Abstract—Planning a time-optimal trajectory for aerial robots is critical in many drone applications, such as rescue missions and package delivery, which have been widely researched in recent years. However, it still involves several challenges, particularly when it comes to incorporating special task requirements as well as the aerial robot’s dynamics into the planning. In this work, we study a case where an aerial manipulator shall pick up a parcel from a moving mobile robot in a time-optimal manner. Rather than setting up the approach trajectory manually, which makes it difficult to determine the optimal total travel time to accomplish the desired task within dynamic limits, we propose an optimization framework, which utilizes the framework of discrete mechanics and complementarity constraints. In the proposed approach, the system dynamics is considered via discrete variational Lagrangian mechanics that provides reliable estimation results according to our experiments. The handover opportunities are automatically determined and arranged based on the desired complementarity constraints. Finally, the performance of the proposed framework is verified with numerical simulations and hardware experiments with our self-designed aerial manipulator.

Index Terms—Aerial manipulator, discrete Lagrangian mechanics, discrete mechanics and complementarity constraints (DMCC), dynamic handover, trajectory planning.

I. INTRODUCTION

FLYING unmanned aerial vehicles (UAVs), especially multicopters, have gained increasing attention over recent years. Compared to other aerial robots, multicopters are lightweight and more flexible, and they can hover over the ground and operate in confined spaces. With various sensors onboard, such as LIDAR sensors [1] and cameras [2], multicopters can be used for photographic and supervisory control duties [3]. In addition, one can equip multicopters with supplementary actors or manipulators to create aerial manipulators, which can further enhance and extend the multicopters’ functional capabilities, so that they are capable of transporting parcels in the 3-D space [4] and performing delicate tasks at a high altitude [5].

However, there are still several challenges for applying multicopters. On the one hand, the onboard battery capacity is limited, so that their operation time and the available payload are restricted, and finding an energy-efficient trajectory is demanded for most drone applications. While the energy-efficient trajectory is not always a time-optimal trajectory, overall, the energy consumption can indeed correlate to the total travel time [6]. Therefore, the shortest possible travel time is usually preferred, even more so in the case of drone racing [7], [8] and timely package delivery [9], [10]. On the other hand, adding sensors or mechanical components to the multicopter system increases its mechanical complexity, complicating both control of the multicopter and planning an admissible trajectory, particularly when one wants to drive the system to its dynamic limit without violating the dynamical constraints.

In this work, we conduct an investigation on a heterogeneous robotic system in which a flying quadrotor equipped with an onboard one-degree of freedom (1-DoF) manipulator wants to approach and pick up an object from a moving mobile robot, as illustrated in Fig. 1. For a flexible and efficient object handover, rather than requiring the mobile robot to halt and wait for the aerial manipulator, this work aims to determine an optimal trajectory for the quadrotor and its onboard manipulator while...
the mobile robot can continue on its way to its destination. To obtain an optimal trajectory for the handover procedure, the potential motion of the mobile robot is essential. The future trajectory of the mobile robot may be predicted based on its recent motion. For instance, neural networks can be used to forecast the observed robot’s near future trajectory [11, 12] or a Bayesian and discrete mechanics and optimal control (DMOC) framework to estimate a long-term trajectory [13]. However, the details of the trajectory prediction for the observed mobile robot are not the subject of this work, and its future trajectory is presumed to be estimated well enough for planning the trajectory of the aerial manipulator. Hence, in this work, we assume that the aerial manipulator has full prior knowledge of the mobile robot’s future trajectory. Given the future or predicted motion of the mobile robot, the proposed method in this study is capable to determine the appropriate handover opportunities, which include the state of the aerial manipulator and the handover time steps. In addition, the entire travel time for the handover procedure is also to be determined, and it should be as short as possible to minimize the operation time without neglecting the overall system dynamics and its limits.

To address trajectory planning, we present an optimization framework for planning the trajectory of a quadrotor and its onboard manipulator. It takes into account the dynamics of the whole system in order to ensure that the estimated trajectory adheres to the dynamic restrictions. By solving the optimization problem, a time-optimal trajectory is represented with a sequence of states of the quadrotor and its manipulator, while the optimal handover opportunities are determined as well.

Previous research works have proposed approaches for controlling the grasping process itself as well as for planning time-optimal trajectories. An overview is provided subsequently.

Object Grasping with Onboard Manipulators: Recent work has examined aerial operation with onboard manipulators in great detail. In our prior work [14], a fixed manipulator is attached under the frame of a quadrotor, and an electromagnet has been implemented to grasp a static object on the ground. Although the coupled structure is robust, it lacks the ability to provide any additional motion to handle the grasping object. As a result, the quadrotor cannot engage in aggressive behavior in order to seize its object. In [15], a 1-DoF manipulator was used to mimic an avian’s grasping trajectory and catch a static cylindrical object at rapid speeds. Both aforementioned studies use a manually planned trajectory represented by polynomials [16], disregarding the system’s dynamics and the effect of the associated mechanics. On the other hand, several research works have also employed complicated manipulators to perform operation tasks. In [17], a 7-DoF manipulator is applied to grasp a moving object that provides a larger manipulating space compared to the aforementioned structure setup. The UAV is controlled by a PID-based controller, and the trajectory planning is not discussed in detail.

Time-Optimal Trajectory Planning: Due to the restricted flight time, planning a time-optimal trajectory is becoming a popular option for UAVs since it can save operation time and increase the efficiency. The majority of work employs a polynomial-based trajectory to guide UAVs in a mission, as it is straightforward to implement and is continuously differentiable, while the travel time is manually specified based on prior knowledge or experiment. Therefore, it is not appropriate for finding a time-optimal trajectory. In [18] and [19], a time-optimal trajectory is generated based on geometric constraints but the quadrotor’s dynamics is neglected, meaning that the resulting trajectory might not be realizable by the quadrotor. In [20], kinodynamic constraints [21] are employed to bound the predicted trajectory, where a polynomial curve is used as the basis of the trajectory. However, only the derivative of the trajectory is constrained using the system’s maximum velocities and accelerations. In more recent work [22], the authors suggest an optimization strategy, which introduces complementarity constraints to distinguish whether the intended trajectory has traveled through the desired path waypoints in 3-D space and the quadrotor’s dynamics is also considered in the optimization problem. Furthermore, in [23], a near-time-optimal flight is realized, and the optimal control input is generated to direct the quadrotor passing through path waypoints at high speed.

In contrast, in this work, we propose an optimization-based trajectory planning algorithm for a quadrotor with a 1-DoF manipulator that enables it to approach and hand over a moving target in a time-optimal manner. Compared to the previous state-of-the-art works, where usually a trajectory is planned only when several essential path waypoints are specified explicitly, the proposed trajectory planner needs to handle a dynamic scenario and provides a joint spatial-temporal solution. Particularly, instead of manually designing a contact trajectory and arranging the handover states with the moving target, the proposed optimization framework automatically determines contact opportunities that satisfy both the studied quadrotor dynamics and automatically finding the optimal handover opportunities throughout the procedure under a variety of desired conditions.

To our best knowledge, this is the first implementation that can handle the handover trajectory planning with a dynamic target for aerial manipulators while considering the full system dynamics and automatically finding the optimal handover opportunities. Moreover, the proposed framework also can be successfully utilized in a classical drone racing trajectory planning, and, as shown in the following, our framework outperforms the state-of-the-art framework, requiring less computational time. In addition, the proposed trajectory planning framework is validated not only in simulations but also in real-world experiments with our customized aerial manipulators.

The rest of this article is organized as follows: In Section II, we illustrate some preliminaries about the modeling of the studied aerial manipulator. Then, in Section III, the proposed discrete mechanics and complementarity constraints trajectory
planning framework is introduced, which comprises the discrete mechanics and optimal control framework associated with the aerial manipulator and the complementarity constraints during the handover procedure. Subsequently, several numerical simulations are conducted to verify the performance of our proposed trajectory planning framework in Section IV. Moreover, the proposed trajectory planning approach is further verified with our aerial manipulator, and the proposed control framework and the real-world experiments are presented in Section V and Section VI, respectively. Finally, Section VII concludes this article.

