

Research

## Autonomous landing of a quadrotor on a moving platform using motion capture system

Ayman Qassab<sup>1</sup> · Muhammad Umer Khan<sup>2</sup> · Bulent Irfanoglu<sup>3</sup>

Received: 25 January 2024 / Accepted: 23 May 2024

Published online: 02 June 2024

© The Author(s) 2024 [OPEN](#)

### Abstract

This paper investigates the challenging problem of the autonomous landing of a quadrotor on a moving platform in a non-cooperative environment. The limited sensing ability of quadrotors often hampers their utilization for autonomous landing, especially in GPS-denied areas. The performance of motion capture systems (MCSs) in many application areas is the motivation to utilize them for the autonomous take-off and landing of the quadrotor in this research. An autonomous closed-loop vision-based navigation, tracking, and control system is proposed for quadrotors to perform landing based upon Model Predictive Control (MPC) by utilizing multi-objective functions. The entire process is posed as a constrained tracking problem to minimize energy consumption and ensure smooth maneuvers. The proposed approach is fully autonomous from take-off to landing; whereas, the movements of the landing platform are pre-defined but still unknown to the quadrotor. The landing performance of the quadrotor is tested and evaluated for three different movement patterns: static, square-shaped, and circular-shaped. Through experimental results, the pose error between the quadrotor and the platform is measured and found to be less than 30 cm.

### Article Highlights

1. Introducing a holistic vision system for quadrotor navigation, tracking, and landing on stationary/moving platforms.
2. Proposing an energy-efficient, smooth, and stable MPC controller validated by Lyapunov analysis.
3. Validating the adept tracking and safe landings of the quadrotor on stationary/moving platforms through three diverse experiments.

**Keywords** Unmanned aerial vehicle · Autonomous vehicle · Localization · Motion capture system · Model predictive control

## 1 Introduction

Quadrotors that belong to the wide class of Unmanned Aerial Vehicles (UAVs) have been undergoing surprising adaptations and improvements, mainly due to their exceptional maneuverability and robust hovering ability. For the past couple of decades, quadrotors have been extensively employed in many application areas, including but not limited

---

✉ Muhammad Umer Khan, [umer.khan@atilim.edu.tr](mailto:umer.khan@atilim.edu.tr); Ayman Qassab, [ayman.qassab@pusulacc.com.tr](mailto:ayman.qassab@pusulacc.com.tr); Bulent Irfanoglu, [bulentirfanoglu@baskent.edu.tr](mailto:bulentirfanoglu@baskent.edu.tr) | <sup>1</sup>Pusula, 06420 Ankara, Turkey. <sup>2</sup>Department of Mechatronics Engineering, Atilim University, 06830 Ankara, Turkey. <sup>3</sup>Department of Electrical and Electronics Engineering, Baskent University, 06790 Ankara, Turkey.



to construction [1], cargo [2], agriculture [3], rescue operations [4], crowd monitoring [5], and surveillance [6]. Recent advancements in techniques and control architectures have significantly bolstered UAVs, furnishing them with a robust foundation for expansion [7]. The strides made in sensor technology and other critical domains of artificial intelligence serve as a catalyst for amplifying their autonomy [8]. To elaborate, the concept of autonomy implies that quadrotors can perform key operations such as flying, hovering, and landing on their own. Among these, autonomous landing is considered the most challenging flight task, mainly due to the environmental conditions and nonlinear dynamics of quadrotors [9, 10].

The major challenges faced in autonomous landing are: accurate localization of the landing platform as well as the quadrotor; state estimation of the platform; and safe landing on a stationary or dynamic platform. A large portion of the existing literature considers the platform to be stationary, whose position can be known or unknown [11–13]. The indoor localization is achieved mainly using computer vision [13, 14], ultra-wideband [15], IMU [16], multimodal sensor fusion [17, 18], wireless sensor networks [19], and motion capture systems [20–22]. To identify the platform using computer vision, certain patterns or markers are placed on the top of the platform [14, 23–25]. The MCSs, independent of such requirements, have proved to be more robust and accurate than other existing approaches [22, 26]. In this work, we assume no communication or perception exists between the quadrotor and the platform; i.e., the environment is non-cooperative. Our study endeavors to demonstrate the autonomous landing proficiency of a quadrotor, leveraging OptiTrack cameras within a complex, dynamic, and unfamiliar setting.

Motion capture systems were initially developed for gait analysis in health care, but are now used in various other fields, such as biomechanics, robotics, sports, virtual reality, and entertainment, among others. Depending upon the nature of the application, the number of cameras can be increased to enhance the captured volume, thus making the system much more expensive. The notable manufacturers producing MCSs are Nokov [27], Vicon [28], Qualisys [29], and OptiTrack [30]. Motion capture is a means to record the movements of specific objects or individual by determining their pose in the physical space. MCSs are mainly classified into four categories based upon the sensor measurements: optical-passive, optical-active, video, and inertial [31]. Among these, optical-passive MCSs have gained recognition for robotics applications in an indoor laboratory setting due to their significant practical value. The author in [22] used the OptiTrack system to localize a mobile robot and compared the results with odometry and kinematic localization. The results demonstrated that higher accuracy was obtained through MCS. In [26], the authors investigated the performance and accuracy of a Vicon motion capture device to record and assess the static and dynamic movements of a mobile robot.

Before the advent of MCS, computer vision and pattern matching were used as the main tools to assist in the autonomous landing of UAVs. Many vision-based algorithms have been developed for this purpose which can be broadly classified into position-based visual servoing [32, 33] and image-based visual servoing [34, 35]. Usually, these algorithms require certain patterns and markers to be affixed to the landing site. In [36], the authors presented a real-time algorithm that identifies an H-shaped landing target using invariant moments. A similar work was proposed by [37], which used a single cross at the center of a circle as a landing pattern. Niu et al. [38] proposed the usage of multiple-scale QR codes for different altitudes. For localization, the empirical formulation is defined which corresponds to the pixel size and UAV's altitude. The authors in [39] also utilized a QR code-based visual positioning system to localize the target object - in this case, a flying drone. In [25], the authors developed a vision-based system capable of detecting fiducial ArUcO markers placed on the landing platform through a monocular camera attached to the quadrotor. The authors in [32] combined large- and small-sized fiducial AprilTag markers to locate the target object with high precision.

Once the localization of the quadrotor and the landing platform is achieved, the motion control of the quadrotor is the next big challenge. The under-actuated nature of the quadrotor makes the takeoff, hovering, tracking, and landing much more difficult. The overall control scheme is usually implemented through a multi-loop to control the position and attitude independently [33]. In a cascaded control designed by [40], MPC was used to plan the landing trajectory; whereas to track the planned trajectory, Incremental Nonlinear Dynamic Inversion (INDI) was employed. The authors of [41] performed the trajectory tracking using linear MPC, and the method was validated in the simulated environment. The MPC-based approach was also utilized by [37] to perform autonomous tracking and landing on a moving car.

The present work proposes a complete autonomous vision-based navigation and control system to perform the autonomous tracking of a stationary or moving platform and the landing of a quadrotor on it. A real-time optical MCS is utilized to track and calculate the poses of both the quadrotor and the landing platform. In this way, using MCS, the necessary condition to keep the target object's geometric properties constantly visible to the quadrotor is eliminated. The developed control architecture guides the quadrotor to the platform by estimating the future movements of the platform. This work proposes a cascaded control scheme, in which MPC is used to perform position tracking while PID is used to control the attitude. This approach follows the fact that, nowadays, a majority of

commercially available flight controllers are pre-equipped with attitude controllers. The experimental results show that the proposed approach can effectively work for both stationary and moving landing platforms and can follow a ground vehicle with a linear velocity of 0.8 m/s. The novelty of this approach is in the introduction of the state-of-the-art MCS to accomplish the challenging quadrotor landing problem in real-world experimental conditions.

The valuable contributions of this work are as follows:

1. An autonomous vision-based navigation, tracking, and control system is proposed that enables the localization, estimation, tracking, and landing of a quadrotor on a stationary or moving platform by employing a real-time MCS.
2. In contrast to the vision-based system described in [25], which relies on the quadrotor initiating tracking and detection from a hovering state with the landing platform in its field of view, necessitating constant visual monitoring to keep the platform within sight, this paper introduces a more holistic approach. Our proposed solution elevates the necessity to maintain continuous observation of the target throughout all stages of operation, encompassing rest, take-off, hovering, and landing. Moreover, the quadrotor can be initially placed anywhere in the workspace.
3. While significant strides have been taken in the pursuit of reference tracking for mobile robots, the prevailing literature frequently confines performance evaluation to metrics such as reference tracking error and motion smoothness [14]. This narrow focus imposes limitations on the robustness and overall performance of the system. In this article, we introduce an optimal model-based controller designed to address these shortcomings. Our approach not only minimizes energy consumption but also prioritizes smooth motion and closed-loop stability, effectively managing induced efforts to enhance overall system performance. Additionally, the closed-loop stability of the system is rigorously ensured through Lyapunov Stability analysis, further bolstering its reliability and robustness.

## 2 Problem description

The overall problem can be classified into three parts: localization of the quadrotor and the landing platform, tracking and estimation of the landing platform, and safe landing of the quadrotor on a stationary or moving platform. The 3D reference coordinate frames are attached to the centers of gravity of the quadrotor and the landing platform as  $X_Q = [x_q \ y_q \ z_q]^T$  and  $X_P = [x_p \ y_p \ z_p]^T$ , respectively. The mobile platform's frame of reference is positioned with an offset of 9 cm off the ground. The overall operation is considered successful when the quadrotor manages to land on the target platform. The three problems introduced earlier are defined as follows:

### Problem 1 Localization of the quadrotor and the landing platform

Build up a framework consisting of a quadrotor, platform, and four OptiTrack cameras to localize the ground vehicle and aerial vehicle to obtain  $X_Q$  and  $X_P$  in real-time.

### Problem 2 Tracking and estimation of the landing platform

The quadrotor tracks the landing platform in the horizontal plane  $(x, y)$  and initiates the landing process if

$$\begin{aligned} e_x &:= \|x_q - x_p\| \leq \sigma_x \\ e_y &:= \|y_q - y_p\| \leq \sigma_y \end{aligned} \quad (1)$$

where  $\sigma_x, \sigma_y \in \mathbb{R}^+$  are defined as the tolerance for the position errors of the landing area.

Design an optimal MPC utilizing the estimated position and velocity of the platform obtained through KF and the actual position and velocity of the quadrotor.

$$u_{x,y} := f_1(x_q, y_q, \dot{x}_q, \dot{y}_q, \tilde{x}_p, \tilde{y}_p, \dot{\tilde{x}}_p, \dot{\tilde{y}}_p)$$

### Problem 3 Landing of the quadrotor on the platform

The landing state is triggered once the quadrotor and the landing platform are aligned in the  $x - y$  plane by fulfilling the conditions in Eq. (1). The adjustment in the quadrotor's height continues until

$$e_z := \|z_q - z_p\| \approx 0 \quad (2)$$

Design a PID

$$u_z := f_2(z_q, \dot{z}_q, z_p, 0)$$

to achieve the vertical regulation of the quadrotor. The height of the platform is a constant, hence velocity would be zero. It is noted that both controllers for horizontal and vertical regulation are combined to generate a unified model-based predictive controller.

