Pontificia Universidad Javeriana

Thesis presented in partial fulfillment of the requirements for the degree of Master of Science

Nonlinear Model Predictive Control for Aggressive Maneuvers in a Variable Pitch Quadrotor based on the Extended Modal Series Method

Ing. Carlos Andrés Devia

supervised by
Ing. Diego Alejandro Patiño Ph.D
Ing. Julian Colorado Ph.D

Bogotá, Colombia
June 2018

Hay varias personas a las que quiero agradecer por diferentes razones, a mi mente vienen sin duda alguna mi director de trabajo de grado el profesor Diego Patiño y mi co-director Julián Colorado por todo su continuo apoyo durante la realización del trabajo en estos últimos dos años. También naturalmente pienso en mi mamá y mi hermano que si bien no me ayudaron de forma directa en la realización del trabajo, se que sin ellos habría sido mucho más complicado y no habría sido lo mismo.

Por último no puedo dejar de reconocer a ciertas personas a quienes no solo les quiero agradecer sino que también quiero dedicarles este trabajo, pues me han acompañado de forma constante durante el último año y medio e hicieron que la experiencia durante mi maestría fuera extremadamente divertida y enriquecedora, que son mis estudiantes.

En particular hay algunos que han generado un gran impacto positivo en mi vida a los cuales quiero hacer la mención especial: a Cami; sin duda alguna eres una de las personas más inteligentes que conozco y me siento muy feliz de haberte conocído, se que llegarás muy lejos y muy pronto podrás estar haciendo trabajos de este y más alto nivel y verdaderamente espero poder estar ahí cuando eso pase. A Carolina, este último año habría sido completamente diferente si usted no hubiera estado conmigo para poder hablar y compartir cuando quería distraerme un poco y pasar un buen momento; su compañia durante todo este tiempo fue algo muy valioso e importante para mi y me trae muy buenos recuerdos. A Eliana, quien muchas veces parecía que leyera mi mente, porque solo con verme sabía lo que estaba pensando y lo que estaba sintiendo; se que siempre puedo contar con usted cuando necesito hablar de algo o simplemente pasar un buen rato. A Alejandra, que tiene una habilidad sorprendente para ponerme a pensar. A Santiago, que siempre (casi siempre) estaba un paso atrás de Cami. Y a muchos más estudiantes con los que compartí y pasé buenos momentos como Maria José, Stephany, Cristian, Manuela, Laura, Jelitza, Ángela, y básicamente todos aquellos cuyos nombres me aprendí.

Estos últimos dos años han sido de los mejores de mi vida y de los más pesados. Debo reconocer que eso es más que por la realización de este trabajo de investigación, por mi trabajo como profesor. A veces me sorprendo a mi mismo al darme cuenta de todo lo que me gusta de dar clases. Desde pensar lo que voy a decir, cómo y porqué lo voy a decir, encontrar cosas nuevas que no sabía o nuevas perspectivas al momento de preparar las clases, buscar formas ingeniosas y divertidas de hacer la clase, de presentar los ejercicios y los talleres. Debo admitir que a veces se me iba la mano con la creatividad en los ejercicios, talleres y parciales, pero simplemente habían cosas muy hermosas para dejar de ponerlas. Disfruto al preparar las clases, al hacer talleres y ejercicios y resolverlos yo mismo, al responder preguntas y sobretodo al ver cómo mis estimados estudiantes van aprendido de cierta medida lo que quiero enseñarles. Todo lo anterior no tiene nada que ver directamente con mi trabajo de investigación, pero para mi está tan íntimamente ligado con el desarrollo de mi maestría que no pude evitar ponerlo acá (y pues es mi trabajo, entonces básicamente puedo poner lo que quiera).

Carlos Andrés Devia
25.05.2018

Contents

Chapter 1

Introduction

The aggressive maneuvers are movements of a system which require or produce operation in the nonlinear part of the model or near the limits of the actuators or the states [1]. These maneuvers are characterized by fast changes in the values of the state variables as well as the use of their whole dynamic range.

The execution of aggressive maneuvers can be divided in two parts: the first part consists on the planning of the trajectory and the second part is the execution. The trajectory planning usually is stated as an optimal control problem which includes in the cost function the characteristics of the desired trajectory: geometric form of the path, obstacle avoidance, the dynamics and constraints of the system. In the literature, it is possible to find many trajectory planning algorithms such as Potential Fields [2], Dynamic Programming [3] and Model Predictive Control [4].

Once the trajectory has been planned is necessary to execute it. Classical methods of control that depend on linear approximations do not work (PID, State-Feedback). This is because given the nature of the trajectory the system operates in highly nonlinear regions and as a consequence the linear approximations are not valid, consequently a linear model-based controller will fail. Taking that into account there are two ways to execute the maneuver.

1. The first one is the Iterative Learning Control [1] [5]. It starts with a trajectory and a control signal which is the result of an optimal control problem that uses a simplified model based on basic principles. Then, it executes the maneuver. After the execution of the maneuver, the result is analyzed and the control signal improved. The process is repeated and in each iteration the control signal is adjusted according to the result of the execution, until the performance is satisfactory. This approach deals with the nonlinearities of the system and the non-modeled aerodynamic effects using the learning algorithm to modify the control signals in order to produce the desired outcome.

2. The second alternative is to consider a complete model of the quadrotor [6]. Then, it is possible to use this model with the desired trajectory and the system constraints to find the control signals as a solution of an optimal control problem. These signals in principle could produce the desired maneuver. In order to increase the robustness of the control, it is implemented in closed loop. By closing the loop the Optimal Control becomes a Model Predictive Control. Given that the system operates in the nonlinear regime, this technique is known as a Nonlinear Model Predictive Control (NMPC).

Besides the NMPC, there are other ways to execute aggressive maneuvers such as Sliding Mode Control [7], Transverse Linearization [8], Quaternions [9] and Neural Networks [10]. However, NMPC has three main advantages over these methods. First, it handles multiple input multiple output (MIMO) systems easily, second it can explicitly take into account the nonlinearities of the model [11] which is highly desirable for this application, and third the restrictions can be added to the controller design.

The advantages of the NMPC come associated with a speed implementation cost. Since it is a control based on a nonlinear real time optimization, the applications are limited by the speed of the optimization algorithm [12]. This implies that in order to use NMPC in a fast dynamic system it is necessary to employ a very efficient nonlinear optimization algorithm [13].

There are many NMPC implementations that differ mainly in the nonlinear optimization algorithm: Recursive Neural Networks [12] Differential Evolution [14], Sequential Quadratic Programming [11], Particle Swarm Optimization [11], Dynamic Particle Swarm Optimization [15], Optimistic Optimization [16], Neighboring External Updates [17], Extended Modal Series Method [18], among others.

Each one of these methods has its own characteristics which make them suitable for certain types of problems. In this particular case, the most relevant method is the Extended Modal Series Method (EMSM) because it is designed for input affine systems and does not require iterations to give a solution. The EMSM is based on a explicit suboptimal control law that depends only on the initial state of the system, which diminishes greatly the computation time of the control signal. This is important because the execution of an aggressive maneuver requires short sample periods for fast responses.

The system that will be used to test this controller is a modification of the classical fixed pitch quadrotor. A quadrotor is an Unmanned Aerial Vehicle (UAV) formed by a cross structure that in the center has the electronics and the battery and four motors with the corresponding propellers at the end of each arm. The movements of the quadrotor are completely determined by the differences of the thrust and torque produced by each motor [19], which in turn are related to the form of the blade and the angular velocity of rotation.

Usually, in order to control a quadrotor, the thrust and torque produced by each of the motors is assumed to be directly proportional to the square of the angular velocity of the motor and the proportionality constants are estimated or measured experimentally. This constant depends on the shape of the blade, in particular the angle of attack.

As a result, the produced thrust can be changed not only through the angular velocity of the motors but also changing the angle of attack of the blades. This last option is available for a special kind of quadrotors and has the advantage of affecting the thrust directly while the change of angular velocity has to pass first through the dynamics of the motor to change the real angular velocity of the propellers and then the thrust, hence it is limited to the bandwidth of the motors [20].

The quadrotors that can change the angle of attack of the blades are called the Variable Pitch Quadrotors and despite being relatively recent, the evident advantage of not being limited to the motor bandwidth has already being used to develop quadrotors that perform aggressive maneuvers with good results [21]. Another advantage of these quadrotors is the possibility to produce negative thrust, which enables them to perform maneuvers impossible for fixed pitch quadrotors such as flips with constant altitude or inverted hover [20].

Because their characteristics, the Variable Pitch Quadrotors are an excelent alternative to perform aggressive maneuvers. However, given that until recently their study has started there is still much to explote related to the execution of aggressive maneuvers, for instance the use of advanced control techniques such as NMPC for this application.

The goal of this work is to implement a NMPC to a Variable Pitch Quadrotor using the EMSM. More precisely the general objective is:

- **Design a Nonlinear Model Predictive Control law based on the Extended Modal Series Method for the execution of Aggressive Maneuvers in a Variable Pitch Quadrotor.**

In order to achieve this result the following specific objectives were stated:

1. Model Mathematically a Variable Pitch Quadrotor including aerodynamic effects such as Blade Flapping, Induced Drag, Translational Drag, Profile Drag, and Parasitic Drag.

2. Compute the trajectories corresponding to the aggressive maneuvers using Pontryagin's Maximum Principle to solve the associated Optimal Control Problem.

3. Implement in simulation a Nonlinear Model Predictive Control for the execution of the aggressive maneuvers using the Extended Modal Series Method.

4. Compare the performance of the Nonlinear Model Predictive Controller with a Quaternion-based one used in previous works.

The first, and main contribution is the implementation of a EMSM-based NMPC [18], which to the best of the author's knowledge has not been done before. This provide insights on the EMSM-based controller design that could produce in improvements on the method and become reference for future works on related topics.

The second contribution of this work is the application of NMPC to the execution of aggressive maneuvers in this system, since it includes the construction of a complete model (not only of the augmented model of the Variable Pitch Quadrotor but also the inclusion of relevant aerodynamic effects) the resulting model can be used for future research in this area.

The partial results of this master work will be presented in the 2018 European Control Conference in Limassol, Cyprus, under the conference article ***C. Devia, M. Roa, D. Patino, and J. Colorado. 2018. Towards a Nonlinear Model Predictive Control using the Extended Modal Series Method. European Control Conference -ECC 2018, June 12-15, Limassol, Cyprus.***

This work is divided as follows: In chapter 2 the complete dynamic system is explained and detailed. In chapter 3 the Extended Modal Series Method is presented, initially the preliminary results are shown and from there the main theoretical contribution of this work is developed. After that, in chapter 4 the control architecture is addressed and each controller described and explained. Finally, chapter 5 presents the simulation results and corresponding analysis ending with overall conclusions and future work. Additional details or theoretical background are located in the appendix .

Chapter 2

Complete Dynamic System Model

This chapter presents the variable pitch quadrotor model. Section 2.1 describes the complete model. An alternate model for the attitude dynamics based on quaternions is presented in section 2.2. Finally drag effects are addressed in section 2.3.

2.1 Variable Pitch Quadrotor

Quadrotors have become a widely studied topic in the recent years. This is due to the fact that they offer many desired characteristics for a wide range of applications. They can achieve vertical landing and take-off [22], perform aggressive maneuvers [23], and can be changed accordingly to the application requirements, varying size, the number of propellers or adding additional sensors.

Basic Quadrotor Model

In order to model the system two different coordinate frames must be defined. The first one, usually denoted by $E = \{\hat{x}, \hat{y}, \hat{z}\}$ corresponding to the inertial frame of reference fixed on the ground and $B = \{\hat{x_b}, \hat{y_b}, \hat{z_b}\}$ located in the center of mass of the quadrotor, where $\hat{z_b}$ point upwards and $\hat{x_b}$ points to the front of the robot, see figure 2.1.

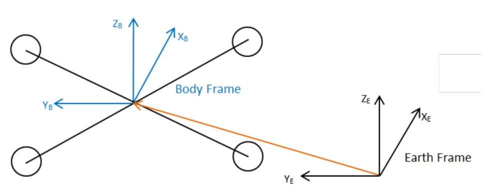

Figure 2.1: E and B frames of reference, taken from [19]

The two frames of reference are related by two vectors: the translational position $\vec{P} = [x, y, z]$ which is the vector from the origin of E to the origin of B and the angular position $\vec{\Theta} = [\phi, \theta, \psi]$ which corresponds to the Euler angles relating E and B. Figure 2.2 shows the relations between the Euler Angles and the axis of the B frame of reference.

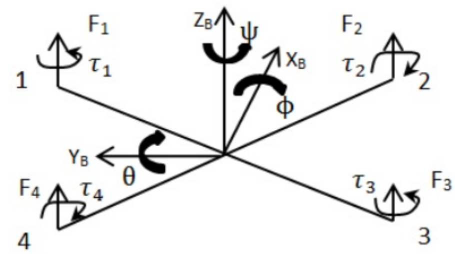

Figure 2.2: B frame of reference, taken from [19]

The equations of motion of the quadrotor can be derived using several methods, including newton's equations and energy principles. The used model that includes the basic dynamics is a combination of the results of previous works [19] [24] [25] [26] [27].

Basic Movements

The inputs of the system are the angular velocity of the four motors, denoted by w_1, w_2, w_3, w_4 respectively. Only through these inputs is possible to move the quadrotor, and since it has six degrees of freedom (three translational and three angular) it is characterized as an under-actuated system. The angular velocity generates thrust F_i and torque τ_i in each motor ($i = 1, 2, 3, 4$). The direction of rotation of each motor is opposite to the adjacent ones, this is done to produce torque both upwards and downwards and to control the yaw angle. It is through the thrust and torque of each motor that the quadrotor moves in space. This is achieved with four basic movements [19], each associated with a corresponding equation:

- **Climb**: this is the movement in the $\hat{z_b}$ direction. This movement is achieved by applying equal thrust to each of the four motors. The climb is represented by the equation of Total Thrust:
$$F_t = F_1 + F_2 + F_3 + F_4 \tag{2.1}$$
- **Roll Rotation**: In order to rotate the roll angle a torque must be applied in the direction of $\hat{x_b}$, this is carried out by increasing the thrust of the motors M1 and M4 and reducing the thrust of M2 and M3. By doing so, a net torque is applied in the $\hat{x_b}$ direction which results in a roll rotation. The thrust of M1 and M4 must be the same, as well as the thrust of M2 and M3, if not, the torque will point in a direction different from $\hat{x_b}$. If L is the distance from the motors to the center of mass, the Roll Torque is given by:
$$\tau_\phi = \frac{L}{\sqrt{2}}(F_1 - F_2 - F_3 + F_4) \tag{2.2}$$
- **Pitch Rotation**: Similar to the roll rotation, a pitch rotation is achieved by applying a torque in the body of the quadrotor, now in the $\hat{y_b}$ axis. This is achieved by increasing the thrust of motors M4 and M3 and decreasing the thrust of motors M1 and M2. Also in order to guarantee that the torque points in the $\hat{y_b}$ direction, the thrust in M1 and

M2 must be the same, as with M3 and M4. If L is the distance from the motors to the center of mass, the Pitch Torque is given by:

$$\tau_\theta = \frac{L}{\sqrt{2}}(-F_1 - F_2 + F_3 + F_4) \tag{2.3}$$

- **Yaw Rotation**: The yaw rotation is achieved by a torque in the direction of the $\hat{z}_b$ axis. This torque is the sum of the contributions of each motor. Then if they all produce the same torque the sum cancels. Hence, in order for a net torque to exist, two opposite motors must increase the angular velocity while the other two need to decrease it, resulting in a net torque in the $\hat{z}_b$ which in turn produces a yaw rotation. The Yaw Torque is given by:

$$\tau_\psi = \tau_1 - \tau_2 + \tau_3 - \tau_4 \tag{2.4}$$

Each of the four movements can be achieved independently with the adequate increase and decrease of angular velocity in the corresponding motors. Also, complex trajectories can be decomposed in each of these four movements in which case the total changes in the angular velocity of the motors will be the composition of the changes required by each of the independent movements.

Complete System of the Quadrotor

With each of the four basic movements defined and associated with an equation, the system dynamics can be addressed. In total 12 variables are needed to describe the system behavior (without taking into account the motor dynamics). The torques over the center of mass τ_ϕ, τ_θ and τ_ψ (roll, pitch and yaw torques respectively) produce angular accelerations $\dot{p}$, $\dot{q}$ and $\dot{r}$ in the B coordinate frame given by:

$$\dot{p} = \left(\frac{I_y - I_z}{I_x}\right) qr + \frac{\tau_\phi}{I_x} \tag{2.5}$$

$$\dot{q} = \left(\frac{I_z - I_x}{I_y}\right) pr + \frac{\tau_\theta}{I_y} \tag{2.6}$$

$$\dot{r} = \left(\frac{I_x - I_y}{I_z}\right) qp + \frac{\tau_\psi}{I_z} \tag{2.7}$$

Where I_x, I_y and I_z are the moments of inertia around the $\hat{x_b}$, $\hat{y_b}$ and $\hat{z}_b$ axis respectively. These accelerations can be integrated and transformed into angular velocities in the E coordinate frame:

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{2.8}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{2.9}$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \tag{2.10}$$

An integration of these angular velocities gives the angular position of the quadrotor $\vec{\Theta}$, which together with the total trust F_t generates translational accelerations in the E frame of reference. These accelerations can be expressed in terms of the translational position by:

$$\ddot{x} = \frac{F_t}{m}[\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi] \tag{2.11}$$

$$\ddot{y} = \frac{F_t}{m}[\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi] \tag{2.12}$$

$$\ddot{z} = \frac{F_t}{m}\cos\phi\cos\theta - g \tag{2.13}$$

Where m is the mass of the quadrotor and g is the gravity.

Motor and Propeller Model

Besides the dynamics of the quadrotor in flight, the model of the motor, the propellers and its coupling needs to be taken into account. There are two main ways to analyze the dynamics of the rotors: the energy method and the blade element method [28].

The detailed model of the propellers during flight is very complicated because when the propellers move their shape changes due to the air that flows around them, as a result the total torque and thrust needs to be calculated using an integral. This is a level of detail that is unnecessary and unfeasible for any application mainly because it introduces many new parameters and variables that need to be estimated because even in test conditions are difficult to measure and second because its effects are negligible. As a result a close form equation can be obtained after doing some approximations of the average shape of the propeller [29].

Finally the thrust and torque produced by each propeller is given by (i denotes each motor):

$$F_i = \rho c R^3 \omega^2 C_{L\alpha}\Big(\frac{\alpha_t}{3} - \frac{L}{2R\omega}(E_1 q - E_2 p)\Big) \qquad i = 1,2,3,4 \tag{2.14}$$

$$\tau_i = \rho c R^4 \omega^2 \Big(\frac{C_{D0} + C_{Di}{\alpha_t}^2}{4} - \frac{C_{L\alpha}\alpha_t \bar{w}}{3Rw}\Big) \qquad i = 1,2,3,4 \tag{2.15}$$

where ρ is the air density, c is the chord of the blade, ω is the angular velocity of rotation of the propeller, $C_{L\alpha}$ is the lift coefficient, D_{D0} and C_{Di} are the parasitic and lift-induced drag coefficient respectively. R is the blade radius, α_t is the angle of attack, L is the distance from the center of the quadrotor to the propeller, $\bar{w}$ is the velocity in the $\hat{z}_b$ direction. E_1 is 0 for motors 2 and 4, 1 for motor 1 and -1 for motor 3. And E_2 is 0 for motors 1 and 3, 1 for motor 2 and -1 for motor 4.

It is important to note that equations (2.14) and (2.15) are only the absolute value of the thrust and torque of each motor. The direction of the thrust is in the $sgn(\alpha_t)\hat{z}_b$ direction and for the torque the directions alternate between adjacent propellers from $\hat{z}_b$ to $-\hat{z}_b$. Because of that, ω in the equations above is independent of the direction of rotation and always a positive number. It is important to note that although the equations seem very complex, at the end they have the following form (i denotes each motor):

$$F_i = K_{1_i}\omega^2\alpha_t + K_{2_i}\omega \qquad i = 1,2,3,4 \tag{2.16}$$

$$\tau_i = B_{1_i}\omega^2 + B_{2_i}\omega^2{\alpha_t}^2 + B_{3_i}\omega\alpha_t \qquad i = 1,2,3,4 \tag{2.17}$$

where the constants K_{1_i}, K_{2_i}, B_{1_i}, B_{2_i} and B_{3_i} can be calculated from measurements.

The motor dynamics can be modeled as a series RCL circuit [23]:

$$v = R_i + L\frac{\partial i}{\partial t} + \frac{w}{K_v} \tag{2.18}$$

where v is the voltage, R_i is the internal resistance, i is the current, w is the rotational velocity of the rotor, K_v is the voltage constant of the motor. On the other hand the torque produced by the motor is:

$$\tau_M = \frac{i - i_o}{K_Q} \tag{2.19}$$

where i_o is the no-load current and K_Q is the torque constant. The previous considerations can be simplified into a first order equation, thus the dynamics of the motor are modeled by [25] with:

$$\dot{\omega}_i = K_{m_i}({\omega_i}^{des} - \omega_i) \qquad i = 1,2,3,4 \tag{2.20}$$

where K_{m_i} is a constant that enclosures the parameters of the motor and ${\omega_i}^{des}$ is the desired angular velocity.

2.2 Quaternion Model

This section focuses on the quaternion-based attitude dynamic model. Appendix A.3 contains a brief explanation of quaternions and their use as rotation operators in space. If the reader is not familiar with this mathematica tool is highly advised to read first appendix A.3.

Recall that when modeling the quadrotor, two reference frames are needed, the inertial frame of reference E and the body frame of reference B. The torques acted on the body frame of reference and together with the thrust were translated into forces in the inertial frame of reference. The relation between these two frames was given by the angular position of the quadrotor represented by Euler's angles ϕ, θ and ψ.

This is the conventional approach and is used in many applications. However, it has two main disadvantages: first, it is computationally expensive since it requires constant calculation of trigonometric functions and secondly it suffers from singularities for some angular positions. The solution is to employ quaternions to represent these rotations in space as explained in the previous subsections.

Using quaternions to represent these rotations implies replacing the angle-based position towards a quaternion-based one. Meaning that the attitude of the quadrotor will be represented by a quaternion $q(t) = q_0(t) + q_1(t)\hat{i} + q_2(t)\hat{j} + q_3(t)\hat{k}$, where q_0, q_1, q_2 and q_3 are the quaternion components.

This change is reflected in the dynamic model equations. Since only the ϕ, θ and ψ angles are replaced only the equations containing these state variables are affected, it means that the angular rate dynamics remain the same while the attitude and translational velocities dynamics change to

$$\begin{aligned} \dot{q}_0 &= \tfrac{1}{2}(-q_1 p - q_2 q - q_3 r) & \dot{q}_2 &= \tfrac{1}{2}(q_0 q - q_1 r + q_3 p) \\ \dot{q}_1 &= \tfrac{1}{2}(q_0 p + q_2 r - q_3 q) & \dot{q}_3 &= \tfrac{1}{2}(q_0 r + q_1 q - q_2 p) \end{aligned} \tag{2.21}$$

$$\ddot{x} = 2\tfrac{F_t}{m}(q_1 q_3 + q_0 q_2) \quad \ddot{y} = 2\tfrac{F_t}{m}(q_2 q_3 - q_0 q_1) \quad \ddot{z} = \tfrac{F_t}{m}(q_0^2 - q_1^2 - q_2^2 - q_3^2) - g \tag{2.22}$$

For details about this result see [30]. Note that although the equations in (2.21) are still non linear the two main disadvantages of the angle attitude dynamics are overcome: no trigonometric functions need to be computed and the equations have no singularities.

2.3 Drag Effects

This section briefly presents the main drag forces presented in the complete model:

- **Blade Flapping** This effect is caused by the translational motion of the propeller. As the propeller rotates the tip velocity of the blade is affected by the translational motion of the quadrotor: while the blade moves in the direction of displacement of the quadrotor the velocity of the tip is greater than while it moves in the opposite direction. This difference in tip velocities generates a flapping movement on the tip of the rotor that can be modeled by:

  $$\beta(\varphi) = \beta_0 + \beta_c \cos(\varphi) + \beta_s \sin(\varphi),$$

  where φ is the azimuth angle of the rotor and β is the flapping angle. This changes the angle of attack of the blade, thus affecting the produced torque. The flapping force on the motor i is given by:

  $$\Delta_i = T_i\Big(A_{\text{flap}}\frac{V_i}{\omega_i} + B_{\text{flap}}\frac{\Omega}{\omega_i}\Big) \tag{2.23}$$

  where $V_i = (v_x, v_y, 0)^T$, $\Omega = (\phi, \theta, \psi)^T$ and

  $$A_{\text{flap}} = \frac{1}{r}\begin{bmatrix} -A_{1c} & A_{1s} & 0 \\ -A_{1s} & A_{1c} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B_{\text{flap}} = \begin{bmatrix} -B_2 & B_1 & 0 \\ B_1 & -B_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.24}$$

  The constants A_{1c} and A_{1s} depend on the geometry of the blade. While B_1 and B_2 can be approximated numerically (see [6]).

