








Visual Servoing NMPC Applied to UAVs for Photovoltaic Array Inspection

Edison Velasco-Sánchez , Luis F. Recalde , Bryan S. Guevara , José Varela-Aldás , Francisco A. Candelas , Santiago T. Puente , and Daniel C. Gandolfo 

I. INTRODUCTION

Abstract—The photovoltaic (PV) industry is seeing a significant shift toward large-scale solar plants, where traditional inspection methods have proven to be time-consuming and costly. Currently, the predominant approach to PV inspection using unmanned aerial vehicles (UAVs) is based on the capture and detailed analysis of aerial images (photogrammetry). However, the photogrammetry approach presents limitations, such as an increased amount of useless data and potential issues related to image resolution that negatively impact the detection process during high-altitude flights. In this work, we develop a visual servoing control system with dynamic compensation using nonlinear model predictive control (NMPC) applied to a UAV. This system is capable of accurately tracking the middle of the underlying PV array at various frontal velocities and height constraints, ensuring the acquisition of detailed images during low-altitude flights. The visual servoing controller is based on extracting features using RGB-D images and employing a Kalman filter to estimate the edges of the PV arrays. Furthermore, this work demonstrates the proposal in both simulated and real-world environments using the commercial aerial vehicle (DJI Matrice 100), with the purpose of showcasing the results of the architecture.

Index Terms—Aerial systems: Perception and autonomy, optimization and optimal control, visual servoing.

NOWADAYS, society is facing global energy challenges and the search for alternative sources to fossil fuels [1]. Renewable energies (RE), such as wind, hydro and solar, are abundant and are not at risk of depletion [2], [3]. The PV sector has grown rapidly and has become the primary source of renewable energy, given the abundance and accessibility of solar power [4], [5]. However, PV installations are limited by changing environments, where elements such as sand, wind, and dust significantly reduce the efficiency of PV modules [6]. In addition, this type of system requires periodic maintenance due to the need for optimal conditions to function properly [7]. Many studies have shown that a large number of PV modules present maintenance-related damage [8], which can even reach 80% of the total installed modules, where the most common problem is strictly related to surface defects such as mismatches, cracks, discolorations, snail trails, and soiling [9], [10], [11], [12].

In recent years, PV modules have been inspected using different techniques. One method is *Electroluminescence-based inspection*, involving the application of direct current to the module of PV array and the measurement of the resulting photoemission [13]. Another approach is *I-V measurements*, where I-V curves are generated using information from PV modules, but the exact location of the failure is not provided [14]. *Infrared thermography* utilizes thermal cameras to capture radiation and convert it into thermogram images [15]. Furthermore, *Visual inspection* by experts offers a simple method for identifying potential defects in PV modules [9]. However, visual inspection can be time-consuming, making it more suitable for small plants; on a large scale, it is almost impractical.

Unmanned aerial vehicle (UAV) technology has experienced accelerated development. It has made a significant contribution to monitoring applications such as oil and pipeline inspection [16], power transmission lines [17], and precision agriculture [18]. This technology has recently been applied to PV inspection, saving valuable time and human resources, as well as ensuring rapid data collection over large areas [19], [20], [21], [22], [23]. Currently, most of the UAV applications in PV inspection are based on photogrammetry, where UAVs are equipped with a Global Positioning System (GPS) receiver and an Inertial Measurement Unit (IMU) [24]. These elements ensure that the system flies at high altitude from the ground, guaranteeing proper horizontal and vertical overlap of the images for orthomosaic reconstruction [25], [26].

Nevertheless, the photogrammetry approach has two limitations. First, GPS accuracy can lead to incorrect placement of the UAV, resulting in an increased amount of useless data during flights. Second, flights at high altitude from the ground can

Manuscript received 1 October 2023; accepted 13 January 2024. Date of publication 31 January 2024; date of current version 9 February 2024. This letter was recommended for publication by Associate Editor M. W. Mueller and Editor G. Loianno upon evaluation of the reviewers' comments. This work was supported in part by the Ministry of Science and Innovation of the Spanish Government under the research Project PID2021-122685OB100 and in part by the training of research Ph.D. staff under Grant PRE2019-088069. (*Edison Velasco-Sánchez, Luis F. Recalde, and Bryan S. Guevara contributed equally to this work.*) (*Corresponding author: Edison Velasco-Sánchez.*)

Edison Velasco-Sánchez, Francisco A. Candelas, and Santiago T. Puente are with the Group of Automation, Robotics and Computer Vision (AUROVA), University of Alicante, 03690 Alicante, Spain (e-mail: edison.velasco@ua.es; francisco.candelas@ua.es; santiago.puente@ua.es).

Luis F. Recalde is with the Instituto de Automática (INAUT), Universidad Nacional de San Juan-CONICET, San Juan J5400, Argentina, and also with the Centro de Investigaciones de Ciencias Humanas y de la Educación (CICHE), Universidad Tecnológica Indoamérica, Ambato 180101, Ecuador (e-mail: fernandorecalde@uti.edu.ec).

Bryan S. Guevara and Daniel C. Gandolfo are with the Instituto de Automática (INAUT), Universidad Nacional de San Juan-CONICET, San Juan J5400, Argentina (e-mail: bguevara@inaut.unsj.edu.ar; dgandolfo@inaut.unsj.edu.ar).

José Varela-Aldás is with the Centro de Investigaciones de Ciencias Humanas y de la Educación (CICHE), Universidad Tecnológica Indoamérica, Ambato 180101, Ecuador (e-mail: josevarela@uti.edu.ec).

Our approach is available to the scientific community in: https://github.com/EPVelasco/VisualServoing_NMPC.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3360876>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3360876

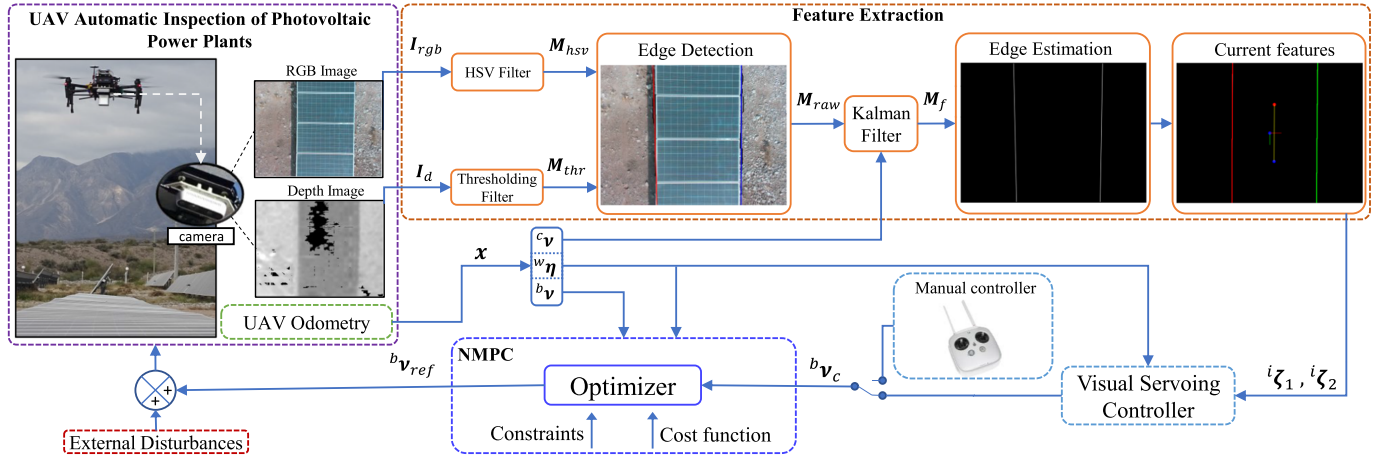


Fig. 1. Pipeline of the system implemented for overflight and inspection of the PV arrays with the visual servoing controller combined with dynamic compensation and constraints based on NMPC using UAVs. The purple area shows the UAV system in conjunction with the test environment. Visual servo controllers and NMPC are explained in Algorithms 1 and 2, respectively. The orange area shows the Feature Extraction from the RGB-D images and is explained in Algorithm 3).