### 3 System overview

#### 3.1 Quadrotor and landing platform

The DJI Tello EDU, a programmable quadrotor, is used in this study. With the help of SDK, it can be interfaced with many tools and programming languages, such as Python, Scratch, Swift, and MATLAB. It is very light-weight (87 g) and has the dimensions of 9.8 cm  $\times$  9.25 cm  $\times$  4.1 cm. Tello EDU has a speed limit of 8 m/s, a maximum flight time of 13 min, and a maximum flight height of 3000 cm. The landing platform having the dimensions of 30 cm  $\times$  30 cm is placed on top of a differential drive robot. Therefore, its position and velocity can be defined based on the movements of the robot, whose maximum velocity is defined as 0.8 m/s.

#### 3.2 Motion capture system (MCS)

MCS, a very sophisticated system, comprises several components: hardware, software, and acquisition methods.

##### 3.2.1 OptiTrack cameras

Four OptiTrack Prime<sup>x</sup>13's 2.7'' square signatures are considered for this study. They are capable of tracking active and passive markers with the positional and rotational accuracies of  $\pm 0.20$  mm and 0.5 deg, respectively. The detailed camera specifications are provided in Table 1.

##### 3.2.2 System architecture

Four lightweight OptiTrack cameras are connected to the host computer using an Ethernet PoE/PoE+ switch through LAN. A special software tool, known as Motive, running on a PC is responsible for the calibration and acquisition of the 3D data. The software can stream live data to other applications using a plugin or SDK. NAT-NET SDK is employed to build client/server applications and to interface Motive with MATLAB. Using this interface, client applications can run on the same PC as the tracking software (Motive). The host PC is carefully chosen to have an i5-9400F processor with six cores to satisfy the required minimum specifications and run the operations smoothly.

**Table 1** Prime<sup>x</sup>13 camera specifications

Resolution	1.3 MP (1280 $\times$ 1024)
Frame Rate (fps)	240
Horizontal FOV (deg)	56
Vertical FOV (deg)	42
Interface	GigE/PoE
Latency (ms)	4.2

### 3.2.3 Motion capture workspace

Similar to other measurement systems, calibration is also considered critical for optical MCSs. Using the OptiTrack calibration wand, the pose of each camera is computed along with the amount of distortion in the captured images. This information is later used to define a 3D capture volume based on 2D images from multiple synchronized cameras through a process known as triangulation. The cameras should be positioned at a certain vantage point to maximize the target capture area. For our case, four cameras are positioned perpendicular to each other to define a workspace volume of 300 cm  $\times$  300 cm (Fig. 1). After the calibration is performed and the volume is defined, retro-reflective markers are carefully installed on the objects of interest. For this study, four markers are placed on both the quadrotor and the platform in a configuration where they are always observable. During the operation, Motive can record the movements of each marker or live stream them to other pipelines.

## 4 System dynamics and modeling

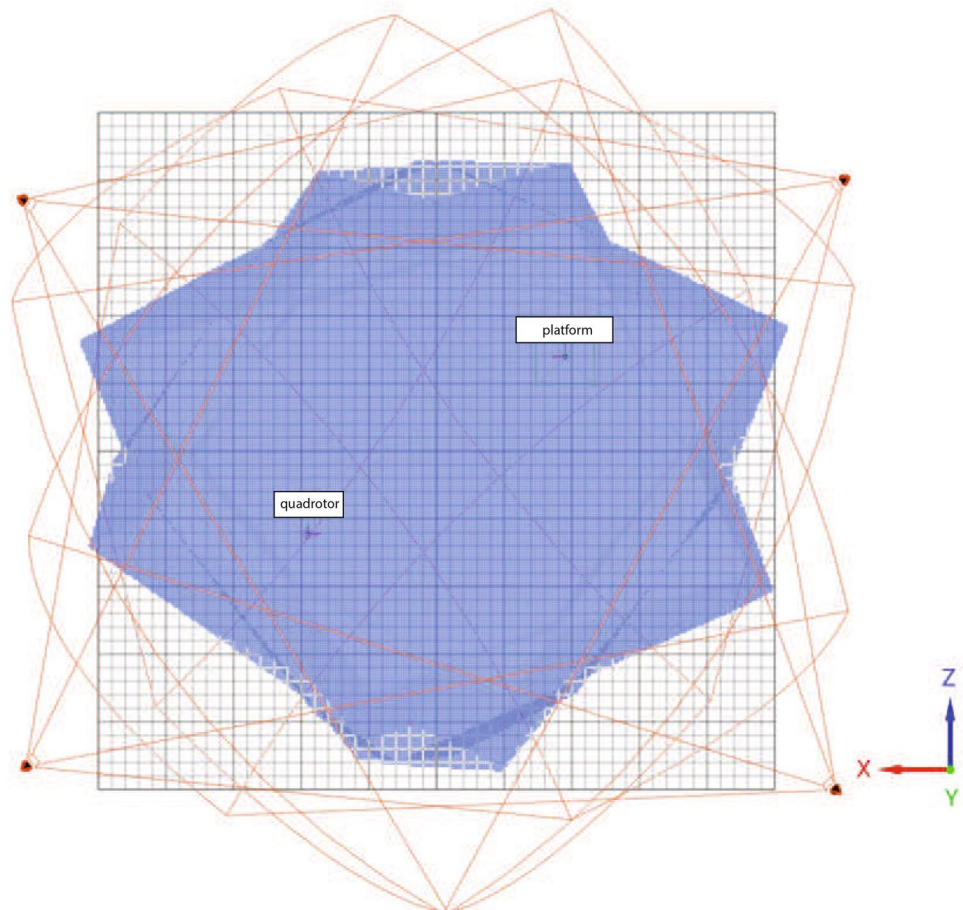
In this section, the kinematic and dynamic model of the quadrotor is presented.

**Assumption 1** The origin and the axes of the body frame coincide with the body center of the quadrotor and the principal axes.

**Assumption 2** The quadrotor is a rigid body and is perfectly balanced.

**Assumption 3** The maximum velocity of the quadrotor is considered greater than the landing platform that is placed on a differential drive robot.

**Fig. 1** Workspace volume defined using MCS



The linear velocity of the quadrotor in the body-fixed frame is defined using  $\mathbf{v}_B = [u \quad v \quad w]^T \in \mathbb{R}^3$ , whereas  $\mathbf{v} = [\dot{x}_q \quad \dot{y}_q \quad \dot{z}_q]^T \in \mathbb{R}^3$  is used to represent its velocity in the inertial frame, which is placed globally. The linear velocities are related by the rotation matrix  $\mathbf{R}(\phi, \theta, \psi) \in \text{SO}(3)$  as:

$$\mathbf{v} = \mathbf{R}\mathbf{v}_B \quad (3)$$

Using the Euler angles, rotation matrix  $\mathbf{R}$  is determined as:

$$\mathbf{R} = \begin{bmatrix} c\psi c\theta & c\theta s\psi & -s\theta \\ c\psi s\phi s\theta - c\phi s\psi & c\phi c\psi + s\phi s\psi s\theta & c\theta s\phi \\ s\phi s\psi + c\phi c\psi s\theta & c\phi s\psi s\theta - c\psi s\phi & c\phi c\theta \end{bmatrix} \quad (4)$$

where  $c$  and  $s$  are abbreviations for the cosine and sine functions, respectively. The angular velocity vector of the quadrotor in the body-fixed frame is defined using  $\boldsymbol{\omega}_B = [p \quad q \quad r]^T \in \mathbb{R}^3$ , whereas  $\boldsymbol{\omega} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T \in \mathbb{R}^3$  is used to represent its angular velocity in the inertial frame. Therefore, the relationship between the angular velocities can be defined by:

$$\boldsymbol{\omega} = \mathbf{T}\boldsymbol{\omega}_B \quad (5)$$

Transformation matrix  $\mathbf{T} \in \text{SO}(3)$  is used to transform the angular velocity from the body frame to the inertial frame:

$$\mathbf{T} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \quad (6)$$

where  $t$  represents the tangent function. The equations of motion are derived using the Newton-Euler method to describe the dynamics of the quadrotor.

$$m(\boldsymbol{\omega}_B \times \mathbf{v}_B + \dot{\mathbf{v}}_B) = \mathbf{f}_B \quad (7)$$

where  $m$  is the mass of the quadrotor and  $\mathbf{f}_B = [f_x \quad f_y \quad f_z]^T \in \mathbb{R}^3$  is the total force. The total torque required for the quadrotor is calculated using the Newton-Euler method:

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\mathbf{I}\boldsymbol{\omega}_B) = \mathbf{m}_B \quad (8)$$

where  $\mathbf{m}_B = [m_x \quad m_y \quad m_z]^T \in \mathbb{R}^3$  is the total torque and  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  is the diagonal inertia matrix that contains  $\{I_{xx}, I_{yy}, I_{zz}\}$ . The dynamic model for the quadrotor defined in Eqs. (7) and (8) remains valid as long as Assumption 1 remains intact. The consideration of the effect of the external forces  $\mathbf{f}_B$  on the quadrotor is combined with Eq. (6) as:

$$\mathbf{f}_B = mg\mathbf{R}^T\hat{\mathbf{e}}_z - f_t\hat{\mathbf{e}}_3 \quad (9)$$

where  $\hat{\mathbf{e}}_z$  and  $\hat{\mathbf{e}}_3$  are the unit vectors along the inertial and the body-fixed  $z$  axes.

The weight  $mg$  introduces the effect of the gravitational force, where  $f_t$  as a total thrust is generated by the rotors. The external moments in the body-fixed frame result in a more detailed version of Eq. (8) as:

$$\mathbf{m}_B = \boldsymbol{\tau}_B - \mathbf{m}_g \quad (10)$$

where  $\boldsymbol{\tau}_B = [\tau_x \quad \tau_y \quad \tau_z]^T \in \mathbb{R}^3$  is the control input torque generated due to the difference in the angular velocities of the rotors. The term  $\mathbf{m}_g$ , known as gyroscopic moment, is included to introduce the combined effect of the rotation of four rotors and the vehicle body, represented as:

$$\mathbf{m}_g = \sum_{i=1}^4 J_p(\boldsymbol{\omega}_B \times \hat{\mathbf{e}}_3)(-1)^{i+1}\Omega_i \quad (11)$$

where  $\Omega_i$  is the angular speed of the  $i$ th rotor and  $J_p$  is the motor's inertia. According to [42], this inertia is found to be small; hence, the effect of gyroscopic moment is ignored in the controller formulation.

The control input  $u_1$  provides the thrust in a vertical axis and is defined as:

$$f_t = u_1 = k_t \sum_{i=1}^4 \Omega_i^2 \quad (12)$$

where  $k_t$  is the thrust coefficient. The other three control inputs ( $u_2, u_3, u_4$ ), dependent on the angular speeds of the rotors, are represented as:

$$\begin{cases} \tau_x = u_2 = k_t l (\Omega_3^2 - \Omega_1^2) \\ \tau_y = u_3 = k_t l (\Omega_4^2 - \Omega_2^2) \\ \tau_z = u_4 = k_d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases} \quad (13)$$

where  $l$  is the distance from the center of the quadrotor to any rotor, as the quadrotor is symmetrical according to Assumption 2. The drag coefficient is introduced using the term  $k_d$ . The complete dynamic model of the quadrotor can be obtained by substituting the control inputs from Eqs. (12) and (13) to Eqs. (9) and (10).

