2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# An NMPC Framework for Tracking and Releasing a Cable-suspended Load to a Ground Target Using a Multirotor UAV

Fotis Panetsos[1], George C. Karras[2] and Kostas J. Kyriakopoulos[3]

***Abstract*— In this work, we present a nonlinear Model Predictive Control (NMPC) scheme for tracking a ground target using a multirotor with a cable-suspended load. The NMPC framework relies on the dynamic model of the UAV with the suspended load and, hence, an estimate of the load state is obtained by fusing the measurements of a downward-facing camera and a load cell with an Unscented Kalman Filter (UKF). Additionally, since the NMPC relies on the future behavior of the system, the trajectory of the ground target throughout the predicted time horizon of the NMPC, is required. Towards this direction, Bézier curves are employed in order to predict the future trajectory of the target, which moves in an arbitrary way. The ultimate goal of the proposed framework is to release the suspended load to the ground target and, consequently, a condition is checked at each time instant that triggers the opening of a gripper, located at the lower edge of the cable. The performance of the proposed control scheme is experimentally validated using an octorotor.**

## I. INTRODUCTION

Multirotor UAVs with cable-suspended loads have been extensively studied in the literature throughout the last decade due to their versatile nature and their applicability to multiple tasks and scenarios such as emergency situations, search and rescue missions, delivery of essential supplies and equipment, and water sampling from contaminated aquatic environments [1]. In contrast to the rigid attachment of the load to the UAV's frame, the utilization of a cable is a flexible and simple mechanical solution that can be easily integrated into different platforms.

However, the degrees of freedom (DoFs) of the system are increased from 6 to 10, due to the existence of the cable-suspended load, while the control inputs remain 4. Additionally, the motion of the vehicle may lead to large swing angles of the load which put the safety of both the UAV and the load at risk. Various schemes are demonstrated in the literature [2] so as to achieve the control of the under-actuated system; however, the majority of the control schemes is typically validated in indoor environments with accurate state feedback, e.g., by utilizing a motion capture system.

Open-loop controllers, i.e., without feedback of the load state, have been proposed in [3] and [4] where a nonlinear hierarchical control law and an Interconnection and Damping Assignment-Passivity Based Control (IDA-PBC) are employed correspondingly to achieve the transportation of the vehicle with minimum swinging motion of the load. An open-loop approach is also proposed in [5], where dynamic programming is implemented for generating swing-free trajectories, according to a linearized model of the system. Afterward, the authors proposed a reinforcement learning approach in [6] so as to address the model inaccuracies. A reinforcement learning control policy is also developed in [7] where the model-free twin delayed deep deterministic policy gradient algorithm is used so as to train a policy which drives the vehicle towards desired waypoints while simultaneously minimizing the oscillations of the load.

The existing control schemes are not limited to swing-free load transportation and various controllers, which aim to take advantage of the full range of motion of the load, are proposed in the literature. More precisely, Mixed Integer Quadratic Program (MIQP) trajectory generation, closed-loop payload control with vision-based state estimation, and reformulation of the trajectory planning problem as a Mathematical Program with Complementarity Constraints (MPCC), presented in [8], [9] and [10] respectively, consider agile and aggressive maneuvers which are essential for tasks such as execution of trajectories through a slalom course, obstacle avoidance, payload throwing, or maneuvering through a narrow window.

Regarding aerial target tracking, various works address the problem of tracking a moving target with a multirotor UAV using visual feedback; however without considering the existence of a cable-suspended load and, hence, the release of the load towards the target. More specifically, in [11], the authors present a visual scheme for tracking a maneuvering target using a UAV with an onboard gimbal camera. The Kernelized Correlation Filter (KCF) tracker [12] is utilized in order to detect the target while the full state of the target is estimated by the Interacting Multi-Model Extended Kalman Filtering (IMM-EKF) algorithm [13] assuming a Singer model with an adaptive mean. Considering the aforementioned estimates, both the gimbal and the vehicle are controlled in order to achieve the tracking of the moving target. However, the IMM-EKF estimator predicts solely the state of the target at the next time instant while also requiring a target motion model, which is a priori unknown.

In [14], the authors propose a framework for real-time target tracking in cluttered environments. More precisely, the trajectory of the target, observed with the aid of markers, is approximated and predicted for a short time horizon using polynomial regression. Based on the predicted target trajectory and the detected obstacles, the $A^*$ method is utilized in

[1]Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Str. Zografou, Athens 15780, Greece fpanetsos@mail.ntua.gr

[2]Dept. of Informatics and Telecommunications, University of Thessaly, 3rd Km Old National Road Lamia-Athens, 35100, Lamia, Greece gkarras@uth.gr

[3]Center of AI & Robotics (CAIR), New York University, Abu Dhabi kkyria@nyu.edu

order to find a grid-based collision-free flight corridor for the UAV. Eventually, a dynamically feasible for the UAV trajectory is generated using quadratic programming. The authors validate their approach experimentally in an indoor environment.

