Positional determination for UAV Precision Landing

Neil Fernandes, Josh Cai, Stephen Wang, Samuel Sun, Lucas Arzoumanian. *Members, Project KITE.*

***Abstract*—Altitude and orientation state tracking is imperative to safe, controlled, and repeatable landings of all unmanned aerial vehicles (UAVs). This paper presents an analytical and empirical analysis of state estimation based on the Extended Kalman Filter (EKF) fusion method using data collected from the following sensors: a camera, a barometer, a time-of-flight (ToF), and 2 different inertial measurement units (IMUs). The ground-truth of the drone's state is determined via motion capture while performing several test cases of the UAV being present in various scenarios. The main goal of this work is to develop a good EKF configuration that can integrate noisy measurements from the modalities the sensor provide, and to determine what combination of sensors produce the most reliable state estimation.**

***Index Terms*—Unmanned Aerial Vehicles, Pose Estimation, Precision Landing, Pattern Recognition**

I. Introduction

PROJECT Kite. is envisioned by a group of 5 fourth year Mechatronics Engineering students who designed and built a Vertical Take Off and Landing (VTOL) UAV using Cheap Off The Shelf (COTS) parts and a custom chassis. This UAV acts as an autonomous inspection system for insulators present on high power transmission lines. The drone possess several COTS parts that creates the sensor suite which helps the UAV estimate its altitude from the ground. These include Inertial Measurement Units (IMUs), a Barometer, and a Time-of-Flight (ToF) sensor. The drone also has a Raspberry Pi Camera Module installed for visual odometry. Currently, the UAV uses the IMU, Barometer and ToF for altitude holding while the drone is in flight. The Raspberry Pi camera is solely used for precision landing. This is done via a pose estimation using an ArUco marker and functions from OpenCV [1]. Figure 1 shows how the standard orthogonal axes would be drawn from the camera's perspective. This estimation from the camera alone is good, but easily susceptible to various forms of obstructions at the same time. This leads to gaps in the continuous stream of measurements, which could decrease landing accuracy. This project attempts to improve the detection accuracy and reliability by fusing other onboard sensors that use different modalities.

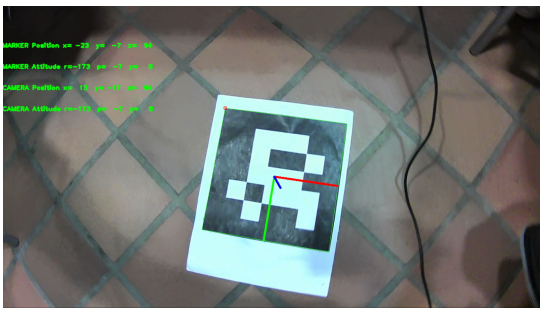

Fig. 1. Screenshot of ArUco location estimation visualization.

A. Related Works

UAVs have been applied in various fields in recent years, such as military strikes, emergency rescue, and geological exploration, mainly due to their low cost and convenient operation as an autonomous system. A successful utilization of unmanned aerial VTOL vehicles in missions that require a high degree of autonomy necessitates accurate and fast updated measurements from the onboard sensors used for both navigation and localization [2]. This also extends to the landing section of the VTOL which is a very important step in protecting the drone as well as the environment external to the UAV [3]. For our implementation of the VTOL glider aircraft, the system would return to a dedicated docking system designed to protect and charge the VTOL aircraft if the system is currently low on battery, or is currently in a high risk environment (examples include being in a dust or rain storm). One challenge that the system faces is a robust and cheap (both computationally and financially) method to make the UAV system land in a precise manner. This also has to be done in a way that the small form factor of the drone's base is not compromised.

In the area of precision landing for UAV technologies, there exist several approaches relying on helipads with either an H or T on them that use pre-trained neural networks to identify the letter and then, using the known geometry of either the letter or the surrounding circle, estimate the 6-DOF pose relative to the landing pattern [4] [5]. A shortcoming of these kinds of methods is losing information on the landing pattern during the approach. Based on the size of the landing pattern and the field-of-view (FOV) of the camera, the quad-rotor will no longer be able to see the landing pattern and be unable to estimate its pose. Cotta et. Al. considered using a vision-based photogrammetric position and attitude sensor (SVGS) to support the precise automated landing of a UAV from an initial altitude above 100 m to ground, guided by an array of landing beacons [6]. SVGS information is fused with other on-board sensors at the flight control unit to estimate the UAV's position and attitude during landing relative to a ground coordinate system defined by the landing beacons [6]. However, this method relies on two different ground stations and several RF readers which meant that the team had to invest more time and resources into the base system, further increasing the already large form factor of the base. The method also relies on several smartphones placed externally, surrounding the base. This poses an environmental risk wherein the base (placed in a remote location with minimal supervision) would cause damage to the SVGS, thereby destroying the overall system at hand.

