Handling Delayed Fusion in Vision-Augmented Inertial Navigation

Ehsan Asadi and Carlo L. Bottasso
Department of Aerospace Engineering, Politecnico di Milano, Milano, Italy

Keywords: Delayed Fusion, Vision-Augmented INS, State Estimation.

Abstract: In this paper we consider the effects of delay caused by real-time image acquisition and feature tracking in a previously documented Vision-Augmented Inertial Navigation System. At first, the paper illustrates how delay caused by image processing, if not explicitly taken into account, can lead to appreciable performance degradation of the estimator. Next, three different existing methods of delayed fusion are considered and compared. Simulations and Monte Carlo analyses are used to assess the estimation error and computational effort of the various methods. Finally, a best performing formulation is identified, that properly handles the fusion of delayed measurements in the estimator without increasing the time burden of the filter.

1 INTRODUCTION

In this paper we consider the handling of delay associated with tracked feature points within a previously documented Vision-Augmented Inertial Navigation System (VA-INS) (Bottasso and Leonello, 2009).

Navigation approaches often use vision systems, since these are among the most information-rich sensors for autonomous positioning and mapping purposes (Bonin-Fontand et al., 2008). Vision-based navigation systems have been in use in numerous applications such as Autonomous Ground Vehicles (AGV) and underwater environments (Dalgleish et al., 2005). Recently, they have been gaining increased attention also in the field of Unmanned Aerial Vehicles (UAV) (Liu and Dai, 2010). Vision systems provide long range, high resolution measurements with low power consumption and low cost. On the other hand, they are usually associated with rather low sample rates, since they often require complex processing of the acquired images, and this limits and hinders their usability in fast and real-time applications such as UAVs.

Several attempts have already been documented in the design and implementation of robust visual odometry systems (Nister et al., 2006; Goedeme et al., 2007). Some authors have proposed the incorporation of inertial measurements as model inputs (Roumeliotis et al., 2002) or states (Qian et al., 2001; Veth et al., 2006; Mourikis and Roumeliotis, 2007), using variants of the Kalman filtering approach to robustly estimate the vehicle motion. The VA-INS of Bottasso and Leonello (2009) combined in a synergistic way vision-based sensors together with classical inertial navigation ones. The method made use of an Extended Kalman Filter (EKF), assuming that all measurements were available with no delay.

However, latency due to the extraction of information from images in real-time applications is one of the factors affecting accuracy and robustness of vision-based navigation systems (Jones and Soatto, 2011). Since image processing procedures required for tracking feature points between stereo images and across time steps are time consuming tasks, visual observations are generated with delay. If delays are small or the estimation is performed off-line, then the use of a classic filtering approach leads to acceptable results. Otherwise, the quality of the estimates is affected by the magnitude of the delay. Consequently, it becomes important to understand how to account for such delay in a consistent manner, without at the same time excessively increasing the computational burden of the filter.

Measurement delay has been the subject of numerous investigations, for example in the context of systems requiring long time visual processing (Pornsarayouth and Wongsaisuwan, 2009). If the delay is rather small, a simple solution is to ignore it, but this implies that the estimates are not optimal and their quality may be affected. Another straightforward method to handle delay is to completely recalculate the filter during the delay period as measurements arrive. Usually this method cannot be used in practical applications because of its large storage cost

Asadi E. and L. Bottasso C..
Handling Delayed Fusion in Vision-Augmented Inertial Navigation.
DOI: 10.5220/0004041403940401
In *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2012), pages 394-401
ISBN: 978-989-8565-21-1

and computational burden. In chemical and biochemical processes, methods have been proposed based on the augmentation of the states (Gopalakrishnan et al., 2011; Tatiraju et al., 1999). Other documented methods fuse delayed measurements as they arrive (Alexander, 1991; Larsen et al., 1998). These methods are effectively implemented in tracking and navigation systems for handling delays associated with the Global Positioning System (GPS).

