IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

# Wildfire Fighting by Unmanned Aerial System Exploiting its Time-Varying Mass

Diego A. Saikin[1], Tomas Baca[1], Martin Gurtner[1], and Martin Saska[1]

***Abstract*— This paper presents an approach for accurately dropping a relatively large amount of fire retardant, water or some other extinguishing agent onto a wildfire from an autonomous unmanned aerial vehicle (UAV), in close proximity to the epicenter of the fire. The proposed approach involves a *risky* maneuver outside of the safe flight envelope of the UAV. This maneuver exploits the expected weight reduction resulting from the release of the payload, enabling the UAV to recover without impacting the terrain. The UAV is tilted to high pitch angles, at which the thrust may be pointed almost horizontally. The vehicle can therefore achieve higher horizontal speeds than would be allowed by conventional motion planners. This high speed allows the UAV to significantly reduce the time spent close to the fire. As a result, the overall high heat exposure is reduced, and the payload can be dropped closer to the target, minimizing its dispersion. A constrained optimal control problem (OCP) is solved taking into account environmental parameters such as wind and terrain gradients, as well as various payload releasing mechanisms. The proposed approach was verified in simulations and in real experiments. Emphasis was put on the real time recalculation of the solution, which will enable future adaptation into a model predictive controller (MPC) scheme.**

## I. Introduction

Wildfires, mainly caused by humans, have been on the rise in recent decades [1]. According to [2], global warming is suspected to have contributed to the intensity and the frequency of these fires by making forests drier and more likely to burn. This trend is not expected to change in the years to come. Reaching a wildfire early is crucial for increasing the probability that a small and manageable fire will not become large and unmanageable. The key advantage of aircraft in wildfire fighting is their speed and their ability to access remote or otherwise difficult-to-reach fires. At the same time, present-day aerial firefighting is expensive, hazardous for the flight crews, and suffers from many limitations such as inability to operate at night.

Autonomous UAVs are potentially more efficient in terms of safety, cost and payload (there is no pilot and there is no cockpit). In addition, UAVs are not restricted from operating at night time or in bad weather. UAVs have recently begun to be employed in fighting wildfires, but are currently only used for *secondary* tasks such as mapping fires, helping in coordinating ground task forces, detecting fires and surveillance. For example, the use of UAVs to ignite prescribed fires has been proposed in [3]. The authors of [4] proposed using swarms of multicopters to hold a long hose and pass over obstacles to reach fire sources. In [5] and [6], UAV and mixed UAV-UGV (*Unmanned Ground Vehicle*) swarms are employed for monitoring wildfires. A broad analysis of possible uses of UAVs in firefighting tasks is provided in [7] but there are no examples of the direct use of UAVs to suppress fires.

Fig. 1: Left – A snapshot from the experimental verification of the proposed system. Right – Motivation for the work, presented here.

The authors of [8] proposed the idea of using fire extinguishing balls as the extinguishing agent, and a system of UAV swarms that would deploy them is presented in [9]. However, besides promising results regarding the effectiveness of these balls as extinguishing agents—which has provided a part of the motivation for our work presented here—no work regarding the maneuvers for accurately dropping them on a fire, and the constrains that affect them was made. A UAV that managed to autonomously gather water and drop it above a target is presented in [10]. However, there is no mention of considerations such as minimizing the heat exposure, the execution time, the dropping distance from the fire, or the inverse relations between these factors.

This paper goes beyond these works and presents a novel method to enable UAVs to optimally and autonomously perform the main task of a wildfire fighting mission—to extinguish the flames.

### A. Motivation and state-of-the-art

The main objective of the proposed approach is to make the UAV deliver the fire extinguishing agent precisely and quickly, and to release it as close as possible to the location of the fire. If the water or fire retardant is dropped from high above a wildfire, it will evaporate or disperse before reaching the target. However, flying too close to the flames,

This work was supported by CTU grant no. SGS17/187/OHK3/3T/13, by the Grant Agency of the Czech Republic under grant no. 17-16900Y, and by project CZ.02.1.01/0.0/0.0/16019/0000765 Research Center for Informatics.

[1]The authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague 6, Czech Republic. diegosaikin@gmail.com, {tomas.baca|gurtnmar|martin.saska}@fel.cvut.cz.

will expose the aircraft to high temperatures. Unless the exposure is for a very brief period of time, there will be a risk of damage. The exposure time can be minimized by flying fast enough. However, there is an inverse relation between the weight of the payload and the maximum achievable speed. There is therefore a trade-off between the amount of extinguishing agent that can be carried, and the proximity to the fire at which it can be dropped. This trade-off is inherent to rotary wing aircraft which must tilt their body, in order to move forward. They have to dedicate a portion of their available thrust to accelerate horizontally and overcome the aerodynamic drag. This tilt reduces the available thrust in the vertical direction, which is what maintains the multicopter in the air. The translational motion of a multicopter in inertial reference frame $\mathbb{I}$ is described by

$$\dot{\mathbf{v}} = \frac{1}{m}\left(\boldsymbol{R}_{\mathbb{I}}^{\mathbb{B}} f_{\mathrm{th}} \mathbf{e}_3 - \mathbf{f}_{\mathrm{D}}(\mathbf{v})\right) - g\mathbf{e}_3, \tag{1}$$

where $\boldsymbol{R}_{\mathbb{I}}^{\mathbb{B}} \in SO(3)$ transforms the force exerted by the UAV motors $f_{\mathrm{th}}$ from body frame $\mathbb{B}$ to $\mathbb{I}$. Parameter $g$ is the gravity acceleration, $\mathbf{e}$ is the standard basis, $m$ is the total system mass, and $\mathbf{f}_{\mathrm{D}}(\mathbf{v})$ is the aerodynamic drag force, given by

$$\mathbf{f}_{\mathrm{D}}(\mathbf{v}) = \rho C_{\mathrm{D}} A \|\mathbf{v}_{\mathrm{AS}}\|^2 \hat{\mathbf{v}}_{\mathrm{AS}}. \tag{2}$$

In (2), $C_{\mathrm{D}}$ is the aerodynamic drag coefficient and $A$ is the cross-section area of the UAV, both of which are assumed to be constant scalars. Constant $\rho$ is the air density, $\mathbf{v}_{\mathrm{AS}}$ is the air speed, which is defined by $\mathbf{v} - \mathbf{v}_{\mathrm{wind}}$ (i.e., the difference between the UAV and the wind velocities), and $\hat{\mathbf{v}}_{\mathrm{AS}}$ is a unit vector pointing in the direction of $\mathbf{v}_{\mathrm{AS}}$.