Researchers have also turned towards using markers to estimate the altitude and pose of the UAV in space. This is

done by calculating the relative height of the UAV with respect to the marker, either on a static or dynamic object. Marut et. Al. utilized ArUco markers to extract pose estimations in order to determine the height of a UAV during the touchdown with high accuracy [7]. De Corso et. Al. took this idea one step ahead and implemented a multi-camera setup to estimate and fuse multiple pose extractions from a board containing multiple ArUco tags [8]. However, these experiments were done indoors and the estimations can be poorly done if the cameras used cannot see the ArUco tag (an example being in a dimly lit area). The methodologies were also not cost efficient and used expensive hardware (Intel Core-i7 processor with an NVIDIA MX Graphics Card) to compute the data streams, something the project did not have the luxury of.

There have been implementations where a GPS-based navigation and landing system has been used to provide a UAV with pose estimation of where the vehicle is with respect to the goal pose [9]. There is also literature suggesting the usage of laser scanners [10] [11], monocular and stereo cameras [12] [13] and RGB-D sensors [14]. However, all these approaches rely on a single exteroceptive sensing modality that is only functional under certain environmental conditions. For example, laser-based approaches require structured environments and cannot handle variations in terrain and vision-based approaches demand sufficient lighting and features. This makes them prone to failure in large-scale environments, in which the environment can change significantly. It is clear that in such scenarios, multiple measurements from these exteroceptive sensors may be available, and the fusion of all these measurements yields increased estimator accuracy and robustness.

Our implementation takes this methodology to fuse the data coming from the IMU, Barometer, and Time-of-Flight (ToF) sensors along with the visual odometry coming in from an off-the-shelf USB Webcam. This was done by virtue of using an Extended Kalman Filter (EKF). The core aim of this work is to develop a modular approach for integrating noisy data from diverse sensors, providing smooth and consistent positional estimates to aid precision landing of the VTOL on a designated pad present on the dock.

In the next section, an introduction of the system, followed by the sensor characterizations and methodology of sensor fusion will be discussed. In Section III, the results and a brief discussions on the findings will be provided, followed by the Conclusion in Section IV and Future Work in Section V.

II. Materials & Algorithmic Formulation

We chose to define an EKF was created to capture the different states of the UAV. This section contains a detailed breakdown of the state transition and sensor model, along with the description of the experimental setup and test procedures.

A. Experimental Setup

To simplify the data collection process, a minimal test jig comprising the selected sensor suite and an Arduino Nano was constructed. The versatile microcontroller was used in the setup because the sensors in the suite use both I^2C and UART, so it was very convenient to perform data acquisition on the microcontroller and log its serial output with a laptop, where they can be synced with the data from camera inference. The sensors used on the experimental setup included the NexiGo USB Webcam for the video feed used to extract visual odometry [15], AltIMU-10 v6 to provide IMU and Barometer data [16] and the TFMini-S to provide altitude data[17]. To ease the process of testing our implementation of the sensor fusion, a test bench was created wherein all the sensors were connected by means of a breadboard. All the sensors except the camera, were connected and powered via an Arduino Nano [18] which was connected to a computer solely used to collect the data. The USB WebCam was used instead of the Raspberry Pi Camera Module to help in easing the process of data acqusition. The usage of the Pi Camera would have meant that the test bench would have had to compromise an SPI connector which was not available at the time of the experiments. With that being said, the USB camera was easily connected to the same computer as the different sensors for the same reason. Figure 2 shows the complete setup described above.

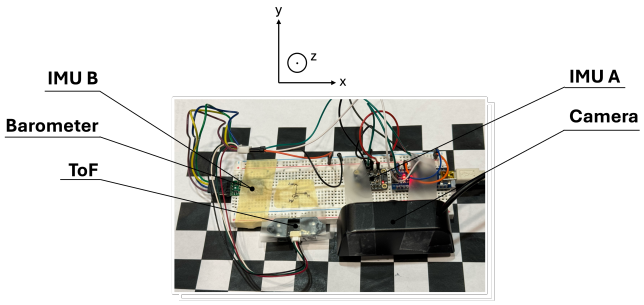

Fig. 2. Experimental Testbench

Figure 3 shows the experimental setup that was done in the Neuromechanics and Assistive Robotics Lab (Arami Lab)