The aim of this paper is to present a modification of the VA-INS of Bottasso and Leonello (2009), based on a delayed fusion process. In the new formulation, tracked feature points associated with delay are incorporated as delayed measurements in a multi-rate multi-sensor data fusion process using a non-linear estimator. More specifically, the paper:

- Analyzes the effects of delay caused by image processing on state estimation, when such delay is not explicitly accounted for in the estimator;
- Considers implementation issues and assesses the performance of three existing delayed fusion methods to incorporate delayed vision-based measurements in the estimator;
- Assesses the quality of the various formulations and identifies the most promising one, in terms of computational burden of the filter and of the quality of its estimates, using simulation experiments and Monte Carlo analysis.

2 VISION-AUGMENTED INERTIAL NAVIGATION

Bottasso and Leonello (2009) proposed a VA-INS to achieve higher precision in the estimation of the vehicle motion. Their implementation used relatively low resolution and relatively high noise low-cost small-size cameras, that can be mounted on-board small Rotorcraft Unmanned Aerial Vehicles (RAUVs). In this approach, the sensor readings of a standard inertial measurement unit (a triaxial accelerometer and gyro, a triaxial magnetometer, a GPS and a sonar altimeter) are fused within an EKF together with the outputs of so-called vision-based motion sensors. The overall architecture of the system is briefly reviewed here.

2.1 Kinematics

The sensor configuration and reference frames used in the kinematic modeling of the system are depicted in Fig. 1.

The inertial frame of reference is centered at point O and denoted by a triad of unit vectors $\mathcal{E} \doteq (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, pointing North, East and down (NED navigational system). A body-attached frame has origin in the generic material point B of the vehicle and has a triad of unit vectors $\mathcal{B} \doteq (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$.

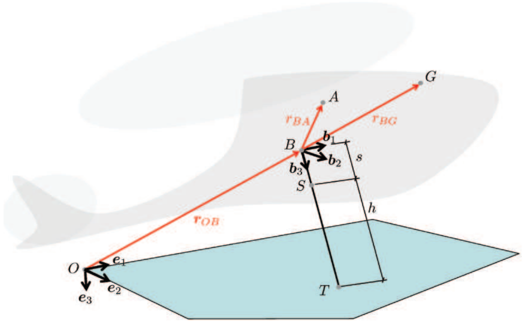


Figure 1: Reference frames and location of sensors.

The components of the acceleration in the body-attached frame are sensed by an accelerometer located at point A on the vehicle. The accelerometer yields a reading $\boldsymbol{a}_{\text{acc}}$ affected by noise $\boldsymbol{n}_{\text{acc}}$:

$$\boldsymbol{a}_{\text{acc}} = \boldsymbol{g}^{\mathcal{B}} - \boldsymbol{a}^{\mathcal{B}}_{A} + \boldsymbol{n}_{\text{acc}}. \tag{1}$$

In this expression, $\boldsymbol{g}^{\mathcal{B}}$ indicates the body-attached components of the acceleration of gravity, where $\boldsymbol{g}^{\mathcal{B}} = \boldsymbol{R}^T \boldsymbol{g}^{\mathcal{E}}$ with $\boldsymbol{g}^{\mathcal{E}} = (0, 0, g)^T$, while $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{q})$ are the components of the rotation tensor which brings triad $\mathcal{E}$ into triad $\mathcal{B}$.

Gyroscopes measure the body-attached components of the angular velocity vector, yielding a reading $\boldsymbol{\omega}_{\text{gyro}}$ affected by a noise disturbance $\boldsymbol{n}_{\text{gyro}}$:

$$\boldsymbol{\omega}_{\text{gyro}} = \boldsymbol{\omega}^{\mathcal{B}} + \boldsymbol{n}_{\text{gyro}}. \tag{2}$$

