AER 1810 Project - Visual-Inertial Relative Pose Estimation for Quadrotor Landing

Matt Brymer

August 23, 2022

1 Introduction and Motivation

Autonomous UAVs, including quadrotors, are becoming increasingly common in applications such as aerial photography, outdoor surveillance of industrial assets and package delivery. A key component of these operations is executing a precise and safe landing at the end of the flight for delivering the payload or recharging the vehicle, for which some measurement of the position of the vehicle relative to the landing target is necessary. Inexpensive GPS units available for most low-cost quadrotors can typically only achieve horizontal position accuracies on the order of 2.5 m CEP [1], which is insufficient for a precise landing. Thus some other method of relative position measurement is necessary for widespread adoption of these systems.

One solution to this issue that has been successfully used in the literature is to use vision from an onboard camera to identify the target and provide a measurement of the relative pose[2][3]. This data can be fused with IMU or other onboard sensing data to produce a refined estimate of the relative pose of the vehicle for use in controlling the landing maneuver.

The end goal of the work surrounding this project overall is to develop the software necessary to allow a low cost quadrotor made from hobby grade components to autonomously take off, navigate to a given GPS location and execute a precision landing on a target. The first step in performing this task is to be able to accurately estimate the relative pose during the final descent phase. For these reasons, this project specifically investigates developing a visual-inertial relative pose estimation methodology for use on a low cost, hobby grade quadrotor platform.

2 Filter Architecture

The problem of visual-inertial pose estimation consists of fusing pose predictions based on integrated IMU data with pose measurements received from the vision pipeline. To simplify the vision problem, an AprilTag 3 fiducial marker is placed on the landing pad [4]. Given the marker dimensions and camera calibration parameters, the AprilTag detector allows the full 6 degree of freedom relative pose of the camera to be identified from a single image. This removes the need for a second camera and expensive stereo image processing to resolve depth, making it more tractable on small computationally constrained platforms.

For this work it is assumed that the landing pad remains fixed, and thus the quantities to be estimated are the position and attitude of the vehicle relative to the AprilTag. The coordinate frames involved include the tag or inertial frame, $\vec{\mathcal{F}}_t$, the vehicle frame, $\vec{\mathcal{F}}_v$, and the camera frame, $\vec{\mathcal{F}}_c$. These frames and the overall estimation problem are illustrated in Figure 1 below.

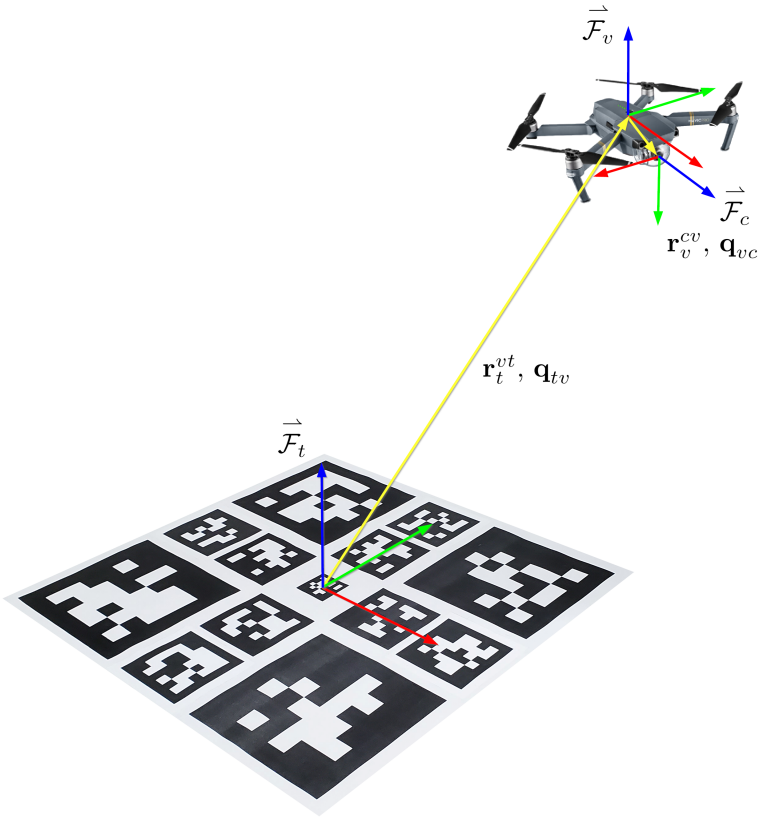

Figure 1: Coordinate frames and state definition for relative pose estimation

Where the pose of the camera frame relative to the vehicle is represented by the position $\mathbf{r}_v^{cv}$ and quaternion $\mathbf{q}_{vc}$, or its equivalent rotation matrix $\mathbf{C}_{vc}$.

Both data sources are fused using an Extended Kalman Filter with the IMU data used in the predictive model and the AprilTag measurements for the correction. The IMU is assumed to be coincident with the centre of gravity of the vehicle, or equivalently the estimate represents the pose of the IMU. Accelerometer and gyroscope biases are estimated in addition to the vehicle pose. A quaternion representation is used to parameterize the vehicle attitude in a Multiplicative Extended Kalman Filter framework where the attitude consists of a nominal attitude to which a small rotation vector perturbation is added[5].

The full state of the filter can thus be defined as:

$\mathbf{r}_t^{vt}$ - Position of vehicle wrt tag
$\mathbf{v}_t^{vt}$ - Velocity of vehicle wrt tag
$\mathbf{q}_{tv}$ - Quaternion attitude of vehicle wrt tag
$\mathbf{C}_{tv}$ - Rotation matrix of vehicle wrt tag
$\mathbf{a}_b$ - Accelerometer bias
$\boldsymbol{\omega}_b$ - Gyroscope bias

Where $\mathbf{C}_{tv}$ is the rotation matrix associated with $\mathbf{q}_{tv}$. The state can be broken up into nominal and perturbation components as follows:

$$\mathbf{r}_t^{vt} = \bar{\mathbf{r}}_t^{vt} + \delta\mathbf{r} \qquad \mathbf{v}_t^{vt} = \bar{\mathbf{v}}_t^{vt} + \delta\mathbf{v}$$
$$\mathbf{q}_{tv} = \bar{\mathbf{q}}_{tv} \otimes \delta\mathbf{q} \qquad \mathbf{C}_{tv} = \bar{\mathbf{C}}_{tv}\delta\mathbf{C}$$
$$\mathbf{a}_b = \bar{\mathbf{a}}_b + \delta\mathbf{a}_b \qquad \boldsymbol{\omega}_b = \bar{\boldsymbol{\omega}}_b + \delta\boldsymbol{\omega}_b$$
$$\delta\mathbf{q} = \exp\frac{\delta\boldsymbol{\theta}}{2} \qquad \delta\mathbf{C} = \exp\left[\delta\boldsymbol{\theta}\right]_\times$$

Where $\delta\mathbf{q}$ and $\delta\mathbf{C}$ are the resulting quaternion and rotation matrices after passing a small rotation vector $\delta\boldsymbol{\theta}$ through their respective exponential maps. This small rotation vector will encode the perturbation component of the rotational state. The Hamilton definition of the quaternion is assumed where the rotation represents a passive transformation from the local to global coordinate system, or equivalently an active transformation of the vehicle frame.

Thus the nominal and perturbation state vectors are defined as:

$$\mathbf{x} = \begin{bmatrix} \bar{\mathbf{r}}_t^{vt} \\ \bar{\mathbf{v}}_t^{vt} \\ \bar{\mathbf{q}}_{tv} \\ \bar{\mathbf{a}}_b \\ \bar{\boldsymbol{\omega}}_b \end{bmatrix} \qquad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{r} \\ \delta\mathbf{v} \\ \delta\boldsymbol{\theta} \\ \delta\mathbf{a}_b \\ \delta\boldsymbol{\omega}_b \end{bmatrix}$$

The measurements are defined as:

$\mathbf{a}_{obs}$ - Observed IMU acceleration, vehicle frame
$\boldsymbol{\omega}_{obs}$ - Observed IMU angular velocity, vehicle frame
$\mathbf{r}_{c,obs}^{tc}$ - Observed AprilTag position wrt camera
$\mathbf{q}_{ct,obs}$ - Observed AprilTag orientation wrt camera

Which are corrupted from their true values by sensor biases and noise:

$$\mathbf{a}_{obs} = \mathbf{a}_v + \mathbf{a}_b + \mathbf{a}_n - \mathbf{C}_{tv}^T\mathbf{g} \qquad \boldsymbol{\omega}_{obs} = \boldsymbol{\omega}_v + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n$$
$$\mathbf{r}_{c,obs}^{tc} = \mathbf{r}_c^{tc} + \mathbf{n}_r \qquad \mathbf{q}_{ct,obs} = \mathbf{q}_{ct} \otimes \exp\frac{\mathbf{n}_\theta}{2}$$

2.1 Motion Model

A kinematic motion model is used to integrate the IMU signals and the sensor biases are modeled with a random walk. This leads to a continuous time motion model for the full state of:

$$\dot{\mathbf{r}}_t^{vt} = \mathbf{v}_t^{vt} \qquad \dot{\mathbf{v}}_t^{vt} = \mathbf{C}_{tv}(\mathbf{a}_{obs} - \mathbf{a}_b - \mathbf{a}_n) + \mathbf{g}$$
$$\dot{\mathbf{q}}_{tv} = \frac{1}{2}\mathbf{q}_{tv} \otimes (\boldsymbol{\omega}_{obs} - \boldsymbol{\omega}_b - \boldsymbol{\omega}_n)$$
$$\dot{\mathbf{a}}_b = \mathbf{b}_a \qquad \dot{\boldsymbol{\omega}}_b = \mathbf{b}_\omega$$

Where $\mathbf{b}_a$ and $\mathbf{b}_\omega$ represent the random walk in the bias terms. With the definition of nominal and perturbation states as above the motion model can be split into models for both the nominal and perturbation states. Defining the motion model for the nominal kinematics as:

$$\dot{\bar{\mathbf{r}}}_t^{vt} = \bar{\mathbf{v}}_t^{vt} \qquad \dot{\bar{\mathbf{v}}}_t^{vt} = \bar{\mathbf{C}}_{tv}(\mathbf{a}_{obs} - \bar{\mathbf{a}}_b) + \mathbf{g}$$
$$\dot{\bar{\mathbf{q}}}_{tv} = \frac{1}{2}\bar{\mathbf{q}}_{tv} \otimes (\boldsymbol{\omega}_{obs} - \bar{\boldsymbol{\omega}}_b) \qquad \dot{\bar{\mathbf{a}}}_b = \dot{\bar{\boldsymbol{\omega}}}_b = 0$$

Then by substituting this expression into the full motion model and making the small rotation approximation $\delta\mathbf{C} \approx (\mathbf{1} + [\delta\boldsymbol{\theta}]_\times)$ the following kinematic model can be derived for the perturbation states:

$$\delta\dot{\mathbf{r}} = \delta\mathbf{v} \qquad \delta\dot{\mathbf{v}} = -\bar{\mathbf{C}}_{tv}[\mathbf{a}_{obs} - \bar{\mathbf{a}}_b]_\times\delta\boldsymbol{\theta} - \bar{\mathbf{C}}_{tv}\delta\mathbf{a}_b - \bar{\mathbf{C}}_{tv}\mathbf{a}_n$$
$$\delta\dot{\boldsymbol{\theta}} = -[\boldsymbol{\omega}_{obs} - \bar{\boldsymbol{\omega}}_b]_\times\delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n$$
$$\delta\dot{\mathbf{a}}_b = \mathbf{b}_a \qquad \delta\dot{\boldsymbol{\omega}}_b = \mathbf{b}_\omega$$

These equations can be expressed in discrete time by integrating them over a small interval Δt using a first order Euler approximation. This leads to the following difference equations for the motion model describing the nominal and perturbation kinematics:

Nominal:
$$\bar{\mathbf{r}}^{vt}_{t,k} = \bar{\mathbf{r}}^{vt}_{t,k-1} + \bar{\mathbf{v}}^{vt}_{t,k-1}\Delta t$$
$$\bar{\mathbf{v}}^{vt}_{t,k} = \bar{\mathbf{v}}^{vt}_{t,k-1} + (\bar{\mathbf{C}}_{tv,k-1}(\mathbf{a}_{obs,k} - \bar{\mathbf{a}}_{b,k-1}) + \mathbf{g})\Delta t$$
$$\bar{\mathbf{q}}_{tv,k} = \bar{\mathbf{q}}_{tv,k-1} \otimes \mathbf{q}\{(\boldsymbol{\omega}_{obs,k} - \bar{\boldsymbol{\omega}}_{b,k-1})\Delta t\}$$
$$\bar{\mathbf{a}}_{b,k} = \bar{\mathbf{a}}_{b,k-1}$$
$$\bar{\boldsymbol{\omega}}_{b,k} = \bar{\boldsymbol{\omega}}_{b,k-1}$$
Perturbation:
$$\delta\mathbf{r}_k = \delta\mathbf{r}_{k-1} + \delta\mathbf{v}_{k-1}\Delta t$$
$$\delta\mathbf{v}_k = \delta\mathbf{v}_{k-1} - \bar{\mathbf{C}}_{tv,k-1}([\mathbf{a}_{obs,k} - \bar{\mathbf{a}}_{b,k-1}]_\times\delta\boldsymbol{\theta}_{k-1} + \delta\mathbf{a}_{b,k-1})\Delta t - \bar{\mathbf{C}}_{tv,k-1}\mathbf{w}_{a,k}$$
$$\delta\boldsymbol{\theta}_k = (\exp[(\boldsymbol{\omega}_{obs,k} - \bar{\boldsymbol{\omega}}_{b,k-1})\Delta t]_\times)^T\delta\boldsymbol{\theta}_{k-1} - \Delta t\ \delta\boldsymbol{\omega}_{b,k-1} + \mathbf{w}_{\omega,k}$$
$$\delta\mathbf{a}_{b,k} = \delta\mathbf{a}_{b,k-1} + \mathbf{w}_{ab,k}$$
$$\delta\boldsymbol{\omega}_{b,k} = \delta\boldsymbol{\omega}_{b,k-1} + \mathbf{w}_{\omega b,k}$$

Where $\mathbf{w}_{a,k} \sim \mathcal{N}(0, \mathbf{Q}_a)$, $\boldsymbol{\omega}_{\omega,k} \sim \mathcal{N}(0, \mathbf{Q}_\omega)$, $\mathbf{w}_{ab,k} \sim \mathcal{N}(0, \mathbf{Q}_{ab})$, $\boldsymbol{\omega}_{\omega b,k} \sim \mathcal{N}(0, \mathbf{Q}_{\omega b})$ are the discretized versions of the IMU process noise and bias random walks integrated over the sampling interval. These can be stacked to give the full noise vector $\mathbf{w}_k$ as:

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_{a,k} \\ \mathbf{w}_{\omega,k} \\ \mathbf{w}_{ab,k} \\ \mathbf{w}_{\omega b,k} \end{bmatrix} \sim \mathcal{N}(0, diag(\mathbf{Q}_a, \mathbf{Q}_\omega, \mathbf{Q}_{ab}, \mathbf{Q}_{\omega b}))$$

The values before and after the correction step of the Kalman filter will be denoted with a check$(\check{\cdot})$ or hat$(\hat{\cdot})$ respectively. The filter update begins with an estimate for the nominal and perturbation states at the previous timestep.

$$\hat{\mathbf{x}}_{k-1} = \begin{bmatrix} \hat{\mathbf{r}}^{vt}_{t,k-1} \\ \hat{\mathbf{v}}^{vt}_{t,k-1} \\ \hat{\mathbf{q}}_{tv,k-1} \\ \hat{\mathbf{a}}_{b,k-1} \\ \hat{\boldsymbol{\omega}}_{b,k-1} \end{bmatrix} \qquad \delta\hat{\mathbf{x}}_{k-1} = \begin{bmatrix} \delta\hat{\mathbf{r}}_{k-1} \\ \delta\hat{\mathbf{v}}_{k-1} \\ \delta\hat{\boldsymbol{\theta}}_{k-1} \\ \delta\hat{\mathbf{a}}_{b,k-1} \\ \delta\hat{\boldsymbol{\omega}}_{b,k-1} \end{bmatrix} \sim \mathcal{N}(0, \hat{\mathbf{P}}_{k-1})$$

In the prediction step the nominal values can be propagated forward to the current timestep k using the difference equations above and the current IMU measurements. As the perturbation states start with a zero mean they will remain that way upon inspection but their covariance will increase. This leads to a prior for the state estimate at the current timestep k.

$$\check{\mathbf{x}}_k = \begin{bmatrix} \check{\mathbf{r}}^{vt}_{t,k} \\ \check{\mathbf{v}}^{vt}_{t,k} \\ \check{\mathbf{q}}_{tv,k} \\ \check{\mathbf{a}}_{b,k} \\ \check{\boldsymbol{\omega}}_{b,k} \end{bmatrix} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{a}_{obs,k}, \boldsymbol{\omega}_{obs,k}) \quad \delta\check{\mathbf{x}}_k = \begin{bmatrix} \delta\check{\mathbf{r}}_k \\ \delta\check{\mathbf{v}}_k \\ \delta\check{\boldsymbol{\theta}}_k \\ \delta\check{\mathbf{a}}_{b,k} \\ \delta\check{\boldsymbol{\omega}}_{b,k} \end{bmatrix} \sim \mathcal{N}(0, \check{\mathbf{P}}_k), \check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}^T_{k-1} + \mathbf{Q}_k$$