- **Induced Drag** The motor experiences a net instantaneous induced drag that directly opposes the direction of apparent wind (V_P) as seen by the rotor and that is proportional to the velocity of V_P:

  $$D_I = K_I V_P \qquad V_P = (V_x, V_y, 0)^T \tag{2.25}$$

  The proportionality constant K_I depends on the geometry of the propeller.

- **Translational Drag** Also known as the momentum drag, it is a drag caused by the induced velocity of the airflow as it goes through the rotor:

$$D_T = \begin{cases} K_{T_1} V_P & \text{low speeds} \\ K_{T_2} V_z^4 V_P & \text{high speeds} \end{cases} \qquad V_P = (V_x, V_y, 0)^T \tag{2.26}$$

  The constants K_{T_1} and K_{T_2} can be calculated using Blade Element Theory.

- **Profile Drag** It is due to the transverse velocity of the rotor blades. It is zero at hover and can be modeled as:

$$D_P = \rho c((c_{d0} + c_{d\alpha})\Omega r^2 + V_z r)V_p \qquad V_P = (V_x, V_y, 0)^T \tag{2.27}$$

  Equation (2.27) can be approximated by a directly proportional relation $D_P \approx K_P V_P$ for ease in the implementation.

- **Parasitic Drag** It is a drag caused by the non-lifting surfaces of the quadrotor. It includes drag arising from the airframe, motors and que guidance and control systems:

$$D_{par} = K_{par}|V|V \qquad V = (V_x, V_y, V_z)^T \tag{2.28}$$

Unlike the previous effects, this drag is present in the complete body of the quadrotor rather than each propeller, as such it is modeled in the overall system and not each motor model. Finally, the drag effects can be summarized by:

$$\begin{cases} \tau_\phi = \frac{L}{\sqrt{2}}(F_{z_1} - F_{z_2} - F_{z_3} + F_{z_4}) \\ \tau_\theta = \frac{L}{\sqrt{2}}(-F_{z_1} - F_{z_2} + F_{z_3} + F_{z_4}) \\ \tau_\psi = \tau_1 - \tau_2 + \tau_3 - \tau_4 - \frac{L}{\sqrt{2}}(F_{y_1} + F_{x_1} + F_{y_2} - F_{x_2} - F_{y_3} - F_{x_3} - F_{y_4} + F_{x_4}) \end{cases} \tag{2.29}$$

Where

$$F_{x_i} = \frac{1}{\omega_i}(\rho c R_p^3 \omega_i^2 C_{L\alpha})\frac{\alpha}{3}\left(\frac{1}{r}(A_{1c}v_x + A_{1s}v_y) - B_2 p + B_1 q\right) - (K_I + K_T + K_P)v_x$$
$$F_{y_i} = \frac{1}{\omega_i}(\rho c R_p^3 \omega_i^2 C_{L\alpha})\frac{\alpha}{3}\left(\frac{1}{r}(A_{1s}v_x - A_{1c}v_y) + B_2 p - B_1 q\right) - (K_I + K_T + K_P)v_y$$
$$F_{z_i} = \rho c R_p^3 \omega_i^2 C_{L\alpha}\frac{\alpha}{3}$$

The model presented in this chapter is the one that will be used throughout the remainder of the work. In particular in chapter 4 where the control is designed. As expected, the way in which the model is divided into the motor-propeller system and the quadrotor system will be reflected in the control design. Furthermore, the attitude and translational dynamics subdivision of the quadrotor model will be present in the control architecture.

Chapter 3

Extended Modal Series Method

This chapter elaborates on the Extended Modal Series Method (EMSM) applied to the solution of an Optimal Control Problem (OCP). Section 3.1 states the OCP to be solved and briefly explains the procedure to solve the problem. Then section 3.2 presents the preliminary results based on [18].

Next, in section 3.3 the main theoretical result of this work is presented, namely an explicit formulation for the systems of ordinary differential equations, which were defined implicitly in the previous section. First the desired system of ordinary differential equations is presented, then the main result follows in theorem 1 with the corresponding proof.

Thanks to the closed form of the system of ordinary differential equations it is possible to approximate a solution. Section 3.4 addresses this problem. Finally in section 3.5 the complete procedure is summarized and the final result shown.

3.1 Problem Statement

Consider the following input-affine dynamic system:

$$\dot{x} = F(x) + G(x)u \tag{3.1}$$

Where $F : \mathbb{R}^n \to \mathbb{R}^n$ and $G : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are analytic vector functions of the states $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ is the system input. Satisfying $F(0) = 0$, $G(0) = 0$. The OCP is stated as follows:

$$\min_{u\in U} \left\{ J(x, u) = \frac{1}{2}\int_{t_o}^{t_f} \left(x^T Qx + u^T Ru\right) dt \right\}$$

Where U is the set of admissible inputs $u(t)$, Q is a positive definite matrix, R is a positive semi definite matrix, t_o is the initial time and t_f is the final time. Subject to the system dynamics of equation 3.1 and with fixed initial and final states, $x(t_o) = x_o$ and $x(t_f) = x_f$ respectively. In order to solve this OCP Pontryagin's Maximum Principle is used [31]. First compute the Hamiltonian corresponding to the system 3.1:

$$H(x, \lambda, u) = \frac{1}{2}\left(x^T Qx + u^T Ru\right) + \lambda^T \Big(F(x) + G(x)u\Big) \tag{3.2}$$

Where $\lambda = \lambda(t) \in \mathbb{R}^n$ are known as the costates. Now minimize the hamiltonian 3.2 with respect to the input u to find the optimal input u^*

$$u^* = \min_{u \in U} \Big\{ H(x, \lambda, u) \Big\}$$

Replace u^* in the hamiltonian 3.2 to obtain the optimal hamiltonian H^*. Finally solve the set of $2n$ differential equations:

$$\dot{x} = \frac{\partial H^*}{\partial \lambda} \qquad\qquad \dot{\lambda} = -\frac{\partial H^*}{\partial \lambda} \tag{3.3}$$

Subject to the boundary conditions $x(t_o) = x_o$ and $x(t_f) = x_f$. Replacing the dynamic system 3.1 in the hamiltonian 3.2 yields the following optimal control input:

$$u^* = -R^{-1} G^T(x) \lambda \tag{3.4}$$

Finally computing the system of differential equations 3.3 gives:

$$\dot{x} = F(x) - G(x) R^{-1} G(x)^T \lambda \tag{3.5}$$

$$\dot{\lambda} = -Qx - \partial_x F(x)^T \lambda + \begin{bmatrix} \lambda^T \partial_{x_1} G(x) R^{-1} G(x)^T \lambda \\ \vdots \\ \lambda^T \partial_{x_n} G(x) R^{-1} G(x)^T \lambda \end{bmatrix} \tag{3.6}$$

The solution to this system of nonlinear differential equations gives the optimal control as well as the optimal state trajectory. However, it often can not be found analytically and approximations are required.

The reminder of this chapter details the use of the Extended Modal Series Method to approximate the solution of this system of partial differential equations. In section 3.2 the solutions of equations (3.5) and (3.6) are approximated by an infinite sum of vector functions g_i and h_i which are themselves solutions of infinite systems of ordinary differential equations. The systems that give rise to the functions g_i and h_i are expressed in closed form in section 3.3. Next, in section 3.4 the closed form is further developed and an approximate solution found. Finally in section 3.5 the complete procedure is summarized and the final result shown.

3.2 Preliminary Results

The Extended Modal Series Method can be used to transform equations 3.5 and 3.6 into a series of ordinary differential equations that have analytic solutions [18]. Let us define the functions

$$\psi(x, \lambda) \triangleq F(x) - G(x) R^{-1} G(x)^T \lambda \tag{3.7}$$

$$\phi(x, \lambda) \triangleq -Qx - \partial_x F(x)^T \lambda + \begin{bmatrix} \lambda^T \partial_{x_1} G(x) R^{-1} G^T(x) \lambda \\ \vdots \\ \lambda^T \partial_{x_n} G(x) R^{-1} G^T(x) \lambda \end{bmatrix} \tag{3.8}$$

where $\psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are nonlinear analytic vector functions (since F and G are assumed to be analytic). With these definitions the system of nonlinear differential equations can be written as:

$$\dot{x} = \psi(x, \lambda) \qquad \dot{\lambda} = \phi(x, \lambda) \tag{3.9}$$

Expanding ψ and ϕ in a Taylor Series around the point $(x, \lambda) = (0, 0)$. It is obtained:

$$\begin{aligned}\dot{x} = \left(\frac{\partial \psi}{\partial x}\Big|_{\substack{x=0\\ \lambda=0}}\right)x + \left(\frac{\partial \psi}{\partial \lambda}\Big|_{\substack{x=0\\ \lambda=0}}\right)\lambda + \frac{1}{2}\begin{bmatrix} x^T\left(\frac{\partial^2 \psi_1}{\partial x^2}\big|_{\substack{x=0\\ \lambda=0}}\right)x \\ \vdots \\ x^T\left(\frac{\partial^2 \psi_n}{\partial x^2}\big|_{\substack{x=0\\ \lambda=0}}\right)x \end{bmatrix} \\ + \begin{bmatrix} x^T\left(\frac{\partial^2 \psi_1}{\partial x \partial \lambda}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \\ \vdots \\ x^T\left(\frac{\partial^2 \psi_n}{\partial x \partial \lambda}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \lambda^T\left(\frac{\partial^2 \psi_1}{\partial \lambda^2}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \\ \vdots \\ \lambda^T\left(\frac{\partial^2 \psi_n}{\partial \lambda^2}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \end{bmatrix} + \dots\end{aligned} \tag{3.10}$$

$$\begin{aligned}\dot{\lambda} = \left(\frac{\partial \phi}{\partial x}\Big|_{\substack{x=0\\ \lambda=0}}\right)x + \left(\frac{\partial \phi}{\partial \lambda}\Big|_{\substack{x=0\\ \lambda=0}}\right)\lambda + \frac{1}{2}\begin{bmatrix} x^T\left(\frac{\partial^2 \phi_1}{\partial x^2}\big|_{\substack{x=0\\ \lambda=0}}\right)x \\ \vdots \\ x^T\left(\frac{\partial^2 \phi_n}{\partial x^2}\big|_{\substack{x=0\\ \lambda=0}}\right)x \end{bmatrix} \\ + \begin{bmatrix} x^T\left(\frac{\partial^2 \phi_1}{\partial x \partial \lambda}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \\ \vdots \\ x^T\left(\frac{\partial^2 \phi_n}{\partial x \partial \lambda}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \lambda^T\left(\frac{\partial^2 \phi_1}{\partial \lambda^2}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \\ \vdots \\ \lambda^T\left(\frac{\partial^2 \phi_n}{\partial \lambda^2}\big|_{\substack{x=0\\ \lambda=0}}\right)\lambda \end{bmatrix} + \dots\end{aligned} \tag{3.11}$$

Assuming that the solution has the form:

$$x(t) = \sum_{i=1}^{\infty} g_i(t) \qquad \lambda(t) = \sum_{i=1}^{\infty} h_i(t) \tag{3.12}$$

Since the trajectory of the states and the costates depends on the time and the initial conditions, it is possible to define the functions Λ and Γ as:

$$x(t) = \Lambda(x_o, t) \qquad \lambda(t) = \Gamma(x_o, t) \tag{3.13}$$

Now multiplying by a factor ϵ the initial condition x_o and replacing in $x(t) = \Lambda(x_o, t)$ and $\lambda(t) = \Gamma(x_o, t)$ it is obtained:

$$x_\epsilon(t) = \Lambda(\epsilon x_o, t) = \epsilon g_1(t) + \epsilon^2 g_2(t) + \cdots = \sum_{i=1}^{\infty} \epsilon^i g_i(t) \tag{3.14}$$

$$\lambda_\epsilon(t) = \Gamma(\epsilon \lambda_o, t) = \epsilon h_1(t) + \epsilon^2 h_2(t) + \cdots = \sum_{i=1}^{\infty} \epsilon^i h_i(t) \tag{3.15}$$

Derivating $x_\epsilon(t)$ and $\lambda_\epsilon(t)$ with respect to time in (3.14) and (3.15) to obtain $\dot{x}_\epsilon(t) = \psi(x_\epsilon(t), \lambda_\epsilon(t))$ and $\dot{\lambda}_\epsilon(t) = \phi(x_\epsilon(t), \lambda_\epsilon(t))$. This gives the following system (using (3.10) and (3.11)):

$$\epsilon\dot{g}_1(t) + \epsilon^2\dot{g}_2(t) + \cdots = \epsilon\left[\left(\frac{\partial\psi}{\partial x}\right)g_1(t) + \left(\frac{\partial\psi}{\partial\lambda}\right)h_1(t)\right] + \epsilon^2\left[\left(\frac{\partial\psi}{\partial x}\right)g_2(t) + \left(\frac{\partial\psi}{\partial\lambda}\right)h_2(t) + \frac{1}{2}\begin{bmatrix} g_1^T(t)\left(\frac{\partial^2\psi_1}{\partial x^2}\right)g_1(t) \\ \vdots \\ g_1^T(t)\left(\frac{\partial^2\psi_n}{\partial x^2}\right)g_1(t)\end{bmatrix} + \begin{bmatrix} g_1^T(t)\left(\frac{\partial^2\psi_1}{\partial x\partial\lambda}\right)h_1(t) \\ \vdots \\ g_1^T(t)\left(\frac{\partial^2\psi_n}{\partial x\partial\lambda}\right)h_1(t)\end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1^T(t)\left(\frac{\partial^2\psi_1}{\partial\lambda^2}\right)h_1(t) \\ \vdots \\ h_1^T(t)\left(\frac{\partial^2\psi_n}{\partial\lambda^2}\right)h_1(t)\end{bmatrix}\right] + \ldots \tag{3.16}$$

$$\epsilon\dot{h}_1(t) + \epsilon^2\dot{h}_2(t) + \cdots = \epsilon\left[\left(\frac{\partial\phi}{\partial x}\right)g_1(t) + \left(\frac{\partial\phi}{\partial\lambda}\right)h_1(t)\right] + \epsilon^2\left[\left(\frac{\partial\phi}{\partial x}\right)g_2(t) + \left(\frac{\partial\phi}{\partial\lambda}\right)h_2(t) + \frac{1}{2}\begin{bmatrix} g_1^T(t)\left(\frac{\partial^2\phi_1}{\partial x^2}\right)g_1(t) \\ \vdots \\ g_1^T(t)\left(\frac{\partial^2\phi_n}{\partial x^2}\right)g_1(t)\end{bmatrix} + \begin{bmatrix} g_1^T(t)\left(\frac{\partial^2\phi_1}{\partial x\partial\lambda}\right)h_1(t) \\ \vdots \\ g_1^T(t)\left(\frac{\partial^2\phi_n}{\partial x\partial\lambda}\right)h_1(t)\end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1^T(t)\left(\frac{\partial^2\phi_1}{\partial\lambda^2}\right)h_1(t) \\ \vdots \\ h_1^T(t)\left(\frac{\partial^2\phi_n}{\partial\lambda^2}\right)h_1(t)\end{bmatrix}\right] + \ldots \tag{3.17}$$

Equations 3.16 and 3.17 hold for arbitrary ϵ. Thus, the coefficients that correspond to each power of ϵ must be equal. This fact generates infinite systems of differential equations, each associated with a power of ϵ and of $2n$ equations (n states and n costates):

$$\epsilon := \begin{cases} \dot{g}_1 = \left(\frac{\partial\psi}{\partial x}\right)g_1 + \left(\frac{\partial\psi}{\partial\lambda}\right)h_1 \\ \dot{h}_1 = \left(\frac{\partial\phi}{\partial x}\right)g_1 + \left(\frac{\partial\phi}{\partial\lambda}\right)h_1 \end{cases} \qquad \epsilon^2 := \begin{cases} \dot{g}_2 = \left(\frac{\partial\psi}{\partial x}\right)g_2 + \left(\frac{\partial\psi}{\partial\lambda}\right)h_2 + \frac{1}{2}\begin{bmatrix} g_1^T\left(\frac{\partial^2\psi_1}{\partial x^2}\right)g_1 \\ \vdots \\ g_1^T\left(\frac{\partial^2\psi_n}{\partial x^2}\right)g_1\end{bmatrix} \\ \quad\cdots + \begin{bmatrix} g_1^T\left(\frac{\partial^2\psi_1}{\partial x\partial\lambda}\right)h_1 \\ \vdots \\ g_1^T\left(\frac{\partial^2\psi_n}{\partial x\partial\lambda}\right)h_1\end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1^T\left(\frac{\partial^2\psi_1}{\partial\lambda^2}\right)h_1 \\ \vdots \\ h_1^T\left(\frac{\partial^2\psi_n}{\partial\lambda^2}\right)h_1\end{bmatrix} \\ \dot{h}_2 = \left(\frac{\partial\phi}{\partial x}\right)g_2 + \left(\frac{\partial\phi}{\partial\lambda}\right)h_2 + \frac{1}{2}\begin{bmatrix} g_1^T\left(\frac{\partial^2\phi_1}{\partial x^2}\right)g_1 \\ \vdots \\ g_1^T\left(\frac{\partial^2\phi_n}{\partial x^2}\right)g_1\end{bmatrix} \\ \quad\cdots + \begin{bmatrix} g_1^T\left(\frac{\partial^2\phi_1}{\partial x\partial\lambda}\right)h_1 \\ \vdots \\ g_1^T\left(\frac{\partial^2\phi_n}{\partial x\partial\lambda}\right)h_1\end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1^T\left(\frac{\partial^2\phi_1}{\partial\lambda^2}\right)h_1 \\ \vdots \\ h_1^T\left(\frac{\partial^2\phi_n}{\partial\lambda^2}\right)h_1\end{bmatrix} \end{cases}$$

Solving the system due to ϵ gives the functions $g_1(t)$ and $h_1(t)$, then substituting in the system generated by ϵ^2 it becomes a system of linear time-invariant non-homogeneous differential equations whose solution is $g_2(t)$ and $h_2(t)$. This process can be repeated recursively in order to obtain the functions $g_i(t)$ and $h_i(t)$ [18].

3.3 Explicit Formulation

Equations (3.16) and (3.17) give a constructive method to form the systems of differential equations. For the ω-th iteration there are $2n$ unknown functions:

$$g_\omega(t) = (g_{\omega,1}(t) \dots g_{\omega,n}(t))^T \quad h_\omega(t) = (h_{\omega,1}(t) \dots h_{\omega,n}(t))^T$$

Let $\xi^\psi_{\omega,l}$ be the function corresponding to $\dot{g}_{\omega,l}$, that is, the component l of the function that multiplies ϵ^ω expanded from ψ. Likewise let $\xi^\phi_{\omega,l}$ be the function corresponding to $\dot{h}_{\omega,l}$. With this notation the system of ordinary differential equations that corresponds to the iteration ω is:

$$\dot{g}_{\omega,l}(t) = \xi^\psi_{\omega,l}(t) \qquad \dot{h}_{\omega,l}(t) = \xi^\phi_{\omega,l}(t) \tag{3.18}$$

Where $1 \leq l \leq n$ and the unknowns are the vector functions $g_\omega(t)$ and $h_\omega(t)$.

The main result is a closed form formula for the functions $\xi^\psi_{\omega,l}(t)$ and $\xi^\phi_{\omega,l}(t)$. This result is stated in the following theorem:

Theorem 1. *Using the previously introduced notation, let $\xi^\psi_{\omega,l}$ be the function corresponding to ϵ^ω for g_l. Likewise let $\xi^\phi_{\omega,l}$ be the function corresponding to ϵ^ω for h_l. Then they can be calculated as follows:*

$$\xi^\psi_{\omega,l} = \sum_{|\alpha|+|\beta|=1}^{\omega} \left\{ \frac{D^\alpha D^\beta}{\alpha!\beta!} \psi_l(0) \Omega(\alpha,\beta,\omega) \right\} \tag{3.19}$$

$$\xi^\phi_{\omega,l} = \sum_{|\alpha|+|\beta|=1}^{\omega} \left\{ \frac{D^\alpha D^\beta}{\alpha!\beta!} \phi_l(0) \Omega(\alpha,\beta,\omega) \right\} \tag{3.20}$$

Where

$$\Omega(\alpha,\beta,\omega) = \sum_{|\theta|=\omega-|\alpha|-|\beta|} \left\{ \prod_{q=1}^{n} \left(\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \leq u \leq \omega\alpha_q \\ \beta_q \leq v \leq \omega\beta_q}} GH \right) \right\} \tag{3.21}$$

$$G = G(\alpha_q, u) = \sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}} \left(\frac{\alpha_q!}{a_1!a_2!\dots a_\omega!} \right) \left(\prod_{t=1}^{\omega} g_{t,q}^{a_t} \right) \tag{3.22}$$

$$H = H(\beta_q, v) = \sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}} \left(\frac{\beta_q!}{b_1!b_2!\dots b_\omega!} \right) \left(\prod_{t=1}^{\omega} h_{t,q}^{b_t} \right) \tag{3.23}$$

$$\sigma(x) = \sum_{t=1}^{\omega} t x_t \tag{3.24}$$

Proof. First rewrite (3.10) and (3.11) using the compact form of Taylor's Formula [32] it is an expansion in $2n$ variables:

$$\begin{aligned} \dot{x} = \psi(x,\lambda) = \psi(\Lambda,\Gamma) &= \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!} \psi(0) \Lambda^\alpha \Gamma^\beta \\ &= \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{\partial^{\alpha_1}_{x_1} \dots \partial^{\alpha_n}_{x_n} \partial^{\beta_1}_{\lambda_1} \dots \partial^{\beta_n}_{\lambda_n}}{\alpha_1! \dots \alpha_n! \beta_1! \dots \beta_n!} \psi(0) \Lambda_1^{\alpha_1} \dots \Lambda_n^{\alpha_n} \Gamma_1^{\beta_1} \dots \Gamma_n^{\beta_n} \end{aligned} \tag{3.25}$$

$$\dot{\lambda} = \phi(x,\lambda) = \phi(\Lambda,\Gamma) = \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!}\phi(0)\Lambda^\alpha\Gamma^\beta$$

$$= \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{\partial_{x_1}^{\alpha_1}\dots\partial_{x_n}^{\alpha_n}\partial_{\lambda_1}^{\beta_1}\dots\partial_{\lambda_n}^{\beta_n}}{\alpha_1!\dots\alpha_n!\beta_1!\dots\beta_n!}\phi(0)\Lambda_1^{\alpha_1}\dots\Lambda_n^{\alpha_n}\Gamma_1^{\beta_1}\dots\Gamma_n^{\beta_n} \tag{3.26}$$

Where α and β are indices with n positions each: $\alpha = (\alpha_i)_{i=1}^n$, $\beta = (\beta_i)_{i=1}^n$ and $|\alpha| = \sum_{i=1}^n \alpha_i$. Also

$$\Lambda_k = \Lambda_k(x_o,t) = \sum_{j=1}^{\infty} g_{j,k}(t) \qquad \Gamma_k = \Gamma_k(x_o,t) = \sum_{j=1}^{\infty} h_{j,k}(t) \tag{3.27}$$

Now, changing the initial conditions $x(t_o) = \epsilon x_o$, we have:

$$\psi(x_\epsilon,\lambda_\epsilon) = \psi(\Lambda(\epsilon x_o,t),\Gamma(\epsilon x_o,t)) = \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!}\psi(0)\Lambda^\alpha(\epsilon x_o,t)\Gamma^\beta(\epsilon x_o,t) = \sum_{j=1}^{\infty}\epsilon^j \dot{g}_j(t) \tag{3.28}$$

$$\phi(x_\epsilon,\lambda_\epsilon) = \phi(\Lambda(\epsilon x_o,t),\Gamma(\epsilon x_o,t)) = \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!}\phi(0)\Lambda^\alpha(\epsilon x_o,t)\Gamma^\beta(\epsilon x_o,t) = \sum_{j=1}^{\infty}\epsilon^j \dot{h}_j(t) \tag{3.29}$$

and

$$\Lambda_k(\epsilon x_o,t) = \sum_{j=1}^{\infty}\epsilon^j g_{j,k}(t) \qquad \Gamma_k(x_o,t) = \sum_{j=1}^{\infty}\epsilon^j h_{j,k}(t)$$

Then, the expansion becomes:

$$\psi(x_\epsilon,\lambda_\epsilon) = \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!}\psi(0)\bar{\Omega}(\alpha,\beta) = \sum_{k=1}^{\infty}\epsilon^k \xi_k^\psi \tag{3.30}$$

$$\phi(x_\epsilon,\lambda_\epsilon) = \sum_{|\alpha|+|\beta|=1}^{\infty} \frac{D^\alpha D^\beta}{\alpha!\beta!}\phi(0)\bar{\Omega}(\alpha,\beta) = \sum_{k=1}^{\infty}\epsilon^k \xi_k^\phi \tag{3.31}$$

Where

$$\bar{\Omega}(\alpha,\beta) = \prod_{q=1}^{n}\Big(\sum_{k=1}^{\infty}\epsilon^k g_{k,q}(t)\Big)^{\alpha_q}\prod_{q=1}^{n}\Big(\sum_{k=1}^{\infty}\epsilon^k h_{k,q}(t)\Big)^{\beta_q} \tag{3.32}$$

The purpose is to calculate the coefficient of ϵ^ω for any ω. First consider $\bar{\Omega}(\alpha,\beta)$, for now truncate the expansion of x_ϵ and λ_ϵ up to the term m. Recalling the multinomial expansion:

$$\Big(\sum_{k=1}^{m}\epsilon^k g_{k,l}(t)\Big)^{\alpha_l} = \sum_{|k|=\alpha_l}\frac{\alpha_l!}{k_1!\dots k_m!}\Big(\prod_{t=1}^{m} g_{t,l}^{k_t}\Big)\epsilon^{\sigma(k)} \tag{3.33}$$

Where $\sigma(x) = \sum_{t=1}^m t x_t$. Since $k = (k_1,\dots k_m)$, and $|k| = \alpha_l$, then:

$$\min|\sigma(k)| = \alpha_l \qquad \max|\sigma(k)| = m\alpha_l$$

Thus we can rewrite (3.33) as:

$$\left(\sum_{k=1}^{m}\epsilon^k g_{k,l}(t)\right)^{\alpha_l} = \sum_{\rho=\alpha_l}^{m\alpha_l} F_l(\rho)\epsilon^\rho \tag{3.34}$$

The same can be done for the h's, yielding:

$$\bar{\Omega}(\alpha,\beta) = \prod_{q=1}^{n}\left(\sum_{\rho=\alpha_q}^{m\alpha_q} F_q(\rho)\epsilon^\rho\right)\prod_{q=1}^{n}\left(\sum_{\rho=\beta_q}^{m\beta_q} G_q(\rho)\epsilon^\rho\right) \tag{3.35}$$

Now, note that the minimum power of ϵ in this expansion is the sum of the minimum powers of each term of the product, i.e. $\sum_{q=1}^{n}(\alpha_q+\beta_q)$, while the maximum power is the sum of each maximum, i.e. $\sum_{q=1}^{n} m(\alpha_q+\beta_q)$.

