33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

UNMANNED AERIAL VEHICLE'S TRAJECTORY OPTIMIZATION IN CONSTRAINED ENVIRONMENTS

Jiří Novák[1] & Peter Chudý[2]

[1]Ph.D. candidate, Faculty of Information Technology, Brno University of Technology
[2]Associate Professor, Faculty of Information Technology, Brno University of Technology

Abstract

The range of multirotor Unmanned Aerial Vehicle (UAV) applications has grown significantly over the last decade. This is to be attributed to their simple mechanical design, along with hovering and maneuvering capabilities, making them a popular choice in applications such as surveillance, aerial photography, cargo transport or infrastructure inspection. Varying mission requirements with respect to UAV target location, payload capacity, speed or time of flight combined with environmental constraints such as no-fly zones can be hard to satisfy without the use of modern trajectory optimization techniques. Trajectory optimization problems are often formulated using optimal control theory. An Optimal Control Problem (OCP) for generalized multirotor UAV introducing environmental constraints has been formulated and solved by a direct transcription approach. Furthermore, a Gauss pseudospectral method has been implemented, enabling to design a framework for UAV trajectory optimization problems in constrained environments.

Keywords: Gauss pseudospectral method, nonlinear programming, optimal control, trajectory optimization, unmanned aerial vehicle

Nomenclature

A_D	m^2	=	Drag area
b_m	$\mathrm{N \cdot m \cdot s \cdot rad^{-1}}$	=	Viscous friction constant
B_{pr}	–	=	Propeller blades
C_D	–	=	Drag coefficient
C_P	–	=	Power coefficient
C_T	–	=	Thrust coefficient
g	$\mathrm{m \cdot s^{-2}}$	=	Gravitational acceleration
h	m	=	Altitude
$\mathbf{I}^b$	$\mathrm{kg \cdot m^2}$	=	Inertia tensor
J_m	$\mathrm{kg \cdot m^2}$	=	Motor moment of inertia
k_e	$\mathrm{V \cdot s \cdot rad^{-1}}$	=	EM force constant
k_m	$\mathrm{N \cdot m \cdot A^{-1}}$	=	Motor torque constant
L	H	=	Inductance
m	kg	=	vehicle mass
O	–	=	Motor rotation orientation
R	Ω	=	Resistance
R_{pr}	m	=	Propeller radius
SoC	$\%$	=	State of charge
V_a	$\mathrm{m \cdot s^{-1}}$	=	Airflow magnitude
W	$\mathrm{m \cdot s^{-1}}$	=	Nominal wind speed
ρ	$\mathrm{kg \cdot m^{-3}}$	=	Air density
μ	–	=	Advance ratio

BEMT	=	Blade Element Momentum Theory
BFF	=	Body-Fixed Frame
BVP	=	Boundary Value Problem
CoM	=	Center of Mass
DC	=	Direct Current
DCM	=	Direction Cosine Matrix
EoM	=	Equations of Motion
ESA	=	European Space Agency
GNC	=	Guidance, Navigation, Control
IDW	=	Inverse Distance Weighting
IPOPT	=	Interior Point OPTimizer
IVP	=	Initial Value Problem
LG	=	Legendre-Gauss
LGL	=	Legendre-Gauss-Lobatto
LGR	=	Legendre-Gauss-Radau
NED	=	North, East, Down
NLP	=	Non-Linear Programming
OCP	=	Optimal Control Problem
ODE	=	Ordinary Differential Equation
SQP	=	Sequential Quadratic Programming
UAV	=	Unmanned Aerial Vehicle
6-DoF	=	6-Degrees-of-Freedom

1. Introduction

An increasing demand for the UAV autonomy drive the advancement in Guidance, Navigation and Control (GNC) solutions. The primary task of a Guidance unit is the construction of a feasible trajectory accounting for the vehicle dynamics while achieving respective mission objectives. The focus of presented work has been on the formulation of optimal control problems for multirotor UAV vehicles in variety of configurations such as shown in Figure 1 while posing operational and environmental constraints.

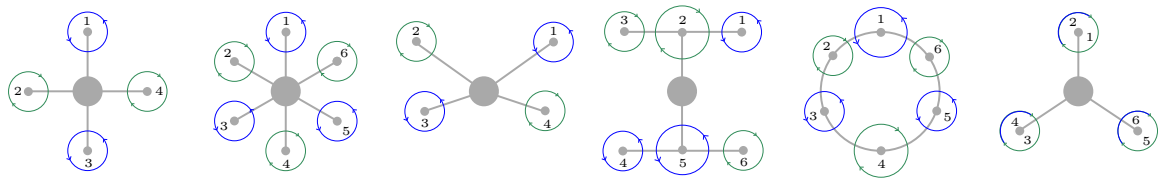


Figure 1 – Multirotor UAV range of configurations.

A trajectory optimization problem is typically solved by utilization of direct or indirect methods. Indirect methods consist of construction of the necessary and sufficient optimality conditions, discretizing these conditions and solving the own optimization problem. The direct method's approach in the OCP assumes a direct discretization of the trajectory, converting the problem to a parameter optimization task. Although the indirect approach yields an accuracy metric for the solution, construction of the adjoint equations is for many applications infeasible. The direct approach was selected due to it's easier formulation and availability of increasingly efficient algorithms for solving them. A generalized framework utilizing Gauss-pseudospectral transcription method for UAV trajectory optimization concerning operational and environmental constraints was implemented under the European Commission funded COMP4DRONES project. The paper provides an example of utilizing simplified kinematic model with no-fly zone constraints as well as full nonlinear 6-Degrees-of-Freedom (6-DoF) Equations of Motion (EoM) driven trajectory optimization problem with operational and environmental constraints.

2. Theoretical background and architecture

Let us consider a general OCP. The goal is to find a control vector $\mathbf{u}(t) \in \mathbb{R}^m$, that minimizes the functional

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\mathrm{d}t. \tag{1}$$

The running cost $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ (Lagrange problem) involves the state vector $\mathbf{x}(t) \in \mathbb{R}^n$ and the control vector along the trajectory. The terminal cost $\Phi : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ (Mayer problem) involves the state vector at initial $t_0 \in \mathbb{R}$ and final $t_f \in \mathbb{R}$ time. Equation (1) is usually referred to as the Bolza form [2]. The state vector is subjected to dynamic constraints characterized by Ordinary Differential Equation (ODE)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \qquad t \in [t_0, t_f] \tag{2}$$

boundary constraints $\Psi : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^q$ and path constraints $C : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^c$ of the form

$$\begin{aligned} \Psi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) &= \mathbf{0} \\ C(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}, t) &\leq \mathbf{0} \end{aligned} \tag{3}$$

with algebraic parameters $\mathbf{p} \in \mathbb{R}^l$. In contrast to Initial Value Problem (IVP) where the dependent variable is imposed at initial time, the Boundary Value Problem (BVP) requires the state vector to be determined at both points t_0 and t_f. It is assumed, that the dynamic constraints (2) involve continuous and differentiable functions. In practice, solving OCP requires iterative Newton-based algorithms on a finite set of variables. The process of converting an OCP to a finite-dimensional general Non-Linear Programming (NLP) problem is referred to as a transcription method [1]. Generally, the transcription method uses the domain discretization to convert the dynamic constraints to a problem with a finite set of variables. For the presented case, the Gauss pseudospectral transcription method [10] was

used to transcribe the OCP. The state and control variables are approximated using global interpolating polynomials on a nonuniform mesh of collocation points to prevent numerical issues such as Runge's phenomenon.

In context of a UAV trajectory optimization, the architecture shown in Figure 2 displays required steps taken to formulate the NLP problem and to find it's solution. In terms of inputs, the dynamic model and it's parameters (such as vehicle mass, moments of inertia or propeller radius) have to be specified. The second set of inputs concerns environment parameters definition. This includes specification of wind field, terrain and parameters such as air density or gravitational acceleration. The third set of inputs provides the required settings to construct the OCP. The objective function being minimized is usually constructed to find the minimum flight time trajectory or a minimum energy/fuel trajectory. The constraints and bounds of the problem as written in equation (3) may specify a range of different requirements on the optimized trajectory. The simplest are the boundary constraints defining requirements such as the initial and final position. Respective variable bounds may be set to respect operational limits of the vehicle such as maximum angular speed of the propellers. Path constraints in the form of inequality constraints are optionally specified to restrict some state or control dependent quantities. Path constraints can be used to define cylindrical no-fly zones or terrain constraints among others. Finally, the solver and its options, such as maximum number of iterations or stopping criteria including tolerances for constraints violation, can be specified by the user. In the transcription phase, the defined inputs are used to compose the OCP problem and check that it is correctly formulated. This is referred to by a preprocessing block (*) in Figure 2. In order to find the solution, the inbuilt gradient based solver has to be provided with functions which are twice continuously differentiable.