Polynomial regression for predicting the target motion is also considered in [15], where a Bézier curve is utilized in order to describe the target trajectory. Additionally, the authors assume that the target velocity and acceleration are bounded and, thus, constraints are imposed on the quadratic programming problem. Given the target trajectory, a kinodynamic searching method is implemented so as to find an appropriate flight corridor consisting of obstacle-free grids and, afterwards, an optimal polynomial trajectory is generated within the safe flight corridor. The authors experimentally verify their approach in both indoor and outdoor environments. In their following paper [16], the visibility of the target is also incorporated into the trajectory planning method so as to avoid target loss. Throughout this work, we adopt the target motion prediction with constrained Bézier regression since this approach reliably estimates the future trajectory of the target.

In this work, an NMPC-based scheme is presented for tracking an unmanned ground vehicle (UGV) using a multirotor with a cable-suspended load. The NMPC is formulated for the nonlinear dynamics of the system and, thus, an Unscented Kalman Filter, which fuses measurements provided by a camera and a load cell, is deployed in order to estimate the state of the load. Regarding the detection of the load, a Convolutional Neural Network (CNN) is utilized so as to avoid the usage of a motion capture system or/and markers. A CNN, accompanied by the KCF tracker, is additionally deployed for uninterruptedly detecting the ground target. Similarly to [15], the past observations of the target are integrated into a constrained Bézier regression problem in order to predict the trajectory of the target. However, in contrast with previous works, the future target motion is not utilized for generating trajectories for the UAV, but is directly incorporated into the predicted horizon of the NMPC scheme, which computes appropriate inputs to the inner attitude control loop of the autopilot, and, thus, the following of the aggressively moving target is achieved. Eventually, the load is released towards the ground target, by opening a gripper, as soon as a condition, which relates the predicted ballistic trajectory of the load and the future target motion, is met. It is mentioned that the swinging motion of the load during the tracking of the target renders the successful release of the load quite challenging even for an experienced human operator. The aforementioned control strategy is deployed on an octorotor in an outdoor environment, as depicted in Figure 1, while a UGV is arbitrarily moving on the ground.

Fig. 1: The multirotor with the cable-suspended load and the UGV deployed in the field.

## II. Equations of Motion

In this section, the equations of motion, that describe the kinematics and dynamics of the system, are presented [17].

Consider the multirotor UAV with the cable-suspended load, as illustrated in Figure 1. Let **B** denote a body-fixed frame, attached to the vehicle's center of mass, and **I** an ENU Inertial frame, the origin of which is located at a constant position. The positions of the vehicle and the load w.r.t. **I** are denoted by ${}^I\mathbf{p} = \begin{bmatrix} {}^Ix & {}^Iy & {}^Iz \end{bmatrix}^T$ and ${}^I\mathbf{p}_L = \begin{bmatrix} {}^Ix_L & {}^Iy_L & {}^Iz_L \end{bmatrix}^T$ respectively. Assuming that the upper edge of the cable coincides with the vehicle's center of mass and that the cable is taut, the cable unit vector ${}^I\mathbf{n}$ is defined by the following equation:

$$ {}^I\mathbf{n} = \frac{1}{l}\left({}^I\mathbf{p}_L - {}^I\mathbf{p}\right) \tag{1} $$

with $l$ denoting the length of the cable.

Given that the cable unit vector has 2 DoFs, it may be parameterized w.r.t. **I** by the angles $\eta_L = \begin{bmatrix} \phi_L & \theta_L \end{bmatrix}^T$ which describe successive rotations around the $x$ and $y$ axes correspondingly:

$$ {}^I\mathbf{n} = \begin{bmatrix} -s\theta_L c\phi_L & s\phi_L & -c\theta_L c\phi_L \end{bmatrix}^T \tag{2} $$

where $s(\cdot) = \sin(\cdot)$ and $c(\cdot) = \cos(\cdot)$. By differentiating Eq. 2, a matrix $J$ is computed which relates the derivative of the cable unit vector ${}^I\dot{\mathbf{n}}$ with the angular velocity $\dot{\eta}_L$:

$$ {}^I\dot{\mathbf{n}} = \underbrace{\begin{bmatrix} s\theta_L s\phi_L & -c\theta_L c\phi_L \\ c\phi_L & 0 \\ c\theta_L s\phi_L & s\theta_L c\phi_L \end{bmatrix}}_{J} \dot{\eta}_L \tag{3} $$

Moreover, by double differentiating Eq. 1 and assuming that the quantities $\dot{\theta}_L^2$, $\dot{\phi}_L^2$, $\dot{\phi}_L \cdot \dot{\theta}_L$, are negligible, the following relationship is derived:

$$ J\ddot{\eta}_L = \frac{1}{l}\left({}^I\dot{\mathbf{v}}_L - {}^I\dot{\mathbf{v}}\right) \tag{4} $$

where ${}^I\mathbf{v} = \begin{bmatrix} {}^Iv_x & {}^Iv_y & {}^Iv_z \end{bmatrix}^T$, ${}^I\mathbf{v}_L = \begin{bmatrix} {}^Iv_{L_x} & {}^Iv_{L_y} & {}^Iv_{L_z} \end{bmatrix}^T$ are the velocities of the vehicle and the load respectively.