generate issues related to image resolution, negatively affecting the surface defects detection process. The results presented in [19] have shown an intrinsic relationship between the ground sampling distance and the detection of surface defects, such as soil, snail trails, and white spots. Thus, resolution is a key factor in achieving accurate detection of surface failures. Therefore, we find it interesting to focus our research here, making it an important contribution to the field of PV inspections.

In this paper, we propose an approach to ensure proper image acquisition of PV modules based on autonomous UAVs, which is capable of capturing details on the surface of the PV array without geo-referenced measurements or path planning methods. The objective of this work is to develop a visual servoing control system combined with a Nonlinear Model Predictive Control (NMPC) capable of accurately tracking the center of the underlying PV array at low-altitude. This system ensures the acquisition of images for later use in the inspection process by experts or AI-based architectures. Fig. 1 shows the pipeline of our approach.

In summary, our contributions are the following.

- A visual servoing control system for aerial vehicles combined with dynamic compensation and constraints based on NMPC, which ensures accurate tracking of the middle of the PV arrays considering different frontal velocities and height constraints without geo-referenced points or path planning methods.
- Lightweight and low run-time PV panel feature extraction method based on RGB-D images and a Kalman filter to mitigate the impact of noise and uncertainties.
- End-to-end validation of the complete system using a commercial aerial vehicle (DJI Matrice 100), both in a simulated environment and in real-world conditions.

II. METHODOLOGY

A. Notation

Throughout the article, we establish the following definitions: an inertia frame denoted as $\mathcal{W} = \{O_{\mathcal{W}}, W_x, W_y, W_z\}$, and the body frame denoted by $\mathcal{B} = \{O_{\mathcal{B}}, B_x, B_y, B_z\}$ located at the

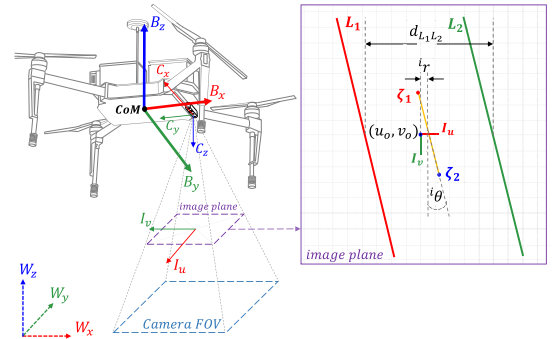


Fig. 2. Reference coordinate frames of the implemented system.

UAV's center of mass (CoM). Furthermore, the camera frame can be defined as $\mathcal{C} = \{O_{\mathcal{C}}, C_x, C_y, C_z\}$, where $O_{\mathcal{C}}$ aligns with the optical center of the camera and with the 2D digital image plane defined as $\mathcal{I} = \{O_{\mathcal{I}}, I_u, I_v\}$ (see Fig. 2). Moreover, this work uses the symbol ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{C}}$ to denote a rotational matrix from the frame \mathcal{C} to \mathcal{B} . Vectors are represented by bold quantities as follows: ${}^w\mathbf{p} \in \mathbb{R}^n$, using the frame \mathcal{W} as a prefix; matrices are indicated in bold capital letters as $\mathbf{P} \in \mathbb{R}^{n \times m}$. Finally, the Q-weighted norm can be formulated as $\|\mathbf{p}\|_{\mathbf{Q}} = \sqrt{\mathbf{p}^T \mathbf{Q} \mathbf{p}}$

B. System Modeling

This section presents the formulation of motion equations considering the UAV and the camera. These equations will serve as the foundation for the visual servoing NMPC formulation applied to the aerial vehicle. Additionally, image feature extraction is carefully formulated to ensure accurate extraction, leveraging the combination of RGB-D information with a Kalman filter.

1) *Quadrotor Differential Kinematics*: This work formulates the equations of motion, considering that the aerial system (DJI Matrice 100) includes an autopilot that ensures tracking of the reference velocities with respect to the frame \mathcal{B} . This consideration takes into account that the roll angle $\phi \approx 0$ and pitch angle $\theta \approx 0$ are approximately zero. Therefore, the location of

the camera ${}^w\eta$ can be formulated as follows:

$$\begin{bmatrix} {}^w x \\ {}^w y \\ {}^w z \\ {}^w \psi \end{bmatrix} = \begin{bmatrix} {}^w x_b + {}^b x_c \cos(\psi) - {}^b y_c \sin(\psi) \\ {}^w y_b + {}^b x_c \sin(\psi) + {}^b y_c \cos(\psi) \\ {}^w z_b + {}^b z_c \\ \psi \end{bmatrix} \quad (1)$$

where $({}^w x_b, {}^w y_b, {}^w z_b)$ is the location of frame \mathcal{B} respect to the frame \mathcal{W} , $({}^b x_c, {}^b y_c, {}^b z_c)$ is the camera's position respect to the frame \mathcal{B} ; finally, the rotational angle with respect to axis W_z can be expressed as ψ (for more details, see the reference coordinate frames in Fig. 2).

Taking it into account and based on [27], the differential kinematics can be formulated as:

$${}^w \dot{\eta} = \mathbf{J}_b(\psi) {}^b \nu \quad (2)$$

where $\mathbf{J}_b \in \mathbb{R}^{4 \times 4}$ is the Jacobian matrix that allows the liner mapping of the UAV's velocities ${}^b \nu = [{}^b v_x \ {}^b v_y \ {}^b v_z \ {}^b \omega_z]^T \in \mathbb{R}^4$ considering that the angular velocities ${}^b \omega_x$ and ${}^b \omega_y$ are approximately zero due to the autopilot of the system.

2) *Quadrotor Dynamics*: Taking the presented autopilot into account and based on [28], the dynamic model can be formulated as follows:

$${}^b \nu_{ref} = \mathbf{M}({}^b \nu, \pi) {}^b \dot{\nu} + \mathbf{C}({}^b \nu, \pi) {}^b \nu \quad (3)$$

where $\mathbf{M} \in \mathbb{R}^{4 \times 4}$ and $\mathbf{C} \in \mathbb{R}^{4 \times 4}$ represent the matrices for inertial and coriolis forces and ${}^b \nu_{ref}$ is the vector of desired velocities of the UAV. Additionally, $\pi \in \mathbb{R}^{18}$ is a vector used to characterize the dynamics of the system.

