

Kalman Filter-Based Drift Detection and Mitigation of Visual-Inertial Odometry in UAVs

Roman Ibrahimov, Teaya Yang, and Mark W. Mueller

Abstract—We present a conceptual framework for an autonomous safety mechanism designed to enhance the reliability of Unmanned Aerial Vehicles (UAVs) that use Visual-Inertial Odometry (VIO) for state estimation. As UAVs increasingly interact with the public, such safety mechanisms are crucial to reducing the likelihood and severity of accidents. VIO drift, which occurs when accumulated estimation errors cause discrepancies between the UAV’s perceived and actual position, poses a significant risk to safe operation. To address this challenge, we propose a Kalman filter-based approach for detecting VIO drift events. Upon detection, the envisioned safety mechanism is designed to adjust state estimation by integrating onboard gyroscope measurements and thrust commands for short durations, aiming to enhance stability and prevent potential crashes before initiating a controlled landing. While this framework provides the foundation for a real-time safety mechanism, the implementation and experiment focus on validating the drift detection component in an offline setting using real UAV flight data. The results demonstrate the effectiveness of the detection method in identifying VIO drift scenarios, highlighting its potential for future real-time applications.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become essential tools in a wide range of applications, from environmental monitoring and disaster response to commercial delivery services and recreational use [1]–[3]. A key component in the successful navigation of UAVs is state estimation, which enables the vehicle to determine its position and orientation in space [4]. Visual-Inertial Odometry (VIO) systems, which fuse data from cameras and inertial measurement units (IMUs), are commonly employed to provide accurate state estimation for UAVs operating in complex environments [5]. VIO combines visual data from onboard cameras, tracking environmental features, with inertial data from the IMU, which measures linear acceleration and angular velocity. By continuously integrating this information, VIO systems can estimate the UAV’s position and orientation, ensuring state estimation even in GPS-denied environments [6].

Despite the advantages of VIO systems, they are prone to drift, a phenomenon where small errors in system or in sensor measurements accumulate over time, leading to significant deviations in the estimated position [7]. VIO drift can pose serious risks to UAV operations, especially in scenarios requiring precise navigation, such as indoor environments or densely populated areas. In these cases, navigation errors can result in collisions, crashes, or loss

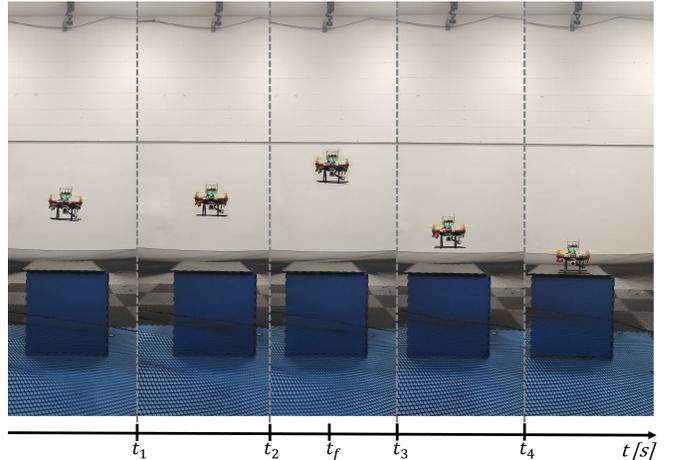


Fig. 1. Conceptual visualization of the safety mechanism: Experimental UAV testbed is hovering with a VIO state estimation. Until t_2 , the vehicle keeps a certain altitude. After t_2 , VIO experiences drift, and the vehicle’s altitude increases. At t_f , a fault (drift) is detected. The safety mechanism is triggered and the vehicle descends after keeping the hovering position for a short time. The image illustrates different phases of the vehicle’s states while flying and serves as a conceptual aid to depict the intended operation of the proposed framework.

of control, highlighting the need for effective drift detection and mitigation mechanisms.

Drift mitigation has been a central concern for VIO systems, especially in environments with few distinct visual features. Traditional approaches to drift detection rely on sensor fusion and loop closure methods. [8] presented a VIO system that uses loop closure detection to correct drift by revisiting previously observed landmarks and re-aligning the state estimate. Similarly, [5] developed a tightly-coupled method that leverages both IMU and visual measurements to enhance drift robustness in feature-sparse environments.

