applied sciences

Article

Zero Reaction Torque Trajectory Tracking of an Aerial Manipulator through Extended Generalized Jacobian

Alberto Pasetto, Yash Vyas and Silvio Cocuzza *

Department of Industrial Engineering, University of Padova, 35131 Padova, Italy
* Correspondence: silvio.cocuzza@unipd.it

Featured Application: Inspection of structures, e.g., offshore/nuclear/eolic plants, bridges, and tall buildings. Placement and retrieval of sensors. Assembly of structures in places not accessible/safe for humans.

Abstract: Aerial manipulators are used in industrial and service robotics tasks such as assembly, inspection, and maintenance. One of the main challenges in aerial manipulation is related to the motion of the UAV base caused by manipulator disturbance torques and forces, which jeopardize the precision of the robot manipulator. In this paper, we propose two novel inverse kinematic control methods used to track a trajectory with an aerial manipulator while also considering resultant UAV base motions. The first method is adapted from the generalized Jacobian formulation used in space robotics and includes the change in system momentum resulting from gravity and UAV control forces in the inverse kinematic control equation. This approach is simulated for a 2 and 3 degree-of-freedom aerial manipulator tracking trajectories with the end-effector. Although the end-effector position error is approximately zero throughout the simulated task, we see significant undesired UAV base motions of several centimeters in magnitude. To ameliorate this by exploiting the kinematic redundancy, we modify the generalized Jacobian by adding an additional task constraint which minimizes the reaction torques from the manipulator, to form the extended generalized Jacobian. While the second approach results in improved precision and reduced base motion by an order of magnitude as compared to the generalized Jacobian, a drawback is the reduction in the available workspace as the solution seeks to minimize the manipulator center of gravity translation. We also demonstrate and compare both approaches in a load picking task. All the algorithms are completed computationally faster than real time in the MATLAB simulations, illustrating their potential for application in real-world experiments.

Keywords: aerial manipulation; UAV; kinematic control; redundancy; generalized Jacobian

check for updates

Citation: Pasetto, A.; Vyas, Y.; Cocuzza, S. Zero Reaction Torque Trajectory Tracking of an Aerial Manipulator through Extended Generalized Jacobian. *Appl. Sci.* **2022**, *12*, 12254. https://doi.org/10.3390/app122312254

Academic Editor: Luigi Fortuna

Received: 28 October 2022
Accepted: 22 November 2022
Published: 30 November 2022

1. Introduction

Autonomous systems such as aerial manipulators have seen an uptake in recent years for conducting safety-critical and resource-intensive applications in industry, such as the inspection, assembly, and maintenance of mechanical structures, and in general for inspection and maintenance tasks. Aerial manipulators offer improved feasibility of executing difficult tasks due to the small size and improved maneuverability in comparison to humans.

The types of aerial robotic manipulation platforms are defined by their base and manipulation design. The most common base is a multicopter unmanned aerial vehicle (UAV) with four or more rotors [1], which can remain hovering while the manipulator completes the required task. The manipulator can have several different designs, and for a large workspace and dexterity multiple joint arms are extensively used.

The types of aerial robotic manipulation platforms are defined by their actuator and manipulation design. Based on the mechanical design of these two aspects, platforms are specialized for various tasks such as grasping, pulling/pushing, sliding, bending, or assembly. The actuator design ranges from single to multiple propeller configurations, with various additional actuators to allow for omni-directional motion. The most common type of platform is a parallel propeller configuration with tilted rotors [1], such as TiltHex [2], AEROX [3], and ODAR [4], equipped with rigid links, grippers, or articulated arms. This propeller configuration has also been extended to include servomotors that can tilt the propeller arms, enabling full omni-directional motion [4,5].

The design and control of aerial manipulators face many challenges compared to other types of mobile robots. Unlike wheeled robots, aerial manipulators do not have a stable base, and therefore the forces and torques generated by the manipulator and interactions with the environment affect the platform as a whole [6]. This impacts the position, attitude, stability, and flight autonomy of the aerial manipulator. Various approaches in controller design ranging from decoupled to coupled have been explored to address this problem. A decoupled approach controls the robot and manipulator separately, and compensates for the interaction forces and related motions through disturbance observers and robust controllers. A coupled approach models the full-body dynamics of both the base and the manipulator. A partially coupled approach is also possible, where the control of both components is independent, but improved through sensing or estimation of interaction wrenches on each other [7].

The coupling of both the kinematics and dynamics of the aerial robot and manipulator results in difficulty in precise positioning and control of the end-effector, which is required for complex tasks such as grasping.

Some existing approaches to address this issue include balancing of the center of gravity (CoG) through attitude control [8,9], or static balancing through mechanical design of the manipulator mount mechanism [10,11]. Nevertheless, the experimental results for these approaches still show significant CoG variations and attitude destabilization during manipulator motion. In general, static balancing through controlling the arm to be aligned with the aerial robot does not guarantee that zero reaction forces and torques are transferred from the manipulator to the robot. The dynamic interaction between the manipulator and UAV in aerial manipulation has also been discussed in [12].

To solve this problem, we require minimization of the manipulator reaction torque disturbances on the base, while it moves to complete its task. This problem has been investigated in the domain of space robotic manipulators through various approaches [13–16]. Of relevance for our application is the generalized Jacobian, which allows for manipulator joint-velocity level control with accurate end-effector trajectory tracking even with large base motions [17]. However, adapting this approach from space to aerial manipulators requires several modifications to the inverse kinematic control algorithm, as aerial manipulators do not operate in an environment free of external forces.

In this research, we utilize the generalized Jacobian and extended Jacobian [18] approaches (i) to track the desired end-effector trajectory even if the UAV moves in the 3D space, and (ii) to minimize the dynamic disturbances transferred to the UAV while the manipulator completes the desired tracking/picking task, in the case of a kinematically redundant arm. The UAV is hovering during the above-mentioned maneuvers. We also consider the change in momentum resulting from external forces on the system such as gravity and UAV controller forces in the inverse kinematic control computation. This is a novel contribution required to adapt the generalized Jacobian formulation from its original assumption of conservation of momentum. The use of an extended generalized Jacobian for trajectory tracking and simultaneous reaction torque minimization in aerial manipulation is another novel contribution. The proposed algorithms are validated in simulation for two types of task: trajectory tracking and load picking.

The UAV base is a multicopter, while the arm is a 2- or 3-degree-of-freedom (DOF) serial link arm. The motions of the base and manipulator are restricted to the 2D plane to simplify the problem and better compare the impact of different Jacobian formulations in terms of end-effector position error, UAV base translations, and reaction torques. We consider three different Jacobian formulations:

1. Generalized Jacobian for an ideal-trolley model (Section 4): the UAV base controller maintains perfect hover conditions vertically and rotationally, with only base horizontal translations permitted. A linear trajectory tracking task is completed by a 2-DOF serial link manipulator utilizing a simplified generalized Jacobian formulation. This provides an initial exploration of the problem and demonstrates some of the limitations of purely using the generalized Jacobian for inverse kinematic control without inclusion of external forces.
2. Generalized Jacobian with external forces (Section 5): the Jacobian includes the impact of all base motions, and the inverse kinematics control equation takes into account the external forces and torques such as gravity and UAV controller forces. Both 2-DOF (nonredundant) and 3-DOF (redundant) serial link manipulator cases are simulated for the trajectory tracking tasks outlined in Section 3.
3. Extended generalized Jacobian (Section 6): the generalized Jacobian is extended to include an additional constraint, which is the minimization of the reaction torques on the UAV caused by the manipulator motion for a redundant 3-DOF serial link manipulator. We validate this algorithm with two trajectory tracking tasks as outlined in Section 3.

Section 2 contains the background for the proposed algorithms, including the dynamic model for planar UAV motions, a general outline of inverse kinematic algorithms for manipulator joint control, and the generalized Jacobian formulation. Section 3 describes the simulation setup, including the trajectory tasks and the kinematic and dynamic parameters used. Sections 4–6 derive the Jacobian formulations and show results for the trajectory tracking tasks. Section 7 considers the scenario where the end-effector is picking a load and compares the generalized Jacobian and extended Jacobian formulations for completing this task. Computational times of the inverse kinematics solution through the generalized and extended generalized Jacobian were measured on a machine featuring an Intel i7 eighth generation exa-core (12 thread) processor with 16 GB RAM, SSD mass storage and using Windows 11 and MATLAB version 2022a. The measured computational times are mentioned in the corresponding Simulation Results subsections.

2. Background Theory

2.1. Unmanned Aerial Manipulator Dynamics and Control Model

In this research, we select a fixed-rotor multicopter configuration as our UAV base, with the manipulator as a serial multiple link arm mounted on the base. We model the unmanned aerial manipulator (UAM) as a dynamically coupled system, i.e., the base and manipulator are regarded as two distinct systems, but the motion of either system affects the other, in addition to the external forces and torques acting on the whole system.

The UAV is modeled as a rigid body subjected to net thrust/attitude torques from the propellers and gravity. The UAV controller tries to keep altitude and inclination angle in hover conditions (i.e., maintaining elevation and minimizing attitude changes).

We model the UAV and manipulator dynamics separately, taking into account the interaction forces, and employ separate control algorithms: (1) PID control to maintain the UAV in hover conditions and (2) inverse kinematic control of the manipulator to perform the desired task.

The reference frames for the UAV are shown in Figure 1, where I denotes the inertial frame and B is the UAV body-fixed frame. We use the conventional simplification for the controller forces and torques for UAVs, where the body of the UAV moves vertically along z_B through a net thrust force and rotates about all three local body axes x_B, y_B, and z_B through attitude torques. In a real-world application, a low-level controller calculates the rotor speeds that result in these control forces and torques. The general dynamic equations relating rotor speeds and simplified thrust/attitude control inputs for a multicopter system in 3D are summarized in [12]. In our analysis, we only consider UAM translations in the 2D plane $x_I z_I$ and rotation about the y_I-axis orthogonal to that plane.

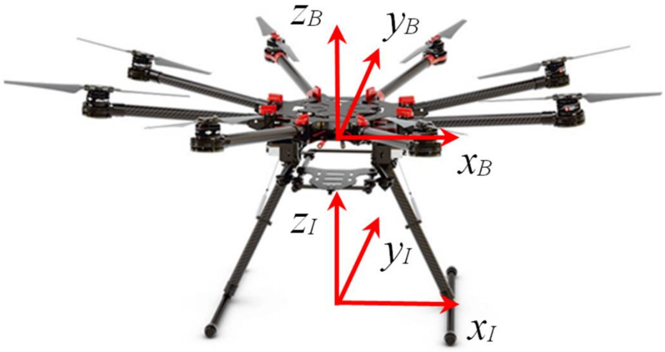

Figure 1. Illustration of reference frames for the UAV system (DJI S1000) [12].

The control actions on the UAV base in the $x_I z_I$ plane are simplified into a vertical thrust force U_1 and roll moment U_2. The mechanical design of a fixed-rotor multicopter does not permit horizontal displacement to be controlled independently, hence horizontal motion is controlled by applying a combination of thrust force and roll moment on the UAV.

We model the UAV center of gravity (CoG) in the inertial frame with the Cartesian position in the 2D $x_I z_I$ plane $[x, z]^T$ and the UAV rotation with the roll angle ϕ about the UAV body y_B-axis (centered at the CoG) parallel to the inertial y_I-axis. The net thrust force U_1 acts directly upwards from the UAV CoG at the roll angle ϕ. U_2 is the net roll moment about the UAV CoG. With m_B as the mass and I_B as the rotational moment of inertia about the CoG, the UAV dynamic equations are the following:

$$\begin{aligned} m_B \ddot{x} &= -U_1 \sin(\phi) \\ m_B \ddot{z} &= -m_B g + U_1 \cos(\phi) \\ I_B \ddot{\phi} &= U_2 \end{aligned} \tag{1}$$

The manipulator is mounted on the base with an offset vector r_m with respect to the UAV CoG, and the manipulator motion exerts reaction forces along the x_I and z_I axes, R_x and R_z, as well as a reaction torque τ_m on the interface point. The UAM model is illustrated in Figure 2. In this paper, we focus on tracking a desired end-effector trajectory while the UAV maintains hover, and on locally minimizing the manipulator reaction torques while the manipulator executes a task.

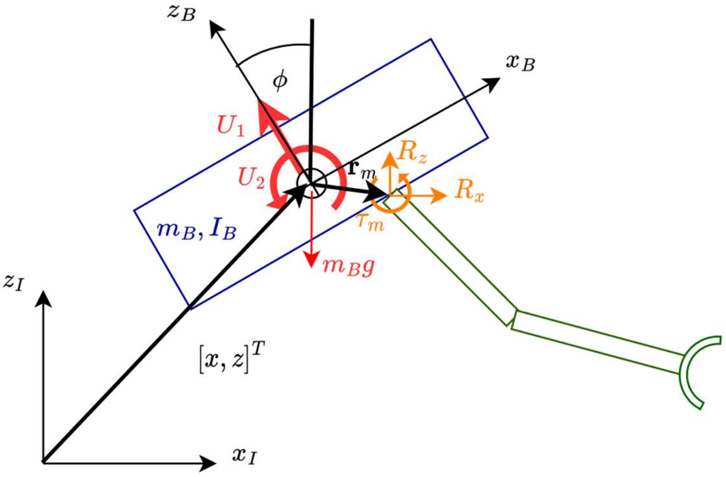

Figure 2. Coordinates and forces of the UAM. UAV forces/torques are in red and manipulator reaction forces/torques are in orange. The UAV is shown in blue and manipulator in green.

