Variable Time-step MPC for Agile Multi-rotor UAV Interception of Dynamic Targets

Atharva Ghotavadekar[1], František Nekovář[2], Martin Saska[2], Jan Faigl[2]

***Abstract*—Agile trajectory planning can improve the efficiency of multi-rotor Uncrewed Aerial Vehicles (UAVs) in scenarios with combined task-oriented and kinematic trajectory planning, such as monitoring spatio-temporal phenomena or intercepting dynamic targets. Agile planning using existing non-linear model predictive control methods is limited by the number of planning steps as it becomes increasingly computationally demanding. That reduces the prediction horizon length, leading to a decrease in solution quality. Besides, the fixed time-step length limits the utilization of the available UAV dynamics in the target neighborhood. In this paper, we propose to address these limitations by introducing variable time steps and coupling them with the prediction horizon length. A simplified point-mass motion primitive is used to leverage the differential flatness of quadrotor dynamics and the generation of feasible trajectories in the flat output space. Based on the presented evaluation results and experimentally validated deployment, the proposed method increases the solution quality by enabling planning for long flight segments but allowing tightly sampled maneuvering.**

***Index Terms*—Aerial Systems: Applications, Motion and Path Planning, Autonomous Vehicle Navigation**

I. Introduction

THE task of multi-target visit, monitoring, and interception planning under a flight budget constraint can be modeled as a variant of the *Orienteering Problem* (OP) [1]. The OP is to maximize the collected reward by visiting a known set of targets with the associated rewards within the given limited travel budget; thus, it can be thought of as a combination of two NP-hard combinatorial problems, the *Travelling Salesman Problem* (TSP) and Knapsack problem [2]. Solutions to the OP generalizations [3], including the Team OP [4], Set OP [5], and Dubins OP[6] can be effectively transferred to real-world use-cases by addition of constraints adhering to the specific vehicle dynamics. In particular, the *Kinematic OP* (KOP) [7], [8] with limited velocity and acceleration-bound point-mass is especially suited for the dynamics of the multi-rotor *Uncrewed Aerial Vehicles* (UAVs). Furthermore, a spatio-temporal reward-collection model is suitable for addressing the real-world dynamics of the targets to be visited and thus improving vehicle utilization under flight time constraints.

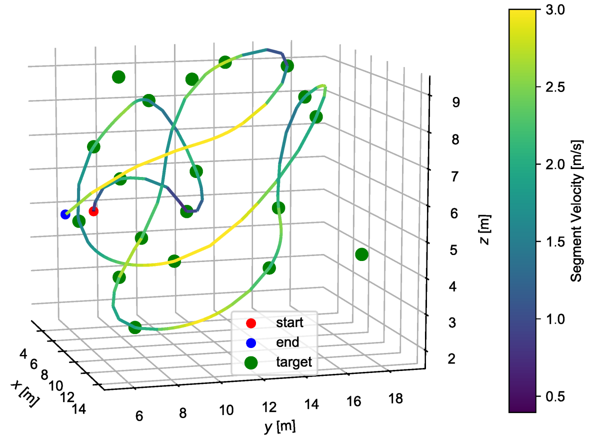


Figure 1. A solution to a 3D variant of the Tsiligirides 2 benchmark instance of the KOP determined by the proposed variable time-step method with the travel budget of 40 s with the visualization of the vehicle velocity for each prediction step.

The herein studied problem is motivated by dynamic environment monitoring formulated as a 3D generalization of the KOP using *Model Predictive Control* (MPC) [9]. The environment is defined by a set of target reward locations placed in a 3D environment representation, each location with the associated reward gain value collected when the location is visited. However, motivated by data gathering tasks [10], such as monitoring forest fires [11], coastal regions [12], terrain surveillance [13], and urban inspections [14], we are accounting for locations and their associated information gained that are highly dynamic due to changing environmental conditions. Hence, the targets' dynamics is modeled in the proposed solution of the addressed variant of the KOP. Since the target dynamics estimation is considered beyond the scope of the paper, it is assumed to be known, provided by an external system or an independent estimator.

The necessity to account for target dynamics puts high demands on trajectory planning, where non-linear MPC-based methods suffer from the fixed length time-step, which significantly increases the computational burden for an increased planning horizon. Since these methods can exploit the vehicle dynamics, we propose a variable time-step approach to increase the length of the prediction horizon without increasing the number of prediction steps. Based on the evaluation results for benchmark instances of the KOP, the proposed method provides improved solutions compared to the existing methods. Furthermore, the practical applicability of the method has been experimentally validated in real-field deployments.

The rest of the paper is organized as follows. An overview

This work was supported by the Czech Science Foundation (GAČR) under research projects No. 22-05762S and No. 22-24425S, the CTU grant No. SGS23/177/OHK3/3T/13 and by the European Union under the project Robotics and advanced industrial production No. CZ.02.01.01/00/22_008/0004590.

[1] Author is with the Department of Electrical and Electronics Engineering, BITS Pilani K.K. Birla Goa Campus, India f20201300@goa.bits-pilani.ac.in

[2] Authors are with the Czech Technical University, Faculty of Electrical Engineering, Technicka 2, 166 27, Prague, Czech Republic, {nekovfra|saskam1|faiglj}@fel.cvut.cz

of the related methods is provided in the following section. The problem is formally introduced in Section III together with the models of the target and reward dynamics, vehicle model, and the utilized cost function. The proposed online replanning is presented in Section IV.

II. Related Work

The studied problem combines trajectory planning with routing, where a multi-point trajectory needs to be determined for a particular sequence of visits to the target regions. Since such a trajectory is sequence-dependent, agile trajectory planning depends on trajectory parameterization, allowing the exploitation of vehicle dynamics. Pioneered quadrotor trajectory generation methods are minimum snap polynomial approaches based on the differential flatness of quadrotor dynamics [15]. B-spline [16], [17] and Bézier curve [18], [19] based approaches produce spatial polynomial paths that require mapping to time domain [20], [21]. In [20], it is shown that if the initial and final velocities are non-zero, a feasible solution may not exist with the aforementioned approaches.

