Autonomously Detect, Track and Land a UAV on a Fiducial Marker

Jonathan Edwards
Computer Engineering
University of Canterbury
Christchurch, New Zealand
jme161@uclive.ac.nz

Richard Green
Computer Science Engineering
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract* - This paper proposes a solution to the problem of being able to autonomously detect, track, and land a UAV on a fiducial marker. The proposed approach successfully detected, tracked and landed a drone on a fiducial marker. The template matching algorithm had an extremely reliable range of up to 3 meters. The drone typically took 20 seconds to orientate itself and land. The Yaw and Centre Point position angles had about a ±5° error margin. Previous research papers had worse performance, were less efficient and had similar error margins. The drone was able to reliably detect, track, and land from an initial height of 40 centimetres to 3 metres away. In the future a Kalman filter will be added to help achieve higher accuracy and performance. An improved controller will also be implemented to achieve better performance maneuverability by the drone.**

***Keywords* – Autonomous, Drone, Fiducial Marker, detect, track, land, Low-Cost, OpenCV, Python**

I. INTRODUCTION

Over the years, research in the Unmanned Aerial Vehicles (UAVs) industry has increased considerably. More importantly, delivery companies and even medical institutions and hospitals have taken a particular interest in UAVs as they remove the human factor to a large degree and thus speed up the delivery process. However, there are challenges as training a user to fly a drone/UAV and successfully land without crashing is a critical aspect of this technology.

A solution to this problem is to equip the drones with artificial intelligence which provides the ability continuously track an object, to identify the landing location and to land safely every time.

There are already several different types of drones on the market that are equipped with artificial intelligence, but these do have hefty price tags.

There are many research papers that have been published and describe the different ways an object can be tracked by using a drone. Techniques published range from colour detection to corner detection and to template matching.

This paper proposes an alternative method, the method consists of programming a low-cost drone with Artificial intelligence that will allow it to land, track, and detect objects autonomously. The proposed tracking and detecting algorithm detects for ArUco (fiducial) markers. This allows the drone to be able to detect and track fiducial marker(s) that are within the drone's field of vision (FOV). If a marker is detected, the algorithm approximates the x, y, and z axis distances as well as the vertical and horizontal angles between the drone and marker reference frame . These angles are fed into a closed loop PID controller which allows the drone to manoeuvre autonomously in the x, y, and z axis while keeping the marker centred in the camera's frame. Once the drone is fully aligned, the proposed method will cause the drone to descend and then to land once it becomes less than 15 cm above the marker.

II. BACKGROUND

A. Colour detection

Three papers were sourced using the HSV method for being able to track and detect objects [1,2,3].
These papers describe a method using the Red Green Blue (RGB) colour space and combining it with the Hue Saturation Value (HSV) colour space. RGB is defined by listing how much red, green and blue is contained in a single value [4]. Each value ranges between 0 – 255. This is an additive method where the more of each colour is added, the brighter it becomes. When combining the amount of red, green, and blue, it is extremely difficult determining how much of each colour is present. This is demonstrated in Figure 1.
HSV on the other hand is comprised of 3 parameters, hue, saturation, and value [4]. The hue value ranges from 0 – 180 and the saturation and value can range between 0 – 255. These values determine the output colour that is seen as shown in Figure 1. The HSV method uses thresholding too and is much better at handling lighting differences.

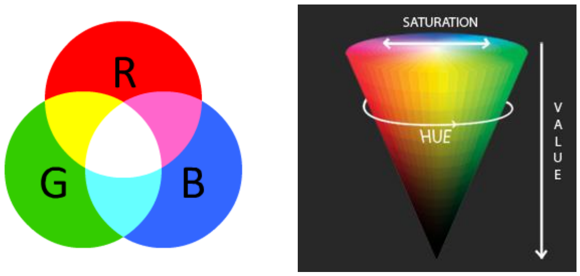

Figure 1 - RGB to HSV colour space's [4]

B. Corner detection

Another approach is to use the Harris Corner algorithm to extract corners and to infer features of an image [5]. The image is first converted to grayscale before running through the algorithm. The detected corners are then filtered, and the strongest corners are then drawn onto the original colour image for display. Corners are valuable as it is typically much easier to match the same corner from two slightly different frames than it is to match an edge or patch of colour as can be seen in Figure 2.