The maximum angle $\overline{\theta}_{\mathrm{nd}}$, at which a UAV can tilt while still maintaining the ability to overcome its own weight, is given by

$$\overline{\theta}_{\mathrm{nd}} = \arccos\left(\frac{mg}{\overline{f}_{\mathrm{th}}}\right), \tag{3}$$

where $\overline{f}_{\mathrm{th}}$ is the maximum available thrust. The maximum achievable horizontal speed $\overline{v}_{\mathrm{nd}}$, for a given mass and available thrust, is given by

$$\overline{v}_{\mathrm{nd}} = \sqrt{\frac{\overline{f}_{\mathrm{th}} \sin(\overline{\theta}_{\mathrm{nd}})}{\rho C_{\mathrm{D}} A}}. \tag{4}$$

Eqs. (3) and (4) explain why increasing the payload weight limits the horizontal speeds. The equations also show that the vehicle becomes more maneuverable after dropping the payload, due to the larger power-to-weight ratio. If the need to maintain altitude is neglected, the pitch angle can be increased above $\overline{\theta}_{\mathrm{nd}}$ and, while the vehicle is allowed to fall, most of its thrust is exploited to accelerate further. The theoretical maximum achievable horizontal speed, that is reachable when the thrust is directed horizontally, meaning $\theta = \frac{\pi}{2}$, is given by

$$\|v_{\max}\mathbf{e}_{1,2}\| = \sqrt{\frac{\overline{f}_{\mathrm{th}}}{\rho C_{\mathrm{D}} A}}. \tag{5}$$

After releasing the payload, the larger power-to-weight ratio allows the UAV to recover from the fall.

It is a challenging task to drop a payload whose weight accounts for a considerable percentage of the overall weight. The commonly-used approach is to treat large mass variations as disturbances, and to increase the robustness of the system to them, rather than planning on the basis of them, or even exploiting them as if they would present an advantage. Examples of the robustness-based approach for mass variations are given in [11], where a $\mathcal{H}_\infty$-based method is proposed, and in [12], where this is attempted by including the variations in the center of mass in the dynamic model, and also in [13], where several control techniques are compared.

Trajectory planning for payload dropping was explored in [14]. However, the payload was supposed to be released from high altitudes, and the payload mass was negligible. Similarly, in [15], a very quick-to-converge trajectory planning method is described but the parameters of the model remain constant throughout the whole trajectory. In addition, the UAV in question in both cases is a fixed-wing aircraft. Generation of aggressive and complex trajectories for quadrotors is discussed in [16], [17], and [18]. These works are relevant and inspirational, but they are aimed at avoiding obstacles or perching on inclined surfaces, and they do not deal with dropping a payload. In [19] and [20], trajectory planning for UAVs carrying cable-suspended payloads (again no dropping) is tackled, but without considering environmental conditions or planning in real time. An example of generating aggressive maneuvers with changing mass is provided in [21], but the mass release is not planned ahead, which is perhaps the main contribution of our paper.

To sum up the contributions, we propose a method for generating a trajectory by solving an OCP, for precise payload dropping in various scenarios. The need to carry a substantial payload, which accounts for $20\%$–$40\%$ of the total mass, implies stretching the boundaries of the flight envelope beyond conventionally safe margins. The trajectory generation is quick enough to enable trajectories to be re-planned in real-time, facilitating the future implementation of a nonlinear MPC scheme.

## II. Dynamic Model Description

The translational dynamics of the UAV model are described in (1) and (2). The system relies on the non-linear controller described in [22], which controls the multicopter motors in such a way that the pitch angle behaves as a first-order linear system. Namely, its dynamic model is given by

$$\dot{\theta} = K_\theta(\theta_{\mathrm{ref}} - \theta), \tag{6}$$

where $\theta$ is the actual pitch angle of the multicopter, $K_\theta$ is the (empirically determined) pitch angle transient factor, and $\theta_{\mathrm{ref}}$ is the reference pitch angle of the multicopter, which is a control input signal. This layered architecture allows us to simplify the model and spares us from modeling more complex features, such as the electric motors or the aerodynamic behavior of the propellers, thus lowering the computational requirements.

### A. Rotation Angles and Reference Frames

We refer to three different reference frames, inertial reference frame $\mathbb{I}$, trajectory reference frame $\mathbb{T}$, and body reference frame $\mathbb{B}$, which are depicted in Fig. 2. Inertial reference frame $\mathbb{I}$ has its $x$ axis pointing north, its $z$ axis pointing to the zenith and its $y$ axis pointing west. Its origin was defined to be in the target location. Trajectory reference frame $\mathbb{T}$ is generated by a rotation from $\mathbb{I}$ around the $z$ axis by the heading angle $\psi$.

The maneuver path is confined to a vertical plane $\mathbb{T}_{xz}$, defined by the $x$ and $z$ axes of $\mathbb{T}$, named $\mathbf{x}_\mathbb{T}$ and $\mathbf{z}_\mathbb{T}$, respectively. The solution is constrained to a plane in order to reduce the computational cost, by solving an optimal control problem in two dimensions instead of three. By also optimizing the orientation of plane $\mathbb{T}_{xz}$, we can still provide a 3D solution. This approach helps us to optimize the trajectory in real time. Solving a full 3D problem, in which the trajectory is not restricted to a 2D plane, would yield insignificant benefits while increasing complexity of the problem exponentially. Based on prior analyses, the optimal solutions in 3D space tend to converge to planar paths, so there is no practical advantage from solving a full 3D problem.

The solution is provided in reference frame $\mathbb{T}$. The heading vector $\mathbf{x}_\mathbb{T} \in \mathbb{R}^2$ is determined by the optimizer and remains fixed through the whole maneuver. The wind velocity $\mathbf{v}_{\text{wind}}$, which is modeled as a constant $\mathbb{R}^3$ vector parallel to the terrain plane, can be divided into two components: one parallel to $\mathbb{T}_{xz}$ named $\mathbf{v}_{\text{wind}\parallel}(\psi)$, and the crosswind that is perpendicular to it named $\mathbf{v}_{\text{wind}\perp}(\psi)$.

