

Incremental Nonlinear Dynamic Inversion based Optical Flow Control for Flying Robots: An Efficient Data-driven Approach

Hann Woei Ho and Ye Zhou*

School of Aerospace Engineering, Engineering Campus, Universiti Sains Malaysia, Malaysia
Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands

The authors contributed equally to this work.

Email: aehannwoei@usm.my, zhouye@usm.my *Corresponding author

Abstract—This paper presents a novel approach for optical flow control of Micro Air Vehicles (MAVs). The task is challenging due to the nonlinearity of optical flow observables. Our proposed Incremental Nonlinear Dynamic Inversion (INDI) control scheme incorporates an efficient data-driven method to address the nonlinearity. It directly estimates the inverse of the time-varying control effectiveness in real-time, eliminating the need for the constant assumption and avoiding high computation in traditional INDI. This approach effectively handles fast-changing system dynamics commonly encountered in optical flow control, particularly height-dependent changes. We demonstrate the robustness and efficiency of the proposed control scheme in numerical simulations and also real-world flight tests: multiple landings of an MAV on a static and flat surface with various tracking setpoints, hovering and landings on moving and undulating surfaces. Despite being challenged with the presence of noisy optical flow estimates and the lateral and vertical movement of the landing surfaces, the MAV is able to successfully track or land on the surface with an exponential decay of both height and vertical velocity at almost the same time, as desired. Video demos can be found at <https://youtu.be/i2xFJNGhfxs>.

I. INTRODUCTION

Recent advancements in the development of small flying robots have opened up their potential for a wide range of indoor applications due to their ability to navigate in confined spaces and lower risk of harm when operating near users [1, 2]. Despite this, achieving full autonomy in these robots remains a challenge due to their limited payload capacities and computational resources.

Extensive research has been conducted to design solutions inspired by biological systems, such as tiny flying insects, which face similar limitations but are still capable of performing complex tasks. These insects heavily rely on visual information for navigation and control, and it is believed by biologists that their movements may be guided by optical flow cues [3, 4].

Therefore, the use of optical flow in Micro Air Vehicles (MAVs) has been a subject of growing interest in recent years due to its computational efficiency. The optical flow, which refers the apparent motion of features in a visual scene, can be used to provide rich information of the surroundings for tasks, such as hovering, landing, and obstacle avoidance [5, 6, 7]. The

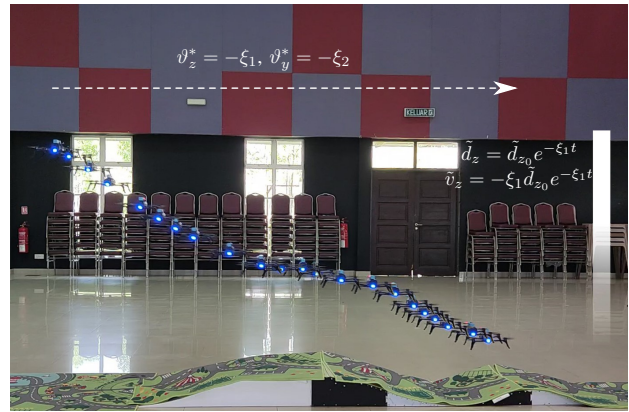


Fig. 1: Optical flow landing of an MAV on an undulating surface while moving laterally using our proposed INDI control scheme. The lateral movement of the MAV was deliberately generated by tracking a constant lateral flow ϑ_y^* to excite the challenging vertical movement of the landing surface as perceived by its bottom camera. The controller also tracked a constant flow divergence ϑ_z^* , resulting both relative height \tilde{d}_z and velocity \tilde{v}_z decay exponentially to zero simultaneously. All vision and control algorithms were run onboard the MAV.

low computational demands and ability to function in GPS-denied environments make optical flow methods particularly well-suited for use in MAVs.

However, the use of optical flow methods poses significant control problems. The optical flow observables are highly nonlinear, being represented as the ratios of the velocities to the height of the MAV, which makes the controller design more challenging. Additionally, the estimation of optical flow observables can be affected by various sources of noise and error, such as sensor noise, motion blur, and lighting changes, further exacerbating the control challenges.

Traditional control approaches, such as PID control and model-based Lyapunov methods [8, 9], which are commonly used in optical flow control for MAVs. These methods are typically designed for systems with linear dynamics and known models, making them sensitive to parameter uncertainties and

model errors. This can lead to poor performance and instability of the control system when using optical flow observables as its control input [10].

As an alternative, adaptive approaches, such as bio-inspired strategies and machine learning [11, 12], can be used to improve the performance of the control system in the presence of nonlinearities. These approaches are based on the idea of using feedback from the system to adapt the control strategy in real-time, making them more robust to unknown dynamics and nonlinearities. However, these methods can be complex to implement and require significant computational resources.

Nonlinear control approaches, such as PID scheduling [13, 14], which involves the tuning of multiple parameters, can also be employed to deal with nonlinear optical flow control. However, the implementation of these methods can be sophisticated, as the parameters need to be carefully tuned to achieve good performance. Furthermore, the use of PID scheduling in optical flow control for MAVs can be problematic as it relies heavily on the accurate knowledge of the system dynamics and parameters, which are often unknown or uncertain in real-world applications. Additionally, the performance of PID scheduling can be sensitive to parameter variations and model errors, which can lead to poor performance and instability of the control system in the presence of nonlinearities.

In contrast, the Incremental Nonlinear Dynamic Inversion (INDI) method has been shown to be a suitable approach for nonlinear systems with unknown models and nonlinear output [15]. INDI is a nonlinear control method that utilizes the inverse of the system dynamics to cancel out the nonlinearities present in the system. By using this inverse, INDI can provide a linear control input that can be used to stabilize the system. It is also robust to parameter uncertainties and model errors, making it an effective method for handling nonlinear optical flow control in MAVs. Furthermore, INDI is a computationally efficient method that can be implemented in real-time, making it a viable option for practical applications.

The current implementation of INDI utilizes a constant control effectiveness matrix G , which is determined by the system dynamics [16, 17]. However, when it comes to optical flow stabilization, landing, or tracking in MAVs, the system dynamics are dependent on the height of the vehicle above the ground. As the height of the vehicle changes, the dynamics of the system also change. Therefore, a constant G matrix is inappropriate for optical flow control in MAVs. To address this issue, it would be beneficial to develop an INDI method that uses a G matrix that changes over time. This would enable the control system to adapt to the changing dynamics of the system, resulting in improved performance and stability.