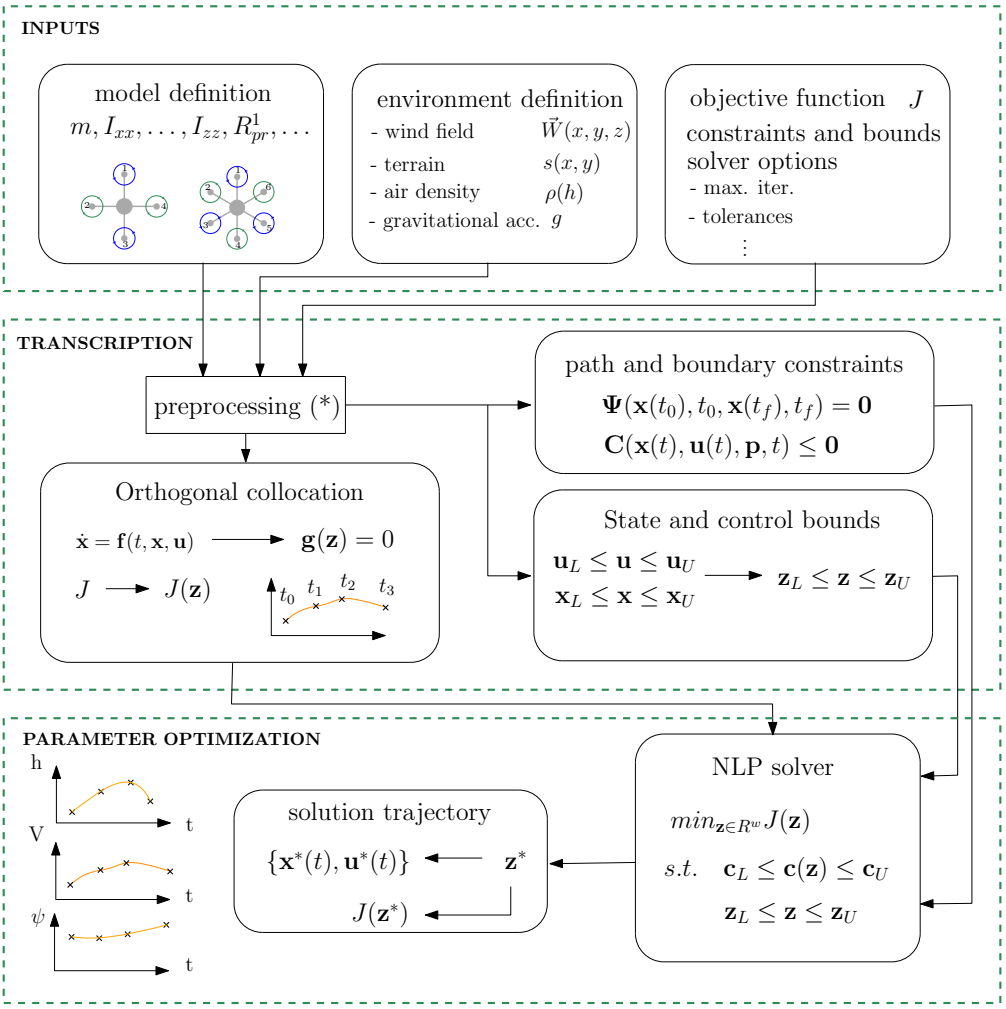


Figure 2 – Trajectory optimization framework block diagram.

For example, using traditional absolute value function may lead to convergence failures. Where non-differentiable functions are critical, these can be usually approximated and substituted so to have the desired properties. Once those cases are handled, an orthogonal collocation is performed to convert the dynamics to a set of equality constraints at the set of time points. An objective function is also discretized so that the running cost is computed by a quadrature. It is important to note that the final time is not fixed, so that it becomes a variable in the optimization problem. In some cases, we might be interested to optimize for other parameters such as battery mass. These remain constant over the entire time domain. The problem can then be solved by algorithms such as Interior Point OPTimizer (IPOPT) [11], sparse Sequential Quadratic Programming (SQP) solver SNOPT [13] or the European Space Agency (ESA) developed nonlinear optimization solver WORHP [14], among others. Once the NLP solver converges with the required tolerances, the trajectory is then easily retrieved from the optimal solution point along with the associated objective function value.

3. Generalized multirotor UAV modeling

Let us assume a rigid-body UAV. Its dynamic quantities are defined in either inertial reference frame $\mathscr{F}_e = (x_e, y_e, z_e, O_e)$ respecting North, East, Down (NED) convention or a Body-Fixed Frame (BFF) $\mathscr{F}_b = (x_b, y_b, z_b, O_b)$. The origins O_e and O_b are located at the vehicle Center of Mass (CoM). The Newton's second law of motion has been used for the derivation of the 6-DoF EoM [3]. A general form of translational equations expressed in the BFF can be written as

$$\mathbf{F}^b = m\left(\dot{\mathbf{v}}^b_{b/e} + \boldsymbol{\omega}^b_{b/e} \times \mathbf{v}^b_{b/e}\right), \tag{4}$$

where m is the UAV's mass, $\mathbf{v}^b_{b/e} = [u \;\; v \;\; w]^T$ is a velocity vector of $\mathscr{F}_b$ with respect to $\mathscr{F}_e$ expressed in $\mathscr{F}_b$ and $\boldsymbol{\omega}^b_{b/e} = [p \;\; q \;\; r]^T$ is the angular velocity vector of $\mathscr{F}_b$ with respect to $\mathscr{F}_e$ expressed in $\mathscr{F}_b$. Equation (4) can be derived from a general form in an inertial frame by expressing the velocity derivative in BFF using an angular velocity vector. The resulting force $\mathbf{F}^b$ acting on the UAV's CoM can be further decomposed. Since aerodynamic and gravitational forces are not generally expressed in BFF, a coordinate frame transformations need to be utilized.

The rotation at motion of a UAV is a result of moments acting on its body. The angular momentum theorem is introduced in a body-fixed coordinate system $\mathscr{F}_b$. Let $\mathbf{M}^b$ be the resulting moment of all external moments acting on the UAV's CoM. The moment equation is formed in terms of the vehicle's angular velocity $\omega^b_{b/e}$ in BFF as shown in equation (5).

$$\mathbf{M}^b = \mathbf{I}^b \dot{\boldsymbol{\omega}}^b_{b/e} + \boldsymbol{\omega}^b_{b/e} \times \mathbf{I}^b \boldsymbol{\omega}^b_{b/e} \tag{5}$$

The inertia tensor $\mathbf{I}^b$ is usually simplified due to the design's symmetry of a UAV body. In other cases, all components of the inertia tensor have to be known. In terms of a modular UAV design, the inertia tensor can be computed using the Parallel axis theorem over all UAV components.

The UAV can be described as a collection of K components such as frame parts, propellers, payload and batteries. Each of the components has it's mass m_k and inertia tensor $\mathbf{I}_k$ expressed in the frame of the respective part as $\mathbf{I}^b_k = \mathbf{I}_k + m_k\left[(\mathbf{P}_j \cdot \mathbf{P}_j)\,\mathbf{E} - \mathbf{P}_j \otimes \mathbf{P}_j\right]$, where $\mathbf{P}_j$ is the displacement vector of the part center of mass to a model center of mass, $\mathbf{E}$ is an identity matrix and $\otimes$ is a tensor product operation. The final mass is then computed as $m = \sum_{k=1}^{K} m_k$ and the inertia tensor as $\mathbf{I}^b = \sum_{k=1}^{K} \mathbf{I}^b_k$.