## 5 Detection, tracking, and estimation

In this section, the problem of detection and tracking of the landing platform is investigated. The information is later used to determine the relative pose between the quadrotor and the platform and to guide the quadrotor towards the platform.

### 5.1 Target detection

The target landing platform is detected by the OptiTrack cameras using the four retro-reflective markers placed on the corners of the platform. It is assumed that there is no transfer of information between the quadrotor and the platform.

### 5.2 Kalman filter-based pose estimation of landing platform

The linear Kalman filter serves two purposes: tracking and estimation of the mobile platform's position and velocity. Assuming that the mobile platform can only move in a 2D plane, its motion can be defined using translation along the  $x$ -axis and  $y$ -axis. To track the mobile platform, the position information  $\{x_p, y_p\}$  must be known at every instant. The KF mathematical model for the mobile platform's tracking is defined as follows:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (14)$$

where  $\mathbf{x}(k) = [x_p(k) \quad y_p(k) \quad \dot{x}_p(k) \quad \dot{y}_p(k)]^T$  is defined as the state vector based upon displacements and velocities, and  $\mathbf{u}(k) = [\ddot{x}_p(k) \quad \ddot{y}_p(k)]^T$  applies the acceleration as the control input at instance  $k$ . The state-space model presented in Eq. (14) has a state-transition matrix  $\mathbf{A}$  and input matrix  $\mathbf{B}$ , defined as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}$$

where  $T_s$  is the control time step.

To account for the uncertainty due to the inaccuracy of the model and/or the inputs (acceleration), a white-noise term is introduced to the model in Eq. (14) as:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \quad (15)$$

It is assumed that  $\mathbf{w}$  is a Gaussian distribution noise with a mean of 0 and a variance  $\mathbf{B}\mathbf{Q}_b\mathbf{B}^T$ , where  $\mathbf{Q}_b$  is a diagonal matrix whose diagonal values are chosen based upon the laboratory tests as 2. The off-diagonal elements of  $\mathbf{Q}_b$  are zeros; this is due to the assumption that the motions along  $x$  and  $y$  axes are uncorrelated.

The measurement vector  $\mathbf{z}$  is defined as:

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (16)$$

where  $\mathbf{H}$  is the output matrix that maps the state vector into the measurement domain and  $\mathbf{v}$  is the measurement noise. The output matrix  $\mathbf{H}$  is chosen as:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The measurement noise  $\mathbf{v}$  is assumed to be Gaussian distributed with a mean of 0 and a variance  $\mathbf{H}\mathbf{R}_b\mathbf{H}^T$ , where the diagonal values of  $\mathbf{R}_b$  are chosen as 2.1 through trial and error.

### 5.3 Velocity estimation of quadrotor

Only the position and attitude of the quadrotor are available through Motive and are either streamed to the ground control station or recorded as logged data. Therefore, the linear and angular velocity of the quadrotor must be estimated. The linear velocity at the inertial frame ( $l$ ) using raw position data ( $x_{raw}$ ) is determined as follows:

$$v_{raw}[N] \approx (x_{raw}[N] - x_{raw}[N-1])/dt \quad (17)$$

where  $dt$  is the step size between two successive data elements. The raw velocity information is then processed through a sliding window filter to obtain smoother data over a finite duration.

$$v[N] = \sum_{i=0}^{m-1} \mu_i v_{raw}[N-i]; \quad \sum_{i=0}^{m-1} \mu_i = 1 \quad (18)$$

The angular velocity of a quadrotor is estimated using the Lie algebra of the rotation matrix as:

$$w_{raw}[N] \approx -(\mathfrak{R}[N]\mathfrak{R}[N-1])/dt)^\vee \quad (19)$$

where  $\vee$  is denoted as the inverse of the Lie map and the rotation matrix  $\mathfrak{R}$  between the inertial frame and body-fixed reference frame is defined as:

$$\mathfrak{R}[N] = \mathcal{R}(q[N])\mathcal{L}(q[N])^T \quad (20)$$

where the auxiliary matrices  $\mathcal{R}$  and  $\mathcal{L}$  are defined using quaternion  $q[N] = [q_0 \quad q_1 \quad q_2 \quad q_3]^T$  obtained from the rotation matrix  $\mathbf{R}$  defined in (4).

$$\mathcal{R}(q) = \begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix};$$

$$\mathcal{L}(q) = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}$$

The filtered angular velocity estimate of angular velocity is obtained using a sliding window as follows:

$$w[N] = \sum_{i=0}^{m-1} \mu_i w_{raw}[N-i]; \quad \sum_{i=0}^{m-1} \mu_i = 1 \quad (21)$$

As the position and attitude data of the quadrotor are expected to vary at a much faster rate, a smaller window length of 4 is chosen based on the experimental tests.

## 6 Model predictive control of quadrotor

The model predictive controller is responsible for determining the desired control inputs to reduce the error between the trajectories of the quadrotor and the platform. This process is performed by solving the constrained tracking problem and minimizing the performance index of the optimization problem. The manipulated variable is computed using the Quadratic Programming (QP) optimization. The quadrotor UAV's control system embodies a bifurcated architecture, delineated in Fig. 2, encompassing a positional regulator and an attitude regulator. Each regulator orchestrates feedback acquisition acquired via MCS, with the positional regulator managing position feedback and the attitude regulator honing in on attitude feedback. Notably, the platform's movements remain pre-established and autonomous from the quadrotor's actions. The linear Kalman Filter (KF) serves as a powerful tool for noise reduction and prediction within the system. By effectively filtering out white noise originating from sensor measurements, it ensures a smoother, more accurate estimate of the platform's position. Additionally, the KF integrates a dynamic model of the landing platform's motion, utilizing both its current state and previous observations to predict future states. This predictive capability not only aids in compensating latency but also enhances the overall responsiveness of the tracking system. The resulting position estimate is then seamlessly integrated into the MPC block, acting as the desired trajectory for precise tracking.

### 6.1 Design of the multi-objective optimization function

The optimization functions for the MPC are designed based on the following discrete form of the system

$$\begin{aligned} x(k+1) &= f(x(k), U(k)) \\ y(k) &= Cx(k) \end{aligned} \quad (22)$$

where  $x$  is a state vector that holds the position and velocity information of the quadrotor,  $f$  represents the continuously differentiable function concerning its arguments, and  $f(0, 0) = 0$ . The position and velocity of the quadrotor obtained as the output are compared with the reference information from the platform.

For the system given in (22), the multi-objective optimization problem solved using QP is defined as:

$$\begin{aligned} \min_{x,u} J(x(k), u(k)) &= \|y(k) - r(k)\|_Q^2 + \|u(k)\|_R^2 \\ &+ \|\Delta u(k)\|_S^2 + \|x(k)\|_P^2 \end{aligned} \quad (23)$$

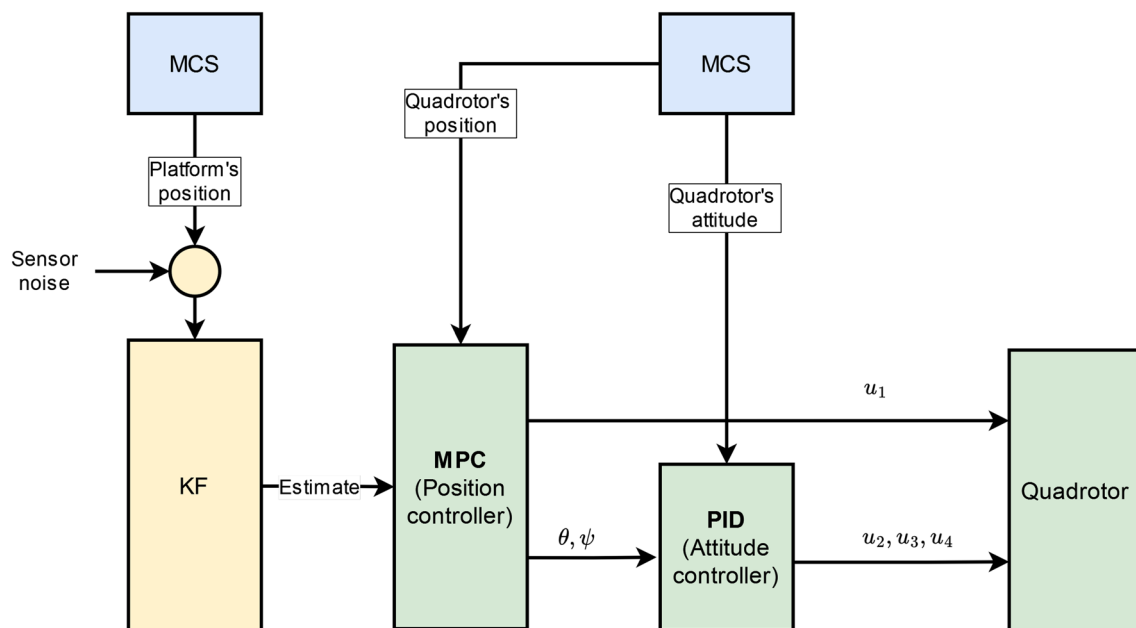


Fig. 2 Vision-based navigation, tracking, and control system

where  $Q$  is a positive diagonal matrix that is employed for adjusting the tracking performance;  $R$  is a positive diagonal matrix to adjust the gain for the control input;  $S$  is a positive diagonal matrix to adjust the weights for the change in control actions; and  $P$  is a symmetric positive-definite matrix that is used to tune penalties.

## 6.2 MPC formulation

Given an auxiliary control law

$$u = \kappa_a(x) \quad (24)$$

and a positive invariant set  $X_f \subset X$  containing the origin  $x = 0$ . Thus for the closed-loop system

$$x(k+1) = f(x(k), \kappa_a(x(k))) \quad (25)$$

and for any  $x(\bar{k}) \in X_f$ , the states and control inputs are defined as

$$\begin{aligned} x(k) &\in X_f, & k &\geq \bar{k} \\ u(k) &= \kappa_a(x(k)) \in U, & k &\geq \bar{k} \end{aligned} \quad (26)$$

The Receding Horizon (RH) method is applied for prediction horizon  $N$  to determine the optimized input sequence [43]

$$u(k), u(k+1), \dots, u(k+N-1)$$

by minimizing the cost function

$$\begin{aligned} \min_{x,u} J(x(k), u(\cdot), N) &= \min \left\{ \sum_{i=0}^{N-1} \|e(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right. \\ &\quad \left. + \|\Delta u(k+i)\|_S^2 + \|x(k+i)\|_P^2 \right\} \\ &\quad + V_f(x(k+N)) \end{aligned} \quad (27)$$

s.t.:  $x(k+1) = f(x(k), U(k))$

$$u_{\min} \leq u(k) \leq u_{\max}, k = 0, \dots, N-1$$

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, k = 0, \dots, N-1$$

$$e_{\min} \leq e(k) \leq e_{\max}, k = 1, \dots, N$$

The RH solution implicitly defines the MPC time-invariant control law as

$$u = \kappa_{RH}(x) \quad (28)$$

**Theorem 1** Let  $X^{RH}(N)$  be the set of states that provides the optimal solution. If, for any  $x \in X_f$  and auxiliary control law  $\kappa_a$ , the condition

$$\begin{aligned} V_f(x(k), \kappa_a(x(k))) - V_f(x(k)) &+ (\|e(k)\|_Q^2 \\ &+ \|\kappa_a(x(k))\|_R^2 + \|\Delta \kappa_a(x(k))\|_S^2 + \|x(k)\|_P^2) \leq 0 \end{aligned} \quad (29)$$

is fulfilled and

$$V_f(x) \leq \alpha_f(\|x\|) \quad (30)$$

where  $\alpha_f(\|x\|)$  is a class  $K$  function, then the origin of the closed-loop system with the MPC-RH control law is an asymptotically stable equilibrium point with the region of attraction  $X^{RH}(N)$ . Moreover, if  $\alpha_f(\|x\|) = b\|x\|^2$  and  $X_f = X^{RH}(N)$ , then the origin defined by  $x = 0$  and  $u = 0$  is exponentially stable in  $X^{RH}(N)$ .