Therefore, the identification of the time-varying G matrix is essential for INDI-based optical flow control. However, the calculation of the G matrix is a computationally intensive task and can be challenging in real-time applications. To address this issue, we propose a novel INDI control scheme that directly identifies its inverse, i.e., $G^\dagger = G^{-1}$. In other words, it could reduce the computation required for the control increment, resulting in a significant reduction in computation time.

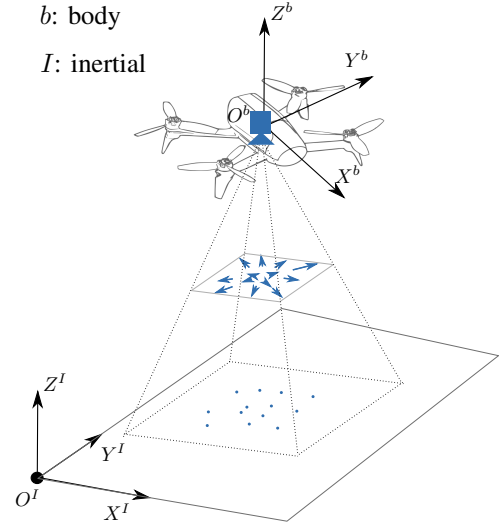


Fig. 2: MAV body ($O^b X^b Y^b Z^b$) and inertial ($O^I X^I Y^I Z^I$) reference frames. O^b is fixed at the center of gravity of the MAV, X^b points forward, Y^b is port, and Z^b points upward. The inertial reference frame is located on the ground and follows the North-East-Up ($X^I - Y^I - Z^I$) system.

Furthermore, this approach eliminates the error introduced by the inversion calculation and thus improves the accuracy of the control system.

Figure 1 illustrates one of the flight tests conducted in this paper using our proposed INDI control scheme, i.e., the optical flow landing of an MAV on an undulating surface while translating. We demonstrate the success of the tracking of the desired constant optical flow setpoints, i.e., ϑ_z^* and ϑ_y^* , with the INDI control, allowing the desired exponential decay of both relative height \tilde{d}_z and velocity \tilde{v}_z to zero in a nearly simultaneous manner while moving laterally. The lateral movement of the MAV was intentionally generated to stimulate the complex vertical movement of the landing surface observed by the onboard downward-facing camera to validate the robustness of the control.

II. PROBLEM FORMULATION

This section describes the dynamics of an MAV and its observed optical flow signal during the landing process onto a moving platform. The attitude of the MAV is stabilized through an inner control loop, which is designed to be accurate and responsive in its operation.

A. The position dynamics of an MAV

Figure 2 defines the body reference frame of an MAV ($O^b X^b Y^b Z^b$) and the inertial reference frame ($O^I X^I Y^I Z^I$). A camera is attached to the body of the MAV and looking downward. Thus, the camera reference frame ($O^c X^c Y^c Z^c$) is assumed to align with the body reference frame. In the rest of the paper, the variables are in the inertial reference frame unless indicated with the superscripts b and c only to avoid confusion.

The position \mathbf{d} and velocity \mathbf{v} of the MAV have the dynamics given by Newton's second law:

$$\dot{\mathbf{d}} = \mathbf{v}, \quad (1)$$

$$\dot{\mathbf{v}} = \mathbf{g} + \frac{1}{M}\mathbf{T}(T, \boldsymbol{\omega}) + \frac{1}{M}\boldsymbol{\delta}(\mathbf{v}, \boldsymbol{\epsilon}), \quad (2)$$

where \mathbf{g} is gravity, M is the mass, \mathbf{T} is the thrust generated by the rotors, and it is a function of the total thrust in Z_b direction T and the attitude in Euler angles $\boldsymbol{\omega} = [\theta, \phi, \psi]$. The term $\boldsymbol{\delta}$ represents the aerodynamic forces due to the velocity \mathbf{v} and other factors $\boldsymbol{\epsilon}$, such as wind, acting on the MAV.

In the tracking moving platform and landing tasks, we can assume that the changes in the yaw angle ψ is small, and we can align the initial MAV position so that the yaw angle ψ can be neglected. The position dynamics in Eq. (2) becomes

$$\dot{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{M} \begin{bmatrix} \cos \phi \sin \theta \\ -\sin \phi \\ \cos \phi \cos \theta \end{bmatrix} T + \frac{1}{M}\boldsymbol{\delta}(\mathbf{v}, \boldsymbol{\epsilon}). \quad (3)$$

The attitude of the MAV is controlled by an inner control loop, thus, the input of the position control is $\mathbf{u} = [\theta, \phi, T]^T$.

B. The observed optical flow signal

The camera attached to the MAV faces downward and perceives optical flow used for the landing on a moving platform. To simplify the system so that the underlying control principles can be explained more clearly, this paper focuses on the problem of tracking and landing on a flat platform with slow, translational degrees of freedom. Its position and velocity vectors are \mathbf{d}^r and \mathbf{v}^r , where the superscript r stands for the reference platform. During movements, the camera will capture the optical flow, which is the velocity of the features on the platform beneath the camera. It is proportional to the ratio of the relative velocity to the height above the platform $d_z - d_z^r$. In this configuration, the camera is able to capture the three-dimensional flow field of the features beneath it, constituting the output vector of this problem: $\mathbf{y} = [\vartheta_x, \vartheta_y, \vartheta_z]^T$.

The optical flow captured in the X - and Y -axes represents the lateral flow of features, which describes the velocity of the features in the image plane. These measurements can be formulated as the ratio of the relative lateral velocity to the height above the platform:

$$\vartheta_x = c_x \frac{v_x - v_x^r}{d_z - d_z^r}, \quad (4)$$

$$\vartheta_y = c_y \frac{v_y - v_y^r}{d_z - d_z^r}, \quad (5)$$

where c_x and c_y are unknown proportionality constants. These constants are intrinsic properties of the optical flow measurement system, which are determined by various factors, such as the resolution, the image acquisition rate, and the focal length of the camera. It can be seen from the above equations that the position of the MAV in the X - and Y -axes are not observable from the lateral flows, they still can be used in stabilizing the position by maintaining zero lateral flows or controlling the velocity with an appropriate constant.

When the MAV approaches the surface, the optical flow captured from the camera shows a divergent pattern. It corresponds to the flow divergence ϑ_z , which represents the velocity of the features in the image plane along the line of sight and can be formulated as

$$\vartheta_z = c_z \frac{v_z - v_z^r}{d_z - d_z^r}, \quad (6)$$

where c_z is an unknown positive constant. When the MAV is approaching the platform, $v_z - v_z^r$ is negative, $d_z - d_z^r$ is always positive, and thus, ϑ_z will be negative. The constant flow divergence approach [15] has been used to control the vertical dynamics of MAVs by keeping the flow divergence a negative constant for landing, i.e., $\vartheta_z = \vartheta_z^*$. As a result, the time response of the distance $d_z - d_z^r$ will decrease exponentially to zero.

