2024 International Conference on Unmanned Aircraft Systems (ICUAS)
June 4-7, 2024. Chania, Crete, Greece

Low-Cost Monocular Vision-based Localization and Feedback Control of UAS for Classroom and Education Settings

Nicholas Grijalva, Hengameh Mirhajianmoghadam, Luis Rodolfo Garcia Carrillo, Hunter Stuckey, Wei Tang

***Abstract*— We introduce a cost-effective real-time 3-dimensional localization and control system of a quad rotorcraft Unmanned Aircraft System (UAS). The intended purpose of the system is to be implemented in low-budget classrooms to teach real-time closed-loop control of UASs and expand interest in the field. The system's localization technique only requires light-emitting diode (LED) markers and a conventional webcam device. Relying on the sampled visual location data, a closed-loop feedback control is developed for real-time stabilization. The system can be implemented with less than 1% of the monetary cost of a conventional motion capture system (MCS), which allows more institutions to purchase and use the materials. To validate our approach, the positional data from the proposed localization system is compared to ground truth data from a conventional MCS. Results from several real-time experiments are provided to validate the reliability and applicability of the localization system in UAS control.**

I. Introduction

Unmanned Aircraft Systems (UAS) are being offered as a solution to civil, industrial, and military demands [1], [2]. Simple tasks such as the delivery of goods, remote visual inspection, and dynamic communication networks can be automated with UASs [2]. Making use of previously known characteristics of a potential target of interest (geometry or color), it would be possible to apply a tracking method like the ones proposed in [3]–[5] to control the 3-dimensional motion of the UAS while performing autonomous tasks.

Real-time target-of-interest localization is often achieved by processing sensory data derived from the surrounding environment. There are many factors to consider when putting together a localization system, including the cost and type of sensors used, type of environment (indoor/outdoor), estimation accuracy/frequency, processing power available, battery consumption, and line-of-sight requirements [6] [7]

Global localization is important because it provides the capability for control of UASs in a global frame. This is very useful for application in classroom demonstrations of closed-loop control systems that run in real time. It has been shown that demonstration of theoretical concepts enhances the amount of material that students retain [8]. One issue is the fact that many secondary education institutions do not have the resources or personnel to use global localization systems.

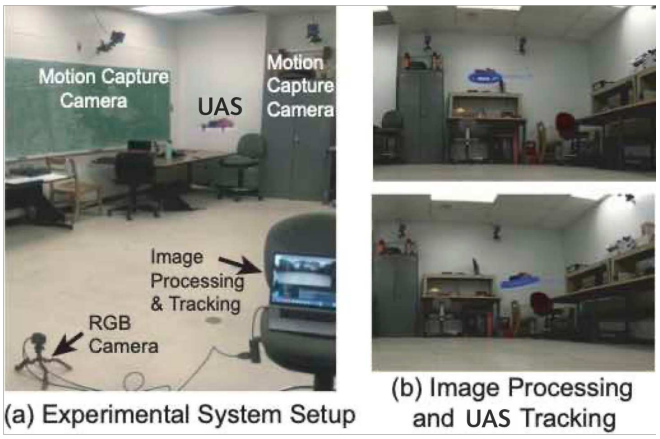

Fig. 1. The proposed vision-based feedback control for UAS (a) the localization system includes the RGB camera, the processing computer, the target UAS, and the MCS system for comparison purposes. (b) the UAS being identified and localized in a screen view, using an ellipse detection algorithm for image processing.

This paper presents the design, development, and implementation of a low-cost monocular vision-based localization and control system for UAS-related applications. Specifically for education and instruction purposes in low-budget environments. The effectiveness of the system is evaluated under several real-time closed-loop control experiments in a laboratory environment, where a UAS is tasked to fly through specific 3-dimensional trajectories, making use of the visual data and a feedback control strategy.

Related Work

Vision-based localization is accomplished by processing video data. These systems can include monocular vision, stereo vision, and combinations of infrared/radio light frequencies for superior accuracy and reliability [9]. Vision-based localization systems are used in many applications today, including self-driving cars, UASs, smartphones, and other devices that can be equipped with such sensors [10] [11]. The main challenge with implementing visual localization in embedded systems, like a UAS, is obtaining

Nicholas Grijalva is with the Department of Computer Science, New Mexico State University, Las Cruces, NM, USA. e-mail: nickgrij@nmsu.edu

Hengameh Mirhajianmoghadam, Hunter Stuckey, Wei Tang, Luis Rodolfo Garcia Carrillo are with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA. e-mail: hmirh@nmsu.edu, hss127@nmsu.edu, wtang@nmsu.edu, luisillo@nmsu.edu

This work was partially supported by the National Science Foundation (NSF) under BPE Award #2131875, ECCS Award #1652944 and ECCS Award #2015573 and by the New Mexico Space Grant Consortium (NMSGC), National Aeronautics and Space Administration (NASA) Cooperative Agreement Number 80NSSC20M0034.