On the other hand, routing problems, as a variant of the TSP [22], are required to adhere to real-world time and/or energy budget constraints and are modeled as the OP [1]. Dubins vehicle is a simplified motion model with constant velocity constrained by a minimum turning radius and a maximum radial acceleration already deployed in Dubins routing problems [6], [23]. However, the advantage of energy efficiency in Dubins paths is offset by possible significant detours, and the model's restriction is more relevant for fixed-wing UAVs than for agile multi-rotor UAVs.

Time optimal trajectories based on Pontryagin's minimum principle that keeps control input at their limits in a bang-zero-bang profile can be calculated axis-wise and then synchronized [24]. The authors of [25] found that an analytical solution for axis synchronization does not exist for certain instances for a jerk-input point-mass vehicle model. In [7], [8], it is further shown that even for the acceleration-input point-mass model, axis synchronization sometimes results in failures due to improper consideration of the robots' inertia, and a method to overcome the synchronization failure is proposed. However, the control input is limited by axis-wise kinematic constraints, the aggregation of which results in a n-dimensional cuboid. In [26], the authors optimize for segment-wise sampling time for a pre-assigned fixed number of time steps and trajectories passing through fixed waypoints. However, the approach does not address the selection of waypoints for maximum reward collection or moving-horizon planning.

The proposed approach overcomes the aforementioned limitations to maximize the reward collection and utilization of the multi-rotor agility in static routing problems and continual monitoring tasks. We propose to utilize the full range of feasible kinematics, equivalent to an n-dimensional sphere with the radius v_{max} and a_{max}, while enforcing velocity and acceleration constraints, respectively. It enables maximum utilization of multi-rotor kinematic constraints as shown in Fig. 2. The contributions are considered as follows.

- 3D KOP generalization with a viable jerk-input constrained trajectory planner for multi-rotor UAV platforms.
- A novel optimal control formulation called the *Variable Time-step MPC* (VT-MPC), where the prediction time-step is a part of the variables optimized over the control horizon in reward-collection planning, which improves performance in the KOP and related receding-horizon monitoring problems.
- The VT-MPC formulation-based planning and executing optimal reward-collecting trajectories for monitoring tasks in real-world flight that leverages the differential flatness property of the quad-copter dynamics combined with a simplified motion primitive.
- Comparative evaluation of the proposed method and state-of-the-art methods to the KOP showing improved performance along with the online re-planning ability of the proposed formulation in a receding-horizon planning.
- Implementation of the method as open source code supporting replicability of the reported results.

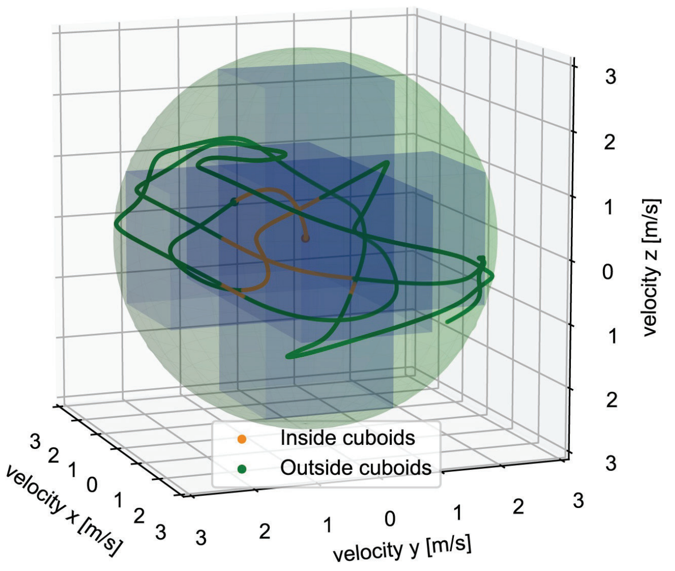


Figure 2. Analysis of the velocity profile from the trajectory depicted in Fig. 1 with respect to (w.r.t.) the kinematic constraints imposed by [8] (blue cuboids) and constraints imposed by the proposed method (green sphere). About 84.4 % of the velocity profile (green trajectory) lies outside the cuboidal constraints but within the proposed spherical constraints.

III. Proposed Models and Problem Formulation

The studied problem and the proposed novel VT-MPC formulation are inspired by previous work on water surface monitoring [9]. Therefore, we follow therein introduced notation further extended to address the VT-MPC.

- t_k is the temporal length of the kth prediction step.
- N_t is the number of targets in the 3D environment.
- N_h is the number of the prediction steps.
- $\mathbf{p}_k$ is the position vector denoting the location of the UAV at the kth step.
- $\mathbf{q}_{r,k}$ is the position vector of the rth target at the kth step.
- $g_{r,k}$ is the reward gain value associated with the rth target at the kth step,
- $\mathbf{x}_k$ denotes the state of the UAV at the kth step.
- $\mathbf{u}_k$ is the input of the model at the kth step.

In the rest of the section, the formal models are introduced for target and reward dynamics (Section III-A), vehicle modeling in Section III-B, and cost function in Section III-C. The proposed problem formulation is presented in Section III-D.

A. Target and Reward Dynamics

The target locations with the associated reward gains at the prediction step k are denoted by the vectors $\mathbf{q}_k$ and $\mathbf{g}_k$, respectively. For the rth target, $\mathbf{q}_{r,k} = [q^x_{r,k}, q^y_{r,k}, q^z_{r,k}]$ represents the target's position vector, and $\mathbf{q}_k = [\mathbf{q}_{1,k}, \dots, \mathbf{q}_{N_t,k}]$ represents combined target position vector. Thus, the Euclidean distance of the UAV at the position $\mathbf{p}_k$ from the target $\mathbf{q}_r$ at the kth prediction step can be computed as

$$d_{r,k}(\mathbf{p}_k, \mathbf{q}_{r,k}) = \sqrt{(p^x_k - q^x_{r,k})^2 + (p^y_k - q^y_{r,k})^2 + (p^z_k - q^z_{r,k})^2}. \tag{1}$$

Assuming that target position dynamics are known, we denote the target position vector at the prediction step k as $\mathbf{q}_k$.