III. DATA-DRIVEN INCREMENTAL NONLINEAR DYNAMIC INVERSION

As depicted in the previous section, the system dynamics in Eq. (3) is input non-affine, the model parameters are unknown, and the full states are not directly measurable. Therefore, this paper proposes to use a data-driven INDI method to control the MAV to track and land on a moving platform.

A. Reformulation of the optical flow tracking control problems

The INDI method can be derived from the system as in Eqs. (1, 3-6) or in a more general form for nonlinear systems:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (7)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \quad (8)$$

where $\mathbf{x} \in \mathcal{R}^n$ is the system state, $\mathbf{u} \in \mathcal{R}^m$ is the system input, $\mathbf{y} \in \mathcal{R}^p$ is the system output or observations, $\mathbf{f} : \mathcal{R}^{n+m} \rightarrow \mathcal{R}^n$ and $\mathbf{h} : \mathcal{R}^n \rightarrow \mathcal{R}^p$ are smooth vector fields. If $p < m$ or $p > m$, it becomes an over-determined or under-determined control problem, and control allocation techniques or weighted least square methods can be used to tackle these problems, respectively. The rest of the paper assumes that the numbers of inputs and outputs are equal, $p = m$, for both theoretical derivations and practical applications.

In optical flow tracking problems, the observation involves the dynamics of the system and the platform as indicated in Eqs. (4-6). Thus, the state space needs to be augmented to incorporate both the states of the system and the referred platform, i.e., $[\mathbf{d}^T, \mathbf{v}^T, \mathbf{d}^{rT}, \mathbf{v}^{rT}]^T$. In biological systems, relative measurements are often utilized in navigation and control. For instance, many animals rely on relative motion cues in their visual system to track moving targets and navigate their environment. Similarly, the formulation of this control task focuses on the relative position and velocity $[(\mathbf{d} - \mathbf{d}^r)^T, (\mathbf{v} - \mathbf{v}^r)^T]^T$, which reduces the dimensionality of the problem. In addition, the relative position in the X - and Y -axes are not observable as mentioned in the previous section. The state of interest can be further reduced and defined as $\mathbf{x} = [\tilde{d}_z, \tilde{v}_x, \tilde{v}_y, \tilde{v}_z]^T = [d_z - d_z^r, v_x - v_x^r, v_y - v_y^r, v_z - v_z^r]^T$.

Thus, the dynamics of the relative position can be reformulated as follows:

$$\dot{\tilde{d}}_z = \tilde{v}_z, \quad (9)$$

$$\dot{\tilde{v}}_x = \frac{1}{M} \cos \phi \sin \theta T + \frac{1}{M} \delta_x - \dot{v}_x^r, \quad (10)$$

$$\dot{\tilde{v}}_y = -\frac{1}{M} \sin \phi T + \frac{1}{M} \delta_y - \dot{v}_y^r, \quad (11)$$

$$\dot{\tilde{v}}_z = \frac{1}{M} \cos \phi \cos \theta T - g + \frac{1}{M} \delta_z - \dot{v}_z^r. \quad (12)$$

B. Incremental nonlinear dynamic inversion

Since the observation \mathbf{y} is nonlinear in the state \mathbf{x} , the first step is to find the input-output relation by differentiating the output until the input appears explicitly:

$$\begin{aligned} \dot{\vartheta}_x &= c_x \left[\frac{\dot{\tilde{v}}_x}{\tilde{d}_z} - \frac{\tilde{v}_x \tilde{v}_z}{\tilde{d}_z^2} \right] \\ &= c_x \left[\frac{(\cos \phi \sin \theta T + \delta_x - M \dot{v}_x^r)}{M \tilde{d}_z} - \vartheta_x \vartheta_z \right], \end{aligned} \quad (13)$$

$$\begin{aligned} \dot{\vartheta}_y &= c_y \left[\frac{\dot{\tilde{v}}_y}{\tilde{d}_z} - \frac{\tilde{v}_y \tilde{v}_z}{\tilde{d}_z^2} \right] \\ &= c_y \left[\frac{(-\sin \phi T + \delta_y - M \dot{v}_y^r)}{M \tilde{d}_z} - \vartheta_y \vartheta_z \right], \end{aligned} \quad (14)$$

$$\begin{aligned} \dot{\vartheta}_z &= c_z \left[\frac{\dot{\tilde{v}}_z}{\tilde{d}_z} - \frac{\tilde{v}_z^2}{\tilde{d}_z^2} \right] \\ &= c_z \left[\frac{(\cos \phi \cos \theta T - Mg + \delta_z - M \dot{v}_z^r)}{M \tilde{d}_z} - \vartheta_z^2 \right]. \end{aligned} \quad (15)$$

The above input-output relation can be written as

$$\dot{\mathbf{y}} = \boldsymbol{\alpha}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}, \dot{\mathbf{v}}^r), \quad (16)$$

which is nonlinear and nonaffine in the input. Now, taking the first-order Taylor series expansion results in:

$$\begin{aligned} \dot{\mathbf{y}} &\approx \dot{\mathbf{y}}_0 + \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}, \dot{\mathbf{v}}^r)}{\partial \mathbf{x}} \Big|_{\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\delta}_0, \dot{\mathbf{v}}_0^r} (\mathbf{x} - \mathbf{x}_0) \\ &\quad + \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}, \dot{\mathbf{v}}^r)}{\partial \mathbf{u}} \Big|_{\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\delta}_0, \dot{\mathbf{v}}_0^r} (\mathbf{u} - \mathbf{u}_0) \\ &\quad + \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}, \dot{\mathbf{v}}^r)}{\partial \boldsymbol{\delta}} \Big|_{\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\delta}_0, \dot{\mathbf{v}}_0^r} (\boldsymbol{\delta} - \boldsymbol{\delta}_0) \\ &\quad + \frac{\partial \boldsymbol{\alpha}(\mathbf{x}, \mathbf{u}, \boldsymbol{\delta}, \dot{\mathbf{v}}^r)}{\partial \dot{\mathbf{v}}^r} \Big|_{\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\delta}_0, \dot{\mathbf{v}}_0^r} (\dot{\mathbf{v}}^r - \dot{\mathbf{v}}_0^r), \end{aligned} \quad (17)$$

where the second or higher-order terms have been neglected in this equation.