2024 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-5788-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICUAS60882.2024.10557106

the highest possible accuracy and robustness with limited computational resources and budget [10].

Visual-Inertial Relative Pose Estimation (VI-RPE) relies on IMU data and a particle filter to localize a UAS equipped with a constellation of markers in a video stream [12]. Ultra Violet Radar (UVDAR) relies on markers separated from the background by an optical filter to localize agents in a leader-follower application [13]. A monocular localization system can be adopted when using a constellation of markers of known geometry. Equipping the UAS with this kind of constellation facilitates its identification and isolation from the background in any given video frame.

An additional challenge for localization tasks involving small robotic agents like UASs is the constrained processing power of embedded autopilots, which factors into the frequency of localization data [14] [15]. This limitation negatively impacts small robotic agents to accurately determine their location and navigate effectively. Several localization methods have reduced complexity or introduced optimizations to prioritize these design goals [15]–[17].

Ultimately, developing an alternative, reliable, and accurate localization system is crucial for any real-time experimental scenario requiring autonomous tasks in the global frame under a low budget.

Main Contributions

In this work, we present the development and implementation of an original localization and control system for UAS applications. The system only relies on a single stationary off-board imaging sensor as well as a known constellation of markers onboard the UAS, which enables a fast and accurate calculation of the 3-dimensional coordinates of the UAS in a controlled laboratory environment. This is done by taking the image data from the imaging sensor and computing the relative location of the known constellation which can be translated to a global localization frame. The proposed system's estimations are compared against ground truth data provided by a conventional Motion Capture System (MCS). We demonstrate the performance of the localization system by testing three real-time pre-programmed flight paths using a Proportional Integral Derivative (PID) controller for stabilizing the motion of the UAS. We also show the application of another type of control, Model Predictive Control (MPC), under a point-regularization test. The proposed single-camera localization's latency and accuracy are also marginally adjustable to meet the end-user's requirements.

Our research has the objective of disseminating the technical details for putting together a low-cost localization system appropriate for education and classroom settings where funds for a state-of-the-art MCS are not available. We expect that this system can broaden teaching and research in localization and control areas, not only for the UAS domain but also for many scenarios relevant to robotic agents like human-robot interaction and path planning.

II. Problem Statement

Advanced localization methods that can achieve high accuracy and latency are also expensive and complex [18]. In most cases, they also require a long setup time and pose potential restrictions concerning usage area [19]. For example, the popular MCS localization area is limited to a fixed-size volume based on image sensor placement. Additionally, the hardware specifications and computer processing demand required by MCSs increase their monetary cost.

MCSs, like the one that we use for validation at the Unmanned Systems Laboratory from New Mexico State University, often require several cameras, which must each be secured to a ceiling/wall with additional hardware. Each of these cameras is specialized and can cost more than 500 times the price of a conventional web camera [20], without considering the additional mounting equipment, software, and hardware needed to run the MCS.

A large number of secondary and technical schools worldwide do not have the appropriate budget for acquiring MCSs, and therefore they are unable to teach control system technology topics with physical real-time systems in a global frame. This makes it difficult for the students in such institutions to get exposed to the subject unless they pursue external extracurricular activities like robotics competitions [21]. Even in those scenarios, students usually don't gain experience working with complex robotics systems like UASs, until university education or even later.

Taking into account these challenges, we propose a vision-based localization and feedback control system that can be implemented with just a single RGB camera, a light-emitting diode (LED) strip, a laptop, and a robotic agent. We focus on the stabilization of a small UAS platform, but the ideas presented here can be applied to other similar and less expensive robots as well. We expect that the ease of setup and low cost of the system can be presented as a solution for academic programs in the areas of localization and control.

III. Vision-Based Localization

The vision-based localization task relies on a single RGB camera, a ring of LED markers installed onboard a target UAS, and a control station computer. In this setup, a Full HD Logitech web camera is used since the color consistency is reliable and the UAS's circle-shaped marker is easily identified by color [4]. A moderately bright pink LED ring powered by a 12v battery is equipped onto the edge of a Parrot Bebop 2 for tracking. In addition to being able to choose a color to contrast with the environment, the LED ring is also able to provide localization in low-light settings. However, the LED-enabled marker is not mandatory, and any solid-colored circle identifier is sufficient for experimenting. The processing system used for the computational workload is a desktop equipped with an Intel Core i5-2400 3.10GHz CPU.

