Robotica (2023), **41**, pp. 3004–3021
doi:10.1017/S0263574723000826

RESEARCH ARTICLE

A novel approach to control four multi-rotor drones in cooperative paired control using relative Jacobian

Keletso Z. Thebe, Rodrigo S. Jamisola Jr. and Larona P. Ramalepa

Department of Mechanical, Energy and Industrial Engineering, Faculty of Engineering and Technology, Botswana International University of Science and Technology, Palapye, Botswana
Corresponding author: Keletso Z. Thebe; Emails: tk19100045@studentmail.biust.ac.bw, kkelkeletso@hotmail.com

Received: 12 August 2022; **Revised:** 22 March 2023; **Accepted:** 28 May 2023; **First published online:** 5 July 2023

Keywords: cooperative drones; relative Jacobian; holistic control; paired control; Jacobian-based control; rotorS gazebo

Abstract

This work presents a new formulation to holistically control four cooperative multi-rotor drones controlled in two pairs. This approach uses a modular relative Jacobian with components consisting of the Jacobians of each individual drone. This type of controller relies mainly on the relative motion between the drones, consequently releasing unnecessary constraints inherent to the control of drones in absolute motion. We present the derivations of all the necessary equations of the modular relative Jacobian to control the four multi-rotor drones. We also present the derivations of the Jacobian for each drone. We implement our proposed method in the Gazebo RotorS simulation using four hexa-rotor drones modeled from Ascending Technologies called firefly drones. We present the simulation results and analyze them to show the effectiveness of our proposed approach.

1. Introduction

Drone control strategies significantly increased in the recent decade coupled with studies in cooperative drone control [1–3], with examples shown in refs. [4–7]. Cooperative control approaches are influenced by the application for which the system is to be used. Some of these approaches include aerial transportation [6], surveying and mapping [8], search and rescue [9, 10], reconnaissance [11], and many others. Many researchers are working on different classes of drone control approaches, for example, formation control, behavior-based approach, leader-follower, virtual leader, etc. [12]. In most of these methods, some excel and some have drawbacks. For example, studies show that the virtual leader and formation control classified methods are much reliable in modern war whereas, on the other hand, have problems when it comes to making full use of the neighbors' communication information for other aspects of control [13].

This paper addresses the problem faced with the usual approach in controlling multi-rotor drones to maneuver in 3D space using tilting heuristics by running the rotors at different speeds as seen in Fig. 1 [14–17].

The use of relative Jacobian, however, increases the degrees of freedom in the null space. Another problem addressed is that most approaches to drone cooperative control use leader-follower and master-slave principles where the drones will rely on the leader drone. This approach is not reliable because when the master or leader drone fails, all the follower drones will also fail. Manually controlling the drone by radio controllers may not be advantageous because there is a chance of human errors in controlling the drones. With the Jacobian controller, the drones become autonomous. This work intends to derive and use new formulations of drone control via relative Jacobian to control four multi-rotor drones. In addition, we also derive and use new formulation in Jacobian-based control of individual drones. This type of controller allows centralized and decentralized control, including obstacle avoidance in the null

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

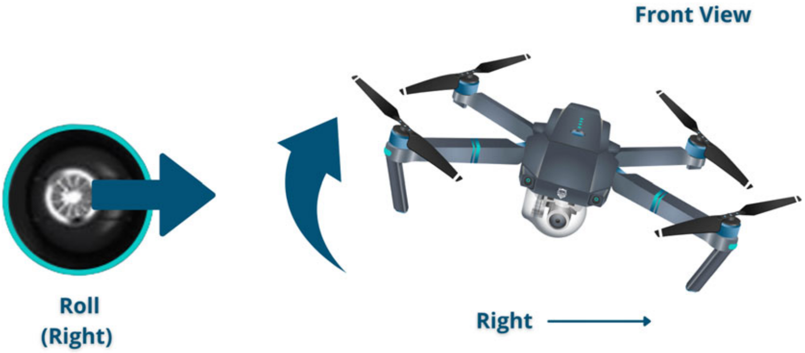

Figure 1. *The usual way of controlling drone tilting heuristics during a right roll maneuver [16].*

space [18]. This work claims the following contribution: (1) a holistic control of four drones as a single robot, (2) paired control of drones similar to dual-arm controllers, (3) Jacobian-based control for each drone, and (4) modular relative Jacobian of four drones with components consisting of individual Jacobian.

The use of Jacobian to control a drone uses the same principle as the use of Jacobian to control a mobile base robot. The only difference is the additional degrees of freedom provided by the z-axis. In the use of Jacobian-based control for multi-drone systems (MDSs), the same principle is used when applied to a multi-arm manipulator system. The main reason for the identical principle is that the Jacobian is derived based on the relative movement of one end-effector with respect to the other. Thus, it is called a relative Jacobian. In the case of multi-arm manipulator systems, the relative movement between end-effectors releases unnecessary constraints in the null space and transforms them into controllable degrees of freedom. This affords more dexterity in the multi-arm manipulator system. The relative movement between end-effectors did not require a physical connection between them. In the same way, we use the relative Jacobian to control the relative movement between drones, where each drone contains both the base and end-effector of the "manipulator." As in the case of a multi-arm manipulator system, the relative movement between drones does not require a physical connection between them. In addition, unnecessary constraints in the null space now become useful, controllable degrees-of freedom for the MDS.

2. Initial implementation of relative Jacobian controller

The Jacobian matrix brought up the solution to the inverse kinematics problem of robotics, this is the problem robots have achieving coordinated motion relative to a fixed reference frame [19]. The work in ref. [20] showed that the Jacobian matrix computes a transformation between the joint velocities and the end-effector velocities. A lot of research has been done on Jacobian control, especially with robot manipulators. Several studies have proved the relative Jacobian controller to be very satisfactory in robot manipulators, and several studies have been published; these include [21–23]. With these studies on the use of relative Jacobian on robot manipulators, we assume the derivation of relative Jacobian can be expressed for the desired number of manipulators or robots intended to be controlled. For example, the study in ref. [21] first introduced a modular relative Jacobian for a dual-arm robot in full space consisting of two manipulators A and B each with their absolute Jacobians $J_A = [J_{pA}; J_{oA}]$ and $J_B = [J_{pB}; J_{oB}]$ expressed in terms of position and orientation velocity components J_{pA}, J_{oA} for manipulator A and J_{pB}, J_{oB} for manipulator B.

This can be extended to more manipulators, like three (3) as in ref. [23] or four (4) in ref. [24]. Therefore, for the example of two manipulators, the modular relative Jacobian:

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

$$J_R = \left[-{}^2\Psi_3 \; {}^2\Omega_1 \, J_A \;\; {}^2\Omega_4 \, J_B \right] \tag{1}$$

where

$${}^i\Psi_j = \begin{bmatrix} I & -S\left({}^ip_j\right) \\ 0 & I \end{bmatrix} \quad \text{and} \quad {}^i\Omega_j = \begin{bmatrix} {}^iR_j & 0 \\ 0 & {}^iR_j \end{bmatrix} \tag{2}$$

the symbol ${}^i\Psi_j$ is the wrench transformation matrix, ${}^i\Omega_j$ is the full space rotation matrix, ${}^ip_j$ is a position vector, ${}^iR_j$ is an orientation matrix, $S({}^ip_j)$ is a skew-symmetric matrix, such that frame j is expressed with respect to frame i. The symbol I is an identity matrix.

As depicted in (1), we can see that for one to derive the relative Jacobian J_R, one only needs the wrench transformation matrix ${}^i\Psi_j$, the rotation matrix ${}^i\Omega_j$, and the Jacobians of the stand-alone manipulators J_A and J_B to form the relative Jacobian J_R. So to extend this to four robots A, B, C, and D, for example, each robot's absolute Jacobian J_A, J_B, J_C, and J_D are all incorporated to form the relative Jacobian of the four robots like in ref. [24].

The following paragraphs show the previous work of relative Jacobian implementation on other robots. This is the foundation of this paper's implementation of cooperative drones.

A more compact expression of relative Jacobian based on individual manipulator Jacobians. In ref. [21], the first derivation of a more compact expression of relative Jacobian for parallel (dual-arm) manipulators was presented. Here, relative Jacobian was expressed in terms of the individual manipulator Jacobians. The aim is to control the combined manipulators as a single manipulator.

Modular relative Jacobian control for bipedal robot. A modular relative Jacobian method that was used for dual arms in ref. [21] was used for generating walking motion for bipedal robots [22]. The two legs that are being experimented with are equal to the two arms of the dual arm, whereas the feet here are the same as the two end-effectors of the dual arm. Each leg has three degrees of freedom (3-DOFs), so combined they form a 6-DOFs bipedal robot. The 3-DOFs relative position of the bipedal robot's feet are controlled as the relative end-effectors of the relative Jacobian, and the other remaining 3-DOFs are used to control the posture of the robot.