The information collecting sensor is modeled using the differentiable Butterworth function denoted by f_b, with the order n_b affecting the function shape and cutoff parameter c_b that models the approximate sensor collection range as

$$f_b(x) = \frac{1}{1 + \left(\frac{x}{c_b}\right)^{n_b}}. \tag{2}$$

The dynamics of the reward gain dynamic state $g_{r,k}$ associated with the rth target at the kth step is determined as

$$g_{r,k+1} = \Big(g_{r,k} + \alpha_g t_k\Big) \cdot \Big(1 - f_b(d_{r,k})\Big), \tag{3}$$

where α_g is the rate of linear increase of the reward gain in time. On the UAV visit of the rth target at the kth step, the distance value $d_{r,k} = 0$ causes the term $1 - f_b(d_{r,k})$ to be nullified; hence, the reward is assumed as fully collected. The used model of information dynamics (to be collected within neighborhoods of discrete points') allows formulating the multiple-target monitoring task as an MPC, with a cost function minimizing the reward gain values of all the targets with regard to the modeled vehicle dynamics. The reward gain is modeled to increase with time when uncollected, to facilitate formulation of persistent monitoring tasks.

B. Vehicle Model

The utilized vehicle model is based on a point-mass motion primitive that is able to produce dynamically feasible UAV trajectories. States and inputs to our discretized algebraic system are considered as the differentially flat output states of multi-rotor dynamics [15]. The UAV state $\mathbf{x_k}$ at the kth prediction step is denoted

$$\mathbf{x}_k = [\mathbf{p}_k, \mathbf{v}_k, \mathbf{a}_k, \psi_k], \tag{4}$$

where the respective elements are position, velocity, acceleration vectors, and heading. We introduce the variable time-step t_k as a system input

$$\mathbf{u}_k = [\mathbf{j}_k, \dot{\psi}_k, t_k], \tag{5}$$

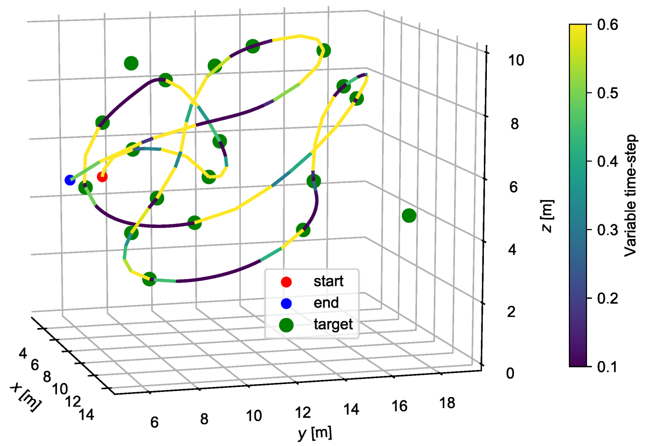


Figure 3. Example of the variable time-step lengths distribution for a data collecting trajectory of the Tsiligirides 2 instance extended to 3D. The optimal solution utilizes shorter time-steps in areas of reward collection and frequent input changes due to maneuvering.

where the respective input elements are jerk, heading rate, and the time-step length optimization variable, which is also referred to as the variable time-step.

An example of the variable time step for a vehicle trajectory is depicted in Fig. 3. The discretized point-mass model with the jerk input is used for state propagation according to (6).

$$\begin{array}{rcl} \mathbf{a}_{k+1} & = & \mathbf{a}_k + \mathbf{j}_k \cdot t_k \\ \mathbf{v}_{k+1} & = & \mathbf{v}_k + \mathbf{a}_k \cdot t_k + \frac{1}{2} \cdot \mathbf{j}_k \cdot t_k^2 \\ \mathbf{p}_{k+1} & = & \mathbf{p}_k + \mathbf{v}_k \cdot t_k + \frac{1}{2} \cdot \mathbf{a}_k \cdot t_k^2 + \frac{1}{6} \cdot \mathbf{j}_k \cdot t_k^3 \\ \psi_{k+1} & = & \psi_k + \dot{\psi}_k \cdot t_k \end{array} \tag{6}$$

C. Cost Functions

The sum of the reward gains for all targets is directly a part of the MPC cost function. Besides, the heading cost $c_\psi(\psi_k)$ is included in the cost function since the multi-rotor vehicle allows for independent heading control, which can be added to the perception constraints to meet the mission objectives. Thus, $c_\psi(\psi_k)$ is defined as the negative of the dot product of the UAV velocity with the current heading in the X-Y plane to ensure the UAV faces its direction of motion. The change in the input is penalized by a scalar c_u. The cost function c_k at the prediction step k is thus defined as

$$c_k = \sum_{r=0}^{N_t} (g^r_k) + k_\psi \cdot c_\psi(\psi_k) + \Delta\mathbf{u}_k^T c_u \Delta\mathbf{u}_k, \tag{7}$$

where k_ψ is a tunable scaling parameter. The summation term denotes the total uncollected reward at the kth step.

A soft constraint on the final position is imposed via the terminal cost c_f defined as

$$c_f = k_f \cdot d(\mathbf{p}_{N_h}, \mathbf{p}_f), \tag{8}$$

where $\mathbf{p}_f$ is the desired final UAV position.