The orientation of frame $\mathbb{B}$ is defined by the pitch $\theta$ and roll $\phi$ angles. Pitch angle $\theta$ varies through the maneuver according to (6). Roll angle $\phi$ is not directly modeled. Instead, it is assumed that the attitude control unit will regulate it in order to compensate for the crosswind, allowing the UAV to follow the planar trajectory. This results in a reduction in the available thrust on the trajectory plane. The thrust reduction coefficient described by

$$c_{\text{th}}^{\mathbb{T}}(\psi) = \sqrt{1 - \left( \frac{\|\mathbf{f}_\text{D}(\mathbf{v}_{\text{wind}\perp}(\psi))\|}{f_{\text{th}}} \right)^2}, \qquad (7)$$

is included into (1) as a factor which adjusts the thrust, and this is therefore the sole effect of the crosswind on the OCP solution.

### B. Trajectory Stages

The dropping maneuver proposed in this paper consists in following a trajectory generated by an optimizer. The trajectory is divided into three consecutive stages, depicted in Fig. 3: *Approach*, *Release* and *Recovery*. We assume that the UAV is capable of reaching the starting point state prior initiation of the maneuver, and is also capable of taking over the control after the end of the trajectory is reached.

The *Approach* stage is the segment of the trajectory from the start point at time $t_0$ until the UAV reaches the release point, which is the location where the payload begins to be released at time $t_{\text{rel}}$. This stage lasts for $\tau_{\text{app}}$, and during this interval the vehicle accelerates towards the release point while losing altitude. The release point is determined by the optimizer, assuming that the payload, once released, follows an ideal ballistic trajectory towards the target.

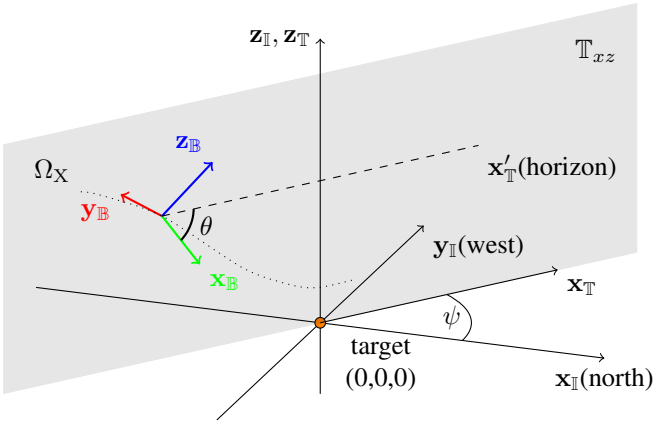



Fig. 2: The reference frames on which the optimal control problem is defined. The trajectory $\Omega_\text{X}$ is confined to a plane $\mathbb{T}_{xz}$ which pivots around the $z$ axis above the origin. The $z$ axis is common to $\mathbb{I}$ and $\mathbb{T}$.

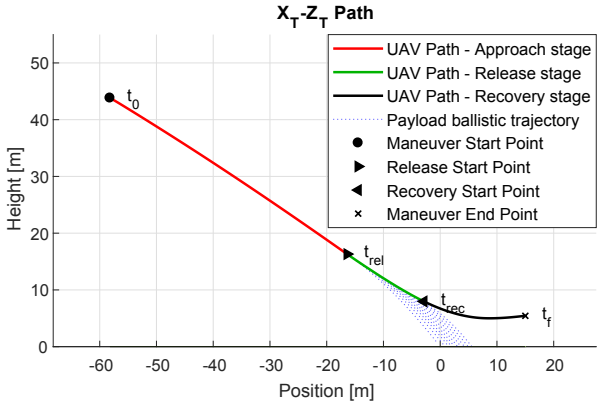



Fig. 3: A typical dropping path generated by the optimizer. The UAV moves from left to right and the target is located at the origin of the axes. The first segment of the trajectory (in red) is the *approach* stage, the second segment (in green) is the *release* stage, and the last segment (in black) is the *recovery* stage.

The *Release* stage is the segment of the trajectory from the release point at time $t_{\text{rel}}$ until the payload is completely released at time $t_{\text{rec}}$. The time lapse for this stage is defined as $\tau_{\text{rel}}$.

The *Recovery* stage is the segment that begins after the end of the payload release at time $t_{\text{rec}}$, and continues until the end of the trajectory at time $t_\text{f}$. The duration of this stage is $\tau_{\text{rec}}$ and its purpose is to diminish the downwards vertical speed acquired during the approach stage, in order to ensure that the vehicle will not impact the ground after releasing the payload.

### C. Payload Releasing Models

The vehicle in charge of delivering the fire extinguishing agent to the fire could employ various methods to release it. For example, the liquids could be released all at once in the form of a balloon, or gradually through an opening on a container which could be either pressurized or not. The relevant payload releasing models are modeled as ordinary differential equations (ODE) which describe the evolution of the total mass $m(t)$ over time as follows:

1) *Immediate Release* mimics a mechanism in which the payload in its totality is immediately released. In this release model, the mass change is represented by a negative delayed Heaviside step function $\mathbb{U}(t)$, and thus the total mass along the trajectory is given by

$$m(t) = m_{\text{UAV}} + m_{\text{pyld}}(1 - \mathbb{U}(t - t_{\text{rel}})), \tag{8}$$

where $m_{\text{UAV}}$ is the mass of the vehicle and $m_{\text{pyld}}$ is the mass of the total payload. It is impossible to describe this behavior with an ODE, so this releasing model is modeled by imposing a discontinuity in the mass parameter.