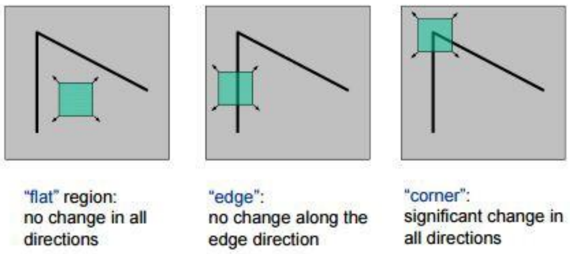

Figure 2 – Corner detection method [5]

Another paper details autonomously landing a UAV on a Helipad [6]. This paper used a corner detection algorithm (similar to the one above) to estimate the attitude between the Helipad and the camera, Figure 4 shows their final result. To achieve their goal, they used both corner detection and corner matching algorithms. Corner matching is similar to Corner detection but instead looking for all possible edges in a frame, it looks for similarities between corners and their quadrants as can be seen in Figure 3. Once computed their method is able to track and approximate the orientation of the H symbol by using Pose estimation [6].

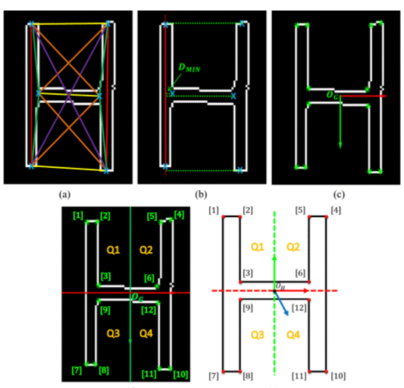

Figure 3 – Corner matching [6]

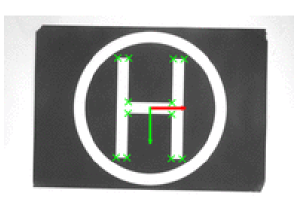

Figure 4 – Final result with using Corner detection + Corner matching [6]

C. Template matching

Another approach is to use Template matching for tracking and detecting objects. Multiple papers use the ArUco (fiducial) marker algorithm to approximate the distance between the drone and the camera [7,8,9]. The fiducial marker algorithm detects for a black and white image within the cameras FOV and determines the unique ID that corresponds to the marker. Fiducial markers are objects that are used to provide a point of reference. These papers also all use the ArUco library, this library contains dictionaries that store different bit size markers [7,8,9]. Figure 5 illustrates an example of a fiducial marker that has a unique ID of 23.

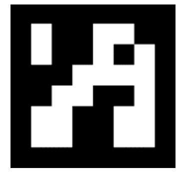

Figure 5 - 6x6 fiducial marker with ID 23

Template matching can either be kept in the RGB or HSV colour space or be converted into the greyscale colour space. Efficient template matching requires the image to be converted from the RGB colour space to the grey-scale colour space. The fiducial marker algorithm allows the greyscale image to be identified. Pose estimation is also used for estimating the marker's position and attitude relative to the camera.

D. Limitations of prior research

Reference 6 which was the paper detailing a method to detect a helipad using corner detection algorithm. It was found to be more accurate than the colour detection method.

However, accuracy was variable as the experiment never got tested on a UAV but on a linear stage instead. However, the algorithm did not attempt to find the object once it had been lost. The speed at which the object could move impacted the chance of it being lost and therefore the speed must be constrained to the drone's performance capabilities [1]. Another consideration was that the drone couldn't see obstacles outside of its view and could easily crash if it were to hit something [3].

Other papers use complex algorithms such as Speeded Up Robust Features (SURF) and Maximally Stable Extremal Regions (MSER) to achieve greater performance [15]. These algorithms produced improved results, but the method was very complex compared to template matching.

Another paper used a more accurate calculation to maintain a certain distance from an object by using dynamic markers This is however very complex to implement by improves the accuracy [19,20].

Pose estimation cannot be calculated using the HSV colour scale easily, thus making it harder to find the orientation of the object. A limitation that one paper faced was that they could not compute the objects orientation easily using the HSV Colour detection. This method is also less accurate and harder to compute then using a greyscale method where Pose estimation can be computed. The paper also talked about needing a good constant light source and being sensitive to noise and disturbances in the environment [2].