3) *Image Features*: The pinhole camera model is considered in this work [29], where the feature point ${}^c \zeta = [{}^c x \ {}^c y \ {}^c z]^T \in \mathbb{R}^3$ expressed in \mathcal{C} can be projected on the digital image plane as follows:

$$\begin{bmatrix} {}^i u \\ {}^i v \end{bmatrix} = \frac{\lambda}{c_z} \begin{bmatrix} {}^c x \\ {}^c y \end{bmatrix} \quad (4)$$

where ${}^i \zeta = [{}^i u \ {}^i v]^T \in \mathbb{R}^2$ is the feature vector projected into the digital image plane \mathcal{I} and λ is the camera focal length.

Our work considers a fixed feature point ${}^i \zeta = \mathbf{0}$; therefore, the temporal variation in the image plane can be expressed in relation to the velocities of the UAV, resulting in the following expression:

$${}^i \dot{\zeta} = \mathbf{J}_f({}^i \zeta, c_z) \mathbf{T}^{-1} {}^b \nu, \mathbf{T} = \begin{bmatrix} {}^B \mathbf{R}_C & [{}^b \mathbf{t}_c]_{\times} {}^B \mathbf{R}_C \\ \mathbf{0}_{3 \times 3} & {}^B \mathbf{R}_C \end{bmatrix} \quad (5)$$

where ${}^b \nu = [{}^b v \ {}^b \omega]^T \in \mathbb{R}^6$ is the stacked vector of the UAV's velocities without the consideration of the autopilot, $[{}^b \mathbf{t}_c]_{\times}$ is a skew symmetric matrix associated to ${}^b \mathbf{t}_c = [{}^b x_c \ {}^b y_c \ {}^b z_c]^T \in \mathbb{R}^3$. Finally, considering two features vectors ${}^i \zeta$, the Jacobian image matrix is $\mathbf{J}_f \in \mathbb{R}^{4 \times 6}$.

4) *Line Features*: This section presents the formulation of the line features expressed by two points that characterize the center of the PV array. These points are projected on the digital image plane as ${}^i \zeta_1 = [{}^i u_1 \ {}^i v_1]^T$ and ${}^i \zeta_2 = [{}^i u_2 \ {}^i v_2]^T$ presented in Fig. 2 (more information related to the extraction of the two points can be found in Section II-D). This work further

formulates these two points in polar coordinates as:

$$\begin{bmatrix} {}^i r \\ {}^i \theta \end{bmatrix} = \begin{bmatrix} \sin({}^i \theta) ({}^i u_1 - \frac{{}^i u_2 - {}^i u_1}{{}^i v_2 - {}^i v_1} {}^i v_1) \\ \arctan(-\frac{{}^i v_2 - {}^i v_1}{{}^i u_2 - {}^i u_1}) \end{bmatrix} \quad (6)$$

where ${}^i \xi = [{}^i r \ {}^i \theta]^T \in \mathbb{R}^2$ is the line features vector, which characterize the middle line of the lateral edges of the PV panels respect to the frame \mathcal{I} .

The time derivative of (6) can be formulated as a function of the camera velocities by combining (5), defining the following formulation:

$${}^i \dot{\xi} = \mathbf{J}({}^i \xi, {}^i \zeta, c_z) {}^b \nu \quad (7)$$

$$\mathbf{J} = \mathbf{J}_l({}^i \xi, {}^i \zeta) \mathbf{J}_f({}^i \zeta, c_z) \mathbf{T}^{-1}$$

where, $\mathbf{J}_l \in \mathbb{R}^{2 \times 4}$ is the Jacobian matrix associated with the line features, and $\mathbf{J} \in \mathbb{R}^{2 \times 6}$ is a compact Jacobian, which includes \mathbf{J}_l and \mathbf{J}_f for simple notation. The formulation (7) is suitable for developing the visual servoing controller as presented in Section II-C, without considering the dynamics of the UAV.

C. Visual Servoing Control System Combined With NMPC

1) *Visual Servoing Controller*: The visual servoing controller proposed in this work aims to guide the UAV in a way that the feature points of the line ${}^i \xi$ move towards the desired values ${}^i \xi_d = [{}^i r_d \ {}^i \theta_d]^T$, ensuring the movement of the UAV over the middle of the PV arrays. Taking this into account, the control error vector can be formulated as ${}^i \tilde{\xi} = {}^i \xi_d - \alpha({}^i \xi)$; however, due to $|{}^i r| \gg |{}^i \theta|$, the operator $\alpha({}^i r, {}^i \theta) = [{}^i r / {}^i r_{\max} \ {}^i \theta]^T$ ensures that the values are of the same magnitude order, considering that ${}^i r_{\max}$ represents half of the image width. Due to the dimensions of the Jacobian matrix $\mathbf{J} \in \mathbb{R}^{2 \times 6}$, there is no unique solution, and it is possible to use the null space projector operator. Furthermore, when considering the nearly perfect tracking velocity ${}^b \nu \equiv {}^b \nu_c$, the control law can be formulated as:

$${}^b \nu_c = \mathbf{J}^\dagger ({}^i \dot{\xi}_d + \mathbf{K} {}^i \tilde{\xi}) + (\mathbf{I}_{6 \times 6} - \mathbf{J}^\dagger \mathbf{J}) {}^b \nu_d \quad (8)$$

note that $\mathbf{J}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1}$ is the Moore-Penrose pseudoinverse, (\mathbf{W}, \mathbf{K}) are positive definite matrices and $(\mathbf{I}_{6 \times 6} - \mathbf{J}^\dagger \mathbf{J})$ is the orthogonal projection of ${}^b \nu_d$.

The vector ${}^b \nu_d$ contains arbitrary values for the associated velocities of the UAV in the frame \mathcal{B} with ${}^b \nu_d = [{}^b v_{xd} \ {}^b v_{yd} \ {}^b v_{zd} \ {}^b \omega_{xd} \ {}^b \omega_{yd} \ {}^b \omega_{zd}]^T$. The appropriate selection of \mathbf{W} can guarantee angular velocities ${}^b \omega_{xd} \approx 0$ and ${}^b \omega_{yd} \approx 0$, making the control proposal compatible with the aerial system (DJI Matrice 100). The arbitrarily long vector can be formulated in order to guarantee any frontal velocity and height over the PV array, which can be written as follows:

$$\begin{aligned} {}^b v_{xd} &= \frac{{}^b v_x^{\max}}{1 + k_1 \|{}^i \tilde{\xi}\|} \\ {}^b v_{zd} &= \dot{\eta}_{zd} + k_2 \tanh({}^w \eta_{zd} - {}^w \eta_z) \end{aligned} \quad (9)$$

where ${}^b v_x^{\max}$ is the maximum desired frontal velocity, (k_1, k_2) are positive scalar values and ${}^w \eta_{zd}$ represents the desired height of the UAV over the PV solar panels. In our application, $\dot{\eta}_{zd} = 0$ and ${}^i \xi_d = 0$, due to the fact that the flights over the PV arrays are maintained at a constant height and the desired line features are constant over time.

Algorithm 1: Visual-Servoing.