II. PRELIMINARIES

A. Quadrotor Modeling

At first, the quadrotor itself, without any additional manipulator, is inspected. The system has 6-DoFs while the number of control inputs is only four, which means that we have an underactuated system. The state of the quadrotor is defined as

$$\mathbf{x}_g = \begin{bmatrix} p^T & \mathbf{q}^T & \mathbf{v}^T & \mathbf{\omega}_B^T \end{bmatrix}^T \in \mathbb{R}^{12}$$

(1)

where the position of the quadrotor is represented by \( p := [x \ y \ z]^T \) and the velocity is denoted as \( \mathbf{v} = \dot{p} := [v_x \ v_y \ v_z]^T \). Moreover, the Euler angles of the quadrotor are denoted as \( \mathbf{q} := [\phi \ \theta \ \psi]^T \), and the angular velocity is expressed as \( \mathbf{\omega}_B \) in the body frame. The input of such a system is represented by the four generated forces from spinning propellers that are denoted as \( \mathbf{u}_g = [f_{\text{motor},1} \ f_{\text{motor},2} \ f_{\text{motor},3} \ f_{\text{motor},4}]^T \), resulting in a collective mass-normalized thrust \( f_{\text{thrust}} = \frac{1}{m_{\text{quadrotor}}} \sum_{i=1}^{4} f_{\text{motor},i} \) along the \( z \)-axis of the body-fixed frame of the quadrotor and a torque \( \mathbf{\tau} \) that is calculated as

$$\mathbf{\tau} = \begin{bmatrix} \sqrt{2} \ell_{\text{frame}} (f_{\text{motor},2} + f_{\text{motor},3} - f_{\text{motor},1} - f_{\text{motor},4}) \\ \sqrt{2} \ell_{\text{frame}} (f_{\text{motor},2} + f_{\text{motor},4} - f_{\text{motor},1} - f_{\text{motor},3}) \\ c_r (f_{\text{motor},3} + f_{\text{motor},4} - f_{\text{motor},1} - f_{\text{motor},2}) \end{bmatrix}$$

(2)

where the diagonal length of the quadrotor frame is denoted as \( \ell_{\text{frame}} \) and the torque coefficient is denoted as \( c_r \). Given the notation above, the general nonlinear quadrotor dynamics is formulated as

$$\dot{\mathbf{x}}_g = \mathbf{f}(\mathbf{x}_g, \mathbf{u}_g) = \begin{bmatrix} \mathbf{v} \\ T(\mathbf{q}) \mathbf{\omega}_B \\ \mathbf{R}_B(\mathbf{\xi}) \begin{bmatrix} 0 & 0 & \mathbf{f}_{\text{thrust}} \end{bmatrix}^T - \begin{bmatrix} 0 & 0 & \mathbf{g} \end{bmatrix}^T \\ \mathbf{J}_{\text{quadrotor}}^{-1} (-\mathbf{\omega}_B \times \mathbf{J}_{\text{quadrotor}} \mathbf{\omega}_B + \mathbf{\tau}) \end{bmatrix}$$

(3)

where the gravitational acceleration is denoted as \( g = 9.8066 \text{ m/s}^2 \), the matrix \( \mathbf{J}_{\text{quadrotor}} \) denotes the quadrotor’s moment of inertia, and the rotation matrix \( \mathbf{R}_B(\mathbf{\xi}) \) denotes the transformation from the body frame to the inertial frame of reference. Furthermore, the mapping matrix between the angular velocities between the body frame and the inertial frame is defined as

$$T(\mathbf{\xi}) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\phi)} & \frac{\cos(\phi)}{\cos(\phi)} \end{bmatrix}.$$
as illustrated in Fig. 2, is represented in the quadrotor frame as \([0 0 - \ell_{\text{offset}}]^\top\). The kinematic and dynamic modeling is described in the following.

**Kinematic Modeling:** Given the state of the quadrotor defined in (1), the position \(p_{\text{arm}}\) of the manipulator’s mass center is denoted as

\[
p_{\text{arm}} = p + R_B(\xi) \left( \ell_{\text{offset},B} + R_m(\alpha) \left[ \begin{array}{c} \ell_{\text{arm}}/2 \\ 0 \\ 0 \end{array} \right] \right)
\]

where the length of the onboard manipulator is denoted as \(\ell_{\text{arm}}\), and we assume that the center of mass is identical of the geometric center of the manipulator. Similarly, the position of the end-effector of the manipulator \(p_{\text{end-effector}}\) can be calculated by

\[
p_{\text{end-effector}} = p + R_B(\xi) \left( \ell_{\text{offset},B} + R_m(\alpha) \left[ \begin{array}{c} \ell_{\text{arm}}/2 \\ 0 \\ 0 \end{array} \right] \right).
\]

Furthermore, the translational velocity of the manipulator center \(v_{\text{arm}}\) can be calculated as

\[
v_{\text{arm}} = v - \frac{\dot{\ell}_{\text{arm}}}{2} R_B(\xi) \begin{bmatrix} \sin(\alpha) \\ 0 \\ \cos(\alpha) \end{bmatrix} + R_B(\xi) \omega_B \times \left( \ell_{\text{offset},B} + \frac{\ell_{\text{arm}}}{2} \right) \begin{bmatrix} \cos(\alpha) \\ 0 \\ -\sin(\alpha) \end{bmatrix}
\]

where the angular velocity of the servo motor is denoted as \(\dot{\alpha}\). In addition, the angular velocity of the manipulator represented in the inertial frame can be calculated as

\[
\dot{\xi}_{\text{arm}} = R_B(\xi) \left( \begin{array}{c} 0 \\ \dot{\alpha} \\ 0 \end{array} \right) + \omega_B.
\]

**Dynamic Modeling:** In Lagrangian mechanics, a mechanical system is described in terms of generalized coordinates, which in this work are specified as \(q := [x y z \phi \theta \psi \alpha]^\top = [p^\top \xi^\top \alpha]^\top\) for the studied aerial manipulator. Then, the Lagrange function for the aerial manipulator is defined as

\[
L(q, \dot{q}) = K(q, \dot{q}) - V(q)
\]

where \(K\) denotes the time derivative accordingly, and the kinetic and potential energies of the aerial manipulator are denoted as \(K\) and \(V\), respectively.

The kinetic energy can be further specified as

\[
K(q, \dot{q}) = \frac{m_{\text{quadrotor}}}{2} \dot{v}^\top v + \frac{1}{2} \omega_B^\top J_{\text{quadrotor}} \omega_B + \frac{m_{\text{arm}}}{2} v_{\text{arm}}^\top v_{\text{arm}} + \frac{1}{2} \xi_{\text{arm,m}}^\top J_{\text{arm,m}} \dot{\xi}_{\text{arm,m}}
\]

where the masses of the quadrotor and its onboard manipulator are denoted as \(m_{\text{quadrotor}}\) and \(m_{\text{arm}}\), respectively, and \(J_{\text{arm,m}}\) represents the inertia of the manipulator respecting to each body-fixed frame as \(O_m\) in Fig. 2. The angular velocity of the manipulator is transformed into the coordinates of its body-fixed rotation axis

\[
\dot{\xi}_{\text{arm,m}} = R_m^\top(\alpha) R_B(\xi) \dot{\xi}_{\text{arm}}.
\]

Then, the potential energy of the aerial manipulator is given by

\[
V(q) = m_{\text{quadrotor}}g z + m_{\text{arm}} g p_{\text{arm}}^\top [0 0 1]^\top.
\]

Furthermore, an arbitrary motion of such a system can be represented by a curve along its configuration manifold \(\mathbb{Q}\) [30], which begins from an initial state \((q(t_0), \dot{q}(t_0))\) to an end state \((q(t_N), \dot{q}(t_N))\) under the influence of control forces throughout the time interval \([0, t_N]\). In continuous mechanics, the integration of the Lagrange function \(L\) in this particular trajectory \(q(t)\) is called the Hamilton action \(\mathcal{S}\)

\[
\mathcal{S} = \int_0^{t_N} L(q(t), \dot{q}(t)) dt.
\]

According to the Hamilton’s principle, among all feasible trajectories connecting the specified initial and final conditions at \(t = 0\) and \(t = t_N\), the true trajectories are those that make \(\mathcal{S}\) stationary. Thus, the motion of the mechanical system between these two time stamps yields

\[
\delta \mathcal{S} = 0
\]

where the variation of the Hamilton action is denoted as \(\delta \mathcal{S}\). Thereby, the Hamilton’s principle is also known as the least action principle, since the action is a local minimum for a true trajectory.

Finally, based on (13), one obtains

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0
\]

which are also called the Euler–Lagrange equations.

When considering non-conservative forces \(f_x\), the least-action principle from (13) can be described as

\[
\delta \int_0^{t_N} L(q(t), \dot{q}(t)) dt + \int_0^{t_N} f_x(q(t), \dot{q}(t)) \delta q dt = 0
\]

which is known as the Lagrange-d’Alembert principle.