The template matching approach as proposed in papers [7,8,9], was by far the simplest and more accurate method. As the biggest limitation that these papers faced were calibration setup, tracking control and time delay between frames. Another problem with template matching and trying to land was that the markers used were too big and often got lost [7].

III. METHOD

A. Drone and Hardware

The DJI Tello Drone has a small horizontal built-in FPV camera that can transmit either video or photos back to the user. It is also low cost, compact and robust quadcopter that was designed for the STEM community as well as hobbyists. The drone includes a first-person view (FPV) camera, an ultrasonic sensor, a 3-axis gyroscope & accelerometer, an inertial measurement unit (IMU), and Wi-Fi connectivity. The drone's camera can transmit either 720p video at 30fps or 5MP photos, has a viewing angle of 82.6 degrees. The drone can be controlled by using either a smartphone or a computer over Wi-Fi, has a flight time of up to 13 minutes or a max flight distance of 100m at max flight speed 8m/s [10]. The camera has between a 10-15s setup window after takeoff. An Intel processor powers the drone which allows it to capture high-quality footage and be enclosed in a small compact form. An image of the drone can be seen in Figure 6. A mirror clip was also added so that the drone could capture images in the downward direction. An image of the drone with the mirror clip attached can be seen in Figure 7.

The drone will be programmed and communicating with a laptop that runs the Windows operating system. The laptop is a 64-BIT machine and is controlled by the Intel Core i7-9750H, it has 16 GB of RAM and is clocked at 2.60 GHz.

Figure 6 - DJT Tello drone

Figure 7 - DJI Tello drone with mirror clip ★

B. Software

The drone will be programmed in Python 3 (Version 3.8.10) and will be programmed in the Visual Studio Code IDE. Plotting was done in the Matlab IDE and (Version R2019b – academic use) is used. The DJI Tello Drone DJITelloPy interface (Version 2.1) is used, this controls the communication over a Wi-Fi link connection to and from the laptop. This interface allows the drone to stream real time video back to the laptop. The interface also controls how data is sent to and from the sensors that are on the drone. The interface also comes equipped with a set of commands that can be fed through the drone to make it move in a specific direction [11].

Open CV is an open-source computer vision and machine learning software library for python [12]. The open-source library contains more than 2500 algorithms that are all extensively well documented and come with sample code. This software library is useful for identifying objects, tracking objects, and for classifying objects. OpenCV (Version 4.5.1) will be used as well as the ArUco (fiducial) marker library.

C. Image Processing

This tracking algorithm is constantly updated when a marker is detected. Prior discussed tracking an object by using a color and edge detection method [1,6]. The problem stated in section II was that both these methods are complex and difficult to get high accuracy. This paper overcomes these limitations by focussing on functionality and accuracy by using an easy to implement algorithm instead.

The proposed image processing algorithm in this paper consists of image conversion, marker detection, data processing and a PID controller. The proposed algorithm as shown in Figure 7, relies on being able to identify a unique ID [16,17], and the algorithm must be fast enough in order to avoid bottlenecking. The algorithm that is used in this paper has the same structure as the one a previous student used for his COSC-428 project at the University of Canterbury [16]. De Gouw's paper in 2019 proposed a limitation in the video feed between the drone and his laptop. The paper reported that the drone was capable of outputting 30fps at a resolution of 640x360. Instead, the algorithm was only getting between 1.1 – 1.5 frames per second and meant that some aspects of accuracy were lost [16]. This paper managed to fix the frame rate to be at 30fps by upgrading the hardware of the drone, and also re-implementing & fixing some bugs in the software's algorithm in Figure 7. This improvement helped improve the accuracy of the experiment.

Camera calibration is important and necessary as it prevents an image from being distorted (normally only has to be done once). Radial distortion caused by imperfect lens and thus a possible blurry image is produced, also as seen in Figure 6. If the camera is not correctly calibrated, it could affect the algorithm to not perform correctly. In this paper, the camera on the drone is calibrated with a checkerboard test similar to Jiang's paper [7].