Other approaches involve incorporating additional sensors, such as LiDAR [9] or GPS [10], to provide additional state information. LiDAR-based odometry, for example, has been shown to complement VIO systems by providing accurate depth information, particularly in GPS-denied environments [11]. While these techniques are effective in improving state estimation accuracy, they come at the cost of increased system complexity, weight, and power consumption—factors that are particularly important for UAVs deployed in resource-constrained or cost-sensitive missions.

While much attention has been given to improving VIO performance, comparatively less effort has been placed on developing safety mechanisms that mitigate the consequences of drift. In critical applications like indoor in-

¹The authors are with the High Performance Robotics Lab, Department of Mechanical Engineering, University of California, Berkeley, CA 94709, USA. {roman.ibrahimov, teaya.yang, mwm}@berkeley.edu

spectations or search-and-rescue missions, drift can lead to catastrophic failures, including crashes or loss of control. To address this issue, several approaches have been proposed that focus on UAV fail-safe procedures.

The use of Kalman filters for drift detection and state estimation has proven effective in many UAV applications [12]. Kalman filters allow for real-time monitoring of sensor residuals, enabling the system to identify when state estimation becomes unreliable due to drift [13]. Several studies have explored the use of Kalman filters in conjunction with onboard sensors to detect anomalies in state estimation and trigger corrective actions [14]–[17].

However, many existing fail-safe mechanisms heavily rely on external data sources such as GPS or external motion capture systems for accurate state estimation. This reliance limits their effectiveness in GPS-denied environments. This highlights the need for autonomous safety mechanisms that can function independently of external infrastructure. Such mechanisms would allow UAVs to continue functioning effectively, even in environments where navigation precision is critical and external assistance is unavailable.

In this paper, we present a conceptual framework for an autonomous safety mechanism designed to detect and mitigate VIO drift in UAVs. Our approach proposes a Kalman filter-based method for detecting drift events, providing a reliable way to identify when state estimation is compromised. While the framework envisions a safety mechanism that could initiate a controlled landing procedure by integrating onboard gyroscope measurements and thrust commands, this study focuses only on the detection component. By shifting the focus from real-time drift correction to prioritizing a safe recovery, this approach aims to reduce the likelihood of crashes and improve UAV safety in environments where navigation precision is critical. This framework is particularly well-suited for GPS and motion-capture-denied environments, where external localization aids are unavailable. It is important to note that this study does not implement real-time control actions, nor does it attempt to navigate the UAV back to its home location.

II. KALMAN FILTER BASED DRIFT DETECTION

In this section, we introduce a Kalman filter-based fault detection framework for monitoring the VIO system's position estimates in the x , y , and z directions. The objective is to detect potential drifts in the VIO output using a residual-based approach. A chi-squared test statistic is then used to identify deviations from expected behaviour.

A. State-Space Model

We model the system using a discrete-time state-space representation. We use a simplified model that allows for easier interpretation of the results and facilitates the implementation. Additionally, the simplified model can ensure computational feasibility for onboard execution, where resources are often limited. The state vector $\mathbf{x}(k)$, representing the 3D position of the system, evolves according to the following state transition equation:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + w(k), \quad \mathbf{x}(k) \in \mathbb{R}^3 \quad (1)$$

where $\mathbf{x}(k) = [x(k), y(k), z(k)]^T$, \mathbf{A} , and $w(k)$ are the state vector at a time k , the state transition matrix, which we assume to be the identity matrix \mathbf{I}_3 for simplicity, and the process noise, modeled as zero-mean Gaussian noise with covariance \mathbf{Q} correspondingly.

The measurement is modeled as follows:

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + v(k), \quad (2)$$

where $\mathbf{z}(k) = [z_x(k), z_y(k), z_z(k)]^T$, \mathbf{H} , and $v(k)$ are the measurement vector at time step k , representing the VIO system's measured position, observation matrix, which we assume to be the identity matrix \mathbf{I}_3 for simplicity, and measurement noise, modeled as zero-mean Gaussian noise with covariance \mathbf{R} .