Then, given α and β, the product $\Lambda^\alpha\Gamma^\beta$ only contributes to ϵ^ω, where $|\alpha|+|\beta| \leq \omega \leq m(|\alpha|+|\beta|)$. Recalling that m is the limit of the truncated series if $m \to \infty$ it is clear that for a given order ω it is unnecessary to consider vectors α and β that satisfy $\omega < |\alpha|+|\beta|$.

This allows to truncate the 'external Taylor series' (equations (3.25) and (3.26)) as well as the 'internal Taylor series' (equation (3.27)) up to ω instead of ∞.

Furthermore we have an expression for $\bar{\Omega}$:

$$\prod_{q=1}^{n}\left(\sum_{\rho=\alpha_q}^{\omega\alpha_q} F_q(\rho)\epsilon^\rho\right)\left(\sum_{\rho=\beta_q}^{\omega\beta_q} G_q(\rho)\epsilon^\rho\right) = \prod_{q=1}^{n}\left(\sum_{\mu=\alpha_q+\beta_q}^{\omega(\alpha_q+\beta_q)} \chi(\mu)\epsilon^\mu\right) \tag{3.36}$$

Where

$$\chi(\mu) = \sum_{\substack{u+v=\mu \\ \alpha_q \leq u \leq \omega\alpha_q \\ \beta_q \leq v \leq \omega\beta_q}} F_q(u)G_q(v) \tag{3.37}$$

Once more, the minimum power of (3.36) is the sum of the minimum powers of each term, and the maximum is the sum of all the maximum powers:

$$\prod_{q=1}^{n}\left(\sum_{\mu=\alpha_q+\beta_q}^{\omega(\alpha_q+\beta_q)} \chi(\mu)\epsilon^\mu\right) = \sum_{W=|\alpha|+|\beta|}^{\omega(|\alpha|+|\beta|)} \Theta(W)\epsilon^W \tag{3.38}$$

Where

$$\Theta(W) = \sum_{|\theta|=W-(|\alpha|+|\beta|)} \left(\prod_{q=1}^{n} \chi(\alpha_q+\beta_q+\theta_q)\right) \tag{3.39}$$

Going back to (3.30) and (3.31) (which now go to ω) we have:

$$\sum_{|\alpha|+|\beta|=1}^{\omega}\left(\frac{D^\alpha D^\beta}{\alpha!\beta!}\psi(0)\sum_{W=|\alpha|+|\beta|}^{\omega(|\alpha|+|\beta|)}\Theta(W)\epsilon^W\right) = \sum_{k=1}^{\omega^2}\epsilon^k\xi_k^\psi \tag{3.40}$$

$$\sum_{|\alpha|+|\beta|=1}^{\omega}\left(\frac{D^\alpha D^\beta}{\alpha!\beta!}\phi(0)\sum_{W=|\alpha|+|\beta|}^{\omega(|\alpha|+|\beta|)}\Theta(W)\epsilon^W\right) = \sum_{k=1}^{\omega^2}\epsilon^k\xi_k^\phi \tag{3.41}$$

Thus it follows directly that

$$\xi_\omega^\psi = \sum_{|\alpha|+|\beta|=1}^{\omega}\left(\frac{D^\alpha D^\beta}{\alpha!\beta!}\psi(0)\Theta(\omega)\right) \qquad \xi_\omega^\phi = \sum_{|\alpha|+|\beta|=1}^{\omega}\left(\frac{D^\alpha D^\beta}{\alpha!\beta!}\phi(0)\Theta(\omega)\right) \tag{3.42}$$

In order to write explicitly $\Theta(W)$, first consider $F_q(\rho)$ and $G_q(\rho)$, which first appear in (3.34). Evaluating at $\epsilon = 1$ and using the multinomial theorem it follows that:

$$F_q(u) = \sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}}\left(\frac{\alpha_q!}{a_1!a_2!\dots a_\omega!}\right)\left(\prod_{t=1}^{\omega} g_{t,q}^{a_t}\right) \tag{3.43}$$

$$G_q(v) = \sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}}\left(\frac{\beta_q!}{b_1!b_2!\dots b_\omega!}\right)\left(\prod_{t=1}^{\omega} h_{t,q}^{b_t}\right) \tag{3.44}$$

Which together with (3.37) and (3.39) give the desired result. □

3.4 Solution

Once a closed form for the system of ordinary differential equations is known, it is possible to elaborate on the solution of the system. In particular it can be transformed into a more suitable form and also an approximate solution proposed, which is exactly what is done in this section.

3.4.1 Non-homogeneous System of Differential Equations

We have the following equations:

$$\dot{g}_{\omega,l} = \sum_{|(\alpha,\beta)|=1}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!}\psi_l(0) \overbrace{\sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q\\ \alpha_q\le u\le\omega\alpha_q \\ \beta_q\le v\le \omega\beta_q}} \underbrace{\left(\sum_{\substack{\sigma(a)=u\\|a|=\alpha_q}} \frac{\alpha_q!}{a!}\prod_{k=1}^{\omega} g_{k,q}^{a_k}\right)}_{G(\alpha_q,u)} \underbrace{\left(\sum_{\substack{\sigma(b)=v\\|b|=\beta_q}} \frac{\beta_q!}{b!}\prod_{k=1}^{\omega} h_{k,q}^{b_k}\right)}_{H(\beta_q,v)} \right]\right]}^{\Omega(\alpha,\beta,\omega)} \right] \tag{3.45}$$

$$\dot{h}_{\omega,l} = \sum_{|(\alpha,\beta)|=1}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!}\phi_l(0) \overbrace{\sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q\\ \alpha_q\le u\le\omega\alpha_q \\ \beta_q\le v\le \omega\beta_q}} \underbrace{\left(\sum_{\substack{\sigma(a)=u\\|a|=\alpha_q}} \frac{\alpha_q!}{a!}\prod_{k=1}^{\omega} g_{k,q}^{a_k}\right)}_{G(\alpha_q,u)} \underbrace{\left(\sum_{\substack{\sigma(b)=v\\|b|=\beta_q}} \frac{\beta_q!}{b!}\prod_{k=1}^{\omega} h_{k,q}^{b_k}\right)}_{H(\beta_q,v)} \right]\right]}^{\Omega(\alpha,\beta,\omega)} \right] \tag{3.46}$$

With $1 \leq l \leq n$ equations (3.45) and (3.46) form the $2n$ system of first order differential equations for the iteration ω with unknowns $g_{\omega,l}$ and $h_{\omega,l}$.

Take (3.45), we want to 'extract' the unknown coefficient $g_{\omega,l}$ as follows:

1. To obtain $g_{\omega,l}$ we require in $G(\alpha_l, u)$ $a_\omega = 1$, this implies that $\sigma(a) = u \geq \omega$.

2. From the third sum we have that (for $q = l$) $u + v = \theta_l + \beta_l + \alpha_l \geq \omega$, since v must be positive .

3. From the second sum we have that:

   $$|\theta| + |\alpha| + |\beta| = \theta_1 + \cdots + \theta_n + \alpha_1 + \cdots + \alpha_n + \beta_1 + \cdots + \beta_n = \omega$$

   Since the vectors θ, β and α correspond to indices the coefficients must be positive and integers, hence we conclude that:

   - For $q = l$, the third sum has only one term given by $u = \omega$, $v = 0$, since $\theta_l + \beta_l + \alpha_l = \omega$.
   - For $q \neq l$, the third sum has only one term given by $u = 0$, $v = 0$, since $\theta_q + \beta_q + \alpha_q = 0$

4. Since $u = \omega$ we have in $G(\alpha_l, u)$ that $\sigma(a) = \omega$, and since $a_\omega = 1$ we conclude that $a = (0, 0, \ldots, 1)$, hence $|a| = \alpha_l = 1$.

5. Since $v = 0$ we have in $H(\beta_l, v)$ that the only option is $b = (0, \ldots, 0)$.

6. For $q \neq l$ since $u = v = 0$, the same applies for $G(\alpha_q, u)$ and $H(\beta_q, v)$. In sumary:

$$\begin{cases} G(\alpha_l, u) = g_{\omega,l} \\ \\ H(\beta_l, v) = 1 \end{cases} \text{for} \quad q = l \qquad\qquad \begin{cases} G(\alpha_q, u) = 1 \\ \\ H(\beta_q, v) = 1 \end{cases} \text{for} \quad q \neq l$$

7. So, we have that:

$$\prod_{q=1}^{n} \left(\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \leq u \leq \omega\alpha_q \\ \beta_q \leq v \leq \omega\beta_q}} G(\alpha_q, u) H(\beta_q, v) \right) = g_{\omega,l}$$

8. Now, in the second sum we have that $|\theta| + |\alpha| + |\beta| = \omega$ and we already know that:

$$\begin{cases} \alpha_l = q \\ \\ \theta_l + \beta_l = \omega - 1 \end{cases} \text{for} \quad q = l \qquad\qquad \begin{cases} \theta_q = 0 \\ \alpha_q = 0 \\ \beta_q = 0 \end{cases} \text{for} \quad q \neq l$$

   However, if $\beta_l \neq 0$ then the sum $H(\beta_l, v)$ would be empty because there exist no b such that $|b| = \beta_l \neq 0$ and $\sigma(b) = 0$. And the emptiness of $H(\beta_l, v)$ would imply that $H(\beta_l, v) = 0$ and the product $\prod_{q=1}^{n}$ would be zero. Thus we conclude that the only choice is $\theta_l = \omega - 1$ and $\beta_l = 0$.

9. As a result the second sum has only one term, given by the index vectors:

$$\begin{aligned} \alpha &= (0, 0, \ldots, \overbrace{1}^{l\text{-th place}}, \ldots, 0) \\ \beta &= (0, 0, \ldots, 0, \ldots, 0) \\ \theta &= (0, 0, \ldots, \omega - 1, \ldots, 0) \end{aligned}$$

10. These vectors also define the first sum, hence we conclude that $g_{\omega,l}$ appears only once in 3.45 and is multiplied by $\frac{\partial}{\partial x_l}\psi(0)$

The same analysis applies for $h_{\omega,l}$, in this case the index vectors become:

$$\alpha = (0, 0, \ldots, 0, \ldots, 0) \qquad \beta = (0, 0, \ldots, \overbrace{1}^{l\text{-th place}}, \ldots, 0) \qquad \theta = (0, 0, \ldots, \omega - 1, \ldots, 0)$$

And the coefficient is $\frac{\partial}{\partial \lambda_l}\psi(0)$. This can be extended to (3.46). Note that these vectors are the only ones that satisfy $|\alpha| + |\beta| = 1$. Hence equations (3.45) and (3.46) can be written as:

$$\dot{g}_{\omega,l} = \frac{\partial}{\partial x_1}\psi_1(0)g_{\omega,1} + \cdots + \frac{\partial}{\partial x_n}\psi_n(0)g_{\omega,n} + \frac{\partial}{\partial \lambda_1}\psi_1(0)h_{\omega,1} + \cdots + \frac{\partial}{\partial \lambda_n}\psi_n(0)h_{\omega,n} + \sum_{|\alpha|+|\beta|=2}^{\omega} \left\{ \frac{D^\alpha D^\beta}{\alpha!\beta!}\psi_l(0)\Omega(\alpha, \beta, \omega) \right\}$$

$$\dot{h}_{\omega,l} = \frac{\partial}{\partial x_1}\phi_1(0)g_{\omega,1} + \cdots + \frac{\partial}{\partial x_n}\phi_n(0)g_{\omega,n} + \frac{\partial}{\partial \lambda_1}\phi_1(0)h_{\omega,1} + \cdots + \frac{\partial}{\partial \lambda_n}\phi_n(0)h_{\omega,n} + \sum_{|\alpha|+|\beta|=2}^{\omega} \left\{ \frac{D^\alpha D^\beta}{\alpha!\beta!}\phi_l(0)\Omega(\alpha, \beta, \omega) \right\}$$

Or in vector form:

$$\begin{bmatrix} \dot{g}_\omega \\ \dot{h}_\omega \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x}\psi & \frac{\partial}{\partial \lambda}\psi \\ \\ \frac{\partial}{\partial x}\phi & \frac{\partial}{\partial \lambda}\phi \end{bmatrix} \Bigg|_{\substack{x=0\\ \lambda=0}} \begin{bmatrix} g_\omega \\ h_\omega \end{bmatrix} + \begin{bmatrix} \chi_\omega^\psi \\ \chi_\omega^\phi \end{bmatrix} \tag{3.47}$$

Where

$$\chi_{\omega,l}^\psi = \sum_{|(\alpha,\beta)|=2}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!}\psi_l(0) \sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \le u \le (\omega-1)\alpha_q \\ \beta_q \le v \le (\omega-1)\beta_q}} \left(\sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}} \frac{\alpha_q!}{a!} \prod_{k=1}^{\omega-1} g_{k,q}^{a_k} \right) \left(\sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}} \frac{\beta_q!}{b!} \prod_{k=1}^{\omega-1} h_{k,q}^{b_k} \right) \right] \right] \right] \tag{3.48}$$

$$\chi_{\omega,l}^\phi = \sum_{|(\alpha,\beta)|=2}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!}\phi_l(0) \sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \le u \le (\omega-1)\alpha_q \\ \beta_q \le v \le (\omega-1)\beta_q}} \left(\sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}} \frac{\alpha_q!}{a!} \prod_{k=1}^{\omega-1} g_{k,q}^{a_k} \right) \left(\sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}} \frac{\beta_q!}{b!} \prod_{k=1}^{\omega-1} h_{k,q}^{b_k} \right) \right] \right] \right] \tag{3.49}$$

And

$$g_\omega = (g_{\omega,1}, \dots, g_{\omega,n})^T \qquad h_\omega = (h_{\omega,1}, \dots, h_{\omega,n})^T$$

$$\chi_\omega^\psi = (\chi_{\omega,1}^\psi, \dots, \chi_{\omega,n}^\psi)^T \qquad \chi_\omega^\phi = (\chi_{\omega,1}^\phi, \dots, \chi_{\omega,n}^\phi)^T$$

It is clear that $\chi_1^\psi = 0$ and $\chi_1^\phi = 0$. Note that in equations (3.48) and (3.49) the products go to $\omega - 1$ instead of ω this is because the case $k = \omega$ would require $|(\alpha, \beta)| = 1$ which cannot happen due to the lower limit of the outer sum. Similarly the upper interval for u and v in the third sum go up to $(\omega - 1)\alpha_q$ and $(\omega - 1)\beta_q$ respectively since the limits $u = \omega\alpha_q$ and $v = \omega\beta_q$ are reached only for $a_\omega = 1$ and $b_\omega = 1$ which would imply $|(\alpha, \beta)| = 1$.

3.4.2 Approximations

In the previous section we found that in each iteration the system that has to be solved is given by equation (3.47), whose solution is:

$$\begin{bmatrix} g_\omega(t) \\ h_\omega(t) \end{bmatrix} = e^{(t-t_o)A} \begin{bmatrix} g_\omega(t_o) \\ h_\omega(t_o) \end{bmatrix} + \int_{t_o}^{t} e^{(t-s)A} \begin{bmatrix} \chi_\omega^\psi(s) \\ \chi_\omega^\phi(s) \end{bmatrix} ds \tag{3.50}$$

Where

$$A = \begin{bmatrix} \frac{\partial}{\partial x}\psi & \frac{\partial}{\partial \lambda}\psi \\ \\ \frac{\partial}{\partial x}\phi & \frac{\partial}{\partial \lambda}\phi \end{bmatrix} \Bigg|_{\substack{x=0\\ \lambda=0}} \tag{3.51}$$

Now, we can expand:

$$e^{(t-s)A} = \sum_{k=0}^{\infty} \frac{1}{k!}(t-s)^k A^k \approx I + (t-s)A + \frac{1}{2}(t-s)^2 A^2 + \frac{1}{6}(t-s)^3 A^3 + \dots \tag{3.52}$$

Taking the expansion to the N-th term, define:

$$\Phi_N(t,s) = \sum_{k=0}^{N} \frac{1}{k!}(t-s)^k A^k \approx I + (t-s)A + \frac{1}{2}(t-s)^2 A^2 + \dots + \frac{1}{N!}(t-s)^N A^N \tag{3.53}$$

For convenience $\Phi_N(t,s)$ can also be expressed as a 2×2 matrix of $n\times n$ matrices:

$$\Phi_N(t,s) = \begin{bmatrix} \Omega_N^{1,1}(t,s) & \Omega_N^{1,2}(t,s) \\ \\ \Omega_N^{2,1}(t,s) & \Omega_N^{2,2}(t,s) \end{bmatrix} \tag{3.54}$$

Thus, the general solution can me approximated by:

$$\begin{bmatrix} g_\omega(t) \\ h_\omega(t) \end{bmatrix} = \Phi_N(t,t_o)\begin{bmatrix} g_\omega(t_o) \\ h_\omega(t_o) \end{bmatrix} + \int_{t_o}^{t} \Phi_N(t,s)\begin{bmatrix} \chi_\omega^\psi(s) \\ \chi_\omega^\phi(s) \end{bmatrix} ds \tag{3.55}$$

Now, recalling the boundary conditions:

- For $\omega = 1$

$$g_\omega(t_o) = x_o \qquad g_\omega(t_f) = x_f \tag{3.56}$$

- For $\omega > 1$

$$g_\omega(t_o) = 0 \qquad g_\omega(t_f) = 0 \tag{3.57}$$

The functions g_ω and h_ω can be explicitly calculated using these boundary conditions:

- For $\omega = 1$ The explicit solution is:

$$\begin{aligned} g_1(t) &= \Omega_N^{1,1}(t)g_1(t_o) + \Omega_N^{1,2}(t)h_1(t_o) \\ h_1(t) &= \Omega_N^{2,1}(t)g_1(t_o) + \Omega_N^{2,2}(t)h_1(t_o) \end{aligned} \tag{3.58}$$

  Evaluating the first equation in $t = t_f$

$$g_1(t_f) = \Omega_N^{1,1}(t_f)g_1(t_o) + \Omega_N^{1,2}(t_f)h_1(t_o) \tag{3.59}$$

  Hence

$$h_1(to) = \Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Big(g_1(t_f) - \Omega_N^{1,1}(t_f)g(t_o)\Big) \tag{3.60}$$

  Replacing $g_1(t_o) = x_o$ gives:

$$\begin{aligned} g_1(t) &= \Omega_N^{1,1}(t)x_o + \Omega_N^{1,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Big(x_f - \Omega_N^{1,1}(t_f)x_o\Big) \\ h_1(t) &= \Omega_N^{2,1}(t)x_o + \Omega_N^{2,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Big(x_f - \Omega_N^{1,1}(t_f)x_o\Big) \end{aligned} \tag{3.61}$$

- For $\omega > 1$ The explicit solution is:

$$\begin{aligned} g_\omega(t) &= \Omega_N^{1,1}(t)g_\omega(t_o) + \Omega_N^{1,2}(t)h_\omega(t_o) + \textstyle\int_{t_o}^{t}(\Omega_N^{1,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{1,2}(t,s)\chi^\phi)_\omega(s))ds \\ h_\omega(t) &= \Omega_N^{2,1}(t)g_\omega(t_o) + \Omega_N^{2,2}(t)h_\omega(t_o) + \textstyle\int_{t_o}^{t}(\Omega_N^{2,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{2,2}(t,s)\chi^\phi)_\omega(s))ds \end{aligned} \tag{3.62}$$

  For convenience define:

$$\begin{aligned} \Theta_{N,\omega}^{g}(t,t_o) &= \int_{t_o}^{t}\Big[\Omega_N^{1,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{1,2}(t,s)\chi_\omega^\phi(s)\Big]ds \\ \Theta_{N,\omega}^{h}(t,t_o) &= \int_{t_o}^{t}\Big[\Omega_N^{2,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{2,2}(t,s)\chi_\omega^\phi(s)\Big]ds \end{aligned} \tag{3.63}$$

such that

$$\begin{aligned} g_\omega(t) &= \Omega_N^{1,1}(t)g_\omega(t_o) + \Omega_N^{1,2}(t)h_\omega(t_o) + \Theta_{N,\omega}^g(t,t_o) \\ h_\omega(t) &= \Omega_N^{2,1}(t)g_\omega(t_o) + \Omega_N^{2,2}(t)h_\omega(t_o) + \Theta_{N,\omega}^h(t,t_o) \end{aligned} \tag{3.64}$$

Taking into account that $g_\omega(t_o) = 0$ we can evaluate $t = t_f$ in the first equation and obtain:

$$g_\omega(t_f) = \Omega_N^{1,2}(t_f)h_\omega(t_o) + \Theta_{N,\omega}^g(t_f,t_o) = 0 \tag{3.65}$$

Hence

$$h_\omega(t_o) = -\left(\Omega_N^{1,2}(t_f)\right)^{-1}\Theta_{N,\omega}^g(t_f,t_o) \tag{3.66}$$

Finally we obtain

$$\begin{aligned} g_\omega(t) &= -\Omega_N^{1,2}(t)\left(\Omega_N^{1,2}(t_f)\right)^{-1}\Theta_{N,\omega}^g(t_f,t_o) + \Theta_{N,\omega}^g(t,t_o) \\ h_\omega(t) &= -\Omega_N^{2,2}(t)\left(\Omega_N^{1,2}(t_f)\right)^{-1}\Theta_{N,\omega}^g(t_f,t_o) + \Theta_{N,\omega}^h(t,t_o) \end{aligned} \tag{3.67}$$

Remarks:

1. Note that the approximation of the matrix exponential implies that all the matrices are really matrices of polynomials of t, thus all the multiplications, integrations and sums are in the end operations of polynomials, which result in a vector of polynomials. It is of interest to calculate the order of this polynomial, this is a long calculation but in the end the order of the polynomial for the ω-th order is

$$N_\omega = N + \omega N_{\omega-1} + 1$$

Where N is the order of the Taylor expansion of the exponential function and $N_0 = N$.