Where the Jacobians $\mathbf{F}_{k-1}$ and $\mathbf{Q}_k$ are derived from the perturbation kinematics model as:

$$\mathbf{F}_{k-1} = \frac{\partial \delta \mathbf{f}}{\partial \delta \mathbf{x}}\Big|_{\substack{\hat{\mathbf{x}}_{k-1} \\ \mathbf{a}_{obs,k}, \boldsymbol{\omega}_{obs,k}}} = \begin{bmatrix} \mathbf{1}_3 & \mathbf{1}_3 \Delta t & & & \\ & \mathbf{1}_3 & -\bar{\mathbf{C}}_{tv,k-1}[\mathbf{a}_{obs,k} - \bar{\mathbf{a}}_{b,k-1}]_\times \Delta t & -\bar{\mathbf{C}}_{tv,k-1}\Delta t & \\ & & (\exp[(\boldsymbol{\omega}_{obs,k} - \bar{\boldsymbol{\omega}}_{b,k-1})\Delta t]_\times)^T & & -\mathbf{1}_3 \Delta t \\ & & & \mathbf{1}_3 & \\ & & & & \mathbf{1}_3 \end{bmatrix}$$

$$\mathbf{Q}_k = \mathbf{W}_{k-1} E[\mathbf{w}_k \mathbf{w}_k^T] \mathbf{W}_{k-1}^T = \mathbf{W}_{k-1} diag(\mathbf{Q}_a, \mathbf{Q}_\omega, \mathbf{Q}_{ab}, \mathbf{Q}_{\omega b}) \mathbf{W}_{k-1}^T$$

$$\mathbf{W}_{k-1} = \frac{\partial \delta \mathbf{f}}{\partial \mathbf{w}_k}\Big|_{\substack{\hat{\mathbf{x}}_{k-1} \\ \mathbf{a}_{m,k}, \boldsymbol{\omega}_{m,k}}} = \begin{bmatrix} \mathbf{0} & & & \\ -\bar{\mathbf{C}}_{tv,k-1} & & & \\ & \mathbf{1}_3 & & \\ & & \mathbf{1}_3 & \\ & & & \mathbf{1}_3 \end{bmatrix}$$

2.2 Observation Model

For the correction step of the EKF the measurement model must be defined. The AprilTag detector returns a pose of the tag relative to the camera corrupted by some sensor noise as:

$$\mathbf{r}_{c,obs}^{tc} = \mathbf{r}_c^{tc} + \mathbf{n}_r \qquad\qquad \mathbf{q}_{ct,obs} = \mathbf{q}_{ct} \otimes \exp \frac{\mathbf{n}_\theta}{2}$$

With $\mathbf{n}_r \sim \mathcal{N}(0, \mathbf{R}_r)$, $\mathbf{n}_\theta \sim \mathcal{N}(0, \mathbf{R}_\theta)$. These outputs can be equated to observed values of the state variables $\mathbf{r}_t^{vt}$ and $\mathbf{q}_{tv}$ through the camera to vehicle coordinate transformation as:

$$\mathbf{r}_{t,obs}^{vt} = -\mathbf{C}_{tv,obs}(\mathbf{C}_{vc}\mathbf{r}_{c,obs}^{tc} + \mathbf{r}_v^{cv}) \qquad\qquad \mathbf{q}_{tv,obs} = (\mathbf{q}_{vc} \otimes \mathbf{q}_{ct,obs})^*$$

The expression for the position measurement can be rearranged to be solely in terms of the vehicle state and noise variables as follows:

$$\mathbf{r}_{t,obs}^{vt} = \exp\left(\frac{\mathbf{n}_\theta}{2}\right)^T (\mathbf{r}_t^{vt} - \mathbf{C}_{tv}\mathbf{C}_{vc}\mathbf{n}_r)$$

The state can again be split into nominal and perturbation components with the small rotation approximation. This allows the observed position measurement to be written in terms of a nominal and perturbation component.

$$\mathbf{r}_{t,obs}^{vt} = \bar{\mathbf{r}}_{t,obs}^{vt} + \delta \mathbf{r}_{obs} = \exp\left(\frac{\mathbf{n}_\theta}{2}\right)^T (\bar{\mathbf{r}}_t^{vt} + \delta \mathbf{r} + \bar{\mathbf{C}}_{tv}[\mathbf{C}_{vc}\mathbf{n}_r]_\times \delta\boldsymbol{\theta} - \bar{\mathbf{C}}_{tv}\mathbf{C}_{vc}\mathbf{n}_r)$$

If this nominal-perturbation split is performed about $\check{\mathbf{x}}_k$ then because the perturbation components have a zero mean, $E[\mathbf{r}_{t,obs}^{vt}] = \check{\mathbf{r}}_t^{vt}$. Thus $\check{\mathbf{r}}_{t,k}^{vt}$ can be subtracted off to derive an error measurement model that relates to the perturbation states and can be used to derive Jacobians for the filter correction.

$$\delta \mathbf{r}_{obs} = \mathbf{r}_{t,obs}^{vt} - \check{\mathbf{r}}_{t,k}^{vt} = (\exp\left(\frac{\mathbf{n}_\theta}{2}\right)^T - \mathbf{1})\check{\mathbf{r}}_{t,k}^{vt} + \exp\left(\frac{\mathbf{n}_\theta}{2}\right)^T (\delta \mathbf{r}_k - \check{\mathbf{C}}_{tv,k}\mathbf{C}_{vc}\mathbf{n}_r)$$

Where the $\delta\boldsymbol{\theta}$ term has been dropped as it becomes zero when linearizing about $\check{\mathbf{x}}_k$ and $\mathbf{n}_k = \mathbf{0}$.

The observed orientation $\mathbf{q}_{tv,obs}$ must also be related to the true orientation plus some measurement noise. Linearizing about $\check{\mathbf{x}}_k$ allows deriving a model for the error between the observed and predicted orientation in terms of the perturbation state.

$$\begin{aligned} &\mathbf{q}_{tv,obs} = \mathbf{q}_{tv} \otimes \exp\left(\frac{\mathbf{n}'_\theta}{2}\right) = \check{\mathbf{q}}_{tv,k} \otimes \delta\mathbf{q} \otimes \exp\left(\frac{\mathbf{n}'_\theta}{2}\right) \\ &\check{\mathbf{q}}_{tv,k}^* \otimes \mathbf{q}_{tv,obs} = \delta\mathbf{q}_{obs} = \exp\left(\frac{\delta\boldsymbol{\theta}}{2}\right) \otimes \exp\left(\frac{\mathbf{n}'_\theta}{2}\right) \\ &\delta\boldsymbol{\theta}_{obs} = 2\log(\delta\mathbf{q}_{obs}) \end{aligned}$$

Where $\mathbf{n}'_\theta = \mathbf{C}_{vc}\mathbf{n}_\theta$ and $\log()$ refers to the quaternion logarithm or inverse exponential map. Assuming that both $\delta\boldsymbol{\theta}$ and $\mathbf{n}_\theta$ are small, $\delta\boldsymbol{\theta}_{obs}$ will be equal to their sum so their Jacobians will be the identity matrix and $\mathbf{C}_{vc}$ respectively.

Therefore, the full measurement model linearized about $\check{\mathbf{x}}_k$ can be summarized as:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{r}^{vt}_{t,obs,k} \\ \mathbf{q}_{tv,obs,k} \end{bmatrix} \qquad \delta\mathbf{y}_k = \begin{bmatrix} \delta\mathbf{r}_{obs,k} \\ \delta\boldsymbol{\theta}_{obs,k} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{vt}_{t,obs,k} - \check{\mathbf{r}}^{vt}_{t,k} \\ 2\log(\check{\mathbf{q}}^{*}_{tv,k} \otimes \mathbf{q}_{tv,obs}) \end{bmatrix} \qquad \mathbf{n}_k = \begin{bmatrix} \mathbf{n}_{r,k} \\ \mathbf{n}_{\theta,k} \end{bmatrix}$$

$$\mathbf{G}_k = \frac{\partial \delta\mathbf{g}}{\partial \delta\mathbf{x}}\Big|_{\substack{\check{\mathbf{x}}_k \\ \mathbf{n}_k=0}} = \begin{bmatrix} \mathbf{1}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$\mathbf{R}_k = \mathbf{N}_k E[\mathbf{n}_k \mathbf{n}_k^T] \mathbf{N}_k^T = \mathbf{N}_k diag(\mathbf{R}_r, \mathbf{R}_\theta) \mathbf{N}_k^T$$
$$\mathbf{N}_k = \frac{\partial \delta\mathbf{g}}{\partial \mathbf{n}}\Big|_{\check{\mathbf{x}}_k} = \begin{bmatrix} -\check{\mathbf{C}}_{tv,k}\mathbf{C}_{vc} & [\check{\mathbf{r}}^{vt}_{t,k}]_\times \\ \mathbf{0} & \mathbf{C}_{vc} \end{bmatrix}$$

2.3 Multiplicative Extended Kalman Filter and Time Delays

The correction step of the EKF can then be performed by calculating the Kalman gain matrix $\mathbf{K}_k$ and the posterior estimate of the perturbation state based on the observed $\delta\mathbf{y}_k$

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{G}_k^T (\mathbf{G}_k \check{\mathbf{P}}_k \mathbf{G}_k^T + \mathbf{R}_k)^{-1}$$
$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k \mathbf{G}_k) \check{\mathbf{P}}_k$$
$$\delta\hat{\mathbf{x}}_k = \mathbf{K}_k \delta\mathbf{y}_k$$
$$\delta\mathbf{x}_k | \delta\mathbf{y}_k \sim \mathcal{N}(\delta\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k)$$

The estimated perturbation state is then added to the nominal state and its mean is reset to zero to complete the filter update.

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{r}}^{vt}_{t,k} \\ \hat{\mathbf{v}}^{vt}_{t,k} \\ \hat{\mathbf{q}}_{tv,k} \\ \hat{\mathbf{a}}_{b,k} \\ \hat{\boldsymbol{\omega}}_{b,k} \end{bmatrix} = \begin{bmatrix} \check{\mathbf{r}}^{vt}_{t,k} + \delta\hat{\mathbf{r}}_k \\ \check{\mathbf{v}}^{vt}_{t,k} + \delta\hat{\mathbf{v}}_k \\ \check{\mathbf{q}}_{tv,k} \otimes \delta\hat{\mathbf{q}}_k \\ \check{\mathbf{a}}_{b,k} + \delta\hat{\mathbf{a}}_{b,k} \\ \check{\boldsymbol{\omega}}_{b,k} + \delta\hat{\boldsymbol{\omega}}_{b,k} \end{bmatrix} \qquad \delta\hat{\mathbf{x}}_k = \begin{bmatrix} \delta\hat{\mathbf{r}}_k \\ \delta\hat{\mathbf{v}}_k \\ \delta\hat{\boldsymbol{\theta}}_k \\ \delta\hat{\mathbf{a}}_{b,k} \\ \delta\hat{\boldsymbol{\omega}}_{b,k} \end{bmatrix} \longleftarrow \mathbf{0}$$

Where $\delta\hat{\mathbf{q}}_k$ is the quaternion corresponding to the posterior rotational perturbation estimate $\delta\hat{\mathbf{q}}_k = \exp(\frac{\delta\hat{\boldsymbol{\theta}}_k}{2})$.

Due the all the steps involved in the image processing pipeline including debayering, rectification and solving the AprilTag pose problem, there may be a significant delay between when the pose is observed by the camera and the tag detection reaches the filter. This creates an issue as it will mean that after fusion the IMU motion model will propagate from a past rather than current state. This could lead to inaccuracies or faster drift, particularly in the case of angular errors which cause gravity to produce a constant acceleration bias. This problem is more generally known as the Out-of-Sequence-Measurement problem [6].

Many methods in the literature exist for handling time delayed measurements, including extrapolating the delayed measurement forward to the current time[7] and maintaining cross-covariance matrices across time[8] among others. For this filter the most straightforward approach was used, which is to store the state and input history until the delayed measurement is received. The measurement is fused with the prediction at the time it occurred and then the prediction steps of the filter are rerun to the current time using the stored inputs. The history before the measurement is then discarded. This process is shown graphically in Figure 2 below.

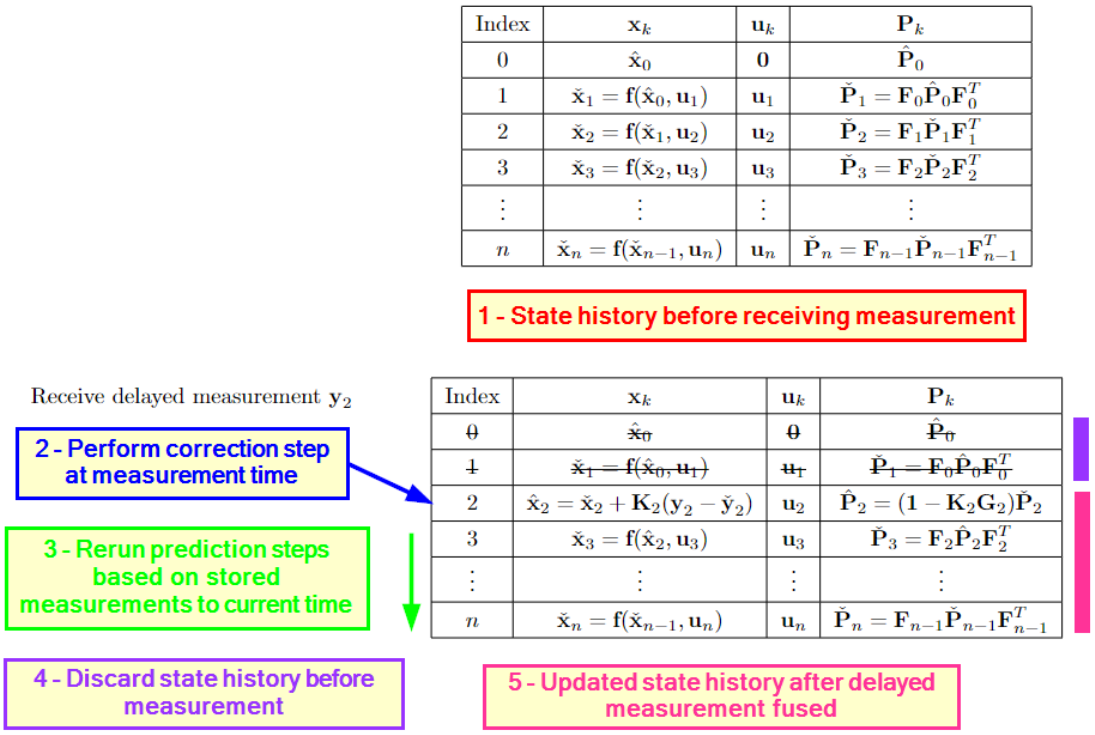

Figure 2: Illustration of the filter recalculation method employed for handling time delayed measurements

This method requires maintaining a history of the state estimate at the end of each filter step($\hat{\mathbf{x}}_k$ for correction step, $\check{\mathbf{x}}_k$ otherwise), the covariance matrix and the inputs since the time of the last received measurement. While not efficient, it is still theoretically consistent and for this application the state dimension is low so the extra computational overhead is not an issue.

3 Filter Implementation and Hardware Platform

3.1 Software Implementation

The filter described in the previous section was implemented in a ROS node written in C++ for ROS Melodic. It takes as inputs IMU data from either simulation or the hardware platform as well as the pose of the detected tag bundle from the AprilTag ROS node[9] and performs the filtering steps to output the relative pose estimate and covariance.

The filter is updated at 100 Hz in both simulation and hardware and fuses tag detections whenever they are received. The delay to apply to the tag detection is calculated dynamically by taking the difference between the current time and the image timestamp and adding an additional static delay. As well, through development it was observed that the AprilTag detector can produce spurious tag detections when the corners are close to the edge of the image. To improve robustness, the corners of the tag are projected into the image plane based on the reported pose and the detection is only accepted if the projected corners are sufficiently far away from the edges of the image. This issue is described in more detail in Sections 5 and 7.

The filter was validated first in simulation using the TRAILab fork of the RotorS simulator[10] before testing in hardware.

3.2 Hardware Platform

The filter was deployed on a custom quadrotor platform developed from hobby grade parts. The quadrotor measures approximately 200 x 148 mm between the motors and has 5.1" diameter propellors powered by a 4S LiPo battery. The low level flight control is performed by a Holybro Kakute F7 flight controller running ArduPilot 4.2.2 on an STM32F7 microcontroller. The filter itself runs on a NVIDIA Jetson Nano 4GB using images obtained from a Basler daA2500-60mci downward facing camera. Serial communication between the Jetson and the flight controller occurs over UART via the MAVLink protocol using the mavros node[11]. The vehicle features a forward facing camera, but this is an analog camera for first person manual flight only. It has a total mass of 1.115 kg and estimated maximum thrust of 36.9 N based on data from the motor manufacturer for this propellor type[12], leading to a thrust to weight ratio of approximately 3.4. The vehicle is shown in Figure 3 below

The downward facing camera is capable of producing images at a resolution of 2592x1944 and frame rate of 60 fps, although the Jetson can only receive them at 30 fps due to bus limitations. The Evetar M118B0418 lens has a focal length of 4.0 mm and aperture of $f/1.8$, producing a field of view of 101 x 76 degrees. To make the image processing time tractable, the image is downsampled by 4x to 640x480. This allows the AprilTag node to produce detections at approximately 15 Hz when run on the CPU.