**Proof** Let  $x^*(k) \in X^{RH}(N)$  and  $U^*(k, N)$  be the optimal solution at  $k$  with prediction horizon  $N$ . Then, at time  $k+1$

$$\tilde{U}(k+1, N) = [u_k^*(k+1), u_k^*(k+N-2), \dots, u_k^*(k+N-1), \kappa_a(x(k+N))] \quad (31)$$

is a feasible solution, so that  $x(k+1) \in X^{RH}(N)$ . Hence, the following condition is satisfied in  $X^{RH}(N)$ .

$$V(x, N) := J(x, \kappa_{RH}(x), N) \geq \|e\|_Q^2 + \|x\|_p^2 \quad (32)$$

At time  $k$ , the sequence

$$\tilde{U}(k+1, N) = [U^*(k, N), \kappa_a(x(k+N))] \quad (33)$$

is feasible for the MPC problem with horizon  $N+1$  and

$$\begin{aligned} J(x, \tilde{U}(k, N+1), N+1) &= V(x, N) - V_f(x(k+N)) + V_f(x(k+N+1)) \\ &+ \|e(k+N)\|_Q^2 + \|\kappa_a(x(k+N))\|_R^2 \\ &+ \|\Delta\kappa_a(x(k+N))\|_S^2 + \|x(k+N)\|_p^2 \leq V(x, N) \end{aligned} \quad (34)$$

so that we have the monotonicity property (concerning  $N$ )

$$V(x, N+1) \leq V(x, N), \forall x \in X^{RH}(N) \quad (35)$$

with  $V(x, 0) = V_f(x), \forall x \in X_f$ . Then

$$V(x, N+1) \leq V(x, N) \leq V_f(x) \leq \alpha_f(\|x(k)\|), \forall x \in X_f \quad (36)$$

and the condition  $V(x, N) \leq \psi(\|x\|), \forall x \in X_f$  is satisfied. Finally

$$\begin{aligned} V(x, N) &= \|e\|_Q^2 + \|\kappa_{RH}\|_R^2 + \|\Delta\kappa_{RH}\|_S^2 + \|x\|_p^2 \\ &+ J(f(x, \kappa_{RH}(x)), u^*(k+1, N-1), N-1) \\ &= \|e\|_Q^2 + \|\kappa_{RH}\|_R^2 + \|\Delta\kappa_{RH}\|_S^2 + \|x\|_p^2 \\ &+ V(f(x, \kappa_{RH}(x)), N-1) \\ &\geq \|e\|_Q^2 + \|\kappa_{RH}\|_R^2 + \|\Delta\kappa_{RH}\|_S^2 + \|x\|_p^2 \\ &+ V(f(x, \kappa_{RH}(x)), N) \\ &\geq \|e\|_Q^2 + \|x\|_p^2 \\ &+ V(f(x, \kappa_{RH}(x)), N-1), \forall x \in X^{RH}(N) \end{aligned} \quad (37)$$

and also the condition  $\Delta V(x) \leq -r(\|x\|), \forall x \in X^{RH}(N)$  is satisfied. So,  $V(x, N)$  is a Lyapunov function, and we conclude by Lyapunov's stability theorem that the equilibrium  $x = 0, u = 0$  is exponentially stable if  $\alpha_f(\|x\|) = b\|x\|^2, X_f = X^{RH}(N)$ .

### 6.3 Hard constraints

Due to the physical saturation of the motor torque input, the hard constraints are introduced to the induced efforts as follows;

$$0 \leq f_t(k+j|k) \leq 4k_t\Omega_{max}^2 \quad (38a)$$

$$-lk_t\Omega_{max}^2 \leq \tau_i(k+j|k) \leq lk_t\Omega_{max}^2 \quad (38b)$$

$$-2k_d\Omega_{max}^2 \leq \tau_z(k+j|k) \leq 2k_d\Omega_{max}^2 \quad (38c)$$

where  $i = x, y$  and  $j = 1, 2, \dots, m - 1$ . The maximum speed,  $\Omega_{max}$ , is the same for all rotors as the motors are identical and the quadrotor is balanced according to Assumption 2.

The inherent nature of MPC assists in predicting future trajectories, thus making it useful for the tracking of moving targets. The predictive model and the MPC optimization problem are solved using the MPC designer toolbox provided by MATLAB. The state-space information is provided to the designer and later used to tune the controller, prediction horizon, and control horizon. The prediction and control horizon are configured as 9 and 3, respectively.

## 7 Experimental results and discussion

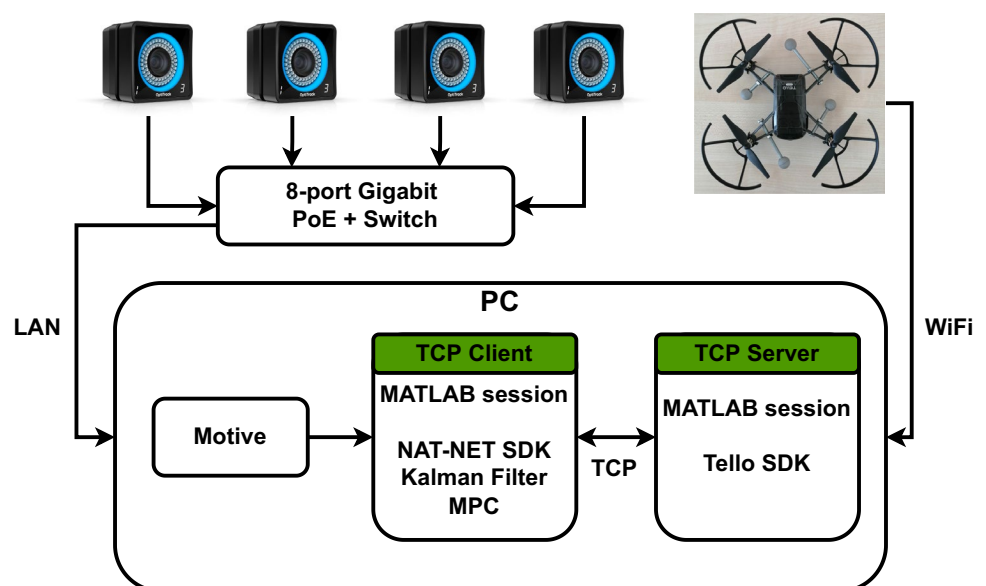
Before initiating the experiments, calibration is performed for the MCS by tracing the motion sensor markers using the Motive. During the operation, the NAT-NET SDK is used by MATLAB to receive the data stream from the Motive. The next step is to build a WiFi communication link between MATLAB and the quadrotor. The mobile platform is set up by uploading the intended trajectory to the open-source hardware (Arduino) using the IDE software. These experimental procedures are repeated for each scenario to ensure consistency and reliable performance. Figure 3 highlights the communication chain concerning the systems and the information flow.

To evaluate the performance of the proposed vision-based system, a series of three different experiments, each including 20 trials, is performed. It is observed that the success of the experiments is reduced when the marker(s) are occluded or the WiFi connection between the PC and the quadrotor is unstable, or the platform travels out of the defined workspace. The overall efficiency of the experiments carried out in this study, though, is recorded as exceeding 90%, thus implying the successful landing of the quadrotor on the stationary/moving platform. Another critical factor influencing landing precision is the resolution of the estimated position of the quadrotor which can fluctuate depending on its distance and angle from the camera or the MCS. This variation in resolution is primarily a result of the principles of triangulation, which are commonly used in the MCS to estimate the position of objects. When the platform is closer to the camera, the MCS can provide higher-resolution data, leading to more accurate position estimates. This can improve the quadrotor's ability to land precisely. Conversely, as the quadrotor moves farther away, the resolution decreases, potentially leading to less precise position estimates.

### 7.1 Stationary target

Initially, the most basic scenario of a stationary landing platform is considered to test the performance of the proposed strategy. From several trials carried out in this way, the results from one such trial are recorded where the relative distance between the quadrotor and the platform is defined as 214 cm. As the process is initiated, the quadrotor takes off, corrects

Fig. 3 Communication link among all the systems



its heading, and then starts moving towards the target by taking a step of 35 cm. This process continues until the pose error defined between the platform and the quadrotor is not acceptable. Figure 4a–d shows the complete sequence of events from the initial position to taking off, maneuvering, and finally landing.

Figure 5 illustrates the initial position of the quadrotor, the stationary platform, and the trajectory of the quadrotor. The sequence begins with the quadrotor taking off; then, it ascends to an appropriate altitude before commencing its approach towards the target. As the pose error along the  $x$  and  $y$  axes falls within the acceptable range, the descent phase begins. Upon reaching a minimal height difference between the quadrotor and the platform, the entire process concludes successfully. Figure 6a, b shows the actual position of the landing platform and its estimation obtained using the KF in a 2D plane. The error norm between the actual and estimated positions for both  $x$  and  $y$  axes is plotted in Fig. 6c. It is evident from the figures that the future prediction of the platform has zero error since it is stationary.

Figure 7a, b illustrates the  $(x, y)$  trajectories of the mobile platform and the quadrotor. Once the estimation information is available, the quadrotor starts moving towards the platform, which is observable at  $t = 14$ s. In the last few seconds of the experiment, the quadrotor maintains its position in the  $x - y$  plane and starts to descend. The changes in the height of the quadrotor due to its movements are recorded and displayed in Fig. 7c. It can be observed that the quadrotor goes through the phases of take-off and hovering before commencing the descent at about  $t = 23$ s. As the pose error between the quadrotor and the platform is minimal, the experiment is concluded successfully.