D. Optimal Control Problem

The combined state vector of the system $\mathbf{X}_k$ is defined in (9) as the concatenated vector of the reward positions $\mathbf{q_k}$, reward gains $\mathbf{g_k}$, the UAV state vector $\mathbf{x_k}$, and the scalar $t_{\text{sum},k} = \sum_{i=0}^{k} t_i$ denoting the total trajectory time (temporal length) up to the prediction step k.

$$\mathbf{X}_k = [\mathbf{q}_k, \mathbf{g}_k, \mathbf{x}_k, t_{\text{sum},k}] \tag{9}$$

The optimal control problem is formulated as the VT-MPC with the prediction horizon of N_h steps

$$\begin{aligned} \min_{\mathbf{X},\mathbf{u}} \quad & \sum_{k=0}^{N_h} (c_k(\mathbf{X}_k, \mathbf{u}_k)) + c_f \\ \text{s.t.} \quad & \|\mathbf{v}_k\| \leq \text{v}_{\text{max}} \\ & \|\mathbf{a}_k\| \leq \text{a}_{\text{max}} \\ & -\text{j}_{\text{max}} \leq \mathbf{j}_k \leq \text{j}_{\text{max}} \\ & -\dot{\psi}_{\text{max}} \leq \dot{\psi} \leq \dot{\psi}_{\text{max}} \\ & \text{t}_{\text{min}} \leq t_k \leq \text{t}_{\text{max}} \\ & t_{\text{sum},N_h} = \text{C}_{\text{max}} \end{aligned} \quad . \tag{10}$$

The UAV platform's specific kinematic bounds are v_{max}, a_{max}, j_{max}, and $\dot{\psi}_{\text{max}}$ on the velocity, acceleration, acceleration jerk, and heading rate, respectively. Time-step length bounds are t_{min} and t_{max}, and the final equation constrains the travel budget C_{max}.

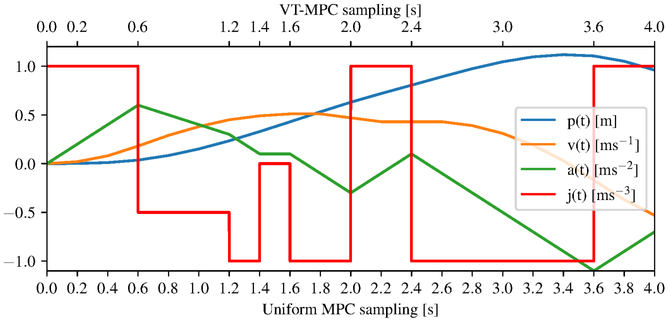


Figure 4. An example single-dimensional jerk-input trajectory where time-step sampling of the VT-MPC is shown as the top x-axis and of the MPC as the bottom x-axis. With the MPC times-step of 0.2 s, the formulation requires 20 prediction steps to reach full fidelity on the prediction horizon 4 s. The VT-MPC set at $t_{\text{min}} = 0.2$ s, $t_{\text{max}} = 0.6$ s and $t_{\text{sum}} = 4$ s utilizes only 10 prediction steps. The first optimization time-step is always constrained at t_{min}.

The difference between standard MPC and the proposed VT-MPC planning method is illustrated in Fig. 4. The MPC models a discrete dynamical system where inputs to the system can change only at fixed time-steps. Assuming a mass-point vehicle model where all dynamical states are constrained, we note the optimization can converge to a situation where constraints on the input or some dynamical state are reached, leading to constant or zero input value respectively on multiple subsequent times-steps. It is supported by the observation that a time-optimal position-velocity trajectory neglecting higher-order dynamics would intercept targets using straight lines, and is a lower-bound to our more complex trajectory with gap decreasing with higher target distances and/or tighter constraints. We take advantage of the observation by formulating the variable time-step, where limited number of prediction steps are optimized on the same time horizon.

IV. Proposed Online Re-planning

The proposed problem formulation can be used for offline trajectory planning and online re-planning for real-world tasks in the presence of target dynamics, including reward gain dynamics. However, the prediction horizon N_h and the travel budget C_{max} need to be adjusted to fit the receding horizon fashion for continual monitoring tasks. Besides, a new parameter is introduced to define the re-planning period t_{replan} to define the maximum time used for the onboard re-planning.

The targets to be monitored exhibit known spatio-temporal dynamics that the planner uses for precise interception. In the absence of the target state measurement, a simulator runs onboard to provide the reward state estimates (the target locations and their reward gains). The optimizer is bound by a maximum allowed planning time equal to 85 % of t_{replan} to converge. The reserve in the amount of 15 % enables other computations and control routines within the onboard system. Once a trajectory is planned, it is sent to the trajectory tracker [27] of the employed MRS UAV System [28].

Algorithm 1: Proposed Online re-planning.

Input: t_{replan}– re-planning time in seconds, $\mathbf{x}_0$ – initial model state, n_{reuses} – maximum number of trajectory reuses.

Data: c – boolean indicating if the solution converged, c_{use} – trajectory reuse counter, $\mathbf{x}$ – the UAV state vector (4), S_{p} – planned trajectory.

```
1  c, $S_p$ ← optimize($\mathbf{x}_0, \infty$)   // Initial planner call.
2  $c_{use}$ ← 1
3  while $c_{use} \leq n_{reuses}$ do
4      if not c then
5          ▷ Since the solution was not found in the
             previous iteration, continue tracking the
             previous trajectory for $t_{replan}$ seconds.
6          $c_{use}$ ← $c_{use}$+1   // Increment use counter.
7      else
8          $c_{use}$← 1   // Solution found, reset counter.
9          set_tracker($S_p$)   // Track the trajectory
10     ▷ Set the state for the next iteration.
11     $\mathbf{x}$ ← $S_p[c_{use} \cdot t_{replan}]$
12     c, $S_p$ ← optimize($\mathbf{x}, t_{replan}$)
```

The online re-planning is summarized in Algorithm 1. It works in iterations, where a found trajectory is reused up to n_{reuses} times if a new trajectory S_{p} is not found in t_{replan} (Algorithm 1, Line 12). Once the tracker is set, it follows the trajectory unless a new trajectory is set.