Modular relative Jacobian for combined 3-arm parallel manipulators. The work in ref. [23] presented a formulation of a modular relative Jacobian for controlling combined manipulators as an individual manipulator having one end-effector. This relative Jacobian was designed for 3-arm parallel manipulators; it is expressed relative to the reference frames at the end-effector of the manipulators. The existing information about each stand-alone manipulator is used in the derivation of the relative Jacobian for the combined system. The motive behind the study of parallel robots includes several applications such as modular micro parallel robots [25], modular control architecture [26], modular designs of parallel robots, etc.

Bio-inspired holistic control through modular relative Jacobian for combined four-arm robots. This work was inspired by other living organisms having a different number of limbs which come in pairs, for example, mammals having four, insects with six limbs, arachnids having eight, etc [24]. This was an attempt to copy the biological method living organisms use in controlling paired legs. In this work, however they used dual arms in place of the paired legs. The experiment was to use these dual arms in opposite and adjacent pairs to achieve holistic control of a large four-legged animal. A modular relative Jacobian was used to control each dual arm as an individual manipulator with a single end-effector and is expressed in terms of the Jacobians of each of the stand-alone manipulators. In this work, the two opposite pairs of legs are treated as single end-effector dual arms, and then, these two dual arms are combined to form a single end-effector four-arm robot. The four-arm controller uses the same principle as a single end-effector controller of a dual arm and thus results in a single end-effector controller of a four-arm. A modular relative Jacobian of the four arms is derived.

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

3. Relative Jacobian cooperative controller implementation on multi-rotor drones

The intention of this paper, however, is to implement this controller on multi-rotor drones and test its efficiency as in cooperative robot manipulators. This work intends to apply the same concept of the relative Jacobian cooperative controller on multi-rotor drones in the same way it is implemented on cooperative manipulators.

The usual way multi-rotor drones maneuver is by using a tilt heuristic by running the rotors at different speeds [14, 15, 27]. This approach has unnecessary constraints, resulting in limited accuracy because there is less control over the movement of the drone. The tilt heuristics are used to compensate for the lack of actuation to change the xy-orientation (rotation about a horizontal axis), especially for quadrotors because they are under-actuated with respect to the full 6-DOF space. This is the case of a coupled movement, such that to change the xy-orientation, the position along the axis perpendicular to the axis of rotation also needs to change. However, this heuristic is not necessary for hexarotors if the propellers are properly configured because there are now six motors to move in the full 6-DOF space. If the propellers are configured properly, then each axis of movement can be independently controlled.

However, using Jacobian-based control the propellers have twist angles of 30 degrees about the y-axis so that the drones move in a quintic path from their initial positions to the desired final positions, without tilting heuristics. The desired positions assigned x_d, the desired orientation assigned x_{do}, desired velocity $\dot{x}_d$, and acceleration $\ddot{x}_d$. Time t is therefore input against all the aforementioned parameters which will result in $x_d(t) = [x_{dp}(t), x_{do}(t)]$, $\dot{x}_d(t) = [\dot{x}_{dp}(t), \dot{x}_{do}(t)]$, and $\ddot{x}_d(t) = [\ddot{x}_{dp}(t), \ddot{x}_{do}(t)]$, respectively. Thus, the drone's desired positions and orientations with their corresponding velocities and accelerations. On the assumption that each drone's initial position or start, time, $t = 0$, when the drone reaches the final desired position, the time will be $t = t_f$. Therefore, we can consider the desired translational motion without any orientation as $\dot{x}_{dp}(0) = \ddot{x}_{dp}(0) = 0$ and $\dot{x}_{dp}(t_f) = \ddot{x}_{dp}(t_f) = 0$.

The desired orientation R_d is determined to compute the rotational motions for each rotor, also ${}^{i}p_i$ which is the drone's arm length with respect to frame i, plus the desired acceleration, α_d, and the rotational velocity, ω_d. In our quintic computations, these values are all initialized to zero, and the path planning of position and orientation use parameterized quintic functions. With regard to orientation, our quintic path is parameterized by θ_d, $\dot{\theta}_d$, and $\ddot{\theta}_d$, and the desired initial and final values for the quintic computations will be: $\theta_d(0) = \ddot{\theta}_d(0) = 0$, $\dot{\theta}(t_f) = \ddot{\theta}_d(t_f) = 0$, $\theta_d(0) = \theta_{\text{init}}$, and $\dot{\theta}_d(t_f) = \theta_{\text{fin}}$.

These desired orientations with respect to xyz-axis are used to compute the desired orientation as shown in (3) below:

$$R_d = R_{z,\phi_d}\, R_{y,\theta_d}\, R_{x,\psi_d} \tag{3}$$

Each individual drone's rotor configurations are configured in the following manner with respect to its body, ${}^{0}R_i$:

$$\begin{aligned}
{}^{0}R_1 &= R_{z,\phi}(\phi = 30^\circ)\;\; R_{y,\theta}(\theta = 30^\circ)\,,\\
{}^{0}R_2 &= R_{z,\phi}(\phi = 90^\circ)\;\; R_{y,\theta}(\theta = -30^\circ)\,,\\
{}^{0}R_3 &= R_{z,\phi}(\phi = 150^\circ)\;\; R_{y,\theta}(\theta = 30^\circ)\,,\\
{}^{0}R_4 &= R_{z,\phi}(\phi = -150^\circ)\;\; R_{y,\theta}(\theta = -30^\circ)\,,\\
{}^{0}R_5 &= R_{z,\phi}(\phi = -90^\circ)\;\; R_{y,\theta}(\theta = 30^\circ)\,, \text{ and}\\
{}^{0}R_6 &= R_{z,\phi}(\phi = -30^\circ)\;\; R_{y,\theta}(\theta = -30^\circ(\,)
\end{aligned}$$

The mass of the drone, $m = 1.5\,\text{kg}$, ${}^{i}p_i = [0.215, 0, 0.037]^T$, matrix of mass inertia, $I_m = [0.0347563, 0, 0; 0, 0, 0.0977]\,\text{kg/m}^2$, the moment proportionality, and rotor force moments constants, $a_f = [0, 0, 0.0045, 0, 0, 0.0001]^T$, these are all the same for all the four drones. The values of a_f are derived empirically. With this, when a drone is moving from its start location towards its goal, there is minimal change in the orientation of the drone. This is because each axis is independently controlled, so when the drone changes position, this movement does not affect its orientation. This proves the fact that for a drone to move with fewer restrictions, all we need is the right combination of drone configurations.

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

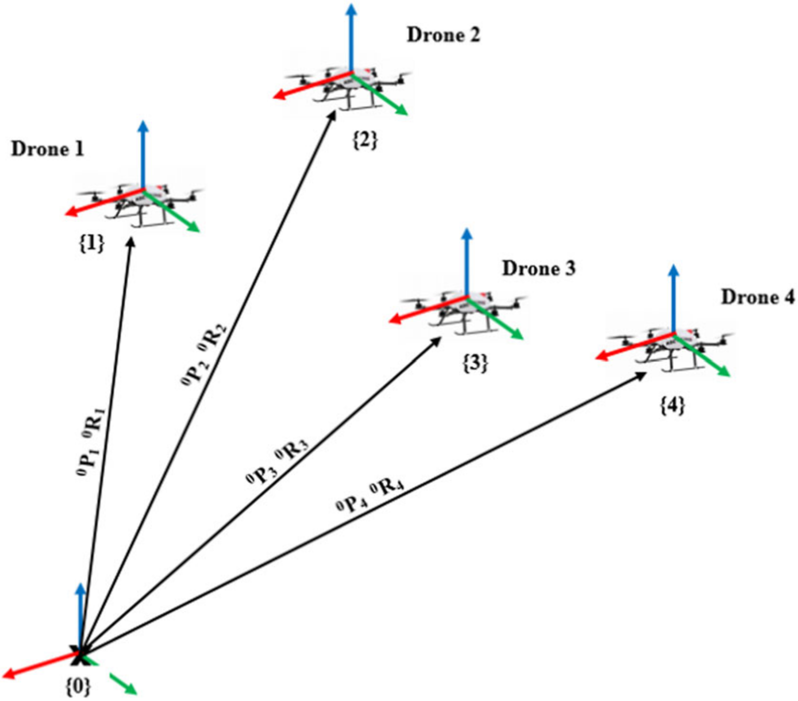

Figure 2. *Demonstration of absolute Jacobian control of individual drones, each drone referencing the world frame.*

This controller will be derived in two cases in the following subsections, 3.1 presents an individual or absolute Jacobian control of multi-rotor drones, and 3.2 presents the implementation of relative Jacobian cooperative control to control four multi-rotor drones in pairs.

3.1. Case 1: Derivation of individual Jacobian control of drones

As the first step to relative Jacobian derivation, we implemented absolute Jacobian for four individual drones. Figure 2 shows the graphical abstract of this implementation taking into consideration the kinematics and dynamics associated with drone control. This figure also shows the coordinate frames used in deriving the dynamic and kinematic models of individual drones. Each drone is attached a reference frame, frame {1} is attached to drone 1, frame {2} attached to drone 2, frame {3} attached to drone 3, and frame {4} attached to drone 4, all drones move with respect to frame 0 which is the world frame.

