

end for

Chapter 7

Results

7.1 TBVP constrained using modificated Model Predictive Control

The trajectory optimization problem shown in Ch.5.2.1 can be resolved implementing the problem shown above using CVX toolbox for matlab. Now considering what written before and taking the considerations on the bounds, is possible to find the trajectory for position, velocity and accelleration. To simplify the understanding of the graphs and evaluate their correctness, it was initially chosen to arrive in a position of symmetry for the three axes. So recalling the optimization problem described previously :

$$min \ \ ||(\vec{a_x} - \vec{a_z})||^2 + ||(\vec{a_y} - \vec{a_x})||^2 \tag{7.1}$$

subject to

$$\mathbf{x} = F\mathbf{x_0} + \begin{bmatrix} H_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_3 \end{bmatrix} \begin{bmatrix} \mathbf{u_x} \\ \mathbf{u_y} \\ \mathbf{u_z} \end{bmatrix}$$

$$\begin{bmatrix} u_{x\ min} \\ u_{y\ min} \\ u_{z\ min} \end{bmatrix} \leq \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_z(k) \end{bmatrix} \leq \begin{bmatrix} u_{x\ max} \\ u_{y\ max} \\ u_{z\ max} \end{bmatrix} \quad \forall k \in [1; N]$$

$$\begin{bmatrix} -\triangle u_{x\ max} \\ -\triangle u_{y\ max} \\ -\triangle u_{z\ max} \end{bmatrix} \leq \begin{bmatrix} \triangle u_x(k) \\ \triangle u_y(k) \\ \triangle u_z(k) \end{bmatrix} \leq \begin{bmatrix} \triangle u_{x\ max} \\ \triangle u_{y\ max} \\ \triangle u_{z\ max} \end{bmatrix} \ \forall k \in [1; N]$$

$$\mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{\ max} \ \forall k \in [1; N]$$

$$\mathbf{x}_5(k) \geq 0 \ \forall k \in [1; N]$$

$$\mathbf{x_0} = \begin{bmatrix} x_0 \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad \mathbf{x}_{target} = \begin{bmatrix} x_t \\ v_{x\,t} \\ y_t \\ v_{y\,t} \\ z_t \\ v_{z\,t} \end{bmatrix} = \begin{bmatrix} p_x \\ 0 \\ p_y \\ 0 \\ p_z \\ 0 \end{bmatrix}$$

defining the target position as $p_x = 10[m]$, $p_y = 10[m]$, $p_z = 10[m]$, the final value of acceleration and velocity equal to 0 and the bound constraints as:

$$T_{min} \leq f_{ubz} \leq T_{max}$$

$$\phi_{\ min} \leq \phi(k) \leq \phi_{\ max}$$

$$\theta_{\ min} \leq \theta(k) \leq \theta_{\ max}$$

with $T_{min} = 0\ [N]$,$T_{max} = 100\ [N]$,$\phi_{\ min} = -\frac{\pi}{2}$,$\phi_{\ max} = \frac{\pi}{2}$, $\theta_{\ min} = -\frac{\pi}{2}$,$\theta_{\ max} = +\frac{\pi}{2}$
That are rewritten recalling the equation () under constraints of the accelleration profile :

$$\vec{\mathbf{a}}_l = \begin{bmatrix} s_\phi c_\theta \\ s_\theta \\ c_\phi c_\theta \end{bmatrix} \cdot \frac{f_{ubz}}{m_q} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

where m_q is the quadrocopter mass equal to 1.49 $[Kg]$ and g the gravitatonal costant equal to 9.81 $[\frac{m}{s^2}]$ that if entered in the previous equation with the constraints on thrust and Euler angles returns:

$$\begin{bmatrix} -33.33 \\ -33.33 \\ -9.81 \end{bmatrix} \leq \begin{bmatrix} a_x(k) \\ a_y(k) \\ a_z(k) \end{bmatrix} \leq \begin{bmatrix} 33.33 \\ 33.33 \\ 33.33 \end{bmatrix}$$

Moreover a bounds on the variation of acceleration in the time unit is introduced to obtain a smooth change of angles:

$$\begin{bmatrix} -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \leq \begin{bmatrix} \triangle a_x(k) \\ \triangle a_y(k) \\ \triangle a_z(k) \end{bmatrix} \leq \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

The following results show how the system dynamic evolves respecting all the bound and reaching the final targets. For the simulation, are applyed two different target point, the first one equal to 0 fot $t = [1 : 3]\ s$ while the second target is set to $p_x = 10[m]$, $p_y = 10[m]$, $p_z = 10[m]$ and 0 value for the acceleration and velocity profile for $t = [12 : 15]\ s$.

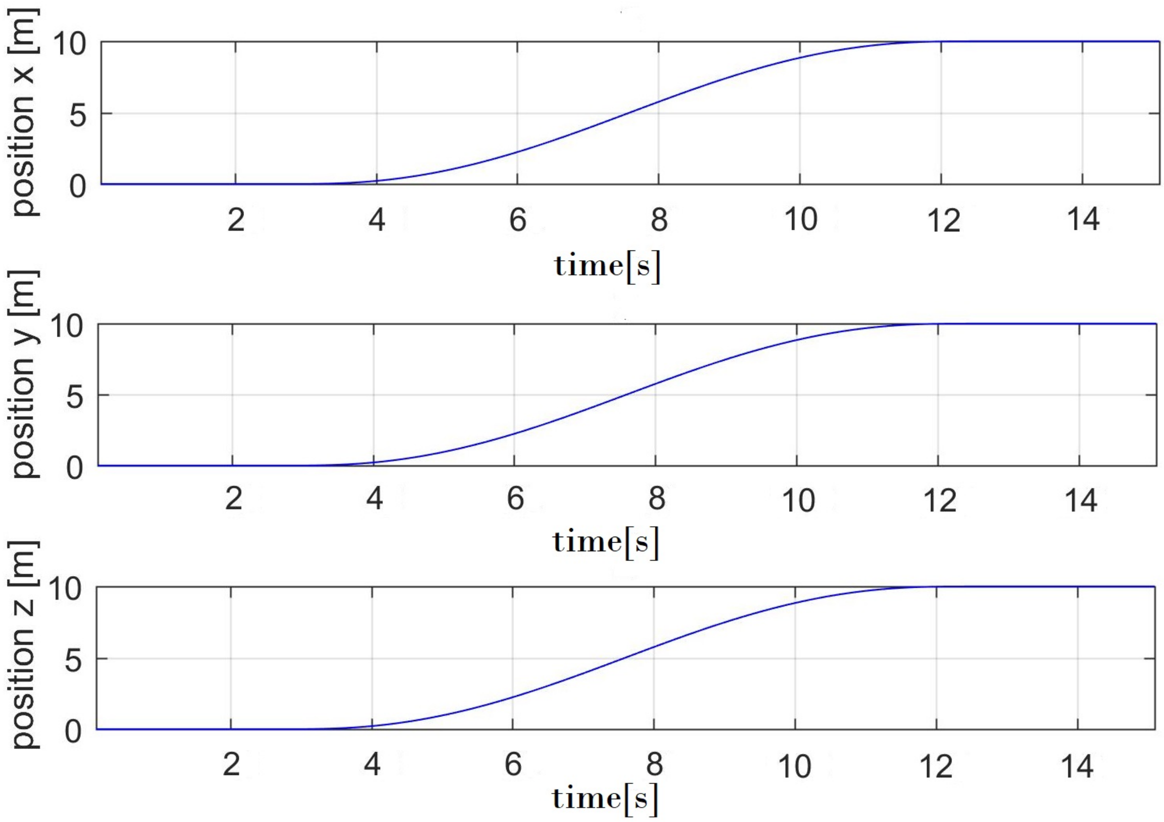

Figure 7.1: Position profile TBVP

As previously anticipated, a target has been chosen in an optimal position such as to make it easier to verify the correct working of the algorithm. From the last plot is possible to see how the positions evolves from the initial position to the final one. Obviously if the position change in time starting to a rest state and reaching a target, also the velocity and acceleration change. Since the boundary condition are set in such a way to have hover condition the velocity have the following behaviour

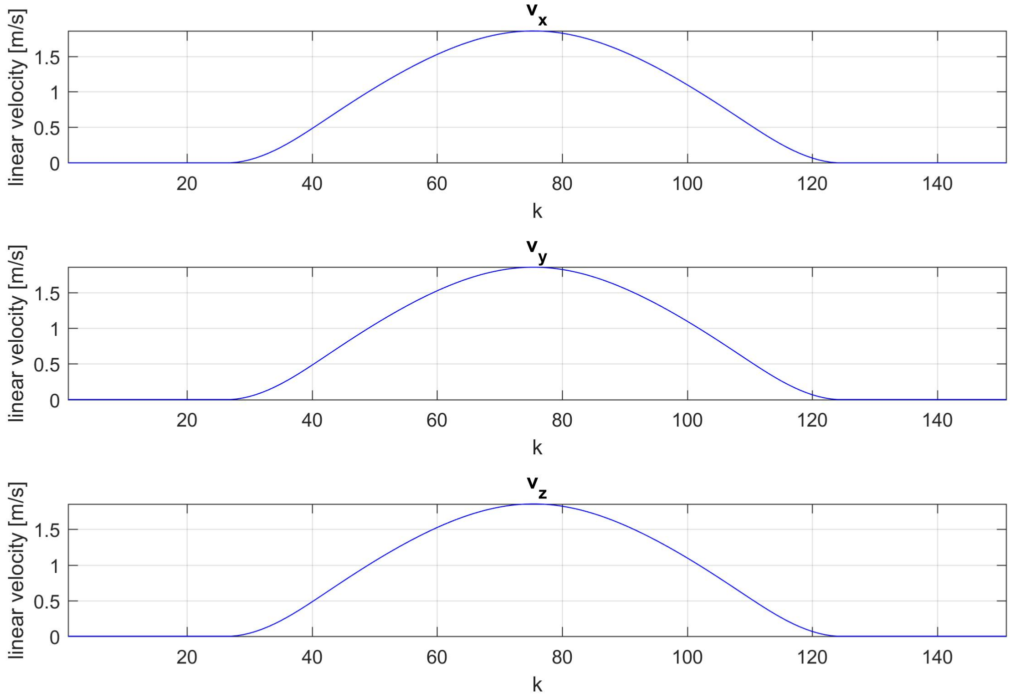

Figure 7.2: Velocity profile

Also the velocity profile show a symmetry inasmuch the system evolves uniformly respect to the 3 axis. This is due to the fact that a cost function has been chosen that minimizes the acceleration differences. In fact also the acceleration will be equal (for this symmetrical target) and are summurize by the picture below:

