Vertical navigation and trajectory tracking for UAV

Pedro Alexandre Pais Pereira
pedro.pais.pereira@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

January 2021

Abstract

The recent technological achievements in sensors and embedded systems have prompted the integration in several civil and military applications of autonomous systems such as Unmanned Aerial Vehicles (UAV) capable of efficiently executing surveying and monitoring missions in locations otherwise considered dangerous or inaccessible. Some of those missions require the acquisition of sensor data close to objects and areas of interest located on the ground. In this thesis, it is necessary to determine the path that will allow the UAV to roam the mission site safely and dynamically by easily avoiding terrain obstacles and while maintaining a constant and close proximity to the ground. This thesis proposes an optimal trajectory generation algorithm based on gradient search methods to generate the flight path for the UAV and an optimization controller capable of precisely track those trajectories. The proposed solution considers a non-linear model dynamics approximation of the autonomous system and determines a discrete trajectory that satisfies Terrain-following and waypoint tracking constraints and minimizes a cost function that aims to reduce the mission's time and acceleration. The trajectory optimizer performs an offline preliminary optimization based on previously acquired data of the terrain elevation in the mission's site. The path planner's performance is tested and discussed for several scenarios with different elevation profiles. In order to precisely perform Trajectory tracking and path following of the determined optimum trajectory in the presence of disturbances, this paper also presents the design, implementation and testing of a Model Predictive Control (MPC). This online planning and control strategy is also an optimization problem that aims to minimize a quadratic function designed to penalize state errors while satisfying the imposed dynamic constraints. The MPC was tested in a physics simulation environment (Gazebo) where a realistic terrain and a model of the Fleet of dRones for radIological inspEction, commuNication anD reScue project [1] UAV was integrated to further validate the stability and robustness of the designed controller.

Keywords: UAV, Optimal trajectory, Terrain-following, Waypoint tracking, Model Predictive Control

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAV) have gained significant popularity among academic researchers and industries which has promoted a reduction in the costs of several sensors, actuators and batteries, making them more practical and accessible, technically enabling their integration in a wide range of civil applications.

In some of those applications, measurements have to be performed close to the objects of interest that are placed on the ground. This is the case for project FRIENDS (Fleet of dRones for radIological inspEction, commuNication anD reScue), where this article is integrated on. Therefore, this paper's goal arises from the requirement that the project's UAV must navigate in close proximity to the ground, where the expected low intensity sources of radiation are located, in order to estimate the radioactivity level of the scenario using reduced sensitivity sensors.

The use of UAVs for monitoring and mapping of radioactive scenarios has gained popularity in recent years due to their simplicity and versatility with the appearance of several academic, military projects and commercial options such as MOBISIC [2] and AVID LLC [3].

For this work, the main interest is to design a strategy capable of realizing autonomous flight by optimizing and following trajectories in space, converting the multi-rotor platform, driven initially by a human operator, into a fully Unmanned Autonomous Vehicle. In the first stage, we address the design of a trajectory optimizer based on the objectives and constraints that follow from the requirements of the project's surveillance and inspection missions. In the second stage, we tackle the problem of precisely and safely tracking this trajectory in real-time by relying on concepts of Model Predictive Control (MPC).

Regarding the optimal trajectory generation, research studies differ mostly on the type of trajectories and on the type of constraints used. The implementation of optimal control strategies based on the discretization of the non-linear dynamics of aerial systems has been widely used to determine terrain-following and avoidance trajectories [4, 5] by augmenting the optimization problem with specific constraints. A different approach addressed in [6] is the use of fuzzy logic methods to model the flight altitude and gradient of the terrain. It is also possible to optimize a continuous spline to perform terrain avoidance flight. This is the case of [7] where the search structure is extended using a spline-RRT* method to generate smooth and continuous paths.

The introduction of the waypoint tracking problem in the trajectory generation has also been solved by many different approaches. One example is proposed in [8] that generates an optimal and smooth trajectory capable of going through waypoints and minimize the trajectory derivatives by approximating the UAV dynamics using a discrete linear model. Paper [9] introduces a formulation where the progress through the waypoints is bounded by complementarity constraints and mixed-integer progress variables. This objective can also be fulfilled by generating a continuous polynomial where the goal is to minimize path derivatives and total segment time [10].

Our approach is based on the work of [11] that presents a fast optimum control algorithm for quadrotors capable of flying through waypoints by assigning a certain number of discretization steps between each waypoint and then optimizing the time for each of the segments as an independent optimization variable, resulting in discrete path with different time steps.

When it comes to precise UAV control in the presence of disturbances, many techniques and augmentations have been studied in a variety of different control designs. The simplest and most common control strategy is the standard PID (Proportional, Integral and Derivative) controller [12, 13] where a feedback mechanism is capable of rejecting disturbances by running the state errors through a closed-loop. A more complex approach is presented in [14, 15] where precise trajectory-tracking control is developed based on a Lyapunov back-stepping technique that approximates the non-linear dynamics of the UAV to a linear state space. In recent years the technological developments prompted the implementation of optimal online planning and control strategies that consider the dynamic model of the autonomous system and determines an optimum trajectory based on user-defined objectives and constraints [16, 17].

Our approach is based on the system simplification presented in [18] where the low-level control is assumed to be performed onboard by an independent autopilot and the translational dynamics is controlled off-board by a Non-linear Model Predictive Control coupled with an Extended Kalman Filter applied to an approximation of the UAV dynamics.

2. Hexarotor Dynamics and Kinematics

In order to derive the equations of motion (EoM) for a multirotor I consider the preliminary assumption that the curvature of the Earth and its rotation can be ignored in the mission's area since the duration and scale of the operation is considerably small when compared to the Earth. From those assumptions we can define two useful coordinate systems as shown in Figure 1:

- A Earth fixed and Inertial frame $\{\mathfrak{I}\}$ ($\mathbf{e}_{\mathbf{x}}^{\mathfrak{I}}$; $\mathbf{e}_{\mathbf{y}}^{\mathfrak{I}}$; $\mathbf{e}_{\mathbf{z}}^{\mathfrak{I}}$) tangent to the earth surface that uses the East, North, Up (ENU) system of coordinates.
- A body fixed frame $\{\mathfrak{B}\}$ ($\mathbf{e}_{\mathbf{x}}^{\mathfrak{B}}$; $\mathbf{e}_{\mathbf{y}}^{\mathfrak{B}}$; $\mathbf{e}_{\mathbf{z}}^{\mathfrak{B}}$) whose center coincides with the center of mass of the vehicle.