We consider the drifts in VIO state estimation as faults. To take this into account, (1) can be extended by adding a deterministic fault vector $f(k)$:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + w(k) + f(k) \quad (3)$$

The affect of the fault $f(k)$ on the measurements $\mathbf{z}(k)$ [18] is modeled with the following extended measurement model:

$$\begin{aligned} \mathbf{x}_f(k+1) &= \mathbf{A}\mathbf{x}_f(k) + w(k) + f_k(k), \quad \mathbf{x}_f(0) = 0, \\ &\begin{cases} f_k = 0, k < t_f, & \text{fault-free} \\ f_k \neq 0, k \geq t_f, & \text{faulty} \end{cases} \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbf{z}(k) &= \mathbf{C}\mathbf{x}(k) + f(k) + v(k), \\ f(k) &= \mathbf{C}\mathbf{x}(k), \end{aligned} \quad (5)$$

where t_f denotes the time instant when fault f takes place, \mathbf{x}_f and \mathbf{C} , which is also assumed to be \mathbf{I}_3 , represent the fault state vector and observation matrix correspondingly.

B. Kalman Filter

The Kalman filter [19] is used to estimate the system state based on the process model and noisy measurements. The predicted state and error covariance are computed as:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k), \quad (6)$$

$$\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^T + \mathbf{Q}, \quad (7)$$

where $\hat{\mathbf{x}}(k+1|k)$ is the predicted state estimate, $\mathbf{P}(k+1|k)$ is the predicted covariance matrix, and \mathbf{Q} is the process noise covariance matrix.

Upon receiving the measurements $\mathbf{z}(k)$, the Kalman gain $\mathbf{K}(k)$ is calculated:

$$\mathbf{S}(k) = \mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \mathbf{R}, \quad (8)$$

$$\mathbf{K}(k) = \mathbf{P}(k+1|k)\mathbf{H}^T\mathbf{S}(k)^{-1}, \quad (9)$$

where $\mathbf{S}(k)$ is the innovation or residual covariance. The state and covariance estimates are then updated as follows:

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \\ &= \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k+1|k)), \end{aligned} \quad (10)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k)\mathbf{H}\mathbf{P}(k+1|k). \quad (11)$$

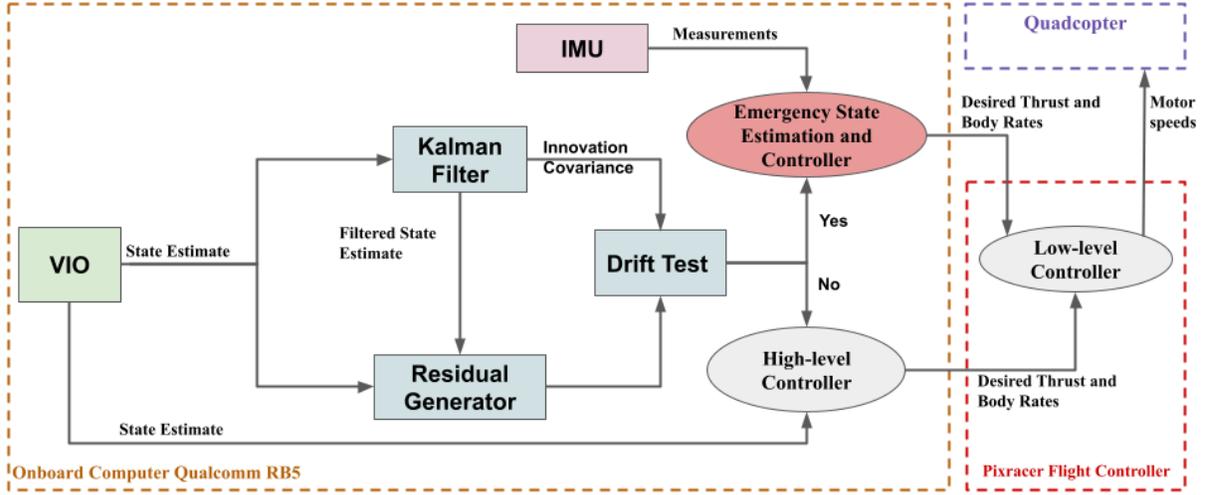


Fig. 2. Block diagram illustrating a possible implementation of the proposed framework on real UAV hardware: Qualcomm RB5 Computer that runs Kalman filter-based drift detection and onboard controllers; Pixracer Flight Controller that receives desired thrust and body rates from RB5 and generates motor speeds.