The kinematic equations, describing the motion of the body-attached reference frame with respect to the inertial one, can be written as

$$\dot{\boldsymbol{v}}^{\mathcal{E}}_{B} = \boldsymbol{g}^{\mathcal{E}} - \boldsymbol{R}[\boldsymbol{a}_{\text{acc}} + \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{r}^{\mathcal{B}}_{BA} + \boldsymbol{\alpha}^{\mathcal{B}} \times \boldsymbol{r}^{\mathcal{B}}_{BA}] + \boldsymbol{R}\boldsymbol{n}_{\text{acc}}, \tag{3a}$$

$$\dot{\boldsymbol{\omega}}^{\mathcal{B}} = \boldsymbol{\alpha}_h(\boldsymbol{\omega}_{\text{gyro}} + \boldsymbol{n}_{\text{gyro}}), \tag{3b}$$

$$\dot{\boldsymbol{r}}^{\mathcal{E}}_{OB} = \boldsymbol{v}^{\mathcal{E}}_{B}, \tag{3c}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{T}(\boldsymbol{\omega}^{\mathcal{B}})\boldsymbol{q}, \tag{3d}$$

where $\boldsymbol{v}_B$ is the velocity of point B, $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{\alpha}$ the angular acceleration, while $\boldsymbol{r}_{BA}$ is the position vector from point B to point A and $\boldsymbol{r}_{OB}$ is from point O to point B. Finally $\boldsymbol{q}$ are rotation parameters, which are chosen here as quaternions, so that matrix $\boldsymbol{T}$ can be written as

$$\boldsymbol{T}(\boldsymbol{\omega}^{\mathcal{B}}) = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^{\mathcal{B}^T} \\ \boldsymbol{\omega}^{\mathcal{B}} & -\boldsymbol{\omega}^{\mathcal{B}}_{\times} \end{bmatrix}. \tag{4}$$

Gyro measures are used in Eq. (3b) for computing an estimate of the angular acceleration. Since this implies a differentiation of the gyro measures, assuming a constant (or slowly varying) bias over the differentiation interval, knowledge of the bias becomes unnecessary. Hence, the angular acceleration is computed as

$$\boldsymbol{\alpha}^{\mathcal{B}} \simeq \boldsymbol{\alpha}_h(\boldsymbol{\omega}_{\text{gyro}}), \tag{5}$$

where $\boldsymbol{\alpha}_h$ is a discrete differentiation operator. The angular acceleration at time t_k is computed according to the following three-point stencil formula based on a parabolic interpolation $\boldsymbol{\alpha}_h(t_k) = \big(3\boldsymbol{\omega}_{\text{gyro}}(t_k) - 4\boldsymbol{\omega}_{\text{gyro}}(t_{k-1}) + \boldsymbol{\omega}_{\text{gyro}}(t_{k-2})\big)/(2h)$, where $h = t_k - t_{k-1} = t_{k-1} - t_{k-2}$.

A GPS is located at point G on the vehicle (see Fig. 1). The velocity and position vectors of point G, noted respectively $\boldsymbol{v}_G^{\mathcal{E}}$ and $\boldsymbol{r}_{OG}^{\mathcal{E}}$, can be expressed as

$$\boldsymbol{v}_G^{\mathcal{E}} = \boldsymbol{v}_B^{\mathcal{E}} + \boldsymbol{R}\boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{r}_{BG}^{\mathcal{B}}, \tag{6a}$$

$$\boldsymbol{r}_{OG}^{\mathcal{E}} = \boldsymbol{r}_{OB}^{\mathcal{E}} + \boldsymbol{R}\boldsymbol{r}_{BG}^{\mathcal{B}}. \tag{6b}$$

The GPS yields measurements of the position and velocity of point G affected by noise, i.e.

$$\boldsymbol{v}_{\text{gps}} = \boldsymbol{v}_G^{\mathcal{E}} + \boldsymbol{n}_{v_{\text{gps}}}, \tag{7a}$$

$$\boldsymbol{r}_{\text{gps}} = \boldsymbol{r}_{OG}^{\mathcal{E}} + \boldsymbol{n}_{r_{\text{gps}}}. \tag{7b}$$