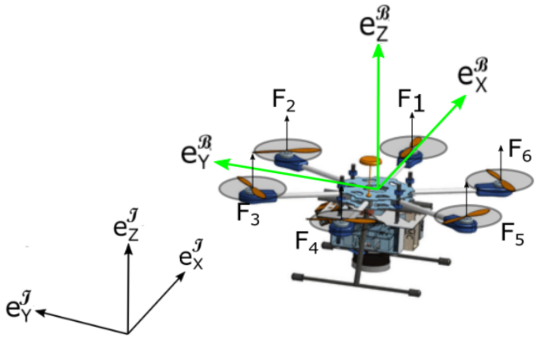


Figure 1: Representation of Inertial and Body frames.

The vehicle configuration is described by the position of the center of mass in the inertial frame $\{\mathfrak{I}\}$ and the orientation of the $\{\mathfrak{B}\}$ frame with respect to the inertial frame [19]. The last is usually defined by means of Euler angles: roll ϕ, pitch θ and yaw ψ. For sake of simplicity, let us denote the vectors for inertial position and body frame orientation by means of $\mathbf{p} = [x,\ y,\ z]^T$ and $\eta = [\phi,\ \theta,\ \psi]^T$, respectively, where $\phi \in \,]-\pi/2,\ \pi/2[$, $\theta \in \,]-\pi,\ \pi[$, and $\psi \in \,]-\pi,\ \pi[$. In order to deduce the equations describing the orientation of the mobile frame relative to the fixed one, we further defined an orthogonal rotation matrix $\mathcal{R}^{\mathfrak{B}\mathfrak{I}}(\eta)$ which can is determined through successive rotations about the three axes.

The FRIENDS' project drone is a generic Hexarotor X geometry, with six rotors, mounted symmetrically along two orthogonal axes. For this UAV model, two main forces act during flight: the gravitational force and the thrust force. These forces acting on the vehicle can then be written, respectively, as

- Force of gravity in $\mathfrak{I}$ frame: $F_g = -mg$
- Rotor's thrust in $\mathfrak{B}$ frame: $F_i = k_n\omega_i^2$

where m is the model's mass, g is the gravitational acceleration, k_n is the thrust constant and $\omega_i \geq 0$ is the rotation speed of the i-th rotor.

As mentioned in [18], we can additionally consider the influence of two important aerodynamic effects that appear in the case of dynamic manoeuvrers. These effects are the blade flapping and induced drag, which induce forces in the x-y rotor plane that contribute to the horizontal stability UAV as shown in [20]. Although these two effects are treated as separate in [20], [21] showed that it is possible to combine these effects into a lumped coefficient $A_d = diag(c_d^x,\ c_d^y,\ 0)$. If the UAV is symmetric in the x-y axis the two coefficients can be considered equal, $c_d^x = c_d^y = c_d$.

This leads to the aerodynamic drag force $F_{a,i}$ defined as:

$$F_{a,i} = F_i A_d \mathrm{v}_{\mathfrak{B}} \tag{1}$$

The roll and pitch torques derive from the propellers' thrust and can be determined by $M_T = \sum r_i \times F_i$, where r_i is the momentum arm of each propeller's thrust in the $\{\mathfrak{B}\}$ referential. Finally, the yaw torque is produced when the angular velocity of propellers rotating "clockwise" differs from the angular velocity of propellers rotating "counterclockwise" creating unbalance in the sum of moments with respect to $\mathbf{e}_\mathbf{z}^{\mathfrak{B}}$.

Assuming that the UAV is a symmetrical and rigid structure we can derive the equations of motion for the full system

$$\dot{\mathbf{p}} = \mathrm{v}_{\mathfrak{I}}\,, \tag{2a}$$

$$\dot{\mathrm{v}}_{\mathfrak{I}} = \frac{1}{m}\left(\mathcal{R}^{\mathfrak{B}\mathfrak{I}}\sum_{i=1}^{6}\left(F_i + F_{a,i}\right) + F_{ext}\right) + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}\,, \tag{2b}$$

$$\dot{\mathrm{w}}_{\mathfrak{B}} = \mathcal{J}^{-1}(-\mathrm{w}_{\mathfrak{B}} \times \mathcal{J}\mathrm{w}_{\mathfrak{B}} + M_{\mathfrak{B}})\,. \tag{2c}$$

where $\mathrm{v}_{\mathfrak{I}}$ is the linear UAV velocity in frame $\{\mathfrak{I}\}$ and $\mathrm{w}_{\mathfrak{B}}$ is the angular velocity of the aerial vehicle around each axis of the $\{\mathfrak{B}\}$ frame. F_{ext} is the vector of external forces acting on the vehicle, $M_{\mathfrak{B}}$ is the vector of aerodynamic torques caused by the propellers and $\mathcal{J}$ is the inertia matrix of the hexacopter model.

In this paper, the attitude dynamic model of the UAV employed in both optimization problems is based on a cascaded approach studied in [22] that assumes that the vehicles low-level control is performed by a separate module based on a closed-loop Proportional architecture that is able to independently track desired roll, ϕ^{des}, pitch, θ^{des}, and yaw, ψ^{des}, angles [18]. Therefore, the rotational dynamics of the Hexacopter can be approximated by a simple and generic model that reliably and consistently represents the dynamic behavior of the internal autopilot modules. These simplified dynamics can either be modeled mathematically if the controller architecture is known or empirically by using experimental data and system identification techniques to approximately determine the parameters of the dynamic simplification.