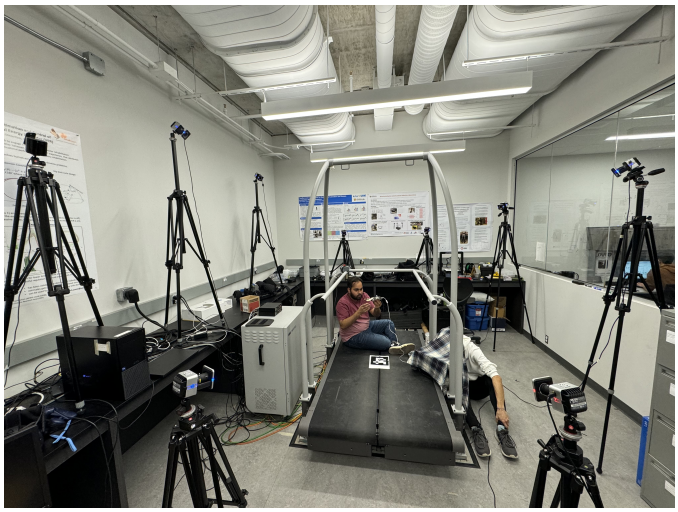

Fig. 3. Experimental Setup in the Arami Lab

B. EKF States

The states of the EKF were chosen to represent all six degrees of freedom and the corresponding first and second derivatives, resulting in displacement, velocity, and acceleration for both position and angle. This approach tallies up to 18 states:

$$p = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T$$

$$\phi = \begin{bmatrix} \alpha & \beta & \gamma & \dot{\alpha} & \dot{\beta} & \dot{\gamma} & \ddot{\alpha} & \ddot{\beta} & \ddot{\gamma} \end{bmatrix}^T$$

$$\hat{x} = \begin{bmatrix} p \\ \phi \end{bmatrix} \tag{1}$$

C. Process Model

The process model for a rigid body is given by (2) and was derived using basic kinematics equations. It is assumed that the UAV can be represented as a rigid body, and that the motion during landing is gentle enough such that centripetal acceleration induced by rapid rotation does not affect the accelerometer measurements.

$$F = \begin{bmatrix} I_{3\times3} & TI_{3\times3} & 0.5T^2I_{3\times3} \\ 0_{3\times3} & I_{3\times3} & TI_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}$$

$$\hat{x}_{k+1} = \begin{bmatrix} F & 0_{9\times9} \\ 0_{9\times9} & F \end{bmatrix} \hat{x}_k \tag{2}$$

D. Sensor Model

The sensor model describes how the EKF state space to the measurement space. In other words, how measurements can be reconstructed from the internal states. This section breaks down such sensor model, which is nonlinear for this project, for each sensor. The nonlinear nature of the sensors and the rotation from the inertial frame to the accelerating body frame makes it necessary to use an EKF rather than a linear KF. The complete sensor measurement vector $\hat{z}$ consists of a 17×1 column matrix and is broken up for readability reasons. Its corresponding Jacobian that linearizes around the estimated mean of the states is also derived.

The measurements relating to the camera can be seen in (3). Here x, y and z related to the state of the UAV.

$$[camera_{3\times1}] = \begin{bmatrix} camera_x \\ camera_y \\ camera_z \end{bmatrix} = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3}$$

The measurements from the ToF and Barometer sensors can be seen in (4) and (5).

$$ToF_z = \frac{z}{cos\alpha \times cos\beta} \tag{4}$$

$$Barometer_p = P_b e^{cz} \tag{5}$$

In both equations, z corresponds to the state and not the measurement. In (5), the constant c is given by

$$c = \frac{-gM}{RT}$$

where,

$$T = 293.15\ K$$
$$R = 8.3144598\ \frac{J}{mol.K}$$
$$M = 0.028964425\ \frac{Kg}{mol}$$
$$g = 9.80665\ \frac{m}{s^2}$$

As mentioned earlier, there are two IMUs aboard the experimental setup. (6) shows the mapping between the IMU measurement and states from the IMUs.

$$[IMU1_{6\times1}] = \begin{bmatrix} IMU1_{\ddot{x}} \\ IMU1_{\ddot{y}} \\ IMU1_{\ddot{z}} \\ IMU1_{\dot{\alpha}} \\ IMU1_{\dot{\beta}} \\ IMU1_{\dot{\gamma}} \end{bmatrix} \tag{6}$$

Splitting (6) into 3×1 rows (first three and last three) for readability, the corresponding models for each of them can be seen in (7) and (8)

$$\left[\begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix}\right]^T \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} + \begin{bmatrix} CC \\ CC \\ CC \end{bmatrix} \tag{7}$$

$$\left[\begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix}\right]^T \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} + \begin{bmatrix} CC \\ CC \\ CC \end{bmatrix} \tag{8}$$

where CC pertains to the Calibration Constant determined for the sensors. This constant is calculated by measuring the mean value of the sensors when the system is at rest, which accounts for any steady-state error in the sensor measurements.