2. This equations make sense only if $\left(\Omega_N^{1,2}(t_f)\right)^{-1}$ is well defined.

Together equations 3.67 and 3.67 allow the explicit computation of the suboptimal control law that solves the associated optimal tracking control problem:

$$u_K(t) = -R^{-1}G^T\left(\sum_{\omega=1}^{K} g_\omega(t)\right)\left(\sum_{\omega=1}^{K} h_\omega(t)\right)$$

3.5 EMSM Summary of Results

To summarize this section: an approximate solution to the Optimal Control Problem

$$\min_{u\epsilon U}\left\{J(x,u) = \frac{1}{2}\int_{t_o}^{t_f}\left(x^TQx + u^TRu\right)dt\right\}$$
$$\text{s.t.} \quad \dot{x} = F(x) + G(x)u \quad x(t_o) = x_o \quad x(t_f) = x_f$$

Was found using the Extended Modal Series Method to approximate the solution of the system of partial differential equations given by the Maximum Principle:

$$\dot{x} = F(x) - G(x)R^{-1}G(x)^T\lambda$$

$$\dot{\lambda} = -Qx - \partial_x F(x)^T \lambda + \begin{bmatrix} \lambda^T \partial_{x_1} G(x) R^{-1} G(x)^T \lambda \\ \vdots \\ \lambda^T \partial_{x_n} G(x) R^{-1} G(x)^T \lambda \end{bmatrix}$$

This was done assuming that the solution had the form:

$$x(t) = \sum_{i=1}^{\infty} g_i(t) \qquad \lambda(t) = \sum_{i=1}^{\infty} h_i(t)$$

After some mathematical manipulation it was found that the functions $g_i(t)$ and $h_i(t)$ satisfied the following system of ordinary differential equations:

$$\dot{g}_{\omega,l} = \sum_{|(\alpha,\beta)|=1}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!} \psi_l(0) \overbrace{\sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \le u \le \omega\alpha_q \\ \beta_q \le v \le \omega\beta_q}} \underbrace{\Big(\sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}} \frac{\alpha_q!}{a!} \prod_{k=1}^{\omega} g_{k,q}^{a_k} \Big)}_{G(\alpha_q,u)} \underbrace{\Big(\sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}} \frac{\beta_q!}{b!} \prod_{k=1}^{\omega} h_{k,q}^{b_k} \Big)}_{H(\beta_q,v)} \right] \right]}^{\Omega(\alpha,\beta,\omega)} \right]$$

$$\dot{h}_{\omega,l} = \sum_{|(\alpha,\beta)|=1}^{\omega} \left[\frac{D^{\alpha,\beta}}{(\alpha\beta)!} \phi_l(0) \overbrace{\sum_{|\theta|=\omega-|(\alpha,\beta)|} \left[\prod_{q=1}^{n} \left[\sum_{\substack{u+v=\alpha_q+\beta_q+\theta_q \\ \alpha_q \le u \le \omega\alpha_q \\ \beta_q \le v \le \omega\beta_q}} \underbrace{\Big(\sum_{\substack{\sigma(a)=u \\ |a|=\alpha_q}} \frac{\alpha_q!}{a!} \prod_{k=1}^{\omega} g_{k,q}^{a_k} \Big)}_{G(\alpha_q,u)} \underbrace{\Big(\sum_{\substack{\sigma(b)=v \\ |b|=\beta_q}} \frac{\beta_q!}{b!} \prod_{k=1}^{\omega} h_{k,q}^{b_k} \Big)}_{H(\beta_q,v)} \right] \right]}^{\Omega(\alpha,\beta,\omega)} \right]$$

With the boundary conditions

- For $\omega = 1$

$$g_\omega(t_o) = x_o \qquad g_\omega(t_f) = x_f$$

- For $\omega > 1$

$$g_\omega(t_o) = 0 \qquad g_\omega(t_f) = 0$$

A solution to these ordinary differential equations was approximated expanding the exponencial function and at last the solution was found to be

- For $\omega = 1$

$$g_1(t) = \Omega_N^{1,1}(t)x_o + \Omega_N^{1,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Big(x_f - \Omega_N^{1,1}(t_f)x_o\Big)$$
$$h_1(t) = \Omega_N^{2,1}(t)x_o + \Omega_N^{2,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Big(x_f - \Omega_N^{1,1}(t_f)x_o\Big)$$

- For $\omega > 1$

$$g_\omega(t) = -\Omega_N^{1,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Theta_{N,\omega}^g(t_f,t_o) + \Theta_{N,\omega}^g(t,t_o)$$
$$h_\omega(t) = -\Omega_N^{2,2}(t)\Big(\Omega_N^{1,2}(t_f)\Big)^{-1}\Theta_{N,\omega}^g(t_f,t_o) + \Theta_{N,\omega}^h(t,t_o)$$

Where

$$\Theta_{N,\omega}^g(t,t_o) = \int_{t_o}^{t} \Big[\Omega_N^{1,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{1,2}(t,s)\chi_\omega^\phi(s) \Big] ds$$

$$\Theta_{N,\omega}^h(t,t_o) = \int_{t_o}^{t} \Big[\Omega_N^{2,1}(t,s)\chi_\omega^\psi(s) + \Omega_N^{2,2}(t,s)\chi_\omega^\phi(s) \Big] ds$$

and

$$\Phi_N(t,s) = \begin{bmatrix} \Omega_N^{1,1}(t,s) & \Omega_N^{1,2}(t,s) \\ \\ \Omega_N^{2,1}(t,s) & \Omega_N^{2,2}(t,s) \end{bmatrix} = \sum_{k=0}^{N} \frac{1}{k!}(t-s)^k A^k \approx I+(t-s)A+\frac{1}{2}(t-s)^2 A^2+\cdots+\frac{1}{N!}(t-s)^N A^N$$

Finally with the functions $g_i(t)$ and $h_i(t)$ the solution of the original OCP can be approximated by:

$$u_K(t) = -R^{-1}G^T\left(\sum_{\omega=1}^{K} g_\omega(t)\right)\left(\sum_{\omega=1}^{K} h_\omega(t)\right)$$

Which is the control law given by this technique and the one that will be used in the angular control of the quadrotor in section 4.3.3.

Chapter 4

Quadrotor Control

This chapter describes the proposed control architecture for the complete dynamic system presented in chapter 2 and each of the controllers. Initially, the control architecture is explained in section 4.1. Next, each of the parts of the control is further detailed.

First, in order to produce the best translational reference points a discrete optimal control problem is solved for each trajectory resulting in time varying reference values for translational position and velocity which are used in the overall online control, this off-line problem is presented in section 4.2.

Because of the decoupled nature of the dynamic system it is natural to divide the control in three cascade controllers: the translational control, the angular control and the motor control. Given that the overall control is a Nonlinear Model Predictive Control (NMPC) each of the three controllers is an optimal control. Only quadratic cost functions will be considered, since no motivation to consider other more complicated cost functions exists. The design of these controllers is presented in section 4.3. Finally the quaternion model-based attitude control is designed in section 4.4.

It is important to clarify the following distinction: whenever the *angular* control is mentioned, it makes reference to the *attitude* control using the angle-based attitude model. While the *quaternion* control refers to the *attitude* control using the quaternion-based model. Both controllers are interchangeable as long as the references are transformed accordingly. The main control architecture described in section 4.1 uses an angular control which later will be changed for the quaternion control described in section 4.4.

4.1 Control Architecture

As mentioned previously, the decoupling of the system dynamics into the attitude, translational and motor dynamics naturally leads to a division in the control. This is motivated by two reasons: First the complete system is not controllable which from the beginning is an issue. However the separated systems are controlable. Second, it is easier to design three small controls than a big one, also the physical insight for each separate dynamic system is clearer than for the complete model.

The control will consist on a single loop, first the translational control computes the optimal accelerations, these are then translated into angular references that enter the angular control. The output of the angular control are the open loop optimal torques and thrust that are feed into the motor controllers and in turn these return values for ω_i^d and α_i which enter the system and then the loop closes with the computation of the new optimal accelerations on the next sampling time. The three controllers (translational, angular and motor) are optimal controllers with fix final time, the final time of one control is the sample time of the previous control, this is done to guarantee the stability of the overall control.

With this in mind the control will be divided into four parts:

1. Translational Control: In order to design this control only the translational dynamics are considered. However with a slight modification: the states remain v_x, v_y, v_z, x, y and z while the inputs are changed to a_x, a_y and a_z, the accelerations in each axis. This is done because if the inputs are taken as ϕ, θ, ψ and F_t then there is no analytic solution to the optimal control problem. On the other hand with this approach an optimal solution can be found and the translation of the optimal inputs a_x^*, a_y^* and a_z^* can be addressed later guaranteeing that the optimality is preserved.

2. Angular References: As mentioned before the output of the translational control are optimal accelerations a_x^*, a_y^* and a_z^*. These need to be translated into optimal angle references ϕ^*, θ^* and ψ^* for the angular control, this problem is only mathematical manipulation and as a result optimal angular references are obtained.

3. Angular Control: Once the angular references are known it is necessary to address the corresponding tracking problem subject to the angular dynamics. This optimal control problem does not have analytic solution and is where the Extended Modal Series Method is applied. In this part the state variables are ϕ, θ, ψ, p, q and r, while the control inputs are the torques τ_ϕ, τ_θ and τ_ψ

4. Motor Control: Once the optimal torques τ_ϕ^*, τ_θ^* and τ_ψ^* and total force F_t^* are known it is up to the motor control to provide them. The values for τ_ϕ^*, τ_θ^*, τ_ψ^* and F_t^* determinate uniquely the desired forces of each motor F_i^*, hence references in the torques and total force translate into force references for each motor allowing a separation which assigns to each motor a different force reference F_i^*. In this part an explicit optimal control is designed for the motor-propeller system in order to achieve the required force F_i.

Figure 4.1 shows graphically the complete control of the drone together with the input and output variables of each block.

4.2 Reference Generation

Before performing the aggressive maneuvers it is necessary to generate the reference points for the quadrotor. These positions will be the references for the outer translational controller during the execution of the maneuver. The generation process must take into account the shape of the trajectory as well as the control limitations.

Since only translational position and velocity references are required for the outer control loop and the purpose of this control is only to generate references, the drone can be modeled

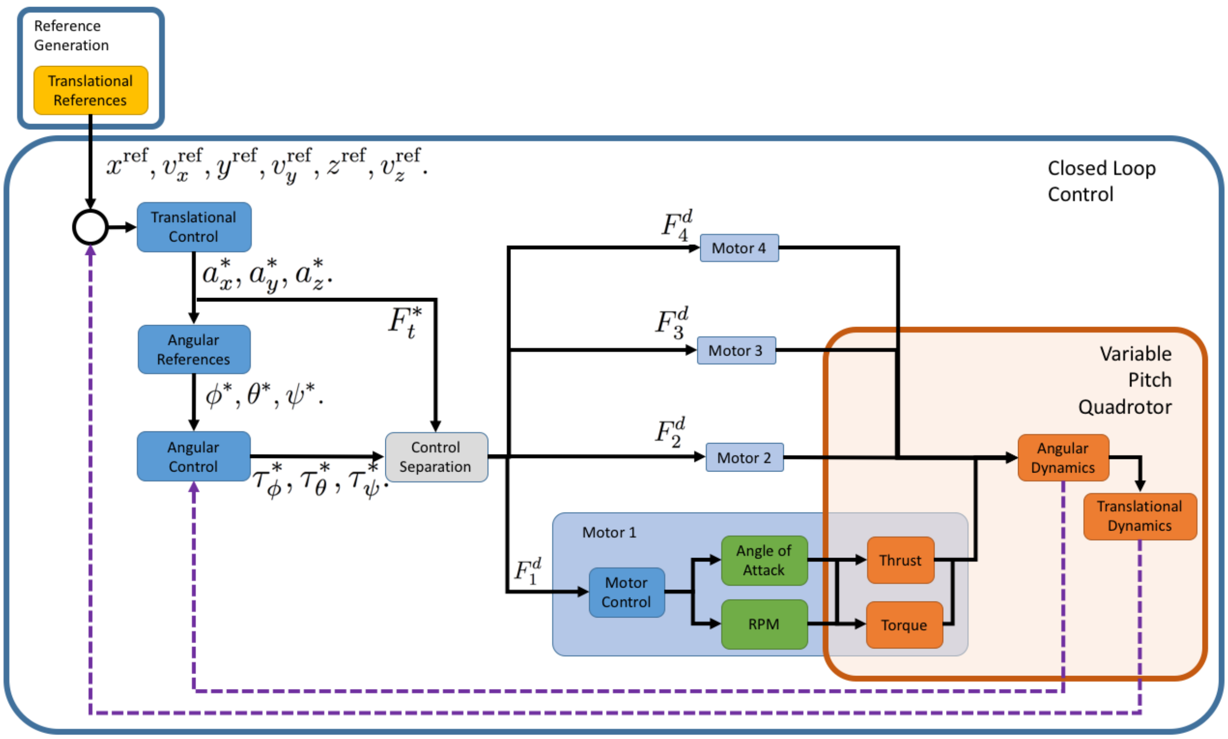

Figure 4.1: Control architecture. The yellow box is the off-line reference generator presented in section 4.2. The blue boxes correspond to control blocks explained in section 4.3 (only the control architecture of motor 1 is shown given that is the same for the others). The orange boxes correspond to dynamic systems whose models were presented in chapter 2. The green boxes represent the inputs of the overall system. The purple dotted lines are measurements of state variables.

as a particle in space with dynamics:

$$\dot{v}_x = a_x \quad \dot{x} = v_x \quad \dot{v}_y = a_y \quad \dot{y} = v_y \quad \dot{v}_z = a_z \quad \dot{z} = v_z$$

Because of the symmetry of the system, it is necessary to consider only one dimension, which in matrix form is written as:

$$\begin{bmatrix} \dot{v}_x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} a_x \tag{4.1}$$

Now, consider the quadratic optimal control problem with limitations on both the absolute control inputs (a_x) and the rate of change of the control inputs ($\dot{a}_x$):

$$\min_{a_x(t)} \left\{ J = \int_{t_o}^{t_f} (x - x^{\text{ref}})^2 Q + a_x^2 R + \left(\dot{a}_x\right)^2 D \right\} \tag{4.2}$$

$$\text{s.t.} \quad \begin{bmatrix} \dot{v}_x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} a_x \qquad \begin{matrix} a_{x_{\min}} \leq a_x \leq a_{x_{\max}} \\ \dot{a}_{x_{\min}} \leq \dot{a}_x \leq \dot{a}_{x_{\max}} \end{matrix}$$

The discrete form of this problem has a known analytical solution, and since the problem is solved offline it is possible to take a small sample time making it very accurate. The first step is to discretize the dynamic system with a sampling time of T_s. Then rewrite the cost function as a sum. The result is the discrete quadratic optimal control problem

$$\min_{a_x[k]}\Big\{J = \sum_{k=1}^{N}(x[k] - x^{\text{ref}}[k])^2Q + a_x^2[k]R + \Delta a_x^2[k]D\Big\} \tag{4.3}$$

$$\text{s.t.} \qquad \begin{bmatrix} v_x[k+1] \\ x[k+1] \end{bmatrix} = A_d \begin{bmatrix} v_x[k] \\ x[k] \end{bmatrix} + B_d a_x[k] \qquad \begin{array}{c} a_{x_{\min}} \leq a_x[k] \leq a_{x_{\max}} \\ \Delta a_{x_{\min}} \leq \Delta a_x[k] \leq \Delta a_{x_{\max}} \end{array}$$

Where $N = \frac{t_o - t_f}{Ts}$ and $a_x[k] = \Delta a_x[k] + a_x[k-1]$ and A_d and B_d are the discrete version of matrices A and B with sampling time T_s.

This optimal control problem can be expressed as a convex optimization problem (for details see appendix A.2), hence if it is feasible the solution is a global minimum. The solution of this optimization problem gives the optimal translational positions and velocities for the execution of the maneuver, including the constraints in the control. These will be the references for the closed loop translational control.

4.3 Closed Loop Control

In this section the four parts of the control architecture are further detailed and explained.

4.3.1 Translational Control

Consider the optimal control problem:

$$\min_{u(t)=(a_x(t),a_y(t),a_z(t))^T} \Big\{J(x,u) = \frac{1}{2}\int_{t_o}^{t_f} \big(x^TQx + u^TRu\big)dt\Big\}$$

Where $x(t) = (v_x(t), v_y(t), v_z(t), x(t), y(t), z(t))^T$ is the state vector. Subject to the modified translational dynamics:

$$\begin{array}{ll} \dot{v_x} = a_x & \dot{x} = v_x \\ \dot{v_y} = a_y & \dot{y} = v_y \\ \dot{v_z} = a_z & \dot{z} = v_z \end{array} \tag{4.4}$$

Since there is no motivation to consider cost for cross variables the matrices Q and R will be diagonal. Furthermore, given that the drone needs to move in space the cost associated with the translational position is zero, and with no preference for one axis over the other we have that:

$$Q = \text{diag}(q, q, q, 0, 0, 0) \qquad R = \text{diag}(r, r, r) \tag{4.5}$$

Given the symmetric nature of the resulting problem it is possible to consider only one dimension and the results will be the same for the other axis:

$$\min_{a_x(t)}\Big\{\frac{1}{2}\int_{t_o}^{t_f}\big(qx^2(t) + ra_x^2(t)\big)dt\Big\} \qquad \text{s.t.} \qquad \dot{v_x} = a_x \quad \dot{x} = v_x$$

Applying Pontryagin Maximum Principle gives the following system of ordinary differential equations:

$$\dot{v_x} = \tfrac{-1}{r}\lambda_1 \qquad \dot{\lambda_1} = -qv_x - \lambda_2$$
$$\dot{x} = v_x \qquad \dot{\lambda_2} = 0$$

Defining $k = \sqrt{q/r}$ the solution is:

$$\begin{aligned}
v_x(t) &= C_1 e^{kt} + C_2 e^{-kt} + \frac{1}{r}C_4 \\
x(t) &= \frac{C_1}{k}e^{kt} - \frac{C_2}{k}e^{-kt} + \frac{1}{r}C_4 t + C_3 \\
\lambda_1(t) &= -rk(C_1 e^{kt} - C_2 e^{-kt}) \\
\lambda_2(t) &= C_4
\end{aligned}$$

And the optimal control is:

$$a_x^*(t) = k(C_1 e^{kt} - C_2 e^{-kt})$$

If we set $x(t_o) = (v_{xo}, x_o)^T$ and $x(t_f) = (v_{xf}, x_f)^T$. The constants can be found solving the linear system:

$$\begin{pmatrix} e^{kt_o} & e^{-kt_o} & 0 & \frac{1}{r} \\ \frac{1}{k}e^{kt_o} & -\frac{1}{k}e^{-kt_o} & 1 & \frac{t_o}{r} \\ e^{kt_f} & e^{-kt_f} & 0 & \frac{1}{r} \\ \frac{1}{k}e^{kt_f} & -\frac{1}{k}e^{-kt_f} & 1 & \frac{t_f}{r} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix} = \begin{pmatrix} v_{xo} \\ x_o \\ v_{xf} \\ x_f \end{pmatrix}$$

The same procedure can be performed on the other axis and as a result the optimal controls $a_x^*(t)$, $a_y^*(t)$ and $a_z^*(t)$ will be known.

4.3.2 Angular References

Next, the problem of translating these optimal accelerations into angles and forces is addressed. Given that all the accelerations are optimal the superscript $*$ will be ignored. It is clear that the modified translational dynamics imply:

$$\begin{aligned}
a_x &= \frac{F_t}{m}(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\psi)) \\
a_y &= \frac{F_t}{m}(\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\psi)\cos(\psi)) \\
a_z &= \frac{F_t}{m}\cos(\phi)\cos(\theta) - g
\end{aligned}$$

In order simplify notation redefine $a_z = a_z + g$. It follows that

$$F_t = m\sqrt{a_x^2 + a_y^2 + a_z^2} \tag{4.6}$$

Now, define:

$$\begin{aligned}
\xi_x &= \frac{ma_x}{F_t} = \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\
\xi_y &= \frac{ma_y}{F_t} = \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\
\xi_z &= \frac{ma_z}{F_t} = \cos(\phi)\cos(\theta)
\end{aligned}$$

After solving:

$$\phi = \sin^{-1}(\xi_x \sin(\psi) - \xi_y \cos(\psi)) \qquad \theta = \tan^{-1}\left(\frac{1}{\xi_z}\sqrt{\xi_x^2 \cos^2(\psi) + \xi_y^2 \sin^2(\psi)}\right)$$

These equations enable the transformation of optimal accelerations into optimal angles which are the inputs of the angular control.

4.3.3 Angular Control

The optimal angular control problem is:

$$\min_{u(t)} \left\{ J(x,u) = \frac{1}{2}\int_{t_o}^{t_f} \left(x^T Q x + u^T R u\right) dt \right\}$$

Where $u(t) = (\tau_\phi(t), \tau_\theta(t), \tau_\psi(t))^T$ and $x(t) = (p(t), q(t), r(t), \phi(t), \theta(t), \psi(t))^T$ Subject to the angular dynamics (2.5), (2.6), (2.7), (2.8), (2.9) and (2.10). This problem does not have analytic solution, but noting that the model is affine with respect to the input, it is possible to apply the Modified Extended Modal Series Method to solve the tracking problem. The first step is to write the system in an affine form:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \underbrace{\begin{pmatrix} \left(\frac{I_y - I_z}{I_x}\right) qr \\ \left(\frac{I_z - I_x}{I_y}\right) pr \\ \left(\frac{I_x - I_y}{I_z}\right) qp \\ p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ q\cos\phi - r\sin\phi \\ \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \end{pmatrix}}_{F(x)} + \underbrace{\begin{pmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{G(x)} \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \tag{4.7}$$

Then the solution of the optimal control problem is given by Pontryagin Maximum Principle equations whose solution can be approximated using the Modified Extended Modal Series Method described in chapter 3.

4.3.4 Motor Control

From the angular control the optimal torques are calculated: τ_ϕ^*, τ_θ^* and τ_ψ^* and from (4.6) the desired total force is known: F_t^d.Now, using equations (2.1), (2.2), (2.3) and (2.4) these inputs can be translated into desired forces for each motor F_i^d:

$$\begin{pmatrix} F_1^d \\ F_2^d \\ F_3^d \\ F_4^d \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} \\ -\frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} \\ 1 & -1 & 1 & -1 \end{pmatrix}^{-1} \begin{pmatrix} F_t^d \\ \tau_\phi^d \\ \tau_\theta^d \\ \tau_\psi^d \end{pmatrix} \tag{4.8}$$

With this decoupling it is possible to solve only one problem for each motor. At a given instant let ω_i be the actual angular velocity, α_i the actual angle of attack, F_i the actual produced force and F_i^d the desired force. Since the change in α_i is assumed instantaneous while the change of ω_i requires time, the angle of attack is changed to the value that minimizes the error $\Delta F_i = |F_i^d - F_i|$, given that the angle of attack is bounded. It is likely that this value will be the maximum or the minimum and $\Delta F_i \neq 0$.

At the same time it is possible to calculate the required value of ω_i that would produce the desired force F_i^d, with an angle of attack of $\bar{\alpha}_i$, let ω_i^d be this value.

Now, the problem is to achieve the desired angular velocity ω_i^d in a fix amount of time t_f (given by the sample time of the control). This can we rewritten as an optimal control problem:

$$\min_u \left\{ \int_{t_o}^{t_f} (r_u u^2) dt \right\} \qquad \text{s.t.} \quad \dot{\omega}_i = K_{m_i}(u - \omega_i) \quad \omega_i(t_o) = \omega_i \quad \omega_i(t_f) = \omega_i^d$$

This problem can be solved analytically. Using Pontryagin's Maximum Principle the optimal corresponding differential equations are

$$\dot{\omega}_i^* = -K_{m_i}\omega_i - -\frac{\lambda K_{m_i}^2}{2r_U} \qquad \dot{\lambda}^* = \lambda K_{m_i}$$

Those solution is:

$$\omega_i = \frac{K_{m_i}}{4r_u}(C_2 e^{K_{m_i}t} - C_1 e^{K_{m_i}t}) \qquad \lambda = C_1 e^{K_{m_i}t}$$

Using the boundary conditions the constants can be calculated as:

$$C_1 = \frac{4r_u(\omega_i e^{-K_{m_i}t_f} - \omega_i^d e^{-K_{m_i}t_o})}{K_{m_i}(e^{K_{m_i}(t_f - t_o)} - e^{K_{m_i}(t_f - t_o)})}$$

$$C_2 = \frac{4r_u(\omega_i e^{K_{m_i}t_f} - \omega_i^d e^{K_{m_i}t_o})}{K_{m_i}(e^{K_{m_i}(t_f - t_o)} - e^{-K_{m_i}}(t_f - t_o))}$$

And finally the optimal control can be found to be:

$$\omega_i^d = u^* = -\frac{C_1 K_{m_i}}{2r_U} e^{K_{m_i}t}$$

Now, as the angular velocity of the motor increases the resulting force will increase as well. Since ω_i^d was calculated with a value for the angle of attack of $\bar{\alpha}_i$ which is lower than the real value of α_i (that is assumed to be saturated) the produced force will be greater than predicted.