2) *Pump* mimics a constant flux pump, which can be either *on* or *off*. Here the mass decreases at a constant rate from $m_{\text{UAV}} + m_{\text{pyld}}$ over a period of time until it reaches $m_{\text{UAV}}$. The mass decrease rate is a constant defined by the properties of the modeled pump. In order to achieve this, the state ODE, described in (13), is expanded as

$$\dot{m} = \begin{cases} \text{const} < 0, & \text{stage} = \text{release}, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

3) *Hole in a Bucket* is modeled based on Bernoulli's equation, in which the fluid is incompressible, and in which viscosity and friction are neglected. The velocity at which the fluid leaves the bucket through a hole in its bottom is given by a modification of the potential/kinetic energy equation

$$v_{\text{f}} = \sqrt{2 g_{\text{f}} h_{\text{f}}}, \tag{10}$$

where $v_{\text{f}}$ is the fluid exit velocity through the hole in the bottom of the bucket, $h_{\text{f}}$ is the height of the fluid column and $g_{\text{f}}$ is the gravity (*felt* by the fluid) given by the sum of the gravity acceleration $g$ and $\dot{v}_z$. Eq. (10) can be modified to describe $m_{\text{pyld}}$ by multiplying both sides of it by the area of the hole in the bottom of the bucket $a_{\text{hole}}$ and the fluid density $\rho_{\text{f}}$. Then, by representing $h_{\text{f}}$ as a function of $m_{\text{pyld}}$ and the bucket base area $a_{\text{bucket}}$, we obtain

$$\dot{m}(t) = -a_{\text{hole}} \sqrt{2(g + \dot{v}_z(t)) \frac{\rho_{\text{f}} m_{\text{pyld}}(t)}{a_{\text{bucket}}}}. \tag{11}$$

Fig. 4 depicts the behavior of the total mass during the release stage for different releasing models. As shown in (11), the rate at which the mass is released is dependent on the acceleration in the $\mathbf{z}_{\mathbb{B}}$ axis. Nevertheless, $\dot{v}_z$ was set to a constant for depiction purposes. Fig. 4 shows that the behavior of a constant flux pump and a bucket with a hole are similar. However, the *Hole in a Bucket* model has a higher computational cost, making it impractical.

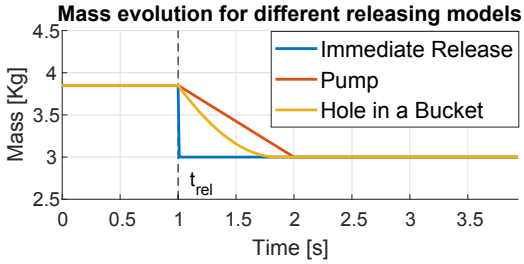



Fig. 4: The evolution of the total mass for different releasing models.

## III. Optimal Control Problem Formulation

Based on task specifications provided by first responders and firefighters, the following assumptions were made: 1) The UAV position, velocity, mass, attitude and aerodynamic drag coefficient are assumed to be accurately known. For precise UAV state estimation, we rely on our previous work, as described in [23]. 2) The UAV is equipped with an on-board controller (e.g. [22]), which is responsible for following the obtained trajectory. 3) Reaching the initial states before performing the maneuver, and also taking over after the maneuver is finished, is the responsibility of an MPC tracker which is described in [24], [25]. 4) The target is defined as a single point on the ground, the location of which is known and on which inertial reference frame $\mathbb{I}$ originates. 5) The terrain in the area surrounding the fire focus is known prior to the trajectory planning, and it is assumed to be any **constant gradient** plane. 6) No obstacles the UAV may need to avoid are located anywhere on the feasible path space. 7) The wind velocity is known, and it is assumed to flow parallel to the terrain plane in a uniform manner. 8) The payload, once released, is assumed to follow an ideal ballistic trajectory (no air drag or diffusion is considered). This is a reasonable assumption in the case of low altitude dropping. 9) The payload is assumed to be a single rigid point, so it does not affect behavior of the UAV in any way other than by altering its total mass and moment of inertia (e.g. no sloshing and no changes in the center of mass). 10) No communication is required, except for the initiation and safety abort commands. The on-board UAV system is able to automatically trigger the payload releasing mechanism at a time dictated by the trajectory planner.

The multi-stage OCP generating the trajectory is formulated as follows:

$$\begin{aligned} \underset{\mathbf{U}(t),\mathbf{X}(t),t_{\text{rel}},t_{\text{f}},t_{\text{ff}},\psi}{\text{minimize}} \quad & J(\mathbf{X}(t_{\text{rel}}), t_{\text{f}}) \\ \text{subject to} \quad & \dot{\mathbf{X}} = f(\mathbf{X}(t), \mathbf{U}(t), m(t), \psi), \\ & \underline{\mathbf{U}} \leq \mathbf{U}(t) \leq \overline{\mathbf{U}}, \\ & \underline{\mathbf{X}} \leq \mathbf{X}(t) \leq \overline{\mathbf{X}}, \\ & G(\mathbf{X}(t_{\text{rel}}), t_{\text{ff}}) = 0, \\ & H(\mathbf{X}(t), \psi) \geq 0, \end{aligned} \tag{12}$$

where $\mathbf{X} \overset{\text{def}}{=} \left(p_x, v_x, p_z, v_z, \theta\right)^T$ is the state vector, $p_x$ and $p_z$ the positions, and $v_x$ and $v_z$ are the velocities of the UAV in the $\hat{x}$ and $\hat{z}$ axes of $\mathbb{T}$, respectively. The elements of $\mathbf{X}$

are bounded between minimum $\underline{\mathbf{X}}$ and maximum $\overline{\mathbf{X}}$ values. $\mathbf{U}$ is the control signal vector given by $\mathbf{U} = \left(\theta_{\mathrm{ref}}, f_{\mathrm{th}}\right)^T$, which is also bounded between minimum $\underline{\mathbf{U}}$ and maximum $\overline{\mathbf{U}}$ values.

Function $f(\mathbf{X}(t), \mathbf{U}(t), m(t), \psi)$—describing the UAV motion in the trajectory reference frame—is derived from (1), (6) and (7) as

$$\begin{pmatrix} \dot{p}_x \\ \dot{v}_x \\ \dot{p}_z \\ \dot{v}_z \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v_x \\ \frac{f_{\mathrm{th}} c_{\mathrm{th}}^{\mathbb{T}}(\psi)\sin(\theta) - f_{\mathrm{D}}(v_x)}{m(t)} \\ v_z \\ \frac{f_{\mathrm{th}} c_{\mathrm{th}}^{\mathbb{T}}(\psi)\cos(\theta) - f_{\mathrm{D}}(v_z)}{m(t)} - g \\ K_\theta(\theta_{\mathrm{ref}} - \theta) \end{pmatrix}. \tag{13}$$