Similarly to [18], for the MPC we choose a first order dynamics to represent the inner-loop behaviour of the UAV represented as follows:

$$\dot{\eta} = \frac{1}{\tau}(K\eta_{cmd} - \eta) \tag{3}$$

where $\eta_{cmd} = [\phi_{cmd},\ \theta_{cmd},\ \psi_{cmd}]^T$ is the vector containing attitude control references and $\tau = [\tau_\phi,\ \tau_\theta,\ \tau_\psi]^T$, $K = [K_\phi,\ K_\theta,\ K_\psi]^T$ are the vectors of time constants and gains of the inner-loop behaviour respectively. Inside the MPC the yaw control and dynamics are simply simulated to reduce the non-linear effect of a varying yaw trajectory, since yaw does not influence the translational dynamics of the UAV.

In the trajectory optimization problem, using the same approximated model as the MPC is important to, not only, reduce the computation time, but also to facilitate the integration of the optimization results in the MPC. However, the model used in the MPC does not allow for constraints in angular velocity or linear acceleration that are required to obtain a smooth trajectory. To this end, the attitude dynamics applied in this optimization problem are approximated by a second-order system that allows the control inputs to be the same as in the MPC but gives a good estimation of the angular velocity. The attitude second-order dynamics is described by the following equations:

$$\dot{\mathrm{w}}_{\mathfrak{I}} = -2\xi\omega\mathrm{w}_{\mathfrak{I}} + \omega^2(K\eta_{cmd} - \eta)\,. \tag{4}$$

where $\mathrm{w}_{\mathfrak{I}} = [\dot{\phi},\ \dot{\theta},\ \dot{\psi}]^T$ is the vector of the Inertial Frame angular rates, $\xi = [\xi_\phi,\ \xi_\theta,\ \xi_\psi]^T$, $\omega = [\omega_\phi,\ \omega_\theta,\ \omega_\psi]^T$ and $K = [K_\phi,\ K_\theta,\ K_\psi]^T$ are the vectors of damping constants, natural frequency and gains for the second order approximation of the attitude dynamics respectively and $\eta_{cmd} = [\phi_{cmd},\ \theta_{cmd},\ \psi_{cmd}]^T$ is the vector of attitude reference inputs.

3. Trajectory Generation Formulation

In this section, we tackled the formulation and implementation of an optimum control trajectory capable of performing safe and efficient terrain following and waypoint tracking while satisfying the UAV dynamic constraints.

As mentioned the attitude dynamics applied in this optimization problem are approximated by a second order system (Eq. (4)). The translational dynamics implemented are a simplified version of equation (2b) where the aerodynamic drag term and external forces are ignored and the individual rotor thrust is substituted by a total and normalized thrust, $\tilde{T}$, input as follows:

$$\dot{\mathrm{v}}_{\mathfrak{I}} = \mathcal{R}^{\mathfrak{B}\mathfrak{I}}(\eta) \begin{bmatrix} 0 \\ 0 \\ \tilde{T} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} . \qquad (5)$$

In order to minimize and set a maximum limit for linear acceleration, we added an integrator to the system dynamics, since the translational dynamics that describe a multi-rotor do not include acceleration as a state. The dynamics of the added velocity integrator is as follows:

$$\dot{\mathrm{v}}_{int} = a \, . \qquad (6)$$

where v_{int} is the state of the integrator that represents the velocity of the UAV and a is the acceleration defined as an input in the integrator's dynamics. In order to link the integrator dynamics with the overall multirotor dynamic model, the velocity v_{int} is constrained to be equal to the velocity of the UAV model, $\mathrm{v}_{\mathfrak{I}}$.

The described system dynamics represent the following state and input vectors:

$$X = \begin{bmatrix} \mathbf{p}^T & \mathrm{v}_{\mathfrak{I}}^T & \eta^T & \mathrm{w}_{\mathfrak{I}}^T & \mathrm{v}_{int}^T \end{bmatrix}^T \qquad (7a)$$

$$U = \begin{bmatrix} \tilde{T} & \phi_{cmd} & \theta_{cmd} & \psi_{cmd} & a^T \end{bmatrix}^T \qquad (7b)$$

3.1. Trajectory Optimization problem

The general statement of a trajectory optimization problem where the goal is to minimize an objective function is:

$$\begin{array}{rl} \text{Minimize} & \mathcal{F}(\chi) \\ \text{w.r.t.} & \chi \\ \text{subject to} & G_i(\chi) = 0, \; i \in \mathcal{E} \\ & H_i(\chi) \leq 0, \; i \in \mathcal{I} \end{array} \qquad (8)$$

where $\mathcal{F}$ is the optimization function, $\chi = [X^T, \; U^T]^T$ is the vector of optimization variables, $\mathcal{E}$ and $\mathcal{I}$ are the sets of equality and inequality constraints, respectively.

From the continuous dynamic equations that describe the UAV, it is now necessary to define a set of discrete constraints that can be used by the optimization problem. From the several discretization methods that could be applied, we decided to use collocation methods since they have numerical advantages over other techniques, such as shooting methods. We incorporate the system dynamics as equality constraints between time steps X_i and X_{i+1} using a first-order backward Euler approximation as follows:

$$0 = X_{i+1} - X_i - f(X_{i+1}, U_{i+1}) \Delta t_k \, . \qquad (9)$$

where Δt_k is the discretization time step for the trajectory segment k that links waypoint $k-1$ to waypoint k and $f(X_{i+1}, U_{i+1})$ is the derivative of the robot states in the collocation point $i+1$ derived from the solution of the dynamic equations.

One of the objectives of this optimization problem is to generate a trajectory capable of passing through pre-determined waypoints defined in the 2D horizontal plane and represented by the vector $p_k^w = [x_k^w, \; y_k^w]$.

In our formulation, each of the K waypoints is allocated to a specific discretization node defined by the number of collocation points, N, between waypoints. To guarantee that the trajectory passes each waypoint $k = 1, \; \ldots, \; K$ within a tolerance σ_w at node kN, a constraint-based formulation can be used, such as

$$(p'_{kN} - p_k^w)^T (p'_{kN} - p_k^w) \leq \sigma_w^2 \qquad (10)$$

where p'_{kN} is the part of the state vector X that represents the horizontal position at node kN.