Based on the Newton-Euler equations of the load, the load acceleration ${}^I\dot{\mathbf{v}}_L$ is defined as:

$$ {}^I\dot{\mathbf{v}}_L = -g\mathbf{e_3} - \frac{\|\mathbf{T}\|}{m_L}{}^I\mathbf{n} \tag{5} $$

where $\|\mathbf{T}\|$ is the cable's tension, $m_L$ is the load's mass, $\mathbf{e_3}$ is the z-axis of the Inertial frame, and $g$ is the gravitational acceleration. By substituting Eq. 5 in 4 and given that $J^T {}^I\mathbf{n} = \mathbf{0}_{2\times 1}$, the following equation is eventually obtained:

$$\ddot{\eta}_L = \frac{1}{l}\left(J^T J\right)^{-1} J^T \left(-g\mathbf{e_3} - {}^I\dot{\mathbf{v}}\right) \tag{6}$$

The above Eq. 6 is defined for $|\phi_L| \neq \pi/2$, which is valid in case of non-aggressive load maneuvers.

As for the multirotor, the rigid body equations of motion under external disturbances [18], i.e., the tension of the cable in this particular case, are utilized for the translational dynamics of the vehicle, whereas a first-order model is used in order to describe the inner attitude control loop, due to the capability of the autopilot to efficiently control the rotational dynamics of the UAV [19]. Eventually, the complete dynamic model of the vehicle with the cable-suspended load, considering also Eq. 6, is:

$$\begin{aligned}
{}^I\dot{\mathbf{p}} &= {}^I\mathbf{v} \\
{}^I\dot{\mathbf{v}} &= -g\mathbf{e_3} + F\,{}^I\mathbf{R}_B\mathbf{e_3} + \frac{\|\mathbf{T}\|}{m}{}^I\mathbf{n} \\
\dot{\phi} &= \left(K_\phi \phi_{ref} - \phi\right)/\tau_\phi \\
\dot{\theta} &= \left(K_\theta \theta_{ref} - \theta\right)/\tau_\theta \\
\dot{\psi} &= \left(K_\psi \psi_{ref} - \psi\right)/\tau_\psi \\
\dot{\eta}_L &= \dot{\eta}_L \\
\ddot{\eta}_L &= \frac{1}{l}\left(J^T J\right)^{-1} J^T \left(-g\mathbf{e_3} - {}^I\dot{\mathbf{v}}\right)
\end{aligned} \tag{7}$$

where $m$ is the mass of the UAV, $F$ denotes the thrust, normalized w.r.t. $m$, ${}^I\mathbf{R}_B$ is the rotation matrix from $\mathbf{B}$ to $\mathbf{I}$ computed by the roll, pitch, yaw angles, i.e., $\phi$, $\theta$, $\psi$ correspondingly, and the parameters $K_\phi$, $K_\theta$, $K_\psi$ and $\tau_\phi$, $\tau_\theta$, $\tau_\psi$ are the gains and time constants of the inner attitude control loop, which are obtained by following a system identification procedure. Finally, the angles $\phi_{ref}$, $\theta_{ref}$, $\psi_{ref}$ denote the reference roll, pitch, yaw angles, i.e., the inputs for the inner attitude control loop, while the actual thrust is computed according to the reference vertical velocity ${}^I v_{z_{ref}}$:

$$F = \frac{g + \left(K_z {}^I v_{z_{ref}} - {}^I v_z\right)/\tau_z}{c\theta c\phi} \tag{8}$$

where $K_z$ and $\tau_z$ are the gain and the time constant of the inner thrust controller.

It is mentioned that the inner attitude control loop of the multirotor's autopilot, specifically the open source Ardupilot firmware [20], is incorporated into the aforementioned dynamic model. Contrastingly, the outer PID position control loop is replaced by the proposed NMPC scheme in order to consider the existence of the cable-suspended load, which alters the translational dynamics of the system, and to control the load angles and angular velocity.

## III. Problem Formulation

In this section, the motion control problem of tracking a ground target using a multirotor with a cable-suspended load is formulated. Consider the case in which a ground vehicle is moving arbitrarily on the ground without sharing any information about its state, i.e., its position ${}^I\mathbf{p}_T$ and velocity ${}^I\mathbf{v}_T$, and, hence, the UAV relies only on the visual information, obtained by the onboard downward-looking camera, so as to fulfill the requested task.