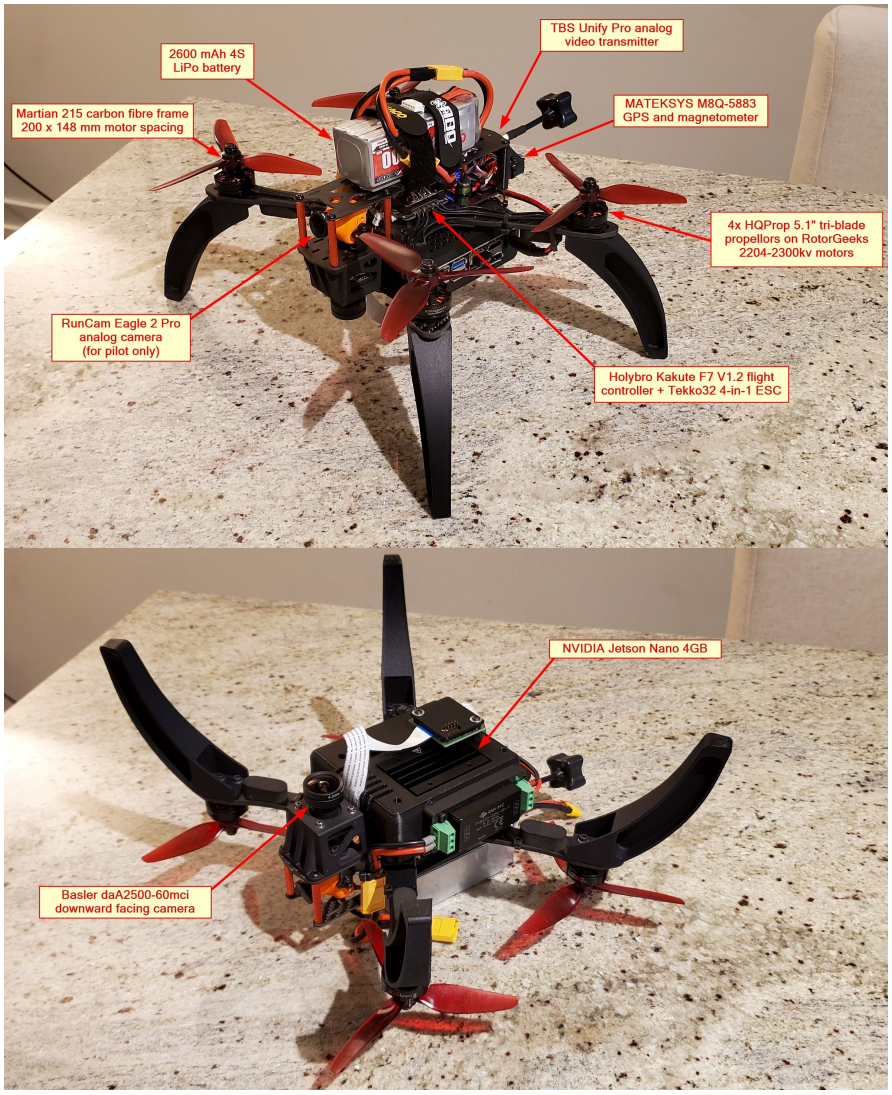

Figure 3: Custom quadrotor platform used for testing the relative pose filter

The camera intrinsic parameters, distortion coefficients as well as camera to IMU transform were derived using the Kalibr toolbox for camera intrinsic and extrinsic calibration[13]. The resulting values are summarized in Tables 1 and 2 below. Figure 4 below shows sample images from the intrinsic and extrinsic calibration tests.

For the hardware tests, a custom target was designed consisting of a bundle of 13 AprilTags of different sizes as shown in Figure 5 below. The bundle has tags of size 75 mm, 152 mm and 305 mm nested within an overall 0.84 m wide square. This pattern allows for some of the tags to still remain visible when the vehicle is offset laterally and has to pitch the camera away from the tag centre to return to the middle. It has a small tag at the centre to maintain detections when close to the ground while still retaining a large total area over which tags can be detected for stable detections at longer ranges.

Camera Intrinsics			
f_x	437.341	κ_1	-0.2765
f_y	438.087	κ_2	0.07382
c_x	328.54	τ_1	0.000256
c_y	239.25	τ_2	0.00126

Table 1: Camera intrinsic calibration results

Camera Extrinsics				
Parameter	x	y	z	w
$\mathbf{r}_v^{cv}$ [mm]	60.4	-1.5	-44.4	
$\mathbf{q}_{vc}$	-0.7035	0.7107	0.0014	-0.0017

Table 2: Camera extrinsic calibration results

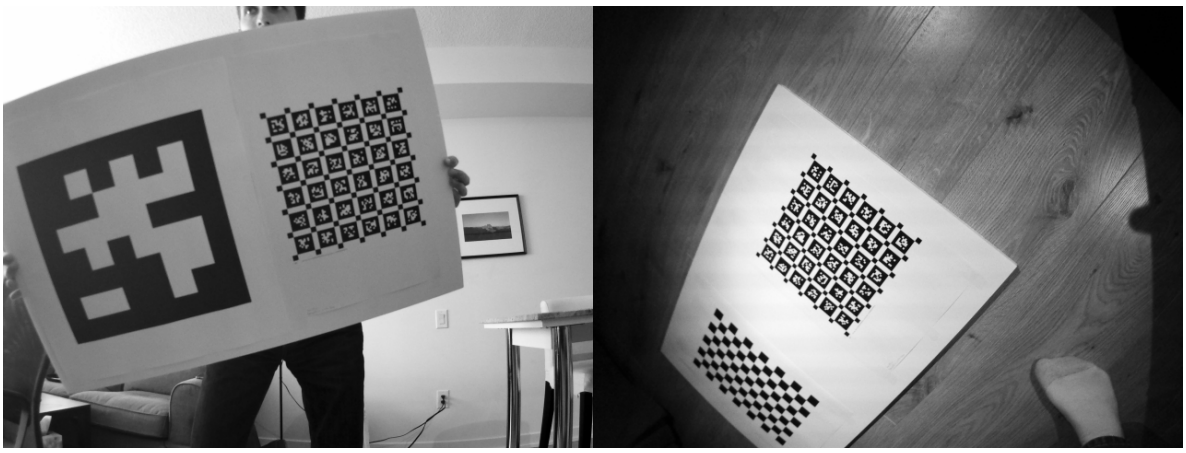

Figure 4: Images from intrinsic and extrinsic calibration of downward facing camera

Figure 5: Custom landing target made from nested AprilTags for hardware experiments

3.3 Filter Parameters

Separate process and measurement noise covariance terms were developed for the simulation and hardware implementations of the filter. In both cases diagonal covariances were assumed for simplicity.

In simulation the IMU process noise terms were guided by multiplying the observed variation in the IMU error by the integration time interval. The AprilTag measurement noise terms were guided by calculating the difference between the reported and ground truth tag pose as shown in Figure 6 below and assuming the observed variation represented an approximate 2σ bound. The IMU bias covariances were selected to be an order of magnitude lower than their corresponding error covariances.

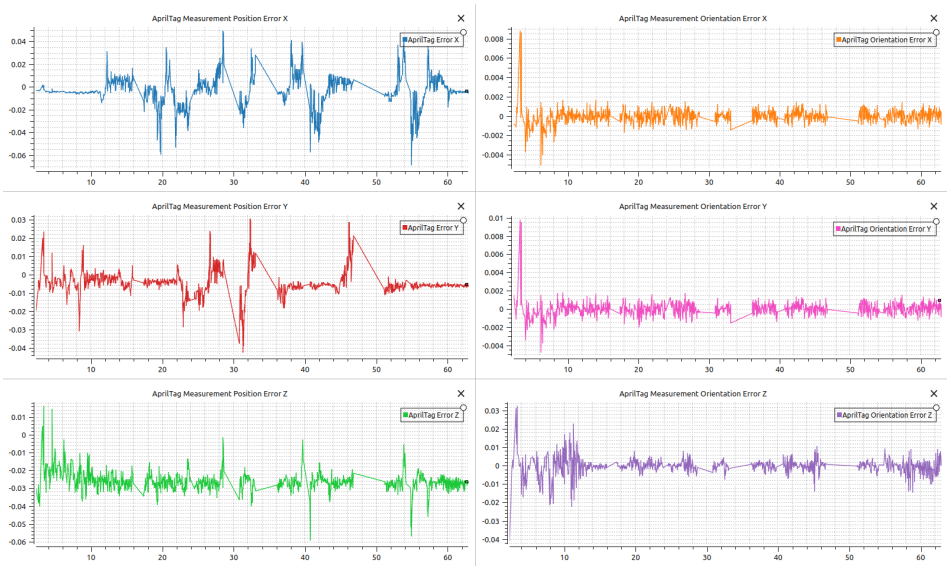

Figure 6: Error between reported and ground truth AprilTag pose in simulation

In hardware, the IMU process noises were guided based on an Allan deviation plot derived from static IMU data using the existing Allan Variance ROS package [14]. The resulting line slopes representing the velocity and acceleration random walks were integrated over the sampling interval to produce the covariance terms. For reference the resulting Allan deviation plot for the accelerometer data is shown in Figure 7 below.

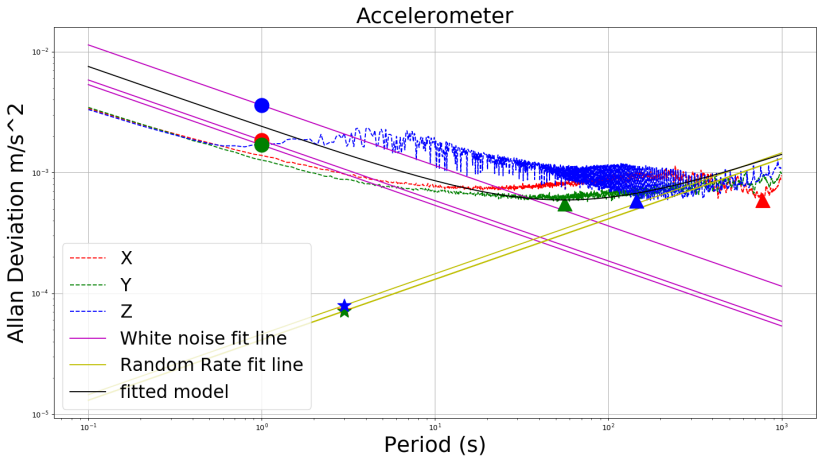

Figure 7: Allan deviation measured from static IMU data on hardware platform

The hardware AprilTag measurement noises were derived by placing a small AprilTag in view of the camera and recording the variation in reported pose over approximately 10 seconds. This test was done at distances of 0.5, 1.5 and 2.5 m and angles of $\pm$ 30 degrees. The maximum variation seen over

the sample windows was taken as an approximate 2σ bound for calculating the covariances. A sample plot of the tag detection image and pose estimate trace from this test is shown in Figure 8 below.

Figure 8: Sample image and pose detection results for hardware AprilTag measurement noise test

The process and measurement noise covariance terms assumed in the filter for both simulation and hardware are summarized in Table 3 below:

Parameter	Simulation			Hardware		
	x	y	z	x	y	z
$\mathbf{Q}_a$	5.0E-4	5.0E-4	5.0E-4	2.5E-4	2.5E-4	2.5E-4
$\mathbf{Q}_\omega$	5.0E-5	5.0E-5	5.0E-5	4.5E-4	4.5E-4	4.5E-4
$\mathbf{Q}_{ab}$	5.0E-5	5.0E-5	5.0E-5	7.0E-6	7.0E-6	7.0E-6
$\mathbf{Q}_{\omega b}$	5.0E-6	5.0E-6	5.0E-6	4.4E-5	4.4E-5	4.4E-5
$\mathbf{R}_r$	1.5E-2	1.5E-2	2.0E-2	1.5E-3	1.5E-3	6.0E-3
$\mathbf{R}_\theta$	1.5E-3	1.5E-3	4.0E-2	1.5E-3	1.5E-3	4.0E-2

Table 3: Process and measurement noise covariances assumed for simulation and hardware implementations of the relative pose filter

Lastly, the static time delays when fusing the AprilTag measurements in addition to the dynamic delay were assumed to be 5 ms in simulation and 85 ms in hardware based on experimentation.

4 Filter Validation: Simulation

As mentioned earlier, the filter was validated first in simulation using the TRAILab fork of the RotorS simulator[10] before testing in hardware. The simulator provides both the dynamics of the vehicle as well as implementing a low-level rate and attitude controller.

To produce more consistent responses between simulation runs, two flight cases were created by recording stick inputs during manual sweeping flights around the AprilTag target at heights of 2 m and 4 m that excited both the pitch and roll axes. These were then replayed as open loop inputs to test the filter response. It should be noted that due to noise added in the simulation there still is some small variation between runs, but overall they are fairly consistent. A sample image of the simulated camera view along with the estimated and ground truth pose visualized in RViz is shown in Figure 9 below.

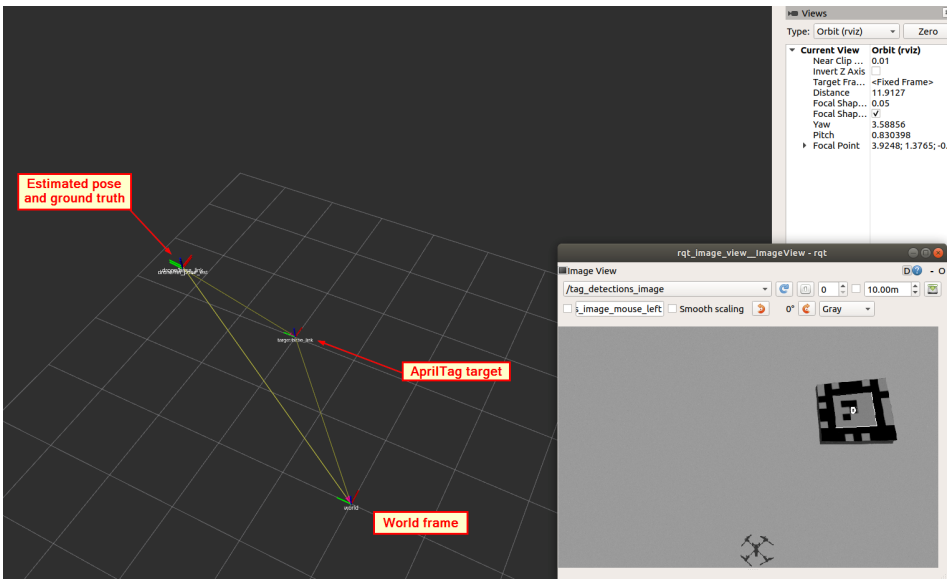

Figure 9: Example simulation results showing the simulated camera view and pose results visualized in RViz for the 2 m height case

The estimated position and orientation of the vehicle for the 2 m flight case along with the position estimation error and 3σ uncertainty bound is shown in Figures 10-12 below.

It can be seen that for the 2 m case the filter generally tracks both the ground truth position and orientation well, with estimation errors below approximately 0.050 m as long as detection is maintained. The camera loses sight of the tag multiple times during the run at the edges of its sweeping trajectory, leading to drift in the state estimate. We observe a drift of 0.61 m in the Y position estimate after 2.5 s of lost detection around 32 s into the event and -0.34 m in Z after 3.8 s of lost detection at 20 s. While not ideal, these are relatively long intervals to go without correction from measurements and thus this performance seems reasonable. It can also be seen that in all cases the estimation error lies within the 3σ uncertainty bounds.