The source code for the proposed localization and experiment control can be found in a ROS package through the following link: `https://github.com/NMSU-Unmanned-Systems-Laboratory/EllipticalTrackingwithControls`

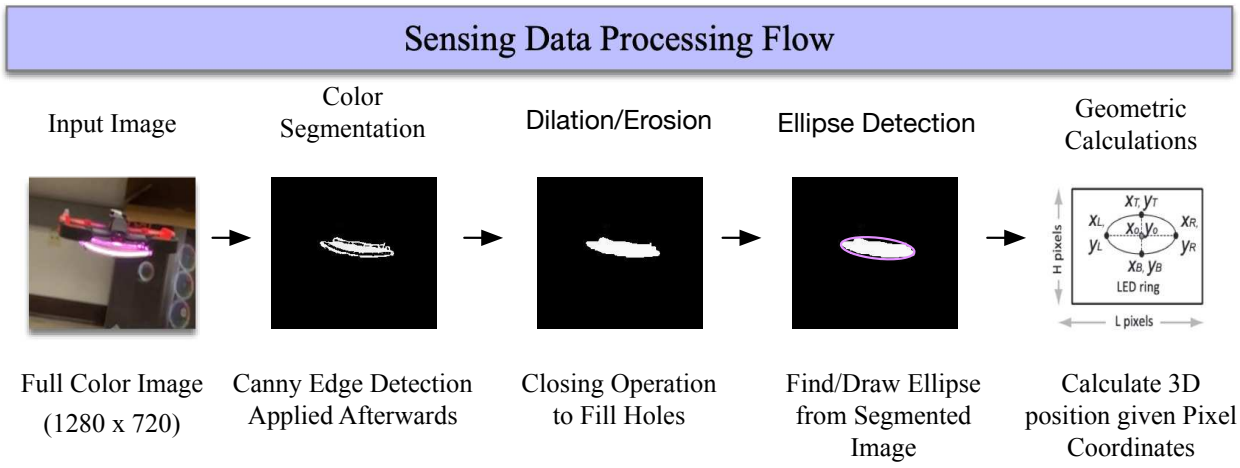

Fig. 2. Signal processing diagram showing the steps required for the localization of the UAS in real-time [4]

A. Elliptical Marker Image-Processing and Tracking

The image-processing flow diagram is shown in Fig. 2. First, the RGB image is converted to a hue-saturation-value (HSV) format and a threshold is applied to the image, filtering out any pixels that do not fit in the desired color range. Then, a canny edge detection filter is applied to extract the edges of the segmented contours left in the image. Next, a morphological closing operator is applied to fill any holes within each contour that could break the elliptical shape.

After image pre-processing, the ellipse detection is done by thresholding the pixel overlap of every remaining connected contour with a predicted ellipse for the size of that corresponding bounding box [22]. The contour pixel overlap is calculated utilizing the OpenCV function `fitEllipse`. The ellipse-fitting in the `fitEllipse` function is performed by a least-squares approximation method [23]. Once the elliptical markers are identified and detected, the pixel coordinates of the ellipse are sent to the spatial localization algorithm for computation of the 3-dimensional location.

B. Localization Calculations

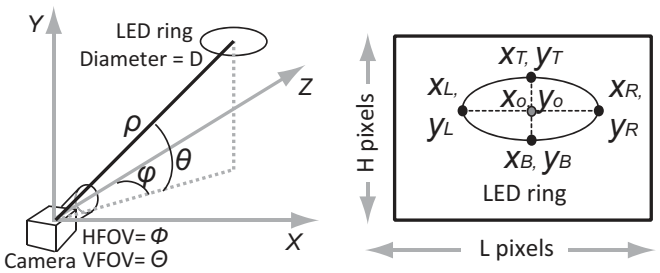

Fig. 3. Left: Previously verified geometry for the localization algorithm. Right: measurement setup of the target ring [24].