Indeed, as demonstrated in [12], a torque transferred to the UAV CoG causes an undesired large horizontal displacement of the UAV, which in turn jeopardizes the end-effector precision. On the other hand, global [19] and nearly-global [20] optimization approaches have been developed in the case of fixed-base and of space manipulators, and their applicability to the case of aerial manipulators will be part of future work.

As in real-world applications, during hover the desired UAV position and attitude are set by a high-level controller and an inner-loop controller calculates the required thrust and attitude torques. The UAV inner-loop controller modeled is a proportional–integral–derivative (PID) controller which minimizes the vertical and roll displacement:

$$\begin{aligned} U_1 &= -k_{Pz} z - k_{Dz} \dot{z} - k_{Iz} \int_0^t z \, dt \\ U_2 &= -k_{P\phi} \phi - k_{D\phi} \dot{\phi} - k_{I\phi} \int_0^t \phi \, dt \end{aligned} \tag{2}$$

It is important to note from (2) that the horizontal displacements cannot be independently controlled. Hence, the minimization of reaction torques from manipulator motions aims to primarily minimize the horizontal displacement. This concept is first explored in an ideal control example in Section 4, where we assume that our controller has instantaneous control response to vertical and roll displacements, and the UAV only translates horizontally. The simple generalized Jacobian used in the inverse kinematic control algorithm only accounts for these horizontal translations in the linear momentum formulation. We then relax the ideal control assumption and account for the full range of base motions as well as the controller/external forces and torques in subsequent sections.

2.2. Kinematics of Manipulators

The configuration of a robotic manipulator system is described in terms of a vector of generalized coordinates $q \in \mathbb{R}^{n_e}$, which are a complete and independent set of scalar coordinates. The degrees of freedom are the minimal set of coordinates needed to describe the system configuration, which in our case correspond to the number of actuated joints in the manipulator n_e, and each coordinate in q is a joint angle.

The forward kinematic equations $\chi_\text{e}(q)$ are a geometric mapping from the manipulator configuration in generalized coordinates q to the desired end-effector pose task p_e in Cartesian space: $p_e = \chi_\text{e}(q)$. In our case, we desire a target end-effector position in the 2D plane, and hence $p_e \in \mathbb{R}^2$, consisting of the x_I and z_I Cartesian coordinates at a given time t.

For kinematic control of the end-effector, we set the corresponding target velocity $v_e \in \mathbb{R}^2$ of the end-effector based on a generated trajectory velocity profile that follows the desired trajectory positions. The relationship between the forward kinematics in position space and the forward differential kinematics in velocity space is described through the Jacobian matrix $J(q) \in \mathbb{R}^{2 \times n_e}$:

$$\dot{p}_e = v_e = J(q)\dot{q} \tag{3}$$

where $J(q) = \frac{\partial \chi_\text{e}(q)}{\partial q}$, and $\dot{q}$ is the vector of joint velocities at time t. For brevity's sake, all Jacobians $J(q)$ will be hereafter denoted as J, with the implicit relationship to the joint state $q(t)$.

We invert this relationship to obtain the inverse differential kinematic relation which defines the required joint velocities to achieve our desired end-effector velocity at time t:

$$\dot{q} = J^{-1} v_e \tag{4}$$

The inverse J^{-1} exists as long as the matrix J is square and invertible, i.e., the manipulator is not in a singular configuration where $Det(J) \approx 0$. These are typically configurations where the links are aligned in a straight line, and we initialize the manipulator configuration to avoid such configurations. In the case of a redundant manipulator ($n_e > 2$), the matrix J is not square and, for a desired Cartesian velocity task, infinite solutions of $\dot{q}$ that achieve the desired v_e are possible. To find the optimal joint velocities, we minimize the objective function $g(\dot{q})$ for desired joint velocities $\dot{q}_0$:

$$g(\dot{q}) = \tfrac{1}{2}(\dot{q} - \dot{q}_0)^T W(\dot{q} - \dot{q}_0) \tag{5}$$

where $W \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite weighting matrix. This objective function is minimized subject to the forward differential kinematic constraint (3) and formulated as a Lagrange multiplier problem:

$$g'(q, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^T W(\dot{q} - \dot{q}_0) - \lambda^T(v_e - J(q)\dot{q}) \tag{6}$$

where $\lambda \in R^{n_e}$ is the vector of Lagrange multipliers. We find the minimum of $g'(\dot{q}, \lambda)$ by solving $\nabla_{(\dot{q},\lambda)} g'(\dot{q}, \lambda) = 0$. This is carried out through setting $\frac{\partial g'(\dot{q},\lambda)}{\partial \dot{q}} = 0$ and $\frac{\partial g'(\dot{q},\lambda)}{\partial \lambda} = 0$, which results in the following solution:

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T)^{-1}v_e + (I - W^{-1}J^T(JW^{-1}J^T)^{-1})\dot{q}_0 \tag{7}$$

If we desire the solution which minimizes joint velocities, it is sufficient to set $\dot{q}_0 = 0$, and if the weighting of all joints is equal (i.e., $W = I$), then the solution to the redundant manipulator kinematic control equation is:

$$\dot{q} = J^T(JJ^T)^{-1}v_e = J^+ v_e \tag{8}$$

with $J^+ = J^T(JJ^T)^{-1}$ as the Moore–Penrose right pseudo-inverse.

Detailed derivation of these inverse kinematics control equations can be found in [21].

2.3. Generalized Jacobian

The inverse kinematics solutions outlined in Section 2.2 are suitable for fixed-base manipulator control, where there is a direct relationship between the joint velocities and end-effector position.

However, in our case our manipulator is mounted on a floating base system (the UAV), and the movement of joints causes reaction forces and torques on the UAV. The base of the manipulator moves due to these forces, which also changes the end-effector and joint and link positions. Hence, for effective tracking an improved method that includes the effect of these forces and torques on the manipulator motion is required.

To achieve this, we employ a method from space manipulator mechanics known as the generalized Jacobian [17]. This Jacobian formulation employs the properties of free-floating manipulators in the absence of gravity, where the linear and angular momentum are conserved. In this work, the momentum balance equations are combined with the kinematic Jacobian formulation derived in Section 2.2 to incorporate the external forces on the whole system, allowing for accurate end-effector trajectory tracking through velocity-level joint control, even if the UAV base moves in 3D space due to the dynamic interaction between UAV and manipulator.

The generalized Jacobian algorithm in [17] is computed with the following steps:

1. The center of gravity (CoG) of the base–manipulator system is found and the inertias of the base and links are reformulated in a CoG centered reference frame.
2. The kinematic equations of the system for velocity-space are defined in the CoG frame. The Jacobians and generalized coordinates for the base component and manipulator component are decoupled:

$$v_e = J_b\dot{q}_b + J_m\dot{q}_m + v_0 \tag{9}$$

where $\dot{q}_b$ and $\dot{q}_m$ are the velocities of the generalized coordinates of the base and manipulator, respectively, J_b and J_m are the corresponding Jacobians, and v_0 is the initial translational velocity of the end-effector.

3. The momentum conservation equations for the system in the CoG frame are defined. Similarly to (9), the base and manipulator components are decoupled:

$$L_0 = \bar{I}_b\,\dot{q}_b + \bar{I}_m\dot{q}_m \tag{10}$$

where L_0 is the initial system momentum, $\bar{I}_b$ and $\bar{I}_m$ are the mass/inertia matrices for the base and manipulator in the CoG frame, the calculation of which is provided in [17]. Note that the momentum is purely rotational and the inertias are time-varying about the instantaneous CoG of the system at time t.

4. Combine (9) and (10) to eliminate the unknown base variables $\dot{q}_b$, to find the relationship between end-effector and joint velocities:

$$v_e = \left(J_m - J_b\bar{I}_b^{-1}\bar{I}_m\right)\dot{q}_m + J_b\bar{I}_b^{-1}L_0 + v_0 = J^*\dot{q}_m + p_0 \tag{11}$$

where we rewrite $p_0 = J_b\bar{I}_b^{-1}L_0 + v_0$ and J^* is the generalized Jacobian. The inverse kinematic relationship to find the desired joint velocities therefore becomes:

$$\dot{q}_m = [J^*]^{-1}(v_e - p_0) \tag{12}$$

The generalized Jacobian for a 2-DOF manipulator fulfilling the 2D task v_e is nonredundant as $v_e \in \mathbb{R}^2$ and $J^* \in \mathbb{R}^{2\times 2}$, so direct inversion is possible. For a manipulator with more than 2 degrees of freedom, $J^* \in \mathbb{R}^{2\times n_e}$ and $n_e > 2$, which results in kinematic redundancy. In this case, we minimize the joint velocities by taking the Moore–Penrose right pseudo-inverse (8). Another alternative for inverse kinematic control of a redundant manipulator is to extend the task-space by adding tasks so that the $v_e \in \mathbb{R}^{n_e}$ and J^* is square. This approach is called the extended Jacobian [18] and we utilize this approach in Section 6 by adding an additional constraint to the generalized Jacobian derived in Section 5, which minimizes the reaction torque on the base due to the manipulator motions.

Our approach for the generalized Jacobian and extended generalized Jacobian derivations in Sections 5 and 6 has some modifications to the original algorithms, as detailed here below. For all the following derivations, steps 1 and 2 are omitted and the kinematic and momentum balance equations are defined about the inertial frame with the generalized coordinates consisting of the base coordinates and joint angles. In Section 5 to Section 6, we relax controller assumptions and develop the complexity of our Jacobian formulation by adding all components of linear and angular momentum as well as modeling all external forces and torques acting on the system.

3. Simulation Setup

The UAM control algorithms derived in subsequent sections are implemented in a MATLAB simulator that recursively solves the kinematics and dynamics of free-base robotic systems [15], which was originally developed for space robotic systems, and that has been adapted to take into account the gravity force and the control forces of the UAV for the purposes of this work. The simulator solves the inverse kinematics aimed at tracking a desired end-effector velocity profile by a time discretization: at each iteration the end-effector desired velocity is given, and the joint velocities and angles are calculated, the timestep used is $\Delta t = 1 \times 10^{-3}$ s.

To validate the algorithms, the joint velocities obtained from the inverse kinematics performed in MATLAB are then imported and imposed in an ADAMS model of the UAM used to perform dynamic simulations. The dynamic simulations are used to validate the proposed solutions and, in particular, to measure the maximum position error of the end-effector, defined as the 2-norm difference between the desired and actual end-effector position. The UAV and manipulator kinematic and dynamic parameters are outlined in Table 1.

Table 1. Kinematic and dynamic UAM model parameters used in simulation experiments.

Parameter	Description	Value	Unit
m_{UAV}	Mass of the UAV	4.2	kg
m_i	Link mass	1	kg
I_{UAV}	Moment of inertia (z) of the UAV	0.4097	kg m^2
$I_{g,i}$	Moment of inertia (z) of Link i	6×10^{-4}	kg m^2
l_i	Link i length	0.13	m
a_i	Distance from joint i to link i CoG	0.065	m
x_o	Distance from UAV CoG to manipulator joint 1 along UAV body x-axis	0	m
z_o	Distance from UAV CoG to manipulator joint 1 along UAV body z-axis	-0.06	m

We tested 2 types of end-effector trajectory tracking tasks:

1. Linear trajectory with x_I displacement of 0.1 m and z_I displacement of 0.06 m from the initial position of the end-effector and back, to be completed within 5 s. This is implemented in the velocity space with a smoothed trapezoidal velocity profile, with an acceleration, constant velocity, and deceleration phase for the forward and backward motions. The equation defining the velocity v along axis x_I or z_I at time t for the forward motion is the following:

$$v = \begin{cases} \frac{1}{2}v_{max}\left(1+\sin\left(\frac{t\pi}{t_a}-\frac{\pi}{2}\right)\right), & t < t_a \\ v_{max}, & t_a \leq t < T - t_a \\ v_{max}\left(1-\frac{1}{2}\left(1+\sin\left(\frac{\pi(t-(T+t_a))}{t_a}-\frac{\pi}{2}\right)\right)\right), & T - t_a \leq t \end{cases} \tag{13}$$

where t_a is the acceleration interval and is equal to 0.4 s; T is the duration of the forward motion; the maximum velocity v_{max} is different for the x_I and z_I components and can be respectively calculated from the maximum displacement in the corresponding direction Δs as $v_{max} = \frac{\Delta s}{T-t_a}$.