The Euler's kinematic equation defines the rate of the Euler angles as a function of UAV's angular velocities in $\mathscr{F}_b$. Let us have an Euler angle vector $\Phi = [\phi \;\; \theta \;\; \psi]^T$ representing the orientation of $\mathscr{F}_b$ with respect to $\mathscr{F}_e$. The kinematic equation in (6) is described in terms of a kinematic transformation matrix $\mathbf{H}$ linking the angular velocity and orientation of the vehicle as a first order ODE.

$$\dot{\Phi} = \mathbf{H}\boldsymbol{\omega}^b_{b/e} \tag{6}$$

The kinematic transformation matrix is given by equation (7). The orientation can be optionally described by a set of quaternions.

$$\mathbf{H} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{7}$$

To compute the UAV's center of mass position in $\mathscr{F}_e$, a UAV's linear velocity in $\mathscr{F}_e$ has to be derived as a function of linear velocity in $\mathscr{F}_b$. This relationship is called the kinematic equation of navigation and it is described by equation (8). Let $\mathbf{p}^e_{cm/e} = [x \;\; y \;\; z]^T$ be a position vector of the CoM expressed in $\mathscr{F}_e$ and $\mathbf{R}_{b/e}$ is the rotational matrix between BFF and the inertial NED frame usually referred to as the Direction Cosine Matrix (DCM).

$$\dot{\mathbf{p}}^e_{cm/e} = \mathbf{R}^T_{b/e}\mathbf{v}^b_{cm/e} \tag{8}$$

The position of a UAV can now be computed based on the knowledge of Euler angles and UAV's linear velocity in BFF. The resulting force $\mathbf{F}^b = [F_x \quad F_y \quad F_z]^T$ acting on the UAV CoM is decomposed into three principal components. Based on the UAV's configuration and required model fidelity, the following forces and moments acting on the body can be introduced. Below we introduce the description of the gravitational force $\mathbf{F}^b_g$, the aerodynamic force $\mathbf{F}^b_a$ and thrust/propeller force $\mathbf{F}^b_t$. As already mentioned, some forces are naturally not expressed in the BFF and need to be transformed via $\mathbf{R}_{b/e}$. The total external force acting on the vehicle is then a sum of its components as in equation (9).

$$\mathbf{F}^b = \mathbf{F}^b_g + \mathbf{F}^b_a + \mathbf{F}^b_t \tag{9}$$

The gravitational force which is naturally expressed in the NED frame's z-axis shall be transformed to BFF using the transformation matrix leading to formulation introduced in equation (10).

$$\mathbf{F}^b_g = \begin{bmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{bmatrix} \tag{10}$$

To express the final propeller force of a generalized multirotor UAV, let N be the number of propellers and R^j_{pr} the radius of propeller j.

$$\mathbf{F}^b_t = \begin{bmatrix} 0 \\ 0 \\ -\frac{4}{\pi^2}\sum_{j=1}^{N}\rho\left(R^j_{pr}\right)^4\omega_j^2 C^j_T(\mu_j,\alpha) \end{bmatrix} \tag{11}$$

The resulting thrust force can be described by equation (11), where ρ is the air density, ω_j is the propeller angular velocity and $C^j_T(\mu_j,\alpha)$ is the value of propeller thrust coefficient for a given value of advance ratio μ_j and angle of attack α. The thrust coefficient depends on the number of blades B^j_{pr}. Advance ratio is defined as a ratio of the free stream fluid velocity to the propeller speed. For a propeller j, it can be computed as introduced in equation (12), where V_a is the relative airflow magnitude.

$$\mu_j = \frac{\pi V_a}{\omega_j R^j_{pr}} \tag{12}$$

The thrust coefficient is usually determined for a set of representative conditions and subsequently interpolated so that $C^j_T(\mu_j,\alpha)$ is a twice continuous differentiable function. Assuming that the UAV is seated in a stationary wind vector field $\vec{\mathbf{W}}(x,y,z) = [W_x \quad W_y \quad W_z]^T$, the relative airflow $\mathbf{V}^b_a$ with respect to the UAV is described by equation (13) using substitutions $c_\gamma := \cos\gamma$ and $s_\gamma := \sin\gamma$ where γ is an arbitrary angle and w_x, w_y, w_z are turbulence and gust effects assumed to be zero when dealing with trajectory optimization tasks.

$$\mathbf{V}^b_a = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}^T \begin{bmatrix} W_x + w_x \\ W_y + w_y \\ W_z + w_z \end{bmatrix} - \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{13}$$

Finally the aerodynamic force can be decomposed into propeller forces generated by the interaction of the incoming airflow with the rotating propellers and the drag force, so that $\mathbf{F}_a^b = \mathbf{F}_{prop}^b + \mathbf{F}_{drag}^b$. Aerodynamic force $\mathbf{F}_{prop}^b$ is complex to quantify and can be modelled in different ways based on the level of complexity and model configuration. Using the blade properties, complex computational models such as Khan's model [12] utilizing the methodology of the Blade Element Momentum Theory (BEMT) for propeller model development, can be adopted. The drag force $\mathbf{F}_{drag}^b$ can be simplified to formulation in equation (14) where C_D is the drag coefficient computed based on a projected shape of the body to the airflow usually determined by the wind tunnel tests and A_D is the drag area which may be modeled as a constant or as a variable changing with orientation of the body with respect to airflow.

$$\mathbf{F}_{drag}^b = \frac{1}{2}\rho C_D A_D \mathbf{V}_a^b \parallel \mathbf{V}_a^b \parallel_2 \tag{14}$$

Let $\mathbf{M}^b = [M_x \quad M_y \quad M_z]^T$ be the resulting moment of all external moments acting on the multirotor UAV's CoM. The total external moment $\mathbf{M}^b$ can be decomposed into five principle components as defined in equation (15).

$$\mathbf{M}^b = \mathbf{M}_t^b + \mathbf{M}_q^b + \mathbf{M}_j^b + \mathbf{M}_g^b + \mathbf{M}_a^b \tag{15}$$

The decomposed moments contain thrust reaction torque $\mathbf{M}_t^b$ due to $\mathbf{F}_t^b$, the propeller drag couple $\mathbf{M}_q^b$, the inertia couple $\mathbf{M}_j^b$ due to rotating parts of the propulsor, the gyroscopic torque $\mathbf{M}_g^b$ and the aerodynamic moment $\mathbf{M}_a^b$. Let $\mathbf{P}_{prop}^j = \begin{bmatrix} p_x^j & p_y^j & p_z^j \end{bmatrix}^T$ be a position vector of propeller j with respect to UAV CoM in the BFF. The thrust reaction torque in equation (16) describes the generalized case for the N-propeller UAV.

$$\mathbf{M}_t^b = \begin{bmatrix} -\frac{4}{\pi^2}\sum_{j=1}^N \rho \left(R_{pr}^j\right)^4 \omega_j^2 C_T^j(\mu_j,\alpha) p_y^j \\ \frac{4}{\pi^2}\sum_{j=1}^N \rho \left(R_{pr}^j\right)^4 \omega_j^2 C_T^j(\mu_j,\alpha) p_x^j \\ 0 \end{bmatrix} \tag{16}$$

Next, the propeller thrust couples described by equation (17) are computed assuming a known power coefficient $C_P^j(\mu_j,\alpha)$ for the propeller j and its rotation orientation $O_j \in \{-1,1\}$.

$$\mathbf{M}_q^b = \begin{bmatrix} 0 \\ 0 \\ -\frac{4}{\pi^3}\sum_{j=1}^N \rho \left(R_{pr}^j\right)^5 \omega_j^2 C_P^j(\mu_j,\alpha) O_j \end{bmatrix} \tag{17}$$

Assuming a j-th propulsor's moment of inertia J_m^j, we can write the inertia couples using equation (18)

$$\mathbf{M}_j^b = \begin{bmatrix} 0 \\ 0 \\ -\sum_{j=1}^N J_m^j \dot{\omega}_j O_j \end{bmatrix} \tag{18}$$

where $\dot{\omega}_j$ is the respective propeller angular velocity. Finally, the propeller's gyroscopic torques of are defined in equation (19), where p and q are the UAV's angular rates.

$$\mathbf{M}_g^b = \begin{bmatrix} -q\sum_{j=1}^N J_m^j \omega_j O_j \\ p\sum_{j=1}^N J_m^j \omega_j O_j \\ 0 \end{bmatrix} \tag{19}$$