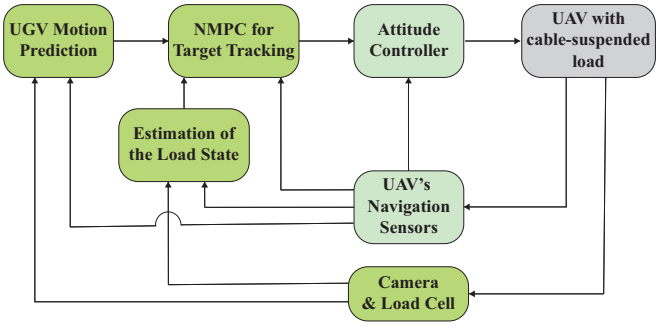



Fig. 2: The proposed control scheme.

Towards this direction, the control scheme, depicted in Figure 2, is deployed so as to achieve the tracking of the ground target and the successful release of the suspended load. The aforementioned control scheme consists of the following parts:

1) CNN-based detection of the cable and estimation of the full state of the load, i.e., $\eta_L$ and $\dot{\eta}_L$, using a UKF which exploits the output of the CNN and the measurements provided by a load cell, located at the lower edge of the cable,
2) Detection of the target using a KCF tracker and a CNN, which indicates an initial region of interest (ROI) for the KCF tracker, and prediction of the target motion, i.e. the position ${}^I\mathbf{p}_T$ and the velocity ${}^I\mathbf{v}_T$ w.r.t. $\mathbf{I}$, based on Bézier curves,
3) Deployment of an NMPC, formulated for the nonlinear dynamics of the system, which integrates the future trajectory of the target into the predicted horizon of the NMPC and produces appropriate references for the inner attitude control loop of the autopilot so as to minimize the tracking error between the target and the load and
4) Check of a condition which, based on the predicted trajectory of the target and the load, triggers the opening of a gripper and the release of the load towards the target.

The above individual parts are explicitly presented in the following sections.

## IV. Estimation of the Load State

### A. Detection of the Load

The estimation of the load angles and angular velocity is a prerequisite for the deployed NMPC. For this purpose, a CNN is applied to the image provided by the downward-facing camera, in order to detect the cable. The CNN-based approach is selected due to its robustness against lighting conditions, typically observed in outdoor environments.

Among the various models from the Keras image segmentation framework [21], the computationally lightweight *mobilenet_segnet* model is selected. Following a supervised

learning approach, images were initially collected and were afterwards labeled [22]. The output of the CNN, after the training procedure, is depicted in Figure 3a.

Given the segmented image produced by the CNN, the edge of the cable and, hence, the pixel coordinates of the load $(u_L, v_L)$ are computed. Since the cable is assumed taut and the upper edge of the cable coincides with the origin of the body-fixed frame $\mathbf{B}$, the 3D position of the load ${}^B\mathbf{p}_L$ w.r.t. $\mathbf{B}$ can be computed after applying the appropriate transformations between the camera frame $\mathbf{C}$ and the body-fixed one $\mathbf{B}$, and considering also that $\|{}^B\mathbf{p}_L\| = l$. Eventually, by dividing the position ${}^B\mathbf{p}_L$ with the cable length $l$ and converting it to $\mathbf{I}$, the cable unit vector ${}^I\mathbf{n}$ is estimated.

### B. Unscented Kalman Filter

In order to retrieve the full state of the load, the load angular velocity is required. Consequently, an Unscented Kalman Filter is applied according to the Newton-Euler equations of the load (Eq. 5). Specifically, the following state transition and measurement equations are utilized in the UKF model:

$$\dot{\mathbf{x}}_L = \begin{bmatrix} {}^I\dot{\mathbf{n}} \\ {}^I\ddot{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} {}^I\dot{\mathbf{n}} \\ \frac{1}{l}\left(-g\mathbf{e_3} - \frac{\|\mathbf{T}\|}{m_L}{}^I\mathbf{n} - {}^I\dot{\mathbf{v}}\right) \end{bmatrix} + \mathbf{w} \tag{9}$$

$$\mathbf{y}_L = {}^I\mathbf{n} + \boldsymbol{\omega} \tag{10}$$

where $\mathbf{w} \in \mathbb{R}^6$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ are process and measurement noise terms and the cable's tension $\|\mathbf{T}\|$ is measured by the load cell. Eventually, after applying the aforementioned UKF, the load angle $\eta_L$ and angular velocity $\dot{\eta}_L$ are extracted based on Eq. 2 and 3.

## V. Estimation of the Target State

### A. Detection of the Target

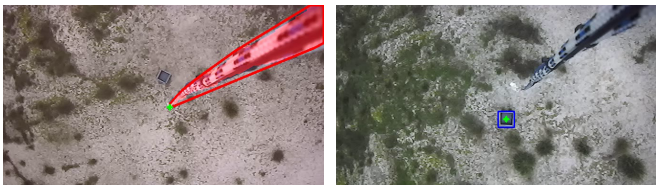
Besides the load state, an estimate of the target state is a prerequisite to achieve the tracking of the ground target and the successful release of the load towards it. Similarly to Section IV-A, a CNN is trained in order to detect the target which moves in an unknown manner.