The velocity profile of the backward motion has the same shape, i.e., the profile is translated of T and has the opposite sign. The velocity profile and initial configuration for a 3-DOF manipulator with the kinematic and dynamic parameters of Table 1 (as modeled in Sections 5 and 6) are shown in Figures 3 and 4.

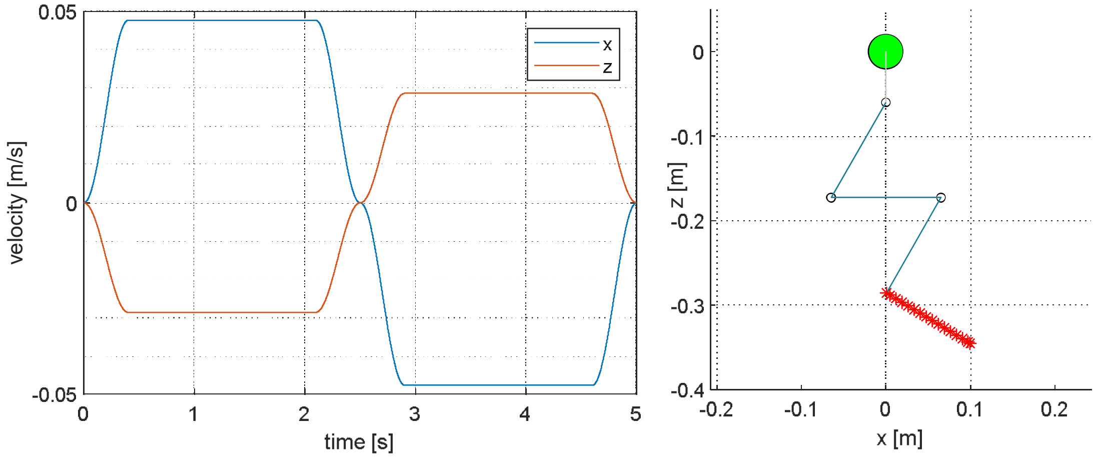

Figure 3. End-effector desired velocities for the line trajectory (**left**) and initial configuration for 3-DOF manipulator (**right**). The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

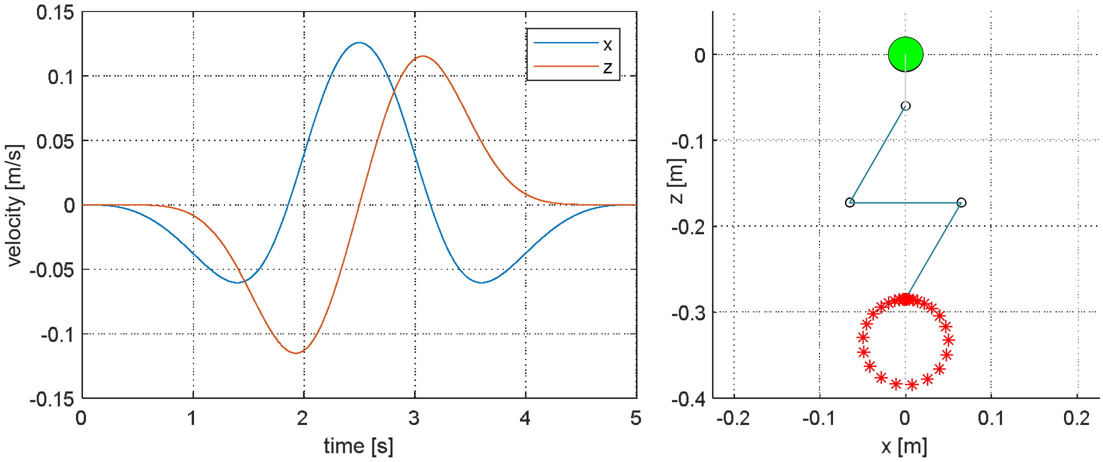

Figure 4. End-effector desired velocities for the circular trajectory (**left**) and initial configuration for a 3-DOF manipulator (**right**). The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

2. Circular trajectory with diameter 0.1 m, to be completed within 5 s, with the initial position of the end-effector as the topmost point of the circle. The velocity profile is generated using the procedure in [22] and is omitted here for brevity. The velocity profile and initial configuration for a 3-DOF manipulator with the kinematic and dynamic parameters outlined in Table 1 (as modeled in Sections 5 and 6) are shown in Figure 4.

In addition to this, for the generalized Jacobian with external forces (Section 5) and extended Jacobian (Section 6) cases we also investigate a load picking test case and compare both approaches, using a separate trajectory profile accounting for the load model. This scenario and its trajectory are elaborated on in Section 7.

For each trajectory, the initial configurations for the joints are set so that the base and manipulator centers of gravity are aligned, the manipulator is in a nonsingular configuration, and the system is at an equilibrium where the control force U_1 and torque U_2 keep the UAM at hover. The starting point of the end-effector is set to the beginning of each trajectory.

The kinematic and dynamic parameters of all the models used in subsequent sections are summarized in Table 1. For the UAV, we used the parameters of the high-payload multicopter DJI S1000 available in our lab (as shown in Figure 1). We modeled heavy links on purpose in order to keep reaction forces and torques resulting from the manipulator motion high, so that their impact is easily observable for comparison purposes between different experiments. In a real-world implementation, the links could be lighter and designed to reduce the impact of the manipulator motion on the system dynamics, resulting in a smaller disturbance and in an improved control of the UAM.

The PID controller tunings in the UAV control forces calculation (2) are summarized in Table 2.

Table 2. PID controller tunings.

Coordinate (*i*)	k_{Pi}	k_{Di}	k_{Ii}
z	37	18	8
ϕ	40	3	35

4. Ideal Control 2-DOF Trolley Model

4.1. Model Derivation

We first formulate a simple model of our UAM system, assuming that our UAV base has perfect vertical and rotational tracking of hover conditions through its controller. This is modeled as a horizontal trolley with a 2-DOF manipulator mounted at the CoG position of the UAV base (G_{UAV}), with mass m_{UAV} and rotational inertia I_{UAV}.

Our generalized coordinates are the joint angles q_1 and q_2. The following parameters are defined: i is the link number, m_i is its mass, l_i is the length, and a_i is the distance from the first joint of the link to the link CoG position G_i. The geometry and dynamic parameters of the system are illustrated in Figure 5.

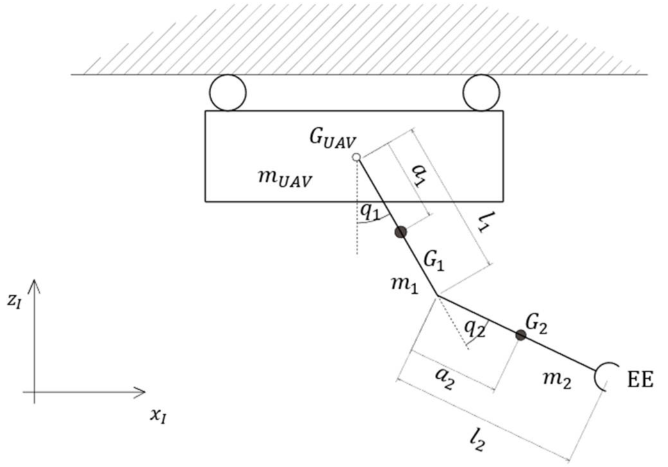

Figure 5. Model for the ideal 2-DOF trolley case.

The end-effector task velocities v_e consisting of x and z velocities v_x and v_z are derived with respect to the joint velocities $\dot{q}$ and kinematic Jacobian J_k:

$$\begin{bmatrix} v_x \\ v_z \end{bmatrix} = \begin{bmatrix} J_{11}(q) & J_{12}(q) \\ J_{12}(q) & J_{22}(q) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) & l_2 \sin(q_1 + q_2) \end{bmatrix} \dot{q} = J_k \dot{q} \tag{14}$$

The joint movements induce motion of the base and manipulator along the horizontal (x_I) axis. We define the linear momentum along this axis:

$$(m_{UAV} + m_1 + m_2)\dot{x}_B + m_1 a_1 \cos(q_1)\dot{q}_1 + m_2\big((l_1 \cos(q_1) + a_2 \cos(q_1 + q_2))\dot{q}_1 + a_2 \cos(q_1 + q_2)\dot{q}_2\big) = 0 \tag{15}$$

We rearrange (15) to solve for the base velocity $\dot{x}_B$, and combine the result with (14) to derive the generalized Jacobian J_G for the 2-DOF trolley model

$$\begin{bmatrix} v_x \\ v_z \end{bmatrix} = \begin{bmatrix} J_{11}(q) + A_1(q) & J_{12}(q) + A_2(q) \\ J_{12}(q) & J_{22}(q) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J_G \dot{q} \tag{16}$$

where:

$$\begin{aligned} A_1(q) &= -\frac{(m_1 a_1 \cos(q_1) + m_2(l_1 \cos(q_1) + a_2 \cos(q_1 + q_2))}{(m_{UAV} + m_1 + m_2)} \\ A_2(q) &= -\frac{m_2 a_2 \cos(q_1 + q_2)}{(m_{UAV} + m_1 + m_2)} \end{aligned} \tag{17}$$

The inverse kinematic control equation for the 2-DOF trolley model is the following:

$$\dot{q} = J_G^{-1} v_e \tag{18}$$

The inverse kinematics is solved through a time discretization and an iterative process: at every timestep k, the end-effector velocities are known and, by assuming that joint angles q_k do not significantly change between two consecutive timesteps k and $k+1$, the generalized Jacobian can be calculated through (16) by substituting $q = q_k$.

4.2. Simulation Results with Ideal Control

We first simulated the UAM with the ideal control assumption, using the inverse kinematic model (18) with the corresponding generalized Jacobian (16). The base controller is assumed to behave perfectly to restrict motion to horizontal translations only (ideal control). The geometric and dynamic parameters for the UAM model are as defined in Table 1.

We performed the linear trajectory illustrated in Section 3 for the trolley model. The inverse kinematics was solved through the procedure shown in Section 4.1 using the MATLAB simulator mentioned in Section 3. The resulting joint velocities were imported in the ADAMS model of the UAM to perform the dynamic simulation of the task, where the ideal control was implemented as a rotation and vertical translation constraint. We measured the end-effector position and compared it to the desired position. The time discretization used for the inverse kinematics is $\Delta t = 1 \times 10^{-3}$ s. The end-effector position error is smaller than 8×10^{-5} m at all times. The stroboscopic trajectories are shown in Figure 6. Figure 7 shows the joint angles, UAV horizontal translation, and torque disturbance transmitted from the manipulator to the base during the task.

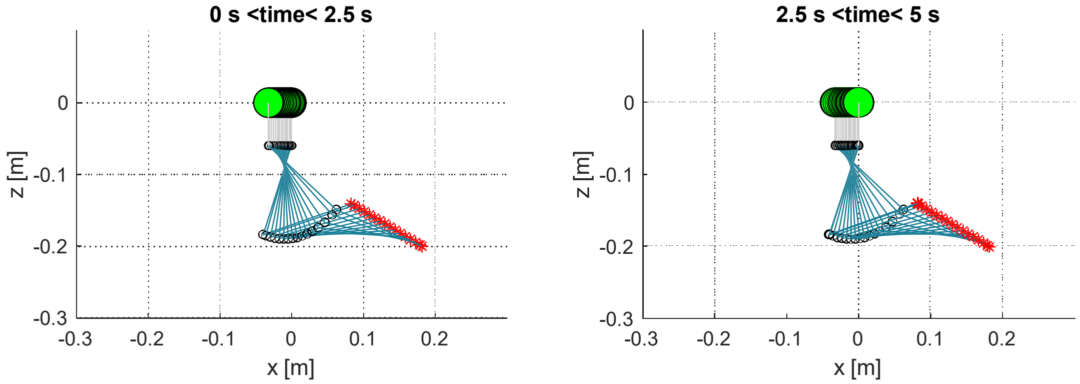

Figure 6. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the linear trajectory task for the 2-DOF trolley model and ideal control assumption. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

Our UAV base translations are minimal (3 cm displacement) and it returns to the original position at the end of the trajectory. The torque disturbance is up to 2 Nm magnitude, which is counteracted by our ideal controller. However, in a real scenario with a not ideal controller, the torque disturbance is likely to cause both vertical and rotational displacements of the base during the manipulation task.