Aerodynamic moments $\mathbf{M}_a^b = \mathbf{M}_{prop}^b$ are assumed to be caused by the interaction of the incoming airflow with the rotating propellers and can be modeled using the BEMT model. The model can be extended to include differential equations of a DC motor controlled by the applied voltage. For j-th motor, we get the differential equations (20) and (21), where L is the motor's inductance, k_e is the

electromagnetic force constant, R is the resistance, k_m is the motor torque constant and b_m is the viscous friction constant. The equation (20) is the electrical circuit equation derived using the Kirchhoff's voltage law with input voltage V_j. The mechanical equation (21) shows that the motor torque is proportional to the armature current. The resulting change in propeller speed is further affected by the viscous friction term and the propeller drag term.

$$L_j \dot{I}_j = V_j - k_e^j \omega_j - R_j I_j \tag{20}$$

$$J_m^j \dot{\omega}_j = k_m^j I_j - b_m^j \omega_j - \frac{4}{\pi^3} \rho \left(R_{pr}^j\right)^5 \omega_j^2 C_P^j(\mu_j, \alpha) \tag{21}$$

The 6-DoF model of the generalized multirotor UAV can be augmented by the DC motor equations. In the following chapters, the motor torque will be used to compute the energy consumption of the UAV.

4. Gauss pseudospectral transcription method

The developed dynamic model can be composed to a system of ODEs $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ forming the dynamic constraint of the OCP. This chapter introduces the transcription method used to discretize the developed dynamic constraints to a set of collocation points in the time domain where they strictly enforce the set of ODEs. Pseudospectral methods employ global interpolating polynomials to approximate the state across the entire time interval. Numerical methods including pseudospectral methods require fixed time interval which is a limiting factor for many OCP formulations. Using a transformation in equation (22), the time interval $t \in [t_0, t_f]$ is mapped to a general interval $\tau \in [-1, 1]$ with free initial and final times.

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0} \tag{22}$$

To minimize error in the quadrature approximation of the integral term of the cost function or dynamic constraints discretization, respective Gauss quadrature support points are selected. The Gauss quadrature has the highest accuracy for given number of points K and is exact for all polynomials of degree $2K-1$ or less. The quadrature points are the Legendre-Gauss (LG) points [9] defined in the interior of the fixed time interval $\tau \in (-1, 1)$ as the roots of the Legendre polynomial $P_K(\tau)$ of degree K. It is given by equation (23).

$$P_K(\tau) = \frac{1}{2^K K!} \frac{\mathrm{d}^\mathrm{K}}{\mathrm{d}\tau^\mathrm{K}} \left[\left(\tau^2 - 1\right)^K\right] \tag{23}$$

Similarly, another set of support points are the Legendre-Gauss-Radau (LGR) points on interval $\tau \in [-1, 1)$ defined as the roots of polynomial $\hat{P}_K(\tau) = P_K(\tau) + P_{K-1}(\tau)$. The third set of support points applicable for pseudospectral collocation method is the Legendre-Gauss-Lobatto (LGL) points. The support points lie on the interval $\tau \in [-1, 1]$ including both boundaries, making the quadrature scheme accurate to $2K-3$ polynomial degree. The LGL points are derived as roots of the polynomial $\tilde{P}_K(\tau) = \left(1 - \tau^2\right) \dot{P}_{K-1}(\tau)$. Pseudospectral methods exploit rapid convergence rate of the quadrature approximation and are able to provide better approximations with fewer number of support points. Having obtained a possible choices of suitable support points, the global polynomial transcription can be obtained using a Lagrange polynomial approximation. The choice of Lagrange polynomials has two benefits when used with quadrature points in a transcription scheme. The function approximation is equal to the true function at the collocation points $x(\tau_j) = X(\tau_j)$ and the unique set of support points tends to bypass the so called Runge phenomenon which leads to higher approximation errors near the boundaries for uniformly spaced support points. Let us define a basis of $K+1$ Lagrange interpolating polynomials defined by the equation (24).

$$\mathscr{L}_i(\tau) = \prod_{j=0,\, j \neq i}^{K} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{24}$$

The state and control vectors are globally approximated in the fixed time domain defined by equation (22) using the Lagrange polynomial basis functions as in equation (25). The control vector polynomial

approximation is presented just for consistency purposes since it is used at the collocation points only.

$$\begin{aligned}\mathbf{x}(\tau) &\approx \mathbf{X}(\tau) = \sum_{i=0}^{K} \mathscr{L}_i(\tau)\mathbf{x}(\tau_i) \\ \mathbf{u}(\tau) &\approx \mathbf{U}(\tau) = \sum_{i=1}^{K} \mathscr{L}_i(\tau)\mathbf{u}(\tau_i)\end{aligned} \tag{25}$$

As will be shown, the function will be evaluated only at the set of support points leaving global approximation of the control vector unnecessary. The support points $M = \{\tau_1, \ldots, \tau_K\}$ (called collocation points) are extended to $M_0 = \{-1, \tau_1, \ldots, \tau_K\}$ to be of size $K+1$. The discretization point $\mathbf{X}_f$ is collocated at the boundary $\tau_f = 1$. Due to boundary constraints of the original OCP, the final state must be included as a set of discrete variables, even they are not part of the global polynomial state approximation. The initial and final states can be linked using a Gauss quadrature. The final state can be integrated from the initial state as shown in equation (26).

$$\mathbf{x}(\tau_f) = \mathbf{x}(\tau_0) + \int_{-1}^{1} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\mathrm{d}\tau \tag{26}$$

Reformulating using the polynomial approximation and a Gauss quadrature, the integration (26) transforms to a set of equality constraints forming equation (27) with w_k being the quadrature weights.

$$\mathbf{X}(\tau_f) - \mathbf{X}(\tau_0) - \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k, t_0, t_f) = \mathbf{0} \tag{27}$$

The mathematical properties of the Gauss quadrature allow an accurate approximation of the final state. In the Lagrange polynomial reformulation, the dynamic constraints in the time domain interior can be satisfied using the properties of the Lagrange polynomial approximation. The state derivative at LG points for $k = 1, \ldots, K$ gives

$$\dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^{K} \dot{\mathscr{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=0}^{K} D_{ki}\mathbf{X}(\tau_k), \tag{28}$$

where $D_{ki} \in \mathbb{R}^{K\times(K+1)}$ is called the LG differentiation matrix. It can be determined analytically during the construction of an NLP problem. The dynamic constraints can now be composed to a set of equality constraints on a set M. Using the derivative approximation (28), we get the equality constraints (29) enforcing the dynamics at the collocation points.

$$\sum_{i=0}^{K} D_{ki}\mathbf{X}(\tau_k) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k, t_0, t_f) = \mathbf{0} \tag{29}$$

Finally, the Gauss quadrature is utilized to integrate the path cost leading to the final form of the objective function as given by equation (30).

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k g(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k, t_0, t_f) \tag{30}$$

With minor changes in the handling of the boundary points, the transcription using LGL or LGR points can be performed. The transcription methods allowed discretized formulation of a general OCP. The obtained NLP problem may include large number of variables and constraints. The problem complexity also extends by presence of equality constraints emerging from dynamic constraints. The set of equality constraints in the NLP leads to dimension reduction which rules out many proven of methods which are successfully used for nonlinear function minimization such as class of evolutionary algorithms including the genetic algorithm or the particle swarm optimization which are popular choice when solving global optimization problems.

The set of variables defining the OCP is composed as seen in equation (31), where, at time τ_k, the control vector is $\mathbf{u}(\tau_k) = [u_1(\tau_k), u_2(\tau_k), \ldots, u_m(\tau_k)]$ and the state vector is $\mathbf{x}(\tau_k) = [x_1(\tau_k), x_2(\tau_k), \ldots, x_n(\tau_k)]$. The variables also include the initial and final times.

$$\mathbf{z}^T = [u_1(\tau_0), u_2(\tau_0), \ldots, x_1(\tau_1), x_2(\tau_1), \ldots, u_m(\tau_K), x_n(\tau_K), t_0, t_f, p_1, \ldots, p_l] \tag{31}$$

Apart from the initial and final time, the set of l algebraic optimization parameters can be optionally defined as $\mathbf{p} = [p_1, p_2, \ldots, p_l]$. The general formulation of the NLP problem can be written as shown in equation (32).