The localization algorithm uses the edge pixel values received from the ellipse detection process in the image. Making use of the known properties of the circular marker, the center of the target UAS in the image, (X_O, Y_O), is calculated as

$$X_O = (X_L + X_R)/2 \tag{1}$$

$$Y_O = (Y_L + Y_R)/2 \tag{2}$$

Afterwards, the tangent value of azimuth angle ϕ and elevation angle θ, see Fig. 3, are acquired as:

$$\tan\phi = 2 \cdot X_O \cdot \frac{\tan(\Phi/2)}{L} \tag{3}$$

$$\tan\theta = 2 \cdot Y_O \cdot \frac{\tan(\Theta/2)}{H} \tag{4}$$

Where Φ and Θ are the Horizontal and Vertical fields of view of the camera respectively, and L and H are the resolutions of the width and height of the camera frame respectively as seen in Fig. 3. Lastly, the radial distance ρ, which is the distance between the center of the target markers and the lens of the camera is obtained as

$$\rho^2 = \left(\frac{D}{2} \cdot \frac{X_O}{|X_O - X_R|}\right)^2 \cdot \left(1 + \frac{1}{\tan^2\phi}\right) \tag{5}$$

The resulting polar coordinates are then translated to Cartesian coordinates and adjusted with a fixed value offset, in such a way that the center coordinates of the single-camera localization align with the chosen global frame.

IV. Closed-Loop Control

To organize the control of the UAS platform into a serialized pipeline, we utilized the Robot Operating System (ROS). Through a series of ROS nodes, we were able to orchestrate communication between multiple running processes. These include the elliptical detection program, the trajectory control program, the UAS onboard computer, and the MCS for verification purposes. Our experiment involves multiple control strategies. One of them consists of a stabilization to a specific point in a 3-dimensional space (regularization) using an MPC strategy. We also conducted experiments where the UAS was tasked with traveling to

distinct waypoints utilizing a PID controller [25]. All of these tests make use of the positional data from our elliptical vision-based system to show its accuracy and applicability. The z-axis position of the UAS was simply held at a constant value and thus is not presented explicitly in the results like the x and y axes.

A. ROS Node Structure

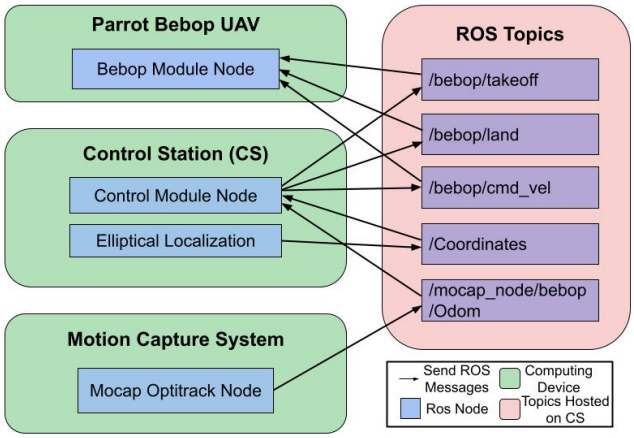

Fig. 4. ROS experimental architecture explained concerning computational hardware. The `/Coordinates` topic is a custom topic where our positional data is published. The other topics are used to receive verification data from the MCS and control for the bebop UAS.

For the closed-loop system, there are three main computational devices to communicate between: The Control Station running the UAS controller and experimental procedures, the processor embedded in the UAS, and the computer reporting the MCS ground truth data. Each of these systems communicates via Ethernet or Wi-Fi through ROS nodes and ROS topics that are hosted on the Control Station computer. Drivers for the Bebop UAS [26], and MCS [27], were used to generate their respective ROS nodes. Fig. 4 depicts the architecture of our ROS system and the topics involved. The topics `/bebop/takeoff`, `/bebop/land`, and `/bebop/cmd_vel` are used for the Bebop commands to enable takeoff, land, and control movements. The controllers take in positional data from the elliptical-based vision-based localization process and output command signals to be sent and enacted by the UAS. The heading of the UAS is stabilized using an onboard IMU signal.

B. Experimental Setup

There are several physical and logical components that we utilized to enable a basic closed-loop control scheme for easy classroom implementation. The network communication between devices is hosted on a general-purpose Linksys EA9400 Wi-Fi Router (used for Optitrack communication), in which both the main controlling computer and the MCS computer are connected via Ethernet. Communication between the computers and the UAS was achieved by configuring the network interface on the Bebop UAS to directly connect to the shared router. We used the off-board Full HD Logitech webcam, placed on the ground, to capture video for the elliptical detection and positional calculations of our localization program. Finally, the Optitrack MCS consisting of eight Prime^{x}22 infrared cameras installed around the perimeter of the room is used for generating ground truth and positional calculation data for comparison. A snapshot depicting a portion of the experimental setup can be seen in Fig.**??**. All components of the experimental setup, aside from the MCS (ground truth) and Bebop UAS, can be purchased and set within a budget less than %1 of a conventional MCS, making it appropriate for implementation by educators in academic programs with a low budget.