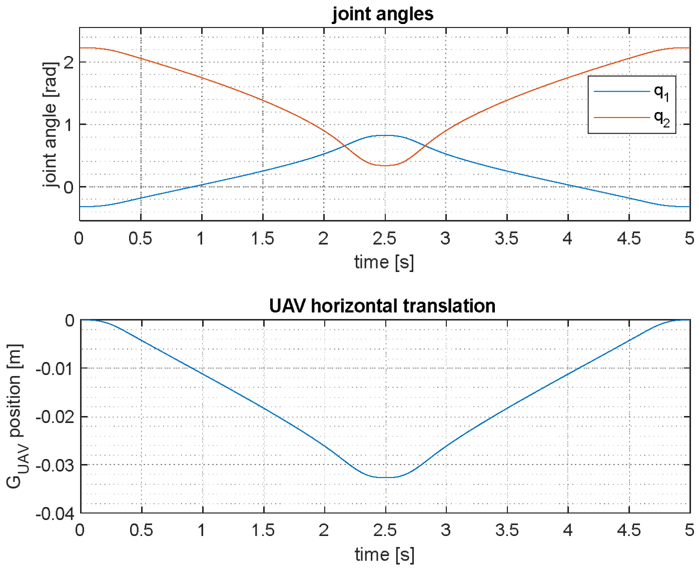

Figure 7. *Cont.*

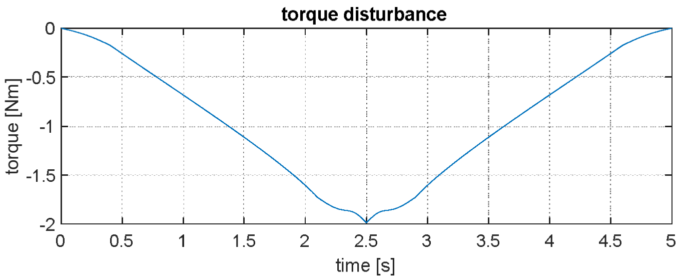

Figure 7. Manipulator (**top**) and UAV (**middle**) motions, and torque disturbance (**bottom**) for the ideal 2-DOF trolley case.

4.3. Simulation Results with PID Control

We simulated the same line trajectory, with the 2-DOF trolley assumption generalized Jacobian (16) and inverse kinematic Equation (18). The resulting joint velocities were then imported in the ADAMS model for the dynamic simulation, but in this case we relaxed the rotation and vertical translation constraints assumption (ideal control). The base motions were instead counteracted with the PID controller force U_1 and torque U_2 defined in (2) in order to validate (16) under more realistic conditions. In the dynamic simulation carried out with ADAMS, the results showed an end-effector position error of 1.3 m, which is equivalent to the UAV base translation. The UAV rotates during the manipulation task due to the torque disturbance. After the task is completed, it takes several seconds for the PID base controller to counteract this motion. At the end of the task, a horizontal translation of 1.3 m of the base remains, which cannot be successfully adjusted by the PID controller. Figure 8 illustrates these results.

From these results, it is evident that our ideal control assumption in the formulation of the generalized Jacobian results in unsuccessful tracking of the linear trajectory, as well as significant base motion, when a realistic PID control is used. This motivated us to develop a complete generalized Jacobian that includes all the displacements of the base, as well as an inverse kinematic method that includes controller feedback forces being applied on the base, and other external forces. The full generalized Jacobian considering all possible base translations and external forces is presented in the next section.

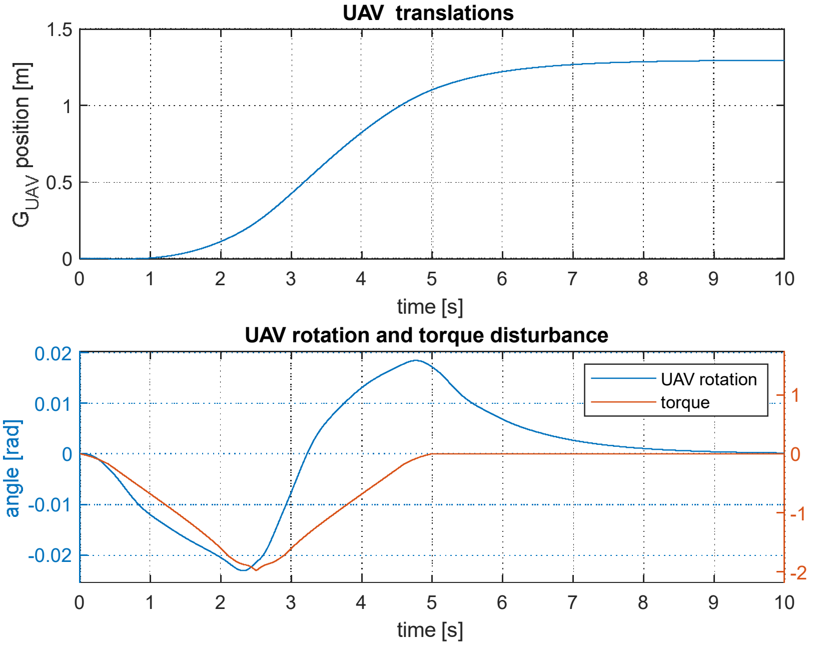

Figure 8. UAV horizontal translation (**top**) and rotation/torque disturbance (**bottom**) during the linear trajectory task for the 2-DOF trolley model with PID control assumption.

5. Generalized Jacobian with External Forces

5.1. Derivation of Generalized Jacobian with External Forces for Trajectory Tracking

A more complete model of the UAM includes the effect of the base translations and rotation as well as the external forces applied by the controller on the UAM and the gravity force. This is based on the generalized Jacobian approach outlined in Section 2.3.

The manipulator modeled is a serial link arm, that is mounted at an offset r_m (as defined in Section 2.1) from the UAV base CoG G_{UAV}. The reaction forces between the base and manipulator are considered internal, and the controller forces U_1 and U_2 as well as the gravity force on the system are considered external.

First, we consider the fact that the change in linear momentum $L(t) \in \mathbb{R}^3$ over time of a mechanical system occurs due to external forces $F(t) \in \mathbb{R}^3$:

$$\frac{dL(t)}{dt} = F(t) \tag{19}$$

where $L(t) = [L_x(t) \;\; L_y(t) \;\; L_z(t)]^T$ is the linear momentum, and $F(t) = [F_x(t) \;\; F_y(t) \;\; F_z(t)]^T$ is the force with its three components along x_I, y_I, and z_I. Similarly, the change in angular momentum $K(t) \in \mathbb{R}^3$ occurs due to external torques $\tau(t) \in \mathbb{R}^3$:

$$\frac{dK(t)}{dt} = \tau(t) \tag{20}$$

where $K(t) = \begin{bmatrix} K_x(t) & L_y(t) & L_z(t) \end{bmatrix}^T$ and $\tau(t) = \begin{bmatrix} \tau_x(t) & \tau_y(t) & \tau_z(t) \end{bmatrix}^T$ are the angular momenta and external torques about each axis. Therefore, the linear and angular momenta of a system with external forces and torques at time t can be obtained by integrating the forces and torques over time, starting from the initial time t_0:

$$\begin{aligned} L(t) &= \textstyle\int_{t_0}^{t} F(t)dt + L(t_0) \\ K(t) &= \textstyle\int_{t_0}^{t} \tau(t)dt + K(t_0) \end{aligned} \tag{21}$$

We discretize these equations to obtain the linear and angular momentum at each timestep $k = \Delta t$, where Δt is the sampling time interval. The discretized forms of (21) at each k are the following:

$$\begin{aligned} L_{k+1} &= F_k \Delta t + L_k \\ K_{k+1} &= \tau_k \Delta t + K_k \end{aligned} \tag{22}$$

The momentum equation is expressed in terms of the mass/inertia matrix of the system $H \in \mathbb{R}^{6\times(6+n_e)}$, base generalized coordinates $\dot{q}_b \in \mathbb{R}^6$, and manipulator generalized coordinates $\dot{q}_m \in \mathbb{R}^{n_e}$:

$$H\begin{bmatrix} \dot{q}_b(t) \\ \dot{q}_m(t) \end{bmatrix} = \begin{bmatrix} L(t) \\ K(t) \end{bmatrix} \tag{23}$$

We partition H into a base component $H_b \in \mathbb{R}^{6\times 6}$ and manipulator component $H_m \in \mathbb{R}^{6\times n_e}$:

$$H = \begin{bmatrix} H_b & H_m \end{bmatrix} \tag{24}$$

We combine the kinematic equation in (9) with (23) and (24), to obtain the end-effector velocity in terms of our Jacobians, mass/inertia matrices, and momenta:

$$v_{e,k+1} = (J_m - J_b\, H_b^{-1}\, H_m)\dot{q}_m + J_b H_b^{-1}\begin{bmatrix} L_k \\ K_k \end{bmatrix} = J_G\, \dot{q}_m + J_b H_b^{-1}\begin{bmatrix} L_k \\ K_k \end{bmatrix} \tag{25}$$

where J_G is the generalized Jacobian of our system. To find the required joint velocities, we solve for $\dot{q}_m$ (dropping the notation of timestep for brevity):

$$\dot{q}_m = J_G^{-1}\left(v_e - J_b H_b^{-1}\begin{bmatrix} L_k \\ K_k \end{bmatrix}\right) \tag{26}$$

To find the current momenta of the system based on the external forces as stated in (22), we first define the following vectors as illustrated in Figure 9:

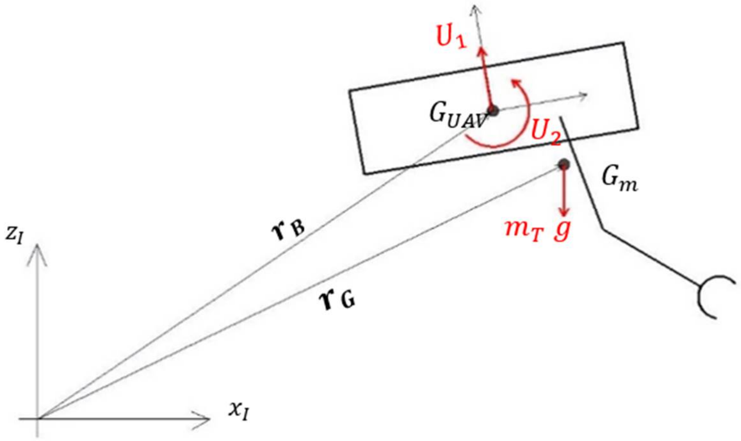

Figure 9. External forces on the UAM for the generalized Jacobian formulation.

- r_B: vector from the inertial frame origin to the UAV CoG G_{UAV};
- r_G: vector from the inertial frame origin to the whole system CoG G_m.

The linear momentum at timestep k under the external forces comprising the controller force and UAM gravity is found with the following equation:

$$L_{k+1} = L_k + \left(A_0 \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_T g \end{bmatrix} \right) \Delta t \tag{27}$$

where A_0 is the rotation matrix of the UAV, and m_T is the total mass of the UAM ($m_T = m_1 + m_2 + m_3 + m_{UAV}$).

The angular momentum at timestep k under the external controller and gravity torques is found with the following equation:

$$K_{k+1} = K_k + \left(-v_{b,k-1} \times L_k + \begin{bmatrix} 0 \\ U_2 \\ 0 \end{bmatrix} + (r_G - r_B) \times \begin{bmatrix} 0 \\ 0 \\ m_T g \end{bmatrix} \right) \Delta t \tag{28}$$

where $-v_{b,k-1}$ is the vector of linear velocities of the base (first 3 rows of $\dot{q}_b$) at timestep $k-1$, assuming that the velocities do not change significantly between the two timesteps.

Each iteration of the presented method can be seen as composed of three steps: (1) the computation of forward kinematics to find the link and end-effector positions from given joint angles and UAV pose; (2) the computation of the dynamics in terms of linear and angular momentum of the global system through (27) and (28); (3) the computation of the inverse kinematics of the manipulator through the generalized Jacobian with given linear and angular momenta of the whole system.

5.2. Simulation Results

In order to validate the method proposed in Section 5.1, the inverse kinematics was solved in MATLAB using a timestep of $\Delta t = 1 \times 10^{-3}$ s and then the resulting joint velocities were used to perform the dynamic simulation of the task in ADAMS.

We first considered the case of the 2-DOF nonredundant manipulator tracking the line trajectory: as opposed to the results obtained with the 2-DOF trolley model with PID control feedback (Section 4.3), the position error of the end-effector is now smaller than 4×10^{-5} m at all times even if the base has horizontal translations bigger than 10 cm. During the operation, the torque transmitted by the manipulator to the base makes the UAV rotate, as a result U_1 has a horizontal component and the base translates along x_I. At the end of the operation, the control torque U_2 stabilizes the rotation of the base, which tends to return to its initial position. The stroboscopic trajectories, joint angles, UAV translations and rotation, and torque disturbance transmitted from the manipulator to the UAV are shown in Figures 10 and 11. The computational time to perform the inverse kinematics for tracking the above-mentioned linear trajectory (5 s) with the 2-DOF manipulator is 2.51 s.