A sonar altimeter measures the distance h along the body-attached vector $\boldsymbol{b}_3$, between its location at point S and point T on the terrain. The sonar altimeter yields a reading h_{sonar} affected by noise n_{sonar}, i.e.

$$h = r_{OB_3}^{\mathcal{E}}/R_{33} - s, \tag{8}$$

$$h_{\text{sonar}} = h + n_{\text{sonar}}, \tag{9}$$

where $r_{OB_3}^{\mathcal{E}} = \boldsymbol{r}_{OB} \cdot \boldsymbol{e}_3$ and $\boldsymbol{R} = [R_{ij}]$, $i,j = 1,2,3$.

Furthermore, we consider a magnetometer which senses the magnetic field $\boldsymbol{m}$ of the Earth in the body-attached system $\mathcal{B}$. The inertial components $\boldsymbol{m}^{\mathcal{E}}$ are assumed to be known and constant in the area of operation of the vehicle. The magnetometer yields a measurement $\boldsymbol{m}_{\text{magn}}$ affected by noise $\boldsymbol{n}_{\text{magn}}$, i.e.

$$\boldsymbol{m}^{\mathcal{B}} = \boldsymbol{R}^T \boldsymbol{m}^{\mathcal{E}}, \tag{10}$$

$$\boldsymbol{m}_{\text{magn}} = \boldsymbol{m}^{\mathcal{B}} + \boldsymbol{n}_{\text{magn}}. \tag{11}$$

Finally, considering a pair of stereo cameras located on the vehicle (see Fig. 2), a triad of unit vectors $\mathcal{C} \doteq (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ has origin at the optical center C of the left camera, where $\boldsymbol{c}_1$ is directed along the horizontal scanlines of the image plane, while $\boldsymbol{c}_3$ is parallel to the optical axis, pointing towards the scene. Considering that P is a fixed point, the vision-based observation model , discretized across two consecutive time instants t_k and $t_{k+1} = t_k + \Delta t$, is

$$\begin{aligned} \boldsymbol{d}(t_{k+1})^{\mathcal{C}_{k+1}} = & -\Delta t\, \boldsymbol{C}^T \Big(\boldsymbol{R}(t_{k+1})^T \boldsymbol{v}^{\mathcal{E}}(t_{k+1}) \\ & + \boldsymbol{\omega}^{\mathcal{B}}(t_{k+1}) \times (\boldsymbol{c}^{\mathcal{B}} + \boldsymbol{C}\boldsymbol{d}(t_k)^{\mathcal{C}_k})\Big) + \boldsymbol{d}(t_k)^{\mathcal{C}_k}, \end{aligned} \tag{12}$$

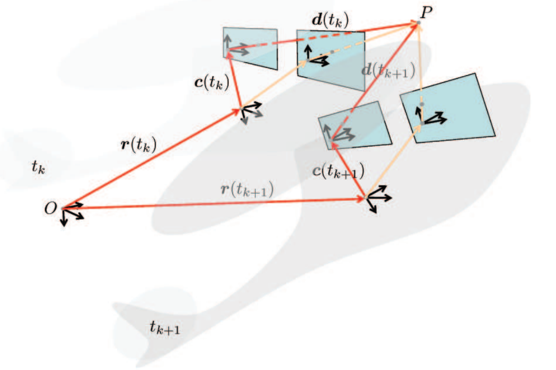


Figure 2: Geometry for the derivation of the discrete vision-based motion sensor.