C. PID Component

To show the control applicability of our proposed localization system, we utilized a PID controller to stabilize the 4 degrees of freedom on the UAS: x, y, z, and yaw. The purpose of this test is to see if the UAS can be safely and succinctly flown utilizing only the proposed localization system and a simple controller so that the experiment can be re-created in simple classroom settings. This test can be used to instruct the properties of the controller, the physical system, and how they interact and affect each other.

The `simple-PID` module in Python was adopted for control purposes [25], with one PID module per degree of freedom, i.e., x, y, z, and yaw. The inputs for this module are the 3-dimensional positional data from our localization system, three constant parameters K_P, K_I, K_D, which we heuristically determined, and a desired set-point coordinate. The module's output, u, is a control signal calculated as

$$u = K_P e(t) + K_I \int e(t)\,dt + K_D \frac{d}{dt} e(t) \tag{6}$$

where

$$e = \text{setpoint coordinate} - \text{measured coordinate} \tag{7}$$

The PID constants for each degree of freedom for the Bebop 2 were obtained through a heuristic method, and are listed in Table I.

TABLE I
PID CONSTANTS FOR PARROT BEBOP 2.

Control Axis	K_P	K_I	K_D
X-Axis	0.35	0.006	0.40
Y-Axis	0.35	0.006	0.40
Z-Axis	0.60	0.003	0.20
Yaw-Axis	1.00	0.000	0.00

D. MPC Component

We addressed the MPC application using the proposed localization system as well. The purpose of this test was to show the implementation of a more complex controller, while also highlighting some of the associated implementation attributes, such as data frequency management. The computational cost of multi-variable systems, as well as inequality and equality requirements on the states, inputs, and outputs, are only a few of the issues that the MPC technique

can address [28]. The advantage of MPC is then enabling optimization of the current time/data slot while taking into account upcoming future events. Therefore, MPC has the capacity to predict motion and implement appropriate control action.

To enable the replication of our MPC, the optimization problem we adopted is formulated as

$$\begin{aligned} \min_{u} \quad & J(s(t), \Delta u(t), p) \\ \text{s.t.} \quad & u_{\min} \leq u(t+j) \leq u_{\max} \\ & s_{\min} \leq s(t+j) \leq s_{\max} \quad j = 1, ..., p \end{aligned} \tag{8}$$

where s is the system state vector, u is the control input vector, $\Delta u(t) = u(t) - u(t-1)$, p is the prediction horizon, and t is the sampling time. $s(t+j|t)$ is the j step ahead prediction of the system state at the current sampling time t. The system's behavior is chosen over the prediction horizon with the help of a previously known or experimentally identified dynamic model.

For experimental system identification purposes, experimental flights under predefined trajectories are conducted, during which control input signals (i.e. inputs) and UAS states (i.e. outputs) data pairs are collected. Then, an extended least-squares (ELS) algorithm is applied to such data. ELS was selected because it is applicable for identifying linear time-invariant (LTI) discrete-time systems, even those exhibiting time delay [29].

The structure of the Parrot Bebop UAS model adopted for the proposed MPC is in the form of

$$s_{k+1} = As_k + Bu_k \tag{9}$$

where the state vector $s_k = [x, v_x, y, v_y]^\top$ represents the UAS position (x, y) and velocity (v_x, v_y) in both x-axis and y-axis. u_k is the input vector $u_k = [u_x, u_y]^\top$, and the output is equal to the state vector. For this system, the time delay is one sampling time ($t = 1$). The parameters identified by ELS and collective data are the elements of matrices A and B, thus the matrices of equation (9) are

$$\begin{aligned} A &= \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -0.69837 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -0.69837 \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 \\ 0.23428 & 0 \\ 0 & 0 \\ 0 & 0.23428 \end{bmatrix} \end{aligned} \tag{10}$$

The Bebop model was tuned in the high-frequency environment of the verification MCS (236Hz). Once the dynamic model is known, the objective function J can defined as

$$\begin{aligned} J = \sum_{j=1}^{p} (||s(t+j|t) - r_d(t+j)||_Q^2 + ||\Delta u(t+j)||_R^2) \\ + ||s(t+p|t) - r_d(t+p)||_Q^2 \end{aligned} \tag{11}$$

where $r_d(t+j)$ is the future reference trajectory over the prediction horizon, Q and R are positive-definite weighting matrices, and the last term is the terminal cost [30]. In MPC, the cost function is usually formulated with the sum of the squared control errors over the prediction horizon.