For the second IMU, the sensor modelling is the exact same thing as the first IMU (with the same constants and states). This would result in obtaining $[IMU2_{6\times1}]$

Putting equations (3)-(8) together, we get the complete sensor measurement vector as shown in (9), mapping the UAV states to the individual sensor measurements.

$$\hat{z} = \begin{bmatrix} camera_{3\times1} \\ ToF_{1\times1} \\ Barometer_{1\times1} \\ IMU1_{6\times1} \\ IMU2_{6\times1} \end{bmatrix} \tag{9}$$

E. Covariances Matrices

The covariance matrices Q and R were chosen with a mix of calculations and trial-and-error. Initial values are set before running the EKF and adjusting accordingly. Using basic kinematics equations, displacement and velocity variances can be calculated as functions of time step and acceleration variance, the latter of which can be set to any initial value.

$$var(\dot{x}) = var(\ddot{x})T$$

$$var(x) = var(\dot{x})T + var(\ddot{x})0.5T^2$$

The final Q matrix is then an 18×18 identity matrix with diagonal elements corresponding to the above equations, depending on if the state is a displacement, velocity, or acceleration. The covariance values in R (17×17) were initially chosen based on the variability and accuracy of each sensor. Accelerometer and gyroscope measurements are far noisier and more prone to error than any other, whereas the camera height and ToF are consistent and accurate. As such, after initial guess and some tuning, the final R values along the diagonal are:

$$\begin{bmatrix}10[11]\ 0.1[11]\ 1\ 50[111]\ 20[111]\ 50[111]\ 20[111]\end{bmatrix} \quad (10)$$

F. Sensor Characterization

Two scenarios from the data collected against the Vicon system are selected as examples to show how the sensors themselves perform. The EKF output will be compared to its input in the next section. Figure 4 shows the height estimation output from the camera, ToF range sensor, barometer, and the Vicon ground truth. The barometer measurement was quite different from the other sensors, and will therefore be weighed less in the subsequent steps.

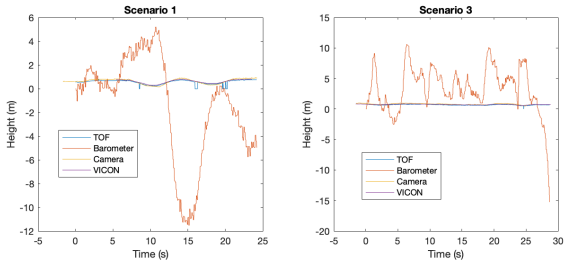

Fig. 4. Data collected from 4 height-measuring sensors.

Without the barometer, Figure 5 shows the remaining height-measuring sensors. A constant offset is applied to the camera measurement to create these plots. Figure 5 demonstrates that as far as height is concerned, there are enough sensors to reconstruct the true height of the jig.

G. Experiment Procedure

A more accurate, independent measurement system for the test jig is crucial for evaluating the performance of the algorithm. For this very reason, in collaboration with the Arami Lab, a set of five test movements were recorded while utilizing the Vicon Motion Capture system available at the lab. These test movements ranged from a simple vertical up and down

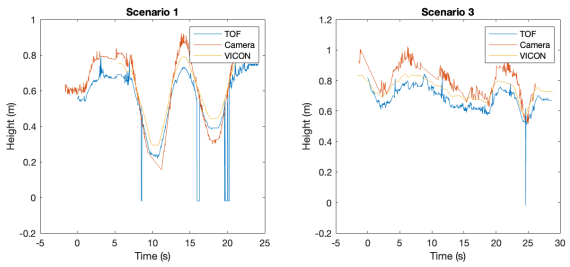

Fig. 5. Data collected from 3 height-measuring sensors.

movement to more extreme movements mimicking an unstable hover performed by the UAV, to an angled landing scenario. One of the test movements also included a rapid movement of the test bench going up and down to mimic the UAV rapidly rising and falling near the landing pad. Out of these movements, the two best measurements from the experiment was taken and used to analyze the performance of the proposed strategy as seen in Section III. To obtain the ground truth of the motion of the jig, i.e., the breadboard setup described in Section II-A, reflective markers were taped to various points of the jig so that the motion capture system can calculate and record the positions of those markers while sensor data from the system is collected. Once attached, the team, together with a PhD student, recorded both pose estimation (from the sensor fusion methodology described) and the motion capture system almost simultaneously.

Once recorded, the data from the motion capture system was post-processed to fill in any gaps that may have occurred during the recording step. This was done so that we could analyze a single coherent stream of ground truth that does not contain any missing data points. This step is also vital so that a more robust comparison can be done with how the motion was predicted with our proposed method, and how the test bench really moved at a specific time instance. Note that the time was not synchronized between the Vicon system and the computer that captured serial output from the Arduino nano; the offset in time was obtained on a case-by-case basis. The team did not have enough time to extract angle ground truth data from the motion capture system.

All test case scenarios and their captured videos can be seen in -https://bit.ly/4aBJbDx

III. Results & Discussion

In this section, we present the results obtained from our implementation of the Extended Kalman Filter. The performance of the algorithm is evaluated through three distinct test scenarios, where the test jig simulated aircraft landing in different motions. For each scenario, the data is processed with different permutations of sensor measurements to simulate external conditions that would make certain sensors unusable/unreliable. For example, one such permutation disregards camera and ToF measurements to simulate low-light environments. All permutations are shown in Table I.