C. Fault Detection

We use the residual between the measured position and the predicted position to detect faults. The residual $\mathbf{r}(\mathbf{k})$ is computed as:

$$\mathbf{r}(\mathbf{k}) = \mathbf{z}(\mathbf{k}) - \mathbf{H}\hat{\mathbf{x}}(\mathbf{k} + 1|\mathbf{k}). \quad (12)$$

The residual represents the difference between the observed and predicted system states, and it serves as an indicator of how well the system model matches the actual measurements. Under normal conditions, the residual is expected to have zero mean with known variance and is driven primarily by noise.

To quantify the significance of the residual, we compute a test statistic $\mathbf{J}(\mathbf{k})$ at each time step. The test statistic $\mathbf{J}(\mathbf{k})$ measures the squared Mahalanobis distance of the residual $\mathbf{r}(\mathbf{k})$ relative to the residual covariance $\mathbf{S}(\mathbf{k})$, and it is defined as:

$$\mathbf{J}(\mathbf{k}) = \mathbf{r}(\mathbf{k})^T \mathbf{S}(\mathbf{k})^{-1} \mathbf{r}(\mathbf{k}). \quad (13)$$

We then follow the chi-squared test to assess whether the observed residual $\mathbf{r}(\mathbf{k})$ is consistent with the expected distribution under fault-free conditions. In this case, we assume that in the absence of faults, the residual $\mathbf{r}(\mathbf{k})$ is driven by Gaussian noise, and the test statistic $\mathbf{J}(\mathbf{k})$ is distributed as:

$$\mathbf{J}(\mathbf{k}) \sim \chi_n^2, \quad (14)$$

where χ_n^2 denotes the chi-squared distribution with n degrees of freedom. The chi-squared test is employed to determine whether the value of the test statistic $\mathbf{J}(\mathbf{k})$ exceeds a threshold \mathbf{J}_{th} , which is determined based on a pre-specified significance level α .

The threshold \mathbf{J}_{th} is derived from the cumulative distribution function of the chi-squared distribution:

$$\mathbf{J}_{th} = \chi_{\alpha}^2(n), \quad (15)$$

where the significance level α represents the probability of a false alarm (i.e., detecting a fault when none exists) and

$\chi_{\alpha}^2(n)$ is the critical value from the chi-squared distribution with n degrees of freedom, corresponding to the chosen α .

If the test statistic exceeds the threshold, a fault is detected:

$$\begin{cases} \mathbf{J}(\mathbf{k}) > \mathbf{J}_{th} \Rightarrow \text{fault detected} \\ \mathbf{J}(\mathbf{k}) \leq \mathbf{J}_{th} \Rightarrow \text{no fault detected} \end{cases} \quad (16)$$

III. EMERGENCY STATE ESTIMATION AND CONTROL STRATEGY

When a VIO drift is detected, we want the UAV to land safely without a severe accident under compromised state estimation. To achieve this, we refer to an emergency safety mechanism [20] that relies solely on onboard IMU. The emergency controller uses gyroscope measurements to estimate the vehicle's state and stabilize the UAV in hover for a defined period. After this short hover period, the UAV begins a controlled descent. The system reduces thrust progressively, ensuring a safe landing. This fail-safe mechanism does not attempt to recover VIO data but instead focuses on a safe and controlled landing once the VIO becomes unreliable.

A. Vehicle Dynamics

The UAV's dynamics are modeled as a rigid body, and the position, velocity, and acceleration in the inertial frame are denoted by x_I , \dot{x}_I , and \ddot{x}_I , respectively. The rotation matrix R_{IB} defines the orientation of the body-fixed frame with respect to the inertial frame, and the body rates $\boldsymbol{\omega}_B = (p, q, r)$ are the angular velocities about the body frame axes.