## 7.2 Dynamic target

### 7.2.1 Square-shaped trajectory

For this experiment, the straight-line distance between the landing platform and the quadrotor is defined as 71 cm. Figure 8a–d shows the complete sequence of events: from the rest position to taking off, tracking, and finally

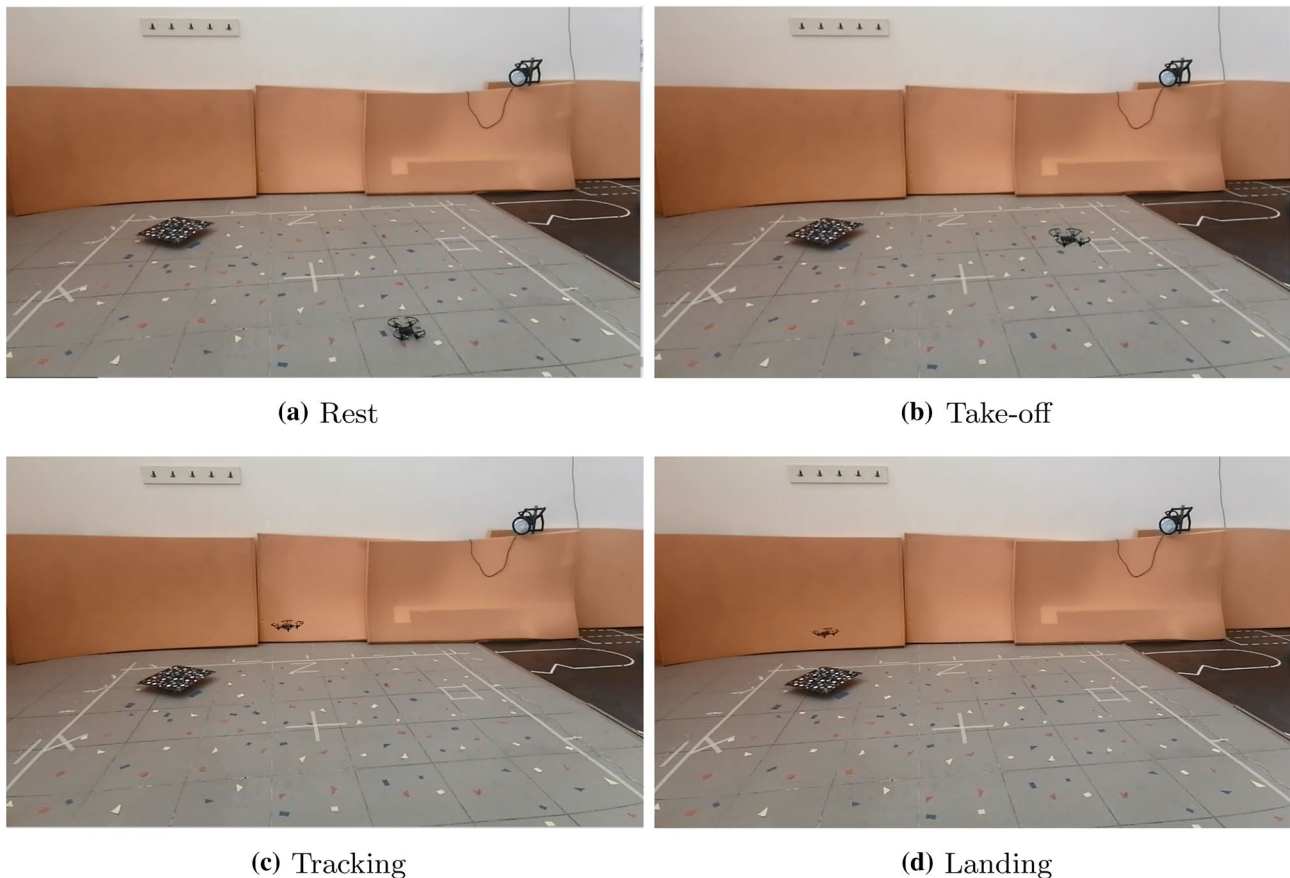
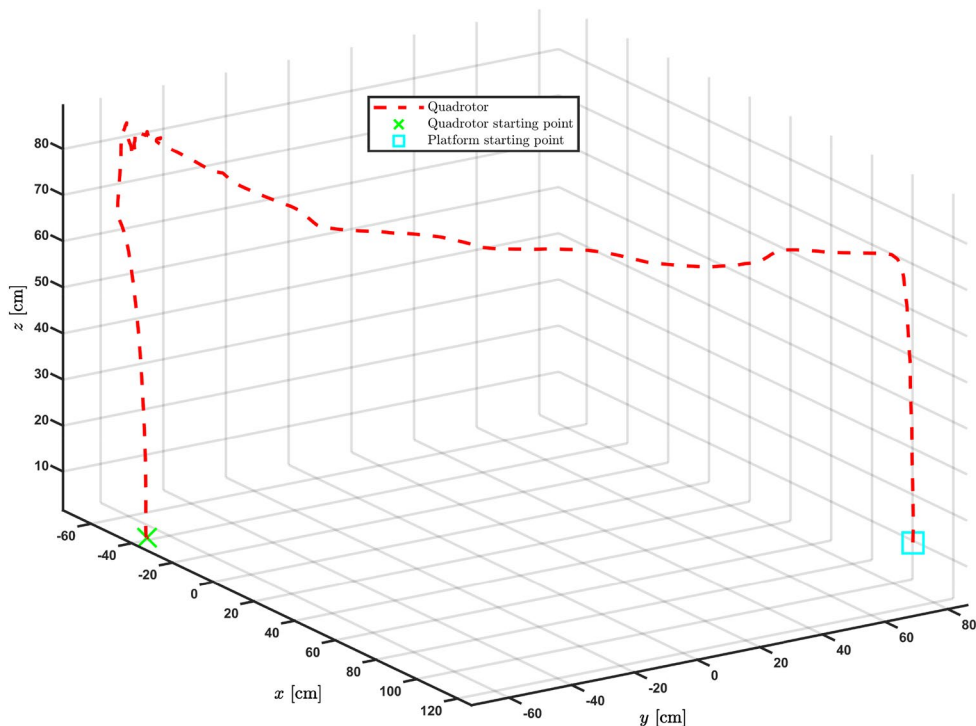
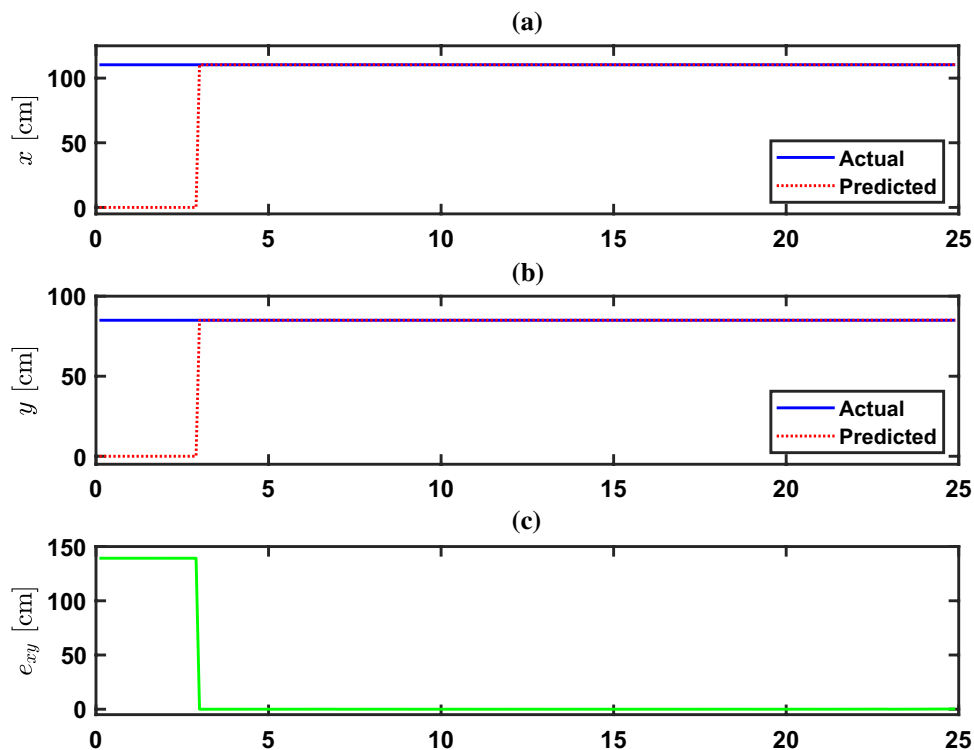


Fig. 4 Sequence of events concerning the quadrotor and the platform in the static scenario

**Fig. 5** Trajectories of the quadrotor and the platform in the static scenario

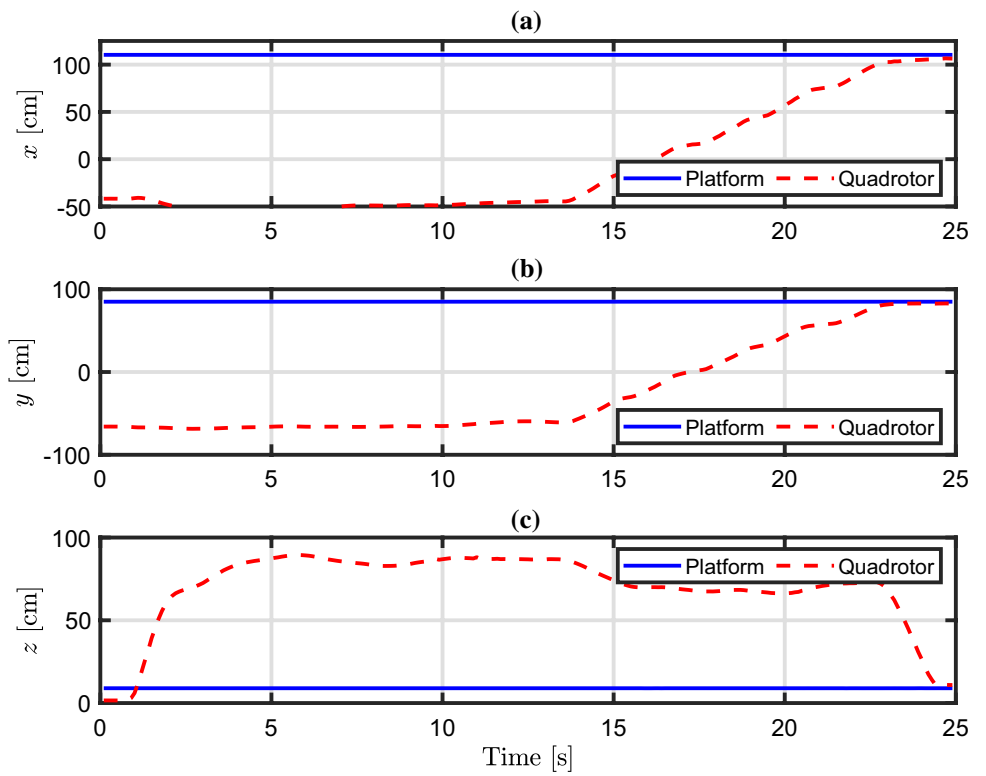


**Fig. 6** Actual and predicted position of the landing platform in the static scenario. **a**  $x$ -axis **b**  $y$ -axis **c** Combined error along  $x$  and  $y$  axes



landing. In this scenario, the target platform maneuvers on a pre-defined trajectory, shown in Fig. 9. The movement of the platform is triggered at  $t + 2$  s. The quadrotor takes off and waits for the first prediction to become available. After each prediction, the quadrotor corrects its heading towards the predicted target and moves closer. The target is assumed to be moving at a constant speed. On the corners, some irregular movements are recorded that are mainly due to the estimation. It can be observed that the MPC thoroughly tracks the reference trajectory with minimum error due to its ability to generate an optimal output at each time step during the navigation.