This equation can be further simplified with a time-scale separation assumption. This assumption holds when the change in the control input is considered significantly faster than the change in the system state with high sampling frequency and instantaneous control effects [18]. Note that in optical flow control problems, this assumption holds only when $1/\tilde{d}_z$ is bounded [15]. This assumption is put into practice by setting a threshold for the height \tilde{d}_z , below which the system will shut down. In addition, it is assumed that the aerodynamic forces $\boldsymbol{\delta}$ and platform dynamics are slower than the system dynamics. This allows the controller to focus on

identifying the direct input-output relationship and generate the control input more efficiently. With these assumptions, Eq. (17) can be further simplified to

$$\dot{\mathbf{y}} \approx \dot{\mathbf{y}}_0 + G(\mathbf{x}_0, \mathbf{u}_0)(\mathbf{u} - \mathbf{u}_0), \quad (18)$$

where $\mathbf{u} = [\theta, \phi, T]^T$ and

$$G(\mathbf{x}, \mathbf{u}) = \frac{1}{M \tilde{d}_z} \begin{bmatrix} c_x \cos \phi \cos \theta T & -c_x \sin \phi \sin \theta T & c_x \cos \phi \sin \theta \\ 0 & -c_y \cos \phi T & -c_y \sin \phi \\ -c_z \cos \phi \sin \theta T & -c_z \sin \phi \cos \theta T & c_z \cos \phi \cos \theta \end{bmatrix}. \quad (19)$$

To solve the above nonlinear control problem, the virtual control input is introduced to replace the differentiated output $\dot{\mathbf{y}} = \boldsymbol{\nu}$, and the control increment can be found by inverting Eq. (18):

$$\mathbf{u} - \mathbf{u}_0 = G(\mathbf{x}_0, \mathbf{u}_0)^{-1}(\boldsymbol{\nu} - \dot{\mathbf{y}}_0). \quad (20)$$

This matrix G is a nonlinear function of Euler angles, ϕ and θ , the total thrust T , and unknown parameters c_x , c_y , and c_z . It also varies nonlinearly with the distance above the platform \tilde{d}_z , which is not directly measurable with the optical flow sensors. Thus, it is very difficult to measure all the relevant variables and calculate matrix G from Eq. (19) at every time step. Another option to find the control increment is to identify G online using least square techniques from input and output data with the time sequence, which can be rewritten as

$$\dot{\mathbf{y}}_k - \dot{\mathbf{y}}_{k-1} \approx G_{k-1} \cdot (\mathbf{u}_k - \mathbf{u}_{k-1}). \quad (21)$$

In this case, the explicit representation of the time-varying matrix G in Eq. (19) is unnecessary, but it needs to be identified and inverted at every time step.

C. Inverted G matrix identification

To reduce the computational load of this data-driven approach, this paper proposes to directly identify the inverted G matrix, which is denoted as $G^\dagger = G^{-1}$, from input and output data with time sequence $t_k, t_{k-1}, t_{k-2}, \dots$ as follows:

$$\mathbf{u}_k - \mathbf{u}_{k-1} \approx G_{k-1}^\dagger \cdot (\dot{\mathbf{y}}_k - \dot{\mathbf{y}}_{k-1}). \quad (22)$$

The parameters in matrix $G^\dagger \in \mathcal{R}^{m \times p}$ can be identified using the Recursive Least Square (RLS) approach. The control command can be calculated directly with the identified matrix \hat{G}^\dagger :

$$\mathbf{u} = \mathbf{u}_k + \hat{G}_{k-1}^\dagger \cdot (\boldsymbol{\nu}_k - \dot{\mathbf{y}}_k). \quad (23)$$

The virtual input $\boldsymbol{\nu}_k$ at the current time t_k can be designed as a feedback linear control with constant proportional, integral, and derivative feedback gains:

$$\boldsymbol{\nu}_k = -K_p(\mathbf{y}_k - \boldsymbol{\vartheta}^*) - K_i \int_0^t (\mathbf{y}_k - \boldsymbol{\vartheta}^*) dt - K_d \dot{\mathbf{y}}_k, \quad (24)$$

where $\boldsymbol{\vartheta}^*$ is a constant vector of the desired optical flow. In the tracking platform task, we will have $\boldsymbol{\vartheta}^* = \mathbf{0}$, and in the landing on a moving platform task, we will have

$\vartheta^* = [0 \ 0 \ \vartheta_z^*]^T$. The desired flow divergence ϑ_z^* for landing is a negative constant, and by following the constant flow divergence, the MAV will have a soft land on the platform with the time response [15]:

$$\tilde{d}_z(t) = \tilde{d}_{z_0} e^{\frac{\vartheta_z^*}{c_z} t}, \quad (25)$$

where \tilde{d}_{z_0} is the initial height above the landing surface.

In the data-based INDI methods, where the matrix G is unknown and time-varying, the control command can be calculated by first identifying G and then calculating its inversion in the traditional way, or directly identifying the inverse matrix G^\dagger using our proposed algorithm. The direct identification of G^\dagger has a lower time complexity and computation load compared to the traditional method. This is because the inversion step, which requires $\mathcal{O}(p^3)$ computational effort, is no longer necessary. Therefore, the direct identification of G^\dagger results in a more efficient INDI control implementation.

The proposed method of directly identifying the inverse matrix G^\dagger also has advantages in terms of accuracy and stability, especially for optical flow control. The matrix G increases to very large values and changes very quickly when the vehicle is approaching the platform, as it is proportional to $1/\tilde{d}_z$. This can result in difficulties for the control system to accurately estimate the matrix G . On the other hand, the inverse matrix G^\dagger tends to converge as the vehicle approaches the platform, which makes the updating process more straightforward and precise. Furthermore, by eliminating the inversion step, the proposed method reduces the chances of numerical errors and improves the stability of the control system, especially when dealing with the possibility of ill-conditioned matrices. In conclusion, the direct identification of the inverse matrix G^\dagger offers both computational efficiency and improved accuracy and stability of the data-driven INDI approach.

IV. NUMERICAL SIMULATIONS

In this section, the feasibility of the proposed optical flow control method will be demonstrated through three-dimensional tracking control simulations. This will be achieved by comparing the INDI control with both 1) the conventional method that identifies the G matrix and calculates its inverse at every time step and 2) the proposed method that directly identifies the inverse matrix G^\dagger . The simulations are performed with an MAV having a mass of $M = 1.2 \text{ kg}$, proportionality constants c_x, c_y, c_z equal to 1, and an initial height of 3 m . For a fair comparison, the initial parameters and the INDI outer loop implementations are the same. The outer loop virtual input is designed using a linear feedback control with a constant proportional gain $K_p = 1$. All these simulations are conducted in the MATLAB environment on a computer with 1.8 GHz CPU and 40 GB of RAM.