The vehicle dynamics are given by the following equations:

$$\ddot{x}_I = R_{IB} \mathbf{c}_B + g_I, \quad (17)$$

$$\dot{R}_{IB} = R_{IB} [\boldsymbol{\omega}_B \times], \quad (18)$$

$$I_{BB} \dot{\boldsymbol{\omega}}_B = I_{BB} \dot{\boldsymbol{\omega}}_B - [\boldsymbol{\omega}_B \times] I_{BB} \boldsymbol{\omega}_B, \quad (19)$$

where $\mathbf{g}_I = (0, 0, -g)$ is the gravitational acceleration in the inertial frame, $\mathbf{c}_B = (0, 0, c)$ is the total thrust force in the body-fixed frame, I_{BB} is the moment of inertia matrix, I_B is

the moment generated by the propellers' differential thrusts, $[\omega_B \times]$ represents the skew-symmetric 3×3 matrix form of the angular velocity vector $\omega_B = (p, q, r)$, and R_{IB} is the rotation matrix characterized by the Euler angles as follows [21]:

$$R_{IB}(\psi, \theta, \phi) = R_3(\psi)R_2(\theta)R_1(\phi) \quad (20)$$

The thrust c and moment l_B are functions of the individual propeller thrusts f_1, f_2, f_3, f_4 and can be expressed as:

$$l_B = \begin{bmatrix} l(f_2 - f_4) \\ l(f_3 - f_1) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix} \quad (21)$$

$$c = \frac{f_1 + f_2 + f_3 + f_4}{m_B}, \quad (22)$$

where κ is an experimentally determined constant that depends on the UAV's dynamic characteristics and system specifications.

B. Emergency State Estimation

In the event of significant VIO drift, the proposed framework can temporarily rely solely on the onboard Inertial Measurement Unit (gyroscope) for state estimation. During this period, the controller is designed to maintain the vehicle in a hovering position, allowing sufficient time to prevent potential crashes. The emergency controller is formulated to regulate the pitch angle θ to mitigate altitude and forward motion drift, while the yaw angle ψ is assumed to remain constant. The control strategy for roll ϕ is not explicitly detailed but follows an analogous approach. As highlighted in the previous subsection, the UAV dynamics are expressed in an XYZ coordinate system. The yawed reference frame, denoted x_Y , decouples the lateral and vertical dynamics, assuming the roll ϕ and pitch θ angles remain small.

We can rewrite (17) as \mathbf{g}_I lies on the axis of rotation \mathbf{R}_3 :

$$\ddot{x}_Y = R_3(\psi)^T R_2(\theta) R_1(\phi) \mathbf{c}_B + R_3(\psi)^T \mathbf{g}_I \quad (23)$$

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} \approx \begin{bmatrix} \sin \theta \cos \phi \\ -\sin \phi \\ \cos \theta \cos \phi \end{bmatrix} c + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (24)$$

We want to estimate the speed of the quadcopter v_x . We denote the onboard estimate of v as \hat{v} . We use a predict-correct estimator to estimate body rate directly from the gyroscope readings. Let Δt represent the time interval of the onboard loop, \hat{q}_0 denote the estimated gyroscope bias, $\hat{q}[k]$ be the gyroscope output at time step k , and I_{xx} refer to the moment of inertia of the body around the x-axis. The estimation is performed in two steps: prediction and correction:

$$\hat{q}^-[k+1] = \hat{q}[k] + \Delta t \frac{l(f_2 - f_4)}{I_{xx}}, \quad (25)$$

$$\hat{q}[k+1] = \lambda \hat{q}^-[k+1] + (1 - \lambda)(\hat{q}[k] - \hat{q}_0), \quad (26)$$

where λ is a filter parameter. We then use Euler integration method to integrate the rate estimate to compute the angle

estimate. The acceleration from (24) is integrated to get a velocity estimate:

$$\hat{\theta}[k+1|n] = \hat{\theta}[k|n] + \hat{q}[k]\Delta t \quad (27)$$

$$\hat{v}_x[k+1|n] = \hat{v}_x[k|n] + c[k] \sin(\hat{\theta}[k|n])\Delta t \quad (28)$$

where $\hat{\theta}[k|n]$ denotes the estimate at step k , with n indicating the external update number.