For the *Immediate Release* model, $m(t)$ is given by

$$m(t) = \begin{cases} m_{\mathrm{UAV}} + m_{\mathrm{pyld}}, & t \in [0, t_{\mathrm{rel}}), \\ m_{\mathrm{UAV}}, & t \in [t_{\mathrm{rel}}, t_{\mathrm{f}}], \end{cases} \tag{14}$$

whereas for *Pump* and *Hole in a Bucket*, (13) is expanded by adding on a new row with either (9) or (10), respectively. $J(\mathbf{X}(t_{\mathrm{rel}}), t_{\mathrm{f}})$ is the cost function, given by

$$J(\mathbf{X}(t_{\mathrm{rel}}), t_{\mathrm{f}}) = w_{\mathrm{t}} t_{\mathrm{f}} + w_{\mathrm{pz}} p_z(t_{\mathrm{rel}}) - w_{\mathrm{vx}} v_x(t_{\mathrm{rel}}), \tag{15}$$

where $w_{\mathrm{t}}$, $w_{\mathrm{pz}}$ and $w_{\mathrm{vx}}$ are weights for the overall maneuver time, altitude upon release, and horizontal speed upon release, respectively. Constraint $G(\mathbf{X}(t_{\mathrm{rel}}), t_{\mathrm{ff}}) = 0$ ensures that the payload, after falling for $t_{\mathrm{ff}}$ seconds, hits the target. Constraint $H(\mathbf{X}(t), \psi) \geq 0$ enforces flying above a minimum height. It should be noted that the term related to the horizontal velocity at release time $v_x(t_{\mathrm{rel}})$ is negated, meaning that the aim of the OCP is to maximize it.

### A. OCP Solution

The multi-stage OCP (12) is solved by a direct *Multiple Shooting* method [26], a general method used for transcribing a continuous-time OCP to a *Nonlinear Programming Problem (NLP)*. The trajectory is discretized into $N$ time steps and is represented as a sequence of states $\mathbf{\Omega}_{\mathrm{X}} = \{\mathbf{X}[0], \mathbf{X}[1], \ldots, \mathbf{X}[N]\}$ propagated according to a sequence of input control signals $\mathbf{\Omega}_{\mathrm{U}} = \{\mathbf{U}[0], \mathbf{U}[1], \ldots, \mathbf{U}[N-1]\}$. Each stage is divided into a preset constant number of evenly spaced time steps $N_{\mathrm{app}}$, $N_{\mathrm{rel}}$ and $N_{\mathrm{rec}}$, which hold $N_{\mathrm{app}} + N_{\mathrm{rel}} + N_{\mathrm{rec}} = N$. In the *Immediate Release* model, where the *Release* stage has zero length, the trajectory consists solely of the *Approach* and *Recovery* stages. As the switching times between the stages are a priori unknown, we invoke a common approach of adding the lengths of the time steps in each stage among the optimization variables, as demonstrated in [27], [28], [29]. The resulting NLP is:

$$\begin{aligned} \underset{\mathbf{\Omega}_{\mathrm{X}}, \mathbf{\Omega}_{\mathrm{U}}, \Delta t_a, t_{\mathrm{ff}}, \psi}{\text{minimize}} \quad & J(\mathbf{X}[k_{\mathrm{rel}}], t_{\mathrm{f}}) \\ \text{subject to} \quad & \mathbf{X}[k+1] = F(\mathbf{X}[k], \mathbf{U}[k], m[k], \Delta t_a, \psi), \\ & \underline{\mathbf{U}} \leq \mathbf{U}[k] \leq \overline{\mathbf{U}}, \\ & \underline{\mathbf{X}} \leq \mathbf{X}[k] \leq \overline{\mathbf{X}}, \\ & G(\mathbf{X}[k_{\mathrm{rel}}], t_{\mathrm{ff}}) = 0, \\ & H(\mathbf{X}[k], \psi) \geq 0, \end{aligned} \tag{16}$$

where $J$ is the cost function defined in (15). $F$ is a 4th order *Runge-Kutta* propagation formula of the dynamic model described in (13), and $\Delta t_a = \frac{\tau_a}{N_a}$ is the length of the time step in stage $a \in \{\mathrm{app}, \mathrm{rel}, \mathrm{rec}\}$. Thus the total length of the trajectory is $t_{\mathrm{f}} = \sum_a \Delta t_a N_a$. Index $k_{\mathrm{rel}}$ corresponds to the release time, that is $k_{\mathrm{rel}} = N_{\mathrm{app}}$.

For the *Approach* and *Recovery* stages, $m[k]$ is constant:

$$m[k] = \begin{cases} m_{\mathrm{UAV}} + m_{\mathrm{pyld}}, & k = 0, \ldots, k_{\mathrm{rel}}, \\ m_{\mathrm{UAV}}, & k = k_{\mathrm{rec}} + 1, \ldots, N, \end{cases} \tag{17}$$

and for the *Release* stage, a constraint inducing the corresponding release model dynamics $\dot{m}(t)$ by a 4th order *Runge-Kutta* integration scheme, is added to the constraints.

The box constraints applied to the state vector $\mathbf{X}$ and the control vector $\mathbf{U}$, and also the free-fall $G$ and minimum height $H$ constraints, are identical to those applied to the continuous time case.

The trajectory heading is determined not by directly optimizing $\psi$, but by optimizing the elements of $\mathbf{x}_{\mathbb{T}}$ instead. This measure facilitates the calculation, by simple algebraic methods, of $\mathbf{v}_{\mathrm{wind}\parallel}$ and $\mathbf{v}_{\mathrm{wind}\perp}$; and also the height above the terrain, which is given by

$$h = p_z - p_x \mathbf{x}_{\mathbb{T}} \cdot \mathbf{r}, \tag{18}$$

where $\mathbf{r}$ is the terrain gradient vector. This method is implemented by adding another nonlinear constraint forcing $\|\mathbf{x}_{\mathbb{T}}\| = 1$.

The optimization vector $\mathbf{\Omega}_{\mathrm{L}}$ is created by concatenating the state trajectory $\mathbf{\Omega}_{\mathrm{X}}$, the control sequence $\mathbf{\Omega}_{\mathrm{U}}$, the time step lengths $\Delta t_a$, the payload free-fall time $t_{\mathrm{ff}}$, and the heading vector $\mathbf{x}_{\mathbb{T}}$.

The NLP (16) was solved by the *CasADi* suite [30], which is an open-source framework for algorithmic differentiation with an interface to NLP solvers. The chosen solver was IPOPT (**I**nterior **P**oint **OPT**imizer), which is well suited for large-scale nonlinear optimization problems [31].