Data: ${}^i\xi_d, {}^i\dot{\xi}_d, {}^i\zeta_1, {}^i\zeta_2, {}^w\eta$
Result: ${}^b\nu_c$ Control value

- 1: ${}^i\xi \leftarrow {}^i\zeta_1, {}^i\zeta_2$; // Image Features
- 2: ${}^i\xi = {}^i\xi_d - \alpha({}^i\xi)$; // Control Error
- 3: ${}^b\nu_{xd} = {}^b\nu_x^{\max} / (1 + k_1 \|{}^i\dot{\xi}\|)$; // Desired frontal Velocity
- 4: ${}^b\nu_{zd} = \dot{\eta}_{zd} + k_2 \tanh({}^w\eta_{zd} - {}^w\eta_z)$; // Desired upper Velocity
- 5: ${}^b\nu_d \leftarrow [{}^b\nu_{xd}, {}^b\nu_{zd}]$; // Arbitrary velocity vector
- 6: $\mathbf{J} \leftarrow [{}^i\xi, {}^i\zeta_1, {}^i\zeta_2, \mathbf{T}]$; // Compact Jacobian
- 7: ${}^b\nu_c = \mathbf{J}^\dagger ({}^i\dot{\xi}_d + \mathbf{K}^i\xi) + (\mathbf{I}_{6 \times 6} - \mathbf{J}^\dagger \mathbf{J}) {}^b\nu_d$; // Control Law

Algorithm 1 explains the procedure for develop the visual servoing controller.

The proposal (8) does not consider the dynamics of the system and lacks information about constraints related to maximum velocities or the workspace within which the system can ensure adequate feature extraction. Consequently, there is a need to formulate an NMPC that incorporates these constraints.

2) *Dynamic Compensation and Constraints With NMPC:* This section introduces a NMPC approach to ensure a nearly perfect tracking velocity of the control actions generated by the visual servoing controller ${}^b\nu \equiv {}^b\nu_c$. This NMPC is designed to compute the control policy while incorporating the dynamics of the system and the restricted operational space as constraints in a finite-time optimal control problem (OCP). Thus, this work utilizes a discrete version of equations (2) and (3). These equations were formulated using the fourth-order Runge-Kutta integrator, considering dt as the sampling time, defined as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), dt) \quad (10)$$

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{J}_b(\psi) {}^b\nu \\ -(\mathbf{M}^{-1}\mathbf{C}) {}^b\nu + \mathbf{M}^{-1} {}^b\nu_{ref} \end{bmatrix}$$

where $\mathbf{x} = [{}^w\eta \quad {}^b\nu]^T \in \mathbb{R}^8$ and $\mathbf{u} = [{}^b\nu_{ref}]^T \in \mathbb{R}^4$ are the states and control actions of the system.

The intermediate cost function related to velocity tracking performance $l_v(\cdot) : \mathbb{R}^4 \rightarrow \mathbb{R}_{\geq 0}$ can be formulated as follows:

$$l_v(\mathbf{x}_k, {}^b\nu_{c_k}) = \|{}^b\nu_{c_k} - \mathbf{C}_v \mathbf{x}_k\|_{\mathbf{Q}} \quad (11)$$

where ${}^b\nu_c$ is the control action generated by the visual servoing controller (8), \mathbf{C}_v is a matrix designed to obtain the velocities of the aerial system. Finally, \mathbf{Q} is a positive definite matrix. Moreover, to ensure smooth control actions, the cost function associated with the control policy $l_u(\cdot) : \mathbb{R}^4 \rightarrow \mathbb{R}_{\geq 0}$ can be structured: $l_u(\mathbf{u}_k) = \|\mathbf{u}_k\|_{\mathbf{R}}$, where \mathbf{R} is a positive definite matrix.

In addition to the equations of motion for the aerial system, constraints related to control actions and workspace can be formulated as $\mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max}$, $\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}$. $(\mathbf{x}_{\min}, \mathbf{x}_{\max}, \mathbf{u}_{\min}, \mathbf{u}_{\max})$ represents the minimum and maximum states and control actions of the aerial system. These constraints ensure proper velocities and height compliance, providing the acquisition of detailed images and adequate feature extraction during low-altitude flights over the PV modules. Therefore, the cost functions presented previously, it is possible

Algorithm 2: Visual-Servoing + NMPC.

Data: ${}^i\xi_d, {}^w\eta, {}^b\nu$ Desired line features and Internal states of the system

- 1 **while** *Inspection is not complete* **do**
- 2 $\mathbf{x} \leftarrow {}^w\eta, {}^b\nu$; // Get internal states of the UAV
- 3 ${}^i\zeta_1, {}^i\zeta_2 \leftarrow$ Image features Extraction;
- 4 ${}^b\nu_c \leftarrow$ Visual-Servoing (${}^i\xi_d, {}^i\zeta_1, {}^i\zeta_2, {}^w\eta$);
- 5 ${}^b\nu_{ref} \leftarrow$ NMPC ($\mathbf{x}, {}^b\nu, {}^b\nu_c$); // ACADOS Optimizer
- 6 APPLY(${}^b\nu_{ref}$) to the UAV
- 7 **end**

to generate the NMPC as a finite-time horizon optimization problem as follows:

$$\begin{aligned} \arg \min_{\substack{\mathbf{x}_1 \dots \mathbf{x}_N \\ \mathbf{u}_1 \dots \mathbf{u}_{N-1}}} & l_v(\mathbf{x}_N, {}^b\nu_{c_N}) + \sum_{k=1}^{N-1} l_v(\mathbf{x}_k, {}^b\nu_{c_k}) + l_u(\mathbf{u}_k) \\ \text{subject to: } & \mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{x}_k, \mathbf{u}_k), k = 1, \dots, N-1 \\ & \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}, k = 1, \dots, N-1 \\ & \mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max}, k = 1, \dots, N. \end{aligned} \quad (12)$$

NMPC is solved using Sequential Quadratic Programming (SQP) in a real-time iteration (RTI) scheme using ACADOS [30] and CasADi [31]. The Algorithm 2 comprehensively explains the Visual Servoing controller combined with the NMPC formulation.

D. Image Features Extraction

We propose a feature extraction of the PV array by approximating to two straight lines the lateral edges of the photovoltaic panels. By identifying the lateral edges of the PV array, we extract the deviation with respect to the center of the image (${}^i r$) and the angle of inclination of the parallel lines generated by these edges (${}^i \theta$). We prolong the line with the average of the slopes of L_1 and L_2 using the midpoint (see image plane in Fig. 2). Once the new slope is calculated, we determine the u, v components of the points ${}^i\zeta_1$ and ${}^i\zeta_2$ as follows:

$$\begin{aligned} {}^i\zeta_{u1,2} &= (u_o \pm p_f \cos({}^i\theta)) \\ {}^i\zeta_{v1,2} &= (v_o \pm p_f \sin({}^i\theta)) \end{aligned} \quad (13)$$

where p_f is a factor that depends on the horizontal distance $d_{L_1 L_2}$, and the desired distance at which the camera wants to capture the lines L_1 and L_2 of the PV array.

1) *Line Detection:* To detect the PV array's vertical lines, RGB images undergo HSV filtering, highlighting white edges, while depth images are thresholded based on camera-to-PV and camera-to-ground distances. Thus, the RGB and D images (I_{rgb}, I_d) are binarized, noise-filtered with morphological operations, and merged. Then, we generate a M_{raw} mask that approximates straight lines using OpenCV's *fitline* algorithm.

We employ a Kalman filter to estimate the states s_k derived from the *fitline* algorithm, influenced by camera velocities ${}^c\nu$. The matrices \mathbf{A} and \mathbf{B} are determined by the dynamic mode decomposition (DMD) methodology [32]. We perform a manual flight over the PV array, varying UAV velocities, altitudes, and attitude. This yields the data set for states s_k and system inputs ${}^c\nu$ used in the estimation of \mathbf{A} and \mathbf{B} .