However, when the load is located above the target, a part of the area, which corresponds to the target, is visually occluded by the load and, as a result, the CNN fails to detect the target consecutively. Hence, a tracker is additionally exploited so as to achieve the detection of the target in a more robust and faster manner. Among the various trackers presented in the literature, the KCF tracker [12] is selected due to its high speed and accuracy and its ability to track the target in the presence of partial occlusions.

The pipeline of the detection is as follows:

1) Extract a ROI by utilizing the trained CNN,
2) Based on the aforementioned ROI, deploy the KCF tracker and
3) Re-detect the target using the CNN after some iterations in order to reduce the accumulated error or when the tracker fails.

Based on the aforementioned pipeline, a ROI, which corresponds to the target, is extracted by means of a bounding box, as depicted in Figure 3b. The center $(u_T, v_T)$ of the bounding box defines the pixel coordinates of the target in the image plane. Since the target moves on the ground, its known height ${}^Iz_T = h$ is exploited in order to compute its 3D position. Consequently, after suitable transformations, an estimate of the target position ${}^I\mathbf{p}_T = \begin{bmatrix} {}^Ix_T & {}^Iy_T & {}^Iz_T \end{bmatrix}^T$ is available at each time instant.



(a) (b)

Fig. 3: (**a**) Detection of the cable. The red-shaded transparent area, surrounded by a convex hull, denotes the detection mask overlaid on top of the original image. The green cross corresponds to the pixels of the load. (**b**) Detection of the target. The green cross corresponds to the center of the bounding box.

### B. Target Motion Prediction

However, the estimate of the target position is not adequate for the deployed control scheme, since the NMPC requires both the position and velocity of the target not only at the current time instant but also throughout the predicted time horizon. In order to predict the future trajectory of the target, Bézier curves are exploited, as proposed in [15].

More precisely, a Bézier curve is defined, based on the order $n$ of the curve, as follows:

$$\mathbf{B}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\mathbf{c}_i \tag{11}$$

where $\tau \in [0,1]$, $\mathbf{B}(\tau) \in \mathbb{R}^2$ (since the target is moving on the x-y plane) is the Bézier curve, $b_{i,n}(\tau), i = 0, \cdots, n$ are the Bernstein basis polynomials of degree $n$ and $\mathbf{c}_i = \begin{bmatrix} c_{x,i} & c_{y,i} \end{bmatrix}^T \in \mathbb{R}^2$ are the $n+1$ control points.

In order to estimate and predict the target motion, it is essential to find the suitable set of control points $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \cdots, \mathbf{c}_n]$. Towards this direction, assuming that $N$ estimates of the target position ${}^I\mathbf{p}_T$ are available according to Section V-A, along with the corresponding timestamps $t$, a set $D = \{(t_1, {}^Ix_{T,1}, {}^Iy_{T,1}), \cdots, (t_N, {}^Ix_{T,N}, {}^Iy_{T,N})\}$ of constant length is constructed, where the timestamp $t_N$ corresponds to the current time. The aforementioned set $D$ is updated once a new measurement is available. Since the aim is to predict the motion of the target up to the future timestamp $t_p$, the following mapping is performed between the timestamps $t \in [t_1, t_p]$ and the parameter $\tau \in [0,1]$ of the Bézier curve:

$$\tau = \frac{t - t_1}{t_p - t_1} \tag{12}$$

The optimal set of control points is computed according to the following optimization problem:

$$\min_{\mathbf{c}} \sum_{i=1}^{N} \left( w_{t_i} \|\mathbf{B}(\tau_i) - {}^I\mathbf{p}_T(t_i)\|_2^2 \right) \tag{13}$$

where the objective is to minimize the distance between the measurements and the Bézier curve with $w_{t_i}$ denoting weights which penalise past measurements [15].

Additionally, in order to identify feasible solutions, constraints are imposed on the above optimization problem. The aforementioned constraints arise from the dynamic limits of the target according to which the predicted velocity and acceleration should lie inside $\left[-u_{max}, u_{max}\right]$ and $\left[-a_{max}, a_{max}\right]$ respectively. By differentiating the Bézier curve, the following constraints are defined for the $x$ axis (similarly for the $y$ dimension):

$$\begin{aligned} -u_{max} \leq n \cdot (c_{x,i} - c_{x,i-1}) / (t_p - t_1) \leq u_{max} \\ -a_{max} \leq n \cdot (n-1) \cdot (c_{x,i} - 2c_{x,i-1} + c_{x,i-2}) / (t_p - t_1)^2 \leq a_{max} \end{aligned} \tag{14}$$

The above constrained Quadratic Programming problem with $2 \cdot (n+1)$ primal variables, i.e., the number of control points, and $2 \cdot n + 2 \cdot (n-1)$ inequality constraints is solved using the OOQP software [23] and, eventually, an estimate of the future trajectory of the target, i.e., ${}^I\mathbf{p}_T(t)$ and ${}^I\mathbf{v}_T(t)$, is available throughout the predicted horizon $\left[t_N, t_p\right]$.