### B. Other Constraints

For brevity, in (16) we omitted the following constraints, the values of which are chosen in such a way that the trajectory is feasible, converges the way that we intend, and does so as quickly as possible. The time step lengths $\Delta t_a$ are constrained to values larger than $1$ ms. In addition, $\theta_{\mathrm{ref}}$ and $\theta$ are both limited to the range $\pm\pi/2$ in order to confine the thrust vector to the 1st and 2nd quadrants. This avoids pointing it downwards, which could be dangerous while not presenting any apparent benefit. In addition, the box constraints for $\mathbf{X}(t_{\mathrm{f}})$ are set in such a way that the maneuver execution can be considered accomplished (e.g. the UAV is past the target, above a certain altitude, moving forward and climbing). Another non-linear constraint imposes a low limit on $h$, in order to force the optimizer to generate trajectories that maintain a minimum safe distance from the terrain throughout the maneuver.

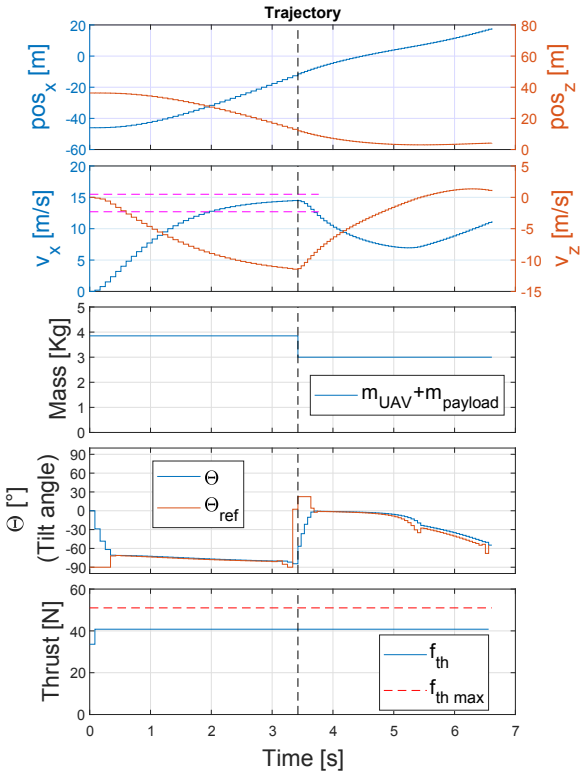



Fig. 5: The state and control signals of the vehicle through a trajectory for the *Immediate Release* payload releasing model. The lower and higher magenta dashed lines on the velocity graph represent $\overline{v}_{\mathrm{nd}}$ and $\|v_{\max}\mathbf{e}_{1,2}\|$, respectively. The vertical black dashed line represents $t_{\mathrm{rel}}$.

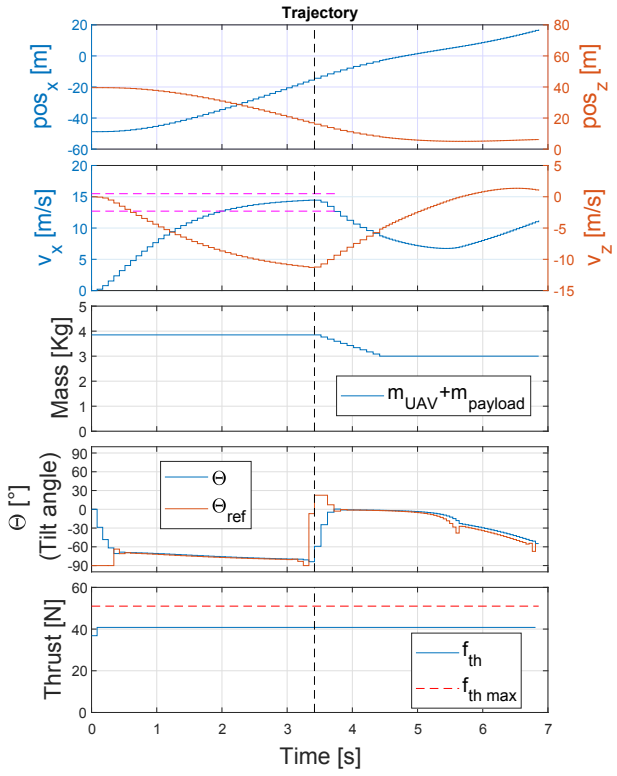



Fig. 6: The state and control signals of the vehicle through a trajectory for the *Pump* payload releasing model. The lower and higher magenta dashed lines on the velocity graph represent $\overline{v}_{\mathrm{nd}}$ and $\|v_{\max}\mathbf{e}_{1,2}\|$, respectively. The vertical black dashed line represents $t_{\mathrm{rel}}$.

## IV. Optimization Results

The trajectory generation results for the *Immediate release* and *Pump* releasing modes are shown in Fig. 5 and Fig. 6, respectively. The horizontal red dashed line on both thrust graphs marks the maximum available thrust for the modeled UAV for trajectory planning. In order to provide a safety margin, and also to prevent the motors reaching saturation, causing control loss, a constraint limiting the planning thrust to about $80\%$ of the total available thrust was enforced. Note that the thrust control signal shown on both figures, maintains its maximum value throughout the maneuver. This demonstrates that the maneuver is as aggressive as the UAV can perform[1]. The optimization was run in Matlab on an Intel Core i7-6820HQ CPU at 2.7 GHz. When $\mathbf{\Omega}_{\mathrm{L}}$ is initialized to values *close* to the results, the optimization time can be brought down to about 100 ms.