The cost function is based on equation (11), which needs well-tuned parameters to achieve trajectory convergence. Several experimental flights were conducted to tune these parameters properly. The prediction horizon is selected as $p = 20$ and the weighting matrices of Q and R are

$$\begin{aligned} Q &= \begin{bmatrix} 0.09 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.09 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \\ R &= \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \end{aligned} \tag{12}$$

The cost function includes the error between all four states of the UAS as well as their desired values $r_d = [r_{d_x}, 0, r_{d_y}, 0]$, where r_{d_x} and r_{d_y} are the value of the desired set-point in x-axis and y-axis, respectively. The desired value for both velocity dynamics is set to zero.

V. Validation and Control Results

To test the real-time flight capability of the proposed vision-based localization, measured localization estimates were compared to and validated against those of the Optitrack MCS during a series of four experiments.

We conducted three experiments under the PID controller to show the robustness of the localization system. These include moving consecutively to three pre-determined waypoints (Multi-Waypoint) (a), following a circle trajectory (b), and moving to a single waypoint (c). As our localization system is not yet able to report the heading of the UAS, we used onboard IMU data to stabilize the bebop UAS's heading to a constant value.

As shown in Fig. 5, the traces of the PID control flight tests are plotted to benchmark the single-camera localization measurements. The experiments show steady and secure flight which is comparable to the performance obtained when using the more expensive MCS for localization purposes. There are minimal deviations from the desired trajectories, showing that more advanced maneuvers and higher-velocity tests could be carried out in the future. The real-time performance of the vision-based localization and control can be seen in recorded videos here:

- Multi-Waypoint Flight: `https://www.youtube.com/watch?v=6aSKvhgFl18`
- Circle Trajectory Flight: `https://www.youtube.com/watch?v=tVHeEKWR5II`
- Single Waypoint Flight: `https://www.youtube.com/watch?v=t9FuLl6j9z0`

Positioning error between the two systems was calculated by measuring the absolute difference between the proposed single-camera localization and the MCS for each time step and then taking an average of the resulting values. These

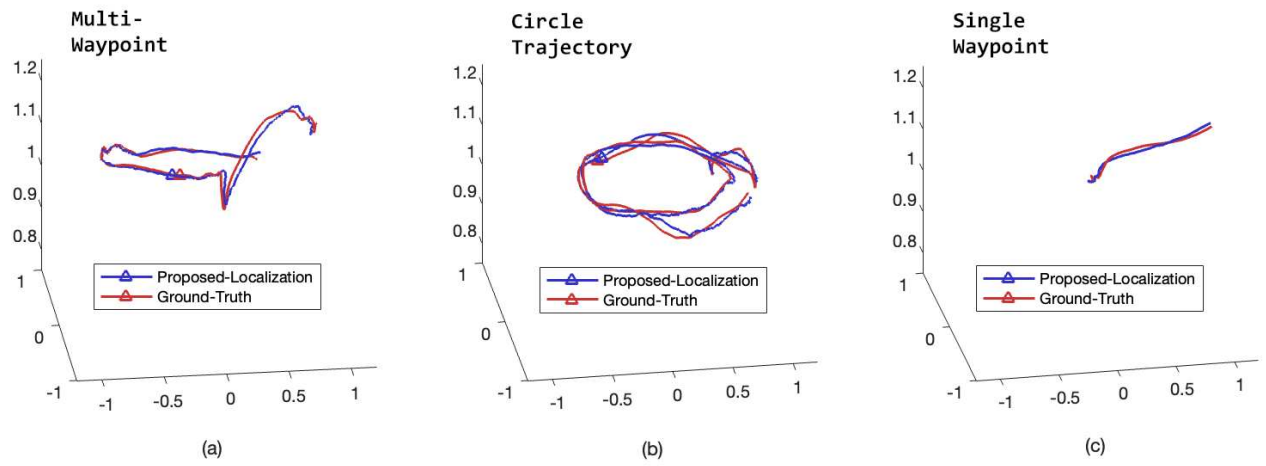

Fig. 5. Experimental results of the localization system during PID control of position. 3D traces for three different flight tests with pre-determined flight trajectories (X, Y, Z coordinate) recorded by the Ellipse tracking system and the benchmark MCS (units in meters).

values are reported in Table II along with the sampling rate of each experiment.

TABLE II
AVERAGE MEASUREMENTS FOR PID FLIGHT TESTS.