The main aim of the trajectory optimizer is to compute a terrain-following path. In our formulation, it is assumed that the terrain being followed is preliminarily available from a known DEM (Digital Elevation Map) file. In order for the solution algorithm to be effective, smooth derivatives of the terrain data are required. This can be achieved by approximating the data matrix with a tensor product cubic B-spline.

The terrain following objective was incorporated in the optimization problem as a weighted norm of the aircraft altitude difference to the desired value above the terrain, h^{des}, as follows:

$$\mathcal{F}_{TF} = \int_{t_0}^{t_f} k_{TF} [z(t) - h^{des} - h_{terrain}]^2 \, dt \qquad (11)$$

where $h_{terrain}$ is the height of the terrain at the multirotor's position and k_{TF} is the weight of this cost function in the total cost of the optimization problem.

The Terrain Avoidance problem requires the definition of additional path constraints to guarantee that the UAV does not fly above a maximum height or below a minimum safety height.

$$h^{-} \leq z(t) - h_{terrain} \leq h^{+} \tag{12}$$

Another goal of the optimization problem is to minimize the total trajectory time. The main problem of a time-optimal trajectory arises when this objective is coupled with the waypoint tracking problem. When we allocated each waypoint to a specific node we are also constraining the time at which the trajectory passes this waypoint. To tackle this problem a time elastic band-based formulation is used where the time to go from one waypoint to the next, ΔT_k, is an optimization variable, resulting in different discretization steps for every trajectory segment of the form

$$\Delta t_k = \frac{\Delta T_k}{N} . \tag{13}$$

where N is the independent variable representing the number of discretization steps (Fig. 2).

In order to optimize the total trajectory time a cost function penalizing the terminal time is added:

$$\mathcal{F}_t = \int_{t0}^{t_f} k_{time} dt = k_{time} \sum_{i=1}^{k} \Delta T_k \tag{14}$$

where k_{time} is the terminal cost factor.

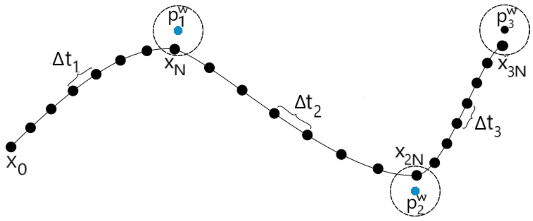


Figure 2: Time Elastic Band Formulation.

Two additional costs were also added to the optimization function: a minimum acceleration cost $\mathcal{F}_a$ and a yaw function $\mathcal{F}_\psi$ that aims to keep ψ equal to the angle between the previous and next waypoints.

$$\mathcal{F}_a = \int_{t0}^{t_f} a^T k_a a \; dt \tag{15a}$$

$$\mathcal{F}_\psi = \int_{t0}^{t_f} k_\psi [\psi - \psi_k^{des}]^2 dt \tag{15b}$$

where a is the acceleration vector of the UAV, ψ_k^{des} is the desired yaw for the waypoint segment $k = 1, \ldots, K$ and k_a, k_ψ are the weight coefficients for the acceleration and yaw costs.

In order to satisfy all the necessary objectives, the problem's cost function is defined as the sum of all the formulated objective functions. The compromise in the solution is determined by the relative weighting applied to each cost which can substantially change the optimum trajectory.

Finally, we introduced constraints in the final and initial states creating a two boundary value problem as follows.

$$\begin{aligned} &X(t_0) = X_0 && \text{bound on initial state} \\ &X_f^{-} \leq X(t_f) \leq X_f^{+} && \text{bound on final state} \end{aligned} \tag{16}$$

An additional constraint for maximum horizontal velocity defined as follows is added to ensure the sensor data collected by the UAV can be correctly used.

$$\mathrm{v}_x^2 + \mathrm{v}_y^2 \leq (V_{xy}^{+})^2 \tag{17}$$

Other than the path constraints some state and input inequality constraints were applied to prevent over actuation leading to undesired high pitch and roll.

$$X^{-} \leq X_i \leq X^{+} \tag{18a}$$

$$U^{-} \leq U_i \leq U^{+} \tag{18b}$$

The Trajectory optimization problem described in this chapter is implemented and tested using the CasADi toolbox. This open-source software is very versatile and contains algorithms of sensitive analysis capable of determining the derivatives of the cost function and constraints without the direct Jacobian and Hessian formulations from the user. From the several NLP solvers that interface with CasADi, we choose to use IPOPT which is an open-source primal-dual interior-point method.

4. Optimal Control Formulation

This section is dedicated to the design and implementation of a continuous-time non-linear Model Predictive Controller, which can follow in real-time the paths computed by the trajectory optimizer described in the previous chapter. The method implemented was based on the work presented in [18, 23], adapted to the problem at hand and software used.

In order for the control optimization algorithm to be able to run in real-time the computation time at each time step must be considerably smaller than the sampling time of the controller. This can be achieved by reducing the number of state variables and simplifying the system dynamics of the robot.

As mentioned, the MPC implements a first-order approximation of the attitude dynamics described in equation (3). The translational dynamics implemented is similar to equation (2b) where, similarly to the trajectory optimization, the individual rotor thrust is substituted by a total and normalized thrust input.

The state-space vector of the optimization control problem is:

$$X_i = \begin{bmatrix} \mathbf{p}^T & \mathrm{v}_{\mathfrak{I}}^T & \eta^T \end{bmatrix}^T \tag{19}$$

And the control input vector, U, consists of tree inputs,

$$U_i = \begin{bmatrix} \tilde{T} & \phi_{cmd} & \theta_{cmd} \end{bmatrix}^T \tag{20}$$

4.1. External Disturbance Estimation

The external disturbances F_{ext} present in the dynamic model of the UAV are estimated by an Extended Kalman Filter (EKF) that employs the same translational model used in the MPC (2b), but with the second-order attitude dynamic approximation that was also used in the trajectory optimization design, (4). The EKF uses the autopilot estimated state information to iteratively determine the external forces acting on the multirotor.