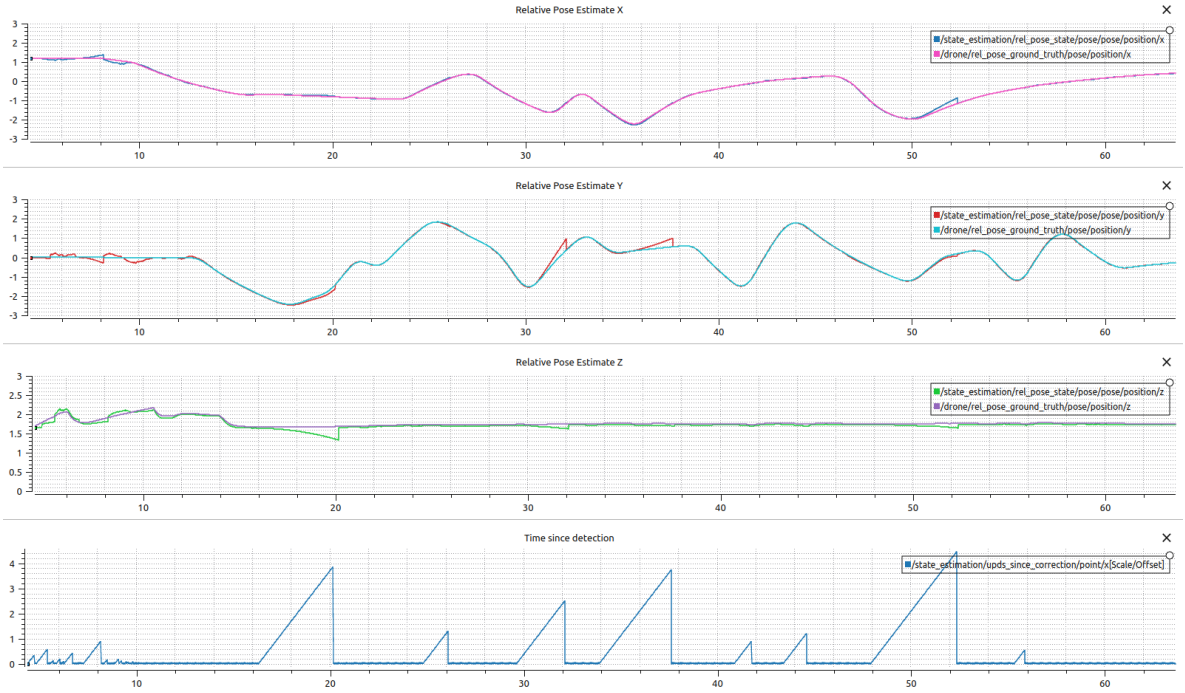

Figure 10: Position estimate vs ground truth for simulated flight at 2 m

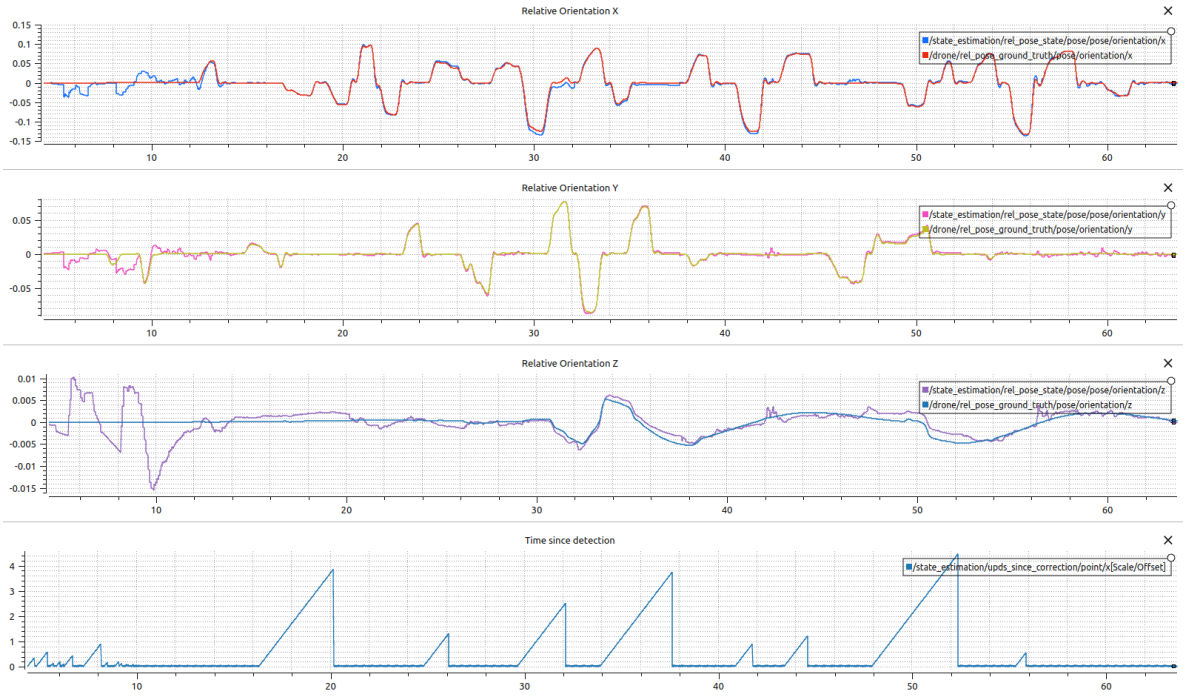

Figure 11: Orientation estimate vs ground truth for simulated flight at 2 m

The same results for the 4 m flight case are shown in Figures 13-15 below.

It can be seen that again for the 4 m case the filter overall tracks the ground truth position and orientation well with estimation errors generally below 0.25 m. We do however begin to see some areas with some localized variability in the pose estimate leading to the error growing beyond the 3σ bounds, particularly the regions around 18, 85 and 100 s into the event.

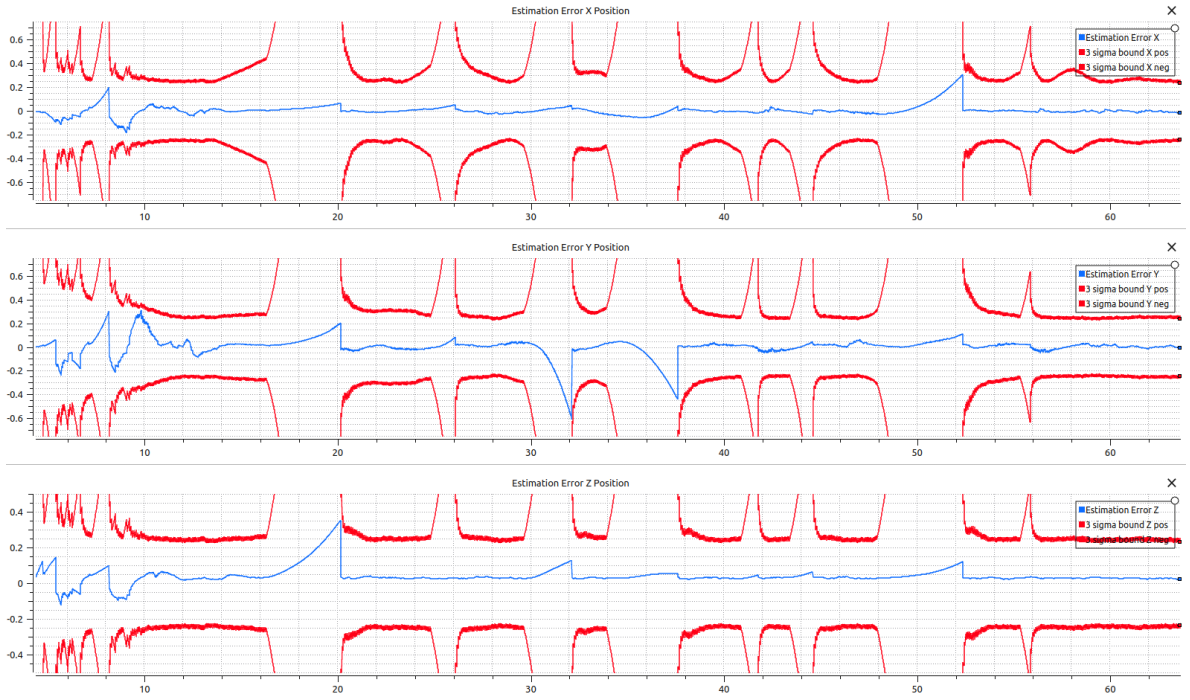

Figure 12: Position estimation error for simulated flight at 2 m

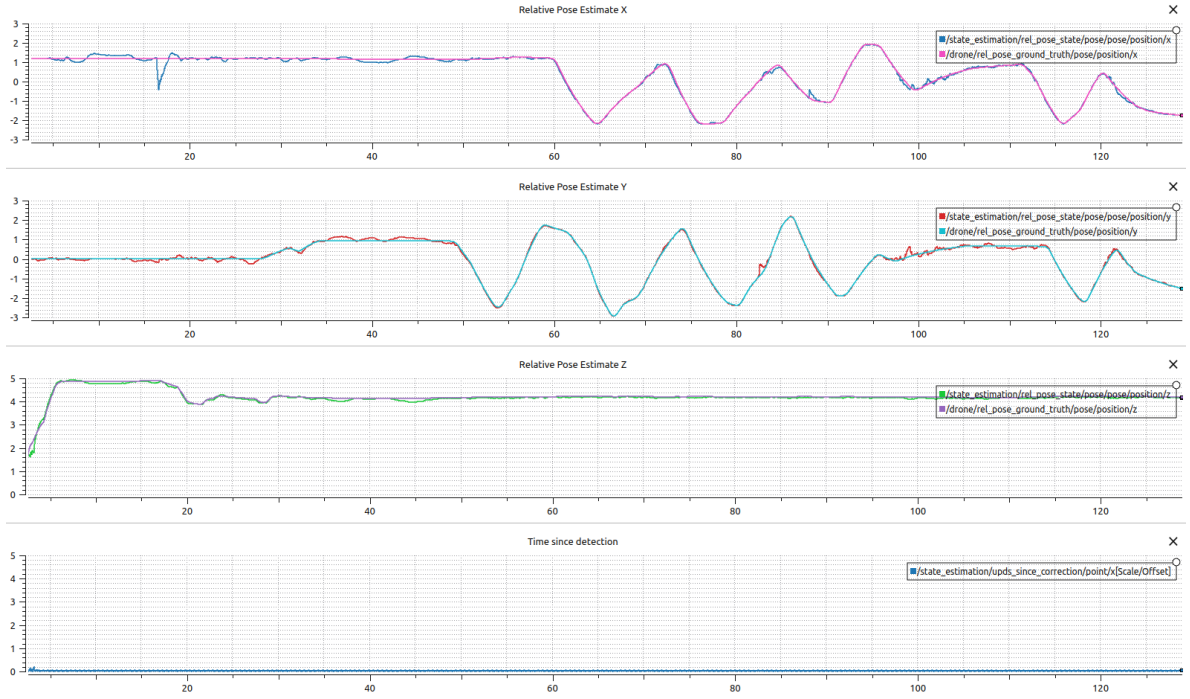

Figure 13: Position estimate vs ground truth for simulated flight at 4 m

Figure 14: Orientation estimate vs ground truth for simulated flight at 4 m

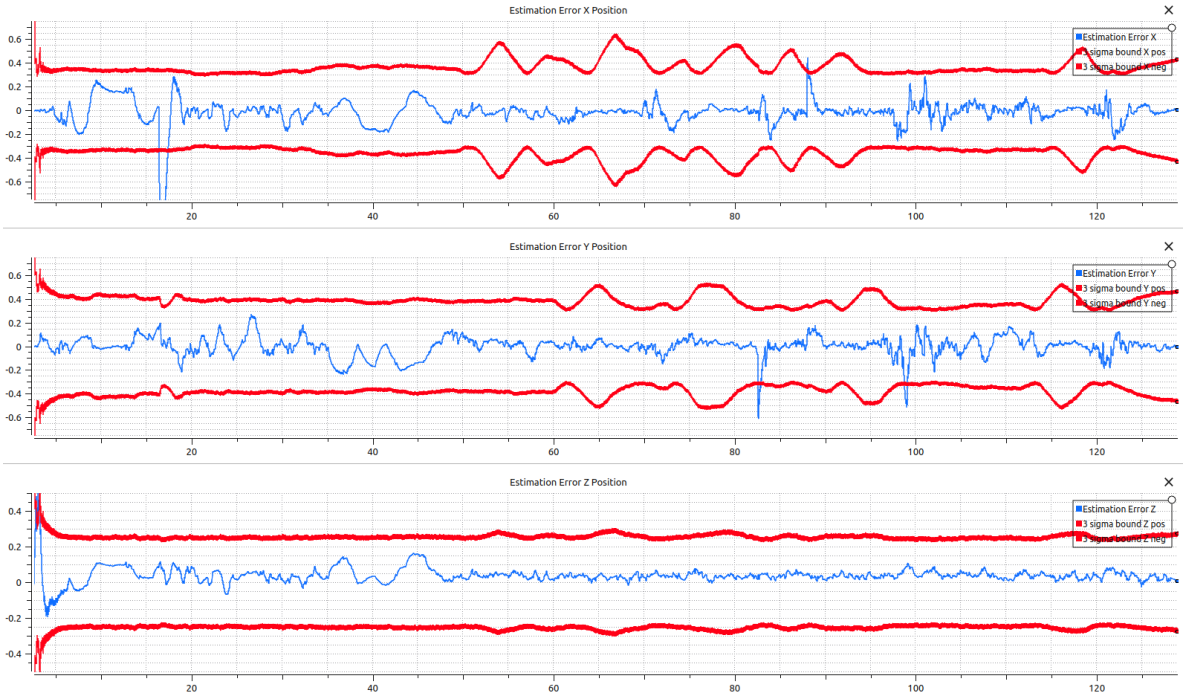

Figure 15: Position estimation error for simulated flight at 4 m

This behaviour is due to the tag being fairly small in the image as shown in Figure 16 below, meaning that variations in which pixel is identified as the corner of the tag have a proportionally larger effect on the orientation estimate. As the AprilTag estimate is the only absolute reference in the filter it has relatively high authority, allowing these variations to make their way into the final estimate.

In this case the simulated tag size is relatively small compared to the height expected in the hardware experiments so this was deemed to not be an issue. Despite the local variability, the performance is overall reasonable and deemed sufficient for relative position control.

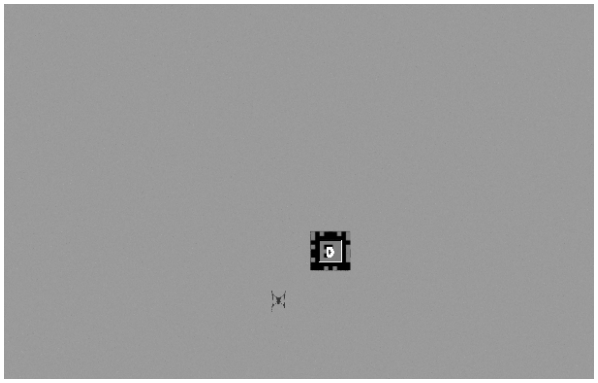

Figure 16: Reference image for the simulated AprilTag as seen in flight at a height of 4 m

5 Filter Validation: Indoors

Before moving to flight experiments, the filter was first tested indoors when moving the vehicle around by hand to validate its performance under controlled conditions. The filter was tested in three cases including:

- Translation of approximately ±0.5 m just past the edges of the tag in the X and Y directions as well as a combination of the two
- Rotations of approximately ±30 degrees at the centre and edges of the tag in both pitch and roll as well as yaw at the centre of the tag
- A combined translation and rotation motion representing flight back and forth across the tag in both directions as well as a combined yaw and translation motion around the centre of the tag

The tests were conducted at heights of approximately 0.4 m, 1 m and 1.8 m. The results were overall similar between heights and so only the results at 1 m are presented here for brevity.

Figures 17 to 19 below show the estimated position and orientation in the translation case along with sample tag detection images to illustrate the motion. As no ground truth data is available, the estimates are instead compared against the relative pose reported by the AprilTag detector.

It can be seen that detection of the tag is maintained throughout the event and that the estimated X and Y positions vary between approximately ±0.5 m, which is in line with the expected results. The orientation estimate also remains relatively constant as expected, minus some variation in the Z component of the quaternion due to an inadvertently applied yaw motion. The estimates closely follow the reported pose from the AprilTag detector with a phase lead, which is due to the filter accounting for the delay in the pose measurements.

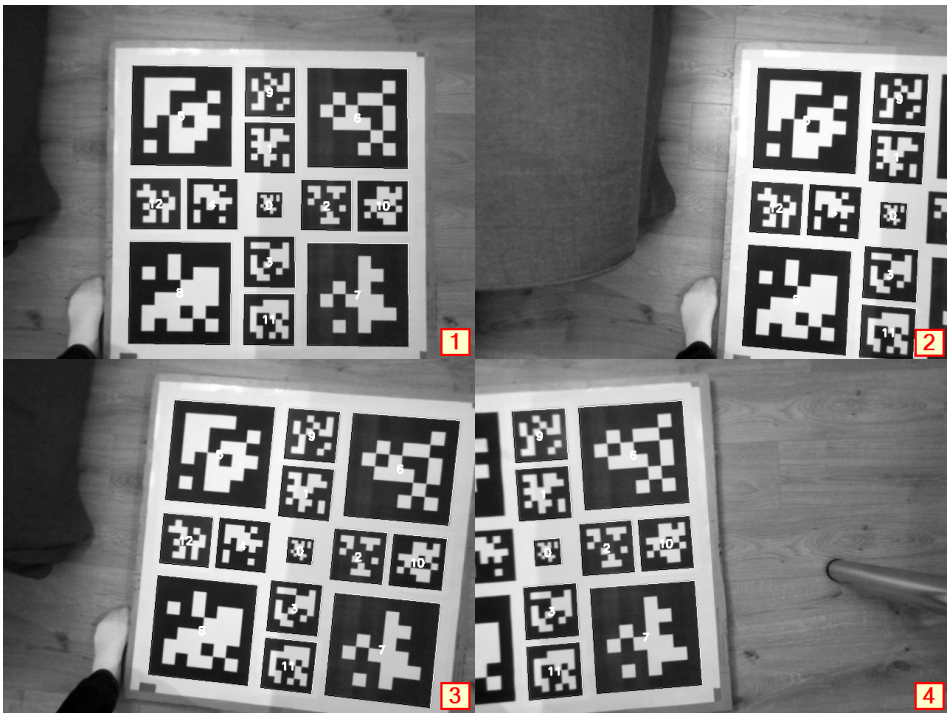

Figure 17: Sample tag detection images illustrating the translation case at a height of 1 m