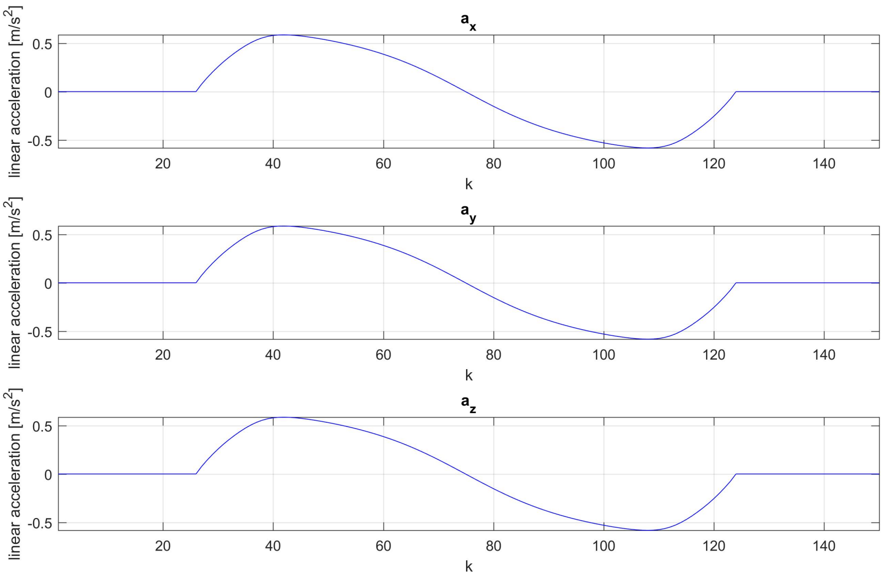

Figure 7.3: Acceleration profile

As can be understood from the responce of acceleration, a hover condition is mantained considering that no accelerations are present on the boundary respecting all the requirements, while in the moving phase the quadrocopter will accelerate for $t = [2,5 - 7,5]$ and decelerate for $t = [7,5 - 12,5]$. Moreover, can be seen how the constraint on the acceleration rate is respected, resulting in a soft variation of accelerations that will be reflected on a corresponding soft variation of angles. Inasmuch the acceleration is linked with the angles and the thrust through the equation (5.12), also these profiles are known :

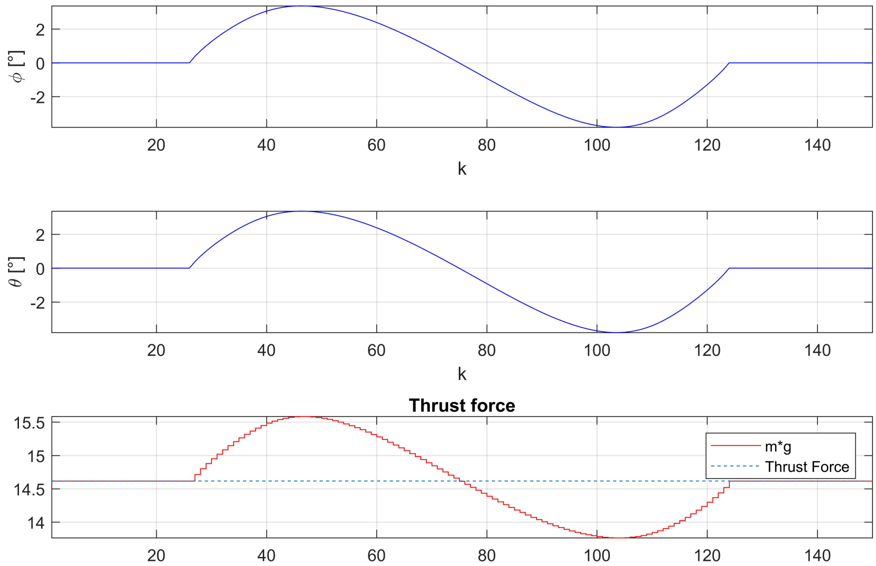

Figure 7.4: Euler angles and total thrust force

the behaviour shown by the Euler angles gives the possibility to state that the goal has been achieved inasmuch as they show a maximum value of 3 degrees and a variation in the unit of time very low, therefore non-sudden angular variations. From this results and considering the equation (5.14), also the torque commands are computed, summarized in the figure below.

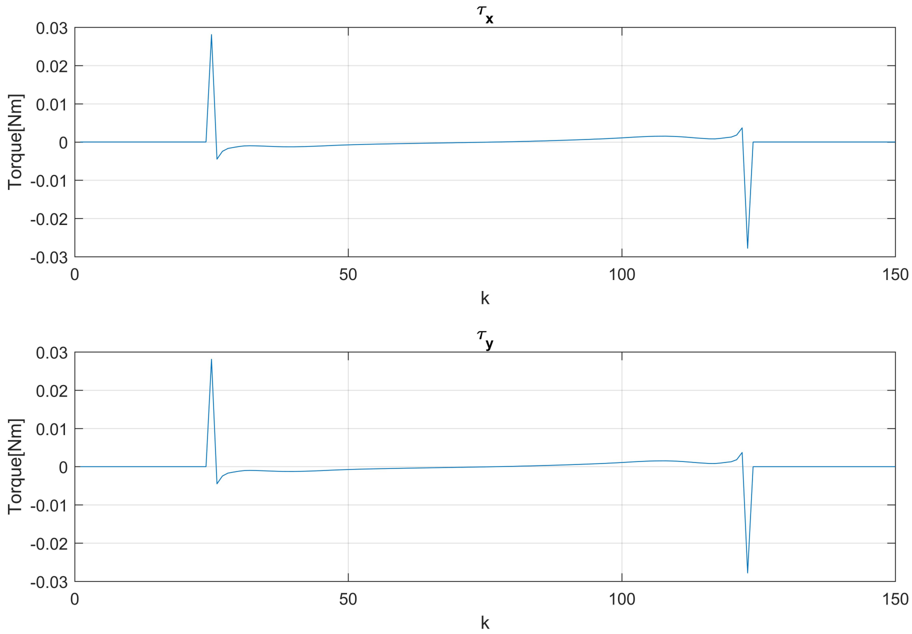

Figure 7.5: Torques commands

Since the goal are achived, all that remains is to understand if the values of torque obtained are right. Considering as angles change, the calculated torques respect what described above, in fact this can be trivially demonstrated considering that a positive torque on the x axis generates a positive variation of the ϕ angle and similarly this happens also for the torque on the y axis, instead for the torque on the axis z the speech is completely different, where no rotation is required by the operator and this could lead to an incorrect evaluation of what happens. In fact, the presence of this torque shows that different angular variations of the propellers-motors lead to a corresponding yaw movement and that a torque on the z axis is necessary to balance and then cancel it. Infact if the commands inputs are dispatched into the corresponding propeller forces through the equation (5.10) the previous results can be clearified.

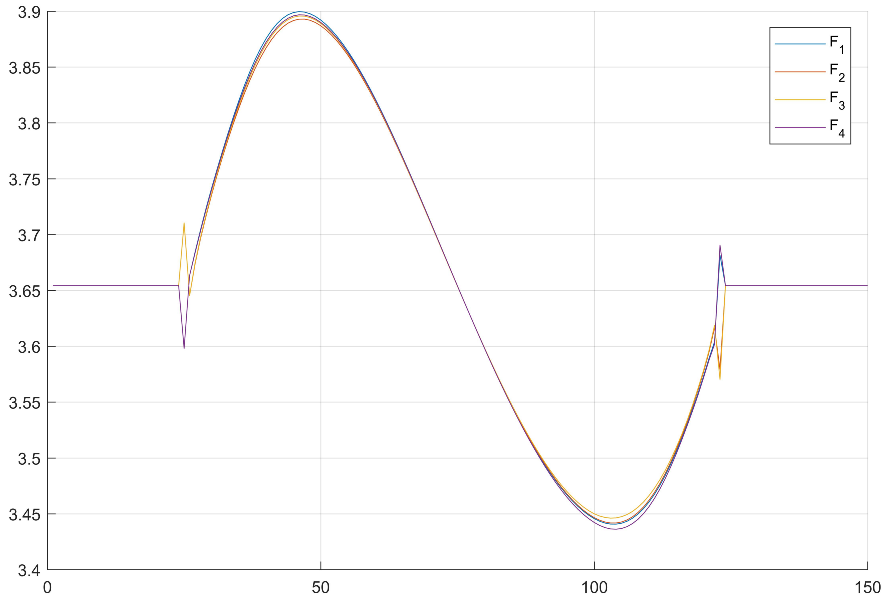

Figure 7.6: Dispatched forces

That can be divided into two couple that generate torques as:

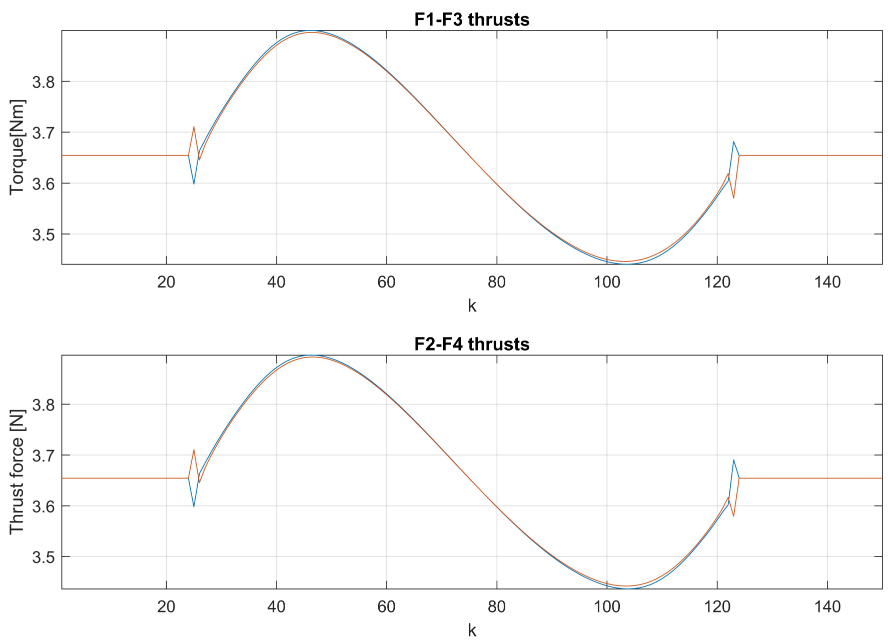

Figure 7.7: Dispatched forces

Where F1-F3 are the forces that drive the torque on the x axis and obviously F2-F4 are the forces that drives the torque on the axis y. For $t = [1 : 2,5]\,s$ there is an hover condition where all the forces are equal and their sum correspond to the weight force maintaining the quadrocopter around its equilibrium point, while at $t = 2,5\,s$ the quadrotor start to move generating torques varying the propeller-motors angular velocity for $0.3\,s$. After this phase the torque is not needed in such a way to maintain the inclination. This condition is obtained with the equality of the 4 forces. While at t=4,8 s the quadrocopter start to change its attitude, inverting its trend to start the deceleration phase. The behaviour of this results shown how the requirements are respected giving the possibility to provide an automated and adaptive optimazed guidance that kept the quadrocopter in any target position in the space with the minimum inclination for all the trajectory. Using the TBVP algorithm developed, that embedd the guidance and control unit, the system can be simulated giving the possibility to understand how it reacts and if the target point is reached. This can be verified in the next picture:

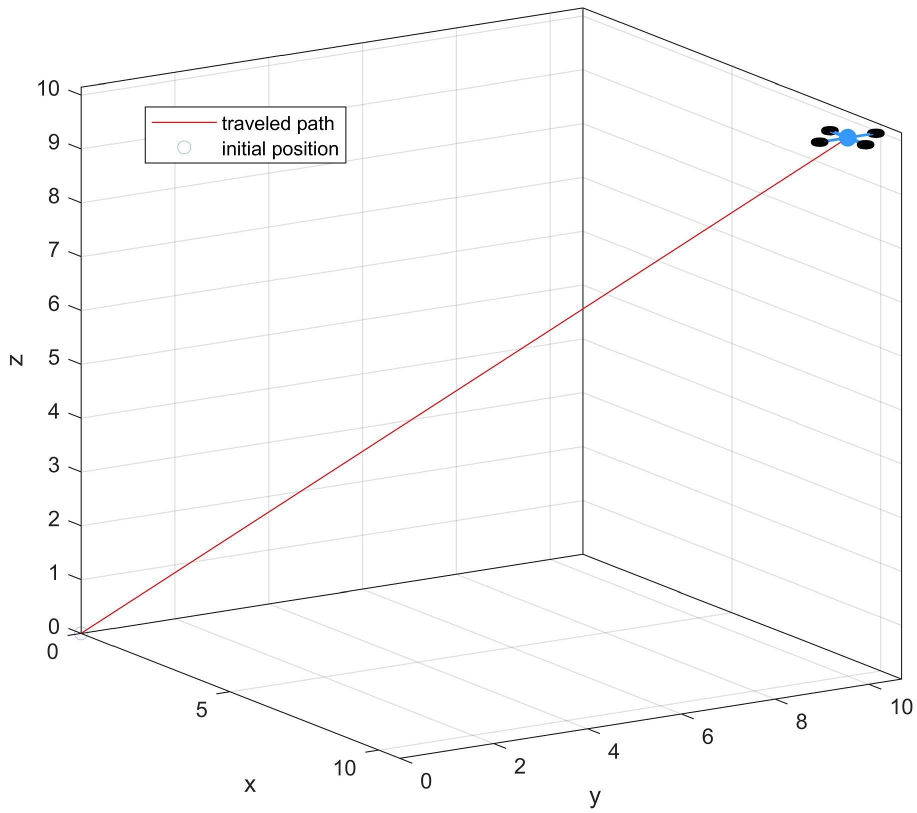

Figure 7.8: Trajectory

that show how the quadrocopter reach the final target with no violation of boundary constraints. Thanks to this simulation, is possible to verify that the attitude is respected and all the calculations are correct, in fact in the following figures can be seen the maximum inclination for θ .

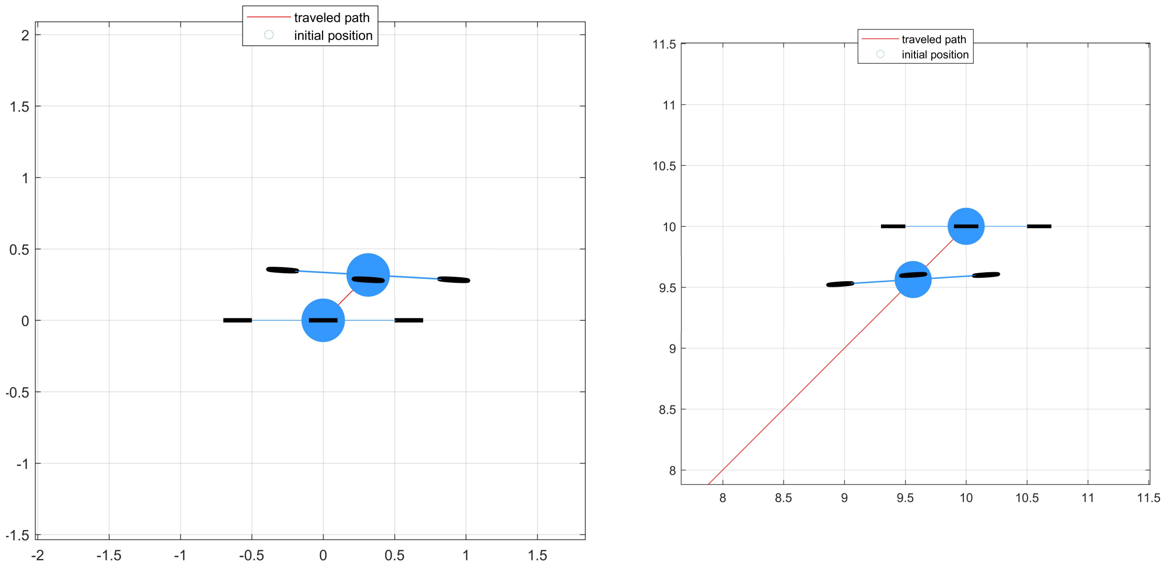

Figure 7.9: Initial phase (left) and Final phase (right)

7.2 Tracking approach using Model Predictive Control

The resolution of the optimization problem using the tracking approach is similar to the previous one. In fact, for the construction of the matrices involved in the optimizatioin problem is possible to use the same set of parameters used in the TBVP optimization problem. The only parameters that will be modulated are ρ and γ, where they represents the weight of each term involved in the performance index.

$$min \;\; \rho((\vec{a_x} - \vec{a_z})^2 + (\vec{a_y} - \vec{a_z})^2) + \gamma((\vec{\mathbf{x}} - \vec{\mathbf{x}_{ref}})^2 + (\vec{\mathbf{a}} - \vec{\mathbf{a}_{ref}})^2) \tag{7.2}$$

subject to

$$\mathbf{x} = F\mathbf{x_0} + H_1\mathbf{u}_x + +H_2\mathbf{u_y} + H_3\mathbf{u_z}$$

$$\begin{bmatrix} -33.33 \\ -33.33 \\ -9.81 \end{bmatrix} \leq \begin{bmatrix} a_x(k) \\ a_y(k) \\ a_z(k) \end{bmatrix} \leq \begin{bmatrix} 33.33 \\ 33.33 \\ 33.33 \end{bmatrix} \forall k \in [1; N]$$

$$\begin{bmatrix} -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \leq \begin{bmatrix} \triangle a_x(k) \\ \triangle a_y(k) \\ \triangle a_z(k) \end{bmatrix} \leq \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \forall k \in [1; N]$$

$$\mathbf{x}_5(k) \geq 0 \ \forall k \in [1; N]$$

$$\mathbf{x_0} = \begin{bmatrix} x_o \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \quad \mathbf{x}_{target} = \begin{bmatrix} x_t \\ v_{x\,t} \\ y_f \\ v_{y\,t} \\ z_t \\ v_{z\,t} \end{bmatrix} = \begin{bmatrix} p_x \\ 0 \\ p_y \\ 0 \\ p_z \\ 0 \end{bmatrix}$$

where $x_{ref} \in \mathbb{R}^{m \cdot N \times 1}$, $\vec{a_x}$, $\vec{a_y}$and$\vec{a_z}$ $\in \mathbb{R}^{1 \times N}$, while ρ and γ are set to 1 in such a way to verify as the algorithm works. The other parameters are defined equal to the problem described before.

The following results show how the system dynamic evolves respecting all the bound and tracking the reference value on N time horizon with $N = 150$ sampling at $T_s = 0.1\,s$. For the simulation, the reference signal have been splitted into two subparts, equal to 0 for $t = [1 : 3)$ s and equal to $p_x = 10[m]$, $p_y = 10[m]$, $p_z = 10[m]$ and 0 value for the velocity and when $t = [3, N]$, while the second reference given to track the command input (the acceleration) is set to 0 for $t = [1 : N]$ s that correspond to have 0 acceleration in the initial time at also when the target is reached.

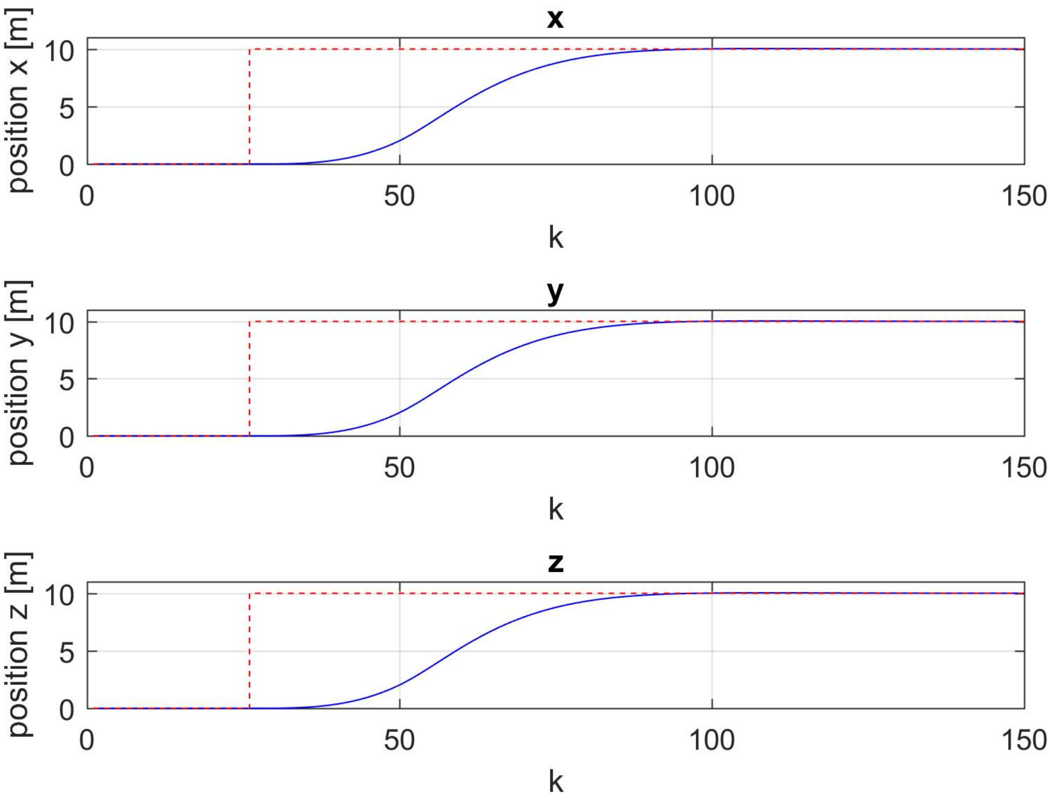

Figure 7.10: Position profiles

As previously done for the TBVP problem, a reference has been chosen in an optimal position such as to make it easier to verify the correct working of the algorithm. As the picture shows, system reach faster the reference. The reason is that the system is much free to evolve inasmuch there is not fixed final time value. Obviously if the position change in time starting to a rest state and reaching a target, also the velocity and acceleration change. Since the boundary condition are set in such a way to have hover condition, the velocity have the following profile.