Flight Measurements	Multi-Waypoints (a)	Circle Trajectory (b)	Single Waypoint (c)
Avg. X error (m)	0.0243	0.0207	0.0093
Avg. Y error (m)	0.0192	0.0356	0.0192
Avg. Z error (m)	0.0090	0.0088	0.0051
Avg. Sampling Rate	51.3018Hz	46.2268Hz	48.9766Hz

The measurement error for the duration of the real-time PID control tests is relatively low (0.0168m approx.), with some higher peaks in error in high-velocity scenarios. The peaks of error occur at points such as the transitions between the waypoints and with certain directional movements in the circular test, particularly the harsher corner motions. The distance between the monocular camera and the UAS in the trials ranges from 1m to around 3.6m, as the tracking space of 3x3x5m for the MCS limits the testing area.

In addition to the three experiments using a PID controller, we conducted a fourth single-point stabilization experiment using an MPC. A Positional comparison between the proposed localization and the MCS can be seen in Fig. 6. This experiment shows additional real-time control applications of the presented localization system.

The MPC in combination with the localization system exhibits oscillations while performing stabilization on a single point. The addition of the MPC optimization function caused a slight delay from the localization estimation to control output. This was due to the UAS model of the MPC being tuned in the high-frequency environment of the MCS (236Hz). Despite the oscillation period, the UAS was still able to be controlled, and with a better model identification, it could perform even better than the PID strategy [31].

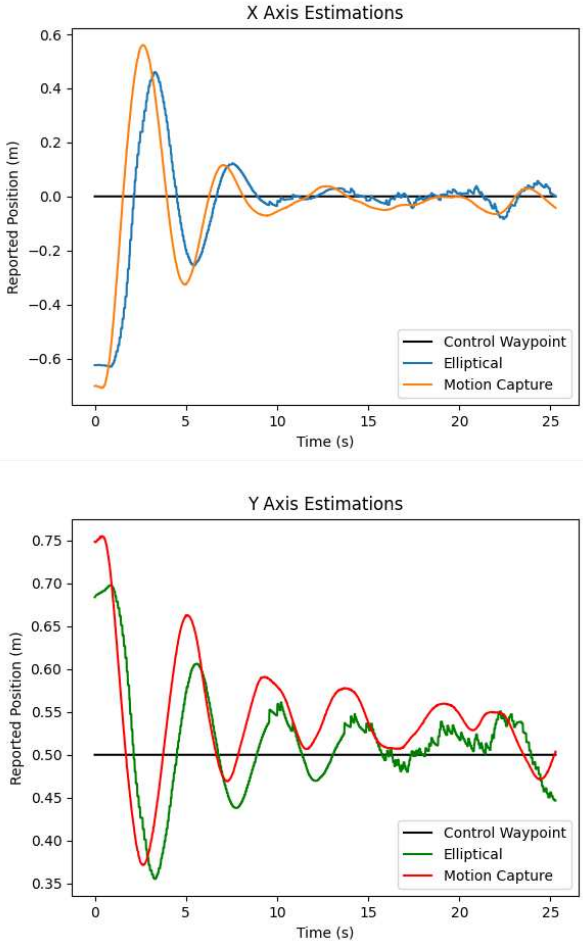

Fig. 6. Real-time MPC control positional results of the UAS converging on the point (x=0.0, y=0.5)

VI. Conclusions

We presented the implementation of a low-cost vision-based localization and feedback control system for Unmanned Aerial Systems, which only requires a single RGB camera, an elliptical marker, and a personal computer. The proposed single-camera-based system provides a cost-effective solution that can localize a dynamic target equipped with the elliptical marker in three dimensions. The vision algorithm is implemented through color-segmentation and visual ellipse detection methods to obtain a relative-to-global localization. The proposed system is simple to set up and costs a fraction of many MCS-enabled solutions. It provides a suitable solution for low-budget classroom instruction on basic control system technology in real-time closed-loop scenarios and can provide a physical demonstration of such subjects. This introduction to physical control examples would provide exposure to the students on the subject earlier and increase interest in the field of UASs.

To demonstrate the performance of the system, four real-time flight tests were conducted. The vision-based localization and control system showed real-time maneuverability when following pre-programmed flight trajectories.

Future work will consider improvements to the localization system to provide an additional degree of localization for the heading of the UAS instead of relying on an onboard IMU. This can be achieved by adding a distinctive 'head' to the elliptical marker. There are also plans to add an interface to improve the parameterization of the localization system so that it can more easily be adjusted and implemented in different environments.