where $\boldsymbol{C}$ are the components of the rotation tensor which brings triad $\mathcal{B}$ into triad $\mathcal{C}$. The tracked feature point distances are noted $\boldsymbol{d}^{\mathcal{C}}$ and $\boldsymbol{d}^{\mathcal{C}'}$ for the left and right cameras, respectively, and are obtained by stereo reconstruction using $\boldsymbol{d}^{\mathcal{C}} = \boldsymbol{p}^{\mathcal{C}}\, b/d$, where $\boldsymbol{p}$ is the position vector of the feature point on the image plane, b is the stereo baseline and d the disparity. This process yields at each time step t_{k+1} an estimate $\boldsymbol{d}_{\text{vsn}}$ affected by noise $\boldsymbol{n}_{\text{vsn}}$

$$\boldsymbol{d}_{\text{vsn}} = \boldsymbol{d}(t_{k+1})^{\mathcal{C}_{k+1}} + \boldsymbol{n}_{\text{vsn}}, \tag{13}$$

for the left camera, and a similar expression for the right one.

2.2 Process Model and Observations

The estimator is based on the following state-space model

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\nu}(t)\big), \tag{14a}$$

$$\boldsymbol{y}(t_k) = \boldsymbol{h}\big(\boldsymbol{x}(t_k)\big), \tag{14b}$$

$$\boldsymbol{z}(t_k) = \boldsymbol{y}(t_k) + \boldsymbol{\mu}(t_k), \tag{14c}$$

where the state vector $\boldsymbol{x}$ is defined as

$$\boldsymbol{x} \doteq (\boldsymbol{v}_B^{\mathcal{E}\,T}, \boldsymbol{\omega}^{\mathcal{B}\,T}, \boldsymbol{r}_{OB}^{\mathcal{E}\,T}, \boldsymbol{q})^T. \tag{15}$$

Function $\boldsymbol{f}(\cdot,\cdot,\cdot)$ in Eq. (14a) represents in compact form the rigid body kinematics expressed by Eqs. (3). The input vector $\boldsymbol{u}$ appearing in Eq. (14a) is defined as measurements provided by the accelerometers $\boldsymbol{a}_{\text{acc}}$ and gyros $\boldsymbol{\omega}_{\text{gyro}}$, and $\boldsymbol{\nu}$ is their associated measurement noise vector.

Similarly, Eqs. (6), (8), (10) and (12) may be gathered together and written in compact form as an observation model $\boldsymbol{h}(\cdot)$ expressed by Eqs. (14b), where the vector of outputs $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = (\boldsymbol{v}_G^{\mathcal{E}\,T}, \boldsymbol{r}_{OG}^{\mathcal{E}\,T}, h, \boldsymbol{m}^{\mathcal{B}\,T}, \ldots, \boldsymbol{d}^{\mathcal{C}\,T}, \boldsymbol{d}^{\mathcal{C}'T}, \ldots)^T. \tag{16}$$

The definition of model (14) is complemented by the vector of measurements $\boldsymbol{z}$ and associated noise $\boldsymbol{\mu}$ vectors

$$\boldsymbol{z} \doteq (\boldsymbol{v}_{\text{gps}}^T, \boldsymbol{r}_{\text{gps}}^T, h_{\text{sonar}}, \boldsymbol{m}_{\text{magn}}^T, \ldots, \boldsymbol{d}_{\text{vsn}}^T, \boldsymbol{d}_{\text{vsn}}^{T\prime})^T, \quad (17a)$$

$$\boldsymbol{\mu} \doteq (\boldsymbol{n}_{v_{\text{gps}}}^T, \boldsymbol{n}_{r_{\text{gps}}}^T, n_{\text{sonar}}, \boldsymbol{n}_{\text{magn}}^T, \ldots, \boldsymbol{n}_{\text{vsn}}^T, \boldsymbol{n}_{\text{vsn}}^T)^T. \quad (17b)$$

The state estimation problem expressed by Eqs. (14–17) was solved using the EKF approach, initially assuming that all measurements are available with no delay.