Condition	Camera	Barometer	ToF
Low Light	X		X
Sloped Surface			X
High Winds		X	
Normal			

TABLE I
ALL SENSOR PERMUTATIONS

A. Scenario 1: Smooth Vertical Landing

This scenario shows a smooth up and down motion with minimal rotation, mimicking a landing sequence with a fixed heading, shown in Figure 6, Figure 7, Figure 8, and Figure 9. These plots show that the system performance is heavily dependant on certain sensors. When all sensors are used, height and angular estimates are passable while the X and Y positions are questionable at best. Those estimates are too inaccurate and too variable to be reliable. In the height estimate, there are obvious dips in the 15 - 20 second range. This is completely due to bad readings from the ToF, which can be proven by looking at the plots for when the ToF is removed: the dips are no longer present. Although brief, these dips can be catastrophic for the system when landing. This shows that the ToF should be filtered for outliers before passing through the EKF.

For other sensors, removing the barometer has little to no effect on the height estimate due to the relative variances between all the sensors that measure height. On the other hand, removing the camera completely ruins the X and Y positional estimates - the IMUs are not sufficient for estimating position.

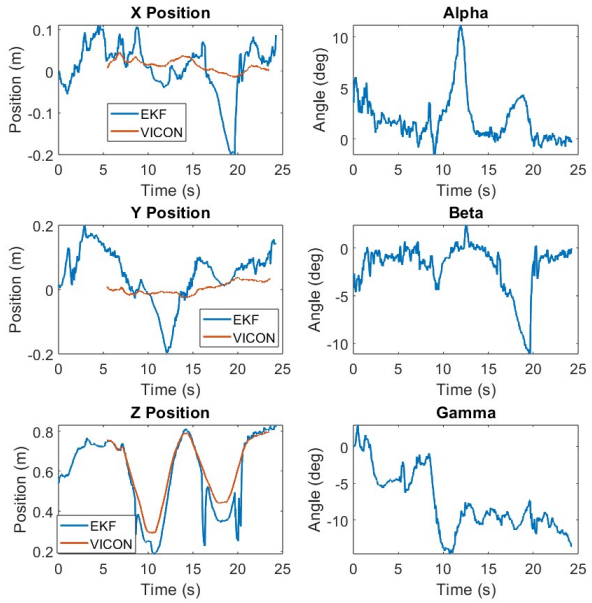

Fig. 6. Scenario 1: All sensors used for state estimation used

Aside from comparing against the ground truth, Figure 10 compares the EKF output with its inputs. Only the z state is shown as an example because multiple sensors could directly measure it.

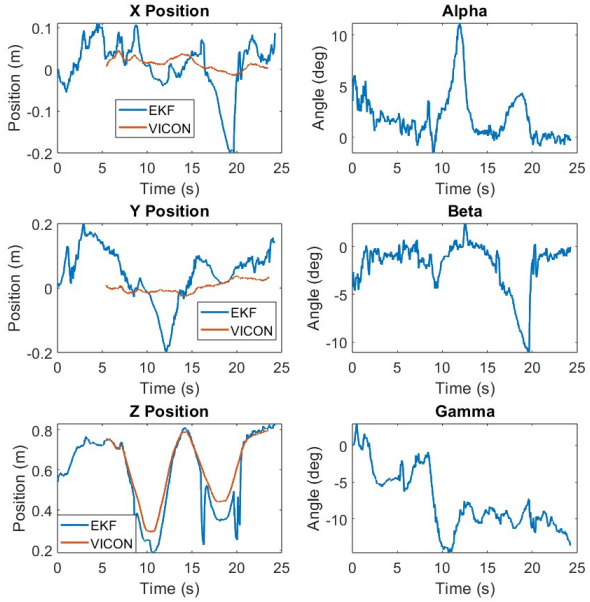

Fig. 7. Scenario 1: All sensors except for Barometer being used for state estimation

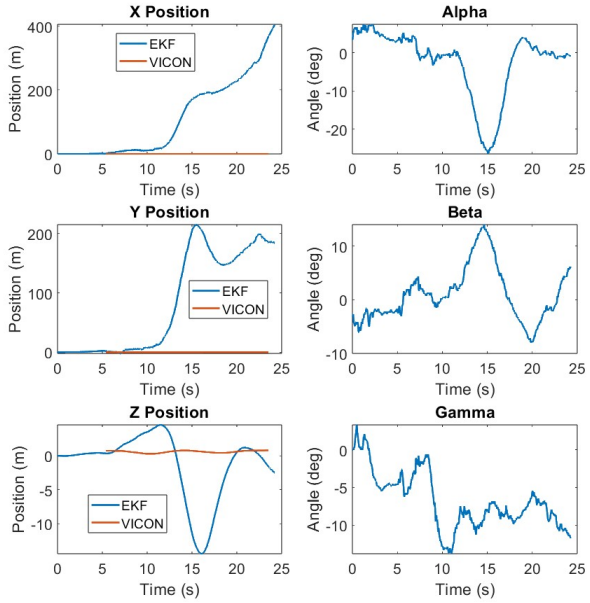

Fig. 8. Scenario 1: All sensors except for ToF and Camera being used for state estimation