### III. TIME-OPTIMAL TRAJECTORY PLANNING

**A. Trajectory Planning Problem**

The method proposed in this work shall automatically establish a time-optimal trajectory for guiding an aerial manipulator toward and grasping a dynamic target. There are several requirements that the planned trajectory should fulfill. The end position of the manipulator \(p_{\text{end-effector}}\) should approach the handover target \(p_{\text{target}}\) at least once during the procedure, giving the end-effector a sufficient time window to grasp the target object and complete the handover task. Remark that this time window should be determined given the potential motion of the dynamic target without any manual arrangement. The total travel time
of the entire procedure $t_N$ should be minimized to improve the cooperation efficiency. Furthermore, the relative velocity and the moving direction between the manipulator and the mobile robot must be limited during the handover in accordance with particular experimental criteria or hardware restrictions. In any case, the planned trajectory should not exceed the dynamic limit of the aerial manipulator.

B. DMOC Framework

In general, the procedure for incorporating the system dynamics into the optimization-based trajectory generation problem begins with deriving ordinary differential equations from the Newton–Euler equations or the Euler–Lagrangian defined in (14), and then discretizing the dynamics using methods, such as the fourth-order explicit Runge–Kutta method (RK4) [22], [31]. On the other hand, the system dynamics can be derived using the discrete variational principle [32], and this formulation also can be utilized as a numerical integrator in the optimization problem. In this approach, the discretization of the Lagrange–d’Alembert principle occurs first and then the equations derived from the variational principle are employed to determine the constraints describing the system dynamics.

Based on the second approach, the DMOC framework, first suggested in [33], can be utilized to plan the optimal trajectories of nonholonomic mobile robots [34], manipulators [24], satellites [33], and quadrotors [35] that are subject to the system dynamics. The DMOC can even be utilized to solve an optimal control problem, with a performance comparable to that of a standard nonlinear model predictive control framework [36], [37]. Thus, in this work, we utilize the DMOC framework to formulate the aerial manipulator’s dynamics constraints as well.

In discrete variational mechanics, the trajectory in the time interval $[0,t_N]$ is first discretized. The discrete configuration is described with a vector $q_k \in \mathbb{Q}$ using generalized coordinates describing the system’s configuration at the time steps $t_k = k\Delta t$, $k \in [0, \ldots, N]$. Moreover, the planned total travel time can be obtained by $t_N := N\Delta t$. Note that the orientation of the quadrotor for planning the time-optimal trajectory is described using Euler angles, as introduced in Section II. Since the expected actions of the aerial manipulator should not be so aggressive as to face issues with singularities (“gimbal lock”), the Euler angle description is used in this article for intuitiveness. However, as the group of unit quaternions and the rotation group using Euler angles can be considered isomorphic [38], the configuration space $\mathbb{Q}$ can be described using quaternions as well.

In the previous work [32], it is shown that the discrete variational mechanics framework can also be used with quaternions.

Given the aforementioned discretization, based on the discrete variational principle [39], the continuous configuration space $\mathbb{Q}$ and its tangent configuration space $\mathbb{T}\mathbb{Q}$ are approximated by the discrete configuration space $\mathbb{Q} \times \mathbb{Q}$, and the discrete form of the Lagrange function $L$ is denoted as $L_d$

$$L_d(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))dt \quad (16)$$

which approximates the action over a time interval between $t_k$ and $t_{k+1}$. Then, based on (12) and (16) the discrete Hamilton action can be calculated as

$$\mathcal{S}_d = \sum_{k=0}^{N} L_d(q_k, q_{k+1}) \quad (17)$$

and its variation should be zero according to the Hamilton’s principle. Furthermore, the integration of the virtual work is approximated by

$$\sum_{k=0}^{N} f_k^T \delta q_k + f_k^T \delta q_{k+1} \approx \int_0^{t_N} f_c(q(t), \dot{q}(t))\delta q dt \quad (18)$$

where the external forces are represented at each discrete time step by the left/right forces, denoted by $f_k^-$ and $f_k^+$, respectively [40]. Particularly, as introduced in [33], both discrete forces can be treated equally around each time step as

$$f_k^- = f_k^+ = \frac{\Delta t}{4} (f_{k+1} + f_k). \quad (19)$$

Thereby, the discrete forced Euler–Lagrangian equations can be derived for each consecutive time interval, yielding

$$\frac{\partial L_d(q_{k-1}, q_k)}{\partial q_k} + \frac{\partial L_d(q_k, q_{k+1})}{\partial q_k} + f_{k+1}^- + f_k^- = 0 \quad (20)$$

where the index $k$ is in the interval $\{1, \ldots, N-1\}$. The boundary conditions for the ending positions can be deduced using the discrete Legendre transforms [24] defined as

$$p_k = \frac{\partial L_d(q_{k-1}, q_k)}{\partial q_{k-1}} + f_k^-$$

$$p_{k-1} = -\frac{\partial L_d(q_{k-1}, q_k)}{\partial q_{k-1}} - f_{k-1}^- \quad (21)$$

where $p_k$ indicates the discrete generalized momentum at the time step $t_k$. Then, (21) is used to approximate the known generalized momentum $p(t_k)$ at the ending time step at $t_0$ and $t_N$ that can be calculated through

$$p(t_k) := \frac{\partial L(q, \dot{q})}{\partial \dot{q}}\bigg|_{q=q(t_k), \dot{q}=\dot{q}(t_k)} \quad (22)$$

Based on (21) and (22), the boundary conditions are defined as

$$\frac{\partial L(q, \dot{q})}{\partial \dot{q}}\bigg|_{q=q(t_0), \dot{q}=\dot{q}(t_0)} + \frac{\partial L_d(q_0, q_1)}{\partial q_0} + f_0 = 0$$

$$-\frac{\partial L(q, \dot{q})}{\partial \dot{q}}\bigg|_{q=q(t_N), \dot{q}=\dot{q}(t_N)} - \frac{\partial L_d(q_{N-1}, q_N)}{\partial q_N} + f_{N-1}^- = 0 \quad (23)$$

given the initial and final conditions of the system $q(t_0), \dot{q}(t_0), q(t_N)$, and $\dot{q}(t_N)$, respectively.

Besides, there are various implementations of $L_d(q_k, q_{k+1})$ that can be chosen. In this work, the Verlet method is selected, which has been applied first in the research of molecular dynamics in [41]. It is straightforward to be implemented in the optimization problems, and provides several attractive numerical properties. Based on the Verlet method, the discrete Lagrange
function is defined as
\[ L_d(q_k, q_{k+1}) = \frac{1}{2} \Delta t L(q_k, \dot{q}_{k,k+1}) + \frac{1}{2} \Delta t L(q_{k+1}, \dot{q}_{k,k+1}) \]
where the average velocity between the two time steps is assumed to be constant and calculated by \( \dot{q}_{k,k+1} = (q_{k+1} - q_k)/\Delta t \).

(24)

Finally, based on the equations above, the optimization problem based on the DMOC framework is denoted as follows:

\[
\min_{\epsilon^k, u^k, q^k} J_d(t_N, u_k, q_k) \\
\text{s.t.} \quad \Delta t = t_N/N \\
q_0 = q(t_0) \quad \text{and} \quad \dot{q}_0 = \dot{q}(t_0) \\
q_N = q(t_N) \quad \text{and} \quad \dot{q}_N = \dot{q}(t_N) \\
q_{\min} \leq q_k \leq q_{\max} \quad \forall k \in \{0, \ldots, N\} \\
\dot{q}_{\min} \leq \dot{q}_{k,k+1} \leq \dot{q}_{\max} \quad \forall k \in \{0, \ldots, N - 1\} \\
u_{\min} \leq u_k \leq u_{\max} \quad \forall k \in \{0, \ldots, N\} \\
\text{discrete Euler–Lagrange equations based on (19) – (24)}
\]

where \( J_d \) denotes the discrete objective function of the optimization problem, and the control input of the system at each time step is represented with \( u_k \). In this work, the control input is composed of four generated forces from motors and the torque of the servo motor that drives the motion of the manipulator. Hence, the external forces are identical to the control input.

C. Complementarity Constraints

Apart from adhering to system dynamics and satisfying the ending conditions, the intended trajectory should drive the aerial manipulator to approach its dynamic target and accomplish the handover procedure. Thus, within the planned trajectory, the end-effector of the aerial manipulator should have a period of time in which it is close enough to the ground mobile robot to hand over the target. However, the timing of this handover is hard to be arranged manually for a dynamic target, and is strongly coupled with the system dynamics. Moreover, the desired behavior of the aerial manipulator during the handover and the noncontact phases may differ significantly. For instance, the relative velocity between the end-effector of the manipulator and the moving target should be decreased during the handover to guarantee a steady contact, while both velocities are completely decoupled during the noncontact phases. Thus, by solving the optimization problem, one should be able to distinguish whether the aerial manipulator is in the handover phase at an arbitrary time step \( t_k \).