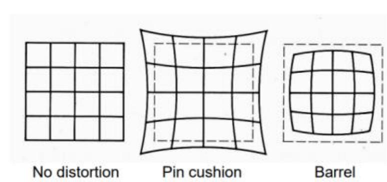

Figure 6 - Image distortion

The proposed algorithm in Figure 7 of this paper uses the grey-scaling with template matching method (ArUco (fiducial) markers [13]. The algorithm is easier to implement, and there are less errors in the accuracy [7,8,9,16]. Detecting for objects using the HSV colour space method (section *II.A*) and the Corner detection (section *II.B*) are both possible [1,2,3,6]. A limitation found in papers [1,2,3,6] was that it required a lot more trial and error to get something working accurately. Another was that they it was more complex for them to find the pose estimation geometries. This is why this paper chose the simpler more accurate approach using ArUco markers.

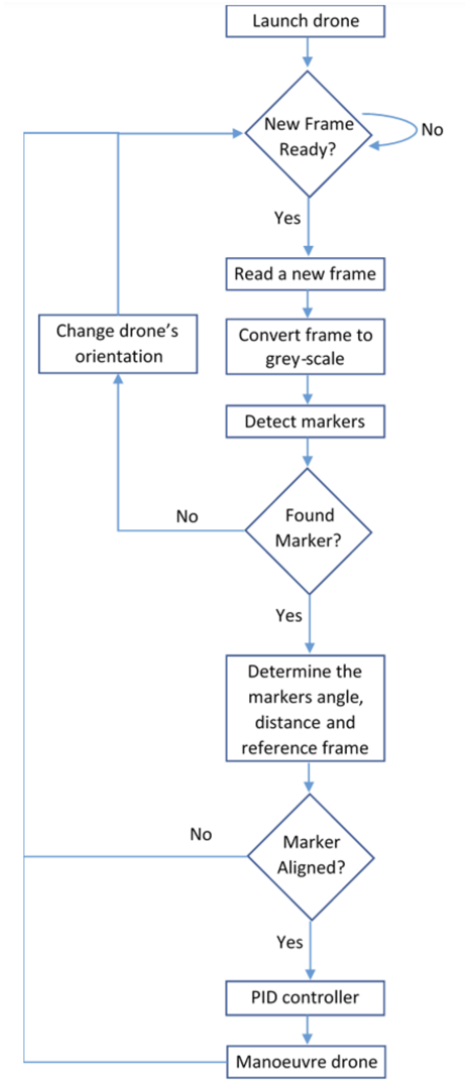

Figure 7 - Flowchart of the implemented Algorithm [16].

D. Marker detection

The ArUco marker is a synthetic square marker composed of a wide black border and an inner binary matrix which determines its unique identifier (ID). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction technique. There are a range of different binary sized marker's that can be chosen, as seen in Figure 5. This paper uses a 6x6-bit marker with ID 23 as well as a Helipad looking marker that contains fiducials within fiducials [9], this can be seen in Figure 8.

A limitation described in a paper is that the fiducial marker got lost when descending [7]. This paper overcame this limitation by adding multiple fiducial markers as did another paper [9], as seen in Figure 8.

Since the DJI Tello drone comes equipped with a horizontal FPV fixed camera, a mirror clip was designed to be attached so that the drone could look vertically down [18], as can be seen in Figure 7. Because of this the fiducial marker was mirrored as can be seen in Figure 8.

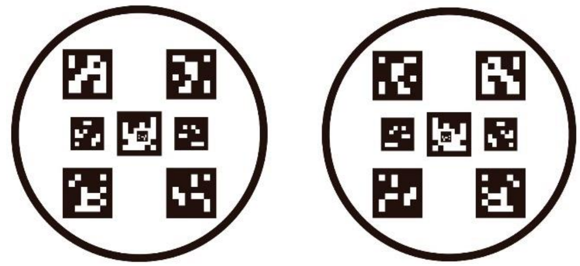

Figure 8 - Helipad setup, original vs mirrored

E. Tracking strategy

This paper uses two vector methods to be able to centre the drone with the marker. The Translational vector method was used for finding the centre point angle between the drone's camera and marker, as seen in Figure 9. The alpha value calculates the angle between the centre of the drone and marker [16,17]. This is calculated using Equation 1. A Closed Loop (CL) PID controller as seen in Figure 10 is used to recenter the drone's X and Y axis in respect to the marker's position inside the camera frame [7].