In addition, more efficient interfaces are also being developed for changing the parameters of the localization system so that it can easily be adapted to work with different environments, targets, and UASs.

References

[1] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and privacy in smart farming: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 34564–34584, 2020.

[2] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, and S. Trimpe, "Wireless control for smart manufacturing: Recent approaches and open challenges," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 441–467, 2021.

[3] L. R. Garcia Carrillo, A. Dzul, R. Lozano, and C. Pégard, *Quad Rotorcraft Control. Vision-Based Hovering and Navigation*. London: Springer-Verlag, August 2012.

[4] H. Stuckey, L. Escamilla, L. R. G. Carrillo, and W. Tang, "Real-time optical localization and tracking of uav using ellipse detection," *IEEE Embedded Systems Letters*, pp. 1–1, 2023.

[5] H. Stuckey, A. Al-Radaideh, L. Sun, and W. Tang, "A spatial localization and attitude estimation system for unmanned aerial vehicles using a single dynamic vision sensor," *IEEE Sensors Journal*, vol. 22, no. 15, pp. 15497–15507, 2022.

[6] T. Savić, X. Vilajosana, and T. Watteyne, "Constrained localization: A survey," *IEEE Access*, vol. 10, pp. 49297–49321, 2022.

[7] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[8] M. K. Ho, M. Littman, J. MacGlashan, F. Cushman, and J. L. Austerweil, "Showing versus doing: Teaching by demonstration," *Advances in neural information processing systems*, vol. 29, 2016.

[9] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1680–1687, 2018.

[10] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[11] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in *2018 26th Telecommunications Forum (TELFOR)*, pp. 420–425, 2018.

[12] L. Teixeira, F. Maffra, M. Moos, and M. Chli, "VI-RPE: Visual-Inertial Relative Pose Estimation for Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2770–2777, 2018.

[13] V. Walter, N. Staub, A. Franchi, and M. Saska, "UVDAR System for Visual Relative Localization With Application to Leader-Follower Formations of Multirotor UAVs," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2637–2644, 2019.

[14] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2502–2509, 2018.

[15] T.-j. Lee, C.-h. Kim, and D.-i. D. Cho, "A monocular vision sensor-based efficient slam method for indoor service robots," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 318–328, 2019.

[16] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, F. N. Martins, and R. F. Vassallo, "Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 582–597, 2018.

[17] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1078–1086, 2017.

[18] metamotion, "Motion capture prices." `https://metamotion.com/FAQ/prices.html`. Accessed: 2024-03-29.

[19] Optitrack, "Quick start guide: Getting started." `https://docs.optitrack.com/quick-start-guides/quick-start-guide-getting-started`. Accessed: 2024-04-19.

[20] Optitrack, *OptiTrack PrimeX 22*, 2024. `https://optitrack.com/cameras/primex-120/`.

[21] F. Robotics, *FIRST Robotics Competition*, 2024. https://www.firstinspires.org/robotics/frc.

[22] A. Keipour, G. A. S. Pereira, and S. A. Scherer, "Real-time ellipse detection for robotics applications," *CoRR*, vol. abs/2102.12670, 2021.

[23] A. Fitzgibbon, M. Pilu, and R. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999.

[24] I. White, E. Curry, D. K. Borah, S. J. Stochaj, and W. Tang, "An optical spatial localization algorithm using single temporal difference image sensor," *IEEE Sensors Letters*, vol. 3, pp. 1–4, March 2019.

[25] M. Lundberg, *simple-pid*, 2022. `https://simple-pid.readthedocs.io/en/latest/#`.

[26] A. L. at SFU, *Bebop Autonomy Driver*, 2018. `https://github.com/AutonomyLab/bebop_autonomy`.

[27] R. device drivers, *OptiTrack ROS Package*, 2014. `https://wiki.ros.org/mocap_optitrack`.

[28] H. Mirhajianmoghadam and M.-R. Akbarzadeh-T, "Predictive hierarchical harmonic emotional neuro-cognitive control of nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 111, p. 104781, 2022.

[29] M. Ahmed, "Rapid parameter estimation algorithms using the els principle," *Systems & Control Letters*, vol. 2, no. 4, pp. 209–216, 1982.

[30] *Model predictive controllers*. Springer, 2007.

[31] H. Mirhajianmoghadam, N. Grijalva, and L. G. Carrillo, "Real-time implementation of model predictive control for fast trajectory tracking of a quad rotorcraft uas," *Seventh IEEE International Conference on Robotic Computing*, 2023.