To indicate the handover phase as well as the handover opportunities, we introduce a set of progress index variables \( \epsilon_k \), where \( k \in \{0, \ldots, N - 1\} \), as inspired by the work from [22]. In our design, each progress index variable \( \epsilon_k \) highlights the handover state at each discrete time step \( t_k \), and it should be nonzero if and only if our desired contact conditions are fulfilled, which are to be detailed later. Hence, if \( \epsilon_k \) is zero, there is no planned contact between the manipulator’s end-effector and the object at the time step \( t_k \).

To implement this design and distinguish the handover timing mathematically, one may formulate it as

\[ \epsilon_k f_c(z) = 0, \quad \text{with} \quad f_c(z) \geq 0 \]

(26)

marked equivalently with the complementarity operator \( \bot \) as

\[ 0 \leq \epsilon_k \bot f_c(z) \geq 0 \]

(27)

where \( f_c(z) \) is the specified nonnegative condition functions of some state \( z \). Equations (26) and (27) are also known as complementarity constraints, where the relationship between the progress index \( \epsilon_k \) and the condition functions \( f_c(z) \) is specified so that at least one of them must remain zero to hold the equality constraint.

According to this design, the first and most straightforward contact condition is based on the Euclidean distance between the end-effector of the manipulator and the target, which is defined as

\[ 0 \leq \epsilon_k \bot \| p_{\text{end-effector}, k} - p_{\text{target}, k} \| \geq 0 \]

(28)

where \( \| \cdot \| \) denotes the Euclidean norm of the given vector. When the end-effector is in the situation of touching and grasping the target, the progress index can be nonzero; on the other phases, the progress index variable should remain zero. Ideally, each progress index variable \( \epsilon_k \) should be binary so that it can unambiguously identify the handover time steps. For example, when the progress index variable equals to one, it indicates that the distance condition is fulfilled and the manipulator is grasping the target; on the contrary, the progress index variable remains zero so that the distance condition can be disregarded in this constraint (28). However, this would lead to a mixed integer problem, which is notoriously difficult to solve. Thus, in our work the range of each progress index variable \( \epsilon_k \) is relaxed and set to be in the interval [0.0, 1.0].

Nevertheless, solely based on condition (28), the handover opportunities may not be indicated, since the progress index can remain at zero no matter which value of the Euclidean norm has been calculated during the whole optimization procedure. To avoid this situation, an additional set of the progress variables \( \kappa \) is introduced, and at each time step the progress index variable \( \epsilon_k \) must satisfy

\[ \epsilon_k = \kappa_k - \kappa_{k+1}, \quad k \in \{0, \ldots, N - 1\} \]

with \( \kappa_0 = \kappa_{\text{init}} \) and \( \kappa_N = 0 \)

(29)

where the positive value of \( \kappa_{\text{init}} \) is specified by the user. Thus, due to \( \kappa_{\text{init}} \) being positive and the final condition of \( \kappa_N = 0 \), the progress index variable \( \epsilon_k \) must be nonzero at a number of time steps. Since \( \epsilon_k \in [0, 1] \), the number of time steps with \( \epsilon_k \neq 0 \) is at least \( \kappa_{\text{init}} \). Therefore, as \( \kappa_{\text{init}} \) increases, the minimum planned contact duration increases.

Remark that unlike the setup based on the similar mechanism for trajectory planning to traverse multiple predefined path waypoints in [13] and [22], where the initial progress variable \( \kappa_{\text{init}} \) is set to one since each intended path waypoint should be passed through only once, in this work, this initial value can be set up...
with a reasonable positive number. The rationale for that is the handover is supposed not to be accomplished solely within one single time step, since a rash grasping makes a solid handover procedure impossible. On the contrary, the handover duration is able to be adjusted using this initial value $\nu_{\text{init}}$, so that the planned trajectory has a sufficient amount of handover time to accomplish the task.

In Fig. 3, a handover demonstration is shown, where the change of both the progress index variables and the progress variables of the produced handover procedure is illustrated with the corresponding time step. In this scenario, the initial value of $\nu_{\text{init}}$ is set to 2, indicating that there are two progress index variables $\nu_k$ should be 1; meanwhile in the planned trajectory the manipulator should maintain contact for grasping at least two time steps. Based on the designed progress conditions in (29) and the contact condition defined in (28), at time steps $t_1$ and $t_2$, the progress index variables become 1 since the end-effector has gotten close enough to the moving grasping target. Note that the results presented here are intended to demonstrate the relationship between the progress variables and the progress index variables; results for planned trajectories will be discussed in further detail in the next section.

Based on the same mechanism introduced above, one can engage more conditions that must be satisfied during the contact phase. In this work, we introduce two additional constraints based on the complementarity constraints. On the one hand, the relative velocity between the end-effector and the handover target should be minimized, so that the impact between the end-effector and the handover target can be reduced. The formulation of this constraint is defined as

$$0 \leq \nu_k \| v_{\text{end-effector},k} - v_{\text{target},k} \| \geq 0. \tag{30}$$

On the other hand, the heading of the quadrotor is set identical to the moving direction of the moving grasping object during the handover procedure, so that the handover procedure will not be disturbed by the landing gears. The heading constraint during the handover procedure is specified as

$$0 \leq \nu_k \| v_{\text{target},k} \times x_{B,k} \| \geq 0, \quad k \in \{0, \ldots, N - 1\} \tag{31}$$

where $\| \cdot \|$ denotes the absolute value. Furthermore, $(\cdot) \times$ denotes the first two components of the given vector, and the vector $x_{B,k}$ denotes the $x$-axis of the quadrotor that is represented in the inertial frame of reference at the time step $t_k$.

### D. DMCC Formulation

In this work, the optimization problem is formulated based on the introduced DMOC framework in (25), together with the complementarity constraints discussed in Section III-C. However, directly incorporating the complementarity constraints into the optimization problem, as a mathematical program with complementarity constraints (MPCC), such as in [42], may make it more difficult to solve since the classical constraint qualifications are failed to hold due to the complementarity constraints [43]. Typically, the MPCC problem needs to be reformulated, and the complementarity constraints may be relaxed with some numerical strategies. As shown in [13] and [22], one can introduce an additional relaxation variable set into the complementarity constraints to mitigate the effect arising from the complementarity constraints. Thereby, the complementarity constraints defined in (28) is reformulated as

$$0 \leq \nu_k \| P_{\text{end-effector},k} - P_{\text{target},k} \| - \nu_k \| \leq 0 \tag{32}$$

where the relaxation variable $\nu_k$ is defined to be positive within a small value. Except numerical requirements, in real-world applications, a slight offset between the position of the end-effector and the handover target is likely acceptable, given that we employ a permanent magnet to grasp the target and the magnetic force can exert its influence at a specific distance.

For the remaining complementarity constraints defined in (30) and (31), another implementation strategy is adopted. As discussed in [44], the equilibrium conditions can be relaxed and replaced with inequalities, such as

$$\nu_k f_c(z) = 0 \implies \nu_k f_c(z) \leq c_{\text{limit}} \tag{33}$$

where $c_{\text{limit}}$ denotes a small constant parameter, e.g., $10^{-2}$.

Moreover, to plan a smooth trajectory and avoid unnecessary aggressive actions throughout the procedure, the cost function penalizes the difference between the optimized control input and the reference control input at each time step. Based on the introduction above, the proposed optimization framework DMCC is formulated as

$$\min \quad J_d = t_N + c_u \Delta t \sum_{k=0}^{N} \| u_k - u_{\text{ref}} \|$$

s.t.

$$\nu_k \| P_{\text{end-effector},k} - P_{\text{target},k} \| - \nu_k \| \leq 0$$

$$\nu_k \| v_{\text{end-effector},k} - v_{\text{target},k} \| \leq c_{\text{limit}}, \text{velocity}$$

$$\nu_k \| v_{\text{target},k} \times x_{B,k} \| \leq c_{\text{limit}}, \text{heading}$$

$$\nu_k \in [0, \nu_{\text{max}}]$$

progress iteration defined in (29)

contrators from the DMOC framework (25) (34)
IV. NUMERICAL SIMULATIONS

In this section, the performance of the proposed trajectory planning framework DMCC is demonstrated in several scenarios. The proposed framework is implemented in Python with the CasADi library [45] in conjunction with the solver IPOPT [46].

A. Trajectory Planning for Aerial Races

One of the challenging applications for the aerial robots is aerial racing, where the UAV travels with its maximal velocity to pass through several static checkpoints in the shortest time. The authors in [22] showed the state-of-the-art performance of trajectory planning while considering the quadrotor’s dynamics through several static gateways in a 3-D space. Their results can even defeat the top human UAV racers. In [22], the authors used standard Newton–Euler equations to formulate the dynamics constraints and discretized them with the RK4 method. In contrast, we implement dynamic constraints using the proposed DMCC framework based on (34) to plan the time-optimal trajectory for the quadrotor in this work.