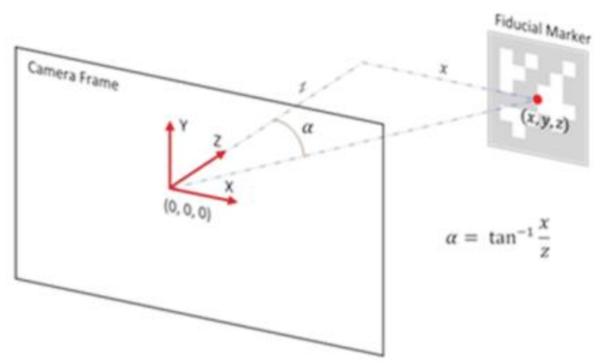

Figure 9 - Centre Point angle calculation [16]

$$\alpha = \tan^{-1}\left(\frac{x}{z}\right) \qquad [1]$$

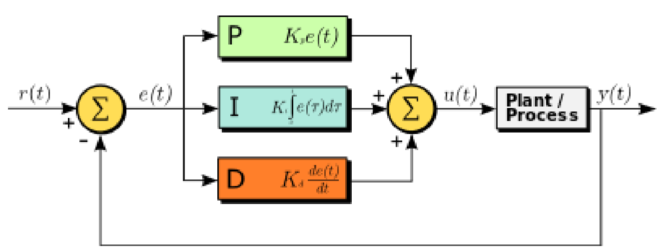

Figure 10 - Closed Loop PID controller

The other method used for finding the yaw centre angle was computed by using the Rodrigues vector method, as seen in Figure 12. The method is first converted to the Euler matrix form by using RQ decomposition. This then provided the Euler angles about the X, Y and Z axis for the marker and only the Z axis rotation matrix was used in this paper as shown by Equation 2 [7]. A CL PID controller was also implemented to allow for smooth angle adjustments between the drone's Z axis and the marker.

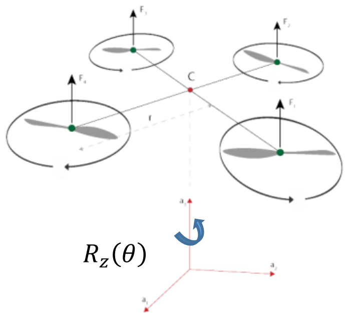

Figure 12 - R_z Yaw angle calculated

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [2]$$

IV. RESULTS

The paper's main focus with tracking was to achieve accurate and reliable data readings. Keeping performance in mind, the drones video feed kept a constant frame rate of 30 fps, but occasionally the frame got distorted, as seen in Figure 13. When a distorted frame got read, a reading of "??" would be outputted to the drone's window and then disregarded. The distorted frame issue enforced the need for the drone to have to be programmed to move at slower speeds, this reduced the number of frames that potentially got missed. Timestamp 15s shown in Figure 14 show no change. directly dependent on the capabilities of the camera

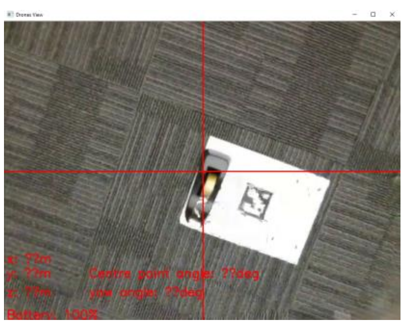

Figure 13 - A distorted drone's view

Figure 11 shows that the proposed method is able to detect, track, and land on a fiducial marker autonomously. It shows 4 frames of the drone aligning, descending, and landing autonomously. Frame 1 is in the top left and frame 4 is in the bottom left. The physical dimension of the fiducial marker used in the experiment is 7 by 7 centimetres. The drones FPV camera is able to detect the marker from up to 3 metres away. But the drone's video feed got distorted more often at the higher heights, this resulted in less accurate results being computed. This was fixed by setting the initial height to a lower value, this was set to 40cm as seen in Descent graph of Figure 14, this achieved the best results.