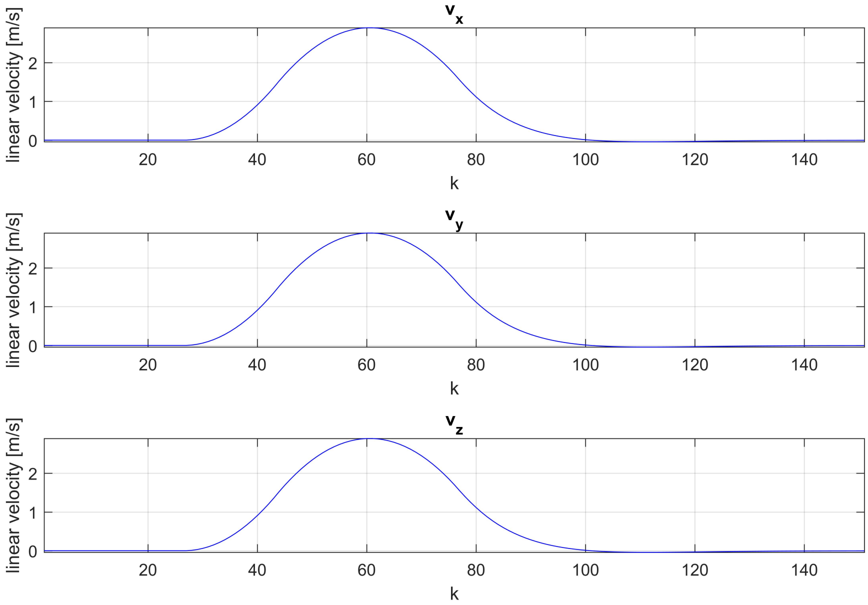

Figure 7.11: velocity profile

The velocity profile shows a symmetry inasmuch the system evolves uniformly respect to the 3 axis. This is due to the fact that a cost function has been chosen that minimizes the acceleration differences and also to track an equal references for the three axes. In fact, the accelerations will be equal for the 3 axes (for this symmetrical references) and are summurized by the picture below:

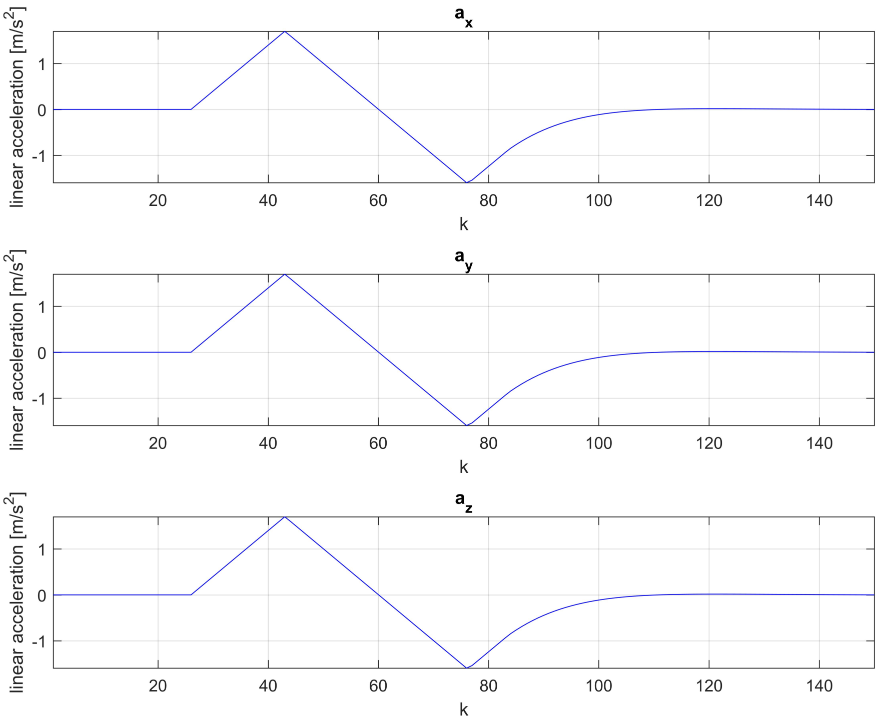

Figure 7.12: Acceleration profiles

As can be understood from the responce of acceleration, a hover condition is maintained considering that no accelerations are present on the boundary respecting all the requirements, while in the moving phase the quadrocopter will accelerate for $t = [2,5-5]$ and decelerate for $t = [5-10]$. Moreover, can be seen how the constraint on the acceleration rate is respected, but differently to the targetting results. There is soft variation of accelerations that will be reflected on a corresponding soft variation of angles by depending to the fact that the references are reached in short time the angle swap is much wide. Infact using the equation (), also the angle profiles are known and are shown below:

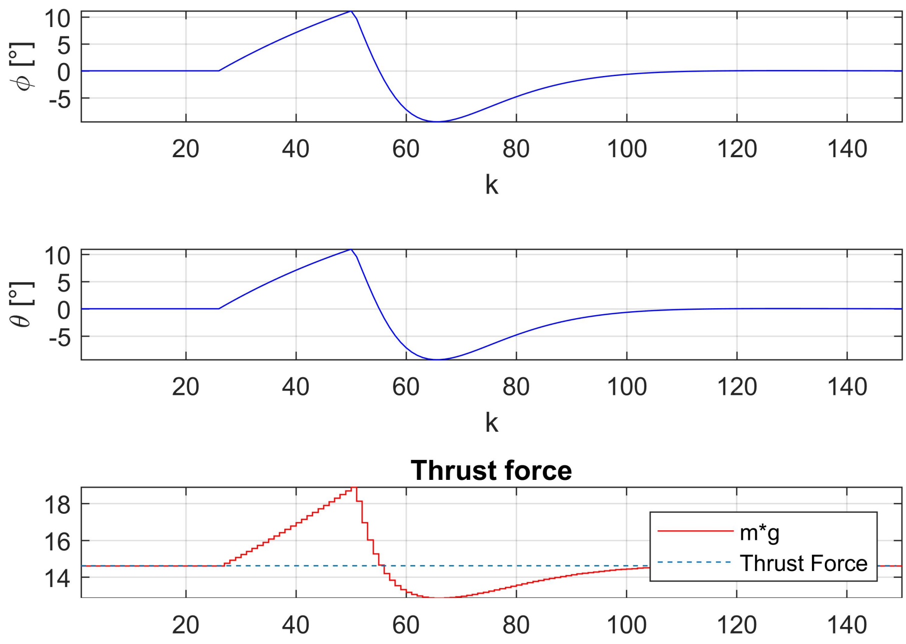

Figure 7.13: Angles profiles

To reach the target in short time, are provided not also a big value of angle but also a major value of thrust. Since this results respect all the bounds, to have a bondend value of angles near 3-5 degrees the optimization problem is reformulated changing the maxim value of velocity and acceleration:

$$\begin{bmatrix} -1.91 \\ -1.91 \\ -1.91 \end{bmatrix} \leq \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix} \leq \begin{bmatrix} 1.91 \\ 1.91 \\ 1.91 \end{bmatrix}$$

$$\begin{bmatrix} -0.6 \\ -0.6 \\ -0.6 \end{bmatrix} \leq \begin{bmatrix} a_x(k) \\ a_y(k) \\ a_z(k) \end{bmatrix} \leq \begin{bmatrix} 0.6 \\ 0.6 \\ 0.6 \end{bmatrix}$$

$$\begin{bmatrix} -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \leq \begin{bmatrix} \triangle a_x(k) \\ \triangle a_y(k) \\ \triangle a_z(k) \end{bmatrix} \leq \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

applying this change of bound is possible to obtain the same trajectory with minor angles, obviously this changing will be reflect on the saturations of this. This is shown in the next figure:

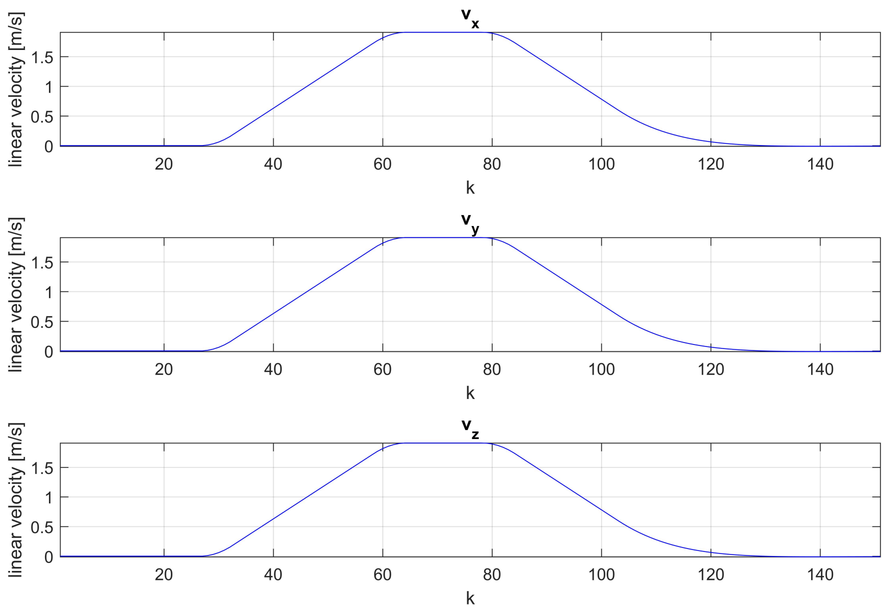

Figure 7.14: Velocity profile

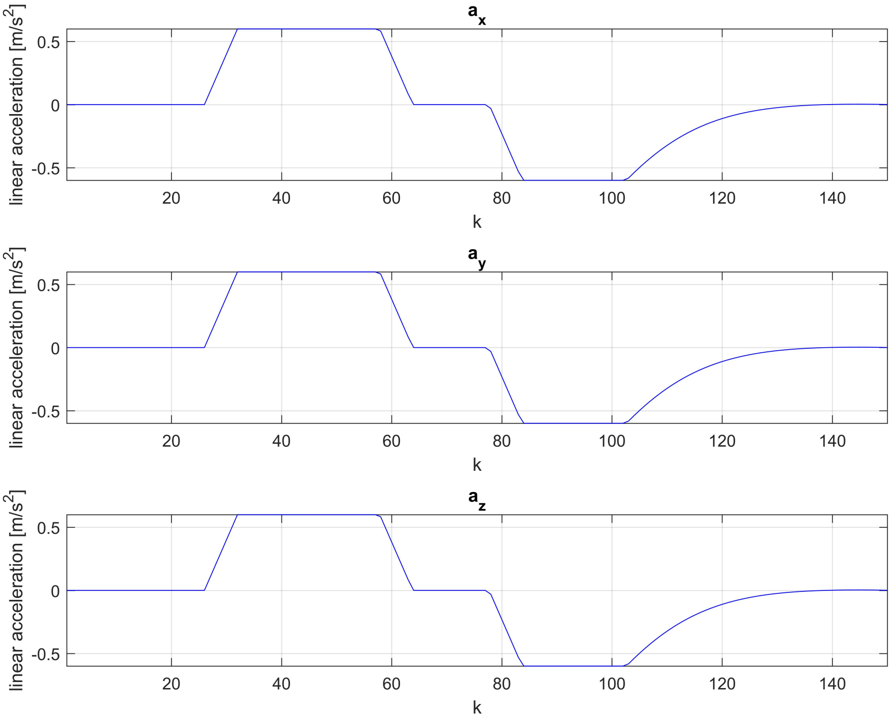

Figure 7.15: acceleration profile

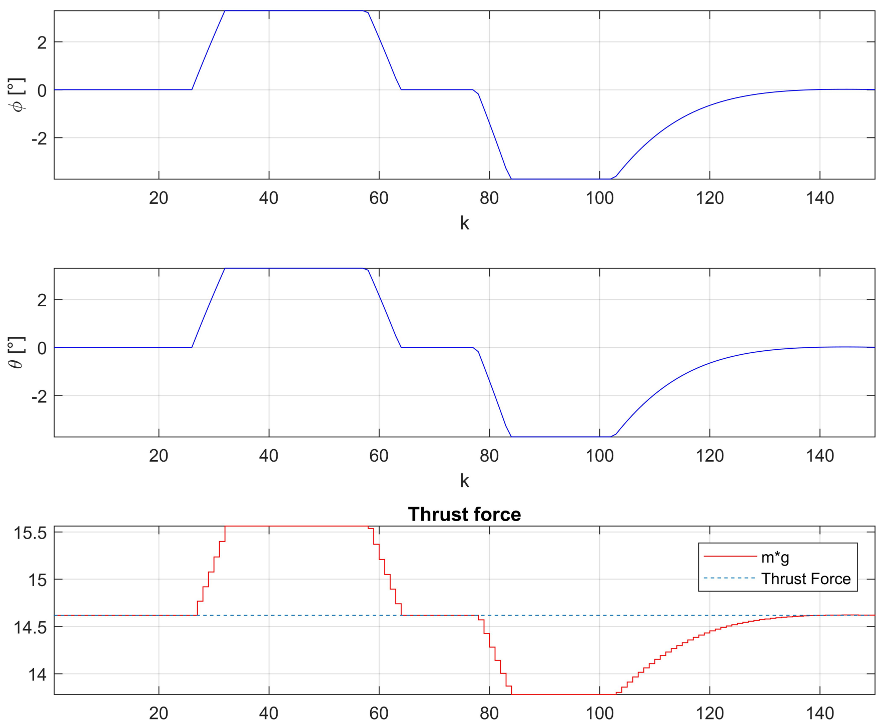

Figure 7.16: angles profiles

This correspond to have an abrupt change of angles and its saturation. The reason are inside the formulation of the objective function where difference between the state and its reference dominates the difference between the couples of accelerations. This can be avoided by the introduction of costants that weights this quantities. This method is used also in standard optimal control as Linear Quadratic Regulation (LQR) or standard Model Predictive Controlo (MPC). So to reach a better results the formulation problem are rewritten in the following form:

$$min \;\; \rho((\vec{a_x} - \vec{a_z})^2 + (\vec{a_y} - \vec{a_z})^2) + \gamma((\vec{\mathbf{x}} - \vec{\mathbf{x}_{ref}})^2 + (\vec{\mathbf{a}} - \vec{\mathbf{a}_{ref}})^2) \tag{7.3}$$

subject to

$$\mathbf{x} = F\mathbf{x_0} + H_1\mathbf{u}_x + +H_2\mathbf{u_y} + H_3\mathbf{u_z}$$

$$\begin{bmatrix} -33.33 \\ -33.33 \\ -9.81 \end{bmatrix} \leq \begin{bmatrix} a_x(k) \\ a_y(k) \\ a_z(k) \end{bmatrix} \leq \begin{bmatrix} 33.33 \\ 33.33 \\ 33.33 \end{bmatrix} \forall k \in [1; N]$$

$$\begin{bmatrix} -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \leq \begin{bmatrix} \triangle a_x(k) \\ \triangle a_y(k) \\ \triangle a_z(k) \end{bmatrix} \leq \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \forall k \in [1; N]$$

$$\mathbf{x}_5(k) \geq 0 \ \forall k \in [1; N]$$

$$\mathbf{x_0} = \begin{bmatrix} x_o \\ v_{x\,0} \\ y_0 \\ v_{y\,0} \\ z_0 \\ v_{z\,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \quad \mathbf{x}_{target} = \begin{bmatrix} x_t \\ v_{x\,t} \\ y_f \\ v_{y\,t} \\ z_t \\ v_{z\,t} \end{bmatrix} = \begin{bmatrix} p_x \\ 0 \\ p_y \\ 0 \\ p_z \\ 0 \end{bmatrix}$$

where ρ is assigned to an higher value and γ to a lower one. This formulation allows to reach our aim without restriction on bounds. Applying this consideration is possible to find a minimized trajectory giving much importance to the term that take into account the difference of couples of acceleration (first term of objective function):

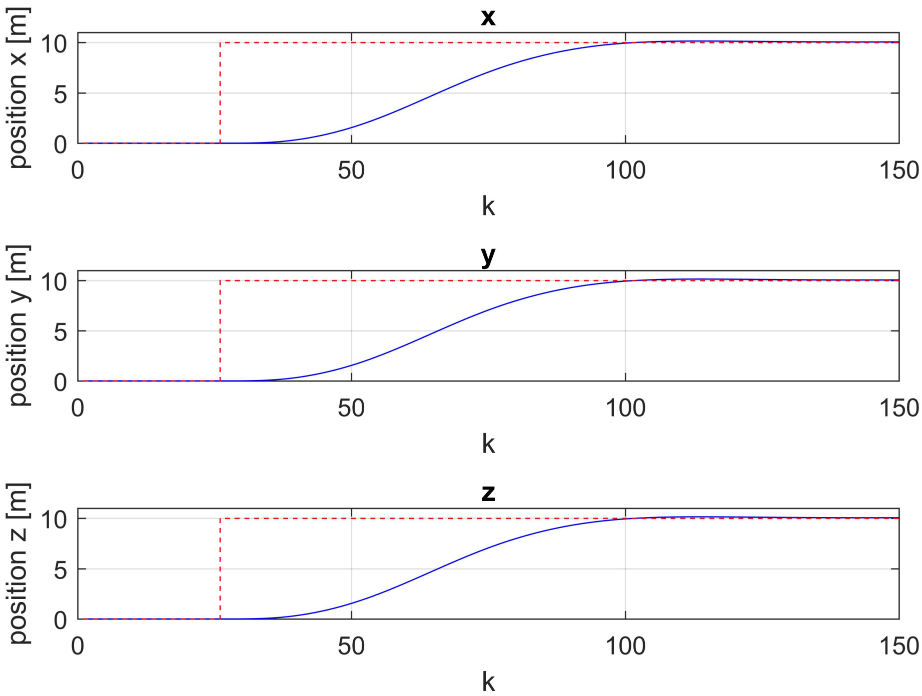

Figure 7.17: position profiles

With the inclusion of the parameters ρ and γ is possible to weight terms involved in the perfomance index. The speed profiles are obviously almost identical as shown from the following figure:

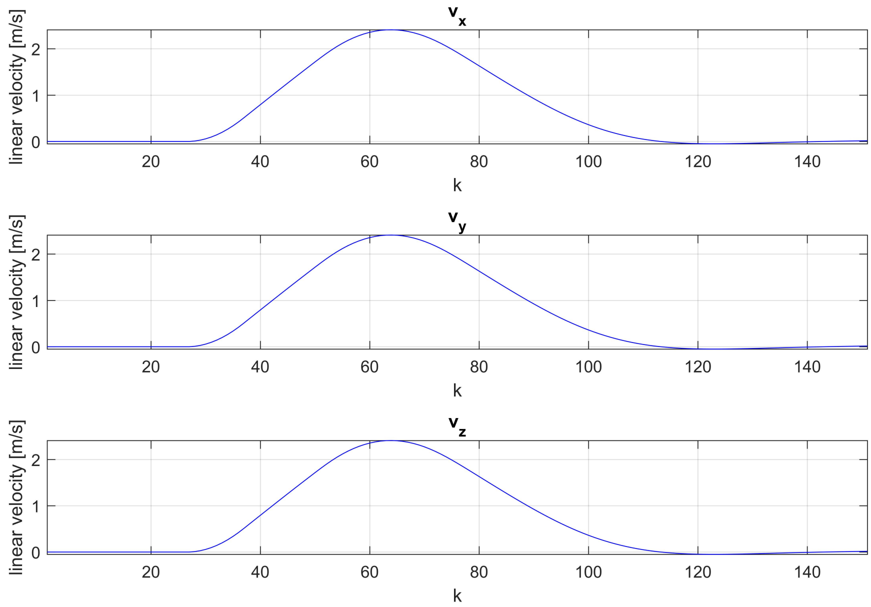

Figure 7.18: velocity profiles

While for the accelerations is shown substantial variation compared to the previous cases given precisely by the inclusion of the weight parameters ρ and γ. This is verified in the figure below:

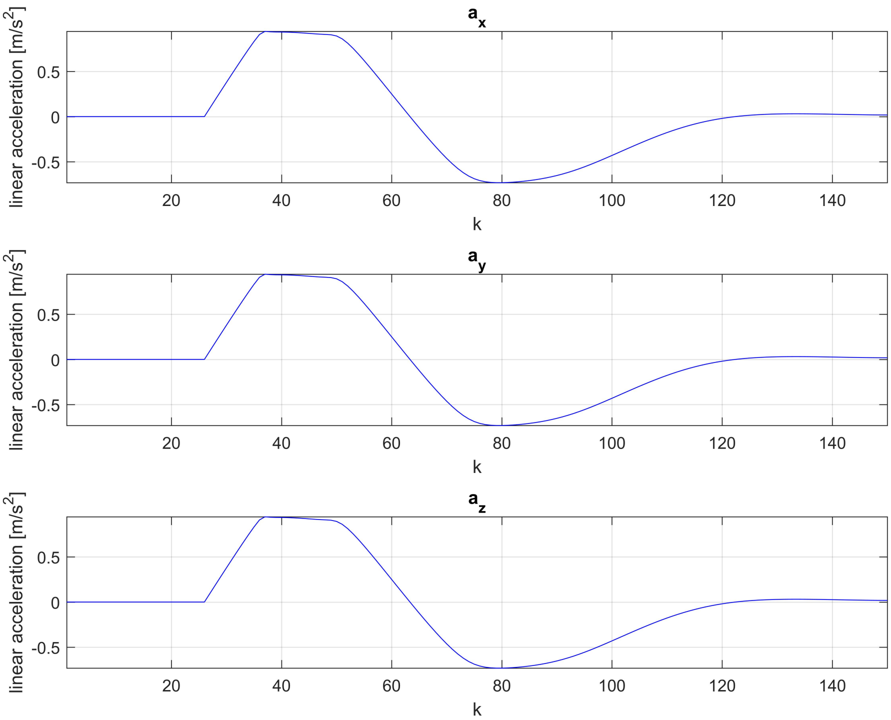

Figure 7.19: acceleration profile

With the insertion of the weight parameters in the tracking technique is possible to led back to the same result obtained using the TBVP constrained technique where the performance index is formed only by the differences of the accelerations couples. This allows to use the results of the TBVP as an ideal case. Thus the angular profiles can be summarized as follows:

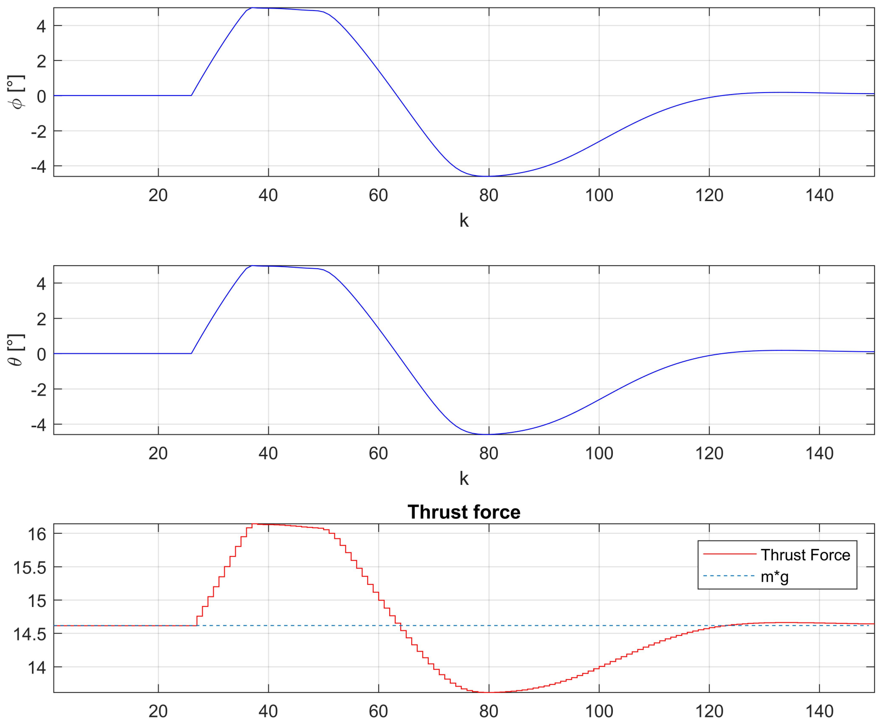

Figure 7.20: angles profile

the behavior shown by the Euler angles gives the possibility to identify a maximum value of + -4.5 degrees and a variation in the unit of time very low, therefore non-sudden angular variations. From the results and considering the equation (), the torque commands are computed, summarized in the figure below.

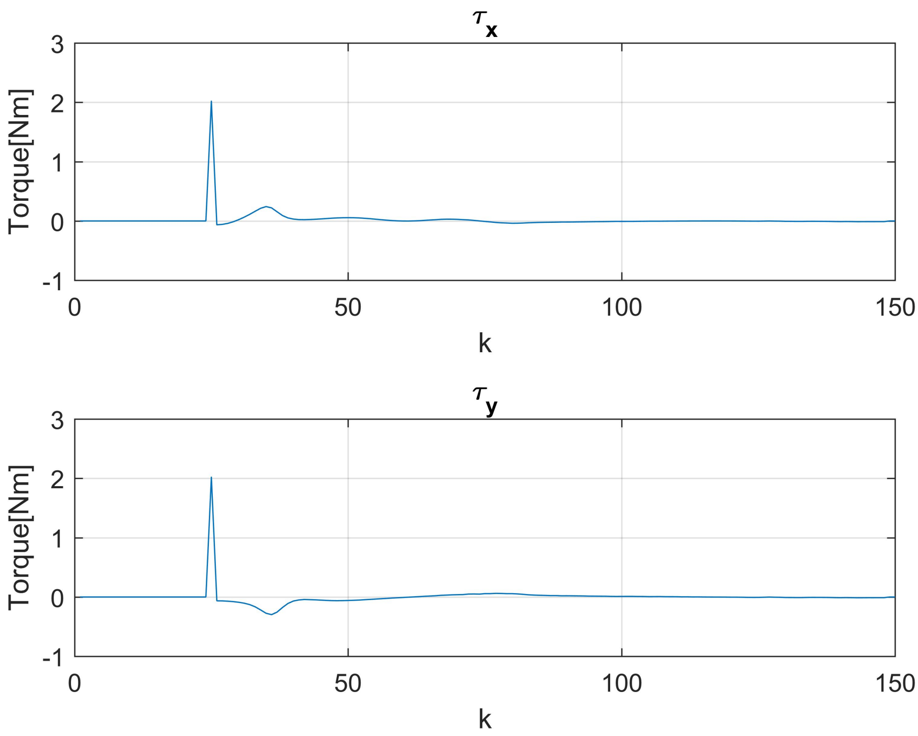

Figure 7.21: Torque profile

All that remains is to understand if values of torque obtained are right. This can be trivially demonstrated considering that a positive torque on the x axis generates a positive variation of the ϕ angle and similarly this happens also for the torque on the y axis, instead for the torque on the axis z the speech is completely different and equal to the TBVP discussion, where no rotation is required by the operator and this could lead to an incorrect evaluation of what happens. In fact, the presence of this torque, shows that different angular variations of the propellers-motors lead to a corresponding yaw movement and that a torque on the z axis is necessary to balance and then cancel it. If the commands inputs are dispatched into the corresponding propeller forces through the equation () the previous results can be clearified.

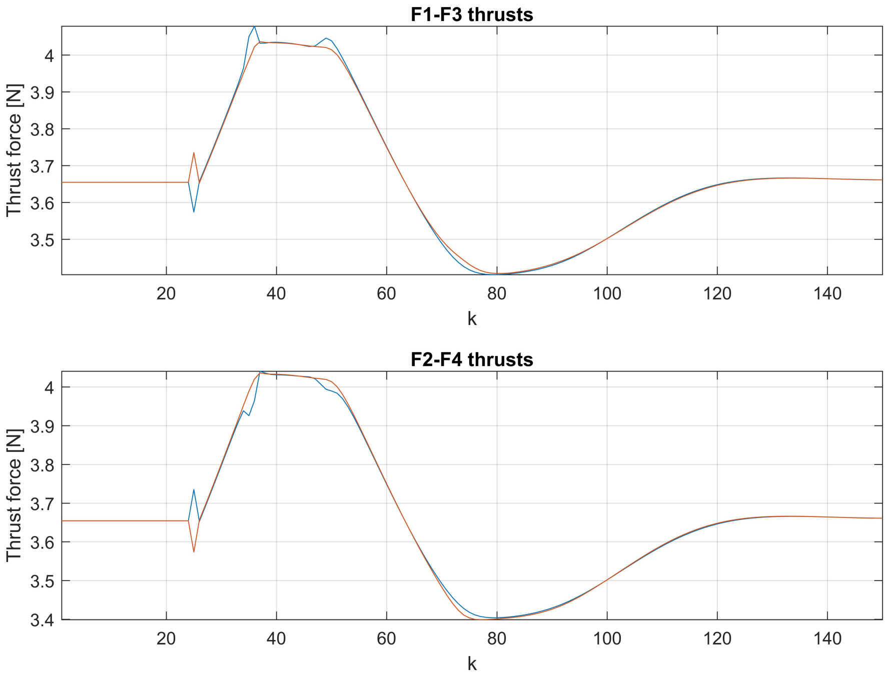

Figure 7.22: Set force profiles

As is possible to understand from these results, by modifing the parameters ρ and γ more importance can be given to the minimization of the difference of couples accelerations respect to the the difference between state vector and its reference. Therefore it is possible to state that the results obtained are consistent with the purpose to be achieved. This allows to use the designed techniques to solve the trajectory optimization problem.