The trajectory tracker produces a full state reference to be tracked by a low-level controller. While the planned trajectory is being tracked by the UAV, the UAV state in t_{replan} seconds in the future is used as the planner input for the next iteration. That ensures continuity of the trajectories produced, despite the rate of re-planning being significantly lower than the rate of the onboard low-level control loop that is set to 100 Hz. In the case a solver does not converge within the given planning time, the previously obtained trajectory is kept for the next period.

In the case the previously calculated trajectory is followed to its end, the UAV stops and hovers at its current position to keep re-planning till a solution is found again. Note that the future state reference and previously found solution can be used as an initial condition to warm-start the optimization by the `do_mpc` framework [29] in the next iteration, thus reducing the computational burden.

The proposed online re-planning has been validated in simulation and real-world deployments, with the results reported in the following section.

V. Results

The performance of the proposed approach has been evaluated in multiple simulations and real-world deployments to examine the impact of introducing the variable time step and to verify the dynamic feasibility of the trajectories generated. 2D and 3D KOP instance trajectories are computed offline on a consumer-grade desktop computer. The real-world feasibility and online re-planning ability is verified with multiple Gazebo simulations and on-field experiments using a UAV research platforms based on the T650, X500 and f450 UAV frames [30] using the MRS UAV System [28] running on the Intel NUC i7-8559U onboard computer, see real-world deployment snapshots in Fig. 5. The UAV is interfaced via *Robot Operating System* (ROS) and utilizes a standard GPS for localization. All the computations are done onboard for real-time experiments involving re-planning.

Figure 5. Snapshots of the UAV performing the monitoring mission in a field test. The UAV is encircled in each image.

The employed MPC framework [29] with IPOPT [31] solver and the MA97 linear solver [32] produces a system state prediction corresponding to the optimal control input over the prediction/control horizon. The predicted UAV positions in the system state prediction are sent to the MRS UAV System interface for trajectory tracking after a temporal Cubic Spline interpolation. The system uses a linear MPC-based trajectory tracker [27] and a low-level controller that uses geometric state feedback in SE(3) [33] for precise reference tracking and execution of fast maneuvers. The trajectories generated are verified to be dynamically feasible both in simulation and real-world experiments.

The reported results consist of simulation deployments in Section V-A, real-world deployments in persistent monitoring presented in Section V-B, and an assessment of the dynamic feasibility of the planned trajectories in Section V-C.

A. Simulation Deployments

Simulation results are presented for the 2D and 3D variants of the KOP for comparison of the proposed method with existing solutions to the KOP. The proposed approach can tackle these variants by the removal of the target dynamics ($\alpha_g = 0$) from (3), such that there is no change in the target reward gains unless collected by the monitoring UAV. Additional terminal constraints on the UAV position and velocity are introduced to the problem, adapting the MPC formulation for a static single-run route planning task. While the proposed method is employed for moving-horizon monitoring, the static KOP is suitable for comparison of performance metrics and the papers' main contribution (the variable timestep) can be better visualized in a static planning scenario. The performance of the proposed VT-MPC is, therefore, evaluated against the existing static time-step MPC approach [9] and other available approaches to the mass-point KOP, which differ in kinematic constraint formulation.

The solution quality reflected as collected reward gains during UAV flight is compared with the state-of-the-art solutions represented by the KOP-1, KOP-6$^{\text{lns}}$ [7], and MPC$_{\text{static}}$ [9]. It is noteworthy that the original mass-point KOP solutions [7] utilize identical kinematic constraints on per-axis velocity and acceleration, obtained by scaling magnitude constraints by a factor of $\sqrt{2}^{-1}$, due to inability to model vector magnitude constraints, and use accelerations as model inputs. Kinematic constraints used by the MPC$_{\text{static}}$ and the VT-MPC formulations are $a_{\max} = 1.5\,\mathrm{m/s^2}$ on acceleration and $v_{\max} = 3.0\,\mathrm{m/s}$ on velocity vector magnitudes. Additional input jerk per-axis constraints used for the VT-MPC are $j_{\max} = 30\,\mathrm{m/s^3}$, sensor function parameters are set to $c_b = 0.05$ and $n_b = 2$.

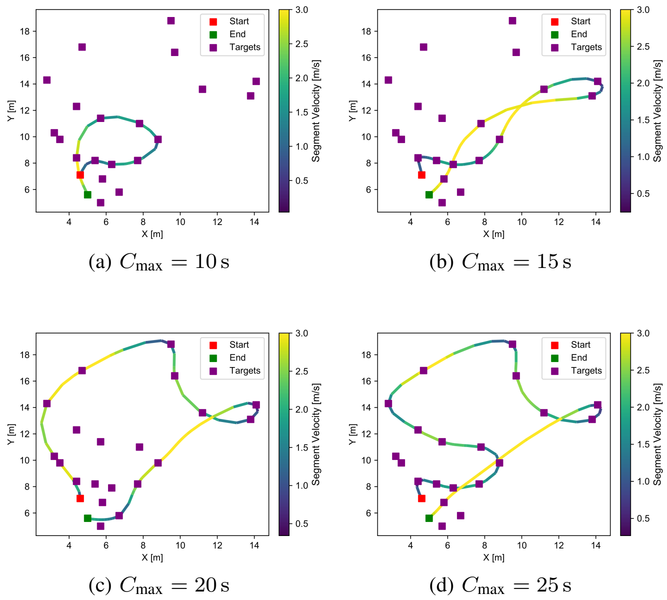
Table I
Collected reward gains for the 2D instances of the KOP.