This estimator will reduce, not only, the effect of unpredictable external disturbances, like wind, but also capture modeling error that the model may have in the attitude and drag approximations applied, achieving zero steady-state tracking error [24].

4.2. Model Predictive Control

MPC is a method of iteratively solving an optimization problem for a finite time horizon and can be generally formulated as:

$$\begin{aligned} \text{Minimize} \quad & \mathcal{F}_N(X,U) \\ \text{w.r.t.} \quad & X, U \\ \text{subject to} \quad & \dot{X}_i = f(X_i, U_i) \\ & X_i \in \mathfrak{X}_{\mathfrak{C}} \\ & U_i \in \mathfrak{U}_{\mathfrak{C}} \end{aligned} \tag{21}$$

where N is the length of the prediction horizon and $\mathfrak{U}_{\mathfrak{C}}$, $\mathfrak{X}_{\mathfrak{C}}$ represent the set of input and state constraints respectively. $X = [X_0, \ \ldots, \ X_N]$ and $U = [U_0, \ \ldots, \ U_{N-1}]$ are state and input sequences for the time horizon.

In this implementation, a real-time iteration scheme based on Gauss-Newton is applied to approximate the non-linear optimization problem and iteratively improve the solution during the runtime. A Multiple shooting technique is employed to discretize the system dynamics with a sampling time Δt over a coarse discrete-time grid $t_0, \ \ldots, \ t_N$.

When formulating the MPC optimization problem, it is important to ensure that it can be solved in the short time available. For that reason, the optimization problem is typically cast into a Quadratic programming (QP) formulation where the objective function used is the common and popular Linear Quadratic Regulator (LQR) of the form:

$$\begin{aligned} \mathcal{F}_N(X,U) = & \|X_N - X_N^{ref}\|_R^2 \\ & + \sum_{i=0}^{N-1} \left(\|X_i - X_i^{ref}\|_Q^2 + \|U_i - U_i^{ref}\|_P^2 \right) \end{aligned} \tag{22}$$

where i is the index along the prediction horizon, Q, R, P are the state error, goal state error and control action error weight matrices respectively, X_i^{ref} is the state reference signal at time step i and U_i^{ref} is the input reference.

In the precise trajectory tracking problem, the state and input references are calculated by the trajectory optimizer and represent the intended dynamics for the UAV, serving as a baseline for the trajectories to be generated by the MPC. However, this controller can also be used to optimize the path to reach a certain reference position or to hover above the current location by setting the reference X_i^{ref} constant in the optimization horizon.

For this simple control problem, apart from the dynamic constraints, the only extra constraints considered are bounds imposed on the control inputs and dynamic states, in order to avoid saturation of the actuators and prevent the system from showing unwanted behaviour.

$$U^- \leq U_i \leq U^+ \,, \tag{23a}$$

$$X^- \leq X_i \leq X^+ \,. \tag{23b}$$

The optimization problem was implemented in a C++ interface for ACADO that can be easily compiled and executed, solving the optimization problem and allowing for a quick preliminary validation test. The OCP can be expressed as a Non-linear Program that can be solved using qpOASES solver.

5. Simulation and Results

The methods presented in Sections 3 and 4 are tested and validated in this section where we present the results from the Python and CasADi trajectory optimization along with the simulation results of the MPC in a ROS and Gazebo environment using the FRIENDS' Hexacopter model and the PX4 autopilot firmware running in software-in-the-loop (SITL) mode. All the following tests and simulations were conducted on a Hp Laptop running Ubuntu 18.04 and equipped with an Intel Core i7-9750H CPU @2.60GHz and 16,00GB of RAM.

5.1. Trajectory Optimization Results

To validate the trajectory optimization algorithm, several different tests were made, varying the terrain profile, waypoint arrangement and even discretization steps. We consider the results of two scenarios to assess the capabilities and difficulties of the selected approach.

On the left side of table 1, we show the second-order attitude dynamic parameters which are constants obtained from the system identification procedure for the FRIENDS Hexacopter model. The right side of the table contains the state, input and path limits of the optimization problem.

Table 1: Trajectory optimization parameters and coefficients.

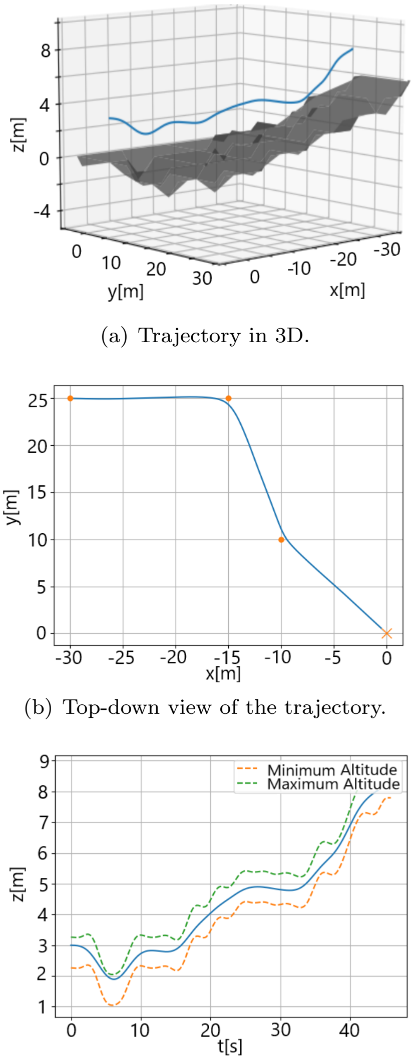
Attitude Constants		Optimization Limits	
K_ϕ	0.9757	V_{xy}^+	1.0 $[m/s]$
K_θ	0.9862	V_z^+	1.0 $[m/s]$
K_ψ	0.9762	$\dot{\phi}^+$	180 $[°/s]$
ω_ϕ	6.2179	$\dot{\theta}^+$	180 $[°/s]$
ω_θ	6.0429	$\dot{\psi}^+$	25.0 $[v/s]$
ω_ψ	3.8762	T^-	7.0 $[m/s^2]$
ξ_ϕ	0.9353	T^+	15.0 $[m/s^2]$
ξ_θ	0.9216	ϕ^+	25.0 $[°]$
ξ_ψ	0.8653	θ^+	25.0 $[°]$
		a^+	1.0 $[m/s^2]$

The first scenario is a simple example that uses real terrain elevation data obtained from the United States Geologic Survey National Elevation Dataset [25]. In this scenario we included three waypoints bounded to a specific node with a tolerance $\sigma_w = 0.5[m]$. The number of discretization steps was chosen to be $N = 80$ based on the horizontal distance and velocity between waypoints. For the height constraints we chose to set the minimum and maximum altitude allowed above the terrain to $h^- = 2.5m$ and $h^+ = 3.5m$ respectively.