2.3 Classic State Estimation using EKF

The EKF formulation is briefly reviewed here using the time-discrete form of Eqs. (14) and assuming $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$ to be white noises with covariance $\boldsymbol{Q}$ and $\boldsymbol{U}$. The prediction stage of states and observations is performed by using the non-linear model equations,

$$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, 0)\,\Delta t, \quad (18a)$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-), \quad (18b)$$

whereas a linear approximation is used for estimating the error covariance and computing the Kalman gain matrices,

$$\boldsymbol{P}_k^- = \boldsymbol{A}_k\,\boldsymbol{P}_{k-1}\,\boldsymbol{A}_k{}^T + \boldsymbol{G}_k\,\boldsymbol{Q}_k\,\boldsymbol{G}_k{}^T, \quad (19a)$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\,\boldsymbol{H}_k{}^T\,\left[\boldsymbol{H}_k\,\boldsymbol{P}_{k-1}\,\boldsymbol{H}_k{}^T + \boldsymbol{U}_k\right]^{-1}. \quad (19b)$$

Matrices $\boldsymbol{A}_k$, $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ are computed by linearizing the non-linear model about the current estimate,

$$\boldsymbol{A}_k = \boldsymbol{I} + \Delta t\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}, \quad \boldsymbol{G}_k = \Delta t\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\nu}}, \quad \boldsymbol{H}_k = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}. \quad (20)$$

Finally, covariance updates and state estimates are computed as

$$\boldsymbol{P}_k = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\,\boldsymbol{P}_k^-, \quad (21a)$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left[\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)\right]. \quad (21b)$$

2.4 Image Processing and Tracking

The idea of VA-INS is based on tracking scene points between stereo images and across time steps, to express the apparent motion of the tracked points in terms of the motion of the vehicle. The identification and matching of feature points is begun with the acquisition of the images; then, strong corners are extracted from the left image with the feature extractor of the KLT tracker (Jianbo and Tomasi, 1994), and a dense disparity map is obtained. Identified feature points are encoded using the BRIEF descriptor (Calonder et al., 2010), and subsequently matches in a transformed image are found by computing distances between descriptors. This descriptor is in general competitive with algorithms like SURF and SIFT, and much faster in terms of generation and matching.

A real-time implementation of the system was based on an on-board PC-104 with a 1.6 GHz CPU and 512 Mb of volatile memory, with the purpose of analyzing the performance and computational effort of the feature tracking process. Images were captured by a Point Grey Bumblebee XB3 stereo vision camera, and resized images with a resolution of 640x480 were used for tracking 100 points between frames. These tests indicate the presence of a 490 millisecond latency (see table 1) between the instant the image is captured and the time the state estimator receives the required visual information.

Table 1: Time consumption of image processing tasks.

Process Task	Computing Time
Image acquisition	100 ms
Resizing, rectification	40 ms
Dense disparity mapping	150 ms
Feature extraction	130 ms
Feature description	50 ms
Feature matching	20 ms
TOTAL	490 ms

3 DELAYED FUSION IN VA-INS

Simulation analyses, presented later, show that the half a second delay of the system is significant enough not to be neglected. In other words, directly feeding this delayed vision-based measurements to the EKF estimator will affect the quality of the estimates.

The outputs of the vision-based motion sensors $\boldsymbol{d}_{vsn}(s)$ and $\boldsymbol{d}'_{vsn}(s)$ from a captured image at time s will only be available at time $k = s + N$, where N is the sample delay. Such delayed outputs are labeled $\boldsymbol{d}_{vsn}^*(k)$ and $\boldsymbol{d}'^*_{vsn}(k)$. On the other hand, measurements from other sensors are not affected by such delay, and are available at each sampling time. For the purpose of handling multi-rate estimation and delay, observations are here partitioned in two groups, one collecting multi-rate non-delayed GPS, sonar and magnetometer readings (labeled rt, for real-time), and the other collecting delayed vision-based observations (labeled vsn, for vision):

$$\boldsymbol{z}^{rt} \doteq (\boldsymbol{v}_{gps}^T, \boldsymbol{r}_{gps}^T, \boldsymbol{h}_{sonar}, \boldsymbol{m}_{magn}^T)^T, \quad (22a)$$

$$\boldsymbol{z}^{vsn^*} \doteq (\boldsymbol{d}_{vsn(1)}^{*T}, \boldsymbol{d}_{vsn(2)}^{*T}, \ldots, \boldsymbol{d}_{vsn(n)}^{*T})^T. \quad (22b)$$

The state estimation process is based on using a proper EKF update for each group. More specifically, the recalculation, Alexander (Alexander, 1991) and Larsen (Larsen et al., 1998) methods are surveyed here for fusing delayed tracked points in the VA-INS structure as they arrive. All methods are briefly reviewed in the following.