Figure 14 shows 3 plots showing real data that is computed by the Tello drone's FPV camera shown in Figure 11. The graph's starts off at the 15s mark as the drone's camera has a 10-15s setup window after take-off. The first plot graphs the descent path of the drone. The second displays the PID controller's input arguments. The last displays the output from the PID controllers. Two PID controllers were implemented to

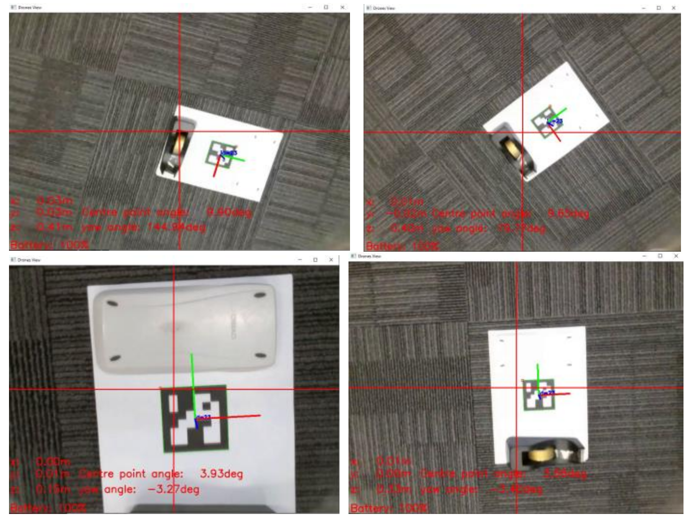

Figure 11 - 4 frames from the drones FPV camera showing it aligning, descending, and landing

manoeuvre the drone smoothly to position the marker in the centre of the drones view as seen in frame 4 of Figure 11. The controller used two separate sets of gains, one for controlling the Yaw-axis angle and the other for controlling the Centring Point angle. Both controllers were specified with 3 gains, a setpoint value of 0° and had a sampling rate of 0.1s. The Yaw-axis gains were P=1.5, I=0.01, D=0.3. The Centring point gains were P=0.6, I=0.01, D=0.06. These gains were tuned by using the Ziegler-Nichols Method as well as changing the controller type (P, PI, PID) [8,21]. A limitation by using this method was that since the 'Process' block of the PID controller only got tested on the real system and the 'Process' EOM equation was not derived. This meant the optimal stable solution could not be found.

When a marker is detected as seen in Figure 11, the drones window view outputs the X, Y, and Z axis distances in metres, as well as the Yaw and Centre Point angles in degrees. For the drone to be able to descend towards the fiducial marker, both the Yaw Angle and the Control Point Angle must be within the $\pm 5.0°$ area margin, graphically shown by the green lines in Figure 14. Table 1 shows the frame results from Figure 11. The first two frames shows the height staying constant at 40cm and both the Yaw Angle and Control Point Angle is outside the $\pm 5.0°$ area margin. This is seen graphically in Figure 14, where the 16 and 25.1 second time stamps are on the left hand side of the red lines. The drone can be seen descending in Frames 3 and 4 as the Yaw Angle and Control Point Angle is inside the $\pm 5.0°$ area margin. This is seen graphically in Figure 14, where the 28.4 and 30 second time stamps are on the right hand side of the red lines. The graphical plots below also shows the descending stage to be at the 27.2 second mark which is dictated by the red lines. The plots also shows that the drone successfully landed on the marker at the 32.3 second mark. The Centring Point angle finished up with an error margin of $+3.9°$ and the Yaw angle finished up with $-3.2°$. These results show that tracking algorithm worked as intended, except for when the drone's camera got distorted or an external disturbance occurred, thus causing the drone to wobble and lose course.