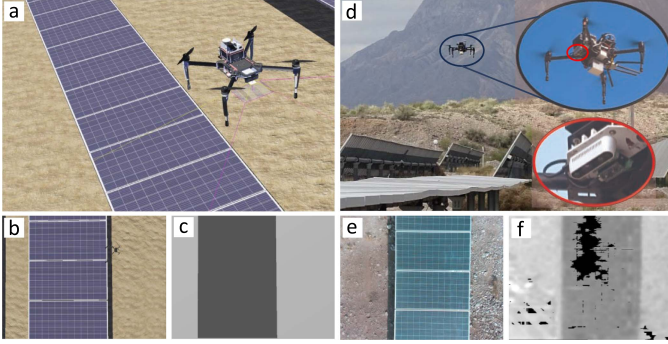


Fig. 3. UAV flying over the PV arrays. The top images (a) and (d) represent the simulated and real experimental environment, respectively. The bottom images (b), (c) and (d), (f) are the color and depth images of the PV array obtained by simulation as well as by the UAV on-board camera.

Algorithm 3: Image Feature Extraction.

```

Data:  $c\nu$ ,  $I_{rgb}$  and  $I_d$  image
Result:  ${}^i\zeta_1, {}^i\zeta_2$  Feature extraction
1 Load the RGB and Depth image;
2 while ( $I_{rgb}$  &  $I_d$ )  $\neq \emptyset$  do
3    $M_{hsv} \leftarrow I_{rgb} \rightarrow HSV_{filter}$ ;
4    $M_{thr} \leftarrow I_d \rightarrow Thresholding_{filter}$ ;
5    $M_{aux} \leftarrow M_{hsv} \cup M_{thr}$ ;
6    $M_{raw} \leftarrow M_{aux} \rightarrow Morphological_{filter}$ ;
7   Vertical contours extraction Loop
8     ( $contour_{left}, contour_{right}$ )  $\leftarrow M_{raw}$ ;
9   end Loop  $x_k \leftarrow filine(contour_{left}, contour_{right})$ ;
10  // Kalman Filter
11   $s_{k+1} = \mathbf{A}s_k + \mathbf{B}c\nu$ ;
12   $\hat{\mathbf{P}}_k = \mathbf{A}\hat{\mathbf{P}}_k\mathbf{A}^T + \mathbf{Q}_k$ ;
13   $\mathbf{K}_k = \mathbf{P}_{k+1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k+1}\mathbf{H}^T + \mathbf{R})^{-1}$ ;
14   $\hat{s}_k = s_{k+1} + \mathbf{K}_k(s_k - \mathbf{H}s_{k+1})$ ; // Estimation
15   $\mathbf{P}_k = (\mathbf{I}_{4 \times 4} - \mathbf{K}_k\mathbf{H})\hat{\mathbf{P}}_k$ ;
16   $M_f \leftarrow \hat{s}_k$ ; // New mask filtered
17  ${}^i\zeta_1, {}^i\zeta_2 \leftarrow M_f$ ; // Feature extraction of Mask filtered
end

```

Algorithm 3 describes the procedure developed for feature extraction, taking RGB-D images I_{rgb} , I_d and camera velocities $c\nu$ as inputs. The output is the M_f mask representing the current features based on the estimated parameters of the lines \hat{s}_k , which correspond to the edges of the PV array.

III. EXPERIMENTS AND RESULTS

This section showcases our method's performance in simulated and real experiments. We outline the experimental setups for the simulation environment and the real platform, including gain weight settings and the controller used. The controller gains were tuned through experimentation, initially in the simulator and later fine-tuned in real experiments.

A. Experimental Setup

1) *Simulation Experiments:* We utilized Webots [33] to simulate PV arrays and DJI Assistant to ensure a similar behavior of the UAV Matrice 100 (see Fig. 3). Webots includes a simulated environment with a RGB-D camera mounted on the UAV. Both simulators communicate via topics using ROS Noetic, allowing integration of the flight dynamics in DJI Assistant with

TABLE I
VISUAL SERVOING AND NMPC PARAMETERS

Parameters	Values	Parameters	Values
\mathbf{K}	$diag(150 \ 0.5)$	\mathbf{W}	$diag(1 \ 1 \ 1 \ 50 \ 50 \ 1)$
k_1	10	k_2	10
\mathbf{Q}	$diag(1 \ 1 \ 1 \ 1)$	\mathbf{R}	$diag(1.4 \ 1.4 \ 1.4 \ 1.4)$
\mathbf{x}_{min}	$(-\infty \ \infty \ 0 \ \pi \ 2 \ 2 \ 2 \ 1.5)$	\mathbf{x}_{max}	$(\infty \ \infty \ 4.5 \ \pi \ 2 \ 2 \ 2 \ 1.5)$
\mathbf{u}_{min}	$(-2 \ 2 \ 2 \ 1.5)$	\mathbf{u}_{max}	$(2 \ 2 \ 2 \ 1.5)$
i_{r_d}	0	i_{θ_d}	0 [rad]
N	10	dt	0.05 [s]

the Webots simulation environment. Furthermore, we considered noise in sensors, with the following Gaussian distribution $w_{\eta_n} \sim \mathcal{N}(-0.005, 0.005)$ and $b_{\nu_n} \sim \mathcal{N}(-0.001, 0.001)$. The simulators were implemented on a computer with an AMD Ryzen 7 3700x 8-core processor, 16 GB of RAM and a NVIDIA GTX 1060 video card with 6 GB of memory, using the Ubuntu 20.04 operating system.

2) *Real Experiments:* We used a DJI Matrice 100 drone in combination with a Jetson Orin-NX onboard computer, equipped with 16 GB RAM and running the Ubuntu 20.04 Focal operating system with Jetpack 5.1.1. As RGB-D sensor, we used an Intel RealSense™ D435i camera. The real experiments were carried out at the Anchipurac Photovoltaic Solar Park, located in the city of San Juan, Argentina (Fig. 3). In this environment, we used a PV array of 80 solar panels with individual dimensions of 2.0×1.0 [m].

In the simulation and real experiments, we used images with a resolution of 640×480 at 30 *fps* and the PV arrays were placed horizontally so that the aerial images fully displayed each solar panel. Furthermore, Table I shows the values of all parameters explained in Section II.

B. Simulation Experiment Results

The first experiment aimed to determine the optimal matrix \mathbf{W} values for the visual servoing controller (8), crucial for the proposed controller. Simulating a scenario where the UAV camera maintains a direct line of sight to the photovoltaic array, we ensured ideal feature extraction with specified values in the null space projector: $b_{v_x}^{\max} = 1$ [m/s] and $w_{\eta_{zd}} = 3$ [m] over the PV arrays.

An inappropriate matrix \mathbf{W} generates control values that do not guarantee $b_{\omega_{xd}} \approx 0$ and $b_{\omega_{yd}} \approx 0$, rendering the visual servoing proposal unsuitable for the aerial platform (DJI Matrice 100). Furthermore, the movement of the aerial vehicle generated by the control law ensures that the error converges to zero ${}^i\tilde{\xi} = [{}^i\tilde{r} \quad {}^i\tilde{\theta}] \rightarrow \mathbf{0}$ as $t \rightarrow \infty$, with a stabilization time of 30 [s]. Finally, the null space projector ensures $b_{v_x} \rightarrow b_{v_{xd}}$ as $t \rightarrow \infty$, with a stabilization time of 30 [s] (see results in Fig. 4(a)).