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^w} \quad & J(\mathbf{z}) \\ \text{subj.to} \quad & \mathbf{c}_L \leq \mathbf{c}(\mathbf{z}) \leq \mathbf{c}_U \\ & \mathbf{z}_L \leq \mathbf{z} \leq \mathbf{z}_U \end{aligned} \tag{32}$$

The vector $\mathbf{z} \in \mathbb{R}^w$ contains the optimization variables with lower and upper bounds $\mathbf{z}_L \in (\mathbb{R} \cup \{-\infty\})^w$ and $\mathbf{z}_U \in (\mathbb{R} \cup \{\infty\})^w$. In context of OCP with no additional parameters, the number of variables can be computed as $w = 2 + K(n+m)$, where n is the number of states, m is the number of control variables, K is the number of time steps and the constant represents the free initial and final time variables. The objective function $J(\mathbf{z}) : \mathbb{R}^w \longrightarrow \mathbb{R}$ and the constraint function $\mathbf{c}(\mathbf{z}) : \mathbb{R}^w \longrightarrow \mathbb{R}^v$ can be generally nonlinear convex or non-convex functions at least twice continuously differentiable.

The state-of-the-art class of algorithm for solving general NLP problems are called the interior point (barrier) methods. The interior point methods define the auxiliary objective modeled by equation (33), where sets I_L and I_U are defined as $I_L = \{i : z_L^i \neq -\infty\}$ and $I_U = \{i : z_U^i \neq \infty\}$.

$$\min_{\mathbf{z}\in\mathbb{R}^w} \Upsilon_\eta(\mathbf{z}) = J(\mathbf{z}) - \eta \sum_{i\in I_L} \ln(z^i - z_L^i) - \eta \sum_{i\in I_U} \ln(z_U^i - z^i) \tag{33}$$

The finite bound constraints are therefore replaced by the logarithmic barrier terms. In case of barrier parameter $\eta > 0$, the auxiliary objective function goes to infinity if any variable approaches it's finite lower or upper bound. A popular implementation of the interior point method is the freely available IPOPT algorithm [11], the primal-dual interior-point algorithm with a filter line-search method, which was in the presented case used to solve the trajectory optimization problem.

There are many more transcription method options available. Common integration schemes such as trapezoidal rule or Simpson's rule can be also used to formulate the equality constraints. Gauss-pseudospectral methods however, benefit from using the Gauss quadrature on a special set of support points and faster convergence. Software solutions such as GPOPS-II [15] or GEKKO [16] implement the pseudospectral method for transcription. Trapezoidal collocation or Runge-Kutta methods transcription are utilized by FALCON.m toolbox [7] developed at Technical University of Munich. To speed up the convergence, the above mentioned software solutions utilize Jacobian and Hessian sparsity patterns of the NLP and automatic differentiation tools. We used the ecosystem of JAX: Autograd [8] to compute objective function gradient, Jacobian of the constraints and Hessian of the Lagrangian function.

5. Implementation and examples

The theoretical background for multirotor UAV modelling and trajectory optimization techniques were outlined. This chapter introduces the energy objective function 5.1, the modelling of terrain and no-fly zone constraints 5.2 and wind profile power law 5.3 specifying the vertical change in wind speed which form feasible options for OCP extension in form of inequality constraints. The chapter also provides a 2D OCP example with circular constraints in a wind field 5.4 following the analytically known Dubins path and a 3D OCP case with set of operational and environmental constraints utilizing the fully nonlinear UAV model.

5.1 Minimum energy objective function

A total energy consumed by the UAV at time t can be derived using the motor torque $\tau(t) = k_m I(t)$ and the angular speed $\omega(t)$ forming the instantaneous power which is then integrated over the time domain to compute the consumed energy [5]. The objective function (34) is derived using the mechanical equation (21) of the DC motor.

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{N} \left(J_m^j \dot{\omega}_j + \frac{4}{\pi^3} \rho \left[R_{pr}^j \right]^5 \omega_j^2 C_P^j(\mu_j, \alpha) + b_m^j \omega_j \right) \omega_j \, \mathrm{dt} \tag{34}$$

The objective function assumes the motor angular speed derivatives $\dot{\omega}_j$ to be the control inputs and motor angular speeds ω_j are assumed to be included in the state vector.

5.2 Terrain and no-fly zone constraints

A terrain constraints were implemented using the Inverse Distance Weighting (IDW) interpolating function $s(\mathbf{p})$. This ensures, that the planned trajectory does not intersect the terrain profile. The terrain constraint is formulated by inequality constraint as in equation (35).

$$z(\tau_k) - s(x(\tau_k), y(\tau_k)) \geq 0, \qquad k \in \{1, \ldots, K\} \tag{35}$$

where $x(\tau_k)$, $y(\tau_k)$ and $z(\tau_k)$ are the collocated position components of the UAV. The terrain surface equation may be modeled by selection of IDW parameter p. Let us consider a rectangular domain $D = [a,b] \times [c,d]$. The multivariate interpolation of the terrain is some function $s(\mathbf{p}) : \mathbf{p} \in D$, where $\mathbf{p} = [x,y]$ represents a location in the domain D. Due to computational efficiency, the Shapard's interpolation was considered as a suitable continuous and differentiable approximation. Search structures like kd-trees are not needed for classical IDW. Given a set of samples $s(\mathbf{p}_i) = z_i$ for $i = 1, 2, \ldots, N$, the interpolating function takes form of equation (36) for $\mathbf{p}$ such that $d(\mathbf{p}, \mathbf{p}_i) \neq 0$ for all i.

$$s(\mathbf{p}) = \frac{\sum_{i=1}^{N} w_i(\mathbf{p}) z_i}{\sum_{i=1}^{N} w_i(\mathbf{p})} \tag{36}$$

In other case, $s(\mathbf{p}) = z_i$ if $\mathbf{p} = \mathbf{p}_i$. The set of weights are given by equation (37) where p is a positive real number called a power parameter which affects shape of the interpolating function.

$$w_i(\mathbf{p}) = \frac{1}{d(\mathbf{p}, \mathbf{p}_i)^p} \tag{37}$$

A no-fly zone is a territory or area, over which the UAV is not permitted to fly. For practical purposes, this work only considers circular no-fly zones. Let us assume zone $i \in \{1, \ldots, U\}$, where U is the number of no-fly zones. The center $[x_i, y_i]$ and radius L_i defines the zone path constraint i given the equation (38), where $x(\tau_k)$ and $y(\tau_k)$ are the position components of the UAV collocated in the discretized time domain.

$$(x_i - x(\tau_k))^2 + (y_i - y(\tau_k))^2 \geq L_i^2, \qquad k \in \{1, \ldots, K\} \tag{38}$$

Since the constraint has to be satisfied at all time points, a single no-fly zone leads to K inequality constraints.

5.3 Wind profile power law

The change of altitude usually affects the nominal wind speed. The so called wind profile power law [6] was used for the vertical extrapolation of the nominal wind speed W in the atmospheric surface layer. In order to estimate the wind speed at certain height z, the power law given by the equation (39) is used to extrapolate $W(z)$ using known wind speed $W(z_A)$ at reference height z_A.

$$W(z) = W(z_A) \left(\frac{z}{z_A} \right)^n \tag{39}$$

The exponent n is an empirically derived coefficient that varies with the stability of the atmosphere. In neutral stability conditions, the value $n = 0.143$ was used for open land surfaces and value $n = 0.110$ was used for over open water surfaces. The applications of wind profile power law include wind power assessments or atmospheric pollution dispersion models.