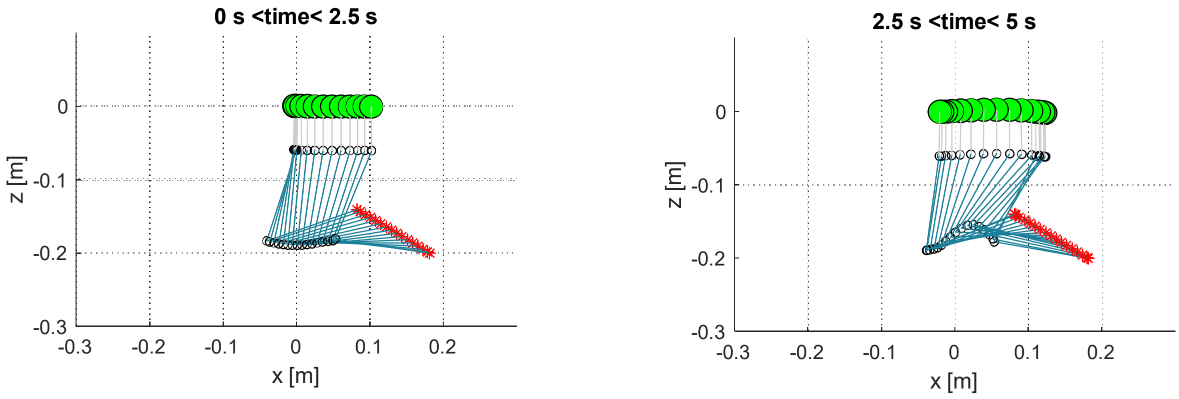

Figure 10. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the linear trajectory task for the 2-DOF UAM using the generalized Jacobian with external forces. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

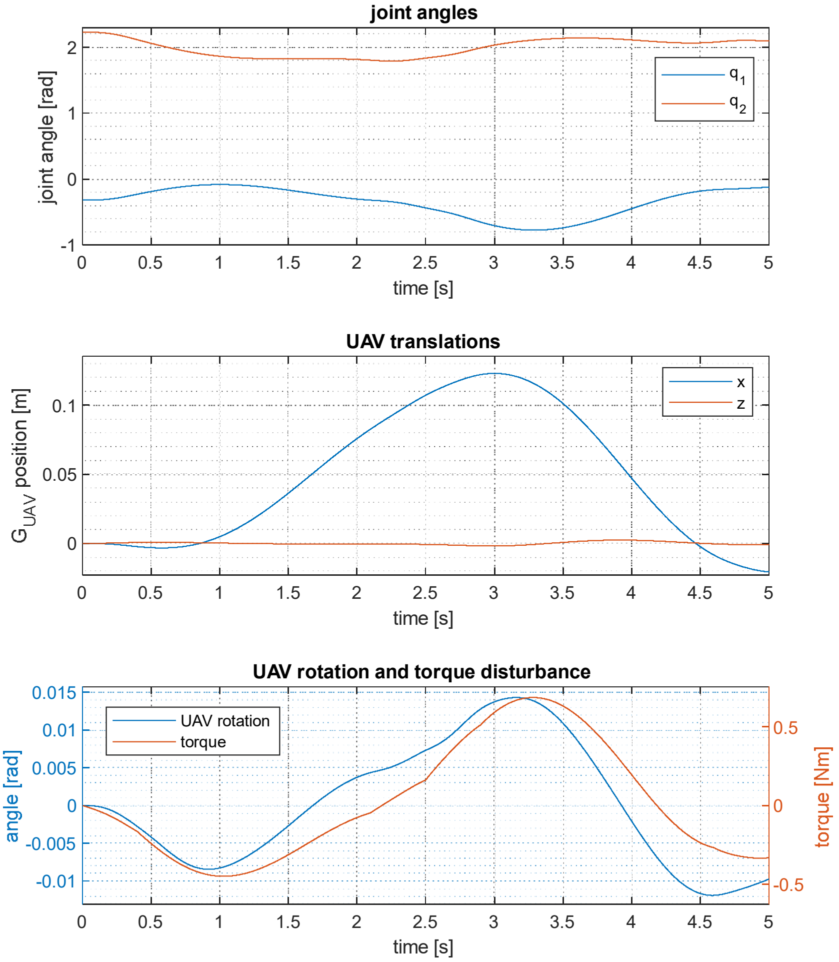

Figure 11. Manipulator (**top**) and UAV motions and torque disturbance (**middle** and **bottom**) for the 2-DOF UAM using the generalized Jacobian with external forces.

We also tracked the same trajectory with a 3-DOF redundant manipulator using the same approach, where the right pseudo-inverse of the generalized Jacobian (8) was used to find the minimum joint velocities solution. The end-effector position error is smaller than 4×10^{-5} m at all times. As in the 2-DOF manipulator case, the base rotates and translates due to the torque disturbances from the manipulator. The stroboscopic trajectories, joint angles, UAV translations and rotation, and torque disturbance transmitted from the manipulator to the UAV are shown in Figures 12 and 13.

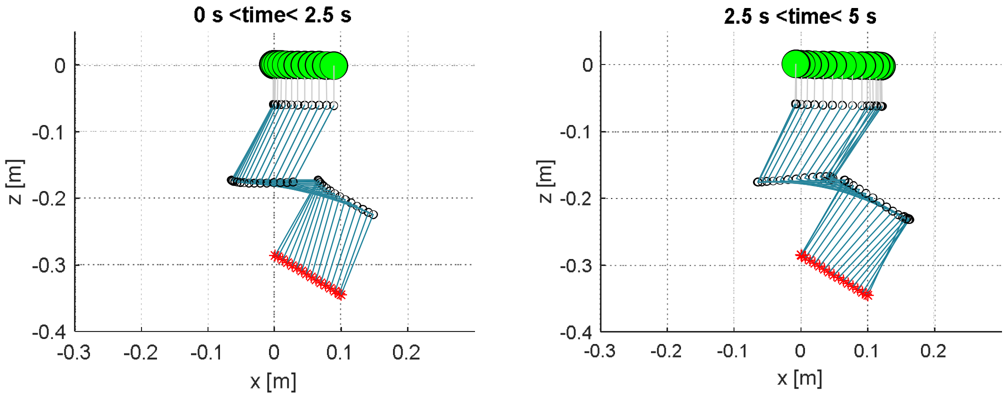

Figure 12. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the linear trajectory task for the 3-DOF redundant UAM using the generalized Jacobian with external forces. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

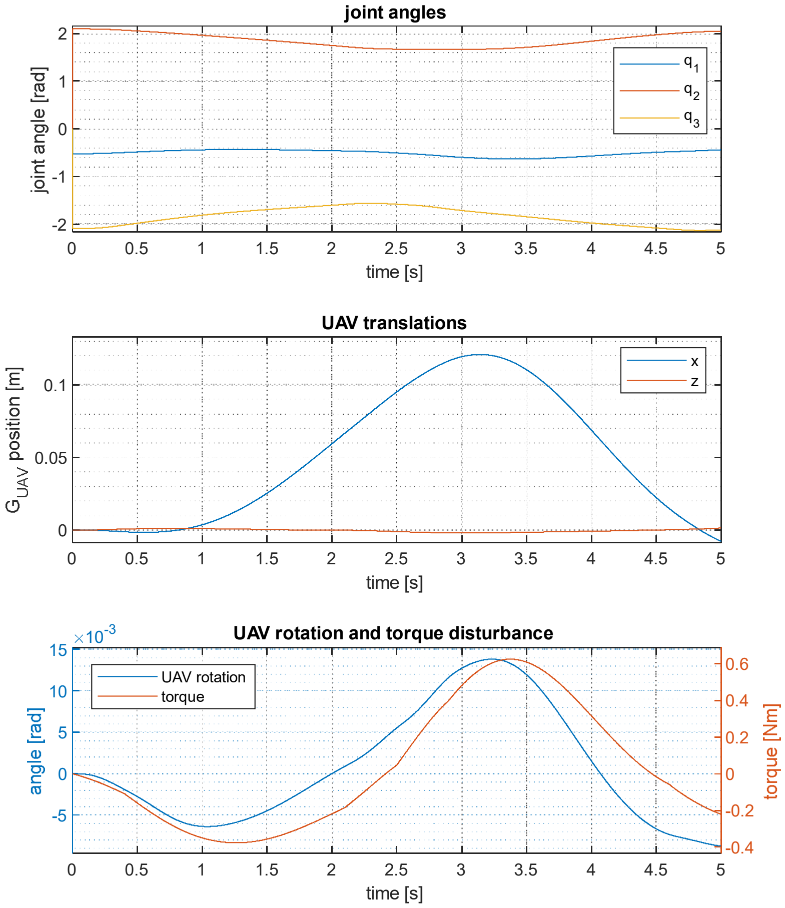

Figure 13. Manipulator (**top**) and UAV motions and torque disturbance (**middle** and **bottom**) for the 3-DOF UAM using the generalized Jacobian with external forces performing a linear trajectory task.

Finally, we tested the method on the circular trajectory with the redundant 3-DOF manipulator. The results in Figures 14 and 15 show that, at the beginning of the task, the UAV translates horizontally following the motion of the end-effector, reaching a maximum displacement of about 9 cm from its initial position, and that in the last part of the task the UAV tends to return to the initial position. In this case, the UAV also shows a visible vertical translation bigger than 1 cm, however, the end-effector position error is always smaller than 12×10^{-5} m.

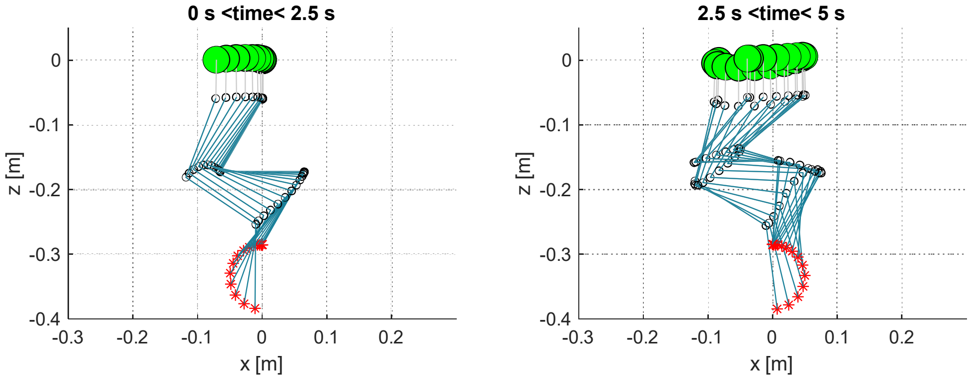

Figure 14. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the circular trajectory task for the 3-DOF redundant UAM using the generalized Jacobian with external forces. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

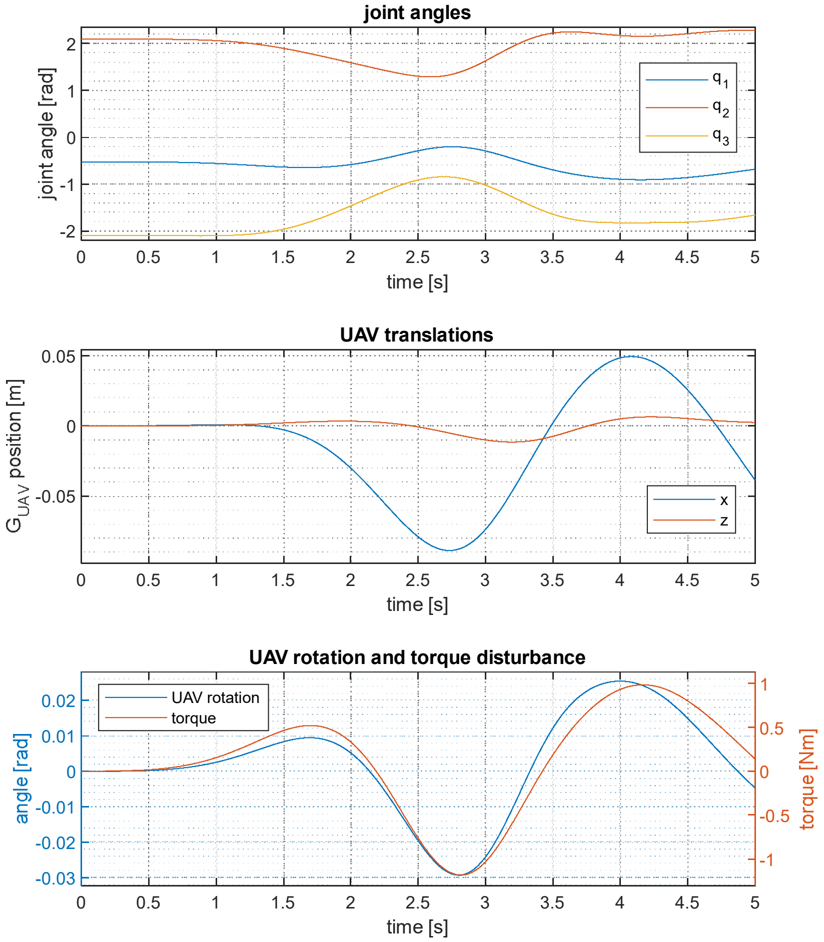

Figure 15. Manipulator (**top**) and UAV motions and torque disturbance (**middle** and **bottom**) for the 3-DOF UAM using the generalized Jacobian with external forces performing a circular trajectory task.

The computational time for the tracking of the linear and circular trajectories using the 3-DOF manipulator is, respectively, 3.96 s and 3.89 s, both faster than the trajectory duration of 5 s.