As a consequence, after some time the real force F_i will match the desired force F_i^d for a value of $\omega_i < \omega_i^d$. From that point α_i will tend to $\bar{\alpha}$ in such a way that the produced force will be kept equal to F_i^d. Finally when $t = t_o$ the angular velocity will be ω_i^d and the produced force will be F_i^d with an angle of attack of $\bar{\alpha}$.

4.4 Quaternion Control

As shown in section 2.2 quaternions are only an alternative form of representing the attitude state of the system and thus are equivalent to an Euler's angle representation in the sense that it is possible to translate from one system to another (for details see appendix A.3). As a result any *quaternion controller* can be regarded as an attitude controller based on Euler's angles which is translated to a quaternion system.

The main contribution of the quaternions is to provide an attitude representation of the dynamic system which does not suffer from singularities and does not involve trigonometric functions. Thus the contribution is in the representation of the dynamic system and not the controller synthesis. The quaternion control raises from the fact that it is unpractical to model the dynamic system using quaternions and then transforming back to Euler's angles to compute the control signal (this transformation defeats the purpose of the quaternion representation).

Taking that into account, there are two main differences when using the quaternion representation to design a controller:

1. Since the differential equations involve only products of the attitude state variables p, q, r, q_0, q_1, q_2 and q_3 it is a second order Taylor multivariable expansion. Unlike the Euler's angle representation that yields an infinite series. As a consequence the use of approximation techniques such as the EMSM may produce more accurate results given that the vector function $F(x)$ does not need to be approximated.

2. The quaternion system representation is not controllable given that the quaternion needs to be unitary. Then, it is not possible to reach every point in the state space. This can be proven differentiating the norm $|q|^2$ with respect to time and replacing equations 2.21.

Given that in the quaternion representation the system is not controllable pole placement control methods such as state feedback are not possible. Hence two alternatives appear: use PID controllers in cascade combination or more advanced techniques such as Backstepping or Sliding Mode Controllers. In this case three PID controllers will be used.

First the angular references are translated into quaternion references. Note that the optimal accelerations that come from the translational controller could directly produce quaternion references solving the system of nonlinear equations:

$$
\begin{gathered}
a_x = 2\sqrt{a_x^2 + a_y^2 + a_z^2}(q_1 q_3 + q_0 q_2) \\
a_y = 2\sqrt{a_x^2 + a_y^2 + a_z^2}(q_2 q_3 - q_0 q_1) \\
a_z = \sqrt{a_x^2 + a_y^2 + a_z^2}(q_0^2 - q_1^2 - q_2^2 - q_3^2) - g \\
q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1
\end{gathered}
\tag{4.9}
$$

For q_0, q_1, q_2 and q_3 .

Let $q^d = q_0^d + q_1^d\hat{\imath} + q_2^d\hat{\jmath} + q_3^d\hat{k}$ denote the reference quaternion. The error quaternion q^e cannot be computed as the difference between q and q^d. Instead it is computed as:

$$q^e = q^* q^d = (q_0 - q_1\hat{\imath} - q_2\hat{\jmath} - q_3\hat{k})(q_0^d + q_1^d\hat{\imath} + q_2^d\hat{\jmath} + q_3^d\hat{k}) \tag{4.10}$$

If $q^e = q_0^e + q_1^e\hat{\imath} + q_2^e\hat{\jmath} + q_3^e\hat{k}$ then

$$\begin{aligned} q_0^e &= q_0 q_0^d + q_1 q_1^d + q_2 q_2^d + q_3 q_3^d \\ q_1^e &= q_0 q_1^d - q_1 q_0^d - q_2 q_3^d + q_3 q_2^d \\ q_2^e &= q_0 q_2^d + q_1 q_3^d - q_2 q_0^d - q_3 q_1^d \\ q_3^e &= q_0 q_3^d - q_1 q_2^d + q_2 q_1^d - q_3 q_0^d \end{aligned} \tag{4.11}$$

Note that if $q = q^d$ then $q^e = 1 + 0\hat{\imath} + 0\hat{\jmath} + 0\hat{k}$.

In order to relate the error quaternion components with the corresponding torques consider the hover attitude $\phi = \theta = \psi = 0$ which translates to the desired quaternion $q^d = 1 + 0\hat{\imath} + 0\hat{\jmath} + 0\hat{k}$. Now the error quaternion has components:

$$q_0^e = q_0 \quad q_1^e = -q_1 \quad q_2^e = -q_2 \quad q_3^e = -q_3$$

Next consider three cases:

1. $q = \alpha_0 + \alpha_1\hat{\imath}$ in this case the attitude of the quadrotor is a rotation along the x axis, hence a roll torque τ_ϕ is required to recover the hover state.

2. $q = \alpha_0 + \alpha_2\hat{\jmath}$ in this case the attitude of the quadrotor is a rotation along the y axis, hence a pitch torque τ_θ is required to recover the hover state.

3. $q = \alpha_0 + \alpha_3\hat{k}$ in this case the attitude of the quadrotor is a rotation along the z axis, hence a yaw torque τ_ψ is required to recover the hover state.

In each case the direction of the torque is given by the sign of α_1, α_2 and α_3 respectively. The direct relation between the error quaternion components q_1^e, q_2^e and q_3^e and the torques τ_ϕ, τ_θ and τ_ψ suggest three SISO parallel controllers for three separate angular dynamics:

1. For ϕ:
$$\dot{p} \approx \frac{\tau_\phi}{I_x} \qquad \dot{q}_1 = \frac{1}{2}p \qquad \Longleftrightarrow \qquad H_p(s) = \frac{1}{2I_x s^2} \tag{4.12}$$

2. For θ:
$$\dot{q} \approx \frac{\tau_\theta}{I_y} \qquad \dot{q}_2 = \frac{1}{2}q \qquad \Longleftrightarrow \qquad H_q(s) = \frac{1}{2I_y s^2} \tag{4.13}$$

3. For ψ:
$$\dot{r} \approx \frac{\tau_\psi}{I_z} \qquad \dot{q}_3 = \frac{1}{2}r \qquad \Longleftrightarrow \qquad H_r(s) = \frac{1}{2I_z s^2} \tag{4.14}$$

The three transfer functions (4.12), (4.13), (4.14) were obtained assuming that the products pq, qr, pr are negligible and near hover attitude. However, the analysis can be extended to any attitude because of the rotational invariance of the complete attitude dynamics. Each transfer function has two poles in the origin and so a P controller will make the system marginally stable. Thus PID controllers are required. The tuning is done using the transfer functions 4.12, 4.13 and 4.14: Consider the usual PID controller:

$$C(s) = \frac{k_i + sk_p + s^2 k_d}{s} \tag{4.15}$$

And the transfer function

$$H(s) = \frac{1}{2Is^2} \tag{4.16}$$

If p_1, p_2 and p_3 are the desired poles then after a straight forward computation

$$\begin{aligned} k_d &= -2I(p_1 + p_2 + p_3) \\ k_p &= 2I(p_1p_2 + p_1p_3 + p_2p_3) \\ k_i &= -2ip_1p_2p_3 \end{aligned} \tag{4.17}$$

Recalling that the quaternion control refers to the attitude control using the quaternion-based model it is necessary to clarify how it integrates in the overall control architecture of figure 4.1. First of all the input references ϕ^*, θ^*, ψ^* are translated into the reference quaternion q^d.

Now, unlike the angular control which is an optimal open loop control, the PID quaternion based control is intrinsically closed loop requiring the measurement of the state variables to produce a control signal. Given that these measurements are not available for every sample time the attitude model of the quadrotor is used instead. The result is shown in figure 4.2.

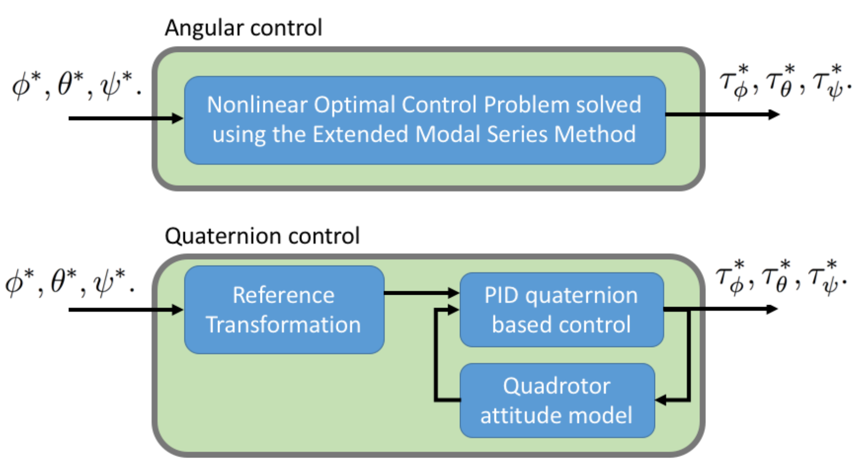

Figure 4.2: Angular and Quaternion controllers.

To summarize, given a desired trajectory an off-line discrete optimal control problem with sample time T_s is solved in order to compute the translational position and velocity references regarding the system as a particle in space. These references are then feed to the translational optimal controller which is has a closed form control law with horizon of prediction $T_{h,t}$ and sample time $T_{s,t} = T_s$.

The output of the translational controller are optimal translational accelerations $a_x^*[k]$, $a_y^*[k]$ and $a_z^*[k]$, the first of which is translated into angular references ϕ^*, θ^* and ψ^* that are the input to the optimal angular controller. This control has horizon of prediction $T_{h,a} = T_{s,t}$ and sample time $T_{s,a}$.

The output of the angular controller are optimal torques $\tau_\phi^*[k]$, $\tau_\theta^*[k]$ and $\tau_\psi^*[k]$, which together with the optimal thrust $F_T^*[k]$ (calculated from $a_x^*[k]$, $a_y^*[k]$ and $a_z^*[k]$) give the desired forces and torques for each motor $F_i^*[k]$, $\tau_i^*[k]$. Next, first one of these forces becomes the reference of each motor control, which is also an optimal controller with prediction horizon $T_{h,c} = T_{s,a}$ and sample time $T_{s,c}$.

At last, the output of these controllers are the inputs of the model, namely, the optimal angular velocities ω_i^* and angles of attack α_i^*. The loop closes when after a time $T_{s,t}$ the translational state variables are used to solve the new optimal translational control problem and the process is repeated.

It is important to note that this type of architecture implies that the angular and motor control are open loop controllers and the only time when the sensed state variables are used is then the translational loop closes. Meaning that if $T_{h,a} = n_a T_{s,a}$, then only in the initial time the measured angular positions will be used in the OCP, for the remaining $n_a - 1$ computations of the control signal the *predicted* angular position is used.

The time diagram of this process is shown in figure 4.3

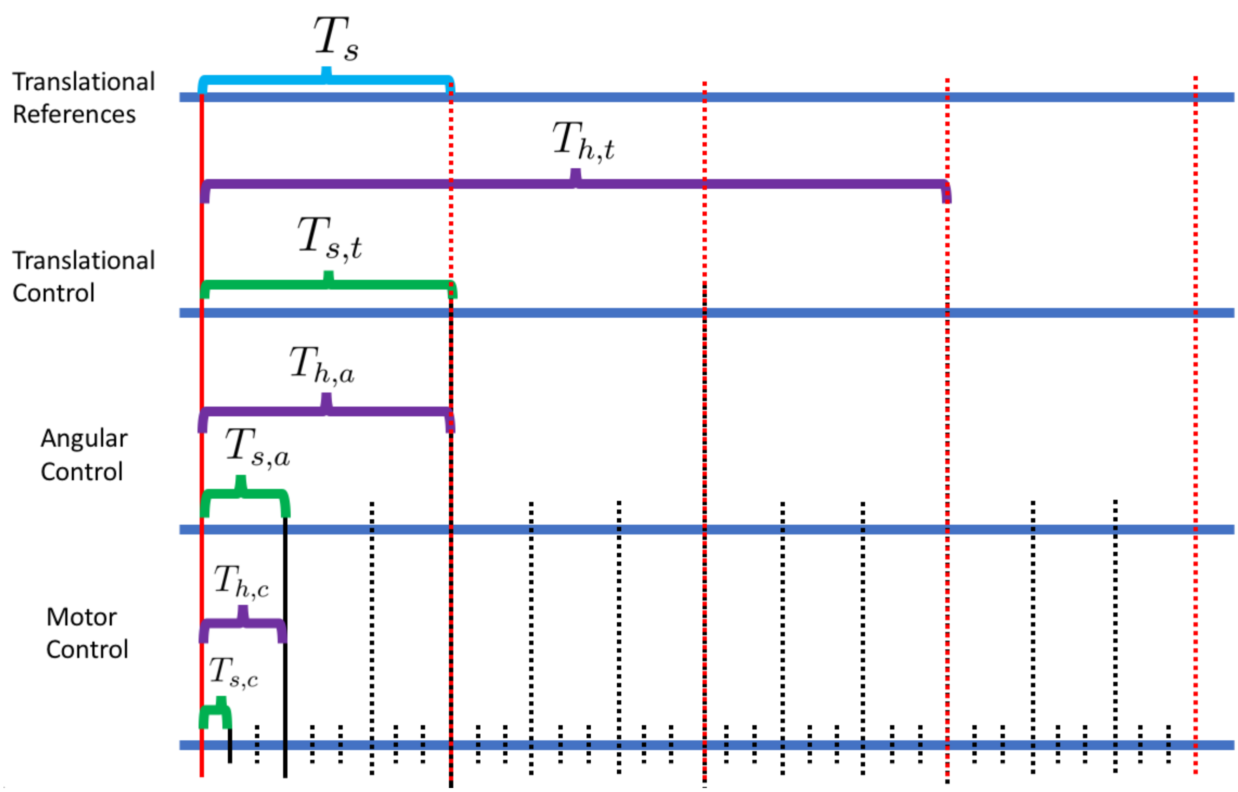

Figure 4.3: Time diagram of the control architecture. The red vertical lines are the times when the loop closes and the measured state variables are available. The prediction horizons are in purple, while the sample times are in green. Note that $T_{s,t} = T_s$, $T_{h,a} = T_{s,t}$ and $T_{h,c} = T_{s,c}$

Chapter 5

Results

This chapter presents the simulation results of the complete control system applied to the variable pitch quadrotor model. The main objective is to test the proposed Nonlinear Model Predictive Control based on the Extended Modal Series Method, this is done in two ways: first the performance of the EMSM controller is analyzed independently and then comparing to the Quaternion based control. In both instances two criteria are used: the Parametric Uncertainty and External Disturbance Robustness.

All simulations were carried out in Simulink with fixed step size and ODE solver Runge-Kutta, the parametric values are shown in appendix A.4. In section 5.1 the trajectories used during the simulations are detailed and explained.

The main results are presented in the following sections. Section 5.2 explains the processing of the simulation output data and analysis methodology. Finally sections 5.3, 5.4 and 5.5 present the complete results with the corresponding analysis. The last section addresses conclusions and future work.

5.1 Trajectories

The trajectories used to test the algorithm are all rectilinear, i.e. formed by concatenation of linear segments. This is done because the most common way to generate trajectories is through the use of waypoints, which are reference positions in space connected by linear displacements.

The time of each linear segment is the same (1 second). Thus, they are only defined by the end points of each segment (given that the starting point is the end of the previous segment). Three trajectories with increasing aggressiveness corresponding to different shapes are considered (figures 5.1, 5.2 and 5.3). In order to further test the controllers each trajectory has three variations changing the length the linear segment ($L_1 = 1$[m], $L_2 = 1.5$[m] and $L_3 = 2$[m]). All the trajectories start at the origin and the first segment corresponds to the take-off.

The three trajectories are:

1. First trajectory: it is a movement in the xy plane keeping the altitude constant. The diamond shape in the plane guarantees a change of velocity in only one direction at the time alternating between the x and y direction. This is a change of direction of

180 degrees. The complete trajectory in space and the corresponding components are shown in figure 5.1 (for line segment length L_3).

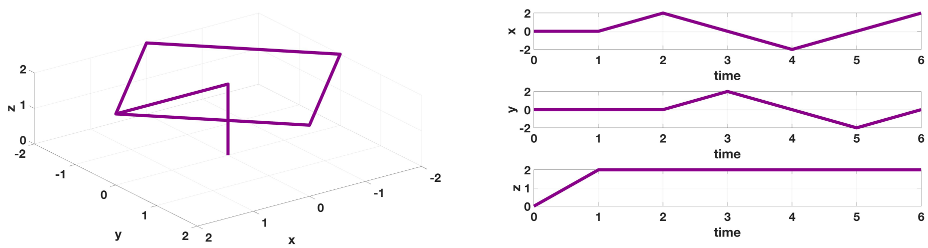

Figure 5.1: First trajectory with segment length L_3

2. Second trajectory: This second trajectory includes non-aggressive changes in altitude. The y trajectory is very similar to the first case while the x trajectory has a more aggressive turn at the beginning. As before the complete trajectory and in each component is shown in figure 5.2 for segment length L_3.

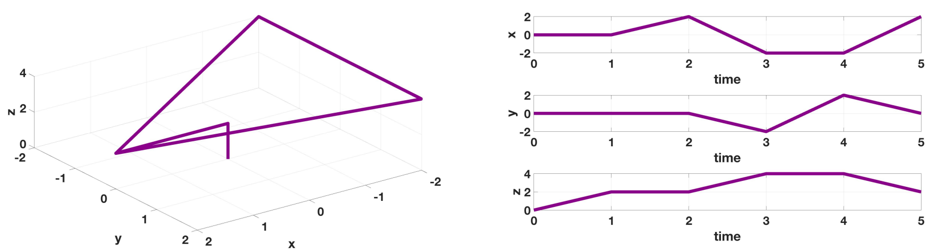

Figure 5.2: Second trajectory with segment length L_3

3. Third trajectory: This third and final trajectory shown in figure 5.3 exhibits a very aggressive change in altitude. The movement in the y axis is the less aggressive of the three with pauses between changes of direction, while the movement in the x axis is more aggressive with two strong changes of direction, that require a higher acceleration and deceleration.

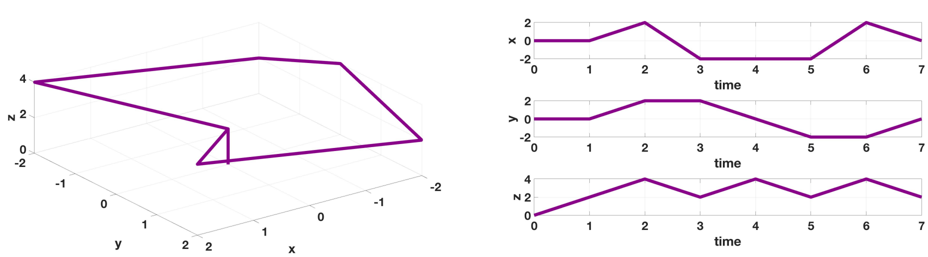

Figure 5.3: Third trajectory with segment length L_3

5.2 Presentation of the Results

The Extended Modal Series Method controller was tested using two criteria:

- *Parameter Uncertainty*: This approach aims to unveil the effect that a change in the values of a set of parameters in the model has in the controller performance. The selected parameters for parameter variation were the moments of inertia I_x, I_y, I_z and the mass M. The percentage of variation was 80%, 90%, 110% and 120% for all the parameters. Seeking to appreciate the effect that every parameter change had on the control performance, all the parameters were changed one by one. As a result for a fix trajectory and segment length this criteria required 17 simulations, 4 for each parameter corresponding to the percentage variations and the reference with nominal values.

- *External Disturbances*: The external disturbances are perturbations on the model represented by offsets in the state variables. Although the quadrotor model has 12 state variables the most common and significant perturbations occur in the v_x and v_y states and emulate side wind currents. These perturbations have three parameters: magnitude, duration and direction. The chosen values for these parameters are:

  - Magnitude: $M_1 = 0.5$[m/s] and $M_2 = 1$[m/s].
  - Duration: 30% (from 40% to 70% of the total time interval), and 60% (from 20% to 80% of the total time interval).
  - Direction: 0 [rad], $\pi/4$ [rad], $\pi/2$ [rad], $3\pi/4$ [rad], π [rad], $5\pi/4$ [rad], $3\pi/2$ [rad], $7\pi/4$ [rad].

  Hence, for a given trajectory and segment length 33 simulations were required, 32 corresponding to the product of possible magnitudes, durations and directions and the reference one without any perturbation.

Two types of overall results were investigated:

1. *Performance of the EMSM controller*: In first instance the robustness of the EMSM controller is tested in face of parameter uncertainty and external disturbances. The first criteria explores the critical model parameters by comparing the performance while varying the values in the simulation model for each of the selected parameters. The second criteria presents the general response of the control to perturbations characterized by different magnitudes, duration and orientation. With the purpose of evaluating the performance of the EMSM the results of all the simulations were normalized with respect to the reference results (the ones obtained without any parameter variation and external disturbance), in the tables these results are labeled as *EMSM Reference Normalized*.

2. *Comparison with the Quaternion controller*: Given that the most common control used in the variable pitch quadrotor is a Quaternion-based one, it is natural to compare it

with the proposed controller. Once more these results are divided into parameter uncertainty and external disturbance criteria. With the objective of comparing the performance of the Quadrotor-based controller all the simulation results were normalized with respect to the EMSM reference results (the ones obtained without any parameter variation and external disturbance). This is done for the parameter uncertainty and external disturbance perturbations, in the tables these results are labeled as *Q/E Normalized*.

The variables used to measure the performance of the control were:

- Square of the norm 2 of the state variables error, denoted $||\delta p||$, $||\delta q||$, $||\delta r||$, $||\delta \phi||$, $||\delta \theta||$, $||\delta \psi||$, $||\delta v_x||$, $||\delta v_y||$, $||\delta v_z||$, $||\delta x||$, $||\delta y||$ and $||\delta z||$.
- Square of the norm 2 of the average energy $||u||$, is the average signal energy over the four control signals.

This set of variables is called *Performance Variables* and to avoid long table captions they appear in all the result tables. For a single trajectory and a given segment length the behavior of the quadrotor during the simulations may be affected by the shape of the trajectory. For instance, if during the velocity disturbance the quadrotor is moving in the direction of the disturbance the position error may be higher than if the quadrotor was moving in a direction perpendicular to the disturbance. These magnifications of the errors produced by the trajectory itself are unavoidable. Thus, in order to obtain trajectory independent conclusions before analysis the results of each trajectory T_1, T_2 and T_3 with segment length L_1 were averaged. Taking into account the above description 4 tables were used for the analysis:

1. Table 5.3 shows the *EMSM Reference Normalized* parametric uncertainty results averaged over the three trajectories with segment length L_1.
2. Table 5.4 shows the *Q/E Normalized* parametric uncertainty results averaged over the three trajectories with segment length L_1.
3. Table 5.5 shows the *EMSM Reference Normalized* external disturbance results averaged over the three trajectories with segment length L_1.
4. Table 5.6 shows the *Q/E Normalized* external disturbance results averaged over the three trajectories with segment length L_1.

In the four tables, the final two rows show the mean and standard deviation of each measured variable (each column). These values represent the global behavior of the variable across every case in both the parametric uncertainty and external disturbance simulations for reference normalization and Q/E normalization.

The reason why tables 5.3, 5.4, 5.5 and 5.6 are not averaged over all the segment lengths is because for L_2 and L_3 the Quaternion controller is unable to control the quadrotor for some cases, and thus the comparison is meaningless. However, it is still possible to average the performance variables in the cases that can be controlled for every trajectory and segment length, the corresponding results are arrange in the following tables:

1. Table 5.7 shows the *EMSM Reference Normalized* parametric uncertainty results averaged over all the performance variables for each trajectory and segment length.

2. Table 5.8 shows the *Q/E Normalized* parametric uncertainty results averaged over all the performance variables for each trajectory and segment length.

3. Table 5.9 shows the *EMSM Reference Normalized* external disturbance results averaged over all the performance variables for each trajectory and segment length.

4. Table 5.10 shows the *Q/E Normalized* external disturbance results averaged over all the performance variables for each trajectory and segment length.

As stated before the values in these tables give a global indication of the measured variables behavior across all the cases (the ones that could be controlled). Although the results of tables 5.7 , 5.8, 5.9 and 5.10 are not averaged by trajectory or segment length in a sense they are independent of the trajectory because the particular effect that the trajectory may have on a single case is averaged over the remaining cases.

Since the tables 5.7, 5.8, 5.9 and 5.10 are all normalized with respect to the nominal EMSM controller performance it is necessary to specifically present this performance, which is shown in table 5.1. The first three rows present the EMSM controller performance for each trajectory corresponding to segment length L_1, while the last row contains the mean which are the values used to normalize the remaining results.