The optimized trajectory is shown in figure 3 where it is possible to see that the waypoint tracking and terrain-following objectives of the trajectory were fulfilled with the altitude of the UAV remaining inside the allowable interval. The mean computation time over 50 optimizations was then determined to be 0.842 [s] for this scenario.

The second scenario is formulated to assess the results of the proposed approach when the terrain data contains significant discontinuities that could represent a cliff or wall. In this case, we introduced a 7 $[m]$ wall in the altitude matrix data with 0.9 $[m]$ of width.

Due to the exclusive vertical formulation of the terrain constraints, we can conclude that as the terrain becomes steeper and more discontinuous the feasible tunnel of the z coordinate becomes less wide and for a vertical wall it will result in an impossible problem where the maximum and minimum altitude constraint surfaces overlap. Therefore, for the following scenario we did not include the maximum altitude constraint but instead relied on the terrain following cost (Eq. (11)) to maintain the UAV close to the desired altitude.



(a) Trajectory in 3D.

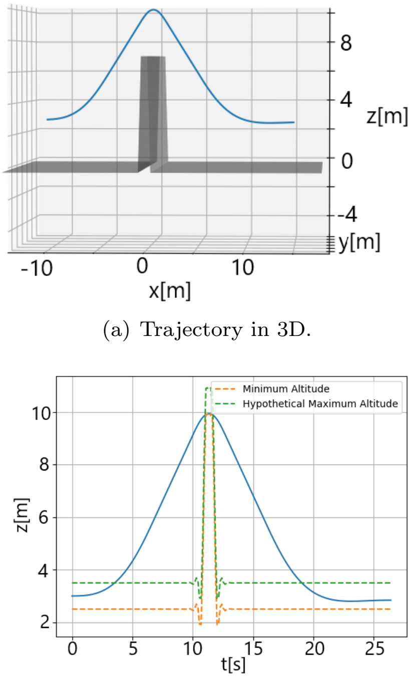
(b) Top-down view of the trajectory.

(c) Trajectory in 3D.

Figure 3: Optimized trajectory for scenario 1.

The effect of this cost weight is illustrated in the two following tests where all constants, weights and constraints are kept constant except the terrain-following weight k_{TF}. The number of discretization steps was chosen to be $N = 200$ and the minimum altitude allowed was $h^- = 2.5m$. We can conclude that if the weight k_{TF} is higher (Fig. 5) the resulting trajectory will climb the wall almost vertically to maintain the vertical distance to the ground as constant as possible if this weight is smaller (Fig. 4) the minimum time objective takes over and the trajectory transposes the wall with an arc-like trajectory.

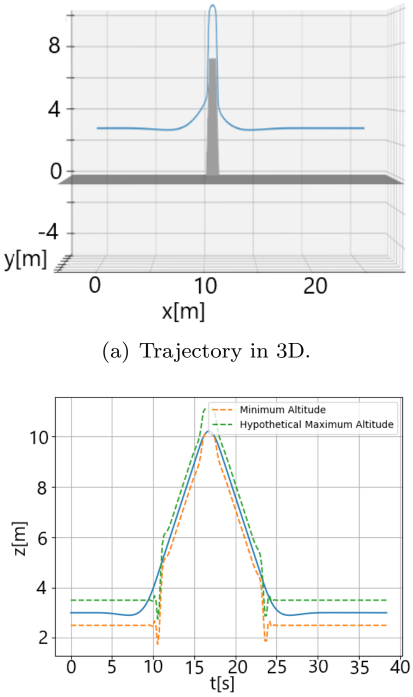
For the first case, the mean computation time for the iterative optimization process, in this case, is 5.719 [s] and for the second test, due to the high gradients of the altitude constraints and the closer proximity to the wall this test requires a higher computation time with a mean of 10.526 [s] over 50 optimizations.



(a) Trajectory in 3D.

(b) Top-down view of the trajectory.

Figure 4: Optimized trajectory for the first test of scenario 3.



(a) Trajectory in 3D.

(b) Top-down view of the trajectory.

Figure 5: Optimized trajectory for the second test of scenario 3.

Although the minimum altitude constraint is always fulfilled in both tests, the fact that the UAV is considered as a point means that in the second test a 3D UAV would collide with the wall in the horizontal plane. To solve this problem further improvement in the UAV representation or the minimum altitude constraint must be performed in future work.

5.2. Online Control Results

In order to test and verify the MPC discussed in section 4, we first designed and integrated a 3D CAD model of the FRIENDS' project Hexacopter (Fig. 6) into the Gazebo+SITL environment. The goal was not only to have a simulation model similar to the real Hexacopter but also to create an environment where it is possible and practical to simulate and test different sensors.

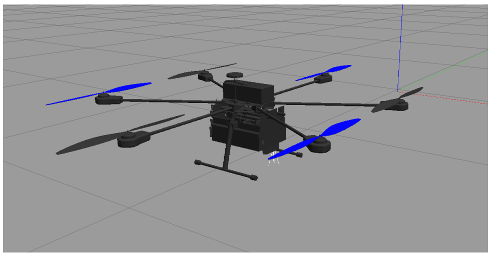


Figure 6: Gazebo model of the FRIENDS Project's drone.

The proposed navigation and control architecture of this work is illustrated in figure 7. Initially, the trajectory optimizer receives 2D waypoints and determines the optimum control path. The optimized trajectory is properly sampled and is ultimately used as a reference to be tracked by the position Modular Predictive Controller.