Figure 3 presents the performance of the INDI control for an MAV landing on a moving platform with a constant flow divergence setpoint $\vartheta_z^* = -0.1 \text{ rad/s}$. The reference platform undergoes a three-dimensional translational movement in a sinusoidal form as shown by the dashed line in the first row. In this simulation, the matrix G is identified using the RLS

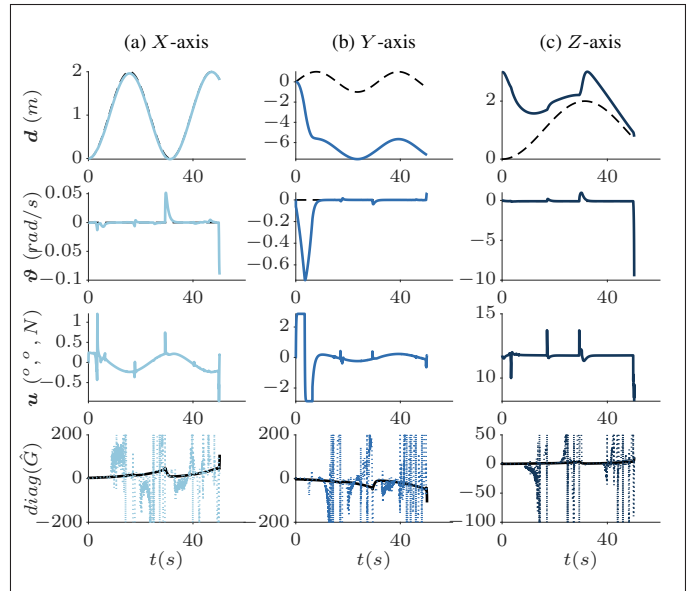


Fig. 3: The MAV lands on a moving platform with INDI control identifying and inverting the G matrix at each step. Each column presents the positions of the MAV d (the dashed line represents the position of the platform d^r), optical flows ϑ (the dashed line represents the tracked setpoint $\vartheta^* = [0, 0, -0.1]$), control inputs $u = [\theta, \phi, T]^T$, and diagonal elements of the identified matrix \hat{G} (the solid line represents the true value) in the (a) X-axis, (b) Y-axis, and (c) Z-axis.

technique, and its inverse is calculated at every time step to generate the control inputs according to Eq. (23). As shown in Eq. (19), the non-diagonal elements of the matrix G are small because the pitch angle θ and roll angle ϕ are small, thus the figures only depict the diagonal elements of the identified matrix \hat{G} . The true values of the matrix G diagonal elements, which correspond to the motion in the X-, Y-, and Z-axes, are calculated from Eq. (19) and presented in solid lines.

It can be seen that, as the MAV approaches the platform, the observed optical flow is highly sensitive to even minor input variations, and the diagonal elements of the matrix G increase rapidly. In this case, it is essential to update the estimated G matrix at a sufficient rate to adapt the control law to ensure a smooth landing. Failure to do so will result in degraded control performance. Therefore, in this optical flow control task, especially when the height \tilde{d}_z is small, the conventional method of identifying and inverting the matrix G is sensitive to the choice of the forgetting factor γ in the RLS approach. To facilitate rapid parameter updates, the simulation employs a value of $\gamma = 0.8$. This choice, however, may lead to oscillations in the estimation. Conversely, large values of γ will impede the update rate of matrix G , resulting in even worse performance. Another practical approach is to set a higher threshold for the height \tilde{d}_z and shut down the system when it falls below the threshold. This approach, however, requires additional sensing and operations.

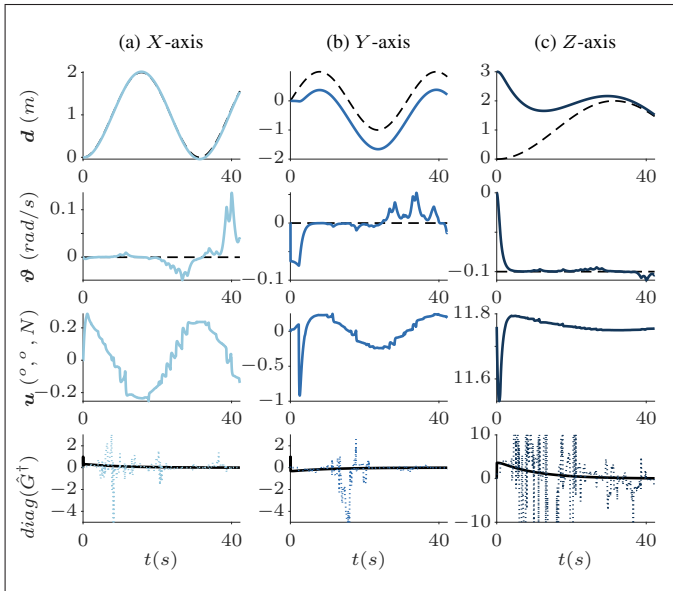


Fig. 4: The MAV lands on a moving platform with INDI control identifying the inverse matrix G^\dagger directly. Each column presents the positions of the MAV d (the dashed line represents the position of the platform d^*), optical flows ϑ (the dashed line represents the tracked setpoint $\vartheta^* = [0, 0, -0.1]$), control inputs $u = [\theta, \phi, T]^T$, and diagonal elements of the inverse matrix \hat{G}^\dagger (the solid line represents the true value) in the (a) X-axis, (b) Y-axis, and (c) Z-axis.

On the other hand, the inverse matrix G^\dagger , which is proportional to the height \tilde{d}_z , converges as the MAV approaches the landing platform as depicted in Figure 4. When the height \tilde{d}_z is small, the directly identified G^\dagger decreases correspondingly to small values and generates small control increments to facilitate a smooth landing of the MAV. In this scenario, the forgetting factor γ may be increased to achieve a more stable parameter update than what is illustrated in the figure. Although both the matrix G and its inverse matrix G^\dagger is time-varying, the direct identification of G^\dagger will be less sensitive to the choice of the forgetting factor in the RLS identification process and the choice of the threshold for the height \tilde{d}_z . Furthermore, the computational efficiency of the proposed INDI control is demonstrated by a decrease in the CPU time required to run a 50-second-simulation from 0.1062 s in the traditional INDI control to 0.0681 s in the proposed INDI control (averaged over 20 simulations).