For this implementation, the initial positions $[x, y, z]$ of the four drones are as follows, drone 1 $[0, 0, 0]$, drone 2 $[0, 2, 0]$, drone 3 $[1, 0, 0]$, and drone 4 $[1, 2, 0]$. And the desired final positions are drone 1 $[0, 3, 1]$, drone 2 $[-2, 6, 1.5]$, drone 3 $[2, -4, 2]$, and drone 4 $[0.5, 5, 2.2]$. The Jacobian controller calculates the drone's rotor speeds automatically. The rotor velocities and Jacobians for each individual drone i are expressed with respect to the frame allocations in Fig. 2. The Jacobian for each drone i, J_i, is expressed as

$$J_i = {}^w\Omega_0 \left[{}^0\Omega_1\ {}^0\Omega_2\ {}^0\Omega_3\ {}^0\Omega_4\ {}^0\Omega_5\ {}^0\Omega_6\right] a_f \tag{4}$$

where a_f is a scalar factor, $[{}^0\Omega_1 \ldots {}^0\ \Omega_6]$ are the rotation matrices of each propeller frame $\{i\}$ with respect to the reference frame $\{0\}$, and the superscript $\{w\}$ means the world frame, such that the rotation matrix is expressed as

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

$$ {}^{i}\Omega_{j} = \begin{bmatrix} {}^{i}R_{j} & 0 \\ 0 & {}^{i}R_{j} \end{bmatrix}. \tag{5} $$

Because this is an absolute controller, reference frame $\{0\}$ corresponds to the world frame $\{w\}$. We define the velocity command for the ith drone to be

$$ V_i = J_i^T\, {}^{w}\Omega_0\, I_m\, u_i + J_i^T\, {}^{w}\Psi_0\, {}^{w}\Omega_0\, I_m\, \dot{x} + J_i^T\, mg \tag{6} $$

where the terms are defined as

$$ {}^{w}\Psi_0 = \begin{bmatrix} S({}^{w}\omega_0) & 0 \\ 0 & S({}^{w}\omega_0) \end{bmatrix} \tag{7} $$

$$ u_i = \ddot{x}_d + k_v(\dot{x}_d - {}^{0}\dot{x}_i) + k_p(x_d - {}^{0}x_i). \tag{8} $$

The symbol ${}^{w}\omega_0$ is the angular velocity of the reference frame $\{0\}$ with respect to the world frame $\{w\}$, I_m is the mass inertia matrix of each individual drone, u_i is the controller of drone i, ${}^{0}x_i$ is the actual position of drone i, ${}^{0}\dot{x}_i$ is the actual velocity of drone i, and $x_d, \dot{x}_d, \ddot{x}_d$ are the desired position, velocity, and acceleration for both drones. The symbol V_i is the rotor velocity for drone i. The values of k_v and k_p are tuned empirically.

3.2. Case 2: Derivation of relative Jacobian for 4 drones

This section will show the derivation of relative Jacobian for four drones as pairs (2) and quads (4); this implementation is similar to ref. [21] on dual arms. In the first case, the four drones were controlled individually; now we will show the derivation of relative Jacobian motion control for the four drones as different pairs and then further expand it to controlling all four as one. For pairs of drones 1 and 2, this is achieved by controlling drone 2 which is an end-effector drone with respect to drone 1 which is the base drone and for the second pair drone 4 is the end-effector drone and is controlled with respect to drone 3. Drone 1 and drone 3 use the world frame as their reference. The frame assignments in Fig. 3 are used in the derivation of this combined relative Jacobian-based control for these drone pairs. The drones use odometry sensor data from motion sensors to estimate the change in position over time. The position $[x, y, z]$ is estimated relative to each drone's starting location. This feedback position data of the base drones (1 and 3) are used to control the end-effector drones (2 and 4), and the results are compared with the desired position. This approach however does not use the leader-follower or master-slave theorems.

Now we present the derivation of relative Jacobian cooperative control for pair 1 (drone 1 and drone 2) and pair 2 (drone 3 and 4). The position of the end-effector drones with respect to the base drones can be expressed as

$$ {}^{1}p_2 = {}^{1}R_0\left({}^{0}p_2 - {}^{0}p_1\right) \quad \text{and} \quad {}^{3}p_4 = {}^{3}R_0\left({}^{0}p_4 - {}^{0}p_3\right) \tag{9} $$

and the velocities of the end-effector drones with respect to the base drones can be expressed below by getting the time derivative of the expression above,

$$ \begin{aligned} {}^{1}\dot{p}_2 &= \left[-{}^{1}R_0\, S\left({}^{0}p_1\right) + {}^{1}R_0\, S\left({}^{0}p_2\right)\right]\, {}^{0}\omega_1 - {}^{1}R_0\, {}^{0}\dot{p}_1 + {}^{1}R_0\, {}^{0}\dot{p}_2 \\ {}^{3}\dot{p}_4 &= \left[-{}^{3}R_0\, S\left({}^{0}p_3\right) + {}^{3}R_0\, S\left({}^{0}p_4\right)\right]\, {}^{0}\omega_3 - {}^{3}R_0\, {}^{0}\dot{p}_3 + {}^{3}R_0\, {}^{0}\dot{p}_4 \end{aligned} \tag{10} $$

where S is a skew-symmetric matrix. Because this controller takes a paired drone control, the reference frame $\{0\}$ corresponds to the world frame $\{w\}$. The symbol ${}^{1}R_0$ is the rotation of the world frame with respect to drone 1, and ${}^{3}R_0$ is the rotation of the world frame with respect to drone 3, which are the base drones, ${}^{0}\omega_1$ and ${}^{0}\omega_3$ are the angular velocities of the base drones with respect to the world frame, and ${}^{0}p_1$, ${}^{0}p_3$ and ${}^{0}p_2$, ${}^{0}p_4$ are the odometry positions of the base drones and the end-effector drones, respectively. The end-effector drone velocities with respect to the world frame are ${}^{0}\dot{p}_2$ and ${}^{0}\dot{p}_4$. The relative angular velocities for the first and second pairs are

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

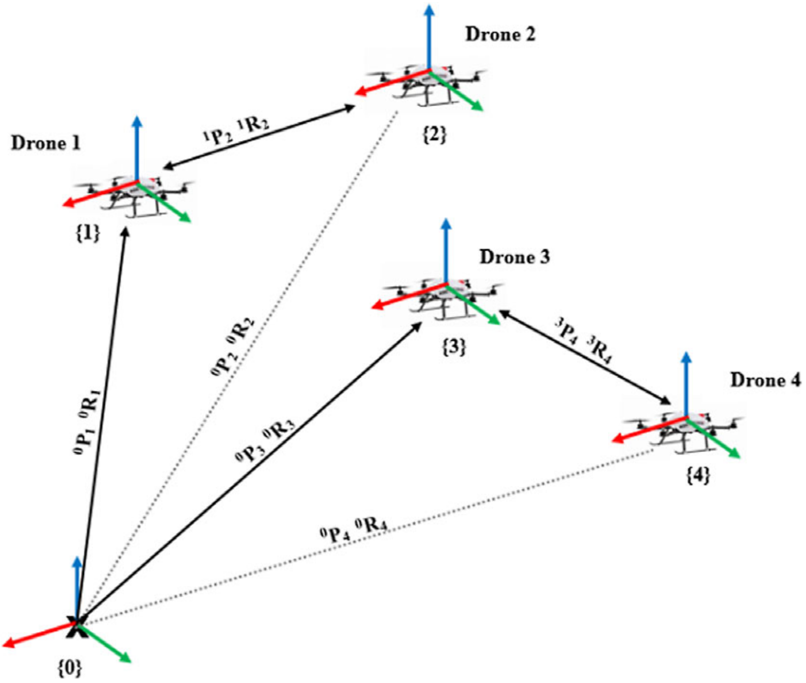

Figure 3. *Relative Jacobian-based control demonstration with frame assignments.*

$$
\begin{gathered}
{}^1\omega_2 = -{}^1R_0\,{}^0\omega_1 + {}^1R_0\,{}^0\omega_2 = {}^1R_0\,\begin{bmatrix}-I & I\end{bmatrix}\begin{bmatrix}{}^0\omega_1\\ {}^0\omega_2\end{bmatrix},\\
{}^3\omega_4 = -{}^3R_0\,{}^0\omega_3 + {}^3R_0\,{}^0\omega_4 = {}^3R_0\,\begin{bmatrix}-I & I\end{bmatrix}\begin{bmatrix}{}^0\omega_3\\ {}^0\omega_4\end{bmatrix}
\end{gathered}
\tag{11}
$$

having ${}^0\omega_2$ and ${}^0\omega_4$ as the end-effector drone angular velocities with respect to the world frame, and I is the identity matrix. The relative Jacobian for controlling the two pairs are

$$
\begin{gathered}
{}^1J_2 = \begin{bmatrix}-{}^1\Omega_0\,{}^1\Psi_2 J_1 & {}^1\Omega_0 J_2\end{bmatrix},\\
{}^3J_4 = \begin{bmatrix}-{}^3\Omega_0\,{}^3\Psi_4 J_3 & {}^3\Omega_0 J_4\end{bmatrix}
\end{gathered}
\tag{12}
$$

where ${}^1J_2$ and ${}^3J_4$ are the relative Jacobians for both relative motion controllers. The symbols ${}^1\Omega_0$ and ${}^3\Omega_0$ are the 6-DOF rotation of the world frame with respect to the base drone, ${}^1\Psi_2$ and ${}^3\Psi_4$ are the wrench transformation matrices of the end-effector frames with respect to the base frames, and J_1, J_2, J_3, J_4 are the absolute Jacobians of each drone.