C. Emergency Controller

In this section, the control strategy for bringing the vehicle to a hover, specifically by reducing the lateral velocities to zero, is developed. Initially, the distinction between the true state and its estimate is ignored, and the effects of discretization are neglected. The controller is designed in continuous time. The equation (24) is linearized around the hover condition by considering $\theta \approx 0$ and $\phi \approx 0$. As a result of small-angle approximations, where $\sin \theta \approx \theta$, $\cos \theta \approx 1$, $\sin \phi \approx \phi$, and $\cos \phi \approx 1$, the equation (24) is simplified as follows:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} \approx \begin{bmatrix} \theta \\ -\phi \\ 1 \end{bmatrix} c + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}. \quad (29)$$

The emergency controller aims to drive the lateral velocities to zero, maintaining a stable hover. The aim for the vertical acceleration is to be zero, which means $c \approx g$. It leads to the following result upon differentiation:

$$\dot{c} \approx 0, \quad \ddot{v}_x \approx \dot{\theta}g. \quad (30)$$

The feedback control law for pitch angle θ is given by:

$$q_c = -2\zeta\omega_n\theta - \frac{\omega_n^2}{g}v_x \quad (31)$$

where ζ is the damping ratio, and ω_n is the natural frequency. The overall system behavior is:

$$\ddot{v}_x + 2\zeta\omega_n\dot{v}_x + \omega_n^2v_x = 0 \quad (32)$$

The actual angular rate q is controlled using:

$$\dot{q} = k_q(q_c - q) \quad (33)$$

Thus, the complete system dynamics are:

$$\frac{1}{k_q}\ddot{v}_x + \ddot{v}_x + 2\zeta\omega_n\dot{v}_x + \omega_n^2v_x = 0 \quad (34)$$

To ensure zero vertical acceleration in the inertial frame, the thrust command $c[k]$ is given by:

$$c[k] = \frac{g}{\cos \hat{\theta}[k] \cos \hat{\phi}[k]}$$

as derived from equation (24).

The emergency controller is not intended to operate for extended periods, as error accumulation would eventually render the state estimate unreliable. During the envisioned landing phase, the controller would transition to an open-loop mode, where thrust is gradually reduced without active feedback corrections. After maintaining a hovering position

for 2 seconds, the total thrust is conceptually reduced, allowing the UAV to descend at a controlled rate of 1 m/s^2 :

$$c = g - 1 \text{ m/s}^2. \quad (35)$$

Once landing is complete, the motors would be turned off, and the UAV would enter an idle mode. This process represents a possible implementation within the proposed framework.

IV. EXPERIMENTAL RESULTS

The proposed VIO drift detection mechanism was tested on logged flight data from a UAV hardware illustrated in Figure 3. The UAV integrates multiple key components, including a Pixracer R15 flight controller, a Qualcomm RB5 computing board, where the proposed mechanism can run, and a RealSense D455 camera. The desired thrust and angular velocity commands from Qualcomm RB5 are sent to the Pixracer R15 flight controller which hosts a low-level controller that translates thrust and angular velocity commands into corresponding motor speeds as shown in Figure 2. State estimation of the vehicle relies on the VIO-based estimator OpenVINS framework and it utilizes IMU and stereo images from the depth camera for state estimation [22].

During the experiments, the output of the "Drift Test" shown in Figure 4 is monitored exclusively during the steady flight stage, which is from $t \approx 52s$ to $t \approx 82s$, using logged flight data. This is because takeoff and landing involve rapid changes in the vehicle's state, which introduce fluctuations not relevant to drift detection.

We set $\alpha = 0.5$, meaning that the threshold is chosen at the median of the chi-square distribution with three degrees of freedom. This value was selected to balance sensitivity and robustness, ensuring that deviations exceeding typical variations in VIO state estimation are detected while minimizing false alarms. Furthermore, the fault detection algorithm runs at the same frequency as the VIO update rate.

To test the effectiveness of our proposed mechanism, we conducted flight tests in a lab environment where the walls lacked distinct visual features. Since OpenVINS depends on environmental visual cues for accurate state estimation, the absence of prominent landmarks led to VIO drift, where the vehicle's estimated position gradually deviated over time. This setup provided a controlled environment to induce drift naturally. The logged flight data from these tests was then used to assess the ability of our detection mechanism to identify VIO drift events in an offline setting.