**Fig. 7** Position measurement of the platform and the quadrotor in the static scenario. **a** x-axis **b** y-axis **c** z-axis



**(a)** Rest



**(b)** Take-off



**(c)** Tracking



**(d)** Landing

**Fig. 8** Sequence of events concerning the quadrotor and the platform in the square-shaped scenario

**Fig. 9** Trajectories of the quadrotor and the platform for the square-shaped scenario

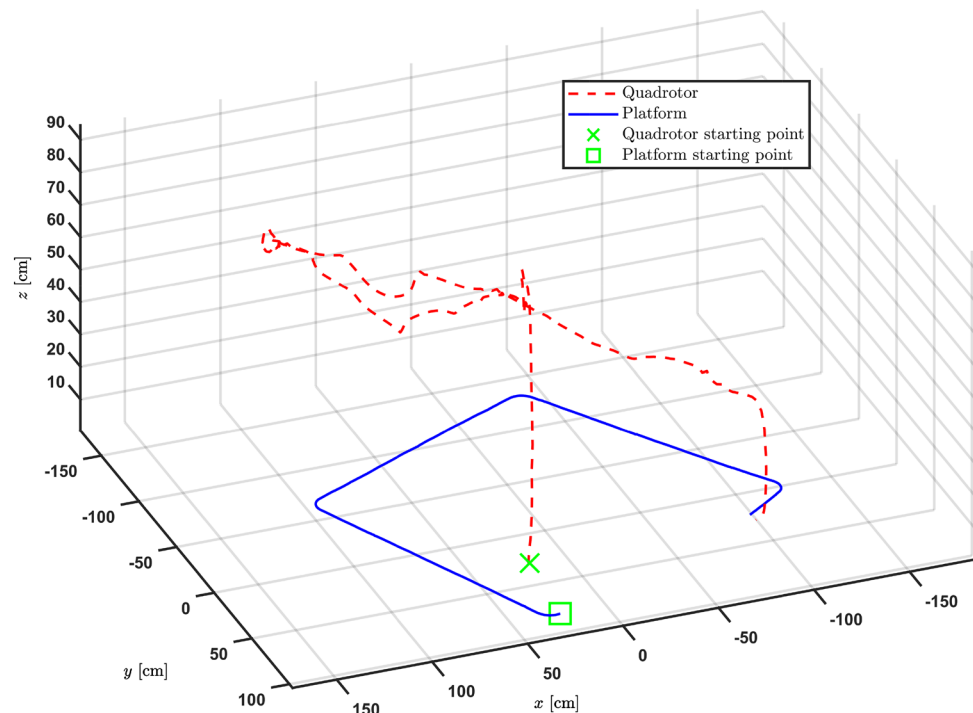


Figure 10a, b shows the actual and the predicted positions of the landing platform along the  $x$  and  $y$  axes, whereas the combined error norm is plotted in Fig. 10c. It can be observed that the estimation obtained through the KF is very much closer to the actual measurement. Small spikes in the prediction error are observed due to the platform being in a dynamic state. The maximum predicted error is recorded as 10 cm, which is considered acceptable.

The position of the quadrotor and the landing platform in the  $x - y$  plane is depicted in Fig. 11a, b. At  $t = 28$  s and  $t = 35$  s, the trajectories of the platform and the quadrotor appear to make close contact; yet, to initiate the landing process, the error along both  $x$  and  $y$  should be within an acceptable range. Figure 11c shows the elevation of the quadrotor and the platform. Though the platform maneuvers on the ground, a small variation of  $\pm 2.5$  cm is recorded in its height by the MCS. The quadrotor initiates its movements - i.e., take-off at  $t = 15$  s and attaining the height of 90 cm. The MPC computes the sequence of events to optimally steer the quadrotor towards the platform and reduce the tracking error. At  $t = 58$  s, the quadrotor makes a successful landing on the moving platform.

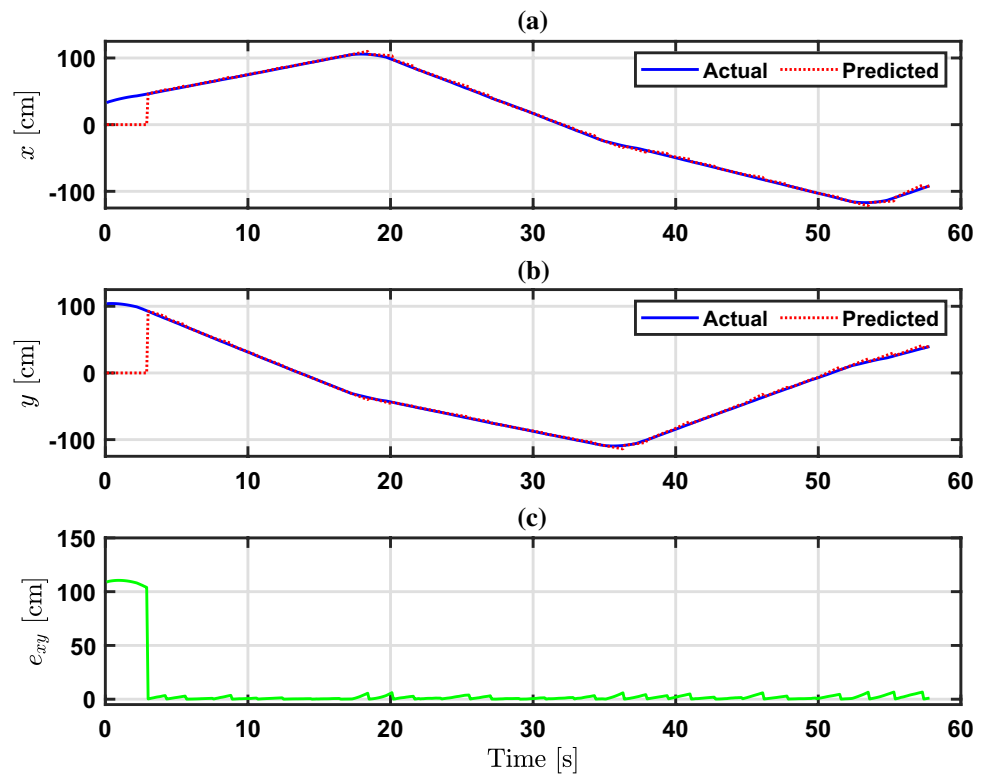
### 7.2.2 Circular-shaped trajectory

The mobile platform, initially placed at a straight line distance of 114 cm from the quadrotor, is programmed to maneuver in a circular-shaped trajectory of a 100 cm diameter. After taking off and stabilizing itself, the quadrotor starts moving towards the predicted location of the platform. Figure 12a–d shows four snapshots to depict each event, while Fig. 13 shows the traveling trajectories of the quadrotor and the mobile platform making a circular motion. After take-off, the quadrotor waits for the prediction concerning the position of the platform to become available. Once done, the quadrotor starts traveling towards the predicted target with a step size of 35 cm. After some maneuvers, the quadrotor successfully performs the landing operation on the moving platform. Though the estimation is performed with high frequency, the time required by the Tello EDU quadrotor to process the command from the navigation and control system somehow limits the overall performance. In this way, the test scenario exposes the technical limitation of the Tello EDU quadrotor.

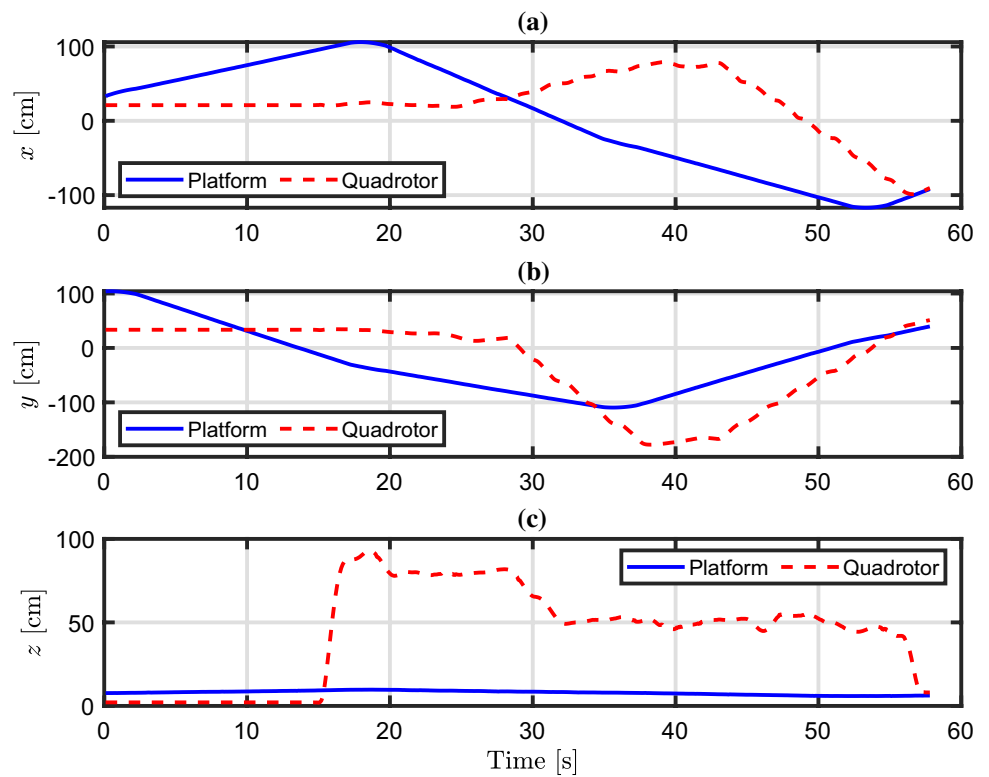
To overcome this limitation, a TCP server/client program is developed that runs on a separate MATLAB session. The TCP server queues up the commands from the main navigation MATLAB program necessary to control the quadrotor. The server continuously monitors the quadrotor status and only sends a command when the quadrotor is available. This solution improves the delay time caused by the quadrotor SDK.

Figure 14a, b plots the actual location of the landing platform and its estimation through KF for the  $x$  and  $y$  axes. The combined error depicting the difference between the actual and estimated data is shown in Fig. 14c. Due to the

**Fig. 10** Actual and predicted position of the landing platform in the square-shaped scenario. **a** x-axis **b** y-axis **c** Combined error along x and y axes



**Fig. 11** Position measurement of the platform and the quadrotor in the square-shaped scenario. **a** x-axis **b** y-axis **c** z-axis



continuous change in the pose of the platform, small variations appear in the error plot; however, more importantly, the obtained steady-state error is bounded and within an acceptable range.