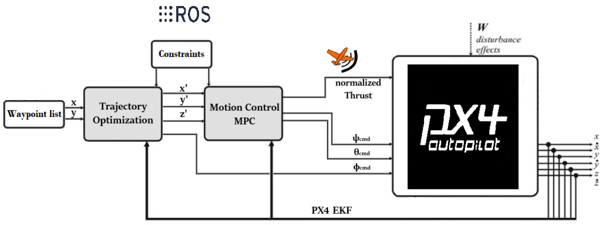


Figure 7: Multirotor Control Architecture.

In our setup, the optimization controller is running at 50 Hz while internally the prediction sampling time is $\Delta t = 0.1$ $[s]$, in this way we achieve a longer prediction horizon with less computational effort. By enforcing the iteration to run roughly 5x faster than the discretization time, we obtain small deviations of the predicted state vector between iterations which facilitates convergence. The prediction horizon is set to be divided into $N = 20$ steps resulting in 2 seconds of prediction time for each iteration.

The MPC parameters used in the following simulations are presented in table 2 and, similarly to the trajectory optimizer, were determined by applying the system identification.

Table 2: FRIENDS hexacopter parameters and control input constraints.

Attitude Constants		Optimization Limits	
K_ϕ	0.9951	V_{xy}^+	2.0 $[m/s]$
K_θ	0.9634	V_z^+	3.0 $[m/s]$
K_ψ	0.9900	T^-	4.0 $[m/s^2]$
τ_ϕ	0.1430	T^+	15.0 $[m/s^2]$
τ_θ	0.1650	ϕ^+	30.0 $[°]$
τ_ψ	0.3020	θ^+	30.0 $[°]$

As for the cost matrices, the implemented weights penalize mainly the position and velocity state errors that are fundamental for accurate trajectory tracking. The input weights are considerably smaller and will allow the optimizer to correct the most penalized errors by adjusting the attitude and mainly the thrust command references calculated by the trajectory optimizer.

Several simulations were executed and analysed to validate the stability and robustness of the designed MPC. In this paper, we chose to present two trajectory tracking simulations with two types of pre-computed trajectories.

5.3. Trajectory Tracking with Wind

We first present the results of the MPC when tracking a smooth continuous polynomial trajectory while the UAV is subject to simulated external forces. This test allows us to validate the capability of the control algorithm in eliminating the effect of unpredictable external disturbances. In order to perform this simulation, we used the wind plugin provided by the PX4 community.

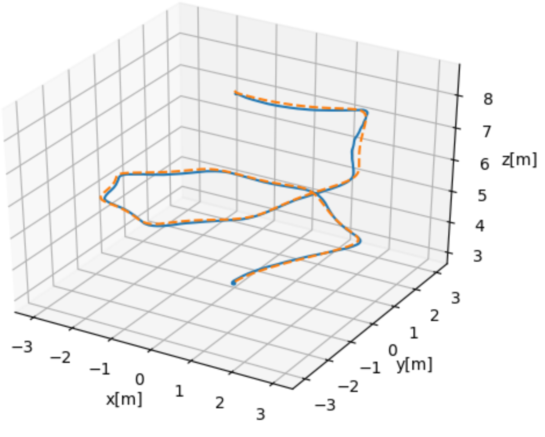
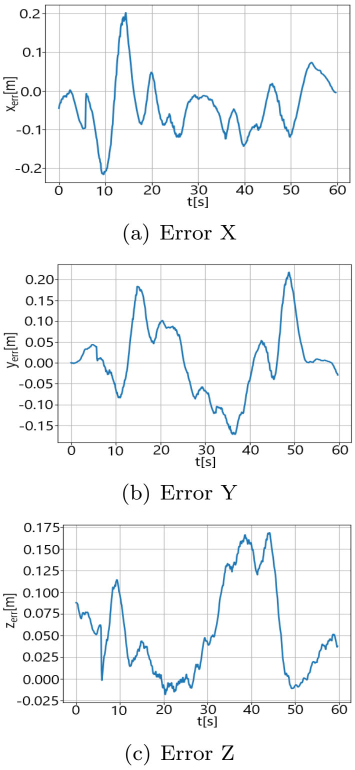


Figure 8: Tested Trajectory with Wind.

Under external wind of around $4[m/s]$, the position error is shown in Figure 9. The MPC controller achieved a maximum error of 20 cm in the sharp corners of the polynomial path, which validates the ability of this controller to eliminate the effects of external disturbances.



(a) Error X

(b) Error Y

(c) Error Z

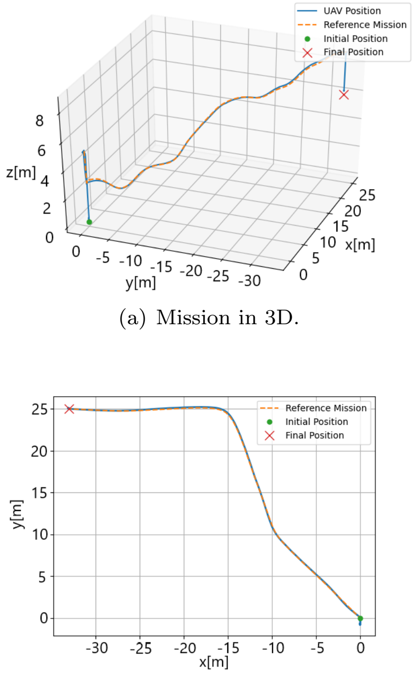
Figure 9: MPC Trajectory Tracking error with Wind.

5.4. Full Control Architecture Simulation

Finally, a simulation is performed where the path optimized in the first scenario of the trajectory generator is used as a reference for the MPC. The mission performed by the simulated UAV can be divided into 3 distinct stages:

Initially, the UAV performs vertical take-off using only the PX4 autopilot controlling loops. Then the second phase starts where the translational control is shifted from the autopilot to the off-board MPC. In this phase, the UAV starts by adjusting the height to the desired altitude above the terrain. After the terrain following trajectory is optimized the execution of the mission resumes. Finally, after arriving at the last waypoint the UAV lands using once again the PX4 autopilot controller and the mission terminates.