To implement a full relative Jacobian-based cooperative controller, we need to use a torque controller instead of a velocity controller because the torque controller is more numerically stable when dynamics parameters are modeled. In our case, we modeled inertia and gravity parameters.

The relative position and relative velocity of frame {2} with respect to frame {1}, together with the relative Jacobian for the same frames, are shown below. This results in the rotor velocities of drone 1 and drone 2. In the same way, the velocities of drone 3 and drone 4 are derived from the relative positions of frame {3} and frame {4} and are shown below

$$
\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} = {}^1J_2^{+}\begin{bmatrix}{}^1\dot{p}_2\\ {}^1\omega_2\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}\dot{\theta}_3\\ \dot{\theta}_4\end{bmatrix} = {}^3J_4^{+}\begin{bmatrix}{}^3\dot{p}_4\\ {}^3\omega_4\end{bmatrix}.
\tag{13}
$$

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

The combined joint space inertia for both drone 1 and drone 2, D_{12}, and the combined joint space inertia for both drone 3 and drone 4, D_{34}, are shown below

$$D_{12} = \begin{bmatrix} J_1^T \, {}^w\Omega_1 \, I_m J_1 & 0 \\ 0 & J_2^T \, {}^w\Omega_2 \, I_m J_2 \end{bmatrix}$$

$$D_{34} = \begin{bmatrix} J_3^T \, {}^w\Omega_3 \, I_m J_3 & 0 \\ 0 & J_4^T \, {}^w\Omega_4 \, I_m J_4 \end{bmatrix} \tag{14}$$

where ${}^w\Omega_1, {}^w\Omega_2, {}^w\Omega_3, {}^w\Omega_4$ are the 6-DOF rotation matrices of each drone with respect to the world frame $\{w\}$, and I_m is the inertia for each drone which is the same for all because the drones are identical.

We use relative task space inertias shown below,

$$\Lambda_{R12} = {}^1J_2 \, D_{12}^+ \, {}^1J_2^T \quad \text{and} \quad \Lambda_{R34} = {}^3J_4 \, D_{34}^+ \, {}^3J_4^T \tag{15}$$

and use dynamically consistent inverse [28] to compute the inverse of each relative Jacobian

$${}^1J_2^+ = D_{12}^+ \, {}^1J_2^T \, \Lambda_{R12}^+ \quad \text{and} \quad {}^3J_4^+ = D_{34}^+ \, {}^3J_4^T \, \Lambda_{R34}^+. \tag{16}$$

The null-space controller is shown below

$${}^1Y_{R2} = I - {}^1J_2^+ \, {}^1J_2 \quad \text{and} \quad {}^3Y_{R4} = I - {}^3J_4^+ \, {}^3J_4 \tag{17}$$

where ${}^1Y_{R2}$ and ${}^3Y_{R4}$ are the null-space terms and I is an identity matrix. The relative task space controllers are shown below

$${}^1u_{R2} = {}^1\ddot{x}_{d2} + k_v\left({}^1\dot{x}_{d2} - {}^1\dot{x}_2\right) + k_p\left({}^1x_{d2} - {}^1x_2\right)$$

$${}^3u_{R4} = {}^3\ddot{x}_{d4} + k_v\left({}^3\dot{x}_{d4} - {}^3\dot{x}_4\right) + k_p\left({}^3x_{d4} - {}^3x_4\right) \tag{18}$$

where ${}^1x_2, {}^3x_4$ are the relative positions and ${}^1\dot{x}_2, {}^3\dot{x}_4$ are the relative velocities of the end-effector drone with respect to the base drone. The symbols ${}^1x_{d2}, {}^1\dot{x}_{d2}, {}^1\ddot{x}_{d2}$ are the desired position, velocity, and acceleration of drone 2 with respect to drone 1, while ${}^3x_{d4}, {}^3\dot{x}_{d4}, {}^3\ddot{x}_{d4}$ are the desired position, velocity, and acceleration of drone 4 with respect to drone 3. The translational parts of ${}^1x_2, {}^1\dot{x}_2$ are ${}^1p_2, {}^1\dot{p}_2$, and the rotational parts are ${}^1R_2, {}^1\omega_2$, respectively. In same way, the translational parts of ${}^3x_4, {}^3\dot{x}_4$ are ${}^3p_4, {}^3\dot{p}_4$, and the rotational parts are ${}^3R_4, {}^3\omega_4$, respectively.

The velocity commands sent to each drone rotor are

$$\dot{q}_{12} = {}^1J_2^+ \, \Lambda_{R12} \, {}^1u_{R2} + {}^1Y_{R2} \, \left(J_1^+ \, {}^w\Omega_1 \, I_m \, u_1\right)$$

$$\dot{q}_{34} = {}^3J_4^+ \, \Lambda_{R34} \, {}^3u_{R4} + {}^3Y_{R4} \, \left(J_3^+ \, {}^w\Omega_3 \, I_m \, u_3\right) \tag{19}$$

where $[\dot{\theta}_1, \dot{\theta}_2] = \dot{q}_{12}$ and $[\dot{\theta}_3, \dot{\theta}_4] = \dot{q}_{34}$. Now we derive the relative Jacobian for motion control of four multi-rotor drones. The idea in this derivation is to combine the dual drone components the same way [24] combined manipulator components as dual arms. An effective individual drone relative Jacobian is applied for each dual-drone pair, this means that now we will have the four drone controller consisting of two drones expressed at the relative reference frames. These two drones in the relative reference frames can now then be combined again and be like a dual-arm robot with an individual manipulator controller. This will put the four drones in a holistic single-drone control through dual-drone pairing. With this implementation, the drones can be holistically controlled two at a time. The relative Jacobian gives us the ability to control different combinations simultaneously. Some of the possible combinations are shown in Eqs. (20) to (24). For dual drones 1 and 2 the relative Jacobian control equation is

$${}^1_4J_2 = \left[-{}^1\Omega_0 {}^1\Psi_2 J_1 \;\; {}^1\Omega_0 J_2 \;\; \mathbf{0} \;\; \mathbf{0}\right], \tag{20}$$

where ${}^i\Psi_j$ and ${}^i\Omega_j$ are as explained in (2). The relative Jacobian for dual drones 3 and 4 is

$${}^3_4J_4 = \left[\mathbf{0} \;\; \mathbf{0} \;\; -{}^3\Omega_0 {}^3\Psi_4 J_3 \;\; {}^3\Omega_0 \, J_4\right], \tag{21}$$

the relative Jacobian for dual drones 1 and 3 is

$${}^1_4J_3 = \left[-{}^1\Omega_0 {}^1\Psi_3 J_1 \;\; \mathbf{0} \;\; {}^1\Omega_0 J_3 \;\; \mathbf{0}\right], \tag{22}$$

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

and the relative Jacobian for dual drones 1 and 4 is

$$ {}_{4}^{1}J_4 = \left[-{}^{1}\Omega_0 {}^{1}\Psi_4 J_1 \ \ \mathbf{0}\ \mathbf{0} \ \ {}^{1}\Omega_0 J_4 \right]. \tag{23} $$

The relative Jacobians, in this case, have four columns corresponding to the four drones, such that column 1 corresponds to drone 1, column 2 corresponds to drone 2, etc. A zero column indicates that its corresponding drone lies in the null space. The velocity controller is used for this derivation, and the velocity control equation is

$$ \begin{aligned} \dot{q}_{1234} = {}_{4}^{1}J_2^{+}\ \Lambda_{R12}\ {}^{1}u_{R2} + (I - {}_{4}^{1}J_2^{+}\ {}_{4}^{1}J_2)\ {}_{4}J_1^{+}\ {}^{w}\Omega_1\ I_m\ u_1\ + \ldots \\ \cdots + {}_{4}^{3}J_4^{+}\ \Lambda_{R34}\ {}^{3}u_{R4} + (I - {}_{4}^{3}J_4^{+}\ {}_{4}^{3}J_4)\ {}_{4}J_3^{+}\ {}^{w}\Omega_3\ I_m\ u_3 \end{aligned} \tag{24} $$

where ${}_{4}J_1 = \left[J_1\ \mathbf{0}\ \mathbf{0}\ \mathbf{0} \right]$ and ${}_{4}J_3 = \left[\mathbf{0}\ \mathbf{0}\ J_3\ \mathbf{0} \right]$.

4. Experimental setup

This is a simulation-based research project, we ran it on an open-source operating system software Linux Ubuntu 18.04. It runs from the desktop to the cloud and all your Internet-connected things. Below is the discussion of the developing tools that are required in setting up our environment.