$C_{\max}$	N_{h}	KOP-1	KOP-6$^{\text{LNS}}$	MPC$_{\text{static}}$	**VT-MPC**	t$_{\text{avg}}$
10	50	95	98	95	**115**	15.5
15	75	180	180	200	**215**	27.1
20	67	250	250	230	**345**	38.2
25	125	325	325	310	**395**	100.6
30	150	390	390	380	**430**	120.8
35	175	430	435	**450**	**450**	167.8
40	200	**450**	**450**	**450**	**450**	238.6

Results for the KOP instances of the 2D Tsiligirides 2 dataset [1] with the maximum available travel budget $C_{\max}[s]$ are depicted in Table I. For each budget, the instance was optimized 10 times to full solver convergence with reward values perturbed as in [9] with the average computational time reported as t$_{\text{avg}}$.

Selected velocity profiles of the dynamically feasible trajectories produced by the proposed planner are depicted in Fig. 6. The distribution of sampled points and computed time-steps are shown in Fig. 7. The same number of the prediction steps as in [9] was used for a fair comparison.

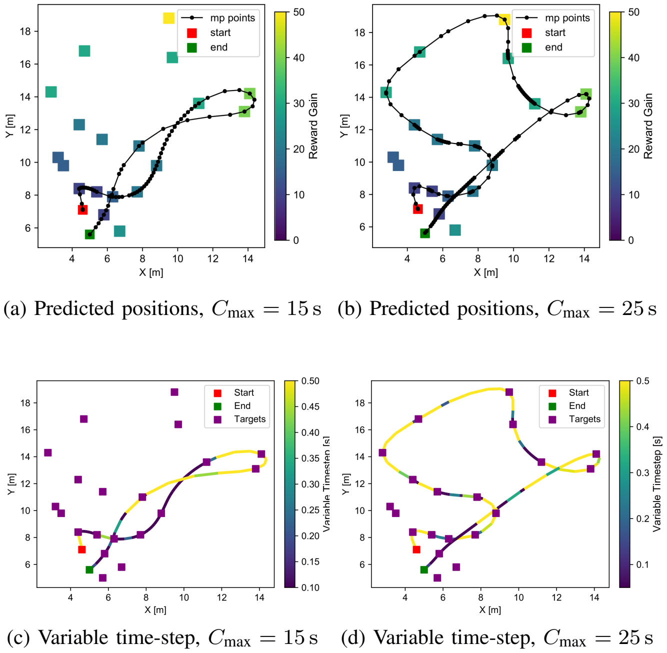
Finally, the evaluation in simulation is performed for the 3D generalization of the KOP that addresses a broader range of real-world scenarios where a UAV needs to visit locations at varying heights. Therefore, the Tsiligirides 2 instance dataset is extended to 3D for validation of the planner performance. The height assignments in the z-axis of each target are such that



(a) $C_{\max} = 10\,\mathrm{s}$ (b) $C_{\max} = 15\,\mathrm{s}$

(c) $C_{\max} = 20\,\mathrm{s}$ (d) $C_{\max} = 25\,\mathrm{s}$

Figure 6. Velocity profiles of the planned trajectories by the proposed VT-MPC planner on the instances from the Tsiligirides 2 dataset.



(a) Predicted positions, $C_{\max} = 15\,\mathrm{s}$ (b) Predicted positions, $C_{\max} = 25\,\mathrm{s}$

(c) Variable time-step, $C_{\max} = 15\,\mathrm{s}$ (d) Variable time-step, $C_{\max} = 25\,\mathrm{s}$

Figure 7. The distribution of the predicted positions (upper row) and variable time-step (bottom row) on 2D trajectories of Tsiligirides 2 instances.

there is a volumetric sparsity in the reward distribution. The bounds on the time-step are $t_{\min} = 0.1\,\mathrm{s}$ and $t_{\max} = 0.9\,\mathrm{s}$. The value of the nominal time-step $T_n = 0.3\,\mathrm{s}$ is used to compute the length of the MPC prediction horizon $N_h = C_{\max}/T_n$. An example of the planned smooth, feasible 3D trajectories along with the distribution of the sampled points is shown in Fig. 8.

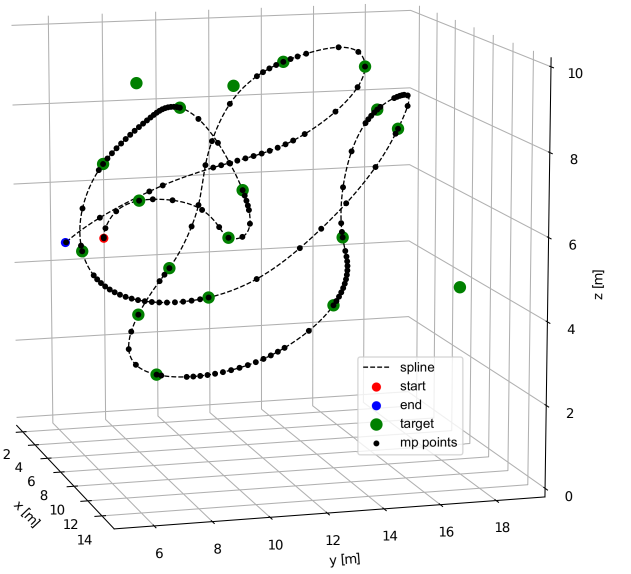


Figure 8. Distribution of predicted positions of the planned trajectory of the 3D Tsiligirides 2 instance with the travel budget $C_{\max} = 40\,\mathrm{s}$.

B. Real-world Deployments in 3D Persistent Monitoring

A series of hardware experiments were performed to emulate a continual monitoring task with targets at known locations in two setups. The first scenario is for static targets with dynamic reward gains. In the second scenario, both targets and reward gains are dynamic.