7.3 TBVP vs Tracking

In the previous sections the parameters and the results of two different techniques have been shown. These two will be compared in order to choose the tecnique that best suits our optimization problem. The first described tecnique (TBVP) will be taken as ideal case. This is possible since it has in the objective function only the difference of the couples accelerations that leads to a minimization of the square of the angles through differential flatness approach. Thus allows to study the problem in the linear form subject to initial and final linear constraints. The second designed technique (Tracking) is developed in such a way to set free the system from final boundary constraints. Both techniques are developed using Model Predictive Control (MPC) theory, to solve constrained discrete time optimal problem. Obviously, considering that the system is non-linear, it was necessary to use the differential flatness technique to return to a suitable form of the objective function in line with the assumption of linear models. Moreover, the assumption of linear models derives from a feedback linearization. The design of several techniques for solving the trajectory optimization problem allows to make a direct comparison between them. The comparison gives the possibility to understand which technique have to be used; to this end, the two techniques TBVP and Tracking are compared. For the Tracking technique only the final version is taken into account, that corresponds to the best version for its realization. The profiles of position and speed are very similar, the substantial differences between the two techniques are shown for the acceleration profiles and therefore the respective angles, for this reason only the latter will be shown and analyzed. By overlapping the results obtained we get the differences between the behavior of the angular profiles for the two techniques taken into consideration. This is shown in the following figure:

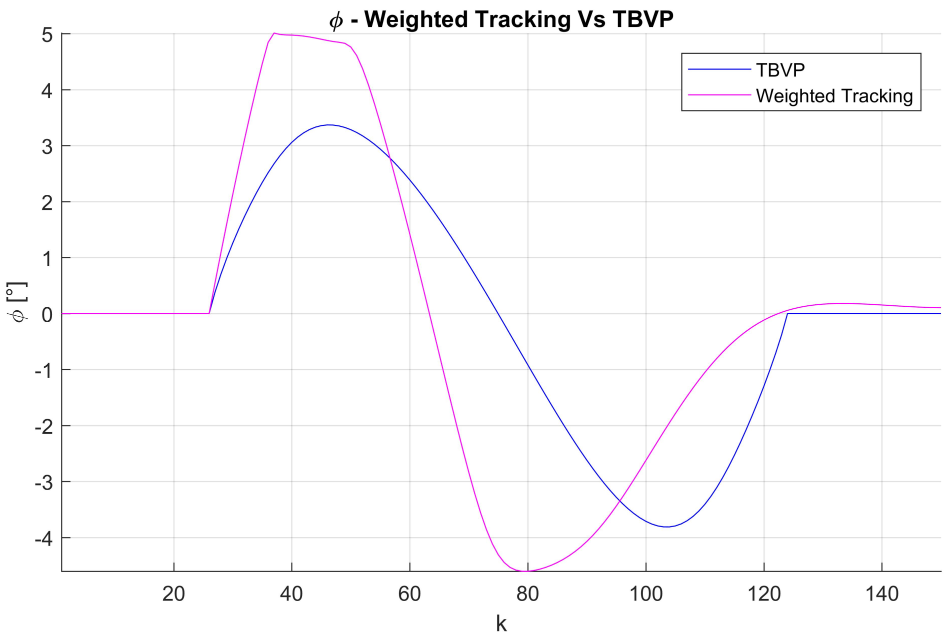

Figure 7.23: TBVP Vs Weighted Tracking

In the figure the blue line represents the behavior of the angular profile obtained by using the TBVP constrained technique, while, the magenta line represents the behavior of the angular profile obtained through the use of the Tracking technique. As can be seen from the figure, the behaviours of the two curves is very similar and the substantial differences between the results derives from the composition of the performance index. In fact, in the TBVP approach the cost function is formed only by the differences between the couples accelerations deriving from the use of the differential flatness method. While, for the Tracking approach the performance index is formed by a term deriving from the differential flatness method and a term that takes into account the minimization between the state and its reference, where this two terms are weighted through ρ and γ parameters. The results shown by both techniques allow to reach the prefixed aim. However, for the real implementation is necessary to consider the running time of algorithms. For the resolution of a single optimization using the TBVP technique, approximately 15-16s are required for the first up to 2 s for the last one. The difference in the running time time between the first solution of the optimization and the last one derives from the fact that matrices are scaled for k for each time step. Since for the first optimization with prediction horizon of $t = 15\,s \rightarrow N = 150$ and sampled at T_s, 900 constraints are needed and this obviously takes a long time. For this reason the problem is reduced for every k instant. Moreover, for the use of this technique, since the initial and final boundary conditions are defined at a determined instant k, it is necessary to define the constraints which describe the dynamics for all the instants k up to the prediction horizon N. While all other happens using the Tracking technique. In fact, this tecnique is just a relaxed version of TBVP technique where the optimization problem is left much free from the Boundary condition allowing the system to evolve freely minimizing the difference between the state and its target. The problem formulated, allows obtaining the minimum distance from the reference minimizing the difference between the couples accelerations. This means that the system does not necessarily have to reach the target but only minimize the distance in the chosen time horizon. Based on these considerations, for Tracking approach it has been possible to choose temporal horizons of predictions much smaller than those chosen for TBVP, which obviously reflects on a notable reduction of the matrices that define the number of constraints and therefore also of the calculation times. By applying the tracking strategy, to get the solution of the optimization problem, 2.6 s are necessary for each k instant over a control time horizon equal to $P = 10$. This

allows to state that a Tracking approach is much more convenient for the realization of Guidance optimization that has the shortest possible running time.
For this reason, this tecnique have been used for the real implementation. Now, defining a properly reference vector, it is possible to track all type of trajectories. For example by choosing a reference trajectory vector with infinity shape, it is possible to show how quadrocopter tracks this kind of references:

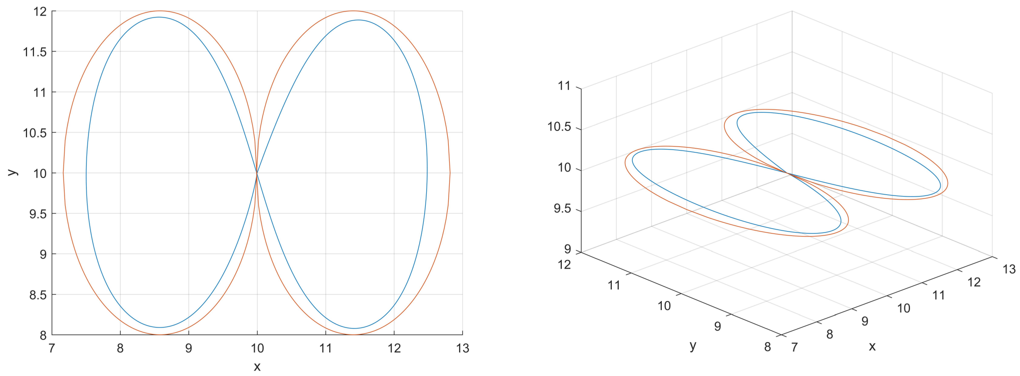

Figure 7.24: Traiettoria Tracking

where the blu line is the reference signal and the red one is the path followed by quadrocopter. Thanks to the properties of tracking approach no infeasibility occurs on the overall trajectory for each time step.

Chapter 8

Conclusion

The obtained results, that have been described in the previous chapter, allow a direct comparison between the new optimized discrete-time constrained guidance and the already present unconstrained polynomial guidance. The main difference between these strategies is the presence of constraints of the dynamic system in one case and the absence of them in the other one. The designed Constrained Guidance method gives the possibility to respect the input and output constraints through the Model Predictive Control technique. While uncostrained polynomial technique does not give this possibility inasmuch is based on the placement of polynomial coefficient to satisfy boundary condition. The new strategy implies an important improvement for a trajectory planning strategy. Moreover, due to this property, using the MPC technique, the implementation of new ideas, such as an obstacle avoidance, could be possible. Therefore, it can be state that the MPC tecnique is more flexible method than the unconstrained polynomial one. Taking into account that this technique is an advanced optimal control method, wich could be used to resolve trajectory optimization problem, means to provide not only guidance but also control.

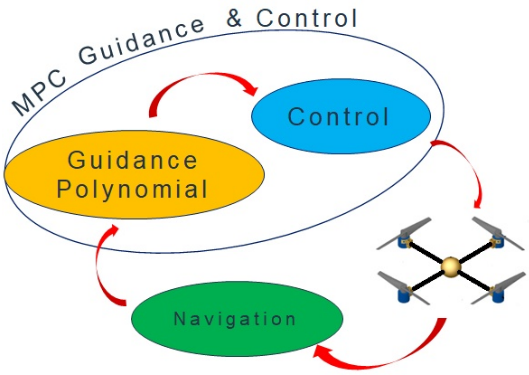

Figure 8.1: Polynomial vs MPC blocks

While on the one hand an improvement could be done, on the other one the running time of algorithm would be influenced. All the simulations was performed on a PC with processor characteristics: dual core, i7 , 2.00 Ghz. The running time of the algorithm takes 1.5-2s to arrive to the solution for Constrained optimization problem using MPC technique in Tracking mode. While polynomial guidance algorithm provide solution in 0.001s since the ammount of calculation is realy low. These results are summarized considering that the running time increases proportionaly w.r.t. the complexity of the algorithm, where for the optimized guidance the complexity of the algorithm is very high which reflects on the future cost.

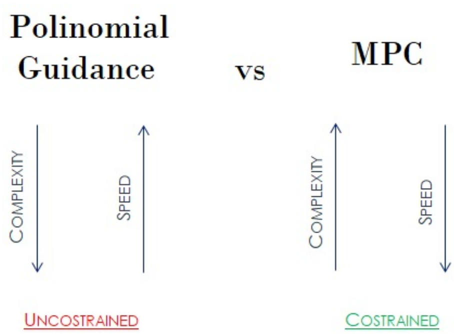

Figure 8.2: Complexity

All these consideration can be reported as advantages and disadvantages in the following table

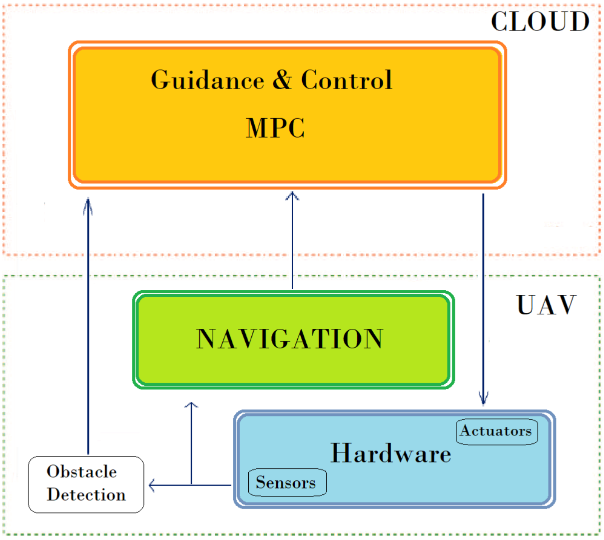

Figure 8.3: Cloud architecture

	Polynomial	*MPC*
Advantages	• *Low run time*	• *Constrained* • *Flexible* • *Guidance & Control*
Disadvantages	• *Unconstrained* • *Less flexible* • *Unconstrained*	• *High runn time CVX*

Table 8.1: Polynomial vs MPC

The running time obtained to arrive to the solution of the optimization problem is not short enough for a direct implementation on the aircraft, where the system is calibrated on much lower sampling times. Furthermore, the computing power that is available on board is not sufficient to solve an optimization problem in the desired time. For this reason, as the literature suggests, to obtain lower running time need for the real implementation, it is necessary to use expedient as cloud real time comunication or solving the problem through specifically designed algorithm. The assumption of one of this approach is mandatory inasmuch the optimization problem must be resolved for each k steps in usefull time.

The Cloud and the UAV could be connected through a wireless network and Internet mobile technologies (4G or 5G in the future) with On-board Connection diagnostic to monitor connection quality. Where due to the bad connection, the on-board control will takes over for a safe navigation.