In the proposed racing scenario from [22], several path waypoints are predefined, as indicated by the green triangles in Fig. 4(a). These waypoints should be traversed by the intended path in the correct order. Furthermore, there is no special requirement about the pass-through velocity and the heading direction in [22]. Thus, in this trajectory planning task, only the Euler distance between the quadrotor and each path waypoint is considered in the proposed DMCC framework, while neglecting the velocity and heading constraints defined in (30) and (31). To make the following comparison fair, despite the individual optimization problem formulations with different numerical integrators, the remaining conditions and parameters are kept the same, as in [22], such as the initial state of the quadrotor and the physical parameters of the quadrotor. Moreover, for fairness, in both cases the numerical solver IPOPT used via CasADi is used for optimization.

As illustrated in Fig. 4, the resulting trajectory from our proposed DMCC framework is very similar to that generated with the method from [22]. Meanwhile, the proposed DMCC framework exhibits a superior numerical performance, see Fig. 5. With more path waypoints, it is straightforward to deduce that the corresponding optimal travel time will grow as the path lengthsen. Both frameworks provide very close predicted optimal travel time. However, the consumption of the computational time shows significant difference. Obviously, more path waypoints necessitate more variables in the optimization problem, which makes it more difficult for the solver to find an appropriate solution. As the number of path waypoints rises, the proposed framework demands less computational time to obtain a comparable performance compared to the work from [22]. The reason of this computational efficiency is due to the discrete mechanics formulation, where the discretization is accomplished using (20) and (23) before solving the optimization problem. The numerical performance of the DMOC-based optimization frameworks is further discussed in [24] and [37], which is beyond the focus of this article.

B. Handover Trajectory Planning

In this section, several dynamic handover scenarios are illustrated to validate the proposed DMCC framework. In particular,
the grasping target is not always in a fixed location. Compared to the aerial races in the previous section, the dynamic system is more complex due to an additional manipulator that is mounted under the frame of the quadrotor. We assume that the end-effector has a magnetic component and that the grasping object is magnetically attractable. The maximum acceptable offset between the end-effector and the grasping object $v_{\max}$ is set to be 0.02 m, assuming that the magnet is strong enough. The other simulation parameters are presented in Table I.

1) Static Object Grasping: In the first scenario, the aerial manipulator starts at the initial position $\left[0.0 0.0 0.65\right]^T$ m, and shall reach the final position $\left[2.5 0.0 0.65\right]^T$ m. During the handover operation, the handover target remains at the fixed position $\left[1.0 0.0 0.4\right]^T$ m. In a previous work [15], a similar scene is set up, but they first studied the motion of a red kite from video footage, such as Fig. 6, and then imitate the red kite’s grasping strategy by generating a polynomial trajectory. On the contrary, in this study, the proposed framework solely involves the system dynamics and the position of the target to automatically generate the dynamics-involved trajectories illustrated in Fig. 7, where the frame projection of the quadrotor on the $x$-$z$-plane is illustrated with blue lines and the manipulator is simplified with green lines, whereas its end-effector is indicated by pink circles.

Compared to the the recorded video footage, it is noteworthy to remark that the trajectories resulting from this work’s approach share some similarities compared with the real behavior of the red kite. Based on the planned trajectories in Fig. 7, the manipulator performs as the claw of the eagle that swings forward to come near to the object prior to grasping it (see Fig. 6 ），then swings back to increase the contact period and reduce the relative velocity with the target when grasping (see Fig. 6 ）。Furthermore, given the positive initial progress variable $\kappa_{\text{init}}$, the end-effector has a period of contact time that it maintains contact with the object in Fig. 7，exhibiting similar behavior compared to the still images 2）and 3）from the red-kite video in Fig. 6.

Moreover, the proposed trajectory planning approach can generate significantly different grasping trajectories depending on the given physical constraints, as illustrated in Fig. 7(a) and 7(b). In Fig. 7(a), the minimum and maximum allowed velocities for the aerial manipulator are specified as $\left[\pm 1.3 \pm 1.3 \pm 0.8\right]^T$ m/s, respectively. Under these conditions, the generated trajectory is smooth, and the attitude of the aerial manipulator does not change significantly during the whole cooperation. In contrast, by increasing the admissible absolute velocity of each axis to $\left[2.6 2.6 0.8\right]^T$ m/s, the proposed framework generates a more aggressive trajectory to accomplish the desired task, as depicted in Fig. 7(b).

In Figs. 8(a) and 9(a), the trace of the end-effector is compared to the position of the grasping object, and the contact period, denoted by a cyan hue, is determined by the estimated results from $\epsilon$. Note that, although the progress index variables $\epsilon_k$ is not

### Table I

| Parameters                          | Value         |
|------------------------------------|---------------|
| UAV mass $m_{\text{quadrotor}}$    | 1.659 [kg]    |
| manipulator mass $m_{\text{arm}}$  | 0.36 [kg]     |
| quadrotor inertia $J_{\text{quadrotor}}$ | diag(0.0348, 0.0459, 0.0977) [kg m$^2$] |
| manipulator inertia $J_{\text{arm}}$ | diag(0,0,0.0019,0,0) [kg m$^2$] |
| manipulator offset $\ell_{\text{offset}}$ | $\left[0 0 -0.05\right]^T$ [m] |
| initial $\kappa_{\text{init}}$     | 2             |
| max. allowed offset between end-effector and the grasping target $v_{\max}$ | 0.02 [m] |
| max. allowed velocity difference by handover $c_{\text{limit, velocity}}$ | 0.01 [m/s] |
| max. allowed heading difference by handover $c_{\text{limit, heading}}$ | 0.1 [rad] |
| max./min. angular velocity $\left[\pm 8.0 \pm 8.0 \pm 2.0\right]^T$ [rad/s] | |
| max./min. manipulator rotation velocity $\pm \pi/2$ [rad/s] | |
| max./min. generated force from a motor | $1.5f_{\text{hover}} / 0.5f_{\text{hover}}$ [N] |

Fig. 5. Performance comparison between the RK4-based approach [22] and the DMCC framework. The results are measured on a machine equipped with an Intel Core i9-8950HK CPU and 32 GB RAM.

Fig. 6. Still pictures from video footage of a red kite.1

Fig. 7. Trajectories generated by the proposed framework for static object grasping intervention. The frame projection of the quadrotor on the $x$-$z$-plane is illustrated with blue lines and the manipulator is simplified with green lines, whereas its end-effector is indicated by pink circles.

1The Slow Mo Guys 2011 Red Kites in Slow Motion—http://youtu.be/AYOxCMZBlkJ
Fig. 7. Two handover scenarios where different velocity constraints are employed in the planning problem. (a) Handover at a fixed position with lower velocity constraints. (b) Handover at a fixed position with higher velocity constraints.

Fig. 8. End-effector trajectory and velocity, and the estimated progress variables during the handover procedure with a static object with lower admissible absolute velocities. (a) End-effector trajectory in the \(z\)-direction. (b) End-effector velocity in the \(x\)-direction. (c) Changes of the variables \(\varepsilon_k\) and \(\kappa_k\).

Fig. 9. End-effector trajectory and velocity, and the estimated progress variables during the handover procedure with a static object with higher admissible absolute velocities. (a) End-effector trajectory in the \(z\)-direction. (b) End-effector velocity in the \(x\)-direction. (c) Changes of the variables \(\varepsilon_k\) and \(\kappa_k\).

set to be binary, the estimated results are close to the ideal value in most studied scenarios, as shown in Figs. 8(c) and 9(c). Nevertheless, a threshold value of 0.2 is generally taken to distinguish between handover and free flight only for illustrating the contact period in the following experiments. Note that this threshold value is unnecessary for solving the optimization problem since the proposed DMCC framework solves the planning problem.
and estimates the handover opportunities simultaneously using the complementarity constraints.

During the contact period, the end-effector stays within the permissible contact range, which is defined by $\nu_{\text{max}}$ in Table I. Furthermore, due to the design of the optimization condition in (30), the end-effector’s velocity during the grasping is nearly zero, minimizing the relative motion between the end-effector and the grasping object, as shown in Figs. 8(b) and 9(b).

2) Moving Object Grasping: The second studied scenario is more challenging since the mobile robot moves with the object to be grasped, and the robot’s motion direction changes during the handover procedure. The mobile robot is moving along a sine trajectory with a constant altitude of $z(t) = 0.4$ m, which begins at $[3.4, 0.55, 0.4]^\top$ and is defined as

$$x(t) = 0.8t + 3.4, \quad y(t) = 0.6 \sin(t + 2.0)$$ (35)

and the aerial manipulator starts at the initial position $[0.0, 0.0, 0.65]^\top$ m, and shall arrive at the final position $[12.0, 0.0, 0.65]^\top$ m.