In the EKF implementation that produced the aforementioned plots, the sensor data is duplicated, where applicable, so that they all refresh at the rate of the fastest sensor. Then a hypothesis was brought up in a discussion: could the duplicated data have an effect on the output? Using a multi-rate implementation provided by MATLAB ™Simulink, the prediction step of the EKF runs every 0.01 seconds, while each sensor runs at a integer multiple of the prediction period

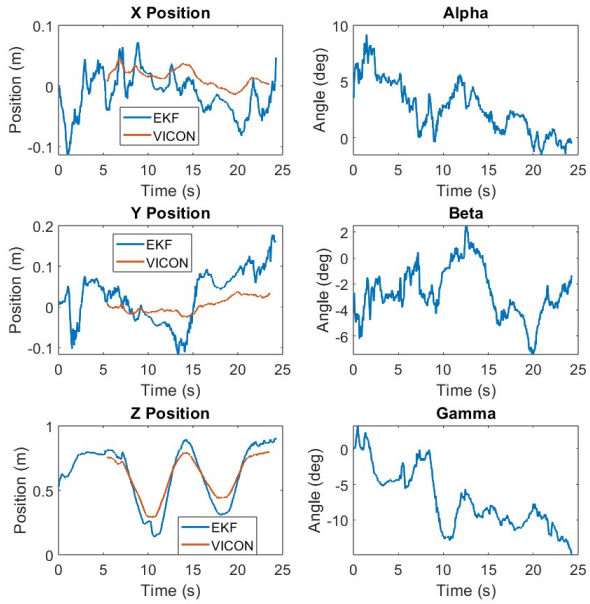

Fig. 9. Scenario 1: All sensors except for ToF being used for state estimation

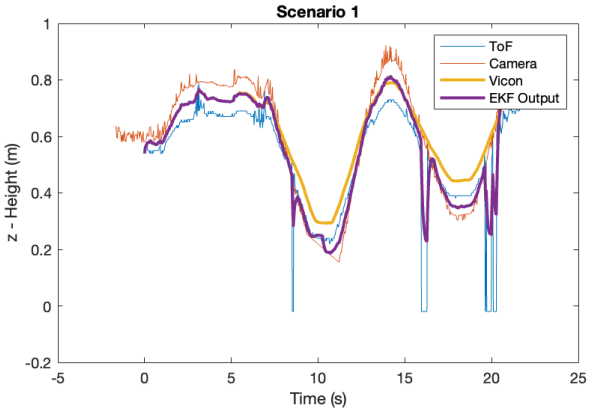

Fig. 10. The sensors that directly measure height is plotted against EKF output

independently. This approach produced Figure 11, showing only the three position states. Apparently, no improvement was made using this approach.

B. Scenario 2: Unstable Landing

This scenario mimicked an unstable landing procedure while the drone approached the landing pad, shown in Figure 12, Figure 13, Figure 14, and Figure 15. This includes several up and down motions along with approximately 90deg yaw turns. The overall system performance remains fairly consistent with the previous test scenario. When all sensors are available, height estimation works very well - good tracking and minimal noise. However, the X and Y position estimates fluctuate -0.2 and 0.2m. The mean estimate seems to be close to 0 but the variation is far too high to be considered reliable. The yaw estimate captures the overall movement of the system but without a ground truth, it is difficult to ascertain how well

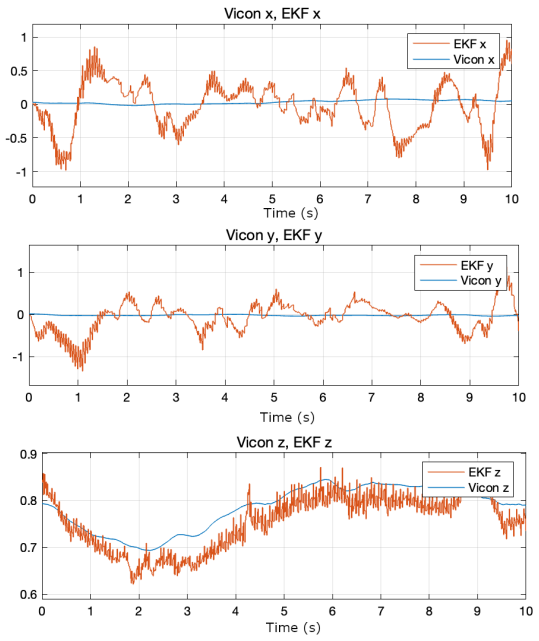

Fig. 11. EKF output when the prediction and update steps run at different rates

it tracks. Given that the EKF estimates a maximum yaw of just over 60deg when the system should have moved 90deg, it is likely that the estimate undershoots. The pitch and roll estimates seem to reasonably estimate a value around 0deg.