Obstacle avoidance

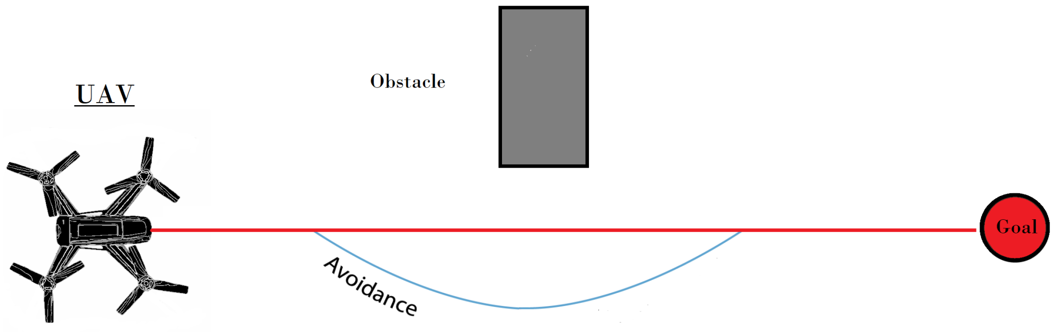

Figure 8.4: Obstacle avoidance

Cloud adoption can be a good choice to get computation times consistent with the system sampling time.

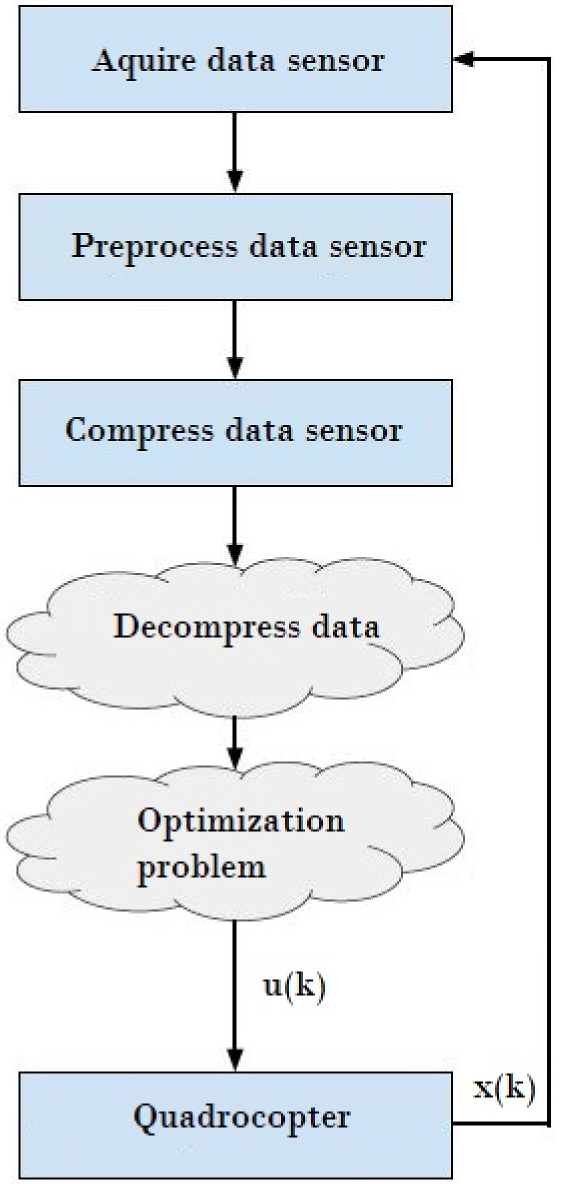

Figure 8.5: Cloud

Bibliography

[1] Chen Yongbo at al. Trajectory optimization of multiple quad-rotor uavs in collaborative assembling task. *Chinese Journal of Aeronautics*, 22 October 2015.

[2] Randal W. Beard. Quadrotor dynamics and control. *Brigham Young University*, 03 October 3, 2008.

[3] Hai bin Duan et al. Max-min adaptive ant colony optimization approach to multi-uavs coordinated trajectory replanning in dynamic and uncertain environments. *Journal of Bionic Engineering*, 2009.

[4] Encyclopedia Britannica. Newtons laws of motion. *https://www.britannica.com/science/mechanics*, 08-03-2018.

[5] E. Canuto. Planetary landing: modelling and control of propulsion descent. *et al. 2, 2013, Zhongguo Kexue Jishu Daxue Xuebao , Vol. 43. In press.*, Febbraio 2013.

[6] Enrico. Canuto. Embedded model control: outline of the theory. *ISA Transactions Vol. 46, 3, pp. 363-377*, 2007.

[7] Enrico Canuto. The embedded model control applied to the entry, descent and landing on mars. *Pisa, Italy, : Convegno Annuale dei Docenti e Ricercatori Italiani in Automatica*, September 2011.

[8] Enrico Canuto. Embedded model control approach to robust control. *Convegno Annuale dei Docenti e Ricercatori Italiani in Automatica*, September 2011.

[9] Enrico Canuto and Luis David. Patras Prieto. Embedded model control: principles and applications. part i. *IEEE, 2007. IEEE Conference on Emerging Technologies and Factory Automation*, 2007.

[10] Molano-Jimenez Andres Canuto Enrico and Perez-Montenegro Carlos. Disturbance rejection in space applications: Problems and solutions. *30th Chinese Control Conference*, July 22-24, 2011.

[11] Perez-Montenegro Carlos. *GUIDANCE AND CONTROL FOR BOREA QUADROTOR PROJECT*. PhD thesis, Politecnico di torino, 2013.

[12] A. Chamseddine, Y. Zhang, and C. A. Rabbath. Trajectory planning and replanning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles. *2012 American Control Conference (ACC)*, 2012.

[13] Guan chen Luo et al. Uav path planning in mixed obstacle environment via artificial potential field method improved by additional control force. *Asian Journal of Control*, 2010.

[14] Catalina Circa. Modelistica e simulazione del sistema di propulsuione del quadrirotore borea. Technical report, Pontificia Universidad Javeriana, 2012.

[15] Wikipedia contributors. *Guidance system*. Wikipedia, The Free Encyclopedia, 2018.

[16] Wikipedia contributors. *Euler angles*. Wikipedia, The Free Encyclopedia, 8 March 2018 17:04 UTC.

[17] Wikipedia contributors. *Rigid body dynamics*. Wikipedia, The Free Encyclopedia., 8 March 2018 17:06 UTC.

[18] Canuto Enrico. Embedded model control calls for disturbance modeling and rejection. *ISA TRANSACTIONS, Vol. 51, pp. 584-595. ISSN 0019-0578.*, 2012.

[19] Canuto E. et al. Embedded model control urges disturbance modelling and rejection. In *Yantai : 30th Chinese Control Conference (CCC), pp 6267 - 6273. ISBN: 978-1-4577-0677-6*, 2011.

[20] Dikmen et al. Attitude control of a quadrotor. In *Istanbul. International Conference on Recent Advances in Space Technologies RAST '09. 4th pp 722 -727*, 2009,ISBN: 978-1-4244-3627-9.

[21] L. W. Tsai et al. Robot analysis: The mechanics of serial and parallel manipulators,. *John-Wiley, NY*, 1999.

[22] Lee et al. Trajectory tracking control of quadrotor uav. In *11th International Conference on Control, Automation and Systems (ICCAS), pp. 281 - 285*, 2011, ISBN 978 1 4577 0835 0.

[23] Su et al. Application of mimu/magnetometer integrated system on the attitude determination of micro satellite. In *International Conference on Intelligent Mechatronics and Automation, Proceedings, pp. 579 - 583, ISBN: 0 7803 8748 1*, . Chengdu,China 2004.

[24] Tanenhaus et al. Utilization of gps/mems-imu for measurement of dynamics for range testing of missiles and rockets. *Position Location and Navigation Symposium*, 1998, pp 244 - 252.

[25] Zhang et al. Receding horizon control for multi-uavs close formation control based on differential evolution. *Science China Information Sciences*, 2014.

[26] Douglas Fields. Galilean relativity" (pdf). *Physics 262-01 Fall 2015, University of New Mexico*, retrieved June 5, 2017.

[27] BOARD SENSORS IDENTIFICATION and CONTROL ON THE DESIGNED TEST BENCH FOR BOREA PROJECT. Marco fontana. Master's thesis, POLITECNICO DI TORINO, 2014.

[28] Reza Kamyar and Ehsan Taher. Aircraft optimal terrain-threat-based trajectory planning and control. *Journal of Guidance, Control, and Dynamics*, 2014.

[29] Jack Langelaan and Steve Rock. Navigation of small uavs operating in forests. In *Rhode Island : American Institute of Aeronautics and Astronautics- Guidance, Navigation and Control Conference,. pp. 1-11.*, 2004.

[30] Lopez-CarmonaTurin. *Hadware Implementation of Measurement Unit for Borea Quadricopter*. PhD thesis, Politecnico Di Torino, 2012.

[31] Mauricio Alejandro Lotufo. Progetto borea: Implementazione algoritmi di guida, navigazione e controllo (gnc). Master's thesis, POLITECNICO DI TORINO, unknown Years.

[32] Raffaello D'andrea Markus Heln. Quadrocopter trajectory generation and control. *ETH zurich*, 2017.

[33] Stefano Mosca. *Modellistica e Simulazione Della Navigazione Di un Quadricottero.* PhD thesis, Turin-Italy : Politecnico di Torino, 2012.

[34] B. Paul. *Kinematics and Dynamics of Planar Machinery.* Prentice-Hall, NJ, 1979.

[35] Carlos Norberto Perez Montenegro and Enrico. Canuto. http://spautomatics.polito.it/s&pa/borea.php. borea official web site. [online]. Technical report, Politecnico di torino, 2013.

[36] C. perez-Montenegro et al. Saturation managing for thr propulsion system and optimal dispatching proposal for miulti-rotor uav. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics(AIM)*, 8-11 July, 2014.

[37] A. Behcet Acikmese Scott R. Ploen and Aron Wolf. A comparison of powered ddescent guidancelaws for mars pinpoint landing. In *AIAA/AAS Astrodynamics specialist conference and exhibit*, 21-24 August 2006, keystone,Colorado.

[38] Ashish Tewari. *ADVANCED CONTROL OF AIRCRAFT, SPACECRAFT AND ROCKETS*. John Wiley & Sons, Ltd, 2011. ISBN 9780470745632., 2011.

[39] RAFAEL YANUSHEVSKY. *Guidance of unmanned aerial vehicles.* Taylor & Francis Group, 2011. ISBN 978-1-4398-5095-4.

[40] Yao and Zhao. Unmanned aerial vehicle dynamic path planning in an uncertain environment. *Robotica*, 2015.

[41] Yu Zhang, Jing Chen, and Lincheng Shen. Real-time trajectory planning for ucav air-to-surface attack using inverse dynamics optimization method and receding horizon control. *Chinese Journal of Aeronautics*, 2013.