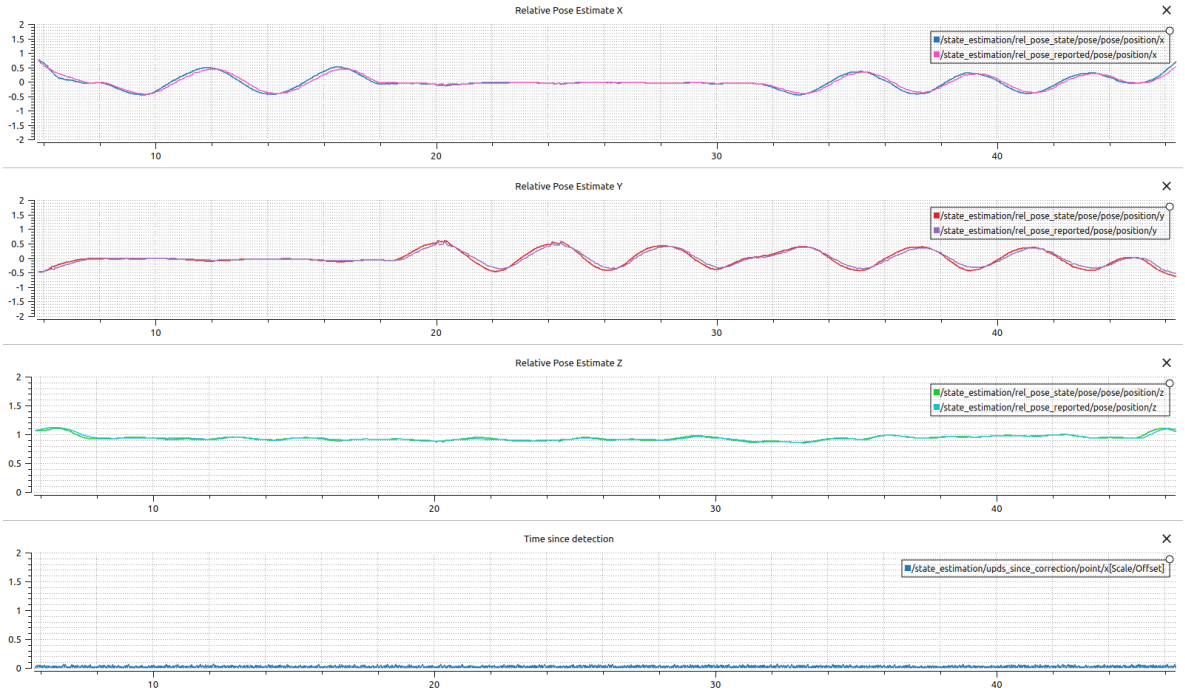

Figure 18: Position estimate results for the translation case at a height of 1 m

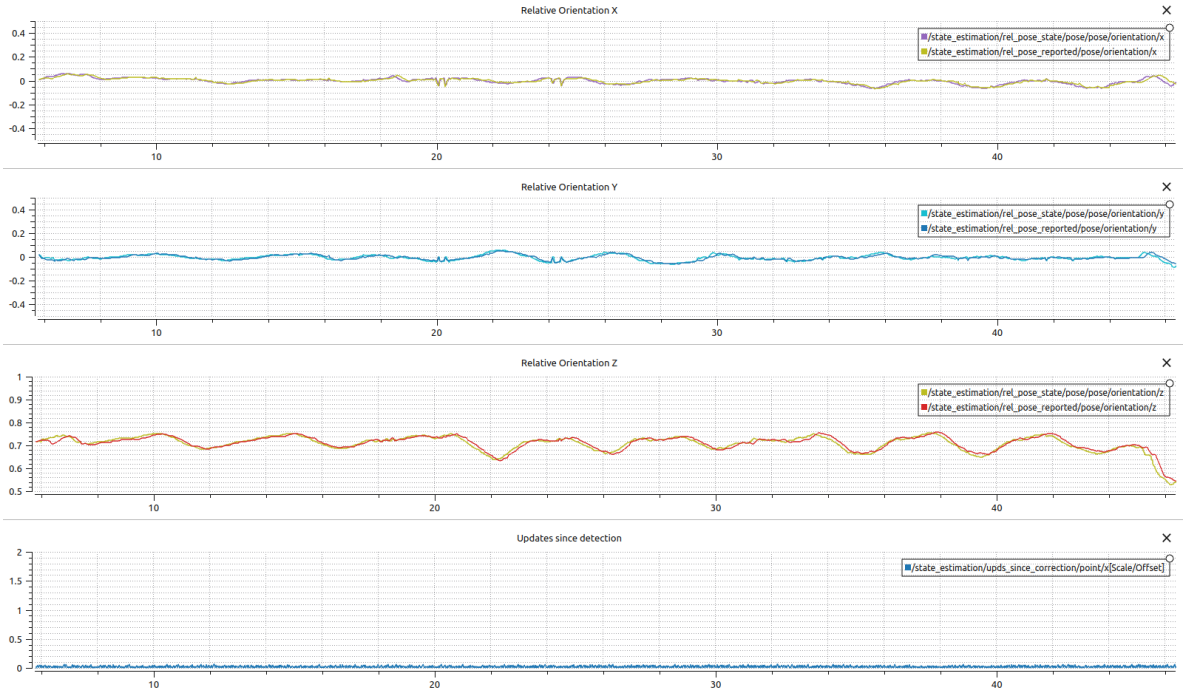

Figure 19: Orientation estimate results for the translation case at a height of 1 m

Figures 20 to 22 below show the same results for the rotation case. For the first 38 seconds of the event the vehicle is centred on the tag during the rotations and detection of the tag is mostly maintained. The result is that the estimated X and Y position remains approximately zero as expected and the X and Y components of the attitude quaternion vary ± 0.2, in line with the expected rotation of approximately ± 30 degrees.

Two anomalies do however occur in this portion of the event. At about 22 seconds detection of the tag is lost for approximately 0.5 s, leading to the position estimate drifting by about 0.33 m in Y and 0.14 m in X before returning to 0 when detection is regained. This drift is due to the fact that errors in the angle estimate lead to a constant acceleration induced by the projected component of the gravitational acceleration, causing a bias in the X or Y direction. This leads to a quadratic growth in estimation error in the absence of tag detections.

This is made worse by the fact that the filter also received a spurious tag detection just as the tag went out of view that was not removed by the tag corners in view criterion as evidenced by the jump in reported orientation at this point in time, further increasing the angular error. The jump in estimated position at about 18 seconds despite maintaining detection can be similarly seen to be due to a spurious reported orientation. This issue is explained in more detail in Section 7. Despite these two anomalies the filter still does a relatively good job of tracking the vehicle pose.

For the latter half of the event the rotations occur when the vehicle is at the edges of the tag, leading to the tag detection being lost for approximately 1 s on each rotation when the camera is pointed away from the tag. This leads to more significant drift in the position estimate of 0.5 - 1 m in most cases or more than 2 m in the worst case. Inspecting the orientation estimates we can again see that there are significant changes in the reported orientation right as the tag goes out of view. While not ideal behaviour, it can be seen that the drift is always away from the centre of the tag which will lead the position controller to fly the vehicle back to where detection can be regained. This case also represents a worst case, as under flight the rotation would be accompanied by a lateral acceleration that will help counteract the drift when detection is lost.

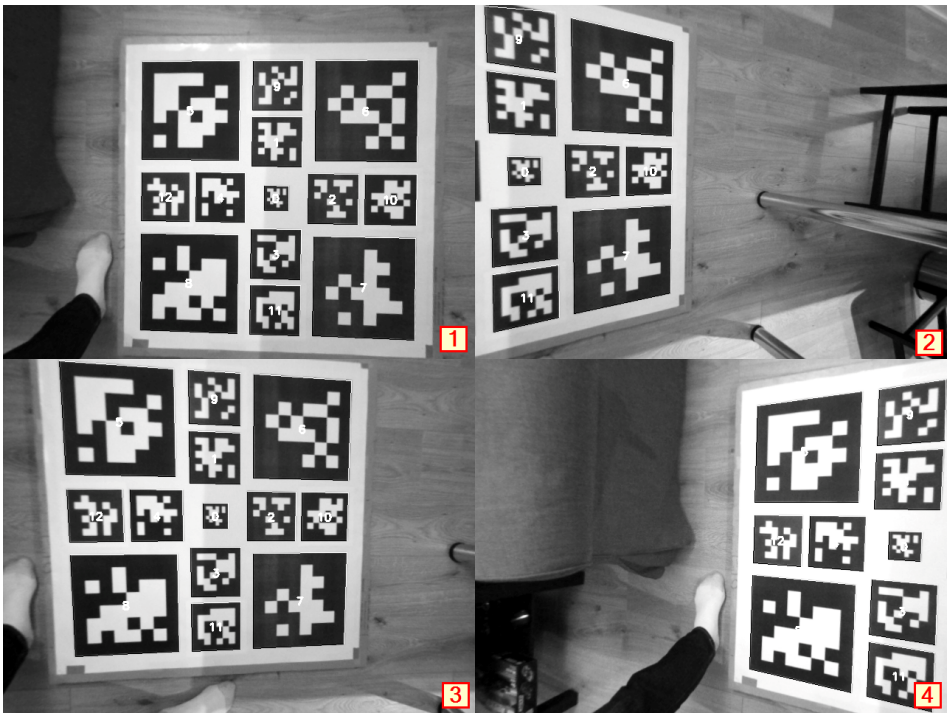

Figure 20: Sample tag detection images illustrating the rotation case at a height of 1 m

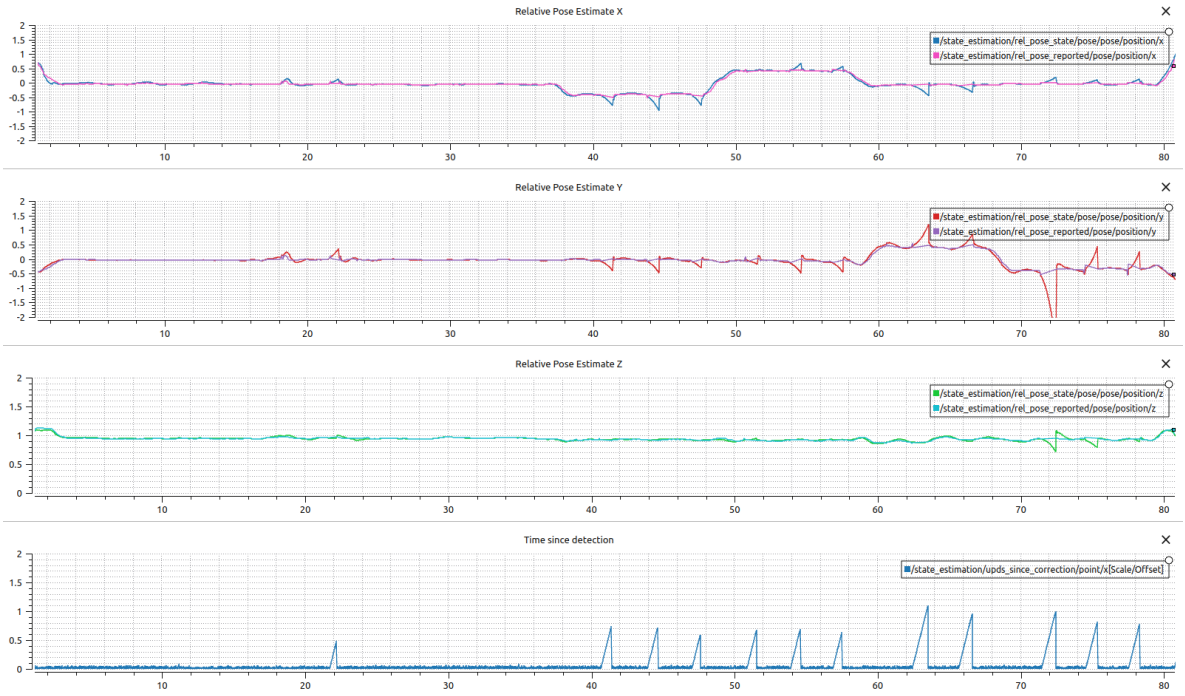

Figure 21: Position estimate results for the rotation case at a height of 1 m

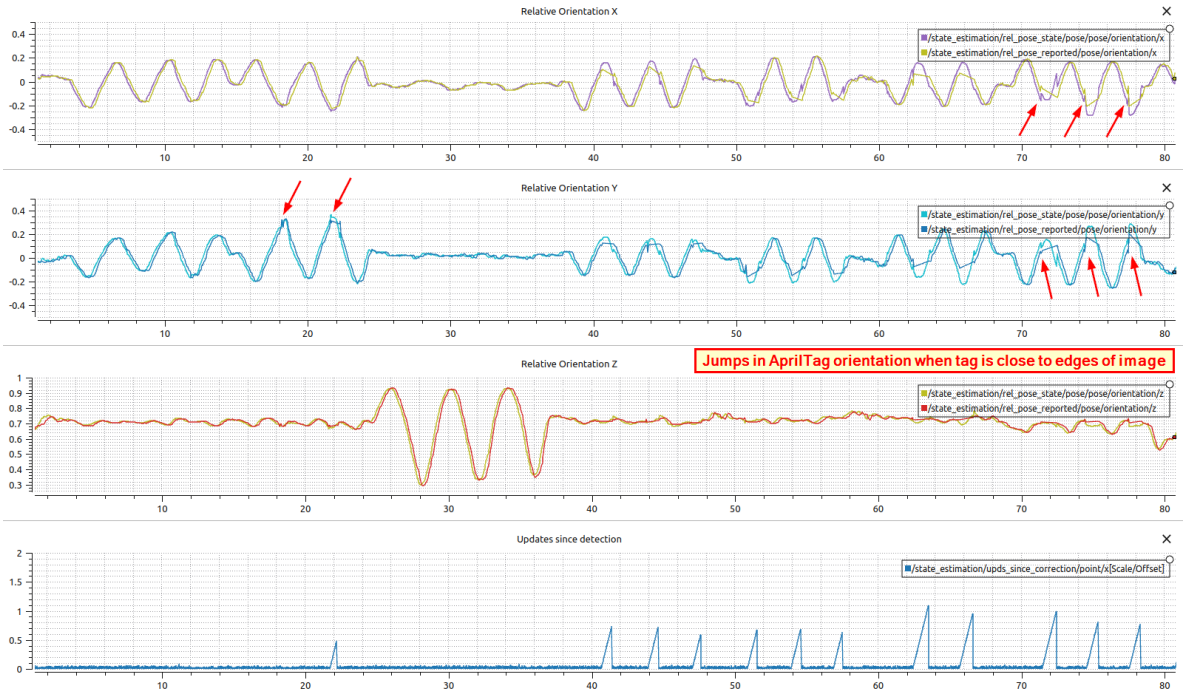

Figure 22: Orientation estimate results for the rotation case at a height of 1 m

Finally, Figures 23 to 25 below show the same results for the combined translation and rotation case. It can be seen that again detection is lost at the extremities of the motion for approximately 0.5 s. The shorter lost detection time and stabilizing effect of the additional translation acceleration leads to less drift with a maximum of approximately 0.97 m.

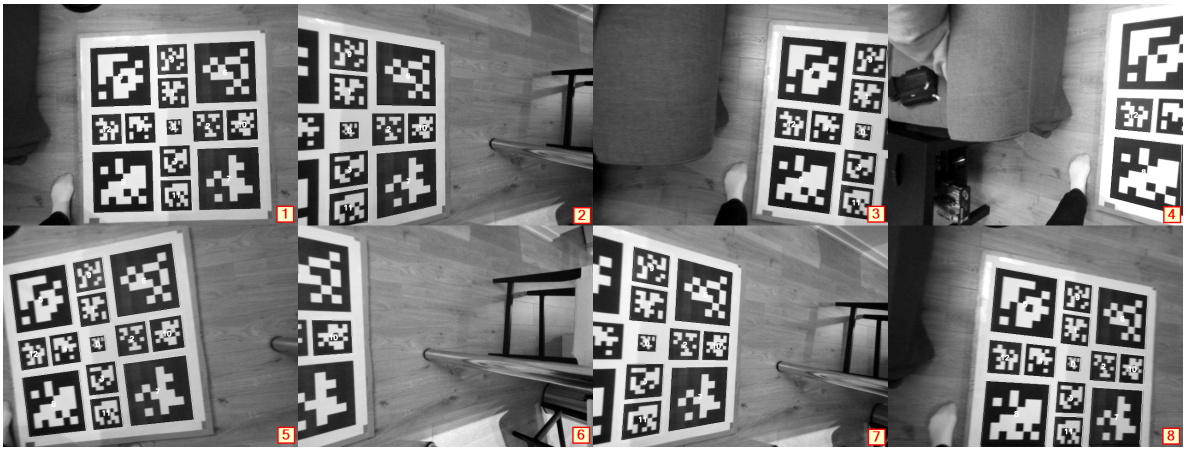

Figure 23: Sample tag detection images illustrating the combined motion case at a height of 1 m

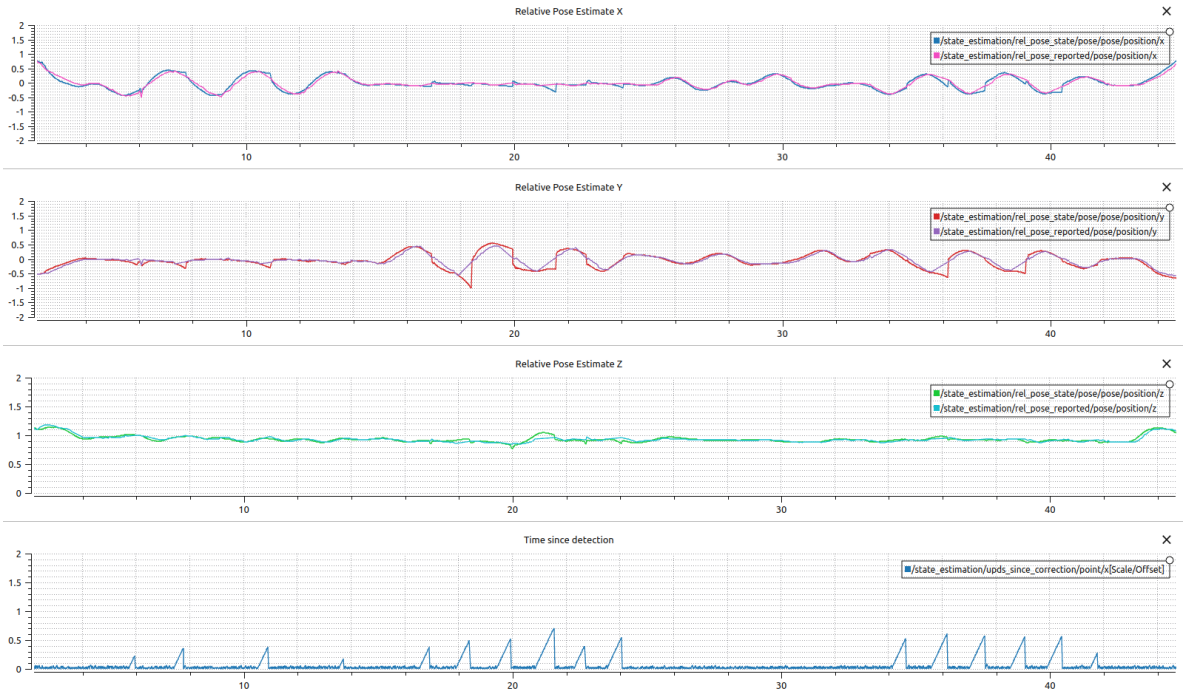

Figure 24: Position estimate results for the combined motion case at a height of 1 m

While the magnitude of the drift on lost detections is not ideal, the overall performance when detection is maintained is reasonable. Given that the filter will be used in combination with a relative position controller to keep the tag in view, the performance was deemed sufficient to proceed to hardware testing. Improving robustness to lost detections through better filtering of the AprilTag orientations or adding an additional attitude reference to address this challenge is an area for future work.