5.4 No-fly zone avoidance using simplified kinematic model

In the first example, a construction of OCP will be investigated in a simplified fashion. A simple 2D kinematic model is used and the notion of no-fly zones utilized. In this simple scenario, the model is given by ODE system (40), where the parameter $\gamma \in [0,1]$ is a weighting factor for the wind field effect on the total speed.

$$\begin{aligned} \dot{x} &= V\sin\psi + \gamma W_x(x,y) \\ \dot{y} &= V\cos\psi + \gamma W_y(x,y) \end{aligned} \tag{40}$$

The control vector of the OCP includes the UAV speed and heading $\mathbf{u} = [V,\psi]^T$. The components of a stationary wind field are modeled as $W_x = W\cos\alpha_W$ and $W_y = W\sin\alpha_W$. The quadratic objective function has no actual physical meaning. The problem may thus be described as minimum control effort problem. Let us assume a general UAV with starting position $\mathbf{x}_0 = [1000, 1000]$ m. The mission target is located at $\mathbf{x}_f = [5000, 5000]$ m. The speed limit of the UAV in zero wind condition is specified with $V_{min} = 0$ m$\cdot$s^{-1} and $V_{max} = 10$ m$\cdot$s^{-1}. The UAV is affected by a wind field with nominal speed $W = 5$ m$\cdot$s^{-1}, direction $\alpha_W = \pi/2$ and the weighting factor $\gamma = 1$. The designed trajectory should avoid two no-fly zones with centers $C_1 = [2500, 2500]$ and $C_2 = [4300, 4300]$ and radii $L_1 = 800$ m and $L_2 = 500$ m. The no-fly zones are modeled using the inequality constraints given by equation (38). The OCP may be formulated as shown in (41).

$$\begin{aligned} &\min_{\mathbf{u}\in U} \int_0^{t_f} V^2(t)\,\mathrm{dt}, \quad \mathbf{u} = [\mathrm{V(t)}, \psi(\mathrm{t}), \mathrm{t_f}]^{\mathrm{T}} && \text{(objective)} \\ &\text{subj.to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), V(t), \psi(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 && \text{(dynamics)} \\ &\mathbf{x}(t_f) = \mathbf{x}_T && \text{(final state constraint)} \\ &(x-2500)^2 + (y-2500)^2 \geq L_1^2 && \text{(no-fly zone constraint)} \\ &(x-4300)^2 + (y-4300)^2 \geq L_2^2 && \text{(no-fly zone constraint)} \\ &V_{min} < V(t) < V_{max} && \text{(velocity constraint)} \\ &|\psi(t)| < \pi && \text{(yaw angle constraint)} \end{aligned} \tag{41}$$

The problem was solved using Legendre-Gauss-Lobatto collocation scheme using $K = 100$ discretization points. The flight time of the optimized trajectory was computed $t_f = 1255.7$ s. The trajectory successfully avoids two no-fly zones and commands lower velocity in a region where tailwind increases the total velocity in the direction of flight. Figure 3 shows the optimized trajectory and Figure 4 shows the UAV's state and control profiles. The trajectory resembles the analytical results of the shortest path 2D problem known as a Dubins path proved by application of Pontryagin's maximum principle.

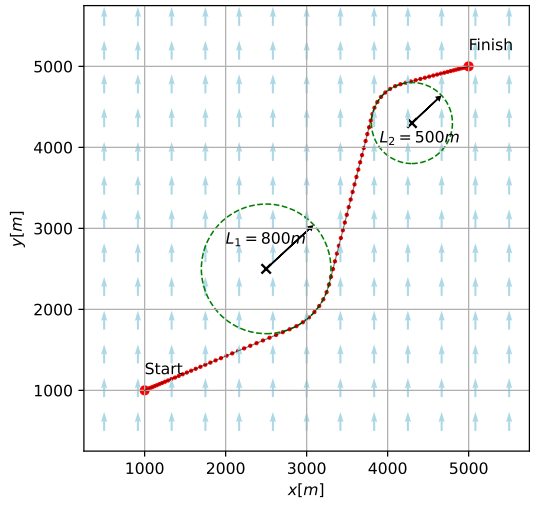


Figure 3 – A 2D trajectory avoiding no-fly zones.

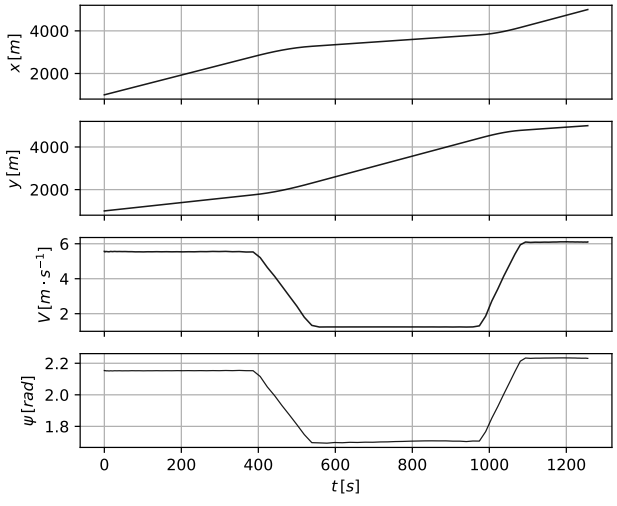


Figure 4 – State and control vector profiles.

5.5 Quadcopter minimum energy problem considering flight battery state of charge

Considering fully nonlinear 6-DoF EoM, the following example will explore a quadcopter's energy optimal trajectory with respect to a cost function presented in subsection 5.1. In some cases, mission requirements may exceed capabilities of a considered vehicle, especially increased battery capacity increases possible flight time, but can be unnecessary for shorter missions where objectives such as maximum payload may be more relevant. Increased vehicle mass due to battery increases the battery capacity based on battery energy density ρ_{bat}. The OCP can now facilitate mass as an optimization parameter. Computation of the battery State of Charge (SoC) is simplified to be $SoC = 100(1 - E_c(t_f)/E_{bat})$, where E_{bat} is total battery energy at an initial time. The quadcopter is required to arrive to a destination with at least SoC_{req} charged battery. Mass of the vehicle has to be now optimized to satisfy this constraint and minimize total consumed energy.

$$
\begin{aligned}
&\min_{\mathbf{u}\in U} E_c(t_f), \quad \mathbf{u} = [u_1(t), u_2(t), u_3(t), u_4(t), t_f, m]^T && \text{(objective)}\\
&\text{subj.to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u_1(t), u_2(t), u_3(t), u_4(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 && \text{(dynamics)}\\
&E_{bat} = E_0 + \rho_{bat}\,(m - m_0) && \text{(battery energy)}\\
&m > m_0 && \text{(minimal mass constraint)}\\
&\mathbf{x}(t_f) = \mathbf{x}_T && \text{(final state constraint)}\\
&\Omega_{min} < \omega_1(t), \omega_2(t), \omega_3(t), \omega_4(t) < \Omega_{max} && \text{(angular speed constraint)}\\
&U_{min} < u_1(t), u_2(t), u_3(t), u_4(t) < U_{max} && \text{(control input constraint)}\\
&h_{min} < h < h_{max} && \text{(altitude constraint)}\\
&SoC > SoC_{req} && \text{(minimum } SoC \text{ requirement)}\\
&|\phi(t)| < \phi_{max} && \text{(roll angle constraint)}\\
&|\theta(t)| < \theta_{max} && \text{(pitch angle constraint)}\\
&|\psi(t)| < \pi && \text{(yaw angle constraint)}
\end{aligned}
\tag{42}
$$

For this example given by the OCP (42), we assume the UAV can be equipped with a number of 18650 Li-ion Battery Cells (2500 mAh) with weight 45 g and energy of 9.25 Wh. Thus we can compute the capacity density to be $e_{bat} = 200\,000\ \mathrm{As \cdot kg^{-1}}$ or energy density of $\rho_{bat} = 205.5\ \mathrm{Wh \cdot kg^{-1}}$. The initial mass of the UAV is assumed to be $m_0 = 1.7$ kg with initial energy of single battery cell $E_0 = 33$ kJ. The physical properties of the quadcopter are adopted from [4]. The drone mission has following specifications:

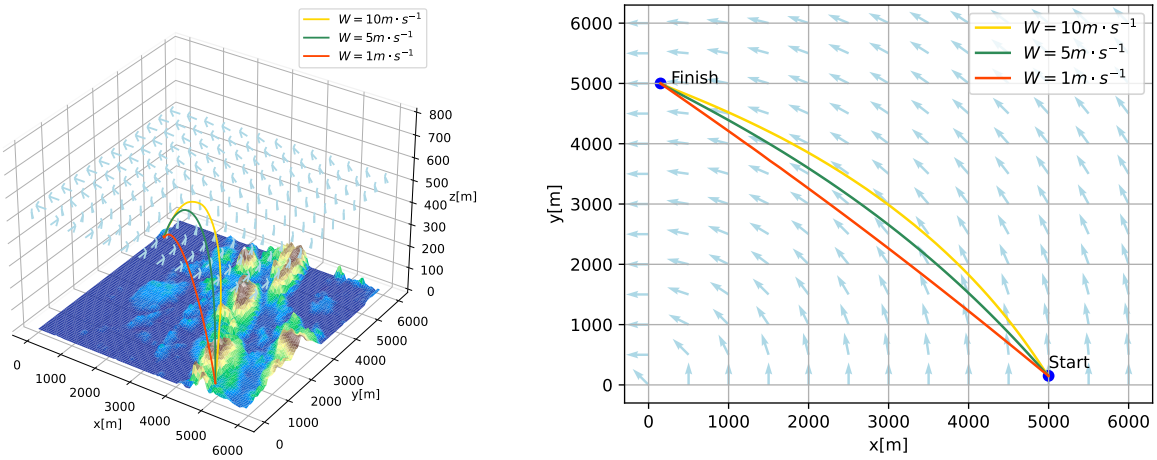
1. The UAV starts in hover mode at position $\mathbf{x}_0 = [5000, 150, 50]$ m over land.
2. The target for inspection is located at offshore location $\mathbf{x}_T = [150, 5000, 10]$ m.
3. The UAV rotors have a maximum angular speeds of $\Omega_{max} = 800\ \mathrm{rad \cdot s^{-1}}$.
4. Angular speeds lower than $\Omega_{min} = 400\ \mathrm{rad \cdot s^{-1}}$ are considered to be insufficient for UAV flight.
5. The control inputs are constrained to $U_{min} = -100\ \mathrm{rad \cdot s^{-2}}$, $U_{max} = 100\ \mathrm{rad \cdot s^{-2}}$.
6. A constraints are given for maximum pitch $\theta_{max} = \frac{\pi}{6}$ rad and roll $\phi_{max} = \frac{\pi}{6}$ rad angles of the UAV.
7. The mission is constrained by maximum $h_{max} = 400$ m and minimum $h_{min} = 10$ m allowed altitude.
8. The UAV is required to arrive with at least 50% charged battery, thus $SoC_{req} = 0.5$.

During the mission, the UAV is under influence of wind field given by equation (43) with constants $C_1 = C_2 = W$ specifying nominal wind speed.

$$
\vec{\mathbf{W}}(x, y) = \begin{bmatrix} \dfrac{-C_1 y}{\sqrt{x^2 + y^2}} & \dfrac{C_2 x}{\sqrt{x^2 + y^2}} \end{bmatrix}^T, \quad C_1, C_2 \in \mathbb{R} \tag{43}
$$

The wind profile power law from subsection 5.3 was accounted for with $z_A = 40\ m$ and $n = 0.110$. The results are given for nominal wind speeds $W = 10\ \mathrm{m \cdot s^{-1}}$, $W = 5\ \mathrm{m \cdot s^{-1}}$ and $W = 1\ \mathrm{m \cdot s^{-1}}$. All solutions

were computed using 280 time steps and an optimal solution was found for all the considered cases. For wind speeds of $W = 10\ \mathrm{m \cdot s^{-1}}$, the total consumed energy was $E_{opt} = 35.14\ \mathrm{kJ}$ with time of flight $t_f = 249.83\ \mathrm{s}$. The mass of the UAV was increased to $m = 1.821\ \mathrm{kg}$ to satisfy state of charge requirements at the destination. Similarly, wind speed $W = 5\ \mathrm{m \cdot s^{-1}}$ yields $E_{opt} = 44.26\ \mathrm{kJ}$, $t_f = 295.47\ \mathrm{s}$ and $m = 1.856\ \mathrm{kg}$. With wind conditions equal to $W = 1\ \mathrm{m \cdot s^{-1}}$, optimal solution of $E_{opt} = 55.57\ \mathrm{kJ}$, $t_f = 342.37\ \mathrm{s}$ and $m = 1.908\ \mathrm{kg}$ was found. Figure 5 shows a 3D plot of all trajectories and their projection to a 2D plot with a wind vector field. It can be seen that with an increasing nominal wind speed, the trajectories increasingly follow the field. Figure 6 shows the comparison of all three selected cases



(a) (b)

Figure 5 – Obtained trajectories in a 3D plot (a) and 2D projection (b) with variety of wind conditions.

with respective velocity profile and battery state of charge. It can be seen that the battery mass is optimized such that the battery is discharged exactly to $50\ \%$ at the endpoint to satisfy the mission's operational constraint. Control inputs in Figure 7 are bound to the propeller angular velocities as $\dot{\omega}_j = u_j,\ \forall j \in 1,\ldots,N$. To solve the trajectory optimization problem, it took IPOPT 1166, 742 and 324

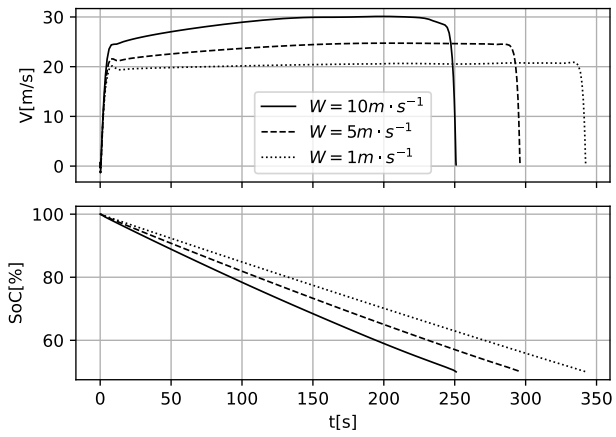


Figure 6 – The velocity and SoC profiles.

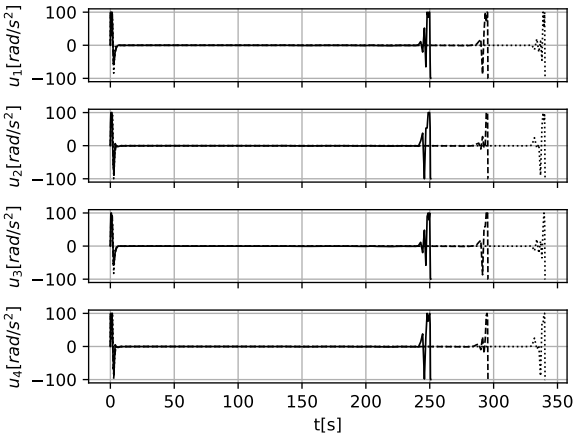


Figure 7 – Auxiliary control inputs.

iterations respectively for cases $W = 10\ \mathrm{m \cdot s^{-1}}$, $W = 5\ \mathrm{m \cdot s^{-1}}$ and $W = 1\ \mathrm{m \cdot s^{-1}}$. The convergence of IPOPT with regards to the objective function value is shown in Figure 8. Both the objective tolerance and the relative inequality and equality constraint tolerances were set to $1\mathrm{e}^{-6}$. Figures 9 shows the individual position and velocity components, angular rates and Euler angles.

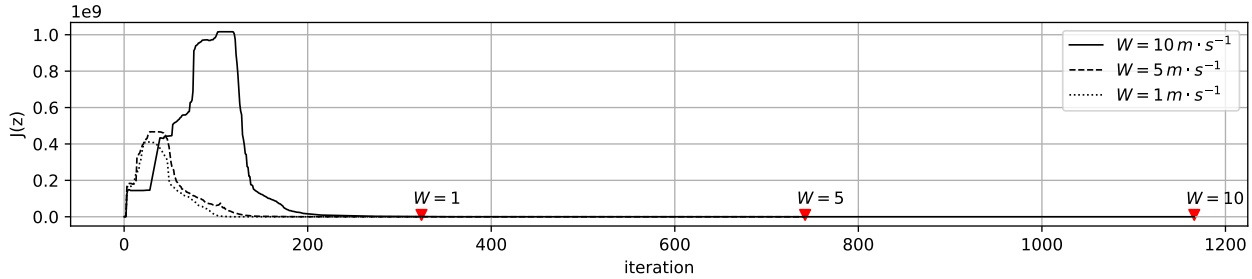


Figure 8 – Objective value convergence with respect to IPOPT iterations.