## VI. Nonlinear MPC for Target Tracking

In order to efficiently track the ground target, an NMPC is formulated for the nonlinear system dynamics defined by Eq. 7. Since the objective of the NMPC scheme is to track the ground target and eventually release the cable-suspended load towards the target, the following cost functions are defined:

$$\begin{aligned} J_p(t) &= \left\| {}^I\mathbf{p}(t) + l\,{}^I\mathbf{n}(t) - {}^I\mathbf{p}_T(t) - \begin{bmatrix} 0 & 0 & z_{des} \end{bmatrix}^T \right\|^2_{Q_p} \\ J_v(t) &= \left\| {}^I\mathbf{v}(t) + l\,{}^I\dot{\mathbf{n}}(t) - {}^I\mathbf{v}_T(t) \right\|^2_{Q_v} \\ J_o(t) &= \left\| \begin{bmatrix} \phi(t) & \theta(t) & \psi(t) - \psi_{des} \end{bmatrix}^T \right\|^2_{Q_o} \\ J_L(t) &= \left\| \begin{bmatrix} \eta_L^T(t) & \dot{\eta}_L^T(t) \end{bmatrix}^T \right\|^2_{Q_L} \end{aligned} \tag{15}$$

where $Q_p \in \mathbb{R}^{3\times3}$, $Q_v \in \mathbb{R}^{3\times3}$, $Q_o \in \mathbb{R}^{3\times3}$ and $Q_L \in \mathbb{R}^{4\times4}$ are diagonal weighting matrices and $J_p$, $J_v$, $J_o$ and $J_L$ are task-specific cost functions which penalize the distance between the load and the target, the relative velocity between the load and the target, the error between the current orientation of the vehicle and the desired one, and the swinging motion of the load. It is mentioned that the UAV should track the ground target while maintaining a certain altitude above it, defined by the variable $z_{des}$. The desired yaw orientation $\psi_{des}$ is maintained constant. Additionally, the swinging motion of the load is minimized so as to avoid unnecessary oscillations and ensure the safety of the load to be delivered.

Considering the total cost function $J_t = J_p + J_v + J_o + J_L$ and the input constraints, the following Optimal Control Problem (OCP) is formulated throughout the horizon $\left[t_N, t_p\right]$:

$$\begin{aligned} &\min_{\upsilon} \int_{t_N}^{t_p} \left( J_t(t) + \|\upsilon(t)\|^2_R \right) \mathrm{d}t + J_t(t_p) \\ &\text{s.t.: } \mathbf{x}(\mathbf{t_N}) = \mathbf{x_N},\ \dot{\mathbf{x}} = f(\mathbf{x}, \upsilon) \text{ (Eq. 7)},\ \upsilon \in \mathbb{U} \end{aligned} \tag{16}$$

where $\mathbf{x} = \begin{bmatrix} {}^I\mathbf{p}^T & {}^I\mathbf{v}^T & \phi & \theta & \psi & \eta_L^T & \dot{\eta}_L^T \end{bmatrix}^T \in \mathbb{R}^{13}$, $\upsilon = \begin{bmatrix} \phi_{ref} & \theta_{ref} & \psi_{ref} & {}^I v_{z_{ref}} \end{bmatrix}^T \in \mathbb{R}^4$ are the state and input vectors, $R$ is the control input cost matrix and, $\mathbb{U}$ is the set of input constraints specified by the roll, pitch and vertical velocity limits.

The aforementioned OCP is solved at each iteration $t_N$, given the predicted target trajectory, and the first control input of the derived control sequence is sent to the inner attitude control loop of the autopilot. The ACADO Toolkit [24], along with the qpOASES solver [25], are utilized for designing and solving in real time the presented OCP.

## VII. Release Condition

Regarding the release condition, at each current timestamp $t_N$, the trajectory of the load is computed assuming that is released. In this case, the load follows a ballistic trajectory, i.e. moves only under the influence of gravity, according to the following equation:

$${}^I\mathbf{p}_L(t) = {}^I\mathbf{p}_L(t_N) + {}^I\mathbf{v}_L(t_N) \cdot t - g\mathbf{e_3} \cdot t^2/2, \tag{17}$$

Given the target's height ${}^I z_T$ and the above equation, the timestamp $t_h$ at which the trajectory of the load intersects with the ${}^I z_T$ is computed. Consequently, the release condition is formulated as follows:

$$\left\| {}^I\mathbf{p}_L(t_h) - {}^I\mathbf{p}_T(t_h) \right\| \leq d \tag{18}$$

where $d$ is the maximum horizontal distance between the load and the target. It is mentioned that the condition implies that the timestamp $t_h$ lies inside the predicted horizon of the target motion, i.e., $t_h \in \left[t_N, t_p\right]$, and, hence, an estimate of the target motion is available. Otherwise, the release condition is not checked.