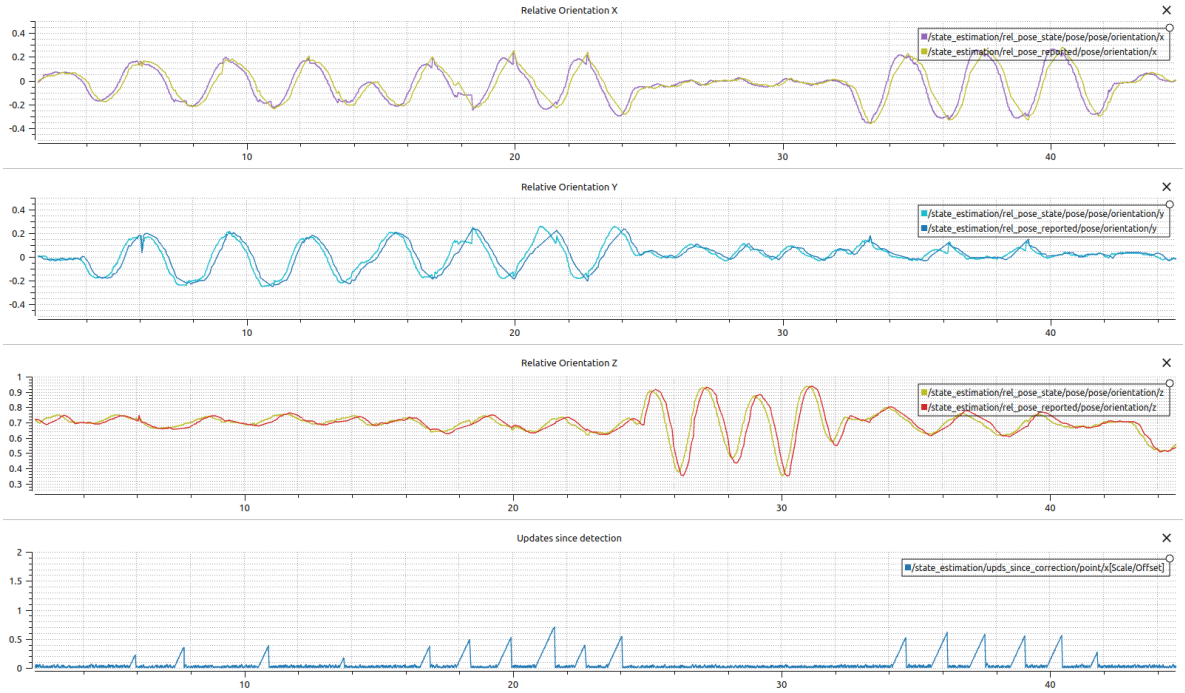

Figure 25: Orientation estimate results for the combined motion case at a height of 1 m

The computed time delay for fusing the AprilTag pose detections was recorded for reference and is shown in Figure 26 below. The delay varies between approximately 0.14 and 0.22 s, illustrating the importance of both including the delay as well as calculating it dynamically.

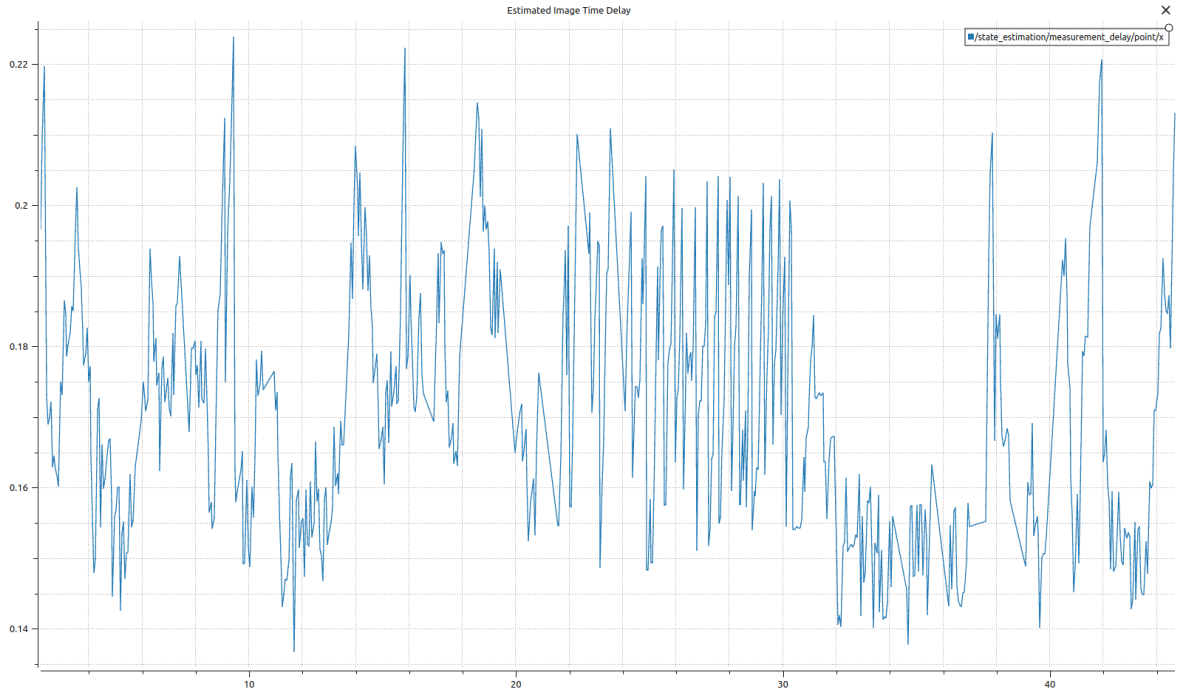

Figure 26: Computed time delay when fusing AprilTag measurements for the combined motion validation case at a height of 1 m

6 Filter Validation: Flight Testing

The filter was finally validated via outdoor flight tests on the quadrotor platform under manual flight. Two cases were evaluated, including a case of sweeping flight back and forth over the target as well as a full landing. Due to variations caused by manual flight, the height in the sweeping flight case varied between 1 - 2.5 m while the landing case began from a height of approximately 4 m.

Figures 27 to 29 below show the position and orientation estimation results for the sweeping flight case along with sample tag detection images. Again, since no ground truth data is available the estimates are compared against the AprilTag reported poses. The filter predicts position estimates between -0.5 and 1.5 m in X and ±1 m in Y that are overall of the correct magnitude compared to what was observed visually and in the tag detection images in relation to the tag size. The bias towards positive X position was likely due to the flight being performed by line-of-sight.

Losses of detection lasting 0.75 - 1.5 s occur at the edges of the trajectory leading to drift in the position estimate of 0.5 - 1.5 m with the severity of the drift depending on both the time for which detections have been lost and the last orientation estimate. Again while not ideal, the performance of the filter is deemed to be overall reasonable when considering that it would be used in conjunction with a relative position controller to keep the tag in view.

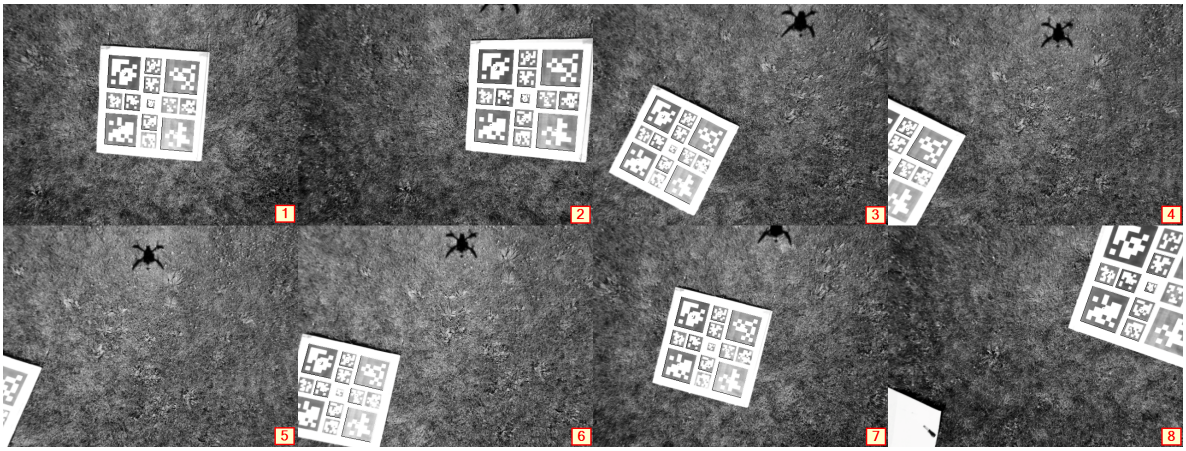

Figure 27: Sample tag detection images illustrating the sweeping flight test case

Figures 30 to 32 below show the same results for the landing case. The estimated and reported poses show good agreement across the event. After maneuvering into a roughly centred position over the target, the vehicle descends starting from 4 m at 9 s into the event before touching down at approximately 20 s. Some variation in both the position and yaw occurs due to external disturbances. The vehicle touches down slightly off centre of the tag due to the flight being performed manually.

Detection of the tag is lost at approximately 19 s when the vehicle is 0.5 m off the ground due to the vehicle being off centre, leading to drift in the position estimate. This loss of detection is likely to occur in most landings, so when deployed in combination with a relative position controller it may become necessary to transition to open-loop control for the final descent phase.

The computed time delay for fusing the AprilTag pose detections was again recorded for reference and is shown in Figure 33 below for the sweeping flight case. The delay is comparable to what was observed indoors and varies between approximately 0.14 and 0.25 s.

Overall, based on these results the estimation performance of the filter is deemed reasonable and sufficient for deployment with a relative position controller for automatically landing the vehicle.