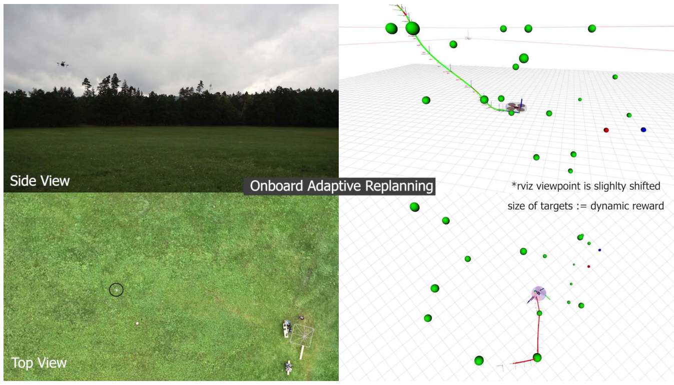
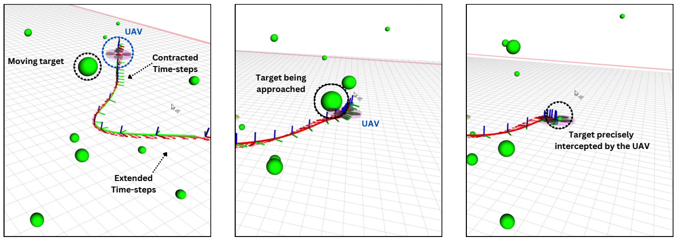

Side View
Onboard Adaptive Replanning
*rviz viewpoint is slighlty shifted
size of targets := dynamic reward
Top View

Figure 9. Experimental field deployment with static target position and dynamic target reward gains in a 3D persistent monitoring task with on-board re-planning.

For the **static target with dynamic reward gains**, the value of the target reward gain increase factor is $\alpha_g = 1$. Synchronized views of the UAV during the experiment with the visualization of the targets in RViz [34] are shown in Fig. 9. The dynamic reward gains model is simulated onboard the UAV, which can be replaced by a perception module that would provide information about the latest target locations and particular reward gains. The online re-planning is performed entirely onboard the UAV. Once collected, the target reward gain drops to 0 and increases as per (3) such that targets with higher gains are prioritized and continual monitoring is achieved.

The **dynamic target and reward gains** setups are designed to demonstrate the real vehicles' ability of precise moving



(a) The planned trajectory is not seen to intercept any moving targets at their current position.

(b) The trajectory is planned with the predicted target position in consideration.

(c) The moving target is intercepted precisely, indicated by the reduced target sphere size.

Figure 10. Visualization of dynamic moving target interception by the vehicle in the 3D persistent monitoring task with on-board re-planning.

target interception as shown in Fig. 10. The approaches are validated with parameters $N_h = 12$, $T_h = 4.0\,\mathrm{s}$, and $t_{\text{replan}} = 0.9\,\mathrm{s}$ for 3D, and $t_{\text{replan}} = 0.5\,\mathrm{s}$ for 2D. The performance of the real-world flights is captured in the accompanying video available at [35].

In 3D interception, simulated targets move around their initial positions following a time-parameterized trefoil knot motion as

$$\begin{aligned} x_r(t) &= \sin(t + t_r) + 2\sin(2(t + t_r), \\ y_r(t) &= \cos(t + t_r) - 2\cos(2(t + t_r)), \\ z_r(t) &= -3\sin(t + t_r). \end{aligned} \tag{11}$$

The positions $x_r(t), y_r(t), z_r(t)$ are displacements of the rth target along the three axes from their initial positions. A phase difference t_r is introduced to each target to ensure that the relative motion around their initial positions is not synchronized. The value of t_r for each of N_t targets is calculated according to $t_r = \frac{2\pi r}{n_t}$.

In 2D interception in the X-Y plane, three targets are represented by UAVs following plane-limited treefoil knot trajectories at a height of $2\,\mathrm{m}$ above the ground while the monitoring UAV is operated at $10\,\mathrm{m}$. Fig. 11 shows a snapshot of the experimental setup. Monitoring UAV velocity and acceleration constraints of 4, 6 and $8\,\mathrm{m/s}$, and 2, 3 and $4\,\mathrm{m/s^2}$ respectively were used to demonstrate the principle of the VT-MPC formulation in real-world experiments.

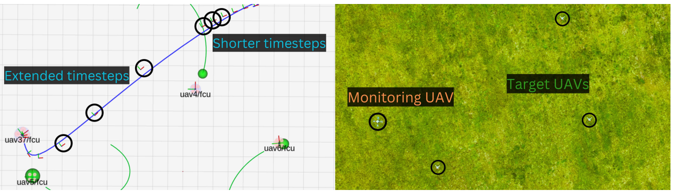


Figure 11. Experimental field deployment for 2D interception with real dynamic targets (target UAVs). The figure shows shorter time-steps (corresponding to precise maneuvers) planned in regions where target interception is predicted.

Although a motion model for the targets is assumed to be known, the continual re-planning ability of the proposed approach allows the motion model to be estimated and updated online via an independent perception and target state estimation, which can run in parallel onboard the UAV.

C. Dynamic Feasibility of Trajectories

The feasibility of the planned trajectories produced by the proposed planner is verified to be dynamically feasible in multiple simulation instances and real-world experiments. The trajectories are tracked by the MPC Tracker and SE(3) controller of the MRS UAV System [28]. A sample tracking error profile is shown in Fig. 12 with the RMSE tracking errors of $0.0420\,\mathrm{m}$, $0.0557\,\mathrm{m}$, and $0.0149\,\mathrm{m}$ in x, y, and z axes, respectively. The performance of the error is similar in all deployments, and the error values are considered sufficiently small, supporting the feasibility of the planned trajectories.