The plot (a) in Figure 4 depicts the position estimates in x , y , and z coordinates, measured over time, alongside detected faults marked by red vertical lines. The positions are derived from the VIO system. We start detecting the faults or drifts after the vehicle enters steady flight stage, which is after 50 seconds. Faults are detected when the residual between the predicted and observed states exceeds a certain threshold, which is indicated by the red lines in the plot. Between 70 and 80 seconds, several faults are detected, corresponding to significant deviations in position estimates. At this point, the

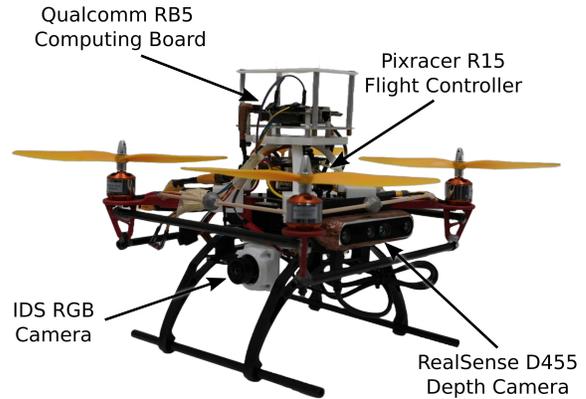


Fig. 3. The experimental UAV testbed includes a RealSense D455 depth camera, which provides both IMU data and stereo images for state estimation. An IDS camera is utilized to capture general RGB images. The proposed mechanism runs on a Qualcomm RB5 board, which communicates desired thrust and angular velocity commands to a Pixracer R15 flight controller.

VIO output is considered unreliable due to the detected drift. The plot (b) illustrates the evolution of the Kalman filter residuals over time, representing the discrepancy between predicted and observed measurements. The residuals in the plot are computed as the norm of the difference between the predicted state and the actual measurements. The figure shows that residuals fluctuate within acceptable limits, but sharp increases in the residual norm are seen around the fault detection times during the steady flight stage.

In (c), we visualize the fault detection statistic J . A horizontal line at J_{th} , the threshold value, indicates the fault detection boundary. The plot shows several peaks in J , and when these exceed the threshold, a fault is declared. We can notice that two of the spikes on the sides correspond to the takeoff and landing phases, while the central spike indicates an actual VIO drift event during the flight.

The final plot presents a histogram of fault detection events throughout the experiment. A similar pattern is observed, with three faults occurring during the steady flight stage between 60 and 80 seconds. This highlights a specific time window in which the system faced challenges in maintaining accurate state estimation.

V. CONCLUSION AND FUTURE WORK

In this work, we presented a framework for an autonomous safety mechanism designed to detect and mitigate VIO drift in UAVs. The approach leverages a Kalman filter-based fault detection system that monitors VIO state estimates and is envisioned to trigger a controlled landing procedure in the event of significant drift. The experimental results demonstrate the effectiveness of the drift detection component, which was evaluated using logged flight data from a laboratory environment with limited visual features. This framework aims to enhance the reliability of UAVs, particularly in GPS-denied environments or during critical missions requiring precise navigation. By providing a structured approach for drift detection, the proposed method contributes to reducing