On the other hand, an appropriate matrix \mathbf{W} can notoriously improve the controller performance. These values ensure $b_{\omega_{xd}} \approx 0$ and $b_{\omega_{yd}} \approx 0$, guaranteeing that the control error vector converges to zero ${}^i\tilde{\xi} = [{}^i\tilde{r} \quad {}^i\tilde{\theta}] \rightarrow \mathbf{0}$ as $t \rightarrow \infty$, with a stabilization time of 7 [s]. Finally, the null space projector also ensures that $b_{v_x} \rightarrow b_{v_{xd}}$ as $t \rightarrow \infty$ with a stabilization time of 7 [s] (see results in Fig. 4(b)).

The second experiment, shown in Fig. 5, compares visual servoing (8) with the visual servoing controller combined with

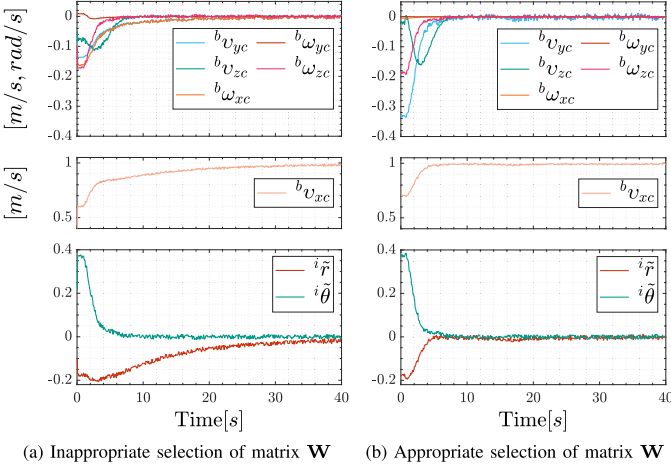


Fig. 4. Simulation results of the visual servoing controller with different values of matrix \mathbf{W} .

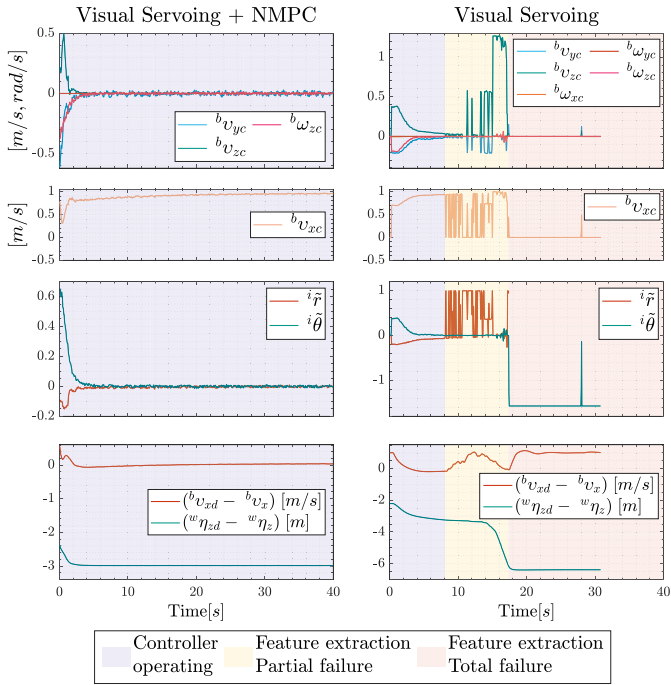


Fig. 5. Comparative results of the visual servoing controller versus the visual servoing controller combined with NMPC in a simulated environment.

NMPC (12), considering the following values in the null space operator: $b_{v_x}^{\max} = 1$ [m/s] and $w_{\eta_{zd}} = 5$ [m]. The limitations of the classical visual servoing controller restrict its ability to define the UAV's workspace. For instance, when a loss of visual features occurs due to a flight altitude outside the workspace, the camera loses line of sight with the visual patterns, decreasing the image feature extraction, and causing controller failures. However, due to the constraints of the NMPC, the altitude of the UAV is limited to 4.5 [m] above the photovoltaic array (see Table I), the control proposal ensures proper extraction and detailed images, guaranteeing that ${}^i\tilde{\xi} = [{}^i\tilde{r} \quad {}^i\tilde{\theta}] \rightarrow \mathbf{0}$ and $b_{v_x} \rightarrow b_{v_{xd}}$ as $t \rightarrow \infty$, with w_{η_z} within the constraints of the workspace in the NMPC.

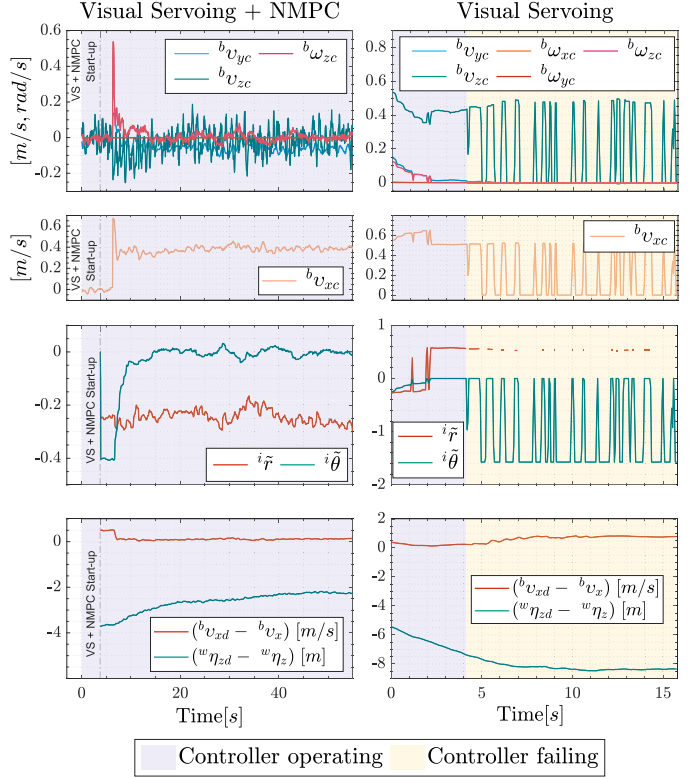


Fig. 6. Comparative results of the visual servoing controller versus the visual servoing controller combined with NMPC in real-world environments.

In Fig. 5 the yellow background presents failures in feature extraction, which cause problems in the visual servoing controller; the pink background represents a total failure of image feature extraction when the controller is not working. Finally, the purple background represents the time when the controller is operating correctly with correct image feature extraction, where visual servoing combined with the NMPC exhibits proper behavior during the whole experiment.

C. Real-World Experiment Results

Several experiments were conducted to demonstrate the performance of both proposal at different heights over PV arrays. The null space operators, $b_{v_x}^{\max} = 0.5$ [m/s] and $w_{\eta_{zd}} = 5$ [m], over the PV arrays are the same as those presented in the second simulation experiments.

Due to the fact that the visual servoing controller does not include the dynamics of the aerial vehicle and the constraints where the image extraction features work properly, this formulation exhibits poor behavior resulting in the controller failure during the experiment (see Fig. 6).