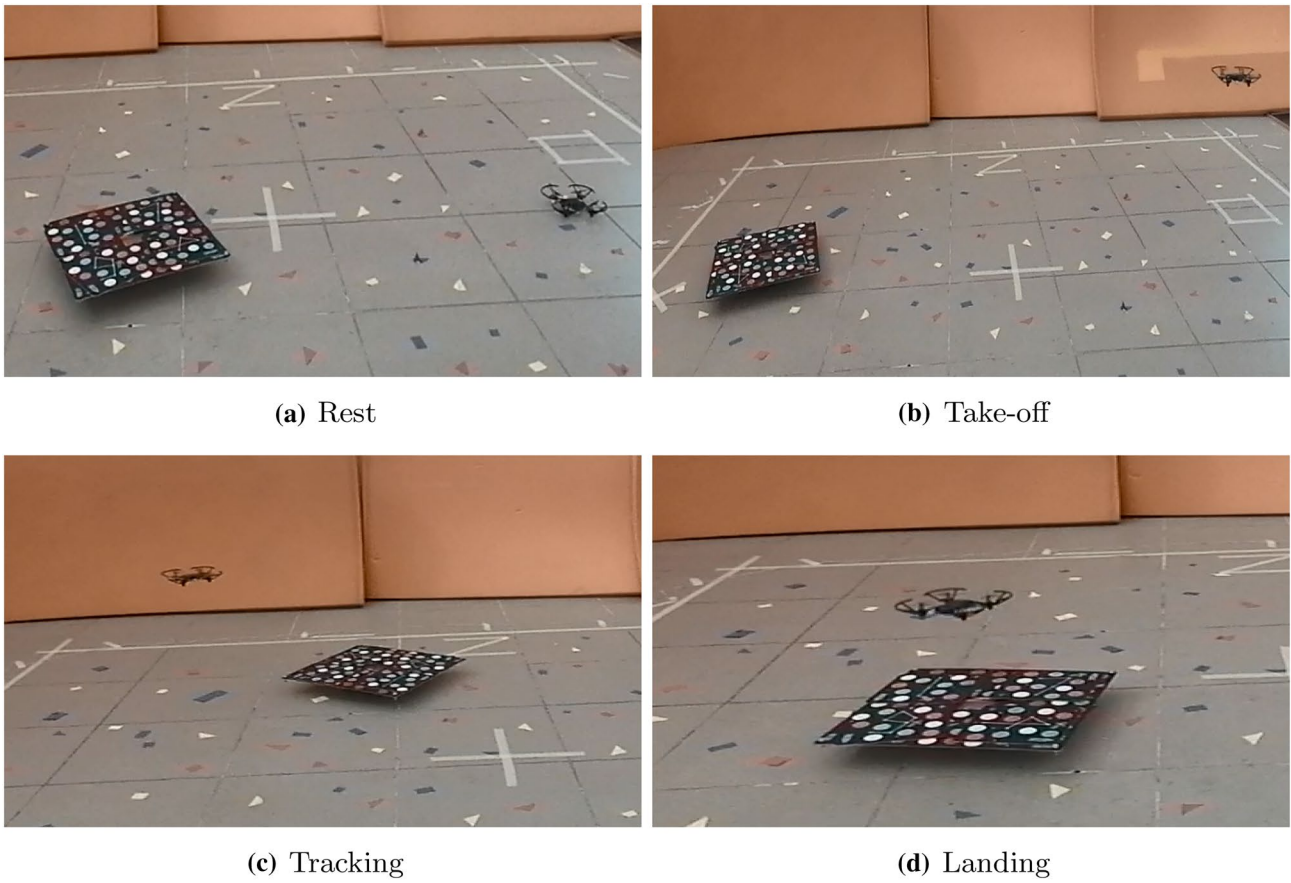
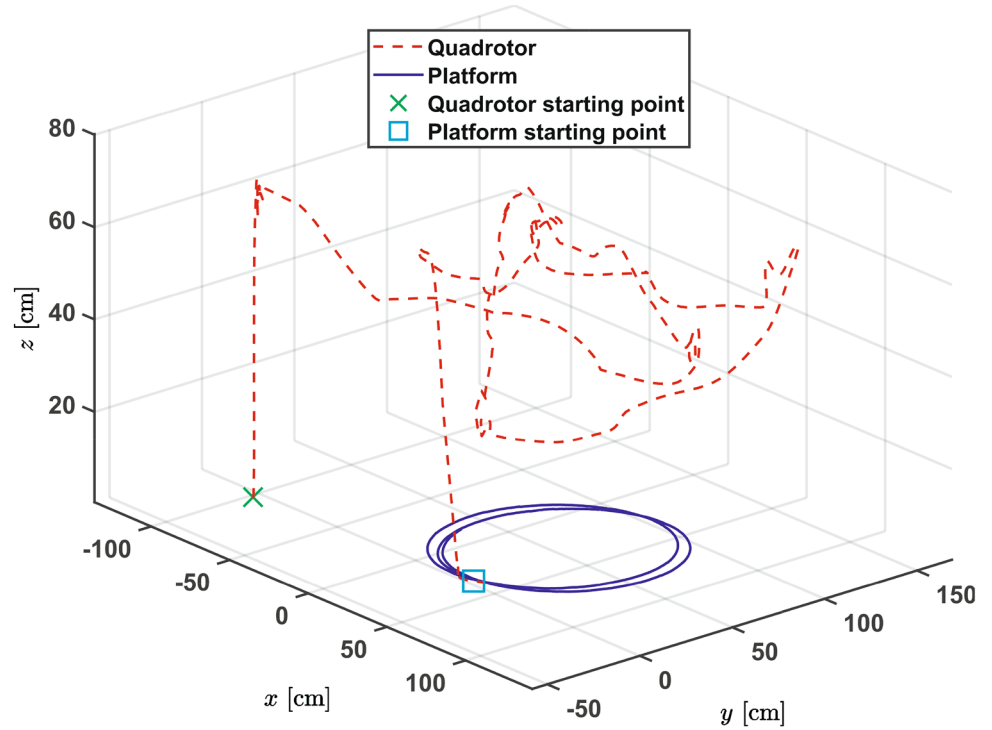
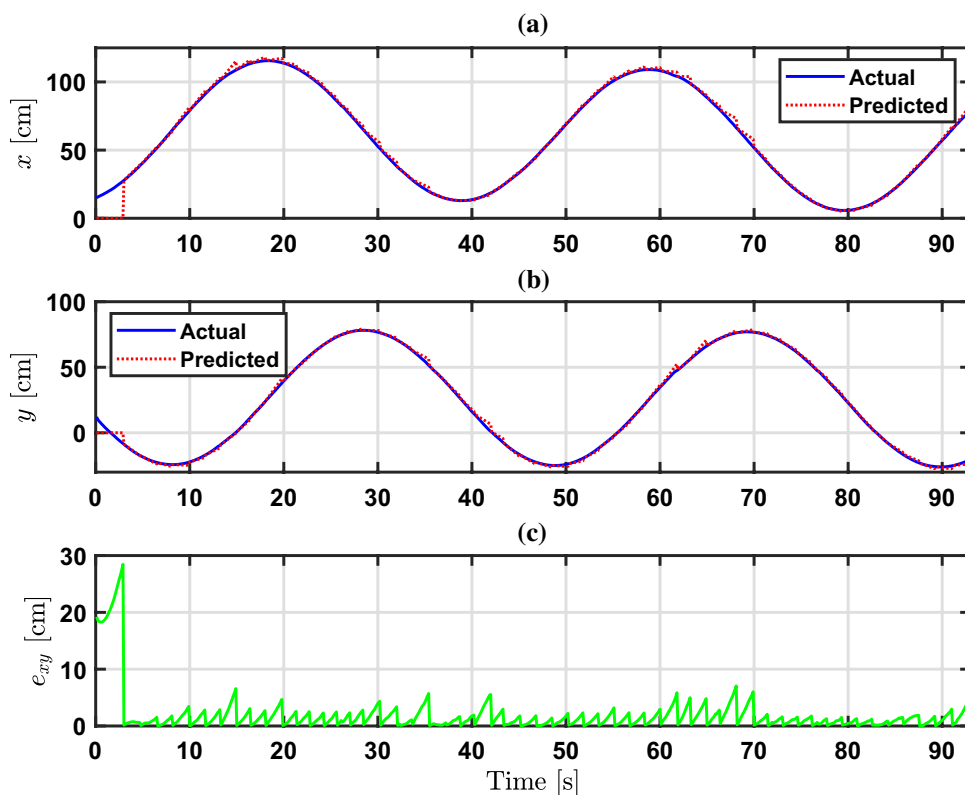


Fig. 12 Sequence of events concerning the quadrotor and the platform in the circular-shaped scenario

Fig. 13 Trajectories of the quadrotor and the platform in the circular-shaped scenario



**Fig. 14** Actual and predicted position of the landing platform in the circular-shaped scenario. **a** x-axis **b** y-axis **c** Combined error along x and y axes



**Fig. 15** Position measurement of the platform and the quadrotor in the circular-shaped scenario. **a** x-axis **b** y-axis **c** z-axis

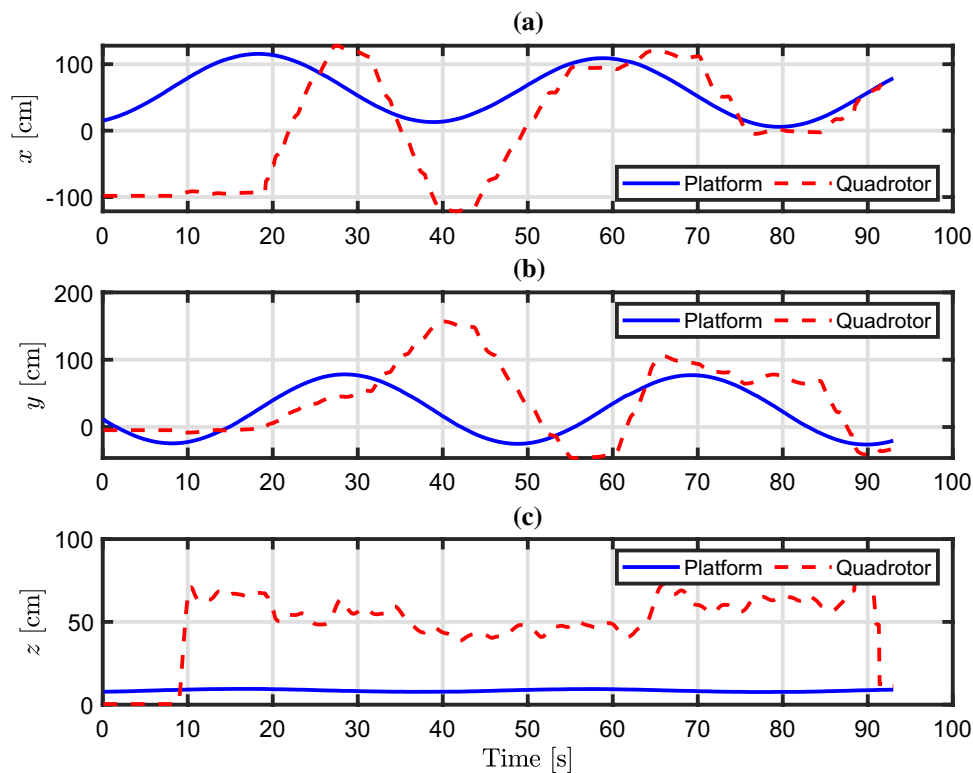


Figure 15a, b shows the movements of the platform and the quadrotor along the x and y axes. Before landing, the quadrotor performs a number of attempts to approach the target; yet, due to the continuously changing conditions, it cannot initiate the landing operation. The landing performance of the quadrotor in terms of the displacement along

the z-axis is depicted in Fig. 15c. It is observed that after take-off, the quadrotor stabilizes itself and starts maneuvering towards the target. The landing is initiated when the platform and the quadrotor are aligned within an acceptable range. Around  $t = 60$ s, the error along both the x and y axes converges towards zero and enters the acceptable range. However, only after a few seconds, the trajectory along the y-axis appears to deviate from its reference. This causes the quadrotor to terminate the landing operation and re-adjust its height, prolonging the entire operation but still landing in a successful second attempt at almost 90 s. This scenario is considered more challenging due to the continuous change in the heading of the platform and the velocities along both axes, which was not the case in any previous scenarios.