Table 1 - Real camera feed data

Frame	Time Stamp (s)	Centre Point angle (deg)	Yaw angle (deg)	Height (cm)
1	16.0	9.9	144.9	40.0
2	25.1	9.65	79.77	40.0
3	28.4	3.58	-3.4	33.0
4	30	3.93	-3.27	15.0

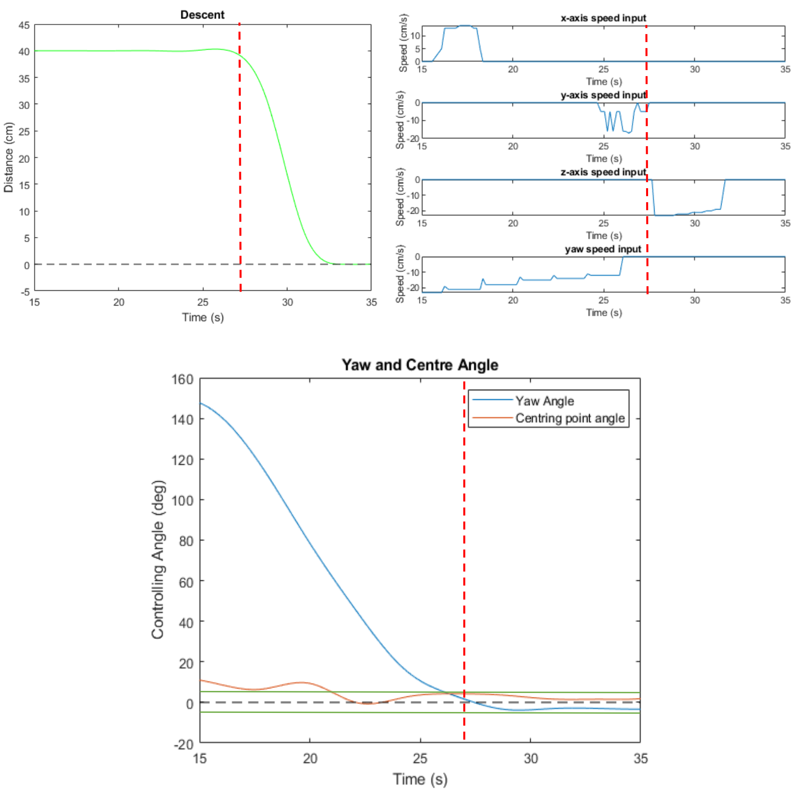

Figure 14 - Plotting the Descent, input axis speeds, Yaw and Centring Point Angle data from the PID controller

V. CONCLUSION

The proposed method successfully detected, tracked, and landed on a fiducial marker with using a DJI Tello drone. This paper chose the template matching approach [7,8,9] over the HSV colour space [1,2,3] and Corner detection [6] approaches due to prioritising accuracy over performance. To find the orientation of an object using the HSV method is harder to implement and prone to giving false detection. The method is also prone to giving false detections due to external noises and disturbances. This paper also looked at the option of implementing a Corner detection method. The method is extremely fast but its accuracy is highly dependent on having a good quality camera. Due to these limits, this paper chose to implement a template matching technique instead. Template matching improves on these issues and increases the chance of developing a successful experiment. Using fiducial markers allowed the algorithm to run on a low cost DJI Tello drone at a constant frame rate of 30fps. Fiducial markers also work in any unknown environment as long as the algorithm is able to detect the marker's unique ID.

This paper printed the fiducial marker to be in a medium to small scale form. This allowed the drone to reliably detect, track, and land on a marker from 40 centimetres to 3 metres away. Past research papers had worse performance, were less efficient and had similar error margins [16]. This was improved by modifying their algorithm, buying new hardware, and reviewing their mistakes. Some further tweaks allowed for the error margin to be minimised from $\pm 10.0°$ to $\pm 5.0°$.

A. *Future Research*

Fiducial markers can be vulnerable to motion blur, especially when used on a drone. Various types of fiducial markers exist, with circular markers being the most resistant to motion blur [8]. Detecting and tracking an object is highly determined by the image processing method that is used. Overall, the primary goal of the paper was to also prioritise accuracy over performance.

In the future a Kalman filter will be used to further improve the algorithm's performance, increase the accuracy of the tracking algorithm, and to reduce any noisy measurements [8,14,15]. An improvement on the PID controller would be desired as at the moment an inefficient method is in place for finding the gains [8]. To improve this, a theoretical method would be implemented so gains could be tried and tested on a simulator. A sensor would also help improve stability. Dynamic markers could also be implemented to improve accuracy [19,20].