On the other hand, visual servoing combined with the NMPC formulation exhibits proper behavior of the aerial vehicle, (see Fig. 6), guaranteeing movement along the PV arrays in the real-world application. However, the control error vector exhibits bounded values, with $|{}^i\tilde{r}| < 0.3$ and $|{}^i\tilde{\theta}| < 0.08$. This behavior is attributed to potential unmodeled dynamics, disturbances during the experiment, the precision of depth camera measurements at this height, and incorrect gain tuning. Additionally, the

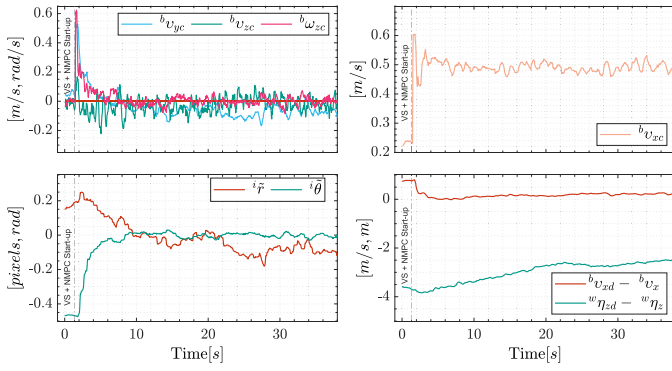


Fig. 7. Results of visual servoing controller combined with NMPC with appropriate gain \mathbf{K} to decrease the $i_{\tilde{r}}$ translation error.

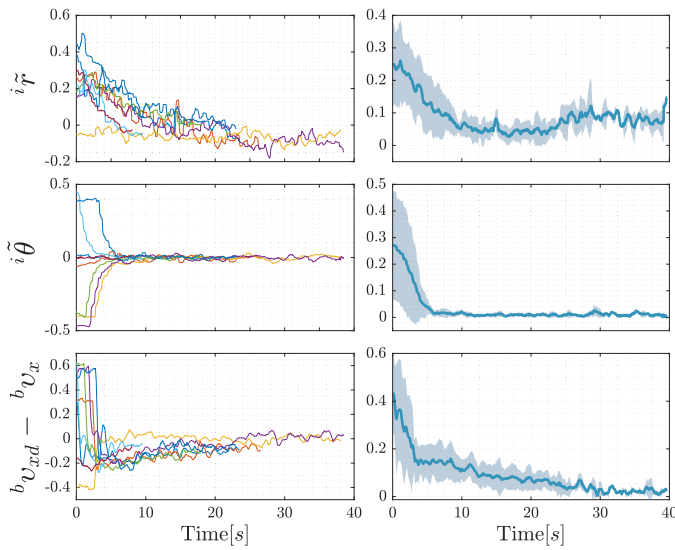


Fig. 8. $i_{\tilde{r}}$, $i_{\tilde{\theta}}$, and $b_{\tilde{v}_x}$ errors of eight experiments of the visual servoing controller combined with the NMPC controller in real world environments. The right plots show the mean and standard deviation of the experiments.

proposal is able to guarantee $b_{v_x} \rightarrow b_{v_{xd}}$ with w_{η_z} inside the workspace.

Given the performance demonstrated in the experimental results presented above, the authors adjusted the gain values, setting $\mathbf{K} = \text{diag}(300, 0.8)$. Simultaneously, modifications were applied to the null space operator, with $b_{v_x}^{\max} = 0.5$ [m/s] and $w_{\eta_{zd}} = 3$ [m]. We have chosen to set the inspection height at 3 [m] as it is deemed adequate. This height ensures the minimum distance within the camera's field of view while maintaining a safe flying altitude. Following these changes, controller performance showed a significant improvement, as indicated in Fig. 7. This achievement is reflected in a maximum final feature error of $|i_{\tilde{r}}| < 0.18$ and $|i_{\tilde{\theta}}| < 0.05$ that can also guarantee $b_{v_x} \rightarrow b_{v_{xd}}$.

The improvement in performance is directly associated with changes in the gain matrix and the height of the PV arrays, both influenced by the unmodeled dynamics of the system and the measurements of the depth camera. Fig 8 shows the errors $i_{\tilde{\xi}}$ and $b_{\tilde{v}_x}$ of eight experiments with different initial conditions using the visual servoing controller with NMPC, where the error vector $[i_{\tilde{r}}, i_{\tilde{\theta}}]$ is ultimately bounded, converging to final values close to

TABLE II
TOTAL RUN-TIME OF THE PROCESS IN EACH ITERATION

Process	Average time (ms)
Feature Extraction:	13.9
Visual Servoing Controller:	1.2
NMPC:	3.0
Total	18.1

zero. The right plots in Fig 8 show the mean ($\bar{x} = (1/n)\sum_i^n |x_i|$) and standard deviation ($\sigma^2 = (1/(n-1))\sum_i^n |x_i - \bar{x}|^2$) of the experiments, where these values tend to converge to a value close to zero along the time. These errors are in an acceptable range for the application, thus, the proposal manages to generate images of the PV arrays for an expert inspection process. Further details of the project are presented in the following videos.¹

D. Run-Time

Table II shows the execution time of the approach. The feature extraction, visual servoing controller, and NMPC processes have a total time of 18.1 [ms]. Therefore, we consider that the application runs in real time because the odometry runs at 20 [m/s] and the camera at 30 [fps].

IV. CONCLUSION

We introduce a visual servoing controller integrated with NMPC for the Matrice 100 UAV's overflight of a PV array. The combination of controllers allows the UAV to have a defined workspace; in this way, the UAV mounted camera has a direct line of sight to the PV array allowing a correct extraction of features for the visual servoing controller. In addition, using a Kalman filter aids in estimating the filtered visual features of RGB-D images, specifically the vertical edges of PV solar panels, facilitating the UAV's stable positioning over the array. This correct positioning enables low-altitude flights for capturing detailed images crucial for expert or AI-based identification of potential panel failures.

In future work, we will improve the detection of PV array vertical lateral edges using neural networks on RGB images. This could decrease the processing run-time and noise in the extraction of visual features. In addition, we will investigate the integration of the vision system and the dynamics of the UAV into a single optimization problem (Perception NMPC).

ACKNOWLEDGMENT

The authors are grateful for the support of the Secretariat of Science, Technology and Innovation (SECITI) - San Juan, Argentina (Project: PIPE 2022) and the staff of the Anchipurac Photovoltaic Solar Park (EPSE) located in the same city, who provided them with the facilities to perform the flight tests with the UAV.

The results of this work are part of the project 'Tecnologías de la Industria 4.0 en Educación, Salud, Empresa e Industria' developed by Universidad Tecnológica Indoamérica.