The full mission can be seen in figure 10 and the three phases are illustrated in figure 11 where the altitude of the UAV is represented as a function of time. We can therefore conclude that the designed tracking controller is also capable of efficiently and precisely track the trajectory optimized by the terrain following algorithm.



(a) Mission in 3D.

(b) Top-down view of the mission.

Figure 10: UAV terrain following Mission.

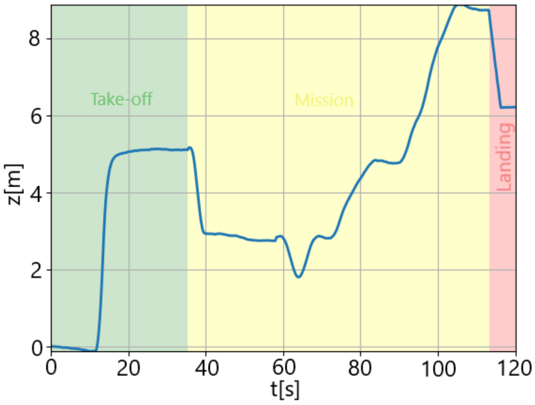


Figure 11: Altitude of the UAV in Mission.

6. Conclusions

This thesis proposes a solution for terrain-following and avoidance with an optimum control trajectory optimization and the design and validation of an online controller capable of following the optimized path even in the presence of external disturbances. The first goal was achieved by a non-linear optimization problem that employed the UAV dynamics to determine an optimum control trajectory that was able to pass through a pre-determined set of horizontal waypoints while keeping the vertical distance to the ground inside a safe and operational window. For the second objective, we tackled the design and implantation of a controller based on a Non-linear Model Predictive Control, where an iterative optimization problem generates control inputs in real-time. Finally, both algorithms were integrated together allowing for the validation of the designed navigation architecture.

References

[1] IPFN. Friends. `https://www.ipfn.tecnico.ulisboa.pt/FRIENDS/index.html`. [Online; accessed 30-12-2020].

[2] K. Boudergui, F. Carrel, T. Domenech, N. Guenard, J.Poli, A. Ravet, V. Schoepff, and R. Woo. Development of a drone equipped with optimized sensors for nuclear and radiological risk characterization. In *2011 2nd International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and their Applications*, pages 1–9, 2011.

[3] C. Cai, B. Carter, M. Srivastava, J. Tsung, J. Vahedi-Faridi, and C. Wiley. Designing a radiation sensing uav system. In *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, pages 165–169, 2016.

[4] Kosari, A., Maghsoudi, H, and Abolfazl, L. Path generation for flying robots in mountainous regions. *International Journal of Micro Air Vehicles*, 9(1):44–60, 2017.

[5] I. Khademi, B. Maleki, and A. Nasser-i Mood. Optimal three dimensional terrain following/terrain avoidance for aircraft using direct transcription method. In *2011 19th Mediterranean Conference on Control Automation (MED)*, pages 254–258, 2011.

[6] M. Rahim and Seyed Malaek. Aircraft terrain following flights based on fuzzy logic. *Aircraft Engineering and Aerospace Technology: An International Journal*, 83:94–104, 2011.

[7] D. Lee and D. H. Shim. Spline-rrt* based optimal path planning of terrain following flights for fixed-wing uavs. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 257–261, 2014.

[8] H. Eslamiat, Y. Li, N. Wang, A. K. Sanyal, and Q. Qiu. Autonomous waypoint planning, optimal trajectory generation and nonlinear tracking control for multi-rotor uavs. In *2019 18th European Control Conference (ECC)*, pages 2695–2700, 2019.

[9] Foehn, P. and Scaramuzza, D. Cpc: Complementary progress constraints for time-optimal quadrotor trajectories, 2020.

[10] Richter, C., Bry, A., and Roy, N. *Polynomial Trajectory Planning for Aggressive Quadrotor*

Flight in Dense Indoor Environments, pages 649–666. Springer International Publishing, 2016.

[11] Falanga, D., Foehn, P., Scaramuzza, D., Kuppuswamy, N., and Tedrake, R. Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload. 2017.

[12] Kada. Belkacem. Robust pid controller design for an uav flight control system. Newswood, 10 2011.

[13] Pounds, P., Bersak, D., and Dollar, A. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. *Autonomous Robots*, 33, 2012.

[14] Sun, L., Beard, R., and Pack, D. Trajectory-tracking control law design for unmanned aerial vehicles with an autopilot in the loop. *Proceedings of the American Control Conference*, pages 1390–1395, 2014.

[15] Xingling, S., Liu, J., Cao, H., Shen, C., and Honglun, W. Robust dynamic surface trajectory tracking control for a quadrotor uav via extended state observer. *International Journal of Robust and Nonlinear Control*, 28, 2018.

[16] Vasan G., Singh A., and Krishna K. Model predictive control for micro aerial vehicle systems (mav) systems. *ArXiv*, abs/1412.2356, 2014.

[17] C. Sferrazza, M. Muehlebach, and R. D'Andrea. Trajectory tracking and iterative learning on an unmanned aerial vehicle using parametrized model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5186–5192, 2017.

[18] Kamel, M., Burri, M., and Siegwart, R. Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles. *IFAC-PapersOnLine*, 50, 2017.

[19] Kamel, M., Alexis, K., Achtelik, M., and Siegwart, R. Fast Nonlinear Model Predictive Control for Multicopter Attitude Tracking on SO(3), 2015.

[20] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, 2012.

[21] S. Omari, M. Hua, G. Ducard, and T. Hamel. Nonlinear control of vtol uavs incorporating flapping dynamics. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2419–2425, 2013.

[22] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 21–28, 2010.

[23] Kamel, M., Stastny, T., Alexis, K., and Siegwart, R. *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System.* 2017.

[24] Borrelli, F., Bemporad, A., and Morari, M. *Predictive Control for Linear and Hybrid Systems.* Cambridge University Press, 2017.

[25] United States Geological Survey. The National Map Viewer. `https://viewer.nationalmap.gov`. [Online; accessed 19-12-2020].