V. FLIGHT TESTS

This section presents the experimental validation of the proposed INDI control scheme implemented on an MAV. Several flight tests were performed in an indoor environment under various challenging conditions: 1) landings with different control parameters on a static and flat landing surface, and 2) hovering and landings over moving and undulating landing surfaces. The experimental setup and the results of all the flight tests are presented and discussed in this section.

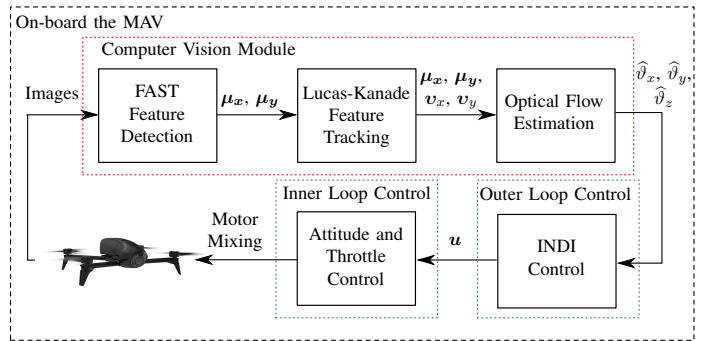


Fig. 5: Flight test setup using an MAV (Parrot Bebop 2) with an open source autopilot (Paparazzi). All the computer vision and control algorithms run onboard the MAV. In the vision module, FAST detects corners (μ_x, μ_y) in each image captured by the downward camera. Then, Lucas-Kanade tracks these corners in the next image and computes the flow vectors (v_x, v_y) . Lastly, optical flow in the X^b -, Y^b -, Z^b -axes $(\hat{\vartheta})$ as defined in this paper are estimated. In the outer loop control module, the INDI controller receives $\hat{\vartheta}$ as the input and commands the attitude and throttle of the MAV (u).

A. Flight Test Setup

The experimental setup uses a Parrot Bebop 2 MAV with the Paparazzi autopilot software¹, as illustrated in Figure 5. It is equipped with standard sensors, including an inertia measurement unit, a sonar sensor, and two cameras (front and bottom). The bottom camera (facing downward) is of particular interest for hovering and landing purposes. The sonar sensor was enabled for logging height during the tests to evaluate the landing performance.

In the autopilot software, two modules were specifically created for the flight tests: the computer vision and the outer loop control, as depicted in Figure 5. The computer vision module processes the images captured from the camera during the flight tests and implements computer vision algorithms, including the Features from Accelerated Segment Test (FAST) algorithm [19] for feature detection and the Lucas-Kanade tracker [20] for feature tracking.

By taking the average of the flow vectors in the X^b - and Y^b -axes (v_x, v_y) obtained from the feature tracking, optical flow \hat{v}_x and \hat{v}_y can be computed. To estimate flow divergence \hat{v}_z , the approach in [10] utilizing Eq. (26) is employed. This involves computing the image distances of every two features between consecutive images κ , and then computing the average ratio of the change in the image distances to its previous image distance with a known time interval Δt , resulting in the estimation of flow divergence \hat{v}_z as follows:

$$\hat{v}_z = \frac{1}{M} \cdot \frac{1}{\Delta t} \sum_{i=1}^M \left[\frac{\kappa_{(t-\Delta t),i} - \kappa_{t,i}}{\kappa_{(t-\Delta t),i}} \right], \quad (26)$$

¹Paparazzi Autopilot: <http://wiki.paparazziuav.org>

where M is the total number of tracked corners. A low-pass filter was introduced to minimize the estimation noise.

The outer loop control module, which implements the INDI control scheme detailed in Section III, receives output from the computer vision module and controls the attitudes and throttle of the MAV. The flight tests were started with manual take-off to a specified position and then switched to automatic mode, where only the required modules executed the designated tests. The height measurement from the sonar sensor was recorded for performance validation and analysis purposes, but not utilized in the control process.

B. Static and flat landing surface

Figure 6 presents multiple landing tests of the MAV on a static and flat landing surface, in which three different setpoints of flow divergence, i.e., -0.1 rad/s , -0.2 rad/s , and -0.3 rad/s , are tracked using the proposed INDI control. As the setpoint value increases, the landing velocity also increases, making the tracking task more demanding for the controller as it must efficiently accommodate the rapidly changing velocity of the MAV.

The results depicted in Figure 6 clearly demonstrate the ability of the controller to accurately track the designated setpoints. It also exhibits a quick response to even larger setpoints, indicating its robustness and efficiency. As a result, for all landings, both height and vertical velocity of the MAV decay exponentially to zero at almost the same time, as desired. Since this control implementation is designed specifically for vertical landings, the inverse matrix \hat{G}^\dagger contains only a single element. It is noteworthy that the identified \hat{G}^\dagger exhibits an exponential decay and convergence, analogous to the change in height. This observation supports the validity of the theoretical derivation of G^\dagger , which is a function of height.

C. Moving and undulating landing surface

The results of the hovering and landing tests conducted on a dynamically changing landing surface, characterized by wave-like fluctuations, are depicted in Figure 7. These flight tests were carried out to verify the efficiency and robustness of the proposed INDI control scheme in the presence of lateral and vertical movements of the landing surface.

The first test, whose results are presented in Figure 7 (a), involved hovering of the MAV over a surface undergoing lateral movement by utilizing the controller to regulate the flow divergence ($\vartheta_z^* = 0 \text{ rad/s}$) and optical flow in the Y^b -axis ($\vartheta_y^* = 0 \text{ rad/s}$). The results in this figure show that both setpoints are well tracked using the INDI controller even for a laterally moving surface. This can be observed from the nearly constant height and unchanged position of the MAV with respect to the surface feature shown in the stacked image.