As given by constraint (31), the planned motion of the aerial manipulator should change its heading to correspond to the motion direction of the mobile robot while satisfying the speed and contact condition in the desired constraints defined in (28) and (30), respectively. In Fig. 10(a), the action of the manipulator is similar to the last previous scenario, which swings forward first and then backward to ensure a sufficient grasping timing. Besides, the aerial manipulator is turning and adjusts its heading to be similar to the motion direction of the mobile robot until the grasping procedure is accomplished, as illustrated in Fig. 10(b).

The details of the proposed trajectory are further illustrated in Fig. 11. The contact altitude of the manipulator is constrained, so that the manipulator can grasp the target during the handover period. The manipulator velocities in both the $x$ and $y$ directions are illustrated in Fig. 11(b) and (c). During the contact period, the velocity of the manipulator is reduced to fit the motion of the mobile robot due to the velocity constraint (30). As expected in Fig. 11(d), the heading is adjusted to the motion direction of the moving target during the contact period.

As demonstrated by numerical simulations, the proposed approach can determine the optimal trajectory for aerial manipulators. During the contact phase, the position of the end-effector corresponds well with that of the grasping target. The same applies for velocity and heading. Without any manual involvement or assistance, the proposed DMCC can provide time-optimal solutions to the trajectory planning problems for nonlinear systems.

V. CONTROL FRAMEWORK

In this section, the estimated trajectory is utilized to direct our self-developed aerial manipulator to grasp an object in a dynamic environment. Besides the proposed time-optimal trajectory generator, an appropriate control framework is required. In this work, we utilize a nonlinear model predictive controller (NMPC) to control the quadrotor to follow the desired optimal trajectory in 3-D space. To reduce the computational complexity of the optimization problem to be solved for the NMPC controller, the onboard manipulator is controlled separately, namely, using a model-based feed-forward controller. The proposed control framework is illustrated in Fig. 12.

In addition, in real-world experiments, we utilize a Pixhawk flight control unit to regulate the angular velocities of the onboard rotors. Thus, the force $f_{\text{motor},i}$ generated from the $i$th motor is not controlled directly by our proposed NMPC algorithm. Instead, the Pixhawk flight control unit can track the desired angular rates using feedback from onboard gyroscopes and an indoor localization system. Moreover, compared to the aerial manipulator’s moment of inertia, the spinning propellers can generate large torques to rotate the whole aerial manipulator, which results in a fast response to desired angular rate commands. Thereby, as illustrated in Fig. 12, the quadrotor’s rotational dynamics is neglected in the NMPC control design, i.e., the NMPC controller assumes that it can directly govern the quadrotor’s angular velocities. Therefore, the control input introduced in Section II-B is converted from $\omega_B$ to $\omega = [\omega_{B,1} f_{\text{thrust}}]^\top$, where the desired angular rate of the quadrotor is denoted as $\omega_{B,1} := [\omega_x \omega_y \omega_z]^\top$. Note that this conversion will not endanger
Moreover, to avoid the singularities in (3), a normalized quaternion \( \hat{q} := [q_w, q_x, q_y, q_z]^T \) is employed to describe the orientation of the quadrotor rather than the Euler angles. Thus, the state of the quadrotor is redefined as \( \mathbf{x} := [\mathbf{p}^\top, \hat{q}^\top, \mathbf{v}^\top]^\top \).

Moreover, due to unknown aerodynamics, unbalanced mass distributions, and other disturbances that are challenging to model directly, the actual quadrotor dynamics in real-world experiments is not identical to the nominal model described in (38). For model-based controllers, the model discrepancy between the nominal model and the real one can have a significant negative impact on control performance. Thus, to improve the model-based controller’s performance, similar to our previous work [14], the system dynamics is augmented with an additional data-driven disturbance observer that is based on a trained Gaussian process regression (GPR) model. As shown in [14], with the augmented system dynamics, the trajectory tracking performance is improved significantly.

In this work, the augmented system dynamics is defined as

\[
\dot{\mathbf{x}} = f_{\text{nominal}}(\mathbf{x}, \mathbf{u}) + f_{\text{ext}}(\mathbf{x})
\]

where the matrix \( R_B(\hat{q}) \) denotes the rotation transformation given the quaternion [48].

Finally, the dynamics of the quadrotor for the NMPC controller is then described as

\[
\dot{\mathbf{x}} = f_{\text{nominal}}(\mathbf{x}, \mathbf{u}) + \frac{1}{2} Q(\omega_{B,d}) \dot{\hat{q}}
\]

where \( Q(\omega_{B,d}) \) denotes a skew-symmetric matrix from \( \omega_{B,d} \) [47], which is defined as

\[
Q(\omega_{B,d}) = \begin{bmatrix}
0 & -\omega_x & -\omega_y & -\omega_z \\
\omega_x & 0 & -\omega_z & \omega_y \\
\omega_y & \omega_z & 0 & -\omega_x \\
\omega_z & -\omega_y & \omega_x & 0
\end{bmatrix}.
\]
Fig. 12. Control framework.

estimated by

$$f_{\text{ext}}(x) = B_z R_B(q(x)) \mu(z) = B_z R_B(q) \begin{bmatrix} \mu_x(z, \alpha, v_{B,x}) \\ \mu_y(z, \alpha, v_{B,y}) \\ \mu_z(z, \alpha, v_{B,z}) \end{bmatrix}$$

where $\mu$ denotes the posterior mean prediction of the trained Gaussian process model given the observed vector $z$, and the matrix $B_z$ defines the mapping between the subspace of the observed vector and the full state, i.e.,

$$B_z = \begin{bmatrix} 0 & 0 \in \mathbb{R}^{3 \times 3} \\ 0 & 0 \in \mathbb{R}^{4 \times 3} \\ I & I \in \mathbb{R}^{3 \times 3} \end{bmatrix}$$

where the identity matrix is denoted as $I$.

The proposed NMPC formulation given the time-optimal trajectory $x_{\text{ref}}$ is

$$\begin{align*}
\min_{x(\cdot), u(\cdot)} & \sum_{k=0}^{N-1} \left( \|x_k - x_{\text{ref},k}\|^2_Q + \|u_k - u_{\text{ref},k}\|^2_R \right) \\
+ & \|x_N - x_{\text{ref},N}\|^2_P \\
\text{s.t.} & \quad x_0 = x(0) \\
& \quad x_{k+1} = x_k + \Delta t f_{\text{RK4}}(x_k, u_k) \quad \forall k \in \{0, \ldots, N-1\} \\
& \quad x_{\text{min}} \leq x_k \leq x_{\text{max}} \quad \forall k \in \{0, \ldots, N\} \\
& \quad u_{\text{min}} \leq u_k \leq u_{\text{max}} \quad \forall \{0, \ldots, N-1\} 
\end{align*}$$

where $Q$ and $R$ are positive-definite weighting matrices and $P$ is a positive semidefinite weighting matrix, respectively. The function $f_{\text{RK4}}$ denotes the fourth-order explicit Runge–Kutta function of the augmented nonlinear dynamics $f_{\text{augmented}}(x, u)$. By solving the optimization problem for each time step, one can acquire the optimized control sequence $u^*$ and the estimated state sequence $x^*$ for the quadrotor, and then send the first optimized control input $u^*_0$ to the flight controller. In this work, the proposed NMPC problem is implemented with the ACADOS library [49], and the optimal control input can be obtained within 0.01 s. Furthermore, the GPR model is implemented with the GPU-accelerated libraries GPytorch [50] and Pytorch [51], and each predicted result for three axes requires only 4 ms with an NVIDIA Quadro M1000 GPU.

VI. EXPERIMENTS

A. Experimental Setup

As illustrated in Fig. 13, the proposed quadrotor with a 1-DoF manipulator consists of a quadrotor with a motor-to-motor diagonal of 330 mm and a manipulator with a length of 182 mm. The latter is powered by a Dynamixel servo motor AX-12 A. The flight control unit is the Pixhawk Cube Black with the open-source PX4 autopilot framework [52]. The onboard computer is a Raspberry Pi 4B+, which establishes the communication between the Pixhawk flight control unit and the laptop, which runs the NMPC controller and the proposed trajectory planner via the local wireless network. The proposed control framework and the communication between devices are implemented in the ROS Noetic environment. Furthermore, the onboard manipulator is placed beneath the center of the quadrotor frame, where the end-effector is simplified with a permanent magnet and a shock absorber is used to attenuate impacts during handover. In experiments, all states of our quadrotor model (38) are directly measured using a tracking system based on six OptiTrack cameras. Moreover, the configuration of the manipulator is also known since the employed servo motor keeps track of the angle of the manipulator. Hence, no state observer is necessary. Please note that, further, plant-model mismatches are partly
compensated for by augmenting the model using the Gaussian process regression, see Section V.