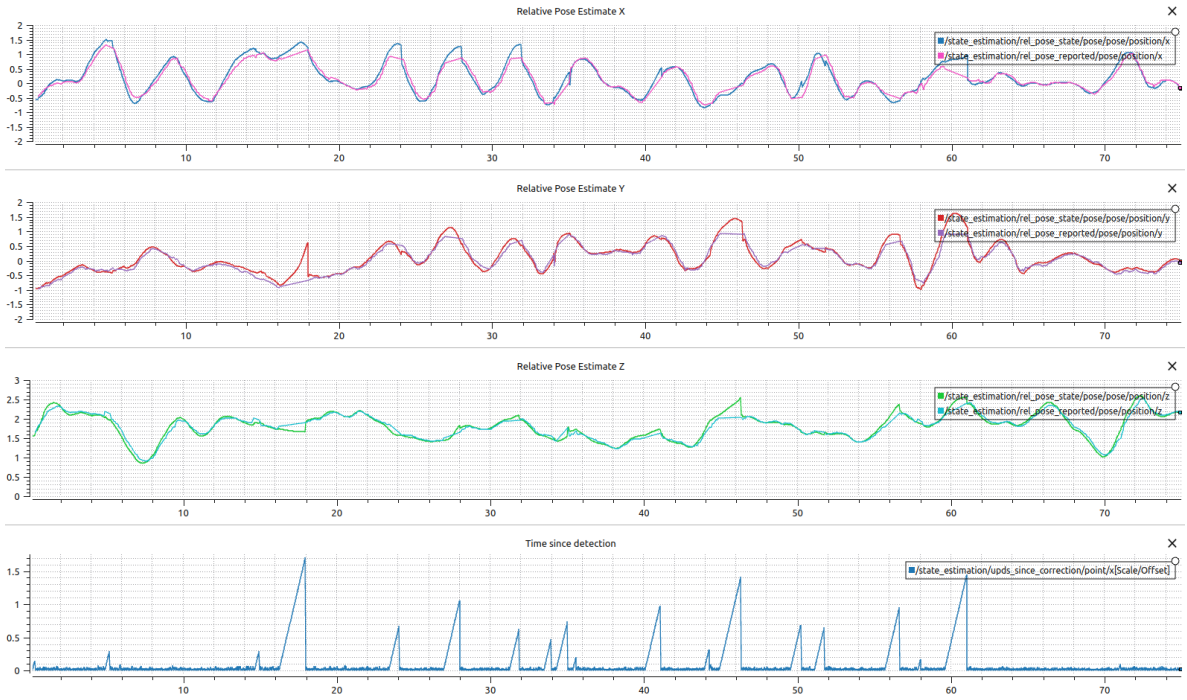

Figure 28: Position estimate results for the sweeping flight test case

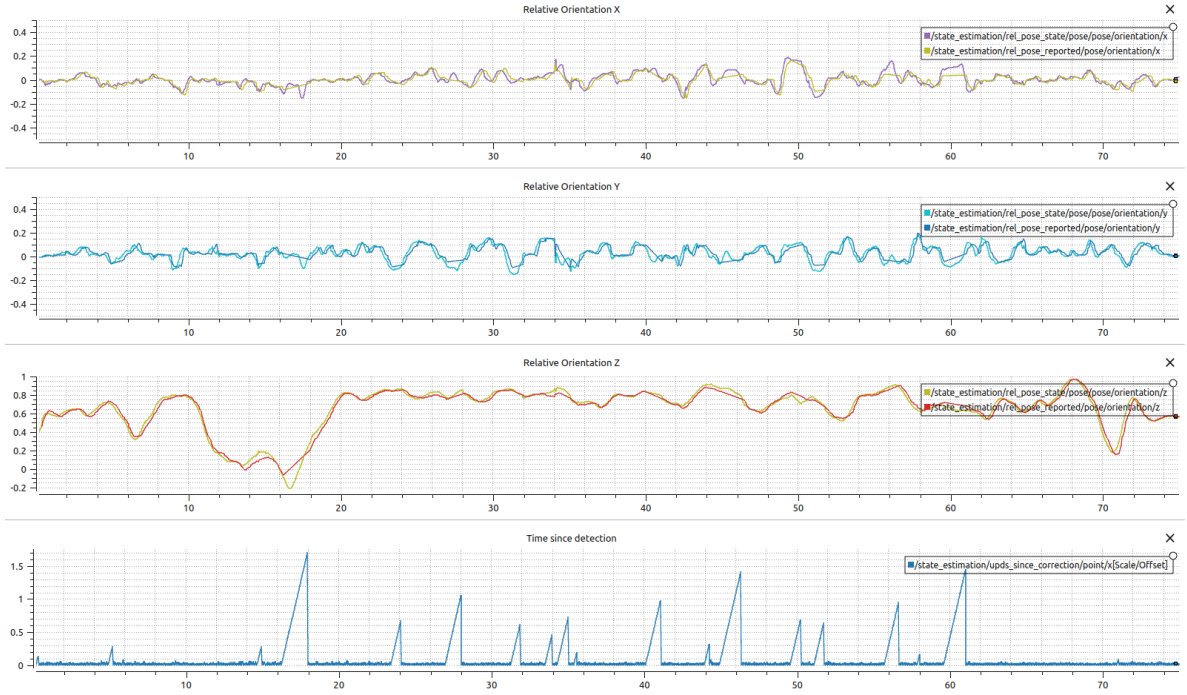

Figure 29: Orientation estimate results for the sweeping flight test case

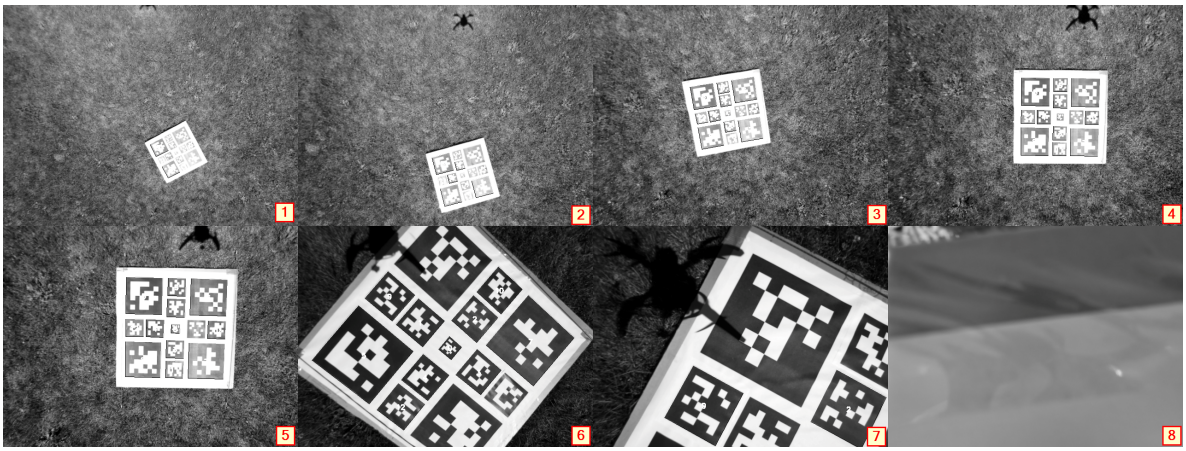

Figure 30: Sample tag detection images illustrating the landing flight test case

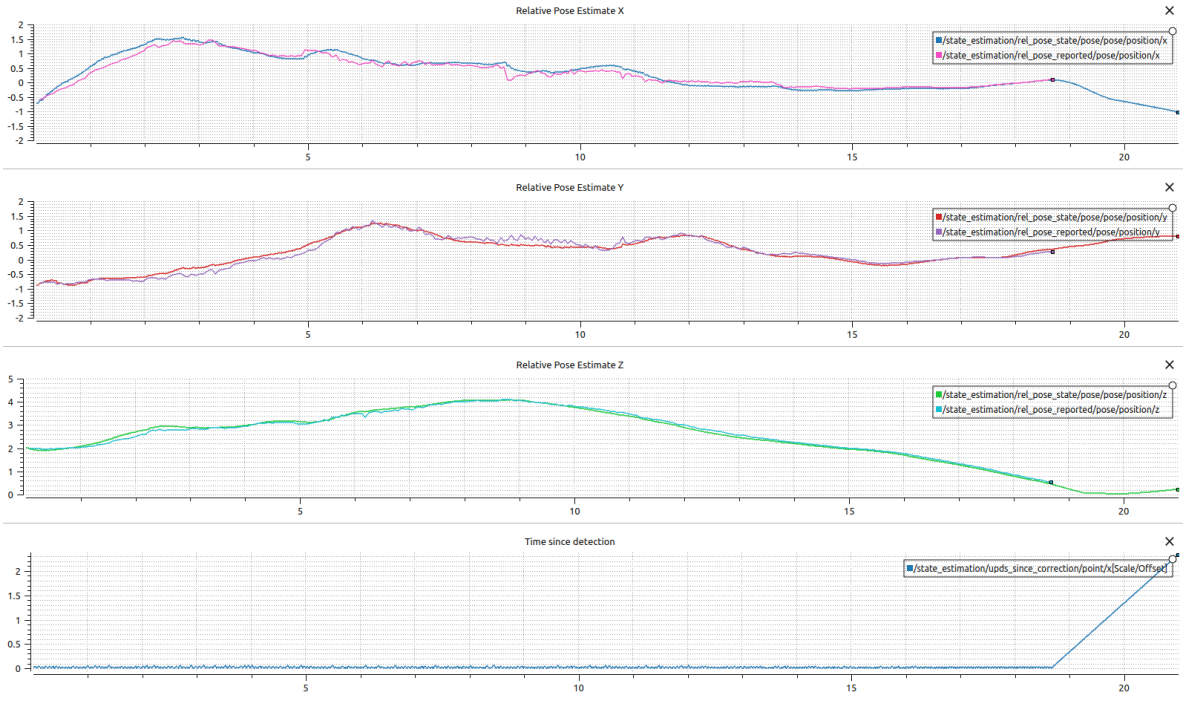

Figure 31: Position estimate results for the landing flight test case

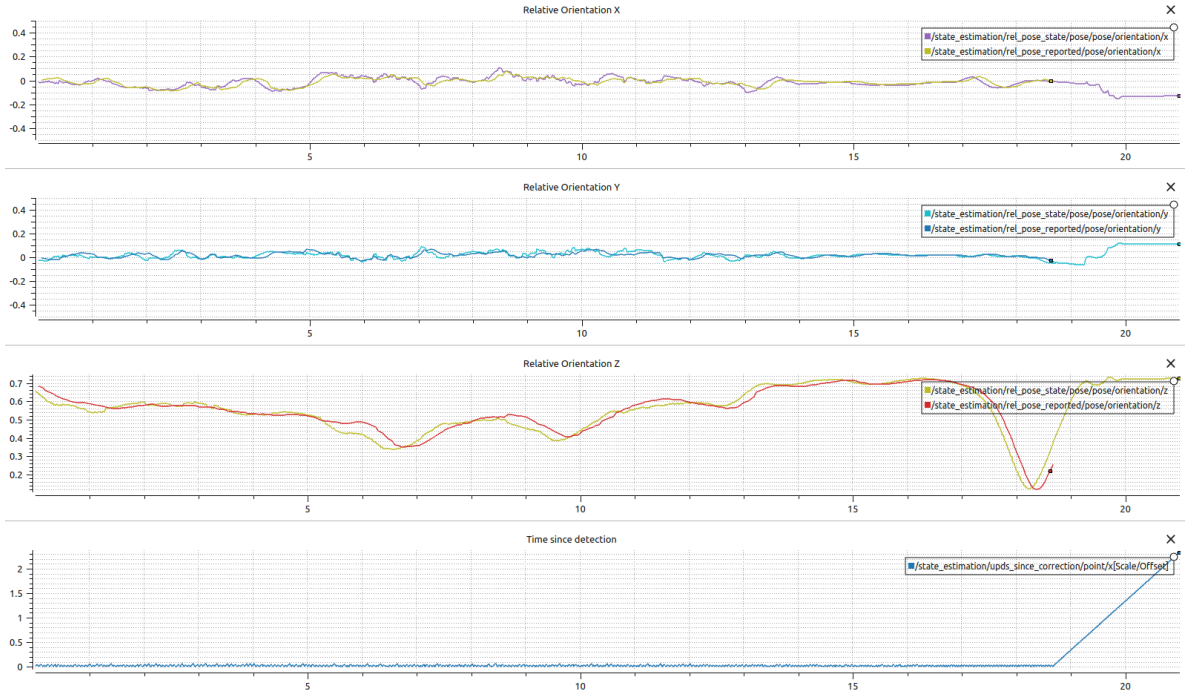

Figure 32: Orientation estimate results for the landing flight test case

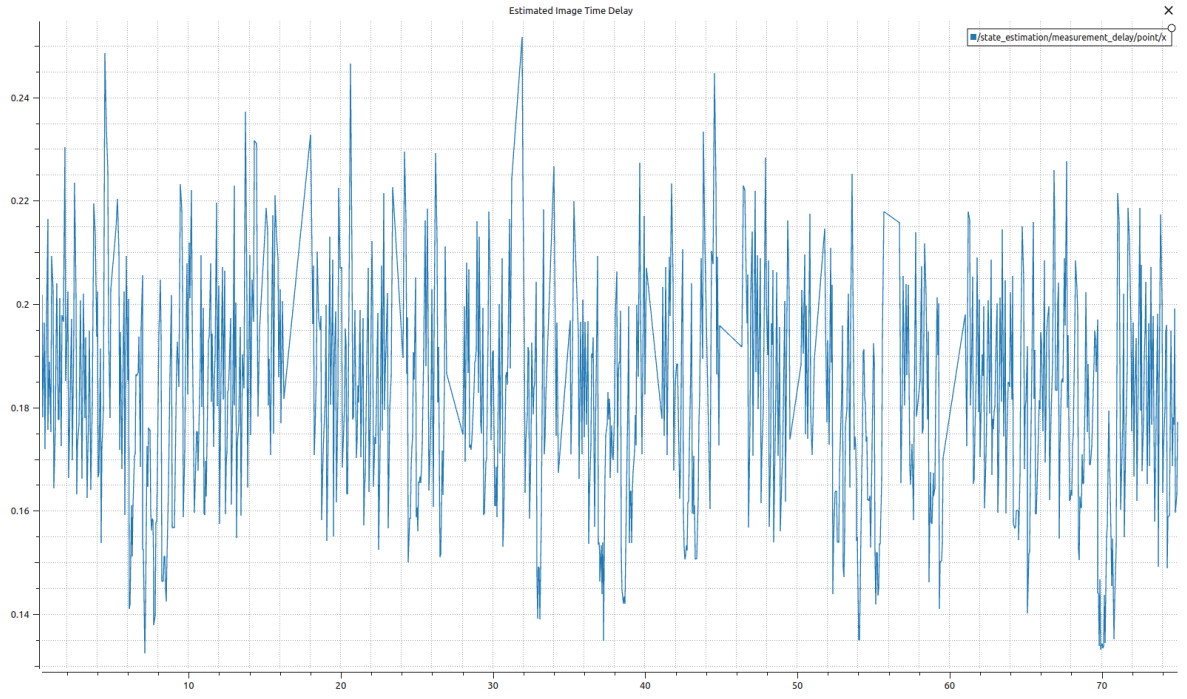

Figure 33: Computed time delay when fusing AprilTag measurements for the sweeping flight test case

7 Challenges

A number of challenges were encountered through the development of the filter that currently limit its performance.

The first was dealing with the delay present in the tag pose measurements. Due to the sequence of image processing steps involved there is a substantial delay between when the tag pose is observed by the camera and when it arrives at the filter for fusion with the IMU data. Synchronizing these two data streams is important because it determines the state that the prediction model propagates from when integrating future IMU data. This is most important for the orientation, as an error in orientation between the estimated and true state will cause gravitational acceleration to appear as a constant acceleration bias in the X or Y direction and quickly integrate to cause a quadratic growth in position estimation error until another detection is received. This means that accurately capturing the delay in the AprilTag detection is important for achieving robustness to lost tag detections when updating the filter based on predictions only becomes necessary.

As mentioned previously the filter calculates a delay based on the difference between the image timestamp and current time to which it adds a constant offset. To study the sensitivity of the estimation performance to the delay time, the indoor combined motion case was replayed in simulation using no delay as well as fixed delay times of 100, 150 and 200 ms. The resulting X and Y position estimates are shown in Figure 34 below.

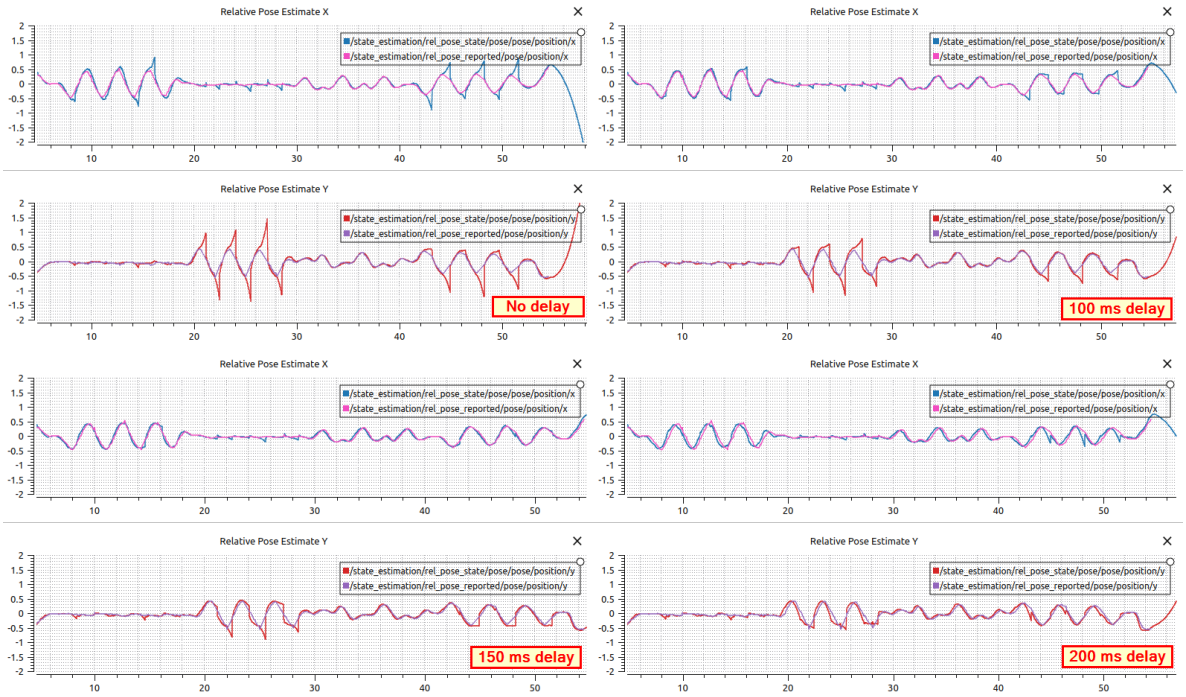

Figure 34: Position estimates in indoor combined motion case at varying assumed detection delays

It can be seen that the drift after lost detections decreases substantially when accounting for the measurement delay. It can also be seen that different points in the event benefit most from different delay times. While a delay of 200 ms shows the least drift for the Y estimate between 20-30 s, it increases the amount of drift between 40-50 s for the X estimate relative to a 150 ms delay. This is in line with our observation earlier that the delay calculated by the filter varies between measurements.

In the current hardware implementation the exact delay is uncertain as the image timestamp is approximated as when the camera driver returns from a retrieve result API call, rather than a time provided by the camera hardware or an acquisition trigger as the camera used only supports hardware triggering and not software triggering. The total delay in the AprilTag pose measurement is thus uncertain, variable and has a significant impact on the estimation accuracy. It therefore remains a

challenge to the filter's performance.

Another challenge encountered was that of spurious AprilTag orientation detections. As mentioned earlier, it has been observed that when the tag corners are close to the edges of the image the detector has a tendency to return orientation estimates that are significantly different than in the preceding frames. This effect has been observed both with only a single tag in view as well as with part of a bundle in view and is illustrated in Figures 35 and 36 below for a single tag and multiple tags respectively.

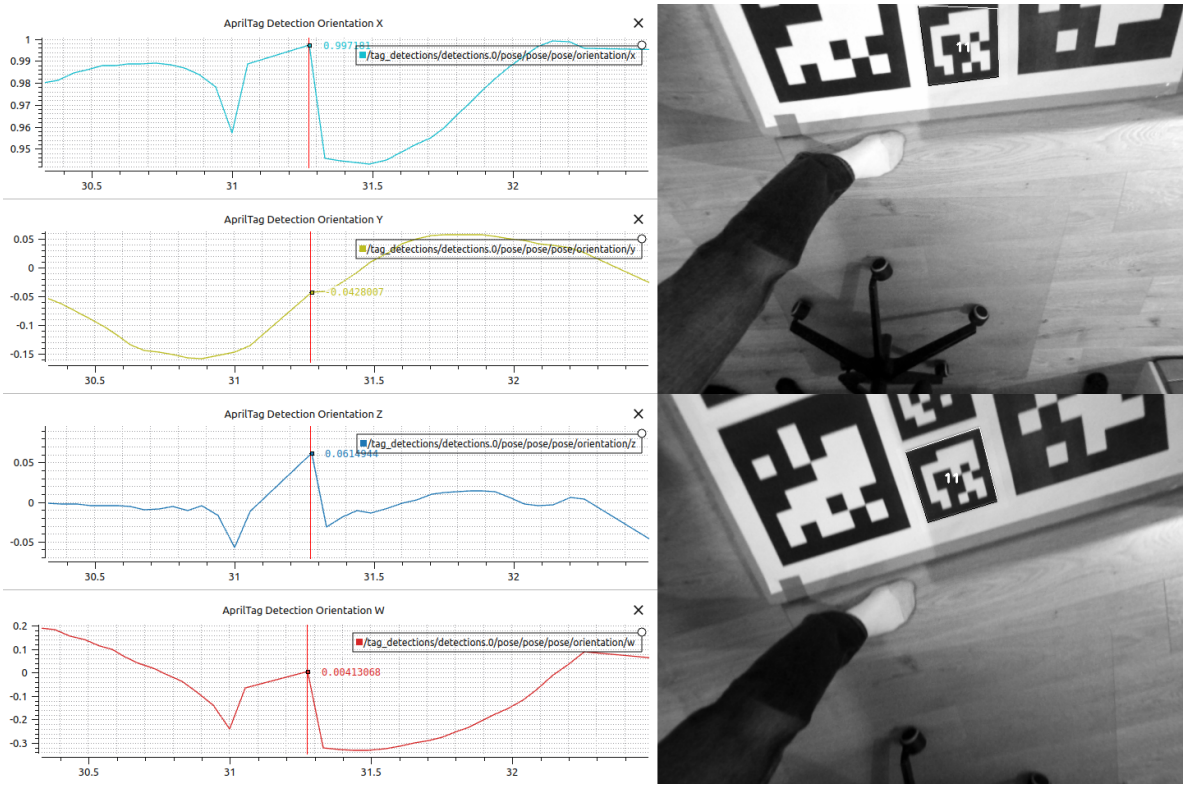

Figure 35: Example of AprilTag detector producing a spurious orientation estimate when a tag is close to the edge of the image

It can be seen in the single tag case that while the bottom half of the quaternion trace is clearly the more correct result, for the two time points highlighted the detector returns a quaternion Z component approximately 0.010 larger than expected. This corresponds to a difference of approximately 10 degrees. This appears to be due to the detector still identifying the tag even though the corner is outside the image and clipping the corner to remain in the image, skewing the orientation estimate. The position estimate on the other hand is mostly unaffected. This effect is not limited to the case of a single tag and can also occur when multiple tags in a bundle are visible as shown in the second example.

The filter does include logic for only using a detection if at least one tag has corners fully inside the image within a certain margin, but this does not appear to be capturing all cases and does not handle the second case where one detected tag is fully inside but another is not. This issue presents a challenge as the tag detections are the only absolute reference on position or orientation, and so inaccurate detections can invalidate the accuracy of future predictions based on the IMU data.