6. Extended Generalized Jacobian for a 3-DOF Redundant Manipulator with Full Base Movement

6.1. Derivation of Extended Jacobian

For a redundant manipulator ($n_e > 2$), i.e., with more than 2 joints actuated in a 2D plane and a 2D velocity tracking task $v_e \in \mathbb{R}^2$, we have infinite solutions and formulate the inverse kinematics problem as a minimization of joint velocities, as illustrated in Section 2.2. We can also leverage the redundancy of the manipulator to add additional tasks/constraints, for example, to minimize the torque disturbance from the manipulator to the base. The general form of the additional constraint for a 3-DOF manipulator is $f_1\dot{q}_1 + f_2\dot{q}_2 + f_3\dot{q}_3 = b$. Adding this constraint to the generalized Jacobian for a 3-DOF serial link manipulator allows us to find a unique solution:

$$\begin{bmatrix} v_x \\ v_z \\ b \end{bmatrix} = \begin{bmatrix} & J_G & \\ f_1 & f_2 & f_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{29}$$

As outlined earlier in (20) and (21), the change in angular momentum is caused by an external torque. The forces/torques acting on the manipulator are shown in Figure 16. The equation of the derivative of the angular momentum of the manipulator about the base CoG is:

$$\frac{dK_{man}}{dt} = -v_B \times L_{man} + \tau_B + \left(\tau_m + r_m \times \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} \right) \tag{30}$$

where v_B is the velocity of the base CoG (the pole of the angular momentum), L_{man} is the linear momentum of the manipulator, τ_B is the torque about the base CoG due to the gravity force ($m_m g$, where m_m is the manipulator mass) and due to the external force on the end-effector (F_e). The term in the brackets is the change in angular momentum of the manipulator caused by the reaction torques exerted by the base on the manipulator and, as this term is equal and opposite to the torque disturbance transmitted from the manipulator to the UAV, we impose that it is null. So, we obtain:

$$\frac{dK_{man}}{dt} = -v_B \times L_{man} + \tau_B \tag{31}$$

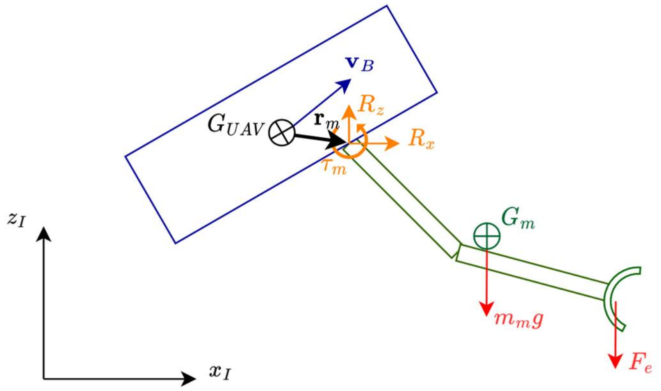

Figure 16. Forces and torques on the manipulator for the extended Jacobian formulation. Centers of gravity are the circles with crosses (black for the UAV, green for the manipulator), external forces are in red, reaction forces/torques in orange.

Similarly to the approach used in (23) and (24), we can express the manipulator angular momentum in terms of the generalized coordinates of the base $\dot{q}_b$ and manipulator $\dot{q}_m$:

$$K_{man} = H_{man} \begin{bmatrix} \dot{q}_b \\ \dot{q}_m \end{bmatrix} = \begin{bmatrix} H_{b,man} & H_{m,man} \end{bmatrix} \begin{bmatrix} \dot{q}_b \\ \dot{q}_m \end{bmatrix} \tag{32}$$

where H_{man} is partitioned into $H_{b,man} \in \mathbb{R}^{3\times 3}$ and $H_{m,man} \in \mathbb{R}^{3\times n_e}$.

We discretize Equation (31): we write the derivative of the angular momentum as $(K_{man,k} - K_{man,k-1}\ )/\Delta t$ and we isolate $K_{man,k}$ which can be written as a function of the unknown joint velocities $\dot{q}_{m,k}$:

$$K_{man,k} = \begin{bmatrix} H_{b,man}(q_{k-1}) & H_{m,man}(q_{k-1}) \end{bmatrix} \begin{bmatrix} \dot{q}_{b,k} \\ \dot{q}_{m,k} \end{bmatrix} = K_{man,\ k-1} + (\tau_{B,k-1} - v_{B,k-1} \times L_{man,k-1})\ \Delta t \tag{33}$$

In (33), we take all joint angles and manipulator momenta for timestep $k-1$ with the assumption that these quantities do not change significantly over one timestep.

As in the derivation of the generalized Jacobian with external forces, we express the generalized velocities of the UAV $\dot{q}_{b,\ k}$ in terms of $\dot{q}_{m,\ k}$ and of the angular momenta of the UAM to obtain the inverse kinematics equation for the extended Jacobian, dropping the discretized notation for joint angles (q_{k-1}) for brevity:

$$\begin{bmatrix} H_{b,man}\ H_b^{-1}\ H_m + H_{m,man} \end{bmatrix} \dot{q}_{m,k} = \left(K_{man,\ k-1} + (\tau_{B,k-1} - v_{B,k-1} \times L_{man,k-1})\Delta t - H_m^T H_b^{-1} \begin{bmatrix} L_k \\ K_k \end{bmatrix} \right). \tag{34}$$

This gives us the constraint in the form of (29), where the term $\begin{bmatrix} H_{b,man}\ H_b^{-1}\ H_m + H_{m,man} \end{bmatrix}$ is $\begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}$ and the term on the right is b. If the manipulator has more than one redundant DOF, the extended Jacobian is formed by adding more tasks and corresponding constraints to the generalized Jacobian in order to make it a square matrix.

6.2. Simulation Results for Trajectory Tracking

We evaluated the reduction of torque disturbance for the linear and circular trajectories, we focused on the horizontal translation of the UAV base and on the ability to track the desired trajectory. We solved the inverse kinematics for the linear and circular trajectories in MATLAB by the procedure shown in Section 6.1 with a timestep of $\Delta t = 1 \times 10^{-3}$ s. As in previous cases, we imposed the resulting joint velocities in the ADAMS model of the UAM to perform the dynamic simulation of the task operations.

Initially, we considered the linear trajectory: the position error is below 8×10^{-5} m and the UAV rotation is smaller than 5×10^{-6} rad. Compared to the generalized Jacobian with external forces (Section 5.2), the torque disturbance is decreased by four orders of magnitude. The manipulator moves while minimizing the horizontal distance between the UAV and manipulator centers of gravity. This results in a drawback of the method since the workspace is reduced. The vertical UAV translation is not visibly affected by the introduction of the extended Jacobian. Figures 17 and 18 show the stroboscopic trajectories, joint angles, UAV translations, UAV rotation, and reaction torque.

For the circular trajectory, the end-effector position error is again smaller than 12×10^{-5} m, the base translation is mainly vertical since it reaches 9.5 mm in the y_I direction and it is smaller than 1 mm in the x_I direction. The torque disturbance transmitted from the manipulator to the UAV and UAV rotation have the same order of magnitude as for the linear trajectory. The stroboscopic trajectories, joint angles, UAV translations and rotation, and torque disturbance are shown in Figures 19 and 20.

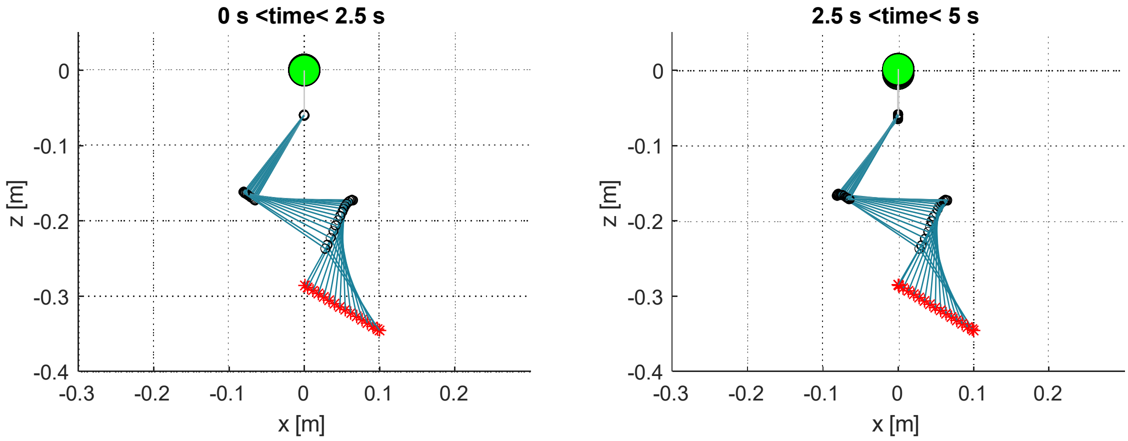

Figure 17. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the linear trajectory task for the 3-DOF redundant UAM using the extended Jacobian. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

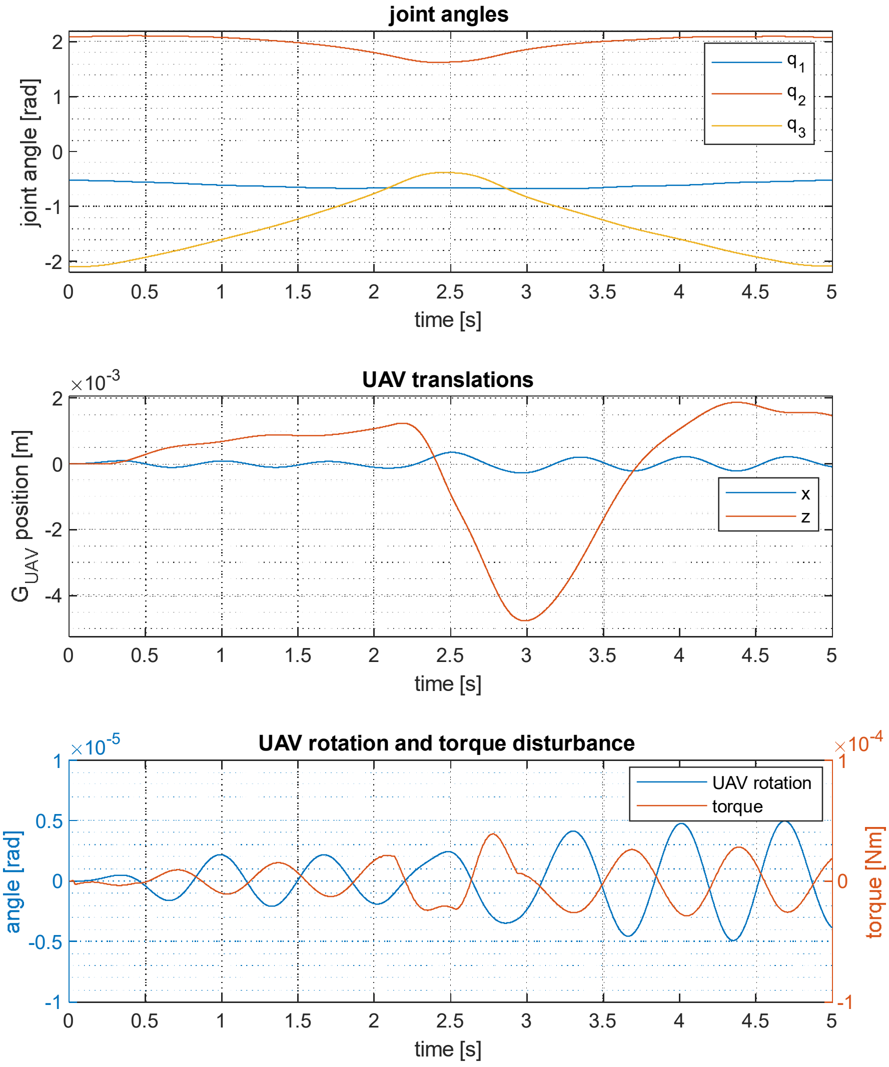

Figure 18. Manipulator (**top**) and UAV motions and torque disturbance (**middle** and **bottom**) for the linear trajectory task with the 3-DOF UAM using the extended Jacobian.