## 8 Conclusion

In the present study, a quadrotor (Tello EDU), a Kalman filter, a model predictive controller, and a motion capture system (OptiTrack) are used to design a vision-based system, whose role is to achieve an autonomous landing of the quadrotor on a moving platform. The tracking and landing on the platform is devised as a constrained tracking problem. An optimal model-based predictive controller is designed to ensure minimum energy consumption, smooth motion, and closed-loop stability while limiting the induced efforts. To provide smooth and noise-free information to the MPC algorithm, a KF is employed to estimate the position and velocity of the platform. This not only helps to accurately predict the future position of the mobile platform, but it also reduces the time required for landing. Several experiments are conducted to test the proposed vision-based system. The experiments involve different movements of the mobile platform, such as square, circular, and stationary. The performance tests reveal that the quadrotor is able to track various types of trajectories and, subsequently, land safely on the moving or stationary platform.

Future work can focus on the improvement of the MPC by introducing different noises and disturbance models. In the presence of such constraints, noises, and disturbances, an NMPC based on Adaptive Dynamic Programming (ADP) can be utilized to achieve fast response and strong robustness. Another potential area for upcoming attempts is to evaluate the performance of the proposed system by utilizing different estimation algorithms, such as an Extended Kalman Filter (EKF), an Unscented Kalman Filter (UKF), and a Particle Filter (PF).

**Acknowledgements** We would like to express our sincere gratitude to Atilim University for their unwavering support and provision of resources throughout this research.

**Author contributions** Conceptualization, A.Q. and M.U.K.; methodology, A.Q., M.U.K., and B.I.; software, A.Q.; validation, M.U.K., and B.I.; formal analysis, A.Q., M.U.K., B.I.; investigation, A.Q. and M.U.K.; resources, M.U.K.; data curation, M.U.K., and B.I.; writing—original draft preparation, A.Q., and M.U.K.; writing—review and editing, M.U.K., and B.I.; visualization, A.Q. and M.U.K.; supervision, M.U.K.; project administration, M.U.K. All authors have read and agreed to the published version of the manuscript.

**Funding** The authors declare no specific funding for this work.

**Data availability** No datasets were generated or analyzed during the current study.

## Declarations

**Competing interest** The authors declare there are no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Elmakis O, Shaked T, Degani A. Vision-based uav-ugv collaboration for autonomous construction site preparation. *IEEE Access*. 2022;10:51209–20.
2. Villa DK, Brandao AS, Sarcinelli-Filho M. A survey on load transportation using multirotor uavs. *J Intell Robot Syst*. 2020;98(2):267–96.
3. Selvaraj R, Kuthadi VM, Baskar S. Real-time agricultural field monitoring and smart irrigation architecture using the internet of things and quadrotor unmanned aerial vehicles. *Agron J*. 2022;115:1–20.

4. Cardona GA, Ramirez-Rugeles J, Mojica-Nava E, Calderon JM. Visual victim detection and quadrotor-swarm coordination control in search and rescue environment (2088–8708). *Int J Electr Comput Eng*. 2021;11(3):2079.
5. De Moraes RS, De Freitas EP. Multi-uav based crowd monitoring system. *IEEE Trans Aerosp Electron Syst*. 2019;56(2):1332–45.
6. Abdul-Samed BN. Oil facilities surveillance using an autonomous quadrotor. *J Petrol Res Stud*. 2022;12(Suppl. 1):211–24.
7. Derrouaoui SH, Bouzid Y, Belmouhoub A, Guiatni M, Siguerdidjane H. Recent developments and trends in unconventional uavs control: a review. *J Intell Robot Syst*. 2023;109(3):68.
8. Kakaletsis E, Symeonidis C, Tzelepi M, Mademlis I, Tefas A, Nikolaidis N, Pitas I. Computer vision for autonomous uav flight safety: an overview and a vision-based safe landing pipeline example. *ACM Comput Surv*. 2021;54(9):1–37.
9. Ghommam J, Saad M. Autonomous landing of a quadrotor on a moving platform. *IEEE Trans Aerosp Electron Syst*. 2017;53(3):1504–19.
10. Deniz S, Wu Y, Shi Y, Wang Z. Autonomous landing of evtol vehicles via deep q-networks. In: *AIAA AVIATION 2023 Forum*; 2023, p 4499.
11. Beck H, Lesueur J, Charland-Arcand G, Akhrif O, Gagne S, Gagnon F, Couillard D. Autonomous takeoff and landing of a quadcopter. In: *2016 international conference on unmanned aircraft systems (ICUAS) 2016*; p. 475–484.
12. Duong TH, Le HP, Le MH. Autonomous landing of a quadcopter on a stationary platform. In: *2021 international conference on system science and engineering (ICSSE), 2021*; p. 56–61.
13. Kamath AK, Tripathi VK, Yogi SC, Behera L. Vision-based fast-terminal sliding mode super twisting controller for autonomous landing of a quadrotor on a static platform. In: *2019 28th IEEE international conference on robot and human interactive communication (RO-MAN), 2019*; p. 1–6.
14. Kumar A. Real-time performance comparison of vision-based autonomous landing of quadcopter on a ground moving target. *IETE J Res*. 2021;69:5455–72.
15. Alonge F, Cusumano P, D'ippolito F, Garraffa G, Livreri P, Sferlazza A. Localization in structured environments with uwb devices without acceleration measurements, and velocity estimation using a Kalman–Bucy filter. *Sensors*. 2022;22(16):6308.
16. Thomas J, Welde J, Loianno G, Daniilidis K, Kumar V. Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor. *IEEE Robot Autom Lett*. 2017;2(3):1762–9.
17. You W, Li F, Liao L, Huang M. Data fusion of uwb and imu based on unscented Kalman filter for indoor localization of quadrotor uav. *IEEE Access*. 2020;8:64971–81.
18. Kumar GA, Patil AK, Patil R, Park SS, Chai YH. A lidar and imu integrated indoor navigation system for uavs and its application in real-time pipeline classification. *Sensors*. 2017;17(6):1268.
19. Fernando E, De Silva O, Mann GK, Gosine R. Toward developing an indoor localization system for mavs using two or three rf range anchors: an observability based approach. *IEEE Sens J*. 2022;22(6):5173–87.
20. Kempke B, Pannuto P, Dutta P. Polypoint: guiding indoor quadrotors with ultra-wideband localization. In: *Proceedings of the 2nd international workshop on hot topics in wireless, 2015*; p. 16–20.
21. Aurand AM, Dufour JS, Marras WS. Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume. *J Biomech*. 2017;58:237–40.
22. Dudzik S. Application of the motion capture system to estimate the accuracy of a wheeled mobile robot localization. *Energies*. 2020;13(23):6437.
23. Jan IU, Khan MU, Iqbal N. Visual landing of helicopter by divide and conquer rule. *IEICE Electron Express*. 2011;8(19):1542–8.
24. Wang L, Bai X. Quadrotor autonomous approaching and landing on a vessel deck. *J Intell Robot Syst*. 2018;92(1):125–43.
25. Qi Y, Jiang J, Jin W, Wang J, Wang C, Shan J. Autonomous landing solution of low-cost quadrotor on a moving platform. *Robot Auton Syst*. 2019;119:64–76.
26. Merriaux P, Dupuis Y, Bouteau R, Vasseur P, Savatier X. A study of vicon system positioning performance. *Sensors*. 2017;17(7):1591.
27. NOKOV Mocap. NOKOV Motion Capture System. Accessed Feb. 9, 2022 [Online], 2022.
28. Vicon. Vicon Motion Capture System. Accessed Mar. 17, 2022 [Online], 2022.
29. Qualisys. Qualisys Motion Capture System. Accessed Feb. 9, 2022 [Online], 2022.
30. A Leyard Company. Optitrack Motion Capture System. Accessed Feb. 9, 2022 [Online], 2022.
31. Furtado JS, Liu HH, Lai G, Lacheray H, Desouza-Coelho J. Comparative analysis of optitrack motion capture systems. In: Janabi-Sharifi F, Melek W, editors. *Advances in motion sensing and control for robotic applications*. Cham: Springer; 2019. p. 15–31.
32. Ghasemi A, Parivash F, Ebrahimian S. Autonomous landing of a quadrotor on a moving platform using vision-based fofpid control. *Robotica*. 2022;40(5):1431–49.
33. Lin J, Wang Y, Miao Z, Zhong H, Fierro R. Low-complexity control for vision-based landing of quadrotor uav on unknown moving platform. *IEEE Trans Ind Inform*. 2021;18(8):5348–58.
34. Keipour A, Pereira GA, Bonatti R, Garg R, Rastogi P, Dubey G, Scherer S. Visual servoing approach for autonomous uav landing on a moving vehicle. *arXiv preprint arXiv:2104.01272*, 2021.
35. Zhang X, Fang Y, Zhang X, Jiang J, Chen X. Dynamic image-based output feedback control for visual servoing of multirotors. *IEEE Trans Ind Inform*. 2020;16(12):7624–36.
36. Jan IU, Khan MU, Iqbal N. Combined application of Kalman filtering and correlation towards autonomous helicopter landing. In: *2009 2nd international conference on computer, control and communication, 2009*, p. 1–6.
37. Baca T, Stepan P, Spurny V, Hert D, Penicka R, Saska M, Thomas J, Loianno G, Kumar V. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J Field Robot*. 2019;36(5):874–91.
38. Niu G, Yang Q, Gao Y, Pun MO. Vision-based autonomous landing for unmanned aerial and mobile ground vehicles cooperative systems. *IEEE Robot Autom Lett*. 2021;7:6234–41.
39. Rafifandi R, Asri DL, Ekawati E, Budi EM. Leader–follower formation control of two quadrotors uavs. *SN Appl Sci*. 2019;1(539):1–12.
40. Guo K, Tang P, Wang H, Lin D, Cui X. Autonomous landing of a quadrotor on a moving platform via model predictive control. *Aerospace*. 2022;9(1):34.
41. Feng Y, Zhang C, Baek S, Rawashdeh S, Mohammadi A. Autonomous landing of a uav on a moving platform using model predictive control. *Drones*. 2018;2(4):34.

42. Mellinger D, Lindsey Q, Shomin M, Kumar V. Design, modeling, estimation and control for aerial grasping and manipulation. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, 2011; p. 2668–2673. IEEE.
43. Clarke DW, Scattolini R. Constrained receding-horizon predictive control. In IEE proceedings D (control theory and applications), 1991; volume 138, p. 347–354. IET.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.