4.1. Robot operating system

Robot operating system (ROS) is an open-source software designed to meet specific challenges robot developers encounter when developing large-scale service robots [29, 30]. ROS has low-cost sensors that bring unquestionably advantages to the robot developers, it gives access to a wide range of robotic applications for any user or developer. It has led to rigorous progressive advances in the development of robotic solutions and has thus become more widespread among the robotics society [31].

ROS consists of several important parts [32]:

1. A set of drivers that reads data from sensors of the simulated robot and sends commands to motors and actuators.
2. A collection of robotic algorithms that can be used to build maps of the world, navigate around the built world, represent and interpret sensor data, manipulate objects, plan motions, etc.
3. Computational infrastructure that can move data around connecting different components of a complex robot system and allows the developer or researcher to incorporate their algorithms.
4. A set of tools like Robot state visualization tools, and tools that make it easy to visualize the algorithms, debug errors, and record sensor data.
5. Extensive set of resources like wiki which documents many aspects of ROS.

4.2. Gazebo simulator

Gazebo simulator is the product of the University of Southern California, which was developed in 2002. The sole purpose of Gazebo was to make a high-fidelity simulator that makes it possible to simulate robots in outdoor environments under various conditions same as in the real world. Gazebo simulator can test the high-performance physics open dynamics engine and the robot sensors for prototype development activities [33]. Gazebo works with ROS together making a powerful combination with the ability to simulate with credibility, flexibility, robustness, and capability of supporting multiple robots performing complex tasks in various distributed environments [34]. Gazebo has various examples of robotic simulations that are easy to find using the models supplied by Gazebo manufacturers. Gazebo simulates three-dimensional dynamics for mechanisms of single and multiple robots, for both inside and outside environments. Even though Gazebo was created to relieve robot developers from the realistic

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

Table I. *Individual drones' initial positions.*

Firefly 1	Firefly 2	Firefly 3	Firefly 4
[0, 0, 0]	[0, 2, 0]	[1, 0, 0]	[1, 2, 0]

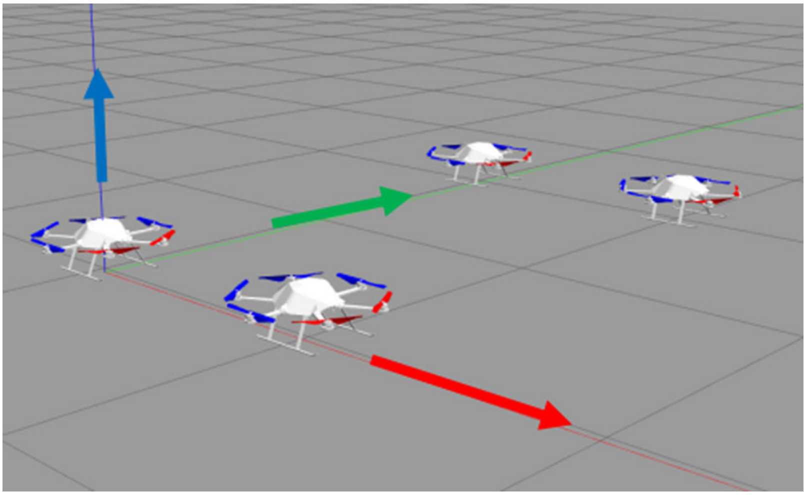

Figure 4. *Drones' initial start positions for our experiments.*

robot simulation in outdoor environments, they mostly use it for indoor simulations. This simulator creates realistic worlds for the robots, and it relies entirely on physics-based characteristics, reflecting the physics when the robot model is pulled, knocked, pushed, etc. It is also an open-source software.

4.3. RotorS simulation framework

RotorS simulation framework is a Gazebo-based micro aerial vehicle platform developed to reduce field testing times and to separate problems of testing, thus making it easier to debug [35]. It was developed by Autonomous Systems lab at ETH Zurich, Switzerland. The simulator has simulated sensors such as an IMU, a generic odometry sensor, and the VI-Sensor, which can be mounted on the simulated drone. The multi-rotor models provided by rotorS include AscTec Hummingbird, AscTec Pelican, and the AscTec Firefly; however, the simulator is not limited to these models only. This paper uses the firefly, which is a hexa-rotor drone. The package comes with example controllers that lay the foundation for researchers and developers to build their controllers from.

5. Implementation results

This section lays out the results of our implementation, and the experiment is divided into two cases in which the first case shows results of individually controlled drones and the second case showing the results of cooperative control of four drones in pairs of two. For both the experiments, the drones' initial positions $[x, y, z]$ are as shown in Table I. These positions are as visualized in Fig. 4

5.1. Case 1: Individual Jacobian control results

Each drone is controlled by its absolute Jacobian J_1, J_2, J_3, and J_4, respectively. The drones do not communicate or depend on other drones rather they exist as stand-alone, any drone can be given any desired

Table II. Individual drones' desired final positions.

Firefly 1	Firefly 2	Firefly 3	Firefly 4
[0, 3, 1]	[−2, 6, 1.5]	[2, −4, 2]	[0.5, 5, 2.2]

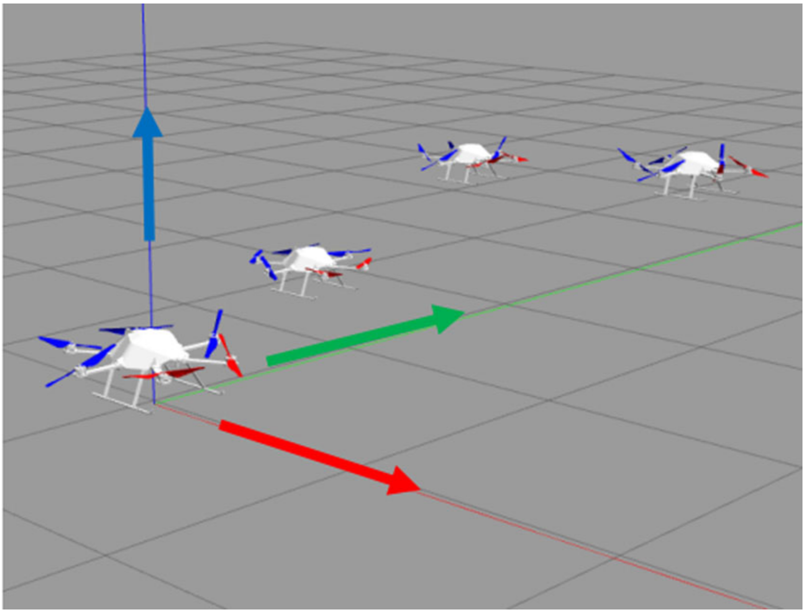

Figure 5. *Individual Jacobian-controlled drones maneuvering in 3D space to their own assigned desired final positions, the drone does not tilt heuristics during this maneuver.*

final position to fly to without taking into consideration the positions of other drones. Each drone's rotor velocities are computed in (6), and each is controlled using the world frame, $\{0\}$, as their reference individually. For this experiment, the desired final positions are as projected in Table II.

The results of this experiment show each drone moving in 3D space to its desired final position in a quintic path without tilting heuristics. Figures 5 and 6 show the demonstration of absolute Jacobian-controlled drones in 3D space, with Fig. 6 showing when they have reached respective desired positions. Graphs shown in Fig. 7 are the feedback from the drone's odometry sensors showing the positions of each drone when moving to their respective desired final positions, and these results show the accuracy of this controller having only small discrepancies which may be the because of the world factors like wind.

5.2. Case 2: Results of cooperative control of four drones

The implementation of the relative Jacobian cooperative controller on four drones is inspired by the implementation of this controller on a four-arm robot [24]. The study aimed at developing a holistic controller for controlling the four-legged animal. The attempt was to interpret the biological method of controlling two paired legs in adjacent and opposite pairs, and they used robot arms to represent the four legs. In our experiment, we used firefly drones instead of robot arms. The purpose of this simulation is to show the overall coordination of the four drones controlled under one individual end-effector drone controller. It is the extension of the concept of individual control where only drone 1 and drone 3 use the world frame as their reference and hence calling them the base drones. And drone 2 uses drone 1 as its reference and drone 4 using drone 3 as its reference hence calling them end-effector drones. Even though

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

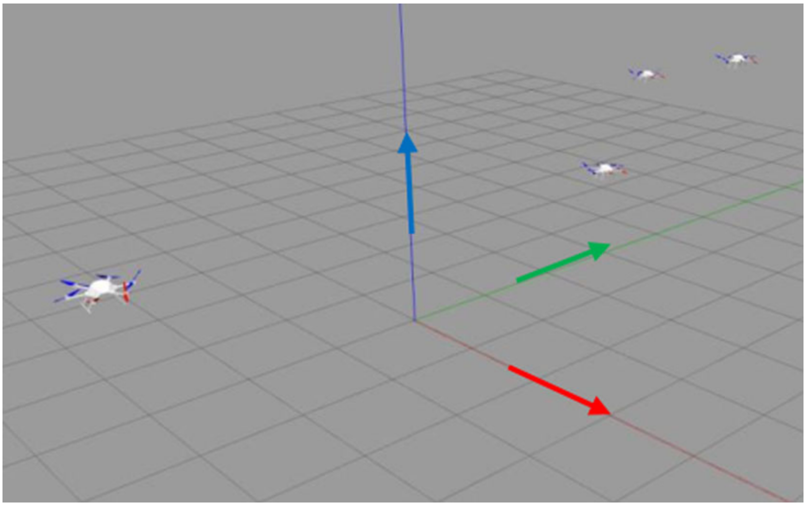

Figure 6. *Individual Jacobian-controlled drones hoovering at their respective desired final positions.*

this implementation was first inspired by the biological limbs of a four-legged animal, this simulation does not mimic the walking of a live animal but uses the principle of coordination between adjacent, opposite, and diagonal pairs. Like in the four-legged animal, there is relative motion among the drones also the concept of controlling the drones is inspired by the way the animal minds control the four legs holistically. The drones are all controlled as if only one drone was being controlled.