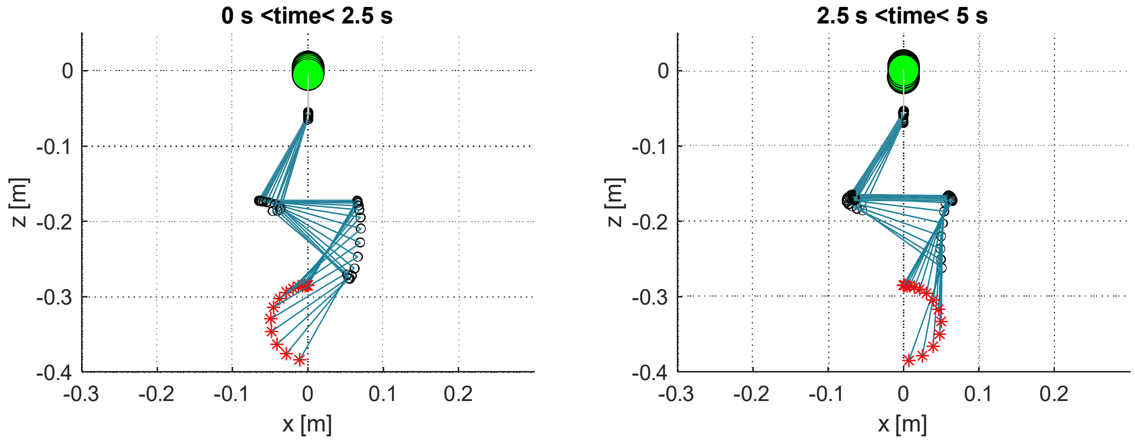

Figure 19. Stroboscopic trajectories for the first half (**left**) and second half (**right**) of the circular trajectory task for the 3-DOF redundant UAM using the extended generalized Jacobian. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses.

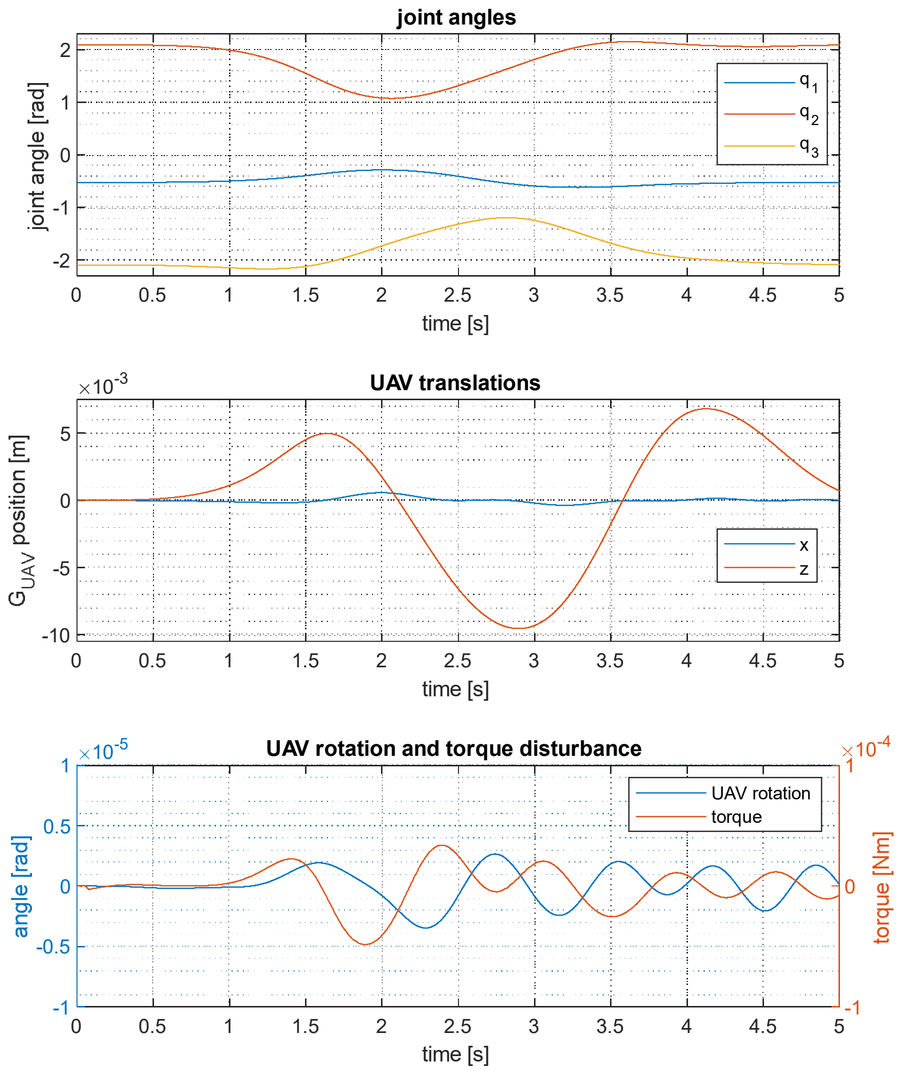

Figure 20. Manipulator (**top**) and UAV motions and torque disturbance (**middle** and **bottom**) for the circular trajectory task with the 3-DOF UAM using the extended generalized Jacobian.

By using the extended generalized Jacobian, the computational time for the tracking of the linear and circular trajectories (with time duration of 5 s) by the 3-DOF manipulator is 3.89 s and 3.88 s, respectively. This result is comparable to the generalized Jacobian method computational speed and faster than the trajectory duration of 5 s.

7. Load Picking Case

7.1. Model Derivation

When using the generalized Jacobian for inverse kinematic control, a concentrated mass payload attached to the end-effector can be considered as an additional external force and torque: the new contributions are accounted for in the terms F_k and τ_k of (22). In the extended generalized Jacobian, the additional torque is also considered in the term τ_B of (31).

The load force on the end-effector consists of three terms: weight force, inertial force, and contact force. We consider the load initially lying on a stiff but elastic frictionless support in order to eliminate discontinuities in force application, which reduces the inaccuracy of taking the force at the previous timestep in the algorithms outlined in Sections 5 and 6. We use a simple spring model for the contact; therefore, after the payload is grasped by the manipulator, the load force applied on the end-effector is defined by the following equations and can thus be calculated at each timestep:

$$F_{Load,EE} = \begin{bmatrix} 0 \\ 0 \\ m_c g \end{bmatrix} - m_c \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{z}_c \end{bmatrix} + F_{contact} \tag{35}$$

$$F_{contact} = \begin{bmatrix} 0 \\ 0 \\ \max(0,\ k_S(z_S - z_c)) \end{bmatrix} \tag{36}$$

where m_c is the mass of the load, x_c, y_c, z_c are the Cartesian coordinates of the load, k_S is the contact spring stiffness, and z_S is the unloaded height of the contact spring.

7.2. Simulation Setup

For the load picking case, the end-effector is required to move horizontally 8 cm and vertically 12 cm through a linear trajectory which lasts 12 s in the following stages:

1. The forward motion (without payload) is accomplished from 0 s $\geq t >$ 1.5 s.
2. The end-effector velocities are then null from 1.5 s $\geq t \geq$ 2 s and the grasp starts at $t = 1.6$ s so we can start to add the payload in this interval without large discontinuities in the load force.
3. The backward motion begins at $t > 2$ s with a slow trapezoidal velocity profile to avoid too fast changes in the load force which would cause the UAV to lose altitude excessively.
4. At $t = 7$ s, $F_{contact}$ is null and the payload is entirely sustained by the manipulator so we increase the end-effector backward velocity.

Desired velocity profiles of the end-effector are obtained through the superposition of profiles defined by (13) appropriately scaled and translated in order to have acceleration continuity. We considered a load mass $m_c = 0.2$ kg and a contact spring stiffness $k_S = 200$ Nm^{-1}, and z_S is chosen such that the load is in equilibrium at its initial position (equal to the position of the end-effector at $t = 1.6$ s). The end-effector velocity profile for the load picking task is shown in Figure 21.

The task is executed by a 3-DOF manipulator. We compare the task execution using both the generalized Jacobian with external forces (26) and the extended generalized Jacobian method, (29) and (34).

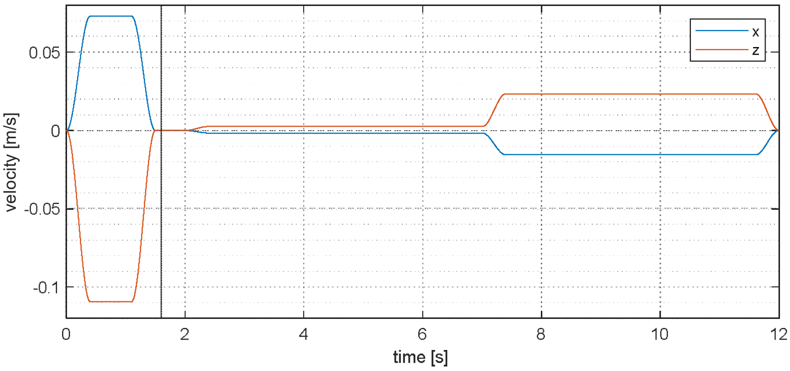

Figure 21. Desired end-effector velocities for the load picking task. The vertical gray line indicates the start time of contact.

7.3. Simulation Results

For the same load picking task, inverse kinematics was solved in MATLAB discretizing with a timestep $\Delta t = 1 \times 10^{-3}$ s, firstly by using the generalized Jacobian and then using the extended generalized Jacobian. Resulting joint velocities were imported in the ADAMS model of the UAM to dynamically simulate the task. The results comparing the stroboscopic trajectories of load picking experiments for both algorithms are shown in Figure 22. The simulated joint angles, UAV translations, and UAV rotations and reaction torques are shown in Figure 23. For both the solution methods, the trajectory is successfully tracked with minimal position errors: for the generalized Jacobian with external forces, the end-effector position error is always smaller than 12×10^{-5} m, and for the extended generalized Jacobian the end-effector position error is smaller than 6×10^{-4} m at all times.

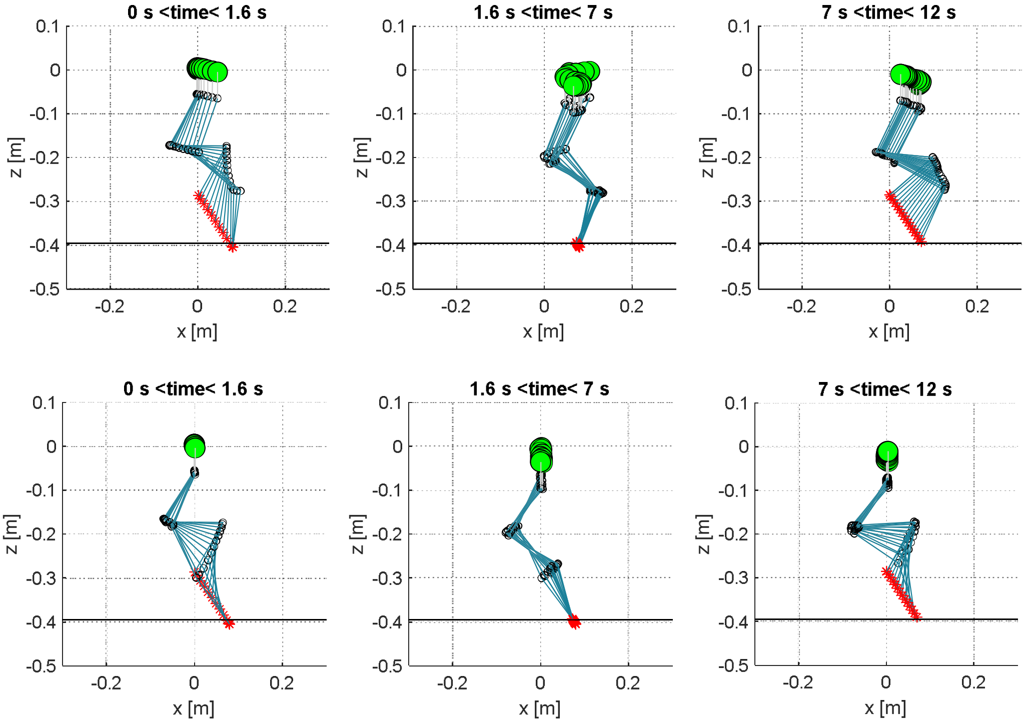

Figure 22. Stroboscopic trajectories for the load picking task for the 3-DOF redundant UAM using the generalized Jacobian with external forces (**top**) and using the extended generalized Jacobian (**bottom**). The left column shows the forward motion until the start of the grasping, the central one shows the slow load lifting part, and the right one the fast part of the backward motion. The UAV base is the large green marker, joints are the black circles, links are the blue lines, and end-effector positions tracked are the red crosses; the black line shows the height of the undeformed contact spring ($z = z_S$).