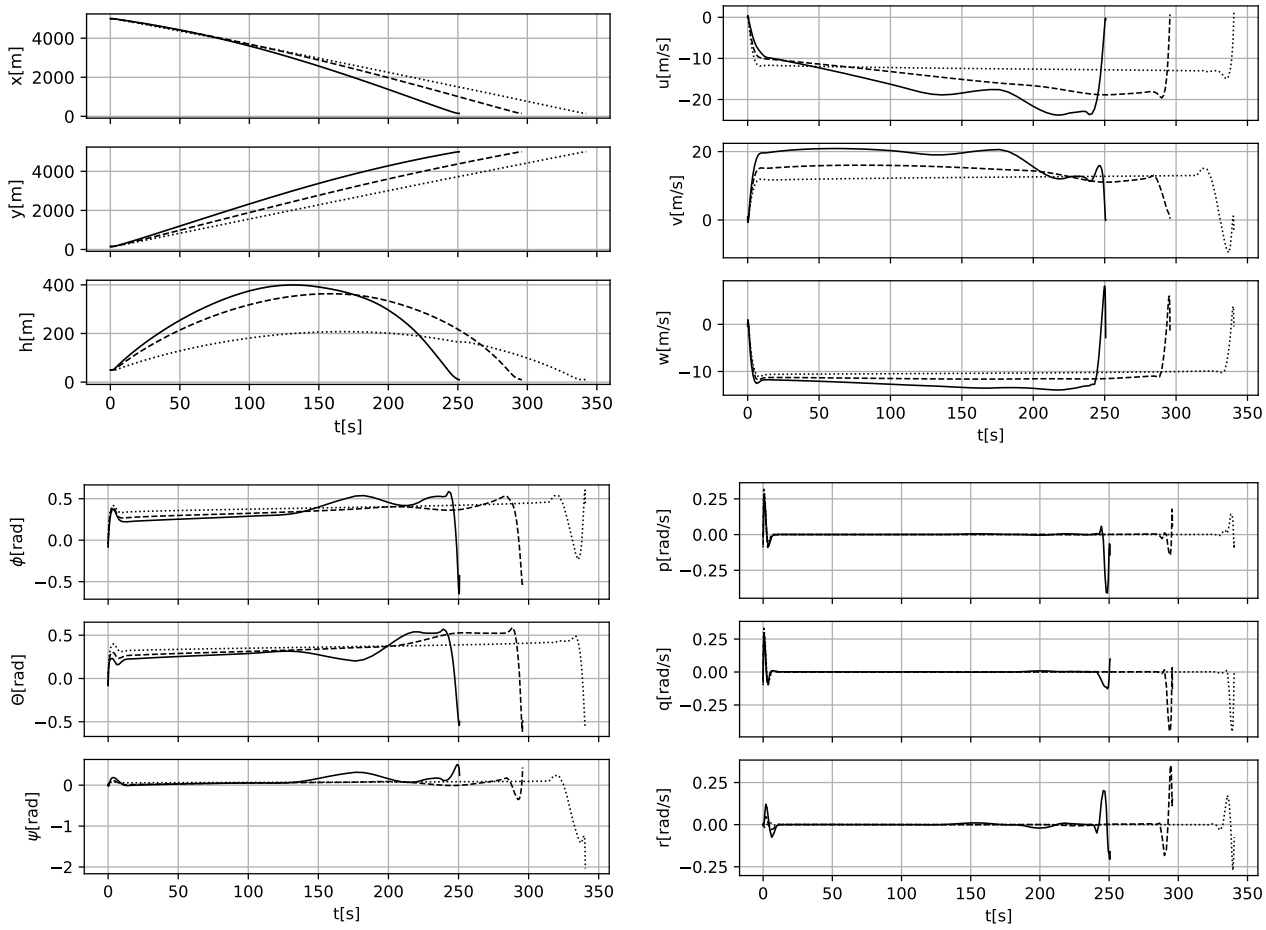


Figure 9 – The optimized position, velocities, orientation and angular velocities of the quadcopter.

6. Conclusion

A framework for UAV trajectory optimization using direct transcription methodology was presented. The work has introduced generalized high-fidelity model for multirotor UAVs so that fits various commonly used drone configurations and presented an interpretable objective function addressing total energy consumption of the vehicle and possibilities to model terrain and no-fly zone constraints. Since wind conditions strongly affect the capabilities and flight times of UAVs, wind fields are modeled and accounted for in the optimization problem definition. The first presented example shows a trajectory computed using simplified kinematic model with two active no-fly zone constraints. The second presented example shows optimization using a fully nonlinear quadcopter model in a wind field while posing constraint on the final battery state of charge. An optimal battery mass is subsequently found to meet the mission requirements. Both cases are solved using the state-of-the-art Gauss pseudospectral methods transcribing the OCP to a parameter optimization problem on a set of discrete points known as the collocation points. Utilizing the terrain constraint was achieved only

when using restricted number of points and it is thus not included in the presented example but shows promising future direction for extending the OCP applicability.

7. Acknowledgments

This project has received funding from the ECSEL Joint Undertaking (JU) for the COMP4DRONES project under grant agreement No 826610.

8. Contact Author Email Address

For more information regarding this research, contact Ing. Jiří Novák at email: inovak@fit.vutbr.cz

9. Copyright Statement

References

[1] Betts J. *Practical methods for optimal control and estimation using nonlinear programming*. 2st edition, Society for Industrial and Applied Mathematics, 2010.

[2] Bertsekas D.P. *Dynamic programming and optimal control*. 1st edition, Athena Scientific, 2000.

[3] Cook M.V. *Flight dynamics principles*. 2st edition, Butterworth-Heinemann, 2008.

[4] Massé C, Gougeon O, Nguyen D and Saussié D. Modeling and Control of a Quadcopter Flying in a Wind Field: A Comparison Between LQR and Structured $\mathscr{H}_\infty$ Control Techniques. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dallas (TX), USA, Vol. 1, 978-1-5386-1354-2, pp 1408-1417, 2018.

[5] Yacef F, Rizoug N, Degaa L and Hamerlain M. Energy-Efficiency Path Planning for Quadrotor UAV Under Wind Conditions. *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic, Vol. 1, 978-1-7281-5954-6, pp 1133-1138, 2020.

[6] Emeis S and Turk M. omparison of Logarithmic Wind Profiles and Power Law Wind Profiles and their Applicability for Offshore Wind Profiles. *Wind Energy: Proceedings of the Euromech Colloquium*, Springer, Berlin, Heidelberg, Vol. 1, 978-3-540-33865-9, pp 61-64, 2007.

[7] Rieck M, Bittner M, Grüter B, Diepolder J, Piprek P, Göttlicher Ch, Schwaiger F, Hosseini B, Schweighofer F, Akman T and Holzapfel F. FALCON.m User Guide. *http://www.falcon-m.com*,Institute of Flight System Dynamics, Technical University of Munich, 2022.

[8] Bradbury J, Frostig R, Hawkins P, Johnson J.M, Leary Ch, Maclaurin D, Necula G, Paszke A, VanderPlas J, Wanderman-Milne S and Zhang Q. JAX: composable transformations of Python+NumPy programs. *http://github.com/google/jax*, 2018.

[9] Garg D, Patterson M, Hager W, Rao A, Benson D and Huntington G. An overview of three pseudospectral methods for the numerical solution of optimal control problems. *Advances in the Astronautical Sciences*, Vol. 135, No. 1, pp 1-11, 2009.

[10] Benson D. *A Gauss pseudospectral transcription for optimal control*. Massachusetts Institute of Technology, 2005.

[11] Wächter A and Biegler L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, Vol. 106, No. 1, pp 25-57, 2006.

[12] Khan W. and Nahon M. A Propeller Model for General Forward Flight Conditions. *International Journal of Intelligent Unmanned Systems*, Vol. 3, No. 2/3, pp 72-92, 2015.

[13] Gill P, Murray W and Saunders M. SNOPT: An SQP Algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, Vol. 12, No. 1, pp 979-1006, 2002.

[14] Büskens Ch and Wassel D. The ESA NLP Solver WORHP. *Springer New York*, Vol. 73, No. 1, pp 85-110, 2013.

[15] Patterson M. and Rao A. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, pp 1-37, 2014.

[16] Beal L, Hill D, Martin R. and Hedengren J. GEKKO Optimization Suite. *Processes*, Vol. 6, No. 8, pp 106, 2018.