## VIII. Results

### A. Experimental Setup

The proposed scheme is evaluated experimentally while using an octorotor in an outdoor environment. The state of the octorotor is provided by the Ardupilot firmware, which fuses measurements by typical navigation sensors via an Extended Kalman Filter. Moreover, the UAV is equipped with the embedded computer Jetson AGX Xavier [26], which communicates with the autopilot through the MAVLink protocol [27]. The control of the vehicle is achieved with the aid of Robot Operating System (ROS) [28], specifically the MAVROS node, an intermediate node between ROS and Ardupilot. Regarding the UAV's sensor suite, the octorotor is additionally equipped with the ZED 2 camera, a load cell and a servo motor which controls the gripper. A base is 3D printed in order to assemble the load cell and the gripper, both located exactly above the load. Both the load cell and the servo motor are directly connected with an Arduino Uno, which communicates serially with the Jetson. As for the load, a bag containing a 0.5 kg load is considered. Finally, the ground target is a UGV Robotnik Summit which is manually driven towards random directions through joystick teleoperation.

As for the computational cost of the proposed framework, the mean processing time, on the powerful Jetson, of the individual parts, that comprise the control scheme, is demonstrated in Table I. It is evident that a real-time performance is achieved during the tracking of the UGV.

| Step | Time[ms] |
|---|---|
| CNN | 32.41 |
| KCF | 5.27 |
| Bézier regression ($N = 20$) | 0.93 |
| NMPC | 3.33 |

TABLE I: The computational cost of the individual parts that comprise the proposed framework.

### B. Experimental Results

In order to validate the ability of the proposed control strategy to uninterruptedly track the UGV, an outdoor experiment was initially conducted without considering the release of the load. Throughout the first experiment, the UGV moved with an average horizontal velocity of $1.5m/s$ and a maximum equal to $2.5m/s$, as estimated by the Bézier regression (Figures 4c, 4d). The deployed vision-based NMPC scheme, which communicates directly with the inner attitude subsystem of the UAV, enables the vehicle to respond quickly to the unknown maneuvers of the target, performed by the operator, and, hence, the continuous tracking of the target was achieved, according to Figures 4a, 4b, where the trajectories of both vehicles in the horizontal plane are depicted. Additionally, despite the agile motion of the UGV, the swinging motion of the cable-suspended load, obtained by the UKF, was not excited and the load angles were maintained below $20^o$, as illustrated in Figure 4e.

Afterwards, a second experiment was realized in order to evaluate the ability of the proposed scheme to successfully release the suspended load towards the target. The UAV initially tracked the vehicle until the release condition of Eq. 18 was satisfied, as depicted in Figure 5a. At that moment, the opening of the gripper was triggered and the load was successfully released and placed into a box, carried by the UGV, as illustrated in Figure 5.

## IX. CONCLUSIONS

In this work, a complete framework is presented for tracking a ground target while using a multirotor UAV with a cable-suspended load. The proposed method relies only on the integration of a downward-looking camera and a load cell, required for the target motion prediction and the load state estimation, and on the existence of an inner-loop controller. Moreover, a gripper is incorporated into the vehicle's sensor suite in order to achieve the release of the load towards the target and, thus, increase the applicability of our method for a variety of scenarios. The efficacy of the whole scheme was validated experimentally.

Regarding our future work, we intend to utilize a forward-looking camera or a LiDAR, so as to consider the presence of obstacles and, generalize our framework.

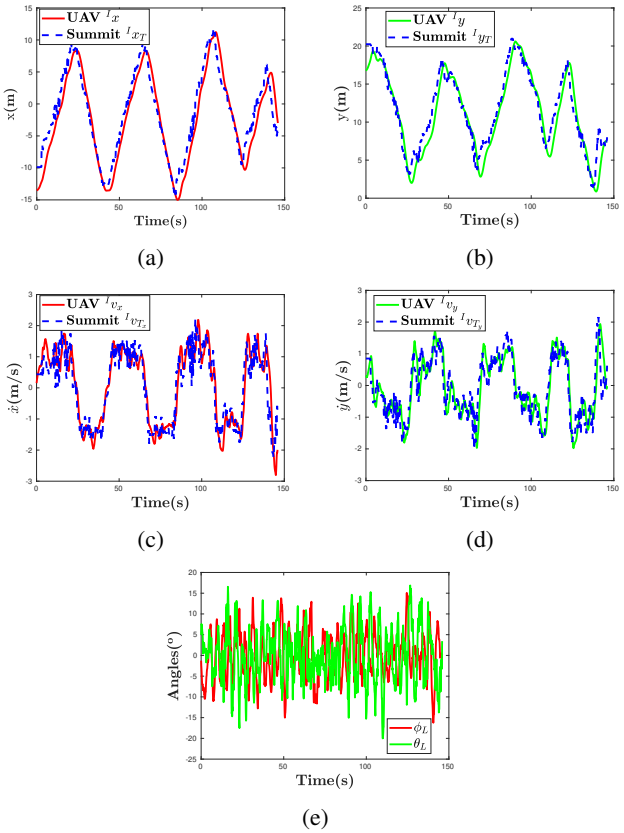



Fig. 4: The position ${}^I\mathbf{p}$ and the velocity ${}^I\mathbf{v}$ of the UAV during the first experiment compared to the UGV motion, i.e., ${}^I\mathbf{p}_T$ and ${}^I\mathbf{v}_T$, as estimated by the Bézier regression. (**a**) Position x. (**b**) Position y. (**c**) Velocity x. (**d**) Velocity y. (**e**) The load angles $\eta_L = \begin{bmatrix}\phi_L & \theta_L\end{bmatrix}^T$.