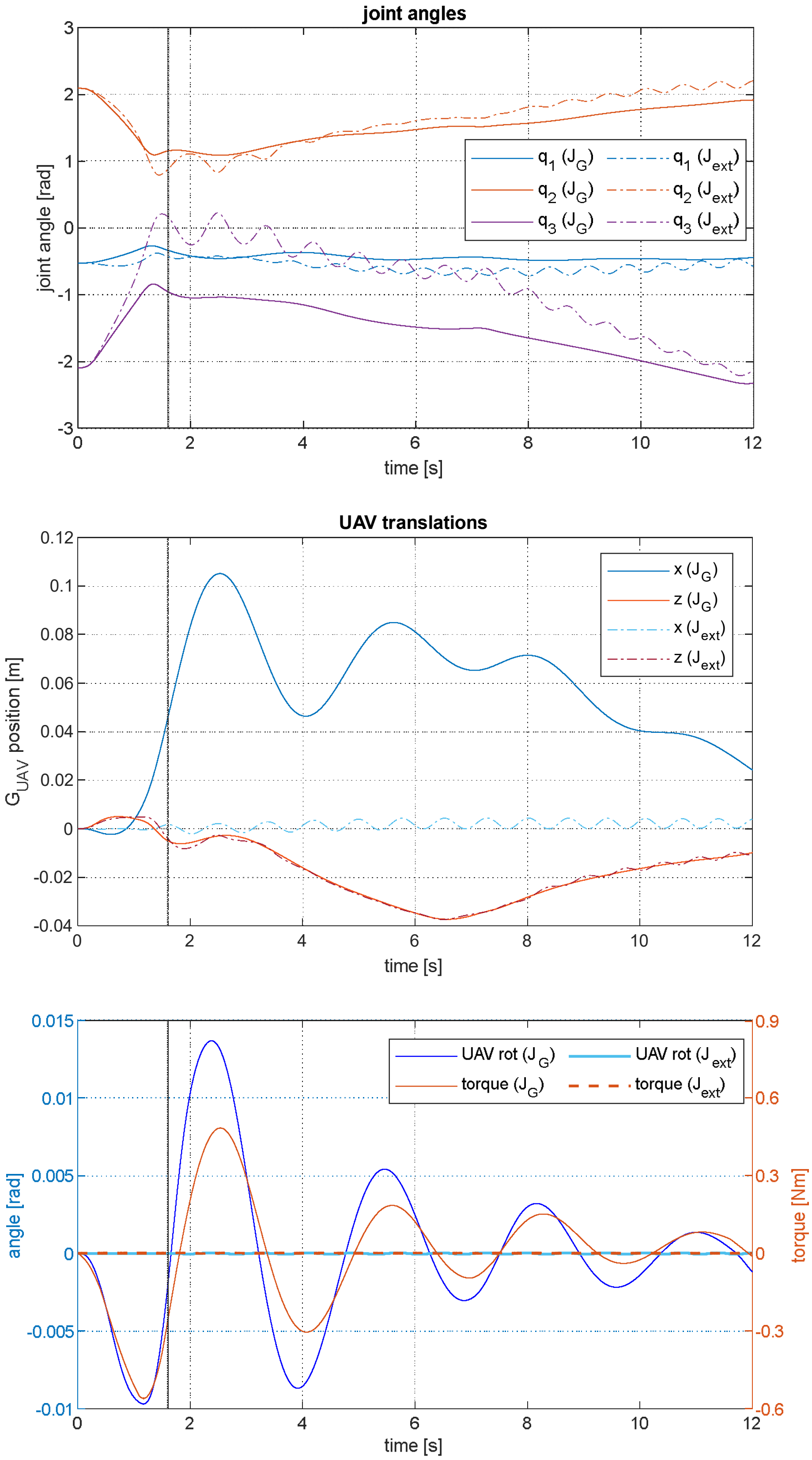

Figure 23. Comparison between generalized Jacobian with external forces (J_G) and extended generalized Jacobian (J_{ext}) solutions for the 3-DOF redundant UAM: manipulator motions (**top**), UAV translations (**middle**), UAV rotation and torque disturbance from the manipulator to the UAV (**bottom**). The vertical dark line indicates the start time of contact for the grasp.

For the generalized Jacobian method, at the beginning of the task the manipulator exerts a torque of almost 0.6 Nm on the base and causes the UAV to rotate and translate. The base UAV also translates after the load grasping along x_I up to 10 cm, overtaking the initial position of the load (8 cm). Then, the UAV moves backward reaching an x_I of about 2.5 cm at the end of the task. In this movement, we can see oscillations on torque disturbance, UAV rotation, and horizontal translation since the UAV control torque requires time to stabilize the angle caused by the initial disturbance from the manipulator.

For the extended generalized Jacobian method, we can see that the torque disturbance is reduced by two orders of magnitude compared to the previous case and is now smaller than 1.5×10^{-3} Nm, the UAV rotation is now smaller than 5×10^{-5} rad, negligible compared to the generalized Jacobian solution. Consequently, the UAV, except for some small oscillations of the millimeter magnitude due to the algorithm, keeps its initial horizontal position. Indeed, we are imposing a zero torque transferred to the UAV during the picking of a load not aligned with the CoG of the system, and this most probably causes these small oscillations. Please note that, in the inverse kinematics algorithm for the extended generalized Jacobian, the external torque in (34) is obtained from a previous timestep, so it is accounted for with a delay of Δt.

For both methods, we can see that the UAV altitude increases in the first second because of the inertia of the manipulator (moving downwards), then the UAV reduces its elevation (along z_I) until the load loses contact with the elastic ground; there are no substantial differences in vertical displacement between the two methods.

For the load picking case (which has a trajectory duration of 12 s), the overall computational time obtained through the generalized Jacobian method is 9.88 s, and the computational time obtained through the extended generalized Jacobian method is comparable: 9.28 s.

8. Conclusions

In this research, we proposed two control methods based on the generalized Jacobian to solve the inverse kinematics of an aerial manipulation and to minimize the UAV base reaction torque and, therefore, its horizontal motion. The main novelty of this work is related to the application of differential kinematics methods to a UAM through the inclusion of external forces and torques acting on the system in the inverse kinematics solution, and to the minimization of the torque disturbance transferred from the manipulator to the UAV. In our case, external forces and torques include weight, UAV control inputs, and forces exerted from the load to be picked to the end-effector. The method can also be extended to other cases where the forces can be modeled, for example to the case of interaction forces with the environment. The effectiveness of both methods was confirmed by the capability of tracking a linear and a circular trajectory, and in a load picking test case. In the extended generalized Jacobian case, the task was carried out with zero reaction torque transferred from the manipulator to the UAV. This is particularly important since the horizontal translation of the UAV is almost zero and, therefore, the aerial manipulator can safely operate in the proximity of walls.

For the generalized Jacobian with external forces method, the base has noticeable rotations and large horizontal translations during task execution due to the UAV dynamics and the UAV controls used. By including an additional task (null manipulator reaction torque), UAV translations are minimized in the case of the extended generalized Jacobian.

The simulated cases showed that using the extended generalized Jacobian method significantly reduces the horizontal translation of the base, however, this is achieved by reducing the available workspace. This is because the main contribution to the torque disturbance on the UAV is the weight of the manipulator, so it is constrained to move reducing the horizontal translation of its center of gravity.

Our MATLAB implementation runs faster than real time, which indicates that the algorithm can be implemented in a real-world system with minimal optimization modifications required. The computational times could be significantly reduced by implementing the algorithm in a compiled programming language (such as C++), which would also enable inverse kinematic control with more complex models of the UAV and manipulator.

Future extensions to this research will be to conduct real-world experiments on a UAM in a planar control scenario. To implement this algorithm on a physical aerial manipulator, we require precise estimation/measure of the UAV pose, end-effector position, joint positions, and contact forces. This will require integration of sensors such as IMUs, force-sensors, and vision systems such as Vicon.

Although the algorithms outlined in this research are validated in 2D, this approach can easily be generalized to 3D systems and different types of manipulators. To address the problem of reduced workspace while minimizing the disturbance torques in the extended generalized Jacobian, we can also investigate the feasibility of adding a controlled counterweight for torque balancing. This would require further modifications to the dynamic modeling of the UAM as well as the inverse kinematic control algorithm. Another topic of research that will be part of future work is the investigation of different solution methods, e.g. using constrained least squares, which would allow for constraints on joint positions and velocities and faster computational times.

Author Contributions: Conceptualization, S.C.; methodology, S.C. and A.P.; software, A.P.; investigation, S.C., A.P. and Y.V.; writing—original draft preparation, A.P. and Y.V.; writing—review and editing, S.C., A.P. and Y.V.; supervision, project administration, funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was carried out in the framework of the project “DynAeRobot—Development and validation of a new dynamically balanced aerial manipulator” (project no. BIRD213590), funded under the BIRD 2021 program promoted by the University of Padova.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ollero, A.; Tognon, M.; Suarez, A.; Lee, D.; Franchi, A. Past, Present, and Future of Aerial Robotic Manipulators. *IEEE Trans. Robot.* **2021**, *38*, 626–645. [CrossRef]
2. Franchi, A.; Carli, R.; Bicego, D.; Ryll, M. Full-Pose Tracking Control for Aerial Robotic Systems with Laterally Bounded Input Force. *IEEE Trans. Robot.* **2018**, *34*, 534–541. [CrossRef]
3. Trujillo, M.Á.; Martínez-de Dios, J.R.; Martín, C.; Viguria, A.; Ollero, A. Novel Aerial Manipulator for Accurate and Robust Industrial NDT Contact Inspection: A New Tool for the Oil and Gas Inspection Industry. *Sensors* **2019**, *19*, 1305. [CrossRef] [PubMed]
4. Park, S.; Lee, J.; Ahn, J.; Kim, M.; Her, J.; Yang, G.-H.; Lee, D. Odar: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1907–1918. [CrossRef]
5. Bodie, K.; Brunner, M.; Pantic, M.; Walser, S.; Pfändler, P.; Angst, U.; Siegwart, R.; Nieto, J. Active Interaction Force Control for Contact-Based Inspection with a Fully Actuated Aerial Vehicle. *IEEE Trans. Robot.* **2020**, *37*, 709–722. [CrossRef]
6. Khamseh, H.B.; Janabi-Sharifi, F.; Abdessameud, A. Aerial Manipulation—A Literature Survey. *Robot. Auton. Syst.* **2018**, *107*, 221–235. [CrossRef]
7. Ruggiero, F.; Lippiello, V.; Ollero, A. Aerial Manipulation: A Literature Review. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1957–1964. [CrossRef]
8. Lee, J.Y.; Cook, Z.; Barzilov, A.; Yim, W. Control of an Aerial Manipulator with an On-Board Balancing Mechanism. In *ASME International Mechanical Engineering Congress and Exposition*; American Society of Mechanical Engineers: Phoenix, Arizona, 2016; Volume 50541, p. V04AT05A019.
9. Yang, H.; Lee, D. Dynamics and Control of Quadrotor with Robotic Manipulator. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 5544–5549.
10. AlAkhras, A.; Sattar, I.H.; Alvi, M.; Qanbar, M.W.; Jaradat, M.A.; Alkaddour, M. The Design of a Lightweight Cable Aerial Manipulator with a CoG Compensation Mechanism for Construction Inspection Purposes. *Appl. Sci.* **2022**, *12*, 1173. [CrossRef]
11. Abuzayed, I.; Itani, A.R.; Ahmed, A.; Alkharaz, M.; Jaradat, M.A.; Romdhane, L. Design of Lightweight Aerial Manipulator with a CoG Compensation Mechanism. In Proceedings of the 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 4 February–9 April 2020; pp. 1–5.

12. Cocuzza, S.; Rossetto, E.; Doria, A. Dynamic Interaction between Robot and UAV in Aerial Manipulation. In Proceedings of the 2020 19th International Conference on Mechatronics-Mechatronika (ME), Prague, Czech Republic, 2–4 December 2020; pp. 1–6.
13. Papadopoulos, E.; Dubowsky, S. On the Nature of Control Algorithms for Free-Floating Space Manipulators. *IEEE Trans. Robot. Autom.* **1991**, *7*, 750–758. [CrossRef]
14. Caccavale, F.; Siciliano, B. Kinematic Control of Redundant Free-Floating Robotic Systems. *Adv. Robot.* **2001**, *15*, 429–448. [CrossRef]
15. Cocuzza, S.; Pretto, I.; Debei, S. Least-Squares-Based Reaction Control of Space Manipulators. *J. Guid. Control Dyn.* **2012**, *35*, 976–986. [CrossRef]
16. Cocuzza, S.; Pretto, I.; Debei, S. Novel Reaction Control Techniques for Redundant Space Manipulators: Theory and Simulated Microgravity Tests. *Acta Astronaut.* **2011**, *68*, 1712–1721. [CrossRef]
17. Umetani, Y.; Yoshida, K. Others Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix. *IEEE Trans. Robot. Autom.* **1989**, *5*, 303–314. [CrossRef]
18. Nenchev, D.N. Redundancy Resolution through Local Optimization: A Review. *J. Robot. Syst.* **1989**, *6*, 769–798. [CrossRef]
19. Tringali, A.; Cocuzza, S. Globally Optimal Inverse Kinematics Method for a Redundant Robot Manipulator with Linear and Nonlinear Constraints. *Robotics* **2020**, *9*, 61. [CrossRef]
20. Tringali, A.; Cocuzza, S. Finite-Horizon Kinetic Energy Optimization of a Redundant Space Manipulator. *Appl. Sci.* **2021**, *11*, 2346. [CrossRef]
21. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*, 1st ed.; Springer Publishing Company, Incorporated: New York, NY, USA, 2008; ISBN 1-84628-641-7.
22. Cocuzza, S.; Pretto, I.; Debei, S. Reaction Torque Control of Redundant Space Robotic Systems for Orbital Maintenance and Simulated Microgravity Tests. *Acta Astronaut.* **2010**, *67*, 285–295. [CrossRef]