When certain sensors are taken away, the system either exhibits nearly no change or becomes completely unreliable. Without a barometer, the system is nearly identical. With multiple reliable sensors also measuring the height of the system (camera and ToF), it makes sense that the less reliable barometer would have very little effect on the estimated height. Removing the ToF has a much larger effect, seen in Figure 15 where the estimated value flattens out between 1 and 6 seconds. Chances are, the camera was either unable to get a measurement or got bad measurements during that time and the barometer alone could not adjust for that error. Once both the ToF and camera are taken out, the height estimate relies solely on the barometer, leading to wildly fluctuating and offset estimates (in the 10 - 20m range). The X and Y position end up relying on only the IMUs, also leading to very bad estimates. Angle estimates are barely impacted.

All this data seems to suggest that the ToF and camera are imperative to a decent estimation - the ToF specifically for height and the camera for X and Y. This also means that extra precautions should be taken to ensure that these sensors are always operational.

IV. Conclusion

The biggest takeaway is that this attitude reference system still needs work before it can be used practically. Assuming that all sensors are being used, the height estimate is generally very satisfactory. However, XY positional error can go as high as 30 or 40cm. Angle estimates are difficult to properly judge given the lack of ground truth despite looking reasonable.

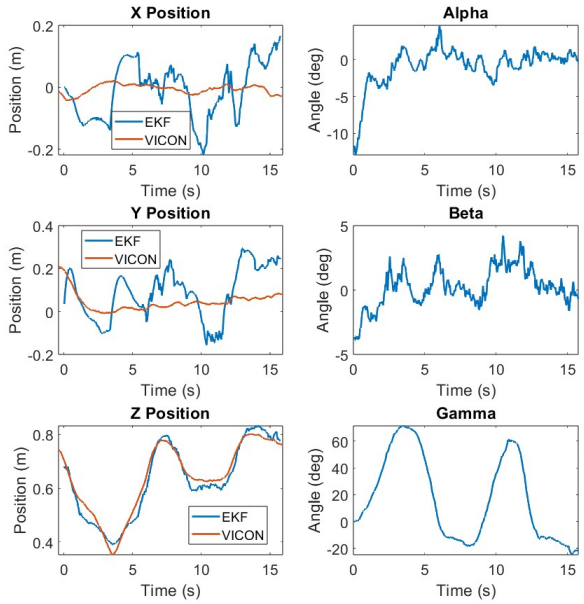

Fig. 12. Scenario 2: All sensors used for state estimation used

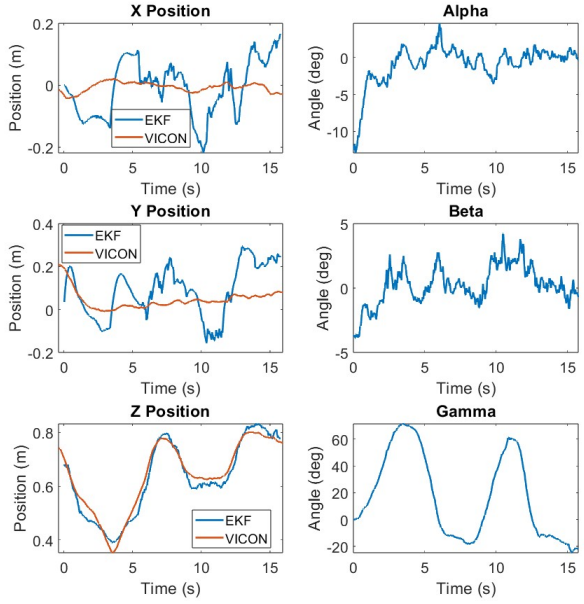

Fig. 13. Scenario 2: All sensors except for Barometer used for state estimation

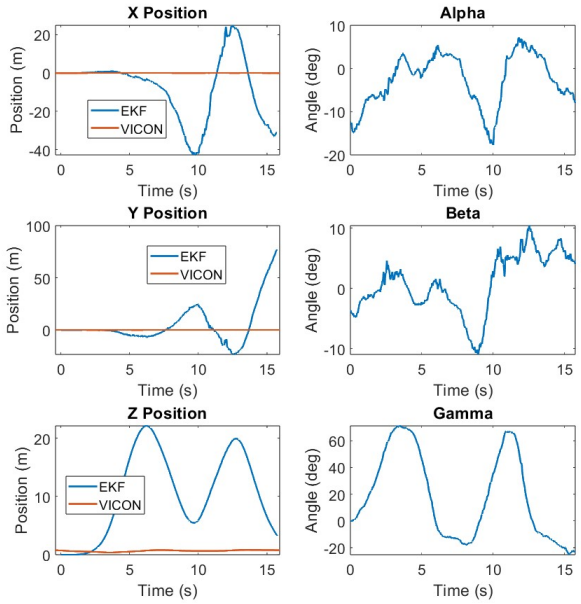

Fig. 14. Scenario 2: All sensors except for ToF and Camera used for state estimation