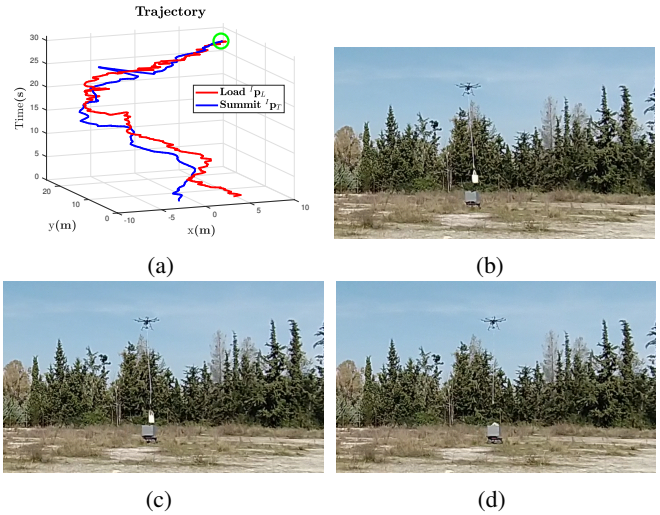



Fig. 5: The successful release of the load towards the target during the second experiment. (**a**) The position of the load ${}^I\mathbf{p}_L(t_h)$, assuming that is released, compared to the position of the target ${}^I\mathbf{p}_T(t_h)$. The green circle indicates that the release condition (Eq. 18) is satisfied. (**b**) The gripper opens. (**c**) The load follows a ballistic trajectory. (**d**) The load is placed inside the box.

## REFERENCES


[1] F. Panetsos, G. C. Karras, S. N. Aspragkathos, and K. J. Kyriakopoulos, "Precise position control of a multi-rotor uav with a cable-suspended mechanism during water sampling," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 1780–1786.

[2] D. Villa, A. Brandao, and M. Sarcinelli-Filho, "A survey on load transportation using multirotor uavs," *Journal of Intelligent & Robotic Systems*, vol. 98, pp. 1–30, 05 2020.

[3] X. Liang, Y. Fang, N. Sun, and H. Lin, "Nonlinear hierarchical control for unmanned quadrotor transportation systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3395–3405, 2018.

[4] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. García, "Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6718–6723.

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

[6] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4902–4909.

[7] F. Panetsos, G. C. Karras, and K. J. Kyriakopoulos, "A deep reinforcement learning motion control strategy of a multi-rotor uav for payload transportation with minimum swing," in *2022 30th Mediterranean Conference on Control and Automation (MED)*, 2022, pp. 368–374.

[8] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2216–2222.

[9] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[10] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Robotics: Science and Systems*, 2017.

[11] H. Cheng, L. Lin, Z. Zheng, Y. Guan, and Z. Liu, "An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1732–1738.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[13] L. Johnston and V. Krishnamurthy, "An improvement to the interacting multiple model (imm) algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 2909–2923, 2001.

[14] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 446–453.

[15] Z. Han, R. Zhang, N. Pan, C. Xu, and F. Gao, "Fast-tracker: A robust aerial system for tracking agile target in cluttered environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 328–334.

[16] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5249–5256.

[17] F. Panetsos, G. C. Karras, K. J. Kyriakopoulos, O. Oikonomides, P. Kolios, D. Eliades, and C. Panayiotou, "A motion control framework for autonomous water sampling and swing-free transportation of a multirotor uav with a cable-suspended mechanism," *Journal of Field Robotics*, vol. 40, no. 5, pp. 1209–1230, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22182

[18] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[19] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317313083

[20] A. P. group. (2016) Ardupilot documentation. [Online]. Available: https://ardupilot.org/ardupilot/

[21] D. Gupta, "A beginner's guide to deep learning based semantic segmentation using keras," 2019. [Online]. Available: https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html

[22] K. Wada, "labelme: Image Polygonal Annotation with Python," https://github.com/wkentaro/labelme, 2016.

[23] E. M. Gertz and S. J. Wright, "Object-oriented software for quadratic programming," *ACM Trans. Math. Softw.*, vol. 29, no. 1, p. 58–81, mar 2003. [Online]. Available: https://doi.org/10.1145/641876.641880

[24] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[25] J. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, 12 2014.

[26] "Jetson Xavier," https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit.

[27] M. M. A. V. Protocol. (2013) Mavlink to ros gateway with proxy for ground control station. [Online]. Available: https://github.com/mavlink/mavros

[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.