VI. REFERENCES

[1] It Nun Thiang, Dr.LuMaw, Hla Myo Tun. "Vision-Based Object Tracking Algorithm With AR.Drone". International Journal of Scientific & Technology, Volume 5, Issue 06, pp. 135-139 June 2016.

[2] Chi-Tinh Dang, Hoang-The Pham, Thanh-Binh Pham and Nguyen-Vu Truong, "Vision based ground object tracking using AR.Drone quadrotor," International Conference on Control, Automation and Information Sciences (ICCAIS), Nha Trang, 2013, pp. 146-151.

[3] Ioannis Dramouskas, Isidoros Perikos, and Ioannis Hatzilygeroudis. "A Method for Performing Efficient Real-Time Object Tracing for Drones". Advances in Smart Systems Research, Vol. 6, pp. 55-64, 2017.

[4] "HSV vs. RGB". Internet: https://handmap.github.io/hsv-vs-rgb/, 26 Oct 2016, [Accessed 11/04/21]

[5] "Introduction to Harris Corner Detector". Internet: Introduction to Harris Corner Detector | by Deepanshu Tyagi | Data Breach | Medium, 16 Mar 2019, [Accessed 16/04/21]

[6] Cosimo Patruno, Massimiliano Nitti, Ettore Stella, and Tiziana D'Orazio. "Helipad detection for accurate UAV pose estimation by means of a visual sensor.". International Journal of Advanced

Robotic Systems (IJARS), Vol. 14, pp. 1-15, Sept. 2017.

[7] Ziwen Jiang. “An Autonomous Landing and Charging System for Drones.” M.Eng. thesis, University of California, Berkeley, 2019.

[8] Christoffer Karlsson. “Vision based control and landing of Micro aerial vehicles.” B.Eng. thesis, Karlstad University, Värmland , 2019.

[9] Liu X, Zhang S, Tian J, Liu L. An Onboard Vision-Based System for Autonomous Landing of a Low-Cost Quadrotor on a Novel Landing Pad. Sensors. 2019

[10] ‘Tello”. Internet: https://www.ryzerobotics.com/tello, [Accessed 21/03/21]

[11] “Tello SDK 1.3.0 documentation”. Internet: https://terra-1-g.djicdn.com/2d4dce68897a46b19fc717f3576b7c6a/Tello%20%E7%BC%96%E7%A8%8B%E7%9B%B8%E5%85%B3/For%20Tello/Tello%20SDK%20Documentation%20EN_1.3_1122.pdf [Accessed 21/03/21]

[12] “About OpenCV”. Internet: https://opencv.org/about.html, [Accessed 14/03/21]

[13] “Detection of ArUco Markers”. Internet: https://docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html, [Accessed 10/03/21]

[14] M. F. Sani and G. Karimian, "Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors". 2017 International Conference on Computer and Drone Applications (IConDA), Kuching, 2017, pp. 102-107.

[15] Abdelkrim Nemra, Luis M. Bergasa, Elena López, Rafael Barea, Alejandro Gómez and Álvaro Saltos, "Robust Visual Simultaneous Localization and Mapping for MAV Using Smooth Variable Structure Filter". Advances in Intelligent Systems and Computing 417, Switzerland, 2015, pp. 557-569.

[16] Daniel de Gouw and Richard Green, “Tracking Fiducial Markers with a Drone”, Computer Vision Lab, University of Canterbury, Tech. Rep., May 2019

[17] James Fairbairn and Richard Green, “ Automatic UAV landing”, Computer Vision Lab, University of Canterbury, Tech. Rep., April 2015

[18] “Mirror Image: Reflection and Refraction of Light”. Internet: https://www.livescience.com/48110-reflection-refraction.html#:~:text=A%20reflection%20appears%20to%20be,direction%20from%20which%20it%20hit. [Accessed 21/03/21]

[19] Raul Acuna and Volker Willert, “Dynamic Markers: UAV landing proof of concept”, Computer Vision Lab, University of Tulsa, Tech. Rep., Jan 2019

[20] F Sani, Moe & Karimian, Ghader. “Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors.”, 2017, pp. 102-107.

[21] “PID Controller-Working and Tuning Methods”. Internet: https://www.electronicshub.org/pid-controller-working-and-tuning-methods/ [Accessed 11/04/21]