For this Case 2, we control the drone positions by using the base drones drone 1 and drone 3. Table III shows the base drones' set desired final positions used for this experiment. The relative positions of drone 2 and drone 4, the end-effector drones with respect to the base drones, have also been set. When the base drones move, the end-effector drones will follow accordingly, thus moving with respect to the respective base drone because the desired relative position is maintained; thus remains the same. This is as shown in Fig. 8.

Figures 9 and 10 show the odometry position comparisons with the desired positions of both pairs. These graphs show the positions of these drones from the initial positions stated in Table I to the desired positions stated in Table III. Note that here the positions of firefly 2 and firefly 4 are the desired relative positions of the end-effector drones, we only change the desired final positions for the base drones only, that is, drone 1 and drone 2 (firefly 1 and firefly 3) and the end-effector drones will follow maintaining the stated desired relative positions.

The general response on positional error of cooperative drones maneuvering from initial position to desired final position is shown in Fig. 11; it depicts the average performance of the relative Jacobian cooperative-controlled drones. The graph shows the error reduction as the drone is maneuvering to its desired final position until the error becomes zero. There is also minimal change in orientation of the drone. It demonstrates the fact that given the right combination of drone configurations, the drones are able to maneuver with less restrictions.

6. Potential applications

Jacobian-based control and cooperative-controlled multi-rotor drones can move precisely because of no tilting heuristics and hence can suit many applications that require precise movements. With the use of good machine learning techniques, relative Jacobian-controlled team of drones can make good surveillance because the drones are stable; hence, the camera technology will give accurate results.

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

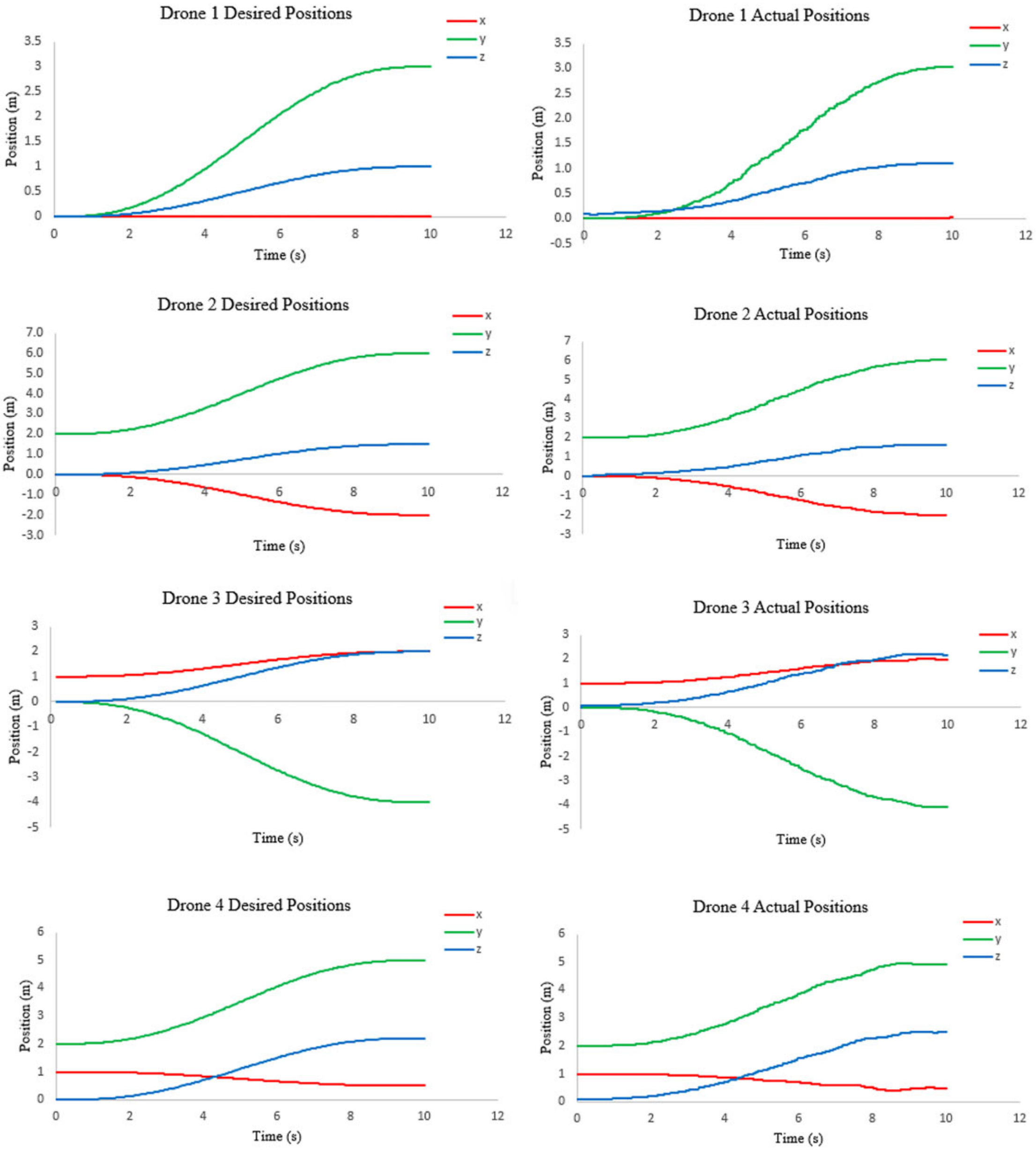

Figure 7. *Odometry position feedback of each drone with the desired positions.*

6.1. Unexploded ordnance disposal

During military training and operations, the use of antitank weapons like grenade launchers, for example, Rocket Propelled Grenade 7 (RPG 7), 84 mm Carl Gustav, etc., is highly used. In some most unfortunate situations, these weapons can have horrible misfired shells or unexploded ordnances. Ammunition is a very complex mechanism that consists of many dependent and interrelated functions; in the event of unexploded ordnance, the fuzing system is most probably the primary point of failure [36]. This however poses safety hazards to the environment [37], and therefore, the need for misfires and unexploded ordnances to be disposed. They can also explode during the disposal period and probably causing harm to humanity. So, multi-rotor drones with attached robot arms or sanction pads can be an alternative way of

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

Table III. *Cooperative-controlled drone pairs desired final positions.*

Firefly 1	Firefly 2	Firefly 3	Firefly 4
[0, 0, 1]	[0, 2, 1]	[1, 0, 1]	[1, 2, 1]

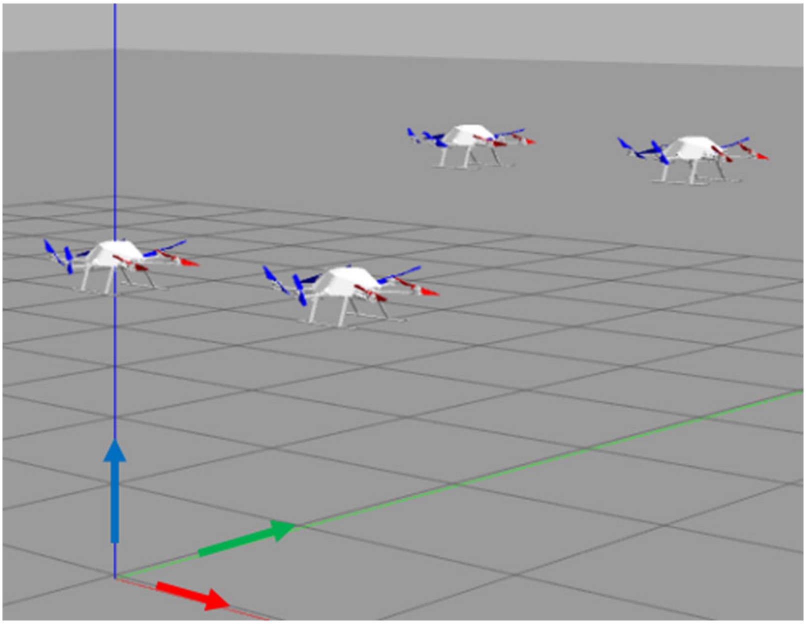

Figure 8. *Cooperative drones hovering at their desired final positions.*

disposing of this ammunition to hinder the harm that can occur to humanity during disposal. Here now we will need a drone that will be mostly stable because disturbances to the unexploded ordnance may cause it to explode. Jacobian-controlled multi-rotor drones move in 3D space without tilting of heuristics this will be the best controller to use for the application of misfired shells or unexploded ordnances because of high stability during maneuvering.