¹<https://drive.google.com/file/d/1f1eLxRuRyWHLqyqgCv5UXXQRtF1UR9WA/view?usp=sharing>

REFERENCES

- [1] R. Heinberg and D. Fridley, *Our Renewable Future: Laying the Path for One Hundred Percent Clean Energy*. Washington, DC, USA: Island Press, 2016.
- [2] M. A. Bhuiyan, Q. Zhang, V. Khare, A. Mikhaylov, G. Pinter, and X. Huang, "Renewable energy consumption and economic growth nexus—A systematic literature review," *Front. Environ. Sci.*, vol. 10, 2022, Art. no. 878394.
- [3] T. Y. Gnangoin, D. F. Kassi, A. J. R. Edjoukou, O. Y. Kongrong, and D. Yuqing, "Renewable energy, non-renewable energy, economic growth and CO₂ emissions in the newly emerging market economies: The moderating role of human capital," *Front. Environ. Sci.*, vol. 10, 2022, Art. no. 1017721.
- [4] B. K. Sovacool, M. L. Barnacle, A. Smith, and M. C. Brisbois, "Towards improved solar energy justice: Exploring the complex inequities of household adoption of photovoltaic panels," *Energy Policy*, vol. 164, 2022, Art. no. 112868.
- [5] G. E. Tverberg, "Oil supply limits and the continuing financial crisis," *Energy*, vol. 37, no. 1, pp. 27–34, 2012.
- [6] J. G. Bessa, L. Micheli, F. Almonacid, and E. F. Fernández, "Monitoring photovoltaic soiling: Assessment, challenges, and perspectives of current and potential strategies," *iScience*, vol. 24, 2021, Art. no. 102165.
- [7] A. Golnas, "PV system reliability: An operator's perspective," *IEEE J. Photovolt.*, vol. 3, no. 1, pp. 416–421, Jan. 2013.
- [8] F. Grimaccia, S. Leva, A. Dolara, and M. Aghaei, "Survey on PV modules' common faults after an O&M flight extensive campaign over different plants in Italy," *IEEE J. Photovolt.*, vol. 7, no. 3, pp. 810–816, May 2017.
- [9] S. Djordjevic, D. Parlevliet, and P. Jennings, "Detectable faults on recently installed solar modules in Western Australia," *Renewable Energy*, vol. 67, pp. 215–221, 2014.
- [10] H. Pedersen, J. Strauss, and J. Selj, "Effect of soiling on photovoltaic modules in Norway," *Energy Procedia*, vol. 92, pp. 585–589, 2016.
- [11] M. Aghaei et al., "Review of degradation and failure phenomena in photovoltaic modules," *Renewable Sustain. Energy Rev.*, vol. 159, 2022, Art. no. 112160.
- [12] J. A. Tsanakas, L. Ha, and C. Buerhop, "Faults and infrared thermographic diagnosis in operating c-Si photovoltaic modules: A review of research and future challenges," *Renewable Sustain. Energy Rev.*, vol. 62, pp. 695–709, 2016.
- [13] L. Pratt, D. Govender, and R. Klein, "Defect detection and quantification in electroluminescence images of solar PV modules using u-net semantic segmentation," *Renewable Energy*, vol. 178, pp. 1211–1222, 2021.
- [14] M. W. Akram, G. Li, Y. Jin, and X. Chen, "Failures of photovoltaic modules and their detection: A review," *Appl. Energy*, vol. 313, 2022, Art. no. 118822.
- [15] Q. Tang, J. Dai, J. Liu, C. Liu, Y. Liu, and C. Ren, "Quantitative detection of defects based on Markov-PCA-BP algorithm using pulsed infrared thermography technology," *Infrared Phys. Technol.*, vol. 77, pp. 144–148, 2016.
- [16] F. Aljalahd, H. Kurdi, and K. Youcef-Toumi, "Autonomous multi-UAV path planning in pipe inspection missions based on booby behavior," *Mathematics*, vol. 11, no. 9, 2023, Art. no. 2092.
- [17] C. Xu, Q. Li, Q. Zhou, S. Zhang, D. Yu, and Y. Ma, "Power line-guided automatic electric transmission line inspection system," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3512118.
- [18] M. N. Tahir, Y. Lan, Y. Zhang, H. Wenjiang, Y. Wang, and S. M. Z. A. Naqvi, "Application of unmanned aerial vehicles in precision agriculture," in *Precision Agriculture*, Q. Zaman, Ed. New York, NY, USA: Academic Press, 2023, ch. 4, pp. 55–70.
- [19] M. Aghaei, A. Dolara, S. Leva, and F. Grimaccia, "Image resolution and defects detection in PV inspection by unmanned technologies," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2016, pp. 1–5.
- [20] X. Li, Q. Yang, Z. Chen, X. Luo, and W. Yan, "Visible defects detection based on UAV-based inspection in large-scale photovoltaic systems," *IET Renewable Power Gener.*, vol. 11, no. 10, pp. 1234–1244, 2017.
- [21] A. K. V. de Oliveira, M. Aghaei, and R. Rüther, "Aerial infrared thermography for low-cost and fast fault detection in utility-scale PV power plants," *Sol. Energy*, vol. 211, pp. 712–724, 2020.
- [22] Y. Zefri, I. Sebari, H. Hajji, and G. Aniba, "Developing a deep learning-based layer-3 solution for thermal infrared large-scale photovoltaic module inspection from orthorectified big UAV imagery data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 106, 2022, Art. no. 102652.
- [23] I. Segovia, J. R. Parra, and F. García, "Unmanned aerial vehicle integrated real time kinematic in infrared inspection of photovoltaic panels," *Measurement*, vol. 188, 2022, Art. no. 110536.
- [24] Y. Zefri, I. Sebari, H. Hajji, and G. Aniba, "In-depth investigation of applied digital photogrammetry to imagery-based RGB and thermal infrared aerial inspection of large-scale photovoltaic installations," *Remote Sens. Appl.: Soc. Environ.*, vol. 23, 2021, Art. no. 100576.
- [25] Y. Zefri, S. Imane, H. Hicham, A. Ghassane, and E. B.-I. Safia, "Applied imagery pattern recognition for photovoltaic modules' inspection: A review on methods, challenges and future development," *Sustain. Energy Technol. Assessments*, vol. 52, 2022, Art. no. 102071.
- [26] D. Hernández-López, E. R. d. Oña, M. A. Moreno, and D. González-Aguilera, "SunMap: Towards unattended maintenance of photovoltaic plants using drone photogrammetry," *Drones*, vol. 7, 2023, Art. no. 129.
- [27] L. F. Recalde, J. Varela, B. S. Guevara, V. Andaluz, J. Gimenez, and D. Gandolfo, "Constrained visual servoing of quadrotors based on model predictive control," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 353–360, 2022.
- [28] B. S. Guevara, L. F. Recalde, J. Varela-Aldás, V. H. Andaluz, D. C. Gandolfo, and J. M. Toibero, "A comparative study between NMPC and baselinefeedback controllers for UAV trajectory tracking," *Drones*, vol. 7, no. 2, 2023, Art. no. 144.
- [29] A. Keipour et al., "Visual servoing approach to autonomous UAV landing on a moving vehicle," *Sensors*, vol. 22, no. 17, 2022, Art. no. 6549.
- [30] R. Verschuere et al., "acados—A modular open-source framework for fast embedded optimal control," *Math. Program. Computation*, vol. 14, no. 1, pp. 147–183, 2022.
- [31] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADI—A software framework for nonlinear optimization and optimal control," *Math. Program. Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [32] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz, and S. L. Brunton, "Dynamic mode decomposition for compressive system identification," *AIAA J.*, vol. 58, no. 2, pp. 561–574, 2020.
- [33] O. Michel, "Cyberbotics Ltd. Webots™: Professional mobile robot simulation," *Int. J. Adv. Robotic Syst.*, vol. 1, no. 1, 2004, Art. no. 5.