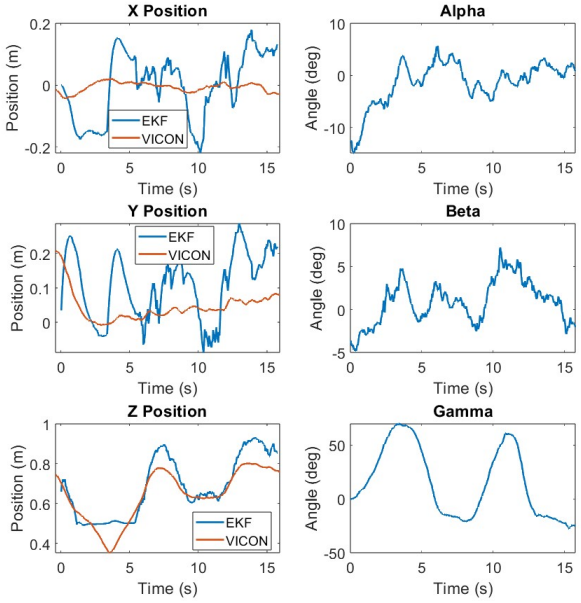

Fig. 15. Scenario 2: All sensors except for ToF used for state estimation used

In the event that the camera and ToF are unusable, the entire system becomes unusable. The IMU is insufficient and unreliable for good positional estimates. The barometer is also unreliable for the small variations in position present during a landing.

V. Recommendations

The biggest required improvement is in the XY positional estimate. There are several possible solutions worth looking into, specifically some pre-processing. Both the accelerometer and camera xy measurements are exceedingly noisy and should be low-passed before use in the EKF. In a similar vein, the ToF data should also be stripped of outliers before use. Despite good height estimations, the system is very sensitive to bad readings from the ToF (albeit rare).

More time should also be spent in parsing the Vicon data to extract angle ground truth. There are almost certainly better and more robust methods of sensor calibration that should be

explored. A GPS can be considered for additional position estimates. Finally, the process model should be reevaluated as it is a simplified model and does not account for interactions between linear and angular acceleration that arise due to centripetal forces.

Acknowledgment

The authors would like to thank Dr. Arash Arami & the Neuromechanics and Assistive Robotics Lab for their help in recording the ground truth data using their Vicon Motion Capture cameras.

References

[1] L. Arzoumanian, J. Cai, N. Fernandes, S. Wang, and S. Sun, "Project kite." April 2024.

[2] M. Al-Sharman, M. A. Al-Jarrah, and M. Abdel-Hafez, "Auto takeoff and precision terminal-phase landing using an experimental optical flow model for gps/ins enhancement," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, vol. 5, no. 1, pp. 011 001–1, 2018.

[3] Y. He, W. Wang, L. Mottola, S. Li, Y. Sun, J. Li, H. Jing, T. Wang, and Y. Wang, "Acoustic localization system for precise drone landing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4126–4144, 2024.

[4] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–380, 2003.

[5] G. Xu, Y. Zhang, S. Ji, Y. Cheng, and Y. Tian, "Research on computer vision-based for uav autonomous landing on a ship," *Pattern Recognition Letters*, vol. 30, no. 6, pp. 600–605, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865509000051

[6] J. L. Silva Cotta, H. Gutierrez, I. R. Bertaska, J. P. Inness, and J. Rakoczy, "High-altitude precision landing by smartphone video guidance sensor and sensor fusion," *Drones*, vol. 8, no. 2, p. 37, 2024.

[7] A. Marut, K. Wojtowicz, and K. Falkowski, "Aruco markers pose estimation in uav landing aid system," in *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2019, pp. 261–266.

[8] T. De Corso, L. De Vito, F. Picariello, K. Wojtowicz, A. Marut, and P. Wojciechowski, "Optical multi-camera uav positioning system via aruco fiducial markers," in *2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2023, pp. 352–357.

[9] R. Van Der Merwe, E. A. Wan, S. Julier, *et al.*, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *Proceedings of the AIAA guidance, navigation & control conference*, vol. 3. Providence, RI Providence, RI, 2004, p. 08.

[10] A. Bachrach, S. Prentice, R. He, and N. Roy, "Range–robust autonomous navigation in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[11] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 20–25.

[12] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 957–964.

[13] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4557–4564.

[14] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Robotics Research: The 15th International Symposium ISRR*. Springer, 2017, pp. 235–252.

[15] NEXIGO, "Nexigo n60 1080p webcam."

[16] Pololu, "Altimu-10 v6 gyro, accelerometer, compass, and altimeter (lsm6dso, lis3mdl, and lps22df carrier)."

[17] Benewake, "Tfmini-s - micro lidar module."

[18] Arduino, "Arduino nano."