### A. Effect of Environmental Parameters

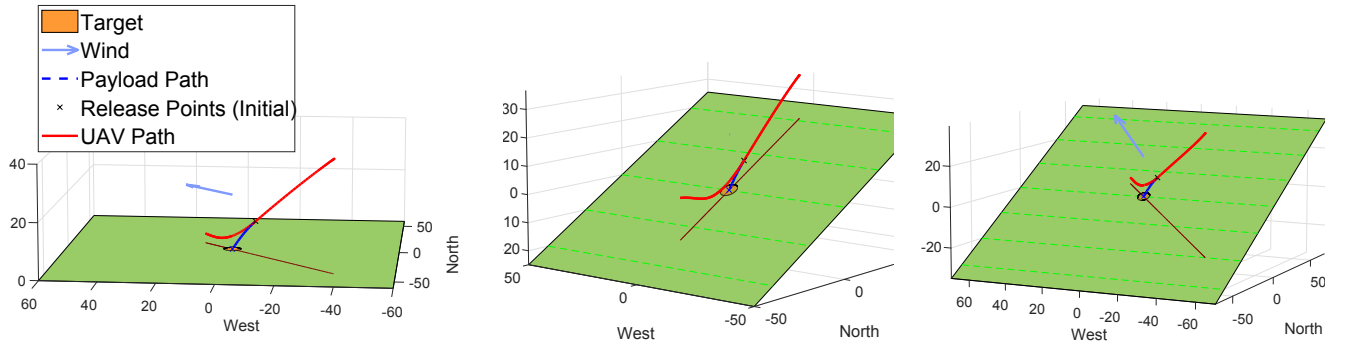
As common sense would imply, the heading angle $\psi$, which the optimizer chooses to engage into the maneuver, is such that the trajectory goes *downwind* and/or *downhill* (so that the horizontal speed is maximized). For example, Fig. 7a, which depicts the resulting path when $\mathbf{v}_{\mathrm{wind}} = 3$ m/s but $\|\mathbf{r}\| = 0$, shows that the optimizer reached a solution in which the maneuver is executed in the same direction as the wind flows. Fig. 7b, on the other hand, depicts the resulting path when the terrain is inclined but there is no wind. Here the maneuver is executed in the opposite direction of $\mathbf{r}$ (i.e. downhill), as expected. Combining these two environmental factors will cause the trajectory to be executed somewhere in between these two outcomes. If the wind flows uphill, depending its strength, the trajectory may even be executed *uphill*, as depicted in Fig. 7c.

[1]Link to video: `https://youtu.be/4o6zVwIzRIQ`

## V. Experimental Evaluation

Following the results presented in Section IV, we decided to pursue further experiments only with the *Immediate Release* model. This was because the differences between the results of the two (*Immediate release* and *Pump*) releasing models are insignificant for the state values, and also for the control signal values, as shown in Fig. 5 and Fig. 6. In addition, the *Hole in a Bucket* releasing model is distinctly more computationally demanding than the other two models, whereas its behavior closely resembles the *Pump* releasing model. Finally, from the practical point of view, new solid fire extinguishing tools can being used as proposed by [8]. These tools are assumed to be suited to be used by UAVs, due to their high extinguishing efficiency for a given weight.

Prior to real deployment, the feasibility and the safety of the maneuver were verified in *Gazebo*. Gazebo is a dynamic




(a) Wind only (b) Terrain gradient only (c) Strong oblique uphill wind and terrain gradient

Fig. 7: Resulting trajectories for various combinations of wind speeds and terrain gradients. The wind direction proceeds from 135° and the terrain slope is 50% pointing north.

simulator that includes a physics engine, and also has the ability to run a ROS instance and to render 3D visualizations of the modeled scenario.[2]

### A. Real Experiments

Experiments were performed to prove the feasibility of the maneuver, and to assess the gain in speed, the dropping altitude and its precision. No real fires were lit and no attempts were made to extinguish a fire in the experiments.

The protocol of the experiments involved making a UAV execute the generated maneuver several times, drop a rigid object and attempt to hit an exact spot with it. Then, the precision and the accuracy of the drops was examined. Afterwards, a final maneuver was performed, in which a *malfunction* on the payload releasing mechanism was simulated by manually disabling it. This last execution was performed to test whether the UAV was capable of recovering from the high vertical speed that it reached, when fully loaded or not. If not—provided that the UAV did not impact the ground when the releasing mechanism *did* work—it would mean that the resulting optimized trajectory did push the boundaries of the UAV, resulting in the most aggressive maneuver it could possibly perform under the designed constraints. In other words, the UAV dropped a given payload mass as close as possible to the target at the highest possible horizontal velocity, while maintaining accuracy and still surviving the maneuver. Two experiment sessions were conducted. The goal of the first session was to ensure the feasibility and the safety of the proposed concept. Therefore, lighter payloads were used and less aggressive maneuvers were performed in that session. The various sources of inaccuracy that were detected and corrected are discussed in greater detail on the project website[3]. The second experiment yielded the improved results discussed here.

The platform used for the experiments was a multicopter with an electromagnet as the releasing mechanism, described in [32]. The chosen payload was a solid metallic object with a high ballistic coefficient which makes it almost immune to aerodynamic perturbations, ensuring that it follows a nearly perfect ballistic trajectory after being released. The experiment used the following parameters: the maximum thrust that the UAV's 6 motors can generate is 51 N. However, as was explained in section IV, the tested trajectory was generated assuming a 20% lower value (i.e. $\overline{f}_{\text{th}} = 40.8$ N). The mass values, $m_{\text{UAV}} = 3.020$ kg and $m_{\text{pyld}} = 0.850$ kg, mean that the payload accounts for 22% of the total weight, and the thrust-to-weight ratio is 1.34 (and it is 1.07 if calculated with the maximum thrust used in the trajectory generation). For these values, $\overline{v}_{\text{nd}} \approx 7.9$ m/s and $\|v_{\max}\mathbf{e}_{1,2}\| \approx 15.5$ m/s. For simplicity, it was also assumed that $\mathbf{v}_{\text{wind}} = 0$ m/s, and $\|\mathbf{r}\| = 0$. The heading angle $\psi$ was constrained to zero, so the trajectory runs along the $\mathbf{x}_{\mathbb{I}} - \mathbf{z}_{\mathbb{I}}$ plane.

### B. Results

In the 2nd experiment, six drop tests were performed. The locations where the payload fell are depicted in Fig. 8. The average measured error was 0.75 m in the $\mathbf{x}_{\mathbb{T}}$ axis and $-1.02$ m in the $\mathbf{y}_{\mathbb{T}}$ axis. The largest error was 4.53 m. The payload dispersion was 1.93 m in the $\mathbf{x}_{\mathbb{T}}$ axis and 0.78 m in the $\mathbf{y}_{\mathbb{T}}$ axis. The maximum horizontal speed that the UAV reached during the maneuver was on an average 14.9 m/s (i.e. a 90% increment above $\overline{v}_{\text{nd}}$). The average release point was at a height of 12.43 m and at a horizontal distance of 11.8 m from the target. In the 7th test, a payload release malfunction was simulated and the UAV impacted the terrain as expected[4]. Except for that test, the payload was always properly released and the UAV never crashed or flew below the minimum allowed altitude.