Furthermore, our omnidirectional mobile robot, holonomic extensible robotic agent (HERA) [53], is holding a metal ball with a diameter of 4.8 cm that is constructed of iron, see Fig. 14. The mobile robot is driven by a separate controller and follows the desired trajectory that, in this study, is assumed to be known a priori.

B. Experimental Results

For the handover hardware experiment, two distinct scenarios are prepared to verify the performance of the GPR-augmented NMPC framework given the planned time-optimal trajectory. Note that some physical and dynamical properties of the actual aerial manipulator vary from those in the numerical simulation in Table I due to the employed hardware and the experiment setup. In particular, the laboratory facilities available for experimentation are limited to an area of about 2 × 2.5 m, which is why safety regulations dictate that the maximum velocity of the quadrotor is reduced markedly compared to the previous simulation scenarios, see Table II, despite the fact that higher velocities would naturally be more interesting. However, it is worth emphasizing that the focus and methodological key contribution of this article is not the tracking controller (of which there are many very capable ones in literature), but rather the trajectory planning method. Qualitatively, the transfer of this article’s findings also to higher velocity settings in experiments should be made easier by the fact that our proposed trajectory planning method explicitly considers the dynamic capabilities of the aerial manipulator through the constraints in the underlying optimization problem. Future work may prove that this is indeed the case. The employed parameter settings used in this work’s real-world experiments are shown in Table II, and parameters not contained in the latter are set to the values from Table I.

1) Static Object Grasping: In the first scenario, the mobile robot is placed on a fixed position during the mission, and the planned trajectory based on the proposed trajectory planning framework is illustrated in Fig. 15. Based on the proposed trajectory planning algorithm and the provided control framework, our aerial manipulator has successfully grasped the static object, and four time steps surrounding the moment of grasping are demonstrated in Fig. 16. Furthermore, Fig. 17(a) illustrates the poses of the aerial manipulator throughout the grasping process in accordance with the indoor localization system in our laboratory based on OptiTrack, while the opacity of each depicted pose increases as time passes.
Fig. 17. Results for a static object to be grasped. (a) Poses of the aerial manipulator in the $x$-$z$-plane, where red circles indicate the grasping object, blue circles indicate the mass-center positions, respectively. (b) Trace of the aerial manipulator’s end-effector in the $x$-$y$-plane. (c) Distance between the aerial manipulator’s end-effector and the desired handover position, where the instant of handover is denoted by a red line.

To guarantee a steady handover flight, it should be noted that in this experiment, as opposed to the numerical simulation setup in Section IV-B, the initial value of $\kappa_{\text{init}}$ is set to 13, so that the contact period is extended. The position of the end-effector in the $x$-$y$-plane is shown in Fig. 17(b), where the position of the target is marked with red circles, and the region in which the permanent magnet may grasp the object is marked with orange circles, and the trace of the end-effector is shown by a dash-dotted line. Moreover, the distance changes between the end-effector and desired handover position are shown in Fig. 17(c), where the acceptable contact distance is marked with orange. Here, the desired handover position refers to the location of the grasping object when it is placed above the mobile robot, as shown in Fig. 14.

Note that, since the permanent magnet is not controllable, once the end-effector has contact with the target ball, the ball remains attached to the end-effector. However, the disturbances caused by the grasped target are not considered in our proposed controller since this work’s focus is the grasping process. As a direct consequence of this, the tracking performance once the grasping is successfully completed is negatively impacted, and the tracking inaccuracy is subsequently increased. Another consequence is that, when the handover process begins, the end-effector cannot consistently maintain the contact condition as desired, as shown in Fig. 17(c).

2) Circular Trajectories With Different $\kappa_{\text{init}}$. In the second experiment, the proposed trajectory planning framework and the designed control framework shall handle a dynamic scenario in which the mobile robot is traveling along a circular trajectory. The latter is illustrated in Fig. 18 with a line dashed in blue. Furthermore, to show the impact of varying initial values of $\kappa_{\text{init}}$ on the planned trajectory, two different $\kappa_{\text{init}}$ are tested, and the planned time-optimal trajectories for hand-over are shown in Fig. 18, where the white dash-dotted line represents the results from $\kappa_{\text{init}} = 3$ and the red line denotes the results from $\kappa_{\text{init}} = 23$, respectively. The planned results show that for a smaller $\kappa_{\text{init}}$, the generated trajectory has a considerably shorter contact duration. On the contrary, the larger $\kappa_{\text{init}}$ lets the aerial manipulator stay in close company with the object for a longer time frame, and the generated trajectory closely follows the trajectory of the mobile robot during this time frame.

The hardware experiment in our laboratory is depicted in Fig. 19, and the real-flight performance given the planned time-optimal trajectories with the corresponding $\kappa_{\text{init}}$ is shown in Figs. 20 and 21, respectively. In both experiments, the planned time-optimal trajectories can be executed with our aerial manipulator based on the proposed NMPC controller, and the moving target ball is grasped successfully. Yet, compared to performance with a smaller $\kappa_{\text{init}}$, the aerial manipulator has a significantly
Fig. 20. Moving object grasping results ($\kappa_{\text{init}} = 3$). (a) Poses of the aerial manipulator in the $x$-$z$-plane, where red circles indicate the grasping object, blue circles indicate the mass-center positions, respectively. (b) Trace of the aerial manipulator’s end-effector is denoted by the dashdotted line and the trace of the grasping object is denoted by red circles in the $x$-$y$-plane, respectively. (c) Distance between the aerial manipulator’s end-effector and the desired handover position, where the instant of handover is denoted by a red line.

longer handover period given the trajectory planned with $\kappa_{\text{init}} = 23$, as shown in Figs. 20(a) and 21(a), which corresponds to the planned trajectories depicted in Fig. 18. Furthermore, the total operation time is also impacted by different $\kappa_{\text{init}}$, and the calculated optimal entire travel time $t_N$ is slightly increased due to the larger $\kappa_{\text{init}}$, as indicated in Figs. 20(c) and 21(c).

VII. CONCLUSION

In this work, we propose an optimization framework based on a discrete mechanics and complementarity constraints (DMCC) framework to design an admissible time-optimal trajectory for an aerial manipulator that accomplishes a handover task, where the timing of the handover is determined automatically, while conforming to the studied aerial manipulator’s dynamics. This framework is initially verified in several numerical simulations, including aerial racing and hand-over scenarios. In real-world experiments, the proposed DMCC framework has proven to generate appropriate trajectories for our customized aerial manipulator to grasp a small object from a mobile robot.

Even though the proposed DMCC framework has shown its superior numerical performance compared to a state-of-the-art approach, a typical computational time to obtain the time-optimal trajectory for the studied scenarios takes over 10 s, which is still too long for recomputing the trajectory in real time. Thus, in future work, a more suitable solver may be sought for reducing the computational time. Besides, the computational time may be further reduced by reimplementing everything efficiently in C++.

Moreover, limited by the size of the hardware experimental site, for safety reasons, the allowed maximum velocity of the aerial manipulator is quite low in the conducted hardware experiments. Given a larger experimental area, future work may test more aggressive handover trajectories not just in simulations but also on hardware.

Furthermore, instead of treating the trajectory prediction of the observed target as an individual task, one may incorporate the target trajectory prediction into the optimization and address the entire issue simultaneously. Moreover, future work may strive to account for the influence of the carried object on the dynamics in the controller after having picked up the object.
REFERENCES

[1] A. C. B. Chiella, B. O. S. Teixeira, and G. A. S. Pereira, “State estimation for aerial vehicles in forest environments,” in Proc. Int. Conf. Unmanned Air. Syst., 2019, pp. 890–898.

[2] R. Ait-Jellal and A. Zell, “Outdoor obstacle avoidance based on hybrid visual stereo slam for an autonomous quadrotor MAV,” in Proc. Eur. Conf. Mobile Robots, Paris, France, 2017, pp. 1–8.

[3] A. V. Borkar and G. Chowdhary, “Multi-agent aerial monitoring of moving convoys using elliptical orbit,” in Proc. IEEE Int. Conf. Robot. Automat., Xi’an, China, 2021, pp. 8999–9005.

[4] G. Loianno and V. Kumar, “Cooperative transportation using small quadrotors using monocular vision and inertial sensing,” IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 680–687, Apr. 2021.