Tra	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
Und.	rad/s			rad			m/s			m			$\times 10^{11}$
T_1	0.0492	0.0419	0.0326	0.0014	0.0011	0.0002	0.0607	0.0675	0.0183	0.0043	0.0034	0.0007	501106
T_2	0.0764	0.0931	0.4262	0.0013	0.0013	0.0054	0.1689	0.1516	0.0385	0.0053	0.0050	0.0018	113945
T_3	0.0380	0.0973	1.3478	0.0013	0.0011	0.0109	0.1516	0.0379	0.1221	0.0055	0.0026	0.0044	281598
Mean	0.0545	0.0774	0.6022	0.0013	0.0012	0.0055	0.1271	0.0856	0.0596	0.0051	0.0037	0.0023	148551

Table 5.1: Nominal Results

Table 5.2 shows the color coding used to analyze the table results.

Color	min	max	Meaning
	-	0.25	4 times better
	0.25	1.1	1.1 to 4 times better
	0.9	1.1	almost the same
	2	4	4 to 1.1 times worse
	4	-	4 times worse
	NaN		

Table 5.2: Color Coding

For the *EMSM Reference Normalized* tables, cells with color and indicate that the results of the reference were better than the corresponding case, while cells with color and indicate that the current simulation results were better than the reference ones. For the *Q/E Normalized* tables, cells with color and indicate that the Quaternion results were better than EMSM ones, while cells with color and indicate that the EMSM results were better than the Quaternion.

5.3 Parametric Uncertainty Robustness

This section analyzes the parametric uncertainty robustness of the proposed controller.

5.3.1 EMSM Controller

Par.	%	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta\phi\|\|$	$\|\|\delta\theta\|\|$	$\|\|\delta\psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
ref	0	1	1	1	1	1	1	1	1	1	1	1	1	1
I_x	0.6	21.0	1.35	4.78	9.64	0.95	3.67	1.02	1.18	0.95	1.18	1.30	0.88	2.61
	0.8	0.77	1.15	2.57	1.01	1.16	2.25	1.01	0.93	0.96	1.13	0.81	0.92	0.91
	1.2	1.48	1.20	0.93	1.47	1.32	0.85	0.95	0.80	1.01	0.88	0.93	1.06	0.87
	1.4	2.00	1.03	0.98	1.28	1.09	0.76	1.04	0.96	1.08	1.10	0.88	1.20	0.95
I_y	0.6	1.03	8.97	2.39	1.28	5.36	1.87	1.04	0.97	0.98	0.99	0.97	0.94	1.62
	0.8	0.93	1.22	0.62	0.92	1.01	0.61	1.06	0.98	1.04	1.01	1.20	0.94	0.80
	1.2	0.83	1.53	0.57	0.99	1.28	0.49	1.02	0.95	1.01	1.02	0.96	0.90	0.75
	1.4	0.89	2.08	0.46	0.98	1.65	0.40	1.00	0.90	0.94	1.23	1.00	0.87	0.75
I_z	0.6	1.00	1.05	0.81	1.03	1.04	0.43	1.00	0.84	0.98	1.04	0.77	0.93	0.67
	0.8	0.86	1.09	0.59	1.08	1.21	0.57	1.04	0.91	0.94	0.91	0.93	0.84	0.80
	1.2	0.89	1.18	1.24	0.99	1.00	1.18	1.04	0.95	0.97	1.07	1.06	0.78	1.21
	1.4	0.92	1.17	0.74	1.24	1.09	0.70	1.04	0.92	1.02	0.99	0.68	1.05	0.98
M	0.8	1.09	1.59	1.16	1.15	1.12	0.90	1.44	1.29	1.24	1.54	1.36	7.64	0.98
	0.9	1.11	1.19	2.90	1.05	1.12	2.43	1.21	1.21	1.05	1.25	1.14	3.17	1.35
	1.1	0.86	0.94	0.47	1.01	1.14	0.50	0.96	0.85	0.94	0.95	0.80	1.49	0.70
	1.2	0.75	0.89	0.13	0.95	1.15	0.21	0.76	0.71	0.96	0.55	0.66	6.21	0.59
Mean		2.20	1.68	1.31	1.59	1.39	1.11	1.04	0.96	1.00	1.05	0.97	1.81	1.03
STD		4.86	1.89	1.18	2.07	1.03	0.92	0.13	0.14	0.07	0.20	0.19	2.01	0.48

Table 5.3: EMSM Reference Normalized Parametric Uncertainty

At first glance table 5.3 shows that the main differences appear in the angular states rather than the translational states except for in the variation of mass. These results are coherent, since the moments of inertia belong to the attitude dynamics while the mass is located in the translational dynamic model. The effect of the mass on the attitude states can be explained by the fact that the attitude loop is contained inside the translational loop.

The complete performance of the angular rate of change state variables translates almost exactly to the angular states. This is expected given that in rough terms the second variables are an integration of the first ones. Another expected result is the direct effect of mass variation in the z state variable

Besides this preliminar analysis there are no other clear patterns in the table. For almost every case some performance variables exhibit a better result while others present the opposite behavior. Thus, no clear critical parameter is revealed. The mean values for each performance variable indicate that the most sensible state variables are the p and q angular rates of change, the ϕ and θ angles and the height z.

Looking at the average energy of the control signals many cases present a decrease with respect to the reference, but there is no clear trend and also in average over all the cases the used energy is the same.

The results in table 5.3 indicate that the EMSM controller is very robust with respect to parametric uncertainty with particular expected results such as the direct effect of the mass uncertainty in the height error and the global effect of the moments of inertia changes in the attitude state variables.

5.3.2 EMSM and Quaternion Controllers

Par.	%	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta\phi\|\|$	$\|\|\delta\theta\|\|$	$\|\|\delta\psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
ref	0	7.21	7.80	1.86	2.66	2.83	1.21	1.03	1.13	0.97	0.91	0.98	0.79	1.35
I_x	0.6	7.41	4.88	0.93	1.21	1.89	0.69	1.05	1.03	1.06	1.18	1.12	0.94	0.88
	0.8	24.6	7.66	4.52	5.51	3.13	3.39	1.07	1.10	0.95	1.04	0.97	1.02	2.85
	1.2	13.9	6.05	4.34	4.98	3.29	3.09	1.16	1.06	1.03	1.09	0.90	1.02	2.84
	1.4	74.6	19.2	7.50	10.5	6.93	6.89	1.05	1.43	1.17	0.98	0.80	1.37	7.04
I_y	0.6	5.50	14.0	1.09	2.06	1.33	0.72	0.95	1.10	1.07	1.00	1.03	0.97	0.96
	0.8	7.09	10.9	4.83	3.78	2.74	2.57	1.04	1.05	0.96	1.25	0.96	1.14	3.12
	1.2	16.9	9.76	3.85	5.99	3.43	3.09	1.25	1.09	1.04	1.28	0.96	0.95	2.66
	1.4	12.3	15.8	3.80	3.99	6.21	2.78	1.10	1.21	0.99	1.08	1.15	0.97	2.54
I_z	0.6	–	–	–	–	–	–	–	–	–	–	–	–	–
	0.8	1082	469.	1086	138.	61.4	770.	1.83	2.55	5.08	0.92	1.16	29.3	137.
	1.2	6.04	5.59	1.49	2.47	2.21	1.30	1.11	1.10	0.95	1.16	0.99	0.90	1.44
	1.4	8.46	7.15	2.70	2.86	2.67	2.01	1.06	1.14	1.00	1.15	1.29	1.10	1.93
M	0.8	227.	167.	422.	27.6	16.2	442.	2.22	1.63	2.47	1.61	1.50	7.36	71.1
	0.9	49.5	24.9	4.99	10.2	6.24	4.26	1.34	1.43	1.19	1.38	0.93	2.69	5.09
	1.1	4.35	4.26	0.99	1.69	1.90	0.78	0.91	0.92	0.95	0.65	0.69	2.47	0.95
	1.2	5.01	3.38	0.62	1.79	1.55	0.47	0.83	0.85	0.98	0.70	0.73	7.24	0.82
Mean		97.1	48.7	97.0	14.0	7.76	77.8	1.19	1.24	1.37	1.09	1.01	3.77	15.2
STD		268.	119.	283.	33.6	14.7	214.	0.35	0.40	1.05	0.23	0.20	7.15	36.8

Table 5.4: Q/E Normalized Parametric Uncertainty

Table 5.4 compares the performance of the EMSM controller with the Quaternion controller for parametric uncertainty. Three trends are highly noticeable:

1. Almost for every case the EMSM is better than the Quaternion control with respect to the attitude states. This is a very important result since it shows that the overall performance of the proposed controller is better than the benchmark, with respect to parametric uncertainty and also in the reference case.

   It is important to keep in mind that this is a normalized result, meaning that it is not absolute in the sense that very low values do not directly indicate that the EMSM is a very good controller. What low values mean, is that the EMSM control is better than the Quaternion-based one, but it could happen that for certain cases the performance of the Quaternion control is very poor while the EMSM remains the same.

2. The empty row corresponding to the 80% variation of the I_z moment of inertia. This happened because in that particular case the Quaternion control was unable to control the system and as a result the comparison was meaningless. A side effect of this is

presented in the following row. The extremely low values for the 90% variation of the I_z moment of inertia most likely indicate that in this case the Quaternion control provided very poor performance (almost unable to control the system).

3. Compared with the attitude states, the translational states show little difference. This is a consequence of the fact that both the EMSM and Quaternion controllers are in the attitude loop, while the translational control is the same. Hence this means that despite the difference in the performance of the attitude controllers, the translational control is able to compensate for them and provide similar results in both cases. Of course, this is a global analysis and there are small particular differences for given cases and performance variables.

Several conclusions can be drawn from this analysis: first that the EMSM is much more robust than the Quaternion control with respect to parameter uncertainty, second that the overall performance of the EMSM is much better than the Quaternion control and finally that despite the differences the translational control is able to compensate for them and provide similar results for both controllers.

5.4 External Disturbance Robustness

This section analyzes the external disturbance robustness of the proposed controller.

5.4.1 EMSM Controller

Table 5.5 shows the EMSM Reference Normalized results with respect to external disturbances. Several clear patterns can be noted:

1. There is a clear global increase in the error of the v_x, v_y, x and y state variables. This is a direct consequence of the wind perturbations.

2. As for the attitude state variables, q shows a minor error increase but not significant. On the other hand the attitude state variables r and ψ exhibit a significant improvement. This effect is more difficult to explain, however a possible explanation is that the continuous perturbations produce a constant attitude correction which not only compensates for the external disturbances but as a side effect improves the r and ψ errors.

3. In most cases the average energy is less than the reference case. However not very significantly.

Beside these patterns the remaining performance variables remain the same with very low standard deviation, meaning that in general the EMSM controller is very robust with respect to external disturbances.

mag	T	ori	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
ref	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
M_1	30%	0	0.91	1.18	1.57	1.03	0.96	1.29	1.47	0.94	0.98	2.38	1.10	0.91	0.90
		$\pi/4$	1.11	1.21	1.42	1.03	1.20	1.26	1.15	1.52	1.06	1.66	2.06	1.05	0.96
		$\pi/2$	0.91	1.18	0.60	1.03	1.18	0.56	1.02	1.96	0.93	1.10	2.84	0.88	0.77
		$3\pi/4$	0.87	1.09	2.02	0.95	1.14	1.82	1.49	1.48	0.96	2.34	1.66	0.69	0.96
		π	0.90	1.04	0.60	0.96	1.06	0.56	1.98	0.94	0.98	3.48	0.96	0.81	0.71
		$5\pi/4$	0.89	1.07	0.30	0.93	1.02	0.36	1.44	1.68	0.95	2.02	3.18	1.02	0.61
		$3\pi/2$	0.87	1.01	0.51	0.86	1.13	0.47	0.95	2.28	1.02	0.85	5.01	0.89	0.69
		$7\pi/4$	0.89	1.08	0.42	1.01	0.92	0.40	1.19	1.54	1.02	1.49	2.90	0.99	0.68
	60%	0	0.93	1.04	0.66	1.09	1.15	0.49	2.53	0.96	0.99	4.99	1.17	0.94	0.80
		$\pi/4$	0.93	1.23	0.60	0.95	1.06	0.59	1.75	1.97	0.98	3.09	3.40	0.85	0.72
		$\pi/2$	1.04	1.24	0.66	1.02	1.25	0.65	1.05	3.42	1.03	1.09	6.46	0.90	0.91
		$3\pi/4$	1.04	1.14	0.96	1.17	1.05	0.76	1.71	2.14	0.93	3.21	3.56	0.79	0.88
		π	0.78	1.27	0.44	1.05	1.09	0.40	2.47	0.91	0.94	5.16	0.87	0.88	0.64
		$5\pi/4$	0.98	1.40	1.22	1.03	1.29	0.91	1.67	2.31	0.99	3.07	4.78	1.10	1.00
		$3\pi/2$	1.10	1.14	1.28	1.07	1.18	0.98	1.04	3.26	1.01	0.97	7.45	0.90	0.92
		$7\pi/4$	0.94	1.12	0.45	1.07	1.16	0.67	1.71	2.19	0.97	2.91	4.47	0.86	0.73
M_2	30%	0	1.01	1.35	0.71	1.40	1.05	0.77	3.21	0.93	1.02	7.41	0.89	1.10	0.93
		$\pi/4$	0.92	1.16	0.47	0.96	1.21	0.54	2.04	3.07	0.91	3.96	5.25	0.80	0.76
		$\pi/2$	0.97	1.15	0.50	1.00	1.23	0.40	1.04	5.21	0.98	1.11	10.1	1.07	0.67
		$3\pi/4$	0.96	1.18	0.45	0.90	1.06	0.42	2.50	2.89	0.93	5.34	4.99	0.86	0.77
		π	0.91	1.17	0.36	1.07	0.97	0.32	4.00	0.87	1.04	9.13	0.90	0.85	0.67
		$5\pi/4$	0.99	1.19	1.04	1.06	1.15	0.91	2.69	3.55	1.09	5.49	8.19	1.25	0.97
		$3\pi/2$	1.13	1.25	1.05	1.12	1.23	0.84	1.02	6.01	0.86	1.07	14.5	0.76	0.92
		$7\pi/4$	1.06	1.17	0.78	0.94	1.08	0.60	2.02	3.42	0.94	3.95	7.68	0.78	0.88
	60%	0	1.03	1.36	0.85	1.13	1.08	0.80	7.01	0.98	0.97	17.1	1.10	0.86	1.06
		$\pi/4$	1.06	1.18	1.11	1.18	0.99	0.82	4.11	5.53	1.02	9.06	11.0	0.88	0.84
		$\pi/2$	1.45	1.14	0.86	1.26	1.16	0.90	1.02	10.5	0.96	0.94	22.8	0.95	1.01
		$3\pi/4$	1.20	1.34	1.65	1.02	1.05	1.32	4.06	6.14	0.96	9.97	12.1	1.01	1.24
		π	0.93	1.30	0.86	1.14	1.22	0.83	7.18	0.99	0.98	18.4	0.98	0.90	0.96
		$5\pi/4$	1.08	1.20	0.46	1.03	1.12	0.38	3.94	6.17	1.05	9.31	14.7	0.81	0.72
		$3\pi/2$	1.28	0.99	0.77	1.13	1.09	0.60	1.07	10.8	0.97	1.22	26.5	0.77	0.91
		$7\pi/4$	1.13	1.27	0.81	1.15	1.21	0.66	3.86	5.79	0.96	8.59	14.0	0.86	0.83
Mean			1.01	1.18	0.83	1.05	1.11	0.74	2.31	3.13	0.98	4.63	6.33	0.91	0.85
STD			0.13	0.10	0.41	0.10	0.09	0.33	1.60	2.60	0.04	4.41	6.40	0.11	0.14

Table 5.5: EMSM Reference Normalized External Disturbance

5.4.2 EMSM and Quaternion Controllers

In table 5.6 the Q/E normalized external disturbance response is presented. This table shares many similarities with table 5.4 and the same analysis can be applied, namely:

1. Globally the attitude states present a lower average error with the EMSM controller than with the Quaternion based one.

2. In this instance two cases cannot be presented because the Quaternion control was unable to control the system (magnitude M_1, duration 30%, orientation π and magnitude M_2, duration 60%, orientation π).

mag	T	ori	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
ref	0	0	7.21	7.80	1.86	2.66	2.83	1.21	1.03	1.13	0.97	0.91	0.98	0.79	1.35
M_1	30%	0	5.84	7.18	1.64	2.55	3.07	1.11	1.51	1.18	1.11	2.24	1.19	1.09	1.22
		$\pi/4$	11.4	6.46	2.83	3.27	2.51	1.95	1.26	1.58	1.08	1.47	1.70	1.35	1.96
		$\pi/2$	5.79	4.37	1.21	2.03	1.93	0.82	1.05	2.12	0.98	1.04	2.95	0.89	0.95
		$3\pi/4$	10.1	7.18	2.64	2.70	2.61	1.78	1.69	1.67	1.03	2.69	2.08	1.04	1.76
		π	–	–	–	–	–	–	–	–	–	–	–	–	–
		$5\pi/4$	214.	190.	842.	53.0	29.6	438.	2.14	2.21	2.74	2.42	2.64	28.2	88.9
		$3\pi/2$	141.	110.	573.	30.4	23.1	279.	1.58	2.38	1.98	1.17	3.98	12.5	61.4
		$7\pi/4$	6.13	5.91	1.11	2.27	1.66	0.94	1.42	1.67	0.98	1.90	2.85	0.97	1.12
	60%	0	11.8	8.72	2.12	3.02	2.87	1.41	2.51	1.16	1.01	4.60	0.99	0.94	1.39
		$\pi/4$	8.53	6.50	2.26	2.39	2.10	1.24	1.71	2.29	1.02	2.74	3.32	1.02	1.43
		$\pi/2$	9.90	6.33	3.43	3.75	2.98	2.23	1.12	3.58	1.10	1.05	6.33	1.23	2.05
		$3\pi/4$	4.45	5.01	3.30	1.88	2.01	2.52	1.91	2.20	1.05	3.55	3.28	1.21	1.28
		π	11.2	9.39	2.96	4.09	3.01	1.81	2.52	1.10	1.10	5.50	0.89	1.03	1.82
		$5\pi/4$	62.6	20.7	8.39	11.9	8.90	6.48	1.92	2.36	0.96	3.48	3.91	1.37	9.36
		$3\pi/2$	8.10	4.74	0.84	2.42	2.06	0.68	1.07	3.59	1.08	1.18	7.60	1.07	1.01
		$7\pi/4$	7.10	5.98	1.70	1.99	2.16	1.32	1.78	2.33	1.05	2.73	4.42	0.93	1.27
M_2	30%	0	7.32	8.56	2.52	2.38	2.43	1.74	3.43	1.17	1.03	7.29	1.06	0.97	1.35
		$\pi/4$	48.6	13.1	5.08	8.36	5.31	4.55	2.38	3.21	1.06	4.45	5.28	1.15	3.82
		$\pi/2$	429.	106.	28.7	86.7	26.8	14.6	2.05	10.4	1.98	2.02	14.4	21.1	17.3
		$3\pi/4$	9.15	7.36	8.59	3.76	3.22	5.50	2.88	3.29	1.04	5.90	5.62	1.19	3.45
		π	50.7	75.9	263.	13.9	8.20	200.	4.41	1.11	1.57	9.74	1.10	3.50	26.9
		$5\pi/4$	7.94	6.50	3.76	3.05	2.49	2.64	2.77	3.68	1.08	5.48	7.94	0.94	1.68
		$3\pi/2$	11.3	9.76	2.28	3.23	3.19	1.43	1.05	5.86	0.96	0.90	13.8	0.83	1.49
		$7\pi/4$	6.35	5.92	2.70	1.86	2.02	2.06	2.16	3.30	0.97	3.99	7.43	0.92	1.46
	60%	0	7.09	6.94	4.72	2.31	2.61	2.88	7.01	1.10	1.04	16.8	1.17	1.08	2.49
		$\pi/4$	9.23	6.99	1.29	2.09	2.65	1.11	3.91	5.68	0.96	8.54	11.1	0.70	1.28
		$\pi/2$	14.2	6.09	4.47	3.91	3.25	3.39	1.03	10.7	1.04	1.00	23.1	0.90	2.74
		$3\pi/4$	156.	56.9	15.5	35.2	14.0	12.0	4.14	6.67	1.20	9.62	11.1	2.77	16.1
		π	–	–	–	–	–	–	–	–	–	–	–	–	–
		$5\pi/4$	8.77	7.08	2.79	2.56	2.32	1.60	4.00	5.95	1.01	9.59	13.7	0.81	1.50
		$3\pi/2$	9.42	4.04	1.46	2.97	1.71	1.12	1.00	10.4	0.99	1.15	24.7	0.85	1.23
		$7\pi/4$	76.5	26.3	8.37	15.4	7.57	6.10	4.38	6.35	1.05	9.30	13.6	1.66	5.62
Mean			44.5	24.3	58.3	10.2	5.85	32.4	2.35	3.60	1.17	4.34	6.60	3.07	8.61
STD			87.6	41.9	182.	18.3	7.42	96.3	1.37	2.85	0.38	3.74	6.34	6.22	19.0

Table 5.6: Q/E Normalized External Disturbance

3. In general the translational states present the same response, indicating that the translational controller is able to compensate for the differences of the attitude controllers in presence of disturbances.

4. On average the mean control energy is also lower for the EMSM than for the Quaternion control.

Also, as explained in table 5.4 these values are Q/E normalized, meaning that a low number does not imply a very good performance of the EMSM controller, but that compared with the Quaternion based one, is better. Overall the main conclusion that can be derived from tables 5.5 and 5.6 is that the EMSM control is very robust with respect to external disturbances and also that across all the attitude state variables has better performance than the Quaternion based control.

5.5 Trajectory Comparison Performance

Tables 5.7, 5.8, 5.9 and 5.10 show the mean of the performance variables for all the trajectories and segment lengths. The empty rows correspond to instances where any case could be controlled and thus no data is available.

Tra	Leng	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
T_1	L_1	1.61	1.75	2.51	1.17	1.28	2.06	1.08	0.97	0.92	0.78	0.85	2.57	1.23
	L_2	1.49	1.42	1.36	1.55	1.56	1.24	0.98	1.00	0.93	0.91	0.99	1.55	1.15
	L_3	1.48	1.16	1.78	1.47	0.98	2.53	1.00	0.95	1.05	0.88	0.96	2.31	1.37
T_2	L_1	1.65	2.02	1.03	1.72	1.63	0.82	0.98	1.01	1.10	1.21	1.17	1.68	1.11
	L_2	1.56	1.38	1.13	1.06	1.23	1.04	0.98	0.99	0.98	1.17	0.91	1.93	1.18
	L_3	1.15	0.41	0.71	0.77	0.66	1.19	123.	300.	4355	11.5	37.3	1362	120.
T_3	L_1	3.36	1.29	0.40	1.88	1.28	0.45	1.05	0.90	0.99	1.15	0.89	1.19	0.75
	L_2	1.44	1.78	1.42	1.17	1.21	1.21	1.02	1.27	1.01	0.92	1.94	1.26	1.51
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–

Table 5.7: EMSM Reference Normalized Parametric Uncertainty Trajectory Comparison

Tra	Leng	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
T_1	L_1	124.	85.3	269.	15.3	9.35	220.	1.42	1.21	1.76	0.79	0.88	6.68	32.0
	L_2	24.2	21.7	56.7	8.82	7.07	37.6	1.10	1.10	1.04	0.87	1.00	2.31	12.0
	L_3	9.73	6.08	5.51	4.22	2.19	6.62	1.09	1.07	1.06	0.87	1.03	2.28	3.36
T_2	L_1	39.9	34.1	16.4	13.6	7.12	7.93	1.01	1.21	1.26	1.25	1.20	3.07	6.90
	L_2	270.	199.	8.68	17.3	9.19	11.8	1.05	1.67	2.86	1.08	0.95	22.9	6.82
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–
T_3	L_1	126.	26.5	5.70	13.3	6.80	5.66	1.14	1.29	1.09	1.21	0.95	1.55	6.66
	L_2	44.6	108.	3.94	11.4	11.5	2.99	1.11	2.12	1.60	1.16	1.27	5.13	2.87
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–

Table 5.8: Q/E Normalized Parametric Uncertainty Trajectory Comparison

In every table the last row is empty meaning that both the EMSM and Quaternion controllers were unable to control the system for trajectory T_3 and segment length L_3, this is due to the high level of aggressiveness that this particular configuration implies. However tables 5.8 and 5.10 also present an empty row for the trajectory T_2 and segment length L_3, the fact that this row is not empty in tables 5.7 and 5.9 means that the EMSM controller was able to control the system in this trajectory while the Quaternion control failed. Besides that three more observations can be made from these tables:

1. Tables 5.8 and 5.10 show that across all the trajectories and segment lengths the EMSM controller is much better than the Quaternion one with respect to the attitude states. Also that the translational control is able to compensate for these differences resulting in very similar translational variables behavior.