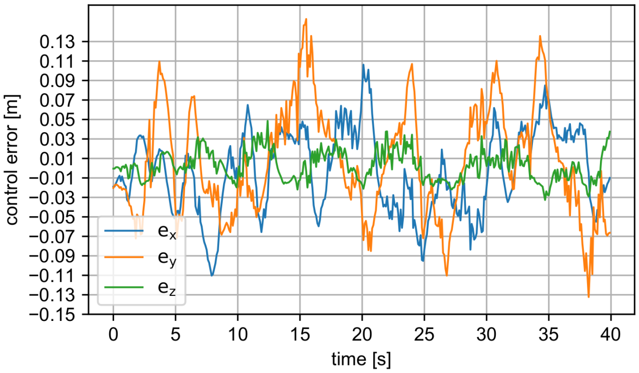


Figure 12. Control errors corresponding to the 3D Trajectory in Fig. 1.

VI. Conclusion

In the paper, we present a novel MCP formulation for precise dynamic target interception utilizing variable time steps. The reported results show that the proposed planner outperforms the state-of-the-art methods for the 2D Kinematic Orienteering Problem. Furthermore, its extension to 3D instances enables adaptation to various tasks. The proposed approach with the variable time steps exhibits two distinct behaviors.

1) Short lengths of the time-step in areas with a high reward (information) gain for precise interception and maneuvering.
2) Long lengths of the time-step in areas without rewards.

A variable length of the time step enables a longer temporal length of the prediction horizon for the same number of steps than the fixed-length steps. Thus, it supports increasing the solution quality without exponentially increasing computational requirements caused by the increase in the number of prediction steps. Based on the empirical evaluation using instances with static targets but with dynamic reward gains, the quality of solutions provided by the IPOPT solver has also increased with the variable time-step approach. The results support that the proposed approach is viable for planning persistent monitoring and surveillance missions. The reported results are backed by multiple real-world deployments that further support the practical feasibility of the proposed approach. The possible future work includes generalization to multi-vehicle problems.

References

[1] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operational Research Society*, vol. 35, pp. 797–809, 09 1984.

[2] D. Pisinger and P. Toth, "Knapsack problems," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds., 1998, pp. 299–428.

[3] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[4] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 464–474, 1996.

[5] R. Pěnička, J. Faigl, and M. Saska, "Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants," *European Journal of Operational Research*, vol. 276, no. 3, pp. 816–825, 2019.

[6] R. Pěnička, J. Faigl, P. Váňa, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, 2017.

[7] F. Meyer and K. Glock, "Kinematic orienteering problem with time-optimal trajectories for multirotor uavs," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 402–11 409, 2022.

[8] F. Meyer, K. Glock, and D. Sayah, "Top-uav: Open-source time-optimal trajectory planner for point-masses under acceleration and velocity constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2838–2845.

[9] F. Nekovář, J. Faigl, and M. Saska, "Multi-vehicle dynamic water surface monitoring," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6323–6330, 2023.

[10] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[11] G. Zhou, C. Li, and P. Cheng, "Unmanned aerial vehicle (uav) real-time video registration for forest fire monitoring," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 3, 2005, pp. 1803–1806.

[12] Q. Laporte-Fauret, V. Marieu, B. Castelle, R. Michalet, S. Bujan, and D. Rosebery, "Low-cost uav for high-resolution and large-scale coastal dune change monitoring using photogrammetry," *Journal of Marine Science and Engineering*, vol. 7, no. 3, 2019.

[13] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for uav-based terrain monitoring," *Autonomous Robots*, vol. 44, no. 6, pp. 889–911, 2020.

[14] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297–309, 2015.

[15] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2520–2525.

[16] F. Stoican, I. Prodan, D. Popescu, and L. Ichim, "Constrained trajectory generation for uav systems using a b-spline parametrization," in *Mediterranean Conference on Control and Automation (MED)*, 2017, pp. 613–618.

[17] V. Usenko, L. von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 215–222.

[18] J. Faigl and P. Váňa, "Surveillance planning with Bézier curves," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 750–757, 2018.

[19] J. Faigl, P. Váňa, and R. Pěnička, "Multi-vehicle close enough orienteering problem with Bézier curves for multi-rotor aerial vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3039–3044.

[20] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal time allocation for quadrotor trajectory generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4715–4722.

[21] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3213–3219.

[22] D. G. Macharet and M. F. M. Campos, "A survey on routing problems and robotic systems," *Robotica*, vol. 36, no. 12, pp. 1781–1803, 2018.

[23] J. Faigl and R. Pěnička, "On close enough orienteering problem with dubins vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5646–5652.

[24] A. Romero, R. Pěnička, and D. Scaramuzza, "Time-optimal online replanning for agile quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, p. 7730–7737, 2022.

[25] M. Beul and S. Behnke, "Fast full state trajectory generation for multirotors," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 408–416.

[26] Z. Zhou, G. Wang, J. Sun, J. Wang, and J. Chen, "Efficient and robust time-optimal trajectory planning and control for agile quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7913–7920, 2023.

[27] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6753–6760.

[28] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, 2021.

[29] F. Fiedler, B. Karg, L. Lüken, D. Brandner, M. Heinlein, F. Brabender, and S. Lucia, "do-mpc: Towards fair nonlinear and robust model predictive control," *Control Engineering Practice*, vol. 140, p. 105676, 2023.

[30] D. Hert, T. Baca, P. Petracek, V. Kratky, V. Spurny, M. Petrlik, M. Vrba, D. Zaitlik, P. Stoudek, V. Walter *et al.*, "Mrs modular uav hardware platforms for supporting research in real-world outdoor and indoor environments," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2022, pp. 1264–1273.

[31] A. Wächter and L. T. Biegler, "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.

[32] HSL, "A collection of fortran codes for large-scale scientific computation," 2013, [cited 15 Nov 2022].

[33] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.

[34] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "RViz: a toolkit for real domain data visualization," *Telecommunications Systems*, vol. 60, no. 2, p. 337–345, 2015.

[35] "Variable Time-step MPC for Agile Multi-rotor UAV Interception of Dynamic Targets," 2024. [Online]. Available: https://mrs.fel.cvut.cz/vt-mpc