[5] C. Ding, L. Lu, C. Wang, and C. Ding, “Design, sensing, and control of a novel UAV platform for aerial drilling and screwing,” IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 3176–3183, Apr. 2021.

[6] H. Alkomy and J. Shan, “An energy analysis of quadrotors with cable-suspended payloads,” in Proc. Int. Conf. Unmanned Air. Syst., Dubrovnik, Croatia, 2022, pp. 49–56.

[7] R. Spicia, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwaiger, “A real-time game theoretic planner for autonomous two-player drone racing,” IEEE Trans. Robot., vol. 36, no. 5, pp. 1359–1403, Oct. 2020.

[8] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, “We are ready for autonomous drone racing? The UZH-FPV drone racing dataset,” in Proc. Int. Conf. Robot. Automat., 2019, pp. 6713–6719.

[9] W. Yuan and L. Rodrigues, “Onboard generation of optimal flight trajectory for delivery of fragile packages,” in Proc. Int. Conf. Unmanned Air. Syst., 2019, pp. 1–8.

[10] M. Kloetzer, A. Burlacu, G. Enescu, S. Caraman, and C. Mahulea, “Optimal indoor goods delivery using drones,” in Proc. IEEE 24th Int. Conf. Emerg. Technol. Factory Automat., Zaragoza, Spain, 2019, pp. 5759–5782.

[11] W. Luo and P. Eberhard, “An LSTM-based approach to precise landing of a UAV on a moving platform,” Int. J. Mech. Syst. Dyn., vol. 2, no. 1, pp. 99–107, 2022.

[12] F. Giuliani, I. Hasan, M. Cristani, and F. Galasso, “Transformer networks for trajectory forecasting,” in Proc. 25th Int. Conf. Pattern Recognit., Milan, Italy, 2021, pp. 10335–10342.

[13] W. Luo and P. Eberhard, “Optimization-based trajectory prediction enhanced with goal evaluation for omnidirectional mobile robots,” in Proc. 18th Int. Conf. Inform. Control, Automat. Robot., 2021, pp. 263–273.

[14] W. Luo, H. Eschmann, and P. Eberhard, “Gaussian process regression-augmented nonlinear model predictive control for quadrotor object grasping,” in Proc. IEEE Int. Conf. Unmanned Air. Syst., Dubrovnik, Croatia, 2022, pp. 11–19.

[15] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, “Avian-inspired grasping of a payload using quadrotors: A reconfigurable cable-driven parallel robot,” in Proc. IFAC-RSJ Int. Conf. Intell. Robots Syst., Daejeon, Korea, 2016, pp. 1623–1630.

[16] C. Y. Son, H. Seo, T. Kim, and H. Jin Kim, “Model predictive control of a multi-rotor with a suspended load for avoiding obstacles,” in Proc. IEEE Int. Conf. Robot. Automat., 2018, pp. 5233–5238.

[17] Z. Samadlakhoshkho, S. Ghorbani, and F. Janabi-Sharifi, “Coupled dynamic modeling and control of aerial continuum manipulation systems,” Appl. Sci., vol. 11, no. 19, 2021, Art. no. 9108.

[18] R. McLachlan and M. Perlmutter, “Integrators for nonholonomic mechanical systems,” J. Nonlinear Sci., vol. 16, no. 4, pp. 383–328, 2006.

[19] W. Van Loock, G. Pipeleers, and J. Swevers, “Time-optimal quadrotor trajectory generation,” in Proc. IEEE/RSJ Int. Conf. Intell. Mech. Syst., 2019, pp. 890–898.

[20] D. Mellinger and V. Kumar, “Minimum snap trajectory generation for trajectory forecasting,” in Proc. 25th Int. Conf. Pattern Recognit., Milan, Italy, 2021, pp. 10335–10342.

[21] W. Luo, J. E. Marsden, and S. Ober-Blobaum, “Discrete mechanics and optimal control,” IFAC Proc. Volumes, vol. 38, no. 1, pp. 538–543, 2005.

[22] M. Kobilarov and G. Sukhatme, “Optimal control using nonholonomic integrators,” in Proc. IEEE Int. Conf. Robot. Automat., Rome, Italy, 2007, pp. 1832–1837.

[23] J. Sun, J. van den Berg, and R. Alterovitz, “Stochastic extended LQR for optimization-based motion planning under uncertainty,” IEEE Trans. Automat. Sci. Eng., vol. 13, no. 2, pp. 437–447, Apr. 2016.

[24] J. M. Wendlandt and J. E. Marsden, “Mechanical integrators derived from a discrete variational principle,” Physica D: Nonlinear Phenomena, vol. 106, no. 3/4, pp. 223–246, Aug. 1997.

[25] O. Junge, J. E. Marsden, and S. Ober-Blobaum, “Discrete mechanics and optimal control,” IFAC Proc. Volumes, vol. 38, no. 1, pp. 141–146, 2018.

[26] J. Solá, J. Deray, and D. Atchuton, “A micro lie theory for state estimation in robotics,” 2018, arXiv:1812.01537.

[27] J. E. Marsden, “Park city lectures on mechanics, dynamics, and symmetry,” in Symplectic Geometry and Topology, vol. 7. Providence, RI, USA: Amer. Math. Soc., 1999, pp. 335–430.

[28] J. E. Marsden and M. West, “Discrete mechanics and variational integrators,” Acta Numerica, vol. 10, pp. 357–514, May 2001.

[29] J. Ismail, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, “Fast trajectory optimization for agile quadrotor maneuverers with a cable-suspended payload,” in Proc. Robot. Sci. Syst. XIII, 2017, pp. 1–10.

[30] C. Kanzow and A. Schwartz, “A new regularization method for mathematical programs with complementarity constraints with strong convergence properties,” SIAM J. Optim., vol. 23, no. 2, pp. 770–798, 2013.

[31] B. Baumrucker, J. Renfro, and L. Biegler, “MPEC problem formulations and solution strategies with chemical engineering applications,” Comput. Chem. Eng., vol. 32, no. 12, pp. 2903–2913, Dec. 2008.

[32] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi—A software framework for nonlinear optimization and optimal control,” Math. Program. Comput., vol. 11, no. 1, pp. 1–56, 2019.

[33] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Math. Program., vol. 106, no. 1, pp. 25–57, 2005.

[34] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, “PAMPC: Perception-aware model predictive control for quadrotors,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Madrid, Spain, 2018, pp. 5200–5207.

[35] S. R. Nekoo, J. A. Acosta, and A. Ollero, “Quaternion-based state-dependent differential Riccati equation for quadrotor drones: Regulation control problem in aerobatic flight,” Robotica, vol. 40, no. 9, pp. 3120–3135, 2022.
Wei Luo (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering, and the doctoral degree (Dr.-Ing.) on the topic “Target Trajectory Prediction-based Object Handover in a 3D Heterogeneous Robot System”, from the University of Stuttgart, Stuttgart, Germany, in 2013, 2016, and 2023, respectively.

He is a member of the research staff with the University of Stuttgart. His research interests include target trajectory prediction, model predictive control, cooperative trajectory planning and applications to heterogeneous swarm robot system.

Jingshan Chen received the B.Sc. degree in mechanical engineering and automation from the China University of Mining and Technology, Beijing, China, in 2016, and the M.Sc. degree in mechatronics, in 2021, from the University of Stuttgart, Stuttgart, Germany, where she is currently working toward the doctoral degree in the fields of robotics, modeling and control with the Institute of Engineering and Computational Mechanics.

She is a member of the research staff with the University of Stuttgart. Her research interests include dynamics of multibody systems and control theory, especially in its applications to robotics.

Henrik Ebel (Member, IEEE) received the B.Sc. and M.Sc. degrees in simulation technology from the University of Stuttgart, Stuttgart, Germany, in 2014 and 2016, and the doctoral degree from the University of Stuttgart on the topic “Distributed Control and Organization of Communicating Mobile Robots: Design, Simulation, and Experimentation”, in 2021.

He is currently a Postdoctoral Researcher and a member of the research staff with the Institute of Engineering and Computational Mechanics, University of Stuttgart. His research interests include multibody system dynamics and control theory, especially in its applications to mechanical systems and in the field of robotics. His research interests also include the cooperation of multiple robotic agents, as well as optimization-based control schemes.

Peter Eberhard received the Diplom-Ingenieur degree in mechanical engineering, in 1991, doctoral degree, in 1996, and the habilitation degree, in 2000, all from the University of Stuttgart. He is currently a Full Professor of Mechanics/Dynamics and has been the Director of the Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany, since 2002. He was Treasurer and Bureau member of the International Union of Theoretical and Applied Mechanics, and served before in many national and international organizations, e.g., as the Chairman of the International Association for Multibody System Dynamics and DEKOMECH (German Committee for Mechanics).