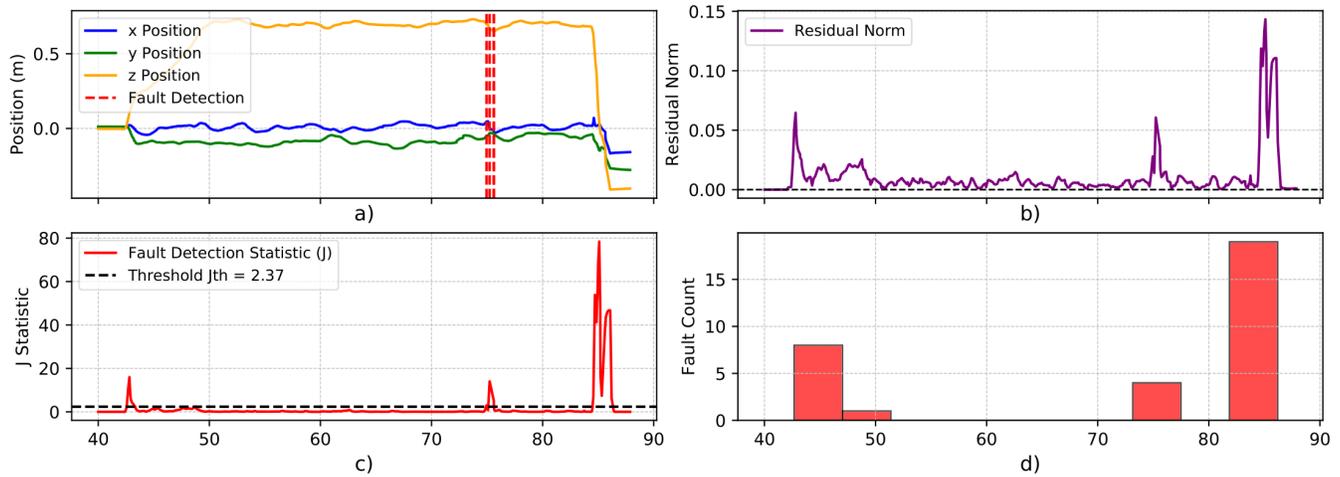


Fig. 4. Overview of drift (or fault) detection during the steady flight stage that is from $t \approx 52s$ to $t \approx 82s$: a) VIO position estimation, b) Kalman filter residual norm, c) fault detection statistic (J), and d) fault event histogram

the risk of crashes and ensuring UAVs operate safely when VIO state estimation is compromised.

For future work, several promising directions can be explored. One critical area is enhancing drift detection during agile UAV maneuvers, where rapid and complex movements can exacerbate VIO drift and challenge detection accuracy. Another valuable avenue is expanding the framework's applicability to a wider range of environmental conditions, such as dynamic lighting and windy weather. Additionally, integrating the drift detection mechanism with high-level mission planning could enable UAVs to autonomously adjust flight paths in response to persistent drift, improving operational reliability in real-world scenarios. A key next step is the real-time implementation of the full framework on UAV hardware, as shown in Figure 2, allowing for onboard drift detection and mitigation strategies to be tested under actual flight conditions.

ACKNOWLEDGEMENT

This work is supported by Caltrans: California Department of Transportation.

REFERENCES

- [1] H. Hildmann and E. Kovacs, "Using unmanned aerial vehicles (uavs) as mobile sensing platforms (msps) for disaster response, civil security and public safety," *Drones*, vol. 3, no. 3, p. 59, 2019.
- [2] S. Jung and H. Kim, "Analysis of amazon prime air uav delivery service," *Journal of Knowledge Information Technology and Systems*, vol. 12, no. 2, pp. 253–266, 2017.
- [3] J. Howard, V. Murashov, and C. M. Branche, "Unmanned aerial vehicles in construction and worker safety," *American journal of industrial medicine*, vol. 61, no. 1, pp. 3–10, 2018.
- [4] F. L. Chernousko, *State estimation for dynamic systems*. crc press, 1993.
- [5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.
- [6] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1885–1892.
- [7] J. Surber, L. Teixeira, and M. Chli, "Robust visual-inertial localization with weak gps priors for repetitive uav flights," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6300–6306.
- [8] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [9] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.
- [10] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [11] B. Zhang, X. Shao, Y. Wang, G. Sun, and W. Yao, "R-lvio: Resilient lidar-visual-inertial odometry for uavs in gnss-denied environment," *Drones*, vol. 8, no. 9, p. 487, 2024.
- [12] J. Z. Wells, R. Kumar, and M. Kumar, "Application of interacting multiple model for future state prediction of small unmanned aerial systems," in *2021 American control conference (ACC)*. IEEE, 2021, pp. 3755–3760.
- [13] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
- [14] W. Wan, H. Kim, N. Hovakimyan, L. Sha, and P. G. Voulgaris, "A safety constrained control framework for uavs in gps denied environment," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 214–219.
- [15] H.-J. Yoon, W. Wan, H. Kim, N. Hovakimyan, L. Sha, and P. G. Voulgaris, "Towards resilient uav: Escape time in gps denied environment with sensor drift," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 423–428, 2019.
- [16] W. Wan, H. Kim, Y. Cheng, N. Hovakimyan, P. Voulgaris, and L. Sha, "Safety constrained multi-uav time coordination: a bi-level control framework in gps denied environment," in *AIAA AVIATION 2021 FORUM*, 2021, p. 2463.
- [17] L. Ma and Y. Zhang, "Fault detection and diagnosis for gtm uav with dual unscented kalman filter," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7884.
- [18] S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*. Springer, 2021.
- [19] G. Welch, "An introduction to the kalman filter," 1995.
- [20] c. W. Mueller and c. D'Andrea, "Critical subsystem failure mitigation in an indoor uav testbed," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 780–785.
- [21] P. H. Zipfel, *Modeling and simulation of aerospace vehicle dynamics*. Aiaa, 2000.
- [22] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.