A few sources of error were identified that are hard to measure. For example, in most tests, when the payload hit the ground, it bounced a few tens of centimeters on the grass, as can be seen in some of the videos. This could explain the positive bias on the fall locations in the $\mathbf{x}_{\mathbb{T}}$ axis. During some of the tests, wind gusts perturbed the UAV when releasing the payload, probably also affecting the dispersion. The bias in the $\mathbf{y}_{\mathbb{T}}$ axis, is larger than the dispersion, and some videos show that the UAV did not pass

[2]Link to the video: `https://youtu.be/S8AgWMuOp4E`

[3] `http://mrs.felk.cvut.cz/mpc-with-time-variable-mass` [4]Link to video: `https://youtu.be/tnGRMUTERq4`

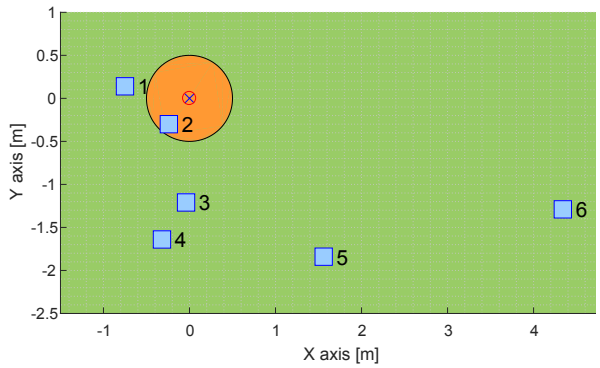



Fig. 8: The blue squares represent the locations where the released payload fell. The orange circle represents location of the target.

above the target as it should have. These observations lead us to believe that perhaps the target was not positioned correctly. Nevertheless, it seems that the error margins achieved in this experiment are compatible with the radius of action of a fire extinguishing ball as explained in [8].

## VI. Conclusions

In this paper, a novel approach to aerial firefighting has been proposed and experimentally verified. During the maneuvers, the UAV flew at close to twice its maximum *conventional* speed for the given payload. Had the target been on fire, the achieved high speed would have considerably reduced the heat exposure. Namely, compared to conventional trajectories, this method would have allowed the UAV to drop a larger payload from a closer distance from the target. When a malfunction in the releasing mechanism was simulated, the UAV was not able to recover from the vertical speed it had gained during the maneuver. The UAV crashed as expected, proving that the generated trajectory was as aggressive as the UAV could perform under the given constraints. Thanks to the short optimization times required to generate the trajectories, the trajectories could be recalculated in real time, enabling the future implementation of an MPC. The dropping accuracy that was achieved is sufficient for deploying fire extinguishing balls for small fires. This is what we will attempt to do in our next round of experiments. Since the dropping accuracy is not dependent on the size of the UAV, this technique could be scaled up for dropping large amounts of water or fire retardant.

## References


[1] J. K. Balch *et al.*, "Human-started wildfires expand the fire niche across the united states," *PNAS*, vol. 114, no. 11, 2017.

[2] J. T. Abatzoglou and A. P. Williams, "Impact of anthropogenic climate change on wildfire across western us forests," *PNAS*, vol. 113, 2016.

[3] E. Beachly *et al.*, "A micro-uas to start prescribed fires," in *ISER*, 2016.

[4] T. M. Whitalker and M. Corson, "Tethered unmanned aerial vehicle fire fighting system," Sep 2017, patent (US9764839B2).

[5] M. Kumar *et al.*, "Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires," *JACIC*, vol. 8, 2011.

[6] C. Phan and H. H. T. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *ICSC*, 2008.

[7] A. Vidovic and D. Diminic, "Possibility of implementing unmanned aerial vehicles in firefighting operations," in *ZIRP*, Apr 2014.

[8] B. Aydin *et al.*, "Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting," *Drones*, vol. 3, no. 1, Feb 2019.

[9] W. Marchant and S. Tosunoglu, "Rethinking wildfire suppression with swarm robotics," in *FCRAR*, 2016.

[10] H. Qin *et al.*, "Design and implementation of an unmanned aerial vehicle for autonomous firefighting missions," in *ICCA*, June 2016.

[11] L. Chang and Y. Jia, "Robust $\mathcal{H}_\infty$ control for tanker station-keeping with mass and inertia variation," in *"CASE"*, Aug 2017, pp. 340–345.

[12] S. D. Lucia *et al.*, "Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach," in *IEEE ECMR*, Sept 2015, pp. 1–6.

[13] I. Sadeghzadeh *et al.*, "Payload drop application of unmanned quadrotor helicopter using gain-scheduled pid and model predictive control techniques," in *Intelligent Robotics and Applications*, 2012.

[14] S. H. Mathisen *et al.*, "Approach methods for autonomous precision aerial drop from a small unmanned aerial vehicle," *20th IFAC World Congress*, pp. 3566–3573, 2017.

[15] C. K. Lai *et al.*, "On-board trajectory generation for collision avoidance in unmanned aerial vehicles," in *2011 Aerospace Conference*.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE ICRA*, May 2011, pp. 2520–2525.

[17] D. Mellinger *et al.*, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 42, pp. 664–674, Apr. 2012.

[18] J. Thomas *et al.*, "Planning and control of aggressive maneuvers for perching on inclined and vertical surfaces," in *ASME*, Aug 2015.

[19] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE ICRA*, 2015.

[20] P. Foehn *et al.*, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *RSS*, 2017.

[21] M. Landolfi *et al.*, "Autonomous guidance navigation and control for agile quadrotors using polynomial trajectory planning and li adaptive control," in *MED*, July 2017.

[22] T. Lee *et al.*, "Geometric tracking control of a quadrotor uav on se(3)," in *CDC*, Dec 2010.

[23] T. Baca *et al.*, "Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle," *JFR*, vol. 36, pp. 874–891, 2019.

[24] T. Baca *et al.*, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *IEEE/RSJ IROS*, 2018.

[25] T. Baca *et al.*, "Embedded model predictive control of unmanned micro aerial vehicles," in *MMAR*, 2016.

[26] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," ser. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984, vol. 17, no. 2, pp. 1603–1608.

[27] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[28] G. Schultz and K. Mombaur, "Modeling and optimal control of human-like running," *IEEE/ASME Transactions on mechatronics*, vol. 15, no. 5, 2009.

[29] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," vol. 439, no. 7072.

[30] J. A. E. Andersson *et al.*, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[31] A. Wächter *et al.*, "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, Mar 2006.

[32] V. Spurný *et al.*, "Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles," *JFR*, vol. 36, no. 1, pp. 125–148, 2019.