Besides the regulation task, Figure 7 (b) depicts the results of a flight test on a surface undergoing lateral movement during landing, rather than hovering. The controller was tasked with tracking a constant flow divergence setpoint, i.e., $\vartheta_z^* = -0.2 \text{ rad/s}$ (a median value of the setpoints was selected from the tests performed in Section V-B) for landing

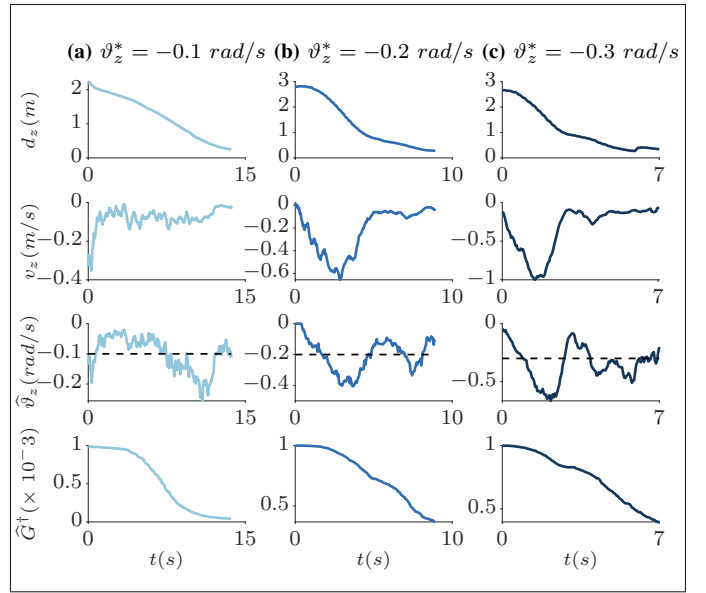


Fig. 6: MAV Landings on a static and flat ground by tracking different flow divergence setpoints: (a) $\vartheta_z^* = -0.1 \text{ rad/s}$, (b) $\vartheta_z^* = -0.2 \text{ rad/s}$, (c) $\vartheta_z^* = -0.3 \text{ rad/s}$. Each column presents the MAV height d_z , vertical velocity v_z , flow divergence $\hat{\vartheta}_z$, and inverse matrix \hat{G}^\dagger for the respective flight test. All the tracked setpoints (ϑ_z^*) are plotted as dashed lines.

and maintaining the regulation of the optical flow in the Y^b -axis ($\vartheta_y^* = 0 \text{ rad/s}$) for following the moving surface. The results indicate that the controller was adapting the changes at the beginning, as the estimated $\hat{\vartheta}_z$ shows a slower response causing a small fluctuation of height at $t \approx 4 \text{ s}$. Eventually, the controller is able to accurately track the setpoint ϑ_z^* and land the MAV with an exponential decay of the height d_z .

In order to evaluate the capabilities of the controller under more challenging conditions, the last flight test stimulated the vertical movement of the landing surface by translating the MAV over an undulating terrain. The lateral movement of the MAV was produced by tracking a setpoint of optical flow in the Y^b -axis ($\vartheta_y^* = -0.8 \text{ rad/s}$). To demonstrate that the MAV can safely land on the uneven terrain, a flow divergence setpoint $\vartheta_z^* = -0.1 \text{ rad/s}$ was chosen. Since the sonar measurement only provides the height relative to the landing surface, an ultra-wideband (UWB) tag was attached to the MAV together with four UWB anchors located at four corners of the test area to measure the absolute height of the MAV using Double-Sided Two-Way Ranging [21]. The absolute height allows us to analyze at which instance the MAV changing its velocity or even stopping. The results show that the controller adapted well to the translation over an undulating terrain as the tracking of the flow divergence ϑ_z^* is accurate leading to the exponential decay of the relative height, as desired. Additionally, the result from the absolute height shows that the MAV was descending in a slow pace when it encountered the first peak ($t \approx 4 \text{ s}$). This is due to