Tra	Leng	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
T_1	L_1	0.99	1.27	1.34	0.90	1.09	1.17	3.22	2.94	0.87	4.86	6.21	0.85	1.05
	L_2	0.92	1.22	1.47	1.09	1.36	1.31	1.93	1.93	0.88	3.31	3.90	0.65	1.13
	L_3	1.10	0.82	1.45	1.10	0.82	2.08	1.58	1.51	1.02	2.18	2.65	1.12	1.26
T_2	L_1	0.96	1.20	0.92	1.19	1.12	0.77	1.80	1.89	1.09	4.73	4.73	1.01	0.95
	L_2	1.14	1.16	1.03	0.97	1.10	1.02	1.34	1.43	0.96	2.64	2.20	1.39	1.17
	L_3	1.13	0.46	0.65	0.81	0.73	1.08	1.32	1.58	3954	1.72	1.05	1678	1.14
T_3	L_1	1.07	1.07	0.24	1.06	1.13	0.26	1.93	4.56	0.98	4.30	8.06	0.86	0.54
	L_2	1.41	1.39	1.40	1.12	1.11	1.45	1.51	8.10	1.05	2.34	8.94	1.01	1.62
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–

Table 5.9: Reference Normalized External Disturbance Trajectory Comparison

Tra	Leng	$\|\|\delta p\|\|$	$\|\|\delta q\|\|$	$\|\|\delta r\|\|$	$\|\|\delta \phi\|\|$	$\|\|\delta \theta\|\|$	$\|\|\delta \psi\|\|$	$\|\|\delta v_x\|\|$	$\|\|\delta v_y\|\|$	$\|\|\delta v_z\|\|$	$\|\|\delta x\|\|$	$\|\|\delta y\|\|$	$\|\|\delta z\|\|$	$\|\|u\|\|$
T_1	L_1	38.0	36.4	154.	9.32	6.41	85.2	3.52	3.15	1.19	4.96	6.13	4.72	16.8
	L_2	13.7	13.9	38.7	6.30	5.45	24.1	2.01	2.05	0.97	3.17	3.95	1.10	8.83
	L_3	7.53	4.69	3.02	3.12	1.89	4.12	1.63	1.61	1.03	2.09	2.60	1.13	2.49
T_2	L_1	8.68	9.88	4.53	3.70	3.42	2.53	1.77	2.10	1.19	4.69	4.79	1.16	2.67
	L_2	115.	71.7	3.69	8.24	5.95	5.77	1.17	2.01	1.57	2.86	2.07	4.28	2.93
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–
T_3	L_1	84.5	24.8	5.86	17.3	7.43	4.07	2.07	5.36	1.11	4.20	8.28	3.08	5.30
	L_2	59.6	284.	3.68	14.1	13.5	3.58	1.86	4.26	1.97	3.31	8.17	5.42	2.57
	L_3	–	–	–	–	–	–	–	–	–	–	–	–	–

Table 5.10: Q/E Normalized External Disturbance Trajectory Comparison

2. The translational variables are more robust to parametric uncertainty than the attitude variables. Nevertheless, there is not a single state variable particularly affected by changes in parametric values.

3. Across all the trajectories and segment lengths the external disturbances translate directly into errors in the v_x, v_y, x and y state variables.

5.6 Conclusions and Future Work

In this thesis a Nonlinear Model Predictive Controller for the a Variable Pitch Quadrotor was designed and tested in simulation. Given that the dynamical system is input affine it has non-linearities that make the PDEs resulting from Pontryagin's Maximum Principle impossible to solve analytically.

In order to overcome this problem, a new method for solving this kind of system of equations was used: the Extended Modal Series Method. It relies on the approximation of the optimal solution by a sum of functions which are themselves solutions to systems of ODEs.

In the original publication, these systems of ODEs were defined implicitly, slowing the implementation of the controller. The first and main contribution of this work was to find a closed form not only for the systems of ODE but also for the control law which enables faster implementation and also further theoretical development.

On the other hand, the Variable Pitch Quadrotor is a modification of the traditional quadrotor, where the angle of attack of the blades in the propellers can change, enabling almost instantaneous changes in thust and torque and also negative thrust generation.

That modification makes the Variable Pitch Quadrotor an ideal system to execute aggressive maneuvers. Given the nature of the task, the system continuously operates in nonlinear regimes of the dynamic system, which make traditional linealization control techniques not feasible.

In order to control the Variable Pitch Quadrotor while it performed the aggressive maneuvers a EMSM-based NMPC was used. The use of an NMPC in this application is the second contribution of this work, together with the complete Variable Pitch Quadrotor model, including relevant drag effects.

In order to test the proposed EMSM-based NMPC, the simulation results were analyzed independently and also compared with a Quaternion-based control. Two criteria were used for the analysis: Parameter Uncertainty and External Perturbation Robustness. Every analysis showed the same general results: the EMSM-based control is more robust to parameter uncertainty and external perturbations than the quaternion-based control.

As a general conclusion it can be stated that the EMSM-based NMPC is a very powerful control technique in the sense that it has all the advantages of a traditional NMPC, but at the same time overcomes the usual disadvantages of computation times by employing the EMSM to approximate the solution of the OCP. The closed form of the control law only increases the implementation speed enabling the application of this control to many other dynamic systems.

Other conclusions that can be drawn from this work are:

- Given that the EMSM control is an optimal controller it produced better performance than the Quaternion-based controller. This was expected, being the EMSM an optimal controller which took into account the system nonlinearities and constraints.

- The overall design of the control architecture divided into cascade controllers corresponding to each of the natural subdivisions of the complete dynamic system not only facilitates the design of each controller but also gives a physical insight that a single controller would otherwise not have.

- The quaternion-based attitude model presents some very interesting advantages over the traditional Euler angles model. In particular the fact that the model is a multivariable polynomial reduces the complexity of the associated PMP partial differential equations, possible enabling more accurate solutions.

- The Variable Pitch Quadrotor is a very versatile dynamic system that not only has the advantages of a traditional quadrotor such as vertical landing and take-off and independent movement in the three spatial axis but also is capable of agile maneuvers thanks to the possibility of almost instantaneous thrust and torque change and also generation of negative thrust.

Finally, throughout the thesis some questions remained unanswered while other appear. Given that they were not in the scope of the research no solution was given to them and are now proposed for future work:

- Continue with the theoretical development of the Extended Modal Series Method control technique. The closed form allows for many questions to be addressed, for instance bounds for the approximation errors, convergence and stability analysis.

- Generalization of the Extended Modal Series Method for more general cost functions and the effects on the overall method.

- Further development in the computational aspects of the implementation, for instance writing faster algorithms or looking for shortcuts in the code implementation of the algorithms in order to make the control signal computation quicker.

- Implementation in simulation in other input affine dynamic systems.

- Experimental validation of the control technique, not necessarily in the Variable Pitch Quadrotor but any fast nonlinear dynamic system.

- Explore in more detail the possibilities of solving the PMP system of partial differential equations associated with the Quaternion-based attitude model.

- Compare the EMSM-based NMPC with other advance control techniques such as Sliding Modes and Backstepping. In order to see whether the advantages presented when compared with the Quaternion control still hold true.

A.1 Taylor Polynomial

Taylor's Theorem gives a formula for the Taylor polynomial:

$$f(a+h) \approx \sum_{k=0}^{N} \frac{1}{k!}(D^k f(a)(h^k)) \tag{A.1}$$

There are, however multiple terms of each sum that give the same result. All that matters is the number of times a index i_j occurs

Denote by α_i the number of times the index i appears for $1 \leq i \leq n$, then it follows that if $\alpha = (\alpha_1, ...\alpha_n)$ then:

1. All the terms of k-th order satisfy that:

$$|\alpha| = \sum_{i=1}^{n} \alpha_i = k \tag{A.2}$$

2. All the terms with equal α give the same result.
3. $h_{i_1} h_{i_2} \dots h_{i_k} = h_1^{\alpha_1} h_2^{\alpha_2} \dots h_n^{\alpha_n}$
4. $\partial_{i_1} \partial_{i_2} \dots \partial_{i_k} = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_n^{\alpha_n}$

Using this notation Taylor's formula becomes [32]:

Theorem 1. *Assume U to be a convex open subset of $\mathbb{R}^n$ and let $f \in C^{k+1}(U, \mathbb{R}^p)$, for $k \in \mathbb{N}_0$. Then we have, for all a and $a+h \in U$*

$$f(a+h) = \sum_{|\alpha| \leq k} \frac{h^\alpha}{\alpha!} D^\alpha f(a) + (k+1) \sum_{|\alpha|=k+1} \frac{h^\alpha}{\alpha!} \int_0^1 (1-t)^k D^\alpha f(a+th)dt \tag{A.3}$$

Where $\alpha! = \alpha_1! \alpha_2! \dots \alpha_n!$, $h^\alpha = h_1^{\alpha_1} h_2^{\alpha_2} \dots h_n^{\alpha_n}$ and $D^\alpha = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_n^{\alpha_n}$

Proof: see [32]

So the k-th order term of the Taylor Polynomial is:

$$\sum_{|\alpha|=k} \frac{h^\alpha}{\alpha!} D^\alpha f(a) = \sum_{|\alpha|=k} \frac{h_1^{\alpha_1} h_2^{\alpha_2} \dots h_n^{\alpha_n}}{\alpha_1! \alpha_2! \dots \alpha_n!} \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_n^{\alpha_n} f(a) \tag{A.4}$$

A.2 Discrete Quadratic Optimal Control Problem

Consider the following discrete quadratic optimal control problem

$$
\begin{aligned}
&\min_{\Delta u_k}\left\{ J=\sum_{k=1}^{N}\big((y_k-y_k^d)^TQ(y_k-y_k^d)+u_k^TRu_k\big)+\sum_{k=0}^{N-1}\Delta u_k^T D\Delta u_k\right\}\\
&\quad\text{s.t.}\\
&\quad x_{k+1}=A_kx_k+B_ku_k \qquad u_{k_{\min}}\le u_k\le u_{k_{\max}}\\
&\quad y_k=C_kx_k \qquad y_{k_{\min}}\le y_k\le y_{k_{\max}}\\
&\quad u_{k+1}=u_k+\Delta u_k \qquad \Delta u_{k_{\min}}\le \Delta u_k\le \Delta u_{k_{\max}}
\end{aligned}
$$

The first step is to transform the inputs into state variables, this is done with the following transformations:

$$
\begin{aligned}
\underbrace{\begin{bmatrix} x_{k+1}\\ u_{k+1}\end{bmatrix}}_{\tilde{x}_{k+1}} &= \underbrace{\begin{bmatrix} A_k & B_k\\ 0 & I\end{bmatrix}}_{\tilde{A}_k}\underbrace{\begin{bmatrix} x_k\\ u_k\end{bmatrix}}_{\tilde{x}_k}+\underbrace{\begin{bmatrix} 0\\ I\end{bmatrix}}_{\tilde{B}_k}\Delta u_k\\
y_k &= \underbrace{\begin{bmatrix} C_k & 0\end{bmatrix}}_{\tilde{C}_k}\underbrace{\begin{bmatrix} x_k\\ u_k\end{bmatrix}}_{\tilde{x}_k}
\end{aligned}
$$

The new state variables transform in the cost function as follows:

$$
\begin{aligned}
(y_k-y_k^d)^TQ(y_k-y_k^d)+u_k^TRu_k &= \begin{bmatrix} y_k^T-y_k^{dT} & u_k^T\end{bmatrix}\begin{bmatrix} Q & 0\\ 0 & R\end{bmatrix}\begin{bmatrix} y_k-y_k^d\\ u_k\end{bmatrix}\\
&= \left(\begin{bmatrix} x_k^TC_k^T & u_k^T\end{bmatrix}-\begin{bmatrix} y_k^{dT} & 0\end{bmatrix}\right)\begin{bmatrix} Q & 0\\ 0 & R\end{bmatrix}\left(\begin{bmatrix} x_k^TC_k\\ u_k\end{bmatrix}-\begin{bmatrix} y_k^{dT}\\ 0\end{bmatrix}\right)\\
&= \left(\underbrace{\begin{bmatrix} x_k^T & u_k^T\end{bmatrix}}_{\tilde{x}_k}\underbrace{\begin{bmatrix} C_k^T & 0\\ 0 & I\end{bmatrix}}_{\Gamma_k^T}-\underbrace{\begin{bmatrix} y_k^{dT} & 0\end{bmatrix}}_{\tilde{y}_k^T}\right)\underbrace{\begin{bmatrix} Q & 0\\ 0 & R\end{bmatrix}}_{\tilde{Q}}\left(\underbrace{\begin{bmatrix} G_k & 0\\ 0 & I\end{bmatrix}}_{\Gamma_k}\underbrace{\begin{bmatrix} x_k\\ u_k\end{bmatrix}}_{\tilde{x}_k}-\underbrace{\begin{bmatrix} y_k^{dT}\\ 0\end{bmatrix}}_{\tilde{y}_k}\right)\\
&= (\Gamma_k\tilde{x}_k-\tilde{y}_k)^T\tilde{Q}(\Gamma_k\tilde{x}_k-\tilde{y}_k)
\end{aligned}
$$

Hence the final cost function is:

$$
J=\sum_{k=1}^{N}(\Gamma_k\tilde{x}_k-\tilde{y}_k)^T\tilde{Q}(\Gamma_k\tilde{x}_k-\tilde{y}_k)+\sum_{k=0}^{N-1}\Delta u_k^TD\Delta u_k \tag{A.5}
$$

This can be written in matrix form, expand the first sum to obtain

$$\sum_{k=1}^{N}(\Gamma_k\tilde{x}_k-\tilde{y}_k)^T\tilde{Q}(\Gamma_k\tilde{x}_k-\tilde{y}_k)=\sum_{k=1}^{N}\tilde{x}_k^T\Gamma_k^T\tilde{Q}\Gamma_k\tilde{x}_k-2\tilde{x}_k^T\Gamma_k^T\tilde{Q}\tilde{y}_k^d+\tilde{y}_k^{d^T}\tilde{Q}\tilde{y}_k^d$$

$$=\tilde{x}_1^T\Gamma_1^T\tilde{Q}\Gamma_1\tilde{x}_1-2\tilde{x}_1^T\Gamma_1^T\tilde{Q}\tilde{y}_1^d+\tilde{y}_1^{d^T}\tilde{Q}\tilde{y}_1^d+\tilde{x}_2^T\Gamma_2^T\tilde{Q}\Gamma_2\tilde{x}_2-2\tilde{x}_2^T\Gamma_2^T\tilde{Q}\tilde{y}_2^d+$$
$$+\tilde{y}_2^{d^T}\tilde{Q}\tilde{y}_2^d+\cdots+\tilde{x}_N^T\Gamma_N^T\tilde{Q}\Gamma_N\tilde{x}_N-2\tilde{x}_N^T\Gamma_N^T\tilde{Q}\tilde{y}_N^d+\tilde{y}_N^{d^T}\tilde{Q}\tilde{y}_N^d$$

$$=\underbrace{\begin{bmatrix}\tilde{x}_1^T & \tilde{x}_2^T & \dots & \tilde{x}_N^T\end{bmatrix}}_{\tilde{x}^T}\underbrace{\begin{bmatrix}\Gamma_1^T\tilde{Q}\Gamma_1 & 0 & \dots & 0\\ 0 & \Gamma_2^T\tilde{Q}\Gamma_2 & \dots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \dots & \Gamma_N^T\tilde{Q}\Gamma_N\end{bmatrix}}_{\widehat{\Gamma}^T\widehat{\tilde{Q}}\widehat{\Gamma}}\underbrace{\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\\ \dots\\ \tilde{x}_N\end{bmatrix}}_{\tilde{x}}$$

$$-2\underbrace{\begin{bmatrix}\tilde{x}_1^T & \tilde{x}_2^T & \dots & \tilde{x}_N^T\end{bmatrix}}_{\tilde{x}^T}\underbrace{\begin{bmatrix}\Gamma_1^T\tilde{Q} & 0 & \dots & 0\\ 0 & \Gamma_2^T\tilde{Q} & \dots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \dots & \Gamma_N^T\tilde{Q}\end{bmatrix}}_{\widehat{\Gamma}^T\widehat{\tilde{Q}}}\underbrace{\begin{bmatrix}\tilde{y}_1^d\\ \tilde{y}_2^d\\ \dots\\ \tilde{y}_N^d\end{bmatrix}}_{\tilde{y}^d}$$

$$+\underbrace{\begin{bmatrix}\tilde{y}_1^{d^T} & \tilde{y}_2^{d^T} & \dots & \tilde{y}_N^{d^T}\end{bmatrix}}_{\tilde{y}^{d^T}}\underbrace{\begin{bmatrix}\tilde{Q} & 0 & \dots & 0\\ 0 & \tilde{Q} & \dots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \dots & \tilde{Q}\end{bmatrix}}_{\widehat{\tilde{Q}}}\underbrace{\begin{bmatrix}\tilde{y}_1^d\\ \tilde{y}_2^d\\ \dots\\ \tilde{y}_N^d\end{bmatrix}}_{\tilde{y}^d}$$

$$=\tilde{x}^T\widehat{\Gamma}^T\widehat{\tilde{Q}}\widehat{\Gamma}\tilde{x}-2\tilde{x}^T\widehat{\Gamma}^T\widehat{\tilde{Q}}\tilde{y}^d+\tilde{y}^{d^T}\widehat{\tilde{Q}}\tilde{y}^d$$

Similarly for the second sum

$$\sum_{k=0}^{N-1}\Delta u_k^T D\Delta u_k=\Delta u_0^T D\Delta u_0+\Delta u_1^T D\Delta u_1+\cdots+\Delta u_{N-1}^T D\Delta u_{N-1}$$

$$=\underbrace{\begin{bmatrix}\Delta u_1^T & \Delta u_2^T & \dots & \Delta u_{N-1}^T\end{bmatrix}}_{\Delta u^T}\underbrace{\begin{bmatrix}D & 0 & \dots & 0\\ 0 & D & \dots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \dots & D\end{bmatrix}}_{\widehat{D}}\underbrace{\begin{bmatrix}\Delta u_1\\ \Delta u_2\\ \dots\\ \Delta u_{N-1}\end{bmatrix}}_{\Delta u}$$

$$=\Delta u^T\widehat{D}\Delta u$$

Hence the total cost function can be written in matrix form as:

$$J=\tilde{x}^T\widehat{\Gamma}^T\widehat{\tilde{Q}}\widehat{\Gamma}\tilde{x}-2\tilde{x}^T\widehat{\Gamma}^T\widehat{\tilde{Q}}\tilde{y}^d+\tilde{y}^{d^T}\widehat{\tilde{Q}}\tilde{y}^d+\Delta u^T\widehat{D}\Delta u \tag{A.6}$$

On the other hand the discrete dynamic system evolution can also be expressed in a compact form evaluating the system $\tilde{x}_{k+1} = \tilde{A}_k \tilde{x}_k + \tilde{B}_k \Delta u_k$ at each sample time

$$
\begin{aligned}
\tilde{x}_1 &= \tilde{A}_0 \tilde{x}_0 + \tilde{B}_0 \Delta u_0 \\
\tilde{x}_2 &= \tilde{A}_1 (\tilde{A}_0 \tilde{x}_0 + \tilde{B}_0 \Delta u_0) + \tilde{B}_1 \Delta u_1 = \tilde{A}_1 \tilde{A}_0 \tilde{x}_0 + \tilde{A}_1 \tilde{B}_0 \Delta u_0 + \tilde{B}_1 \Delta u_1 \\
\tilde{x}_3 &= \tilde{A}_2 (\tilde{A}_1 \tilde{A}_0 \tilde{x}_0 + \tilde{A}_1 \tilde{B}_0 \Delta u_0 + \tilde{B}_1 \Delta u_1) + \tilde{B}_2 \Delta u_2 \\
&= \tilde{A}_2 \tilde{A}_1 \tilde{A}_0 \tilde{x}_0 + \tilde{A}_2 \tilde{A}_1 \tilde{B}_0 \Delta u_0 + \tilde{A}_2 \tilde{B}_1 \Delta u_1 + \tilde{B}_2 \Delta u_2 \\
&\vdots \\
\tilde{x}_N &= \tilde{A}_{N-1} \tilde{A}_{N-2} \dots \tilde{A}_1 \tilde{A}_0 \tilde{x}_0 + \tilde{A}_{N-1} \tilde{A}_{N-2} \dots \tilde{A}_1 \tilde{B}_0 \Delta u_0 + \tilde{A}_{N-1} \tilde{A}_{N-2} \dots \tilde{A}_2 \tilde{B}_1 \Delta u_1 + \dots \\
&\qquad \dots + \tilde{A}_{N-1} \tilde{A}_{N-2} \tilde{B}_{N-3} \Delta u_{N-3} + \tilde{A}_{N-1} \tilde{B}_{N-2} \Delta u_{N-2} + \tilde{B}_{N-1} \Delta u_{N-1}
\end{aligned}
$$

Arranging matrix form

$$
\underbrace{\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \vdots \\ \tilde{x}_{N-2} \\ \tilde{x}_{N-1} \\ \tilde{x}_N \end{bmatrix}}_{\tilde{x}} = \underbrace{\begin{bmatrix} \tilde{A}_0 \\ \tilde{A}_1 \tilde{A}_0 \\ \tilde{A}_2 \tilde{A}_1 \tilde{A}_0 \\ \vdots \\ \tilde{A}_{N-3} \dots \tilde{A}_0 \\ \tilde{A}_{N-2} \dots \tilde{A}_0 \\ \tilde{A}_{N-1} \dots \tilde{A}_0 \end{bmatrix}}_{\tilde{A}} \tilde{x}_0 + \underbrace{\begin{bmatrix} \tilde{B}_0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \tilde{A}_1 \tilde{B}_0 & \tilde{B}_1 & 0 & \dots & 0 & 0 & 0 \\ \tilde{A}_2 \tilde{A}_1 \tilde{B}_0 & \tilde{A}_2 \tilde{B}_1 & \tilde{B}_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \tilde{A}_{N-3} \dots \tilde{A}_1 \tilde{B}_0 & \tilde{A}_{N-3} \dots \tilde{A}_2 \tilde{B}_1 & \tilde{A}_{N-3} \dots \tilde{A}_3 \tilde{B}_2 & \dots & \tilde{B}_{N-3} & 0 & 0 \\ \tilde{A}_{N-2} \dots \tilde{A}_1 \tilde{B}_0 & \tilde{A}_{N-2} \dots \tilde{A}_2 \tilde{B}_1 & \tilde{A}_{N-2} \dots \tilde{A}_3 \tilde{B}_2 & \dots & \tilde{A}_{N-2} \tilde{B}_{N-3} & \tilde{B}_{N-2} & 0 \\ \tilde{A}_{N-1} \dots \tilde{A}_1 \tilde{B}_0 & \tilde{A}_{N-1} \dots \tilde{A}_2 \tilde{B}_1 & \tilde{A}_{N-1} \dots \tilde{A}_3 \tilde{B}_2 & \dots & \tilde{A}_{N-1} \tilde{A}_{N-2} \tilde{B}_{N-3} & \tilde{A}_{N-1} \tilde{B}_{N-2} & \tilde{B}_{N-1} \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \vdots \\ \Delta u_{N-2} \\ \Delta u_{N-1} \\ \Delta u_N \end{bmatrix}}_{\Delta u}
$$

We obtain

$$
\tilde{x} = \tilde{A} \tilde{x}_0 + \tilde{B} \Delta u \tag{A.7}
$$

Replacing in the cost function

$$
J = (\tilde{A}\tilde{x}_0 + \tilde{B}\Delta u)^T \widehat{\Gamma}^T \widehat{\tilde{Q}} \widehat{\Gamma} \tilde{A}\tilde{x}_0 + \tilde{B}\Delta u - 2(\tilde{A}\tilde{x}_0 + \tilde{B}\Delta u)^T \widehat{\Gamma}^T \widehat{\tilde{Q}} \tilde{y}^d + \tilde{y}^{d^T} \widehat{\tilde{Q}} \tilde{y}^d + \Delta u^T \widehat{D} \Delta u \tag{A.8}
$$

Several of the terms on this sum do not depend on Δu hence are irrelevant in the optimizatio problem with respect to Δu and can be ignored. Taking that into account and expending yields:

$$
J = 2\Delta u^T \tilde{B}^T \widehat{\Gamma}^T \widehat{\tilde{Q}} (\widehat{\Gamma} \tilde{A} \tilde{x}_0 - \tilde{y}^d) + \Delta u^T (\tilde{B}^T \widehat{\Gamma}^T \widehat{\tilde{Q}} \widehat{\Gamma} \tilde{B} + \widehat{D}) \Delta u \tag{A.9}
$$

All that is left is to express the constraints in a convenient form, first recall the constraints on the input and the output:

$$
u_{k_{\min}} \leq u_k \leq u_{k_{\max}} \begin{cases} u_k \leq u_{k_{\max}} \\ -u_k \leq -u_{k_{\min}} \end{cases} \qquad y_{k_{\min}} \leq y_k \leq y_{k_{\max}} \begin{cases} y_k = C_k x_k \leq y_{k_{\max}} \\ -y_k = -C_k x_k \leq -y_{k_{\min}} \end{cases}
$$