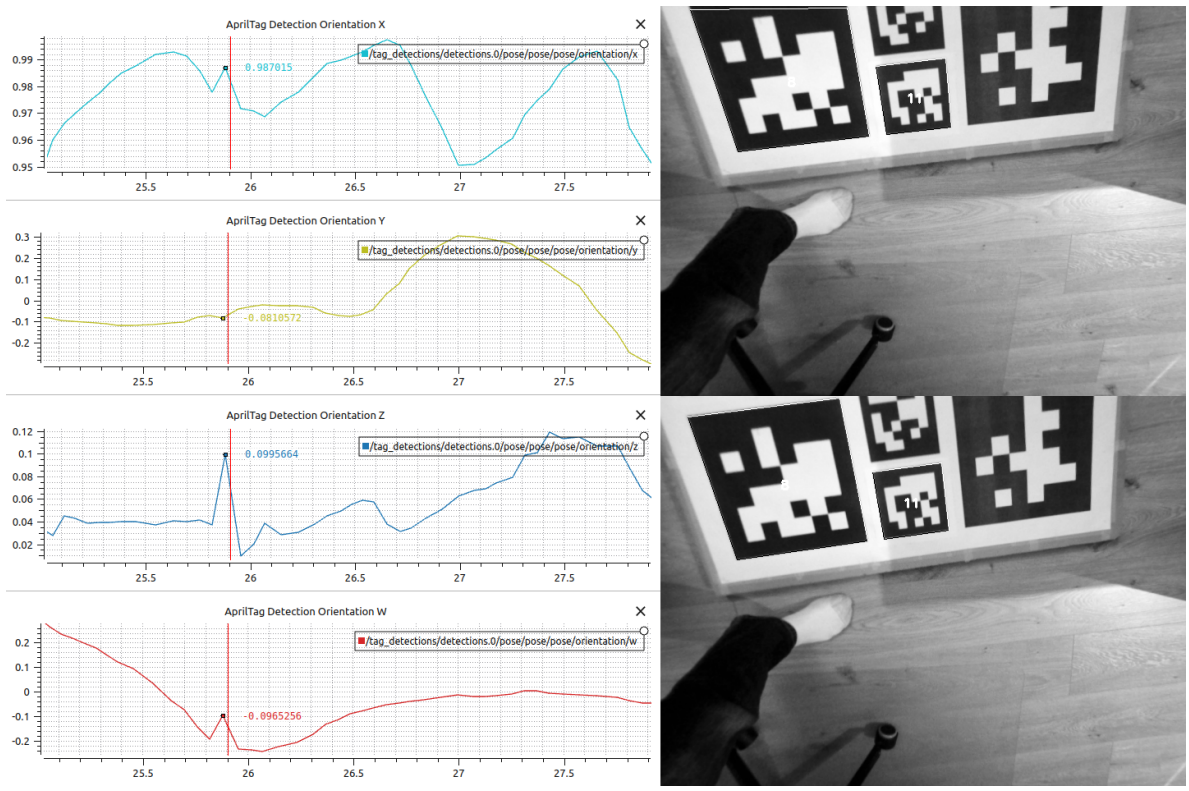

Figure 36: Example of AprilTag detector producing a spurious orientation estimate when one tag in a bundle is close to the edge of the image

The third challenge encountered has been drift when tag detections are lost. While long term drift is expected without correction in a Kalman filter, the level of drift observed has been relatively large over short time spans for small amounts of true motion. A specific example of this is illustrated in Figures 37 and 38 below, where it can be seen that rotation of the camera away from the tag during flight to maintain position results in detection being lost for 1.7 s and an ensuing drift of 1.17 m in the Y position estimate and 0.19 m in X.

The drift is driven partially by the delay and spurious detection issues mentioned previously, but the main contributor is that errors in the orientation estimate have the effect of a constant acceleration bias in the X and Y directions leading to quadratic error growth when detection is lost. This challenge is unfortunately embedded in the filter architecture and due to using the accelerometer data on the predictive side of the filter.

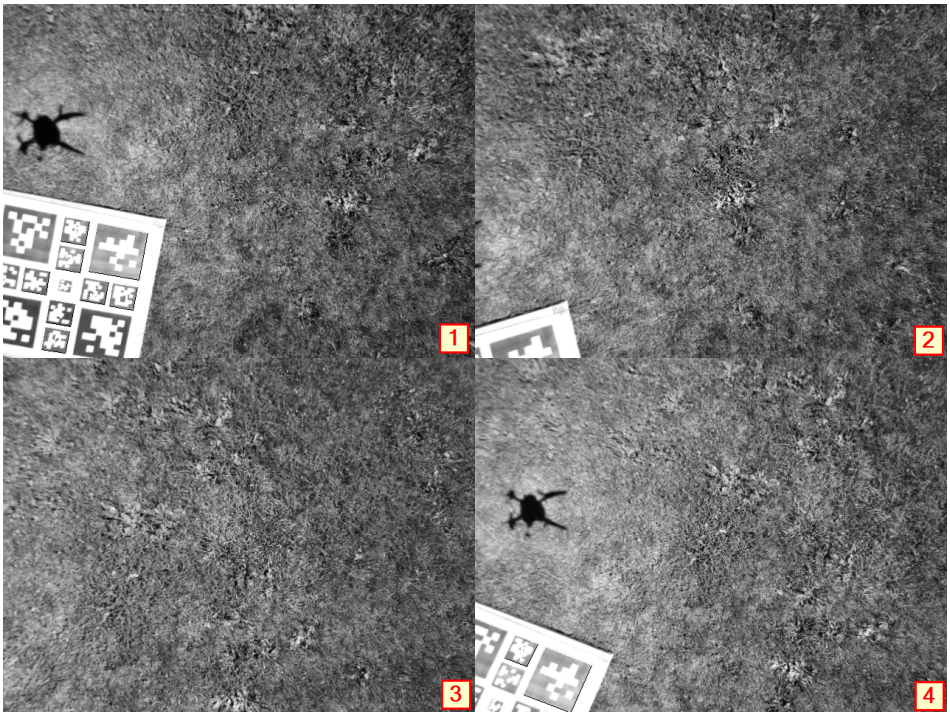

Figure 37: Sample images from a corrective maneuver in outdoor flight leading to loss of detection and drift in the position estimate

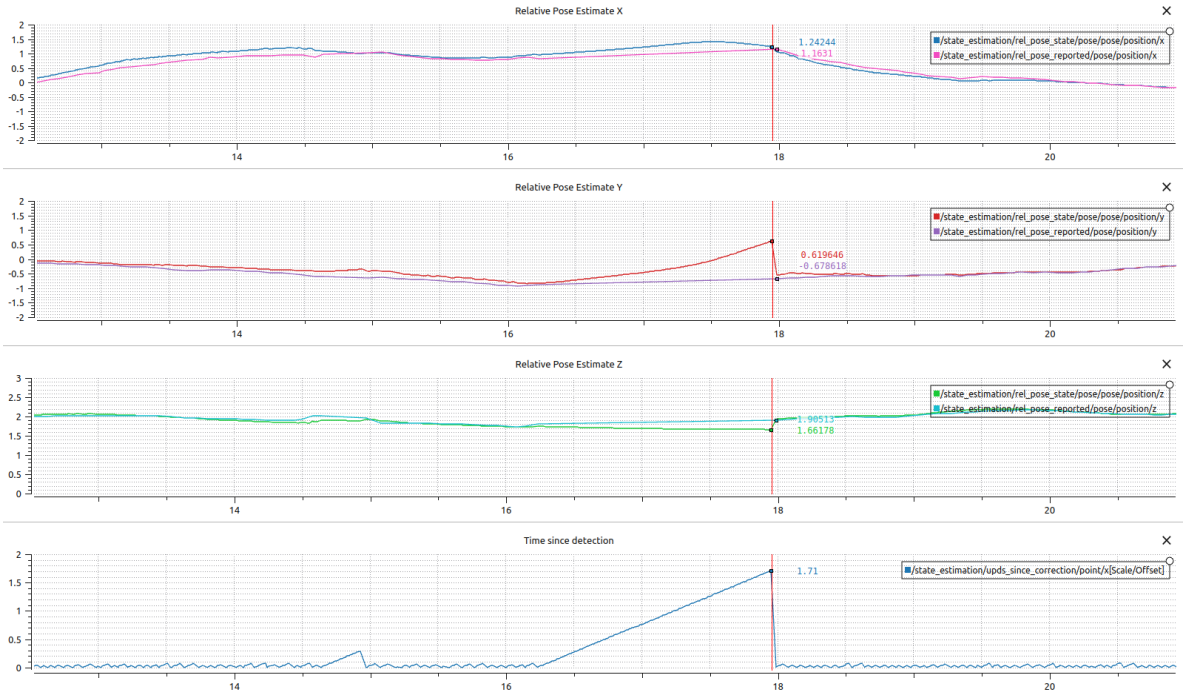

Figure 38: Estimated position in a corrective maneuver in outdoor flight displaying large drift in the position estimate

The last challenge encountered was that of tag visibility under bright sunlight. During flight testing it was found that despite appearing relatively dark and non-reflective on the ground, when viewed by the vehicle from 4 m above the tag began to reflect light and appear washed out, compromising the ability of the AprilTag detector to successfully locate the tags. This is illustrated in Figure 39 below, where despite being fully visible only 2 of 13 tags are successfully identified. As a countermeasure the camera exposure time was lowered to the minimum value allowed by the camera API of approximately 45 μs, but this was insufficient to eliminate the issue.

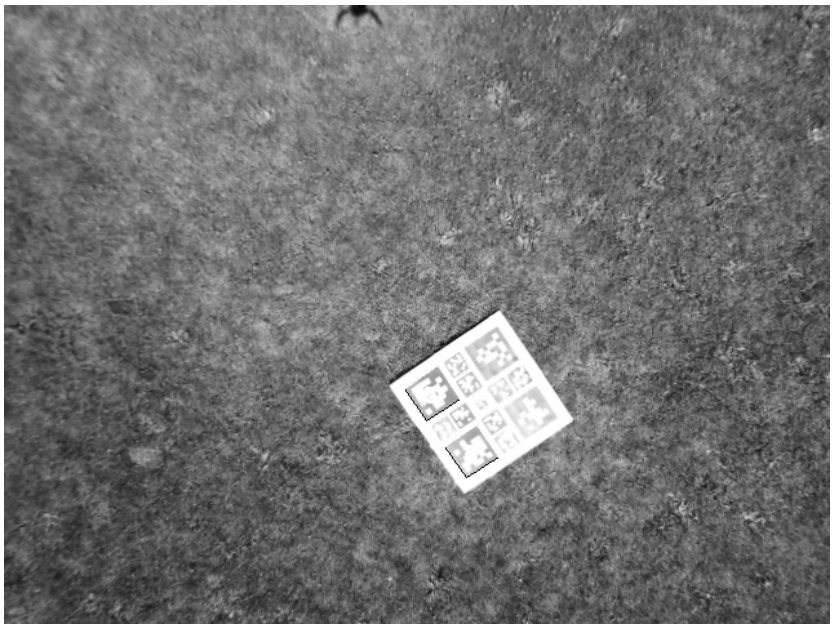

Figure 39: Sample image showing tag image becoming washed out when viewed from above in bright sunlight, leading to only 2 of 13 tags in the bundle being detected

8 Future Work

Based on the learnings over the course of the project a handful of areas have been identified for future work.

The first is to deploy the existing filter in tandem with a relative position controller and overall state machine to perform the task of relative flight and landing. While the current implementation still faces challenges in terms of drift under lost detections, it is likely still sufficient for accomplishing the task under nominal conditions and may identify additional areas for improvement.

The second general area is improvements in the vision system and how the AprilTag detections are handled. It was identified that two significant challenges that currently exist are spurious tag orientation detections and uncertainty in the detection time delay. A better method of rejecting or filtering spurious tag detections or use of an image acquisition trigger(hardware or software) could both be worthwhile future improvements. A more sophisticated camera driver capable of dynamically adjusting the image resolution and cropping the image based on the tag detections could also allow better detection at longer ranges without significantly increasing computational cost.

Improvements could also be made to the filter architecture. The filter currently only has the AprilTag detection in its measurement model, meaning that it relies on prediction only as soon as detection is lost. Adding another orientation reference for use in the absence of detections, perhaps from a simpler IMU only method such as the Mahony filter[15], could reduce the level of drift. Another option could be to use a motor thrust model in the predictive model to move the accelerometer data to the correction side of the filter [16]. Both could potentially reduce the orientation error in the absence of detections and slow the growth of error. The vehicle also has a barometer which was left unused. Making use of this could potentially allow for reduced height error during the final descent phase after detection is lost, provided that buffeting from ground effects is not too severe. As well, a more advanced filter architecture such as a Sigma Point or Iterated Extended Kalman Filter could be investigated for improved estimation accuracy.

Lastly, the image processing pipeline could be moved to the GPU of the Jetson Nano. The standard AprilTag ROS node is a CPU implementation, but recently GPU implementations have become available from NVIDIA for Jetson devices[17]. While technically available during the project time period on the Nano, this option was not pursued due to being developer releases with dependency mismatches in operating systems and ROS versions requiring complicated containerized solutions for use. Once fully available this could potentially increase the run rate, reduce the need for downsampling the image, reduce the delay time and free up CPU usage for more advanced control methodologies.

9 Conclusion

This project studied the problem of visual-inertial relative pose estimation for landing of low cost quadrotor UAVs. The problem was simplified through the use of an AprilTag fiducial marker on the target and a quaternion based MEKF was developed to estimate the relative pose along with IMU biases while accounting for delays in the vision pipeline. The filter was evaluated in simulation under sweeping flights and found to be capable of achieving estimation errors within 0.050 m at a height of 2 m above the target and 0.25 m at a height of 4 m as long as detections were maintained. Loss of tag detection was found to lead to drift in the position estimate of up to 0.61 m horizontally after 2.5 s of no detections due to orientation errors producing a constant apparent acceleration bias.

The filter was deployed on a custom quadrotor platform powered by a NVIDIA Jetson Nano capable of producing AprilTag detections at approximately 15 Hz. The filter was tested indoors under controlled motions by hand and found to produce estimates of the correct magnitude and in line with the AprilTag reported detections, although exact error measurements could not be made due to a lack of ground truth data. Drift in the position estimate upon loss of detection was again observed with a maximum drift of approximately 0.97 m after 0.5 s in the combined motion case.

The filter was also validated in outdoor flight tests including sweeping flights over the target and a simulated landing. Exact estimation error measurements again could not be made, but the predicted position estimates were in line with what was observed visually and in the recorded tag detection images based on the physical tag dimensions. Losses of detection between 0.75 - 1.5 s again resulted in drift in the position estimate of 0.5 - 1.5 m with the severity depending on the length of time for which detection was lost and the last orientation estimate.

Numerous challenges were identified through the development including variable time delay in the AprilTag pose measurements of up to 0.25 s, spurious orientation estimates from the AprilTag detector when tags are close to the edge of the image, reduction in tag visibility under bright sunlight and drift upon loss of detection as mentioned above.

Overall the filter achieves relatively good estimation performance when detections are maintained, and should be sufficient to deploy with a relative position controller and state machine for executing landing maneuvers or relative motions under nominal conditions. Many areas for future work were identified including improving the quality of the AprilTag detections and vision pipeline, updates to the filter architecture to enhance accuracy and reduce the susceptibility to drift and moving the vision pipeline to a GPU implementation.

References

[1] u-blox, "SAM-M8Q Data Sheet." https://content.u-blox.com/sites/default/files/SAM-M8Q_DataSheet_%28UBX-16012619%29.pdf, 2020. Doc No. UBX-16012619 Rev. 06.

[2] K. Ling, D. Chow, A. Das, and S. L. Waslander, "Autonomous maritime landings for low-cost vtol aerial vehicles," in *2014 Canadian Conference on Computer and Robot Vision*, pp. 32–39, 2014.

[3] A. Paris, B. T. Lopez, and J. P. How, "Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions." https://arxiv.org/abs/1909.11071, 2019.

[4] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible Layouts for Fiducial Tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1898–1903, 2019.

[5] F. L. Markley, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, 2003.

[6] S. Challa, R. J. Evans, and X. Wang, "A bayesian solution and its approximations to out-of-sequence measurement problems," *Information Fusion*, vol. 4, no. 3, p. 185–199, 2003.

[7] T. Larsen, O. Andersen, and N. Poulsen, "Incorporation of time delayed measurements in a discrete-time Kalman filter," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 4, p. 3972–3977, 1998.

[8] R. van der Merwe, E. Wan, and S. Julier, "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004.

[9] D. Malyuta, C. Brommer, D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, "Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition," *Journal of Field Robotics*, p. arXiv:1908.06381, Aug. 2019.

[10] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. Cham: Springer International Publishing, 2016.

[11] "mavros Package Summary." https://wiki.ros.org/mavros, 2022. Accessed: 21-Aug-2022.

[12] "Rotorgeeks 7075 series 2204 2300kv." https://rotorgeeks.com/motors/standard-22xx-and-larger/rotorgeeks-7075-series-2204-2300kv, 2022. Accessed: 21-Aug-2022.

[13] P. Furgale, H. Sommer, J. Maye, J. Rehder, and L. Oth, "Kalibr: A unified camera/imu calibration toolbox." https://github.com/ethz-asl/kalibr, 2022.

[14] R. Buchanan, "Allan Variance ROS." https://github.com/ori-drs/allan_variance_ros, 2022.

[15] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[16] J. Svacha, G. Loianno, and V. Kumar, "Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1109–1116, 2019.

[17] NVIDIA, "Isaac ROS Apriltag." https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_apriltag.