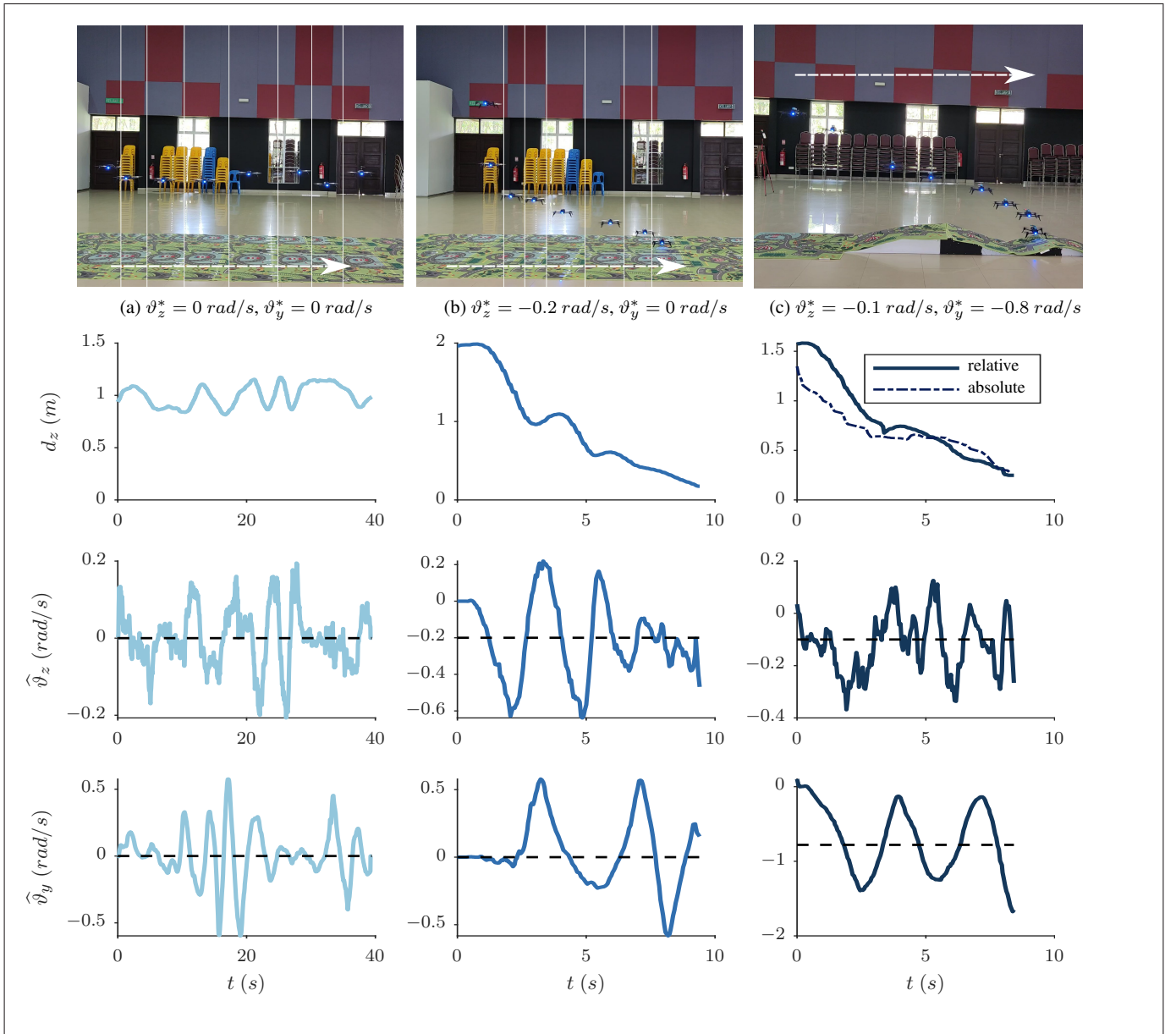


Fig. 7: MAV Hovering and Landings on a moving and undulating ground. The columns (a) and (b) depict the hovering (by tracking $\vartheta_z^* = 0 \text{ rad/s}$, $\vartheta_y^* = 0 \text{ rad/s}$) and landing (by tracking $\vartheta_z^* = -0.2 \text{ rad/s}$, $\vartheta_y^* = 0 \text{ rad/s}$) results of the MAV, respectively, above a moving and flat surface, while the column (c) displays the landing (by tracking $\vartheta_z^* = -0.1 \text{ rad/s}$) results of the MAV during translation (by tracking $\vartheta_y^* = -0.8 \text{ rad/s}$) over an undulating ground. Each column shows the stacked images of the flight scene, MAV height d_z , flow divergence $\hat{\vartheta}_z$, and optical flow in the Y^b -axis $\hat{\vartheta}_y$ for the respective flight test. All the tracked setpoints (ϑ_z^* , ϑ_y^*) are plotted as dashed lines.

the fact that the MAV detected its movement closer to the landing surface at that instance. After it moved over the first peak, the MAV continued to descend and land safely on the next peak of the undulating surface while translating.

VI. CONCLUSION

This paper formulated a novel data-driven Incremental Nonlinear Dynamic Inversion (INDI) method to address nonlinear optical flow control problems with rapidly changing

and height-dependent system output dynamics. The control effectiveness matrix G in INDI for optical flow control was found to be time-varying with the distance above the platform, which is not directly measurable. Consequently, an online identification of G from input and output data, followed by an inversion, at every time step was deemed necessary to calculate the control increments. To this end, the direct identification of the inverse matrix G^\dagger was proposed in order to eliminate

the inversion step, thus reducing the chances of numerical errors and enhancing the accuracy and stability of the data-driven INDI approach. The proposed method was applied and compared with the conventional method in numerical simulations for controlling an MAV during landing on a moving platform. It showed an improved control performance and also computational efficiency. The proposed method was also validated through real-world flight tests, successfully demonstrating the ability to track and land on various surfaces with desired height and velocity decay. Future work will focus on 1) extending the incremental model to deal with measurement noises and delays, scaleless system outputs, or fast dynamics, and 2) integrating this efficient optical flow control method with a guidance strategy from other visual cues for precise navigation of small flying robots in obstacle-dense environments.

ACKNOWLEDGMENTS

This study is sponsored by Malaysian Ministry of Higher Education with the Fundamental Research Grant Scheme (FRGS) [grant number FRGS/1/2020/TK0/USM/03/11].

REFERENCES

- [1] Enrica Soria. Swarms of flying robots in unknown environments. *Science Robotics*, 7(66):eabq2215, 2022.
- [2] David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance¹. *The International Journal of Robotics Research*, 39(2-3):250–265, 2020.
- [3] Z Jane Wang, James Melfi Jr, and Anthony Leonardo. Recovery mechanisms in the dragonfly righting reflex. *Science*, 376(6594):754–758, 2022.
- [4] Thomas S Collett. Insect vision: controlling actions through optic flow. *Current Biology*, 12(18):R615–R617, 2002.
- [5] Zhitao Yu, Gioele Zardini, Andrea Censi, and Sawyer Fuller. Visual confined-space navigation using an efficient learned bilinear optic flow approximation for insect-scale robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4250–4256. IEEE, 2022.
- [6] Florian Mahlknecht, Daniel Gehrig, Jeremy Nash, Friedrich M Rockenbauer, Benjamin Morrell, Jeff De-laune, and Davide Scaramuzza. Exploring event camera-based odometry for planetary robots. *IEEE Robotics and Automation Letters*, 7(4):8651–8658, 2022.
- [7] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaz9712, 2020.
- [8] Franck Ruffier and Nicolas Franceschini. Optic flow regulation in unsteady environments: A tethered mav achieves terrain following and targeted landing over a moving platform. *Journal of Intelligent & Robotic Systems*, 79:275–293, 2015.
- [9] Bruno Herissé, Tarek Hamel, Robert Mahony, and François-Xavier Russotto. Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on robotics*, 28(1):77–89, 2011.
- [10] H W Ho, G C H E de Croon, E van Kampen, QP Chu, and Max Mulder. Adaptive gain control strategy for constant optical flow divergence landing. *IEEE Transactions on Robotics*, 34(2):508–516, 2018.
- [11] Guido CHE De Croon, Julien JG Dupeyroux, Christophe De Wagter, Abhishek Chatterjee, Diana A Olejnik, and Franck Ruffier. Accommodating unobservability to control flight attitude with optic flow. *Nature*, 610(7932):485–490, 2022.
- [12] Michael O’Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022.
- [13] Farid Kendoul. Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems. *The International Journal of Robotics Research*, 33(2):237–267, 2014.
- [14] Andrea Cesetti, Emanuele Frontoni, Adriano Mancini, Primo Zingaretti, and Sauro Longhi. A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of intelligent and robotic systems*, 57:233–257, 2010.
- [15] Ye Zhou, Hann Woei Ho, and Qiping Chu. Extended incremental nonlinear dynamic inversion for optical flow control of micro air vehicles. *Aerospace Science and Technology*, 116:106889, 2021.
- [16] Rasmus Steffensen, Agnes Steinert, and Ewoud JJ Smeur. Nonlinear dynamic inversion with actuator dynamics: An incremental control perspective. *Journal of Guidance, Control, and Dynamics*, pages 1–9, 2022.
- [17] Ewoud JJ Smeur, Guido CHE de Croon, and Qiping Chu. Cascaded incremental nonlinear dynamic inversion for mav disturbance rejection. *Control Engineering Practice*, 73:79–90, 2018.
- [18] Ronald van’t Veld, Erik-Jan Van Kampen, and Qiping Chu. Stability and robustness analysis and improvements for incremental nonlinear dynamic inversion control. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 1127, 2018.
- [19] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision—ECCV 2006*, pages 430–443. Springer, 2006.
- [20] JY Bouquet. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.
- [21] Poh Ling Ching, Shu Chuan Tan, and Hann Woei Ho. Ultra-wideband localization and deep-learning-based plant monitoring using micro air vehicles. *Journal of Aerospace Information Systems*, 19(11):717–728, 2022.