Can be written in matrix form using the augmented state $\tilde{x}_k$ as:

$$\begin{bmatrix} y_k \\ u_k \end{bmatrix} = \begin{bmatrix} C_k & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq \begin{bmatrix} u_{k_{\max}} \\ y_{k_{\max}} \end{bmatrix} \Longrightarrow \Gamma_k \tilde{x}_k \leq \tilde{y}_{\max}$$

$$\begin{bmatrix} -y_k \\ -u_k \end{bmatrix} = \begin{bmatrix} -C_k & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq - \begin{bmatrix} u_{k_{\min}} \\ y_{k_{\min}} \end{bmatrix} \Longrightarrow -\Gamma_k \tilde{x}_k \leq -\tilde{y}_{\min}$$

Arranging in matrix form

$$\underbrace{\begin{bmatrix} \Gamma_1 & 0 & \dots & 0 \\ 0 & \Gamma_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \Gamma_N \\ -\Gamma_1 & 0 & \dots & 0 \\ 0 & -\Gamma_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & -\Gamma_N \end{bmatrix}}_{S_\gamma} \underbrace{\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_N \end{bmatrix}}_{\tilde{x}} \leq \underbrace{\begin{bmatrix} \tilde{y}_{1_{\max}} \\ \tilde{y}_{2_{\max}} \\ \vdots \\ \tilde{y}_{N_{\max}} \\ -\tilde{y}_{1_{\min}} \\ -\tilde{y}_{2_{\min}} \\ \vdots \\ -\tilde{y}_{N_{\min}} \end{bmatrix}}_{\tilde{y}_{\lim}} \tag{A.10}$$

Replacing $\tilde{x}$ gives

$$S_\gamma \tilde{x} = S_\gamma(\tilde{A}\tilde{x}_0 + \tilde{B}\Delta u) = S_\gamma \tilde{A}\tilde{x}_0 + S_\gamma \tilde{B}\Delta u \leq \tilde{y}_{\lim} \tag{A.11}$$

Now recalling the constraints in the rate of change of the input

$$\Delta u_{k_{\min}} \leq \Delta u_k \leq \Delta u_{k_{\max}} \begin{cases} \Delta u_k \leq \Delta u_{k_{\max}} \\ -\Delta u_k \leq -\Delta u_{k_{\min}} \end{cases}$$

In matrix form:

$$\underbrace{\begin{bmatrix} I & 0 & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & I \\ -I & 0 & \dots & 0 \\ 0 & -I & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & -I \end{bmatrix}}_{S_I} \underbrace{\begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix}}_{\Delta u} \leq \underbrace{\begin{bmatrix} \Delta u_{1_{\max}} \\ \Delta u_{2_{\max}} \\ \vdots \\ \Delta u_{N_{\max}} \\ -\Delta u_{1_{\min}} \\ -\Delta u_{2_{\min}} \\ \vdots \\ -\Delta u_{N_{\min}} \end{bmatrix}}_{\Delta u_{\lim}} \tag{A.12}$$

Finally all the constraints can be written in a single equation:

$$\underbrace{\begin{bmatrix} S_\gamma \tilde{B} \\ S_I \end{bmatrix}}_{S} \Delta u \leq \underbrace{\begin{bmatrix} \tilde{y}_{\lim} - S_\gamma \tilde{A}\tilde{x}_0 \\ \Delta u_{\lim} \end{bmatrix}}_{M_{\lim}} \tag{A.13}$$

So the original optimal control problem can be written as a quadratic optimization problem

$$\min_{\Delta u} \Big\{ J = 2\Delta u^T \tilde{B}^T \widehat{\Gamma}^T \widehat{\tilde{Q}} (\widehat{\Gamma} \tilde{A} \tilde{x}_0 - \tilde{y}^d) + \Delta u^T (\tilde{B}^T \widehat{\Gamma}^T \widehat{\tilde{Q}} \widehat{\Gamma} \tilde{B} + \widehat{D}) \Delta u \Big\} \quad \text{s.t.} \quad S\Delta u \leq M_{\text{lim}} \tag{A.14}$$

This optimization problem is convex, thus if a solution exist then it is a global solution.

A.3 Quaternions

A.3.1 Quaternion definition and properties

Quaternions are an extension of the complex numbers in the sense that a quaternion is written as

$$q = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} \tag{A.15}$$

where q_0, q_1, q_2 and q_3 are real numbers and the quaternion units $\hat{i}$, $\hat{j}$ and $\hat{k}$ satisfy the relations:

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1 \tag{A.16}$$

Quaternions can also be thought as vectors with four components in the sense that if q and r are the quaternions:

$$\begin{aligned} q &= q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} \\ r &= r_0 + r_1\hat{i} + r_2\hat{j} + r_3\hat{k} \end{aligned} \tag{A.17}$$

Then their sum is another quaternion defined as expected:

$$r + q = (r_0 + q_0) + (r_1 + q_1)\hat{i} + (r_2 + q_2)\hat{j} + (r_3 + q_3)\hat{k} \tag{A.18}$$

However, unlike common vectors the relations of the quaternion units A.16 allow the definition of the multiplication of quaternions using the distributive law:

$$\begin{aligned} rq =& (q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k})(r_0 + r_1\hat{i} + r_2\hat{j} + r_3\hat{k}) \\ =& q_0r_0 + q_0r_1\hat{i} + q_0r_2\hat{j} + q_0r_3\hat{k} + q_1r_0\hat{i} + q_1r_1\hat{i}^2 + q_1r_2\hat{i}\hat{j} + q_1r_3\hat{i}\hat{k} \\ &+ q_2r_0\hat{j} + q_2r_1\hat{j}\hat{i} + q_2r_2\hat{j}^2 + q_2r_3\hat{j}\hat{k} + q_3r_0\hat{k} + q_3r_1\hat{k}\hat{i} + q_3r_2\hat{k}\hat{j} + q_3r_3\hat{k}^2 \\ =& (q_0r_0 - q_1r_1 - q_2r_2 - q_3r_3) + (q_0r_1 + q_1r_0 - q_2r_3 + q_3r_2)\hat{i} \\ &+ (q_0r_3 + q_1r_3 + q_2r_0 - q_3r_1)\hat{j} + (q_0r_3 - q_1r_2 + q_2r_1 + q_3r_0)\hat{k} \end{aligned} \tag{A.19}$$

This multiplication is known as a quaternion product. The quaternion units suggest another way to represent quaternions and that is as a sum on an scalar and a vector:

$$q = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} = q_o + \vec{q} \tag{A.20}$$

Where $\vec{q} = q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ can be interpreted as a vector in the space. Using this last form the addition and multiplication of quaternions can be expressed in a more compact form: If $q = q_0 + \vec{q}$ and $r = r_0 + \vec{r}$ then

$$\begin{aligned} r + q &= (r_o + q_0) + \vec{q} + \vec{r} \\ rq &= r_0q_0 - \vec{r} \cdot \vec{q} + r_0\vec{q} + q_0\vec{r} + \vec{r} \times \vec{q} \end{aligned} \tag{A.21}$$

As an extension of the complex numbers it is natural to extend the notions of norm and complex conjugate as the following operations: Let $q = q_0 + \vec{q} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ be a quaterion, then its norm $|q|$ is defined as:

$$|q|^2 = q_0^2 + |\vec{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 \tag{A.22}$$

and the complex conjugate:

$$q^* = q_0 - \vec{q} = q_0 - q_1\hat{i} - q_2\hat{j} - q_3\hat{k} \tag{A.23}$$

A.3.2 Quaternions as rotation operators

Let $\vec{v}$ be a vector in space, that is $\vec{v} \in \mathbb{R}^3$. Although $\vec{v}$ is not a quaternion it can be identified with the quaternion $v = 0 + \vec{v}$, v is called a pure quaternion since it does not have scalar part. This identification is bijective and allows the definition of a quaternion acting on a vector using the quaternion multiplication simply replacing the vector with the corresponding pure quaternion: If $\vec{v}$ is a vector and q a quaternion then

$$q\vec{v} \equiv q(0 + \vec{v}) = -\vec{q} \cdot \vec{v} + q_0\vec{v} + \vec{q} \times \vec{v} \tag{A.24}$$

This product is not necessarily a pure quaternion, and so can not be identified with a vector in space, which is what would be desired of a rotation operator acting on a vector. In order to guarantee that the resulting quaternion is pure the definition of the rotation operator given by the quaternion q is:

$$R_q(\vec{v}) = qvq^* \tag{A.25}$$

There v is the quaternion corresponding to $\vec{v}$. Although equation A.25 gives a pure quaternion it still does not represent a rotation operation since the norm of the resulting vector needs not to be the same as the original one (which is required for a rotation). In order to preserve the norm of the vector $\vec{v}$ an unit quaternion q is required.

If q is an unit quaternion then using the scalar plus vector notation we have that

$$q = q_0 + \tilde{q}\hat{u} \tag{A.26}$$

Where $\hat{u}$ is a unit vector and $q_0^2 + \tilde{q}^2 = 1$, this last condition implies that an angle α exists such that $q_0 = \cos(\alpha)$ and $\tilde{q} = \sin(\alpha)$. Replacing the unit quaternion q can be written as

$$q = \cos(\alpha) + \sin(\alpha)\hat{u} \tag{A.27}$$

And so at last we obtain the desired result:

Theorem 2. *Let q be an unit quaternion. Then the rotation operator given by the quaternion q is defined as:*

$$\begin{aligned} R_q : \mathbb{R}^3 &\longrightarrow \mathbb{R}^3 \\ \vec{v} &\longmapsto R_q(\vec{v}) = qvq^* \end{aligned}$$

Where v is the pure quaternion associated with $\vec{v}$. If $q = q_0 + \vec{q}$ then the operator can be written explicitly as:

$$R_q(\vec{v}) = (q_0^2 - |\vec{q}|^2)\vec{v} + 2(\vec{q}\cdot\vec{v})\vec{q} + 2q_0(\vec{q}\times\vec{v}) \tag{A.28}$$

This operator is linear and satisfies $|R_q(\vec{v})| = |\vec{v}|$. Further if $q = \cos(\alpha) + \sin(\alpha)\hat{u}$ then the axis of rotation is the vector $\hat{u}$ and the angle of rotation is 2α.

Proof. See [33] □

Since R_q is a linear operator fixing a basis B it has a corresponding matrix $[R_q]_B$. In the canonical basis E the corresponding matrix is given by:

$$[R_q]_E = \begin{bmatrix} 2(q_0^2+q_1^2)-1 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 2(q_o^2+q_2^2)-1 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 2(q_0^2+q_3^2)-1 \end{bmatrix} \tag{A.29}$$

Since quaternions $q = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ and Euler's angles $\{\phi, \theta, \psi\}$ can be used to represent rotations in space it is possible alternate between the two explicitly using the following equations [30] :

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \tag{A.30}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan 2(2(q_0q_1+q_2q_3), q_0^2-q_1^2-q_2^2+q_3^2) \\ \arcsin(2(q_0q_2-q_3q_1)) \\ \arctan 2(2(q_0q_3+q_1q_2), q_0^2+q_1^2-q_2^2-q_3^2) \end{bmatrix} \tag{A.31}$$

A.4 Parametric values for the simulation

Quadrotor parameters:

1. Mass: $m = 0.43$[Kg]
2. Moments of inertia: $I_x = 4.856\times10^{-3}$, $I_y = 4.856\times10^{-3}$, $I_z = 8.801\times10^{-3}$
3. Distance form the center of mass to the motors $L = 0.2$m
4. Propeller moment of inertia: $I_r = 1\times10^{-5}$
5. Propeller lift constant #1: $K_1^i = 3.88\times10^{-7}$
6. Propeller lift constant #2: $K_2^i = 0$
7. Propeller torque constant # 1: $B_1^i = 9.96\times10^{-9}$
8. Propeller torque constant # 2: $B_3^i = 4.33\times10^{-9}$

9. Maximum angle of attack $\alpha_{max_i} = 1.2\text{rad}$
10. Minimum angle of attack $\alpha_{min_i} = -1.2\text{rad}$
11. Nominal angle of attack $\bar{\alpha}_i = 0.5\text{rad}$
12. Maximum x-axis velocity $v_{x_{max}} = 20\frac{\text{m}}{\text{s}}$
13. Maximum y-axis velocity $v_{y_{max}} = 20\frac{\text{m}}{\text{s}}$
14. Maximum z-axis velocity $v_{z_{max}} = 20\frac{\text{m}}{\text{s}}$
15. Maximum x-axis acceleration $a_{x_{max}} = 5\frac{\text{m}}{\text{s}^2}$
16. Maximum y-axis acceleration $a_{y_{max}} = 5\frac{\text{m}}{\text{s}^2}$
17. Maximum z-axis acceleration $a_{z_{max}} = 5\frac{\text{m}}{\text{s}^2}$
18. Motor dynamic constant $K_{m_i} = 100$.

Control parameters:

1. Reference generation optimal control problem cost function matrices:

$$Q = \text{diag}(10, 10) \qquad\qquad R = 0.04$$

2. Translational optimal control problem cost function matrices:

$$Q = \text{diag}(1, 1, 1, 0, 0, 0) \qquad\qquad R = \text{diag}(0.1, 0.1, 1)$$

3. Extended Modal Series Method cost function matrices:

$$Q = 1 \times 10^{-5}\text{diag}(1, 1, 1, 5, 5, 5) \qquad\qquad R = \text{diag}(4, 4, 4)$$

4. Motor input cost function weight $r_u = 10$.
5. Angular control sampling time $T_{\text{ang}} = 2[\text{ms}]$.
6. Angular prediction horizon $H_{p_{\text{ang}}} = 50[\text{ms}]$.
7. Translational control sampling time $T_{\text{tra}} = 50[\text{ms}]$.
8. Translational prediction horizon $H_{p_{\text{tra}}} = 0.5[\text{s}]$.
9. Motor control sampling time $T_{\text{mot}} = 80[\mu\ \text{s}]$.
10. Motor prediction horizon $H_{p_{\text{mot}}} = 2[\text{ms}]$.
11. Reference generation optimal control problem saturations:

$$\text{Sat} = \begin{bmatrix} 4 & 4 & 10 & 10 & 20 & 20 \end{bmatrix}$$

A.5 Algorithms

A.5.1 Extended Modal Series Method Algorithm

Algorithm 1 OCP solution using EMSM

1: **procedure** EXTENDED MODAL SERIES METHOD ▷ Sub optimal control law
2: Input the functions F and G, the matrices Q and R and the performance criteria e_1 and e_2.
3: Construct the functions ψ and ϕ (equations 3.7 and 3.8)
4: $K \leftarrow 1$ ▷ Initialize the order of the approximation
5: $J_0 = 0$ ▷ Initialize the value of the cost function
6: Expand ψ and ϕ in a Taylor Series around $(0, 0)$ up to order K (for $\psi(x_\epsilon; \lambda_\epsilon)$ change $\phi(0)$ for $\psi(0)$)
7: Approximate the vector functions $x_\epsilon(t)$ and $\lambda_\epsilon(t)$ by

$$x_\epsilon(t) \approx \sum_{k=1}^{K} \epsilon^k g_k(t) \qquad \lambda_\epsilon(t) \approx \sum_{k=1}^{K} \epsilon^k h_k(t)$$

8: Replace $x_\epsilon(t)$ and $\lambda_\epsilon(t)$ in the approximations and expand.
9: $k \leftarrow 0$
10: **for** $k \leq K^2$ **do** ▷ To obtain the coefficient for ϵ^K
11: **if** $k = K$ **then**
12: Replace ϵ^k with 1
13: **else**
14: Replace ϵ^k with 0
15: **end if**
16: **end for**
17: Evaluate and assign the previous expression to $\Psi_K(t)$ for the ψ approximation and to $\Phi_K(t)$ for the ϕ approximation.
18: Form the $2n$ system of differential equations $\dot{g}_K(t) = \Phi_K(t)$, $\dot{h}_K(t) = \Psi_K(t)$.
19: Solve the system of differential equations, using the boundary conditions

$$\begin{cases} g_1(t_o) = x_o \\ h_1(t_f) - 2Pg_1(t_f) = 0 \end{cases} \quad \text{for } K = 1 \text{ or } \quad \begin{cases} g_K(t_o) = 0 \\ h_K(t_f) - 2Pg_K(t_f) = 0 \end{cases} \quad \text{for } K > 1$$

20: Update the control law and the corresponding state trajectory

$$u_K(t) = -R^{-1}G^T \left(\sum_{k=1}^{K} g_k(t) \right) \left(\sum_{k=1}^{K} h_k(t) \right) \qquad x_K(t) = \sum_{k=1}^{K} g_k(t)$$

21: Evaluate the control performance $J_K = \frac{1}{2} \int_{t_o}^{t_f} \left(x_K^T(t) Q x_K(t) + u_K^T(t) R u_K(t) \right) dt$
22: **if** $\frac{|J_K - J_{K-1}|}{|J_K|} > e_1$ or $\|x(t_f) - x_f\| > e_2$ **then**
23: $K \leftarrow K + 1$
24: Go back to step 6 ▷ Increase the order of the approximation
25: **end if**
26: **end procedure**

A.5.2 Explicit formula for $\xi^{\psi}_{\omega,l}$

The algorithm consists only in the evaluation of the formula given by (3.19) and (3.20). It is the same for $\xi^{\psi}_{\omega,l}$ and $\xi^{\phi}_{\omega,l}$, changing only in the function that is derivated in step 4 (ψ_l for $\xi^{\psi}_{\omega,l}$ and ϕ_l for $\xi^{\phi}_{\omega,l}$), therefore only one is shown in algorithm 2.

Algorithm 2 Explicit formula for $\xi^{\psi}_{\omega,l}$

1: **procedure** MALGUS FORMULA
2:     $\xi^{\psi}_{\omega,l} = 0$
3:     **for** $1 \leq |\alpha| + |\beta| \leq \omega$ **do**
4:         Calculate $\frac{D^\alpha D^\beta}{\alpha!\beta!}\psi_l(0)$
5:         $X_3 \leftarrow 0$
6:         **for** $|\theta| = \omega - |\alpha| - |\beta|$ **do**
7:             $X_2 \leftarrow 1$
8:             **for** $1 \leq q \leq n$ **do**
9:                 $X_1 \leftarrow 0$
10:                 **for** $u + v = \alpha_q + \beta_q + \theta_q$ **do**
11:                     **if** $\alpha_q\beta_q \neq 0$ **then**
12:                         **if** $\alpha_q \leq u \leq \omega\alpha_q$ **then**
13:                             Calculate $F_q(u)$ from (3.43)
14:                         **end if**
15:                         **if** $\beta_q \leq v \leq \omega\beta_q$ **then**
16:                             Calculate $G_q(v)$ from (3.44)
17:                         **end if**
18:                     **else**
19:                         $F_q(u)G_q(v) = 0$
20:                     **end if**
21:                     $X_1 = X_1 + F_q(u)G_q(v)$
22:                 **end for**
23:                 $X_2 = X_2X_1$
24:             **end for**
25:             $X_3 = X_3 + X_2$
26:         **end for**
27:         $\xi^{\psi}_{\omega,l} = \xi^{\psi}_{\omega,l} + X_3$
28:     **end for**
29: **end procedure**

Bibliography

[1] O. Purwin and R. D'Andrea, "Performing and extending aggressive maneuvers using iterative learning control," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 1–11, 2011.

[2] A. A. Paranjape, K. C. Meier, X. Shi, S. . Chung, and S. Hutchinson, "Motion primitives and 3-d path planning for fast flight through a forest," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 2940–2947, 2013.

[3] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation : Safe and efficient load manipulation with aerial robots," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[4] H. M. Pari, M. Khosravi, and M. Haeri, "Model predictive based dynamic path planning for single target tracking and formation," in *3rd International Conference on Control, Instrumentation, and Automation, ICCIA 2013*, pp. 340–344, 2014.

[5] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[6] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Australasian Conference on Robotics and Automation, ACRA*, 2012.

[7] N. K. Ure and G. Inalhan, "Autonomous control of unmanned combat air vehicles: Design of a multimodal control and flight planning framework for agile maneuvering," *IEEE Control Systems*, vol. 32, no. 5, pp. 74–95, 2012.

[8] S. Spedicato, G. Notarstefano, H. H. Bülthoff, and A. Franchi, *Aggressive maneuver regulation of a quadrotor UAV*, vol. 114 of *Springer Tracts in Advanced Robotics*, pp. 95–112. Springer, 2016.

[9] A. Pretorius and E. Boje, "Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive maneuvers," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 12208–12213, 2014.

[10] A. V. Terekhov, J. . Mouret, and C. Grand, "Stochastic optimization of a neural network-based controller for aggressive maneuvers on loose surfaces," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 4782–4787, 2010.

[11] F. Xu, H. Chen, X. Gong, and Q. Mei, "Fast nonlinear model predictive control on fpga using particle swarm optimization," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 310–321, 2016.

[12] Z. Yan and J. Wang, "Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 746–756, 2012.

[13] Z. Li, Y. Xia, C. . Su, J. Deng, J. Fu, and W. He, "Missile guidance law based on robust model predictive control using neural-network optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1803–1809, 2015.

[14] G. H. Negri, V. H. B. Preuss, M. S. M. Cavalca, and J. De Oliveira, "Differential evolution optimization applied in multivariate nonlinear model-based predictive control," in *2015 Latin-America Congress on Computational Intelligence, LA-CCI 2015*, 2016.

[15] A. Mozaffari, N. L. Azad, and J. K. Hedrick, "A nonlinear model predictive controller with multiagent online optimizer for automotive cold-start hydrocarbon emission reduction," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4548–4563, 2016.

[16] J. Xu, T. Van Den Boom, L. Busoniu, and B. De Schutter, "Model predictive control for continuous piecewise affine systems using optimistic optimization," in *Proceedings of the American Control Conference*, vol. 2016-July, pp. 4482–4487, 2016.

[17] B. Sabetghadam and A. A. Safavi, "Fast nonlinear model predictive control for teleoperation systems using computationally efficient optimization techniques," in *22nd Iranian Conference on Electrical Engineering, ICEE 2014*, pp. 1371–1376, 2014.

[18] S. S. Sajjadi, N. Pariz, A. Karimpour, and A. Jajarmi, "An off-line nmpc strategy for continuous-time nonlinear systems using an extended modal series method," *Nonlinear Dynamics*, vol. 78, no. 4, pp. 2651–2674, 2014.

[19] Y. Sun, "Modelling, identification and control of a quad-rotor drone using low-resolution sensing," master thesis, Graduate College of the University of Illinois, Urbana, Illinois, 2012.

[20] B. Michini, J. Redding, N. Kemal Ure, M. Cutler, and J. P. How, "Design and flight testing of an autonomous variable-pitch quadrotor," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2978–2979, 2011.

[21] M. Cutler and J. P. How, "Analysis and control of a variable-pitch quadrotor for agile flight," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 137, no. 10, 2015.

[22] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007*, vol. 2, pp. 1670–1689, 2007.

[23] M. Cutler, N. Kemal Ure, B. Michini, and J. P. How, "Comparison of fixed and variable pitch actuators for agile quadrotors," in *AIAA Guidance, Navigation, and Control Conference 2011*, 2011.

[24] J. Puerta, J. Sánchez, I. Bataller, C. Fu, and P. Cervera, "Ar drone identification and navigation control at cvg-upm," in *XXXIII Jornadas de Automática CEA-IFAC*, 2012.

[25] G. Martin, "Modelling and control of the parrot ar.drone," *ADFA Journal of Undergraduate Engineerig Research*, vol. 5, no. 1, 2012.

[26] T. Luukkonen, "Modelling and control of quadcopter." Independent research project in applied mathematics, August 2011.

[27] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems*, vol. 28, no. 2, pp. 51–64, 2008.

[28] A. Pedersen, "Test and modelling of four rotor helicopter rotors," master thesis, Technical University of Denmark, October 2006.

[29] P. . Bristeau, P. Martin, E. Salaun, and N. Petit, "The role of propeller aerodynamics in the model of a quadrotor uav," in *2009 European Control Conference, ECC 2009*, pp. 683–688, 2014.

[30] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *2013 European Control Conference, ECC 2013*, pp. 3864–3869, 2013.

[31] D. Naidu, *Optimal Control Systems.* Electrical Engineering Textbook Series, Idaho State University: CRC Press, 2003.

[32] J. J. Duistermaat and J. A. C. Kolk, *Multidimensional Real Analysis I: Differentiation.* Cambridge Studies in Advanced Mathematics, Cambridge: Cambridge University Press, 2004.

[33] J. B. Kuipers, *"Quaternions and Rotation Sequences, A Primer with Applications to Orbits, Aerospace and Virtual Reality.* Princeton, New Jersey: Princeton University Press, 1990.