6.2. Patrols and videography

As videography is one of the most common application of multi-rotor drones, there are a lot of applications of multi-rotor drones based on videography. This is because videography is relatively not expensive to conduct due to the low costs of viewing devices like cameras and low costs for operator training [38]. Sometimes, there is a need to record steady and more concentrated videos; this can be useful in filmmaking or patrols performed by the military or police. This will require a more stable drone, and if there is a need to capture a large area, relative Jacobian-based cooperative multi-rotor drones can be used. Cooperative drones controlled with this controller can be an effective supplement to the current patrolling business in road traffic patrolling with complex urban buildings and road conditions and a limited ground perspective. One of the problems that drones on patrol is having the patrolling cooperative drones visit and revisit the target area to cover the areas that may have been missed [39, 40]. Relative Jacobian-based cooperative control is the solution to this problem because the drones are more stable than other drone controllers and hence will not miss areas to be covered. The captured videos can be merged, and it will be as if only one large coverage camera was used. This shows the good outcomes of the relative Jacobian-based cooperative controller.

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

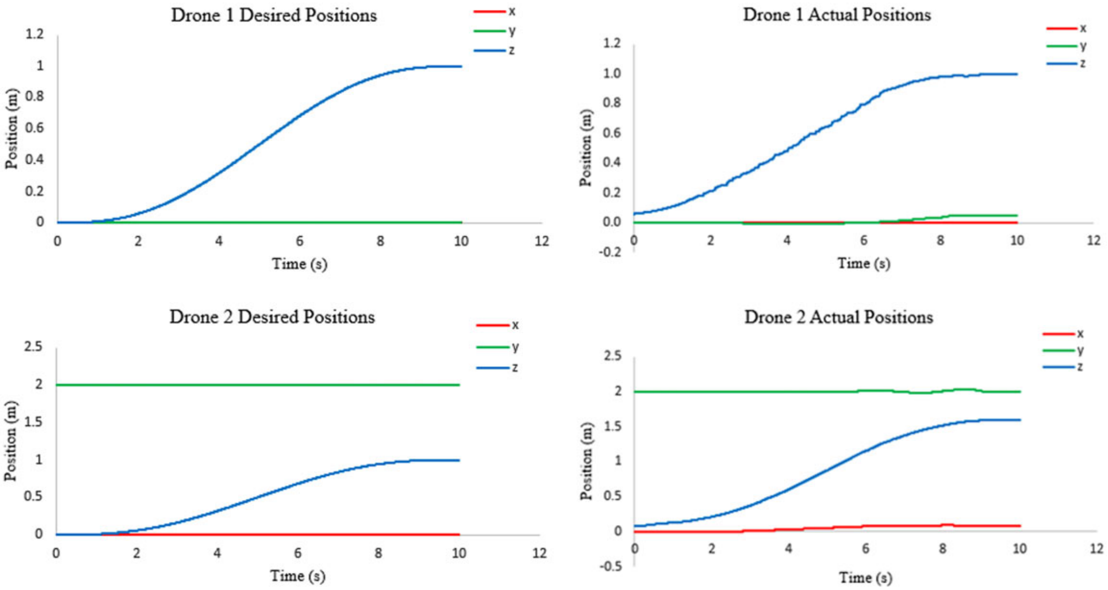

Figure 9. *Pair 1, drone 1 and drone 2 odometry position feedback with their desired positions.*

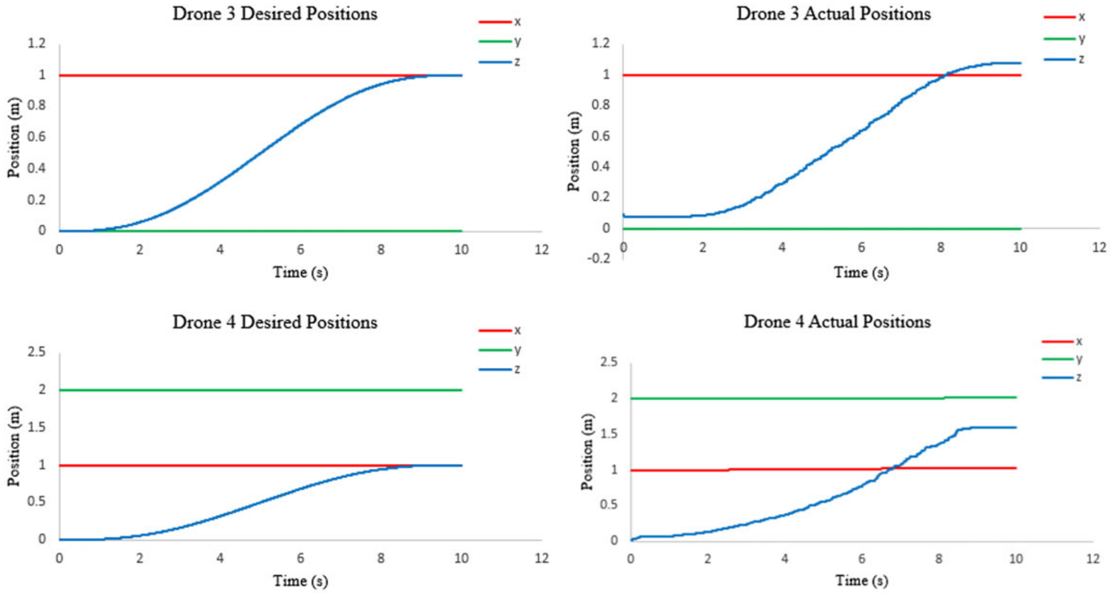

Figure 10. *Pair 2, drone 3 and drone 4 odometry position feedback with their desired positions.*

7. Conclusions and future work

This study showed implementation results of individual Jacobian control and relative Jacobian-based cooperative motion control of AscTec firefly hexa-rotor drones using RotorS Gazebo platform. A new formulation is presented that allows Jacobian-based control of individual drone and relative Jacobian-based control of a pair of drones in terms of relative position and orientation. Lastly, a new formulation is presented that allows relative Jacobian-based control of four drones, using paired relative position and orientation control. However, in these implementation relative orientation remained constant and only

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

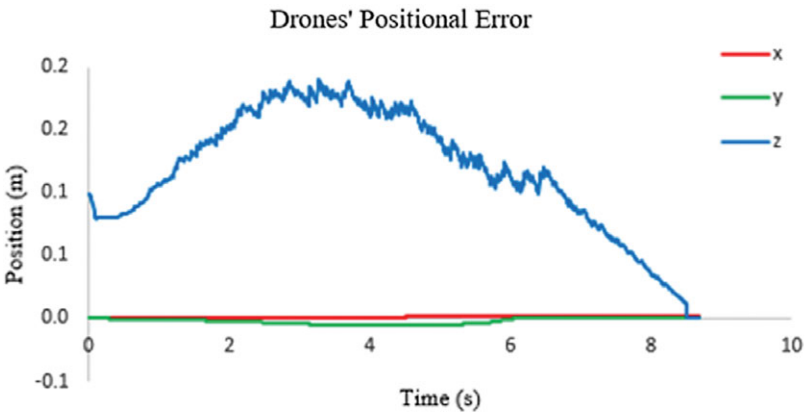

Figure 11. *Performance of the base drone among the relative Jacobian cooperatively controlled drones.*

relative position control is changed. In future work, this controller will be implemented on real hardware. Work is in progress to build own DIY drones to implement this controller on. Another platform to implement this controller on is the DJI M200 v2 series and DJI Matrice 300 drones. Also as part of future work, both position and orientation control will be implemented on both individual and four cooperative drones.

Author contributions. Keletso Z. Thebe is the main researcher conducting this research having Larona P. Ramalepa, his fellow colleague assisting with the mathematical equations implementation. They did data gathering and conducted experiments together and writing of the article. Rodrigo S. Jamisola Jr. is the principal investigator of the research project, and he derived all the equations and also assisted in structuring the writing of the research article.

Financial support. This research is funded by Botswana International University of Science and Technology (BIUST) Drones Project with project number P00015.

Competing interests. The authors declare that they have no competing interest.

Ethical approval. Not applicable.

Acknowledgments. The authors would like to acknowledge Olebogeng Mbedzi for the assistance in ROS and Gazebo platform setup and support throughout the implementation. Also, the authors thank the funding support on this work from the Botswana International University of Science and Technology (BIUST) drones project number P00015.

References

[1] L. Yu, G. He, S. Zhao, Z. Wang and L. Shen, "Immersion and invariance-based sliding mode attitude control of tilt tri-rotor UAV in helicopter mode," *Int. J. Control Autom. Syst.* **19**(2), 722–735 (2021).

[2] L. P. Ramalepa and R. S. Jamisola, "A review on cooperative robotic arms with mobile or drones bases," *Int. J. Autom. Comput.* **18**(4), 536–555 (2021).

[3] T. Elijah, R. S. Jamisola, Z. Tjiparuro and M. Namoshe, "A review on control and maneuvering of cooperative fixed-wing drones," *Int. J. Dyn. Control* **9**(3), 1332–1349 (2021).

[4] S. Kaviri, A. Tahsiri and H. D. Taghirad, "A cooperative control framework of multiple unmanned aerial vehicles for dynamic oil spill cleanup," *J. Braz. Soc. Mech. Sci. Eng.* **43**(6), 1–17 (2021).

[5] J. Hu and A. Lanzon. Cooperative Control of Innovative Tri-Rotor Drones Using Robust Feedback Linearization. **In:** *2018 UKACC 12th International Conference on Control (CONTROL)*, Sheffield, UK (2018) pp. 347–352.

[6] P. Zhang, Y. Fang, X. Liang, H. Lin and W. He, "Nonlinear cooperative control of double drone-bar transportation system," *arXiv preprint* arXiv:2011.07515, (2020).

[7] E. Skjervold. Autonomous, Cooperative UAV Operations Using COTS Consumer Drones and Custom Ground Control Station. **In:** *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA (2018) pp. 1–6.

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press

[8] G. Antonelli, “Development of Efficient Swarm Intelligence Algorithm for Simulating Two-Dimensional Orthomosaic for Terrain Mapping Using Cooperative Unmanned Aerial Vehicles,” **In:** *The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems*, (Elsevier, 2020) pp. 75–93.

[9] F. A. de Alcantara Andrade, A. R. Hovenburg, L. N. de Lima, C. D. Rodin, T. A. Johansen, R. Storvold and D. B. Haddad, “Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control,” *Sensors* **19**(19), 4067 (2019).

[10] W. Youn, H. Ko, H. Choi, I. Choi, J. H. Baek and H. Myung, “Collision-free autonomous navigation of a small UAV using low-cost sensors in GPS-denied environments,” *Int. J. Control Autom. Syst.* **19**(2), 953–968 (2021).

[11] Y. Liu, H. Liu, Y. Tian and C. Sun, “Reinforcement learning based two-level control framework of UAV swarm for cooperative persistent surveillance in an unknown urban area,” *Aerosp. Sci. Technol.* **98**, 105671 (2020).

[12] W. Yuan, Q. Chen, Z. Hou and Y. Li. Multi-UAVs Formation Flight Control Based on Leader-Follower Pattern. **In:** *IEEE 36th Chinese Control Conference (CCC)*, Dalian, China (2017) pp.1276–1281.

[13] Y. Qu, X. Zhu and Y. M. Zhang. Cooperative Control for UAV Formation Flight Based on Decentralized Consensus Algorithm. **In:** *International Conference on Intelligent Robotics and Applications*, Montreal, QC (2012) pp. 357–366.

[14] A. Abhishek, A. Duhoon, M. Kothari, S. Kadukar, L. Rane and G. Suryavanshi. Design, Development, and Closed-Loop Flight-Testing of a Single Power Plant Variable Pitch Quadrotor Unmanned Air Vehicle. **In:** *Proceedings of the 73rd American Helicopter Society Annual Forum, Vertical Flight Society (VFS), Fairfax, Virginia. U.S*, (2017) pp. 205–218.

[15] R. Kumar, A. Nemati, M. Kumar, R. Sharma, K. Cohen and F. Cazaurang, “Tilting-Rotor Quadcopter for Aggressive Flight Maneuvers Using Differential Flatness Based Flight Controller,” **In:** *Dynamic Systems and Control Conference*, vol. 58295 (American Society of Mechanical Engineers, 2017) pp. V003T39A006.

[16] UAV Coach, How to fly a drone: A beginner’s guide to multirotor systems, Available at https://uavcoach.com/how-to-fly-a-quad-copter-guide/ (October 05, 2021).

[17] A. M. Deshpande, R. Kumar, A. A. Minai and M. Kumar, “Developmental Reinforcement Learning of Control Policy of a Quadcopter UAV with Thrust Vectoring Rotors,” **In:** *Dynamic Systems and Control Conference*, vol. 84287 (American Society of Mechanical Engineers, 2020) pp. V002T36A011.

[18] S. O. Park, M. C. Lee and J. Kim, “Trajectory planning with collision avoidance for redundant robots using Jacobian and artificial potential field-based real-time inverse kinematics,” *Int. J. Control. Autom. Syst.* **18**(8), 2095–2107 (2020).

[19] K. J. Waldron, S. L. Wang and S. J. Bolin, A study of the Jacobian matrix of serial manipulators, (1985).

[20] X. M. Niu, G. Q. Gao, X. J. Liu and Z. D. Bao, “Dynamics and control of a novel 3-DOF parallel manipulator with actuation redundancy,” *Int. J. Autom. Comput.* **10**(6), 552–562 (2013).

[21] R. S. Jamisola and R. G. Roberts, “A more compact expression of relative Jacobian based on individual manipulator Jacobians,” *Robot. Auton. Syst.* **63**, 158–164 (2015).

[22] R. S. Jamisola, O. Mbedzi, T. Makati and R. G. Roberts. Investigating Modular Relative Jacobian Control for Bipedal Robot. **In:** *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Bangkok, Thailand (2019) pp. 428–432.

[23] R. S. Jamisola Jr, C. Mastalli and F. Ibikunle, “Modular relative jacobian for combined 3-arm parallel manipulators,” *Int. J. Mech. Eng. Robot. Res.* **5**(2), 90–95 (2016).

[24] R. S. Jamisola and C. Mastalli. Bio-Inspired Holistic Control Through Modular Relative Jacobian for Combined Four-Arm Robots. **In:** *2017 18th International Conference on Advanced Robotics (ICAR)*, Hong Kong, China (2017) pp. 346–352.

[25] M. Ghafarian, B. Shirinzadeh, A. Al-Jodah, T. K. Das, W. Wei and T. Shen, “Modeling and prototype experiment of a monolithic 3-PUU parallel micromanipulator with nano-scale accuracy,” *Smart Mater. Struct.* **29**(7), 075023 (2020).

[26] J. Maaß, N. Kohn and J. Hesselbach, “Open modular robot control architecture for assembly using the task frame formalism,” *Int. J. Adv. Robot. Syst.* **3**(1), 2 (2006).

[27] P. Zheng, X. Tan, B. B. Kocer, E. Yang and M. Kovac, “TiltDrone: A fully-actuated tilting quadrotor platform,” *IEEE Robot. Autom. Lett.* **5**(4), 6845–6852 (2020).

[28] R. Featherstone and O. Khatib, “Load independence of the dynamically consistent inverse of the Jacobian matrix,” *Int. J. Robot. Res.* **16**(2), 168–170 (1997).

[29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. B. Ng, “ROS: An open-source robot operating system,” *ICRA Workshop Open Source Softw.* **3**(2), 5 (2009).

[30] M. L. Aarizou and N. E. Berrached, “ROS-based telerobotic application for transmitting high-bandwidth kinematic data over a limited network,” *Int. J. Control Autom. Syst.* **17**(2), 445–453 (2019).

[31] A. Koubâa, eds. *Robot Operating System (ROS)*, vol. 1 (Springer, Cham, 2017).

[32] M. Quigley, B. Gerkey and W. D. Smart. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System* (O’Reilly Media, Inc., 2015).

[33] Z. B. Rivera, M. C. De Simone and D. Guida, “Unmanned ground vehicle modelling in Gazebo/ROS-based environments,” *Machines* **7**(2), 42 (2019).

[34] Y. Zhai, L. Liu, W. Lu, Y. Li, S. Yang and F. Villecco, “The Application of Disturbance Observer to Propulsion Control of Sub-Mini Underwater Robot,” **In:** *International Conference on Computational Science and Its Applications*, (Springer, 2010) pp. 590–598.

[35] F. Furrer, M. Burri, M. Achtelik and R. Siegwart, “Rotors–A Modular Gazebo MAV Simulator Framework,” **In:** *Robot Operating System (ROS)*, (Springer, 2016) pp. 595–625.

[36] D. S. Williams, “The management of unexploded ordnance,” *Aust. Def. Force J.* **119**, 43–50 (1996).

[37] S. D. Billings, “Discrimination and classification of buried unexploded ordnance using magnetometry,” *IEEE Trans. Geosci. Remote Sens.* **42**(6), 1241–1251 (2004).

[38] S. E. Smith, L. Pearlstine and J. Stenberg, "Comparison of aerial videography and digital camera imagery for improving land cover classification," *Surv. Land Inf. Sci.* **62**(3), 163 (2002).

[39] K. S. Kappel, T. M. Cabreira, J. L. Marins, L. B. de Brisolara and P. R. Ferreira, "Strategies for patrolling missions with multiple UAVs," *J. Intell. Robot. Syst.* **99**(3), 499–515 (2020).

[40] D. Klaška, A. Kučera and V. Řehák. Adversarial Patrolling With Drones. **In:** *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, Auckland, New Zealand (2020) pp. 629–637.

Cite this article: K. Z. Thebe, R. S. Jamisola Jr. and L. P. Ramalepa (2023). "A novel approach to control four multi-rotor drones in cooperative paired control using relative Jacobian", Robotica **41**, 3004–3021. https://doi.org/10.1017/S0263574723000826

https://doi.org/10.1017/S0263574723000826 Published online by Cambridge University Press