3.1 Recalculation Method

A straightforward estimate can be obtained simply by recalculating the filter throughout the delay period. As the vision-based measurements are not available in the time interval between s to k, one may update states and covariance using only non-delayed measurements in this time interval. As soon as vision measurements originally captured at time s are received with delay at time k, the estimation procedure begins from s by repeating the update procedure while incorporating both non-delayed measurements and lagged vision-based measurements.

The computational burden of this implementation of the filter in the VA-INS is critical, because of the need of fusing a fairly large set of measurements. Therefore the approach, although rigorous and straightforward, is not a good candidate for the implementation on-board small size aerial vehicles.

3.2 Alexander Method

In this method (Alexander, 1991), a correction term $\boldsymbol{M}_*$ is calculated based on Kalman information and added to the filter estimates when the delayed measurements are received:

$$\boldsymbol{M}_* = \prod_{i=1}^{N} \left(\boldsymbol{I} - \boldsymbol{K}'_{s+i} \boldsymbol{H}^{rt}_{s+i} \right) \boldsymbol{A}_{s+i-1}. \tag{23}$$

In the above equation, $\boldsymbol{K}'_{k-i}$ is used to distinguish it from $\boldsymbol{K}^{rt}_{k-i}$. The Kalman gain and error covariance are updated at time s as if the measurements were available without delay. Then, at time k when measurements $\boldsymbol{z}^{vsn^*}_k$ become available with delay, their incorporation and the state update is obtained by using the following correction term in the Kalman equation

$$\delta\hat{\boldsymbol{x}}_k = \boldsymbol{M}_* \boldsymbol{K}^{vsn}_s \left(\boldsymbol{z}^{vsn^*}_k - \boldsymbol{H}^{vsn^*}_s \hat{\boldsymbol{x}}_s \right). \tag{24}$$

The problem of implementing Alexander method in the VA-INS arises since it is not possible to identify which points are tracked until all image processing tasks at time k are completed. Consequently, the global measurement model $\boldsymbol{H}^{vsn^*}_s$, including the submodels of all tracked feature points in a new scene, is unknown at time s. Moreover, the uncertainty $\boldsymbol{U}^{vsn^*}_s$ related to each point is unknown, since this is changed by its distance and position in the image plane.

3.3 Larsen Method

In our VA-INS, the successive tracked points, their uncertainty and consequently the exact measurement model will be unknown until images are completely processed. Therefore, a method is needed that does not require information about $\boldsymbol{z}^{vsn^*}_k$ until new measurements arrive.

Larsen extended Alexander approach, by extrapolating delayed measurements to the present ones (Larsen et al., 1998):

$$\boldsymbol{z}^{vsn(int)}_k = \boldsymbol{z}^{vsn^*}_k + \boldsymbol{H}^{vsn^*}_k \hat{\boldsymbol{x}}_k - \boldsymbol{H}^{vsn^*}_s \hat{\boldsymbol{x}}_s. \tag{25}$$

Larsen shows that the correction term is calculated based on Kalman information in a way that closely resembles Alexander method, i.e.

$$\boldsymbol{M}_* = \prod_{i=1}^{N} \left(\boldsymbol{I} - \boldsymbol{K}^{rt}_{s+i} \boldsymbol{H}^{rt}_{s+i} \right) \boldsymbol{A}_{s+i-1}, \tag{26}$$

where the Kalman gain and covariance are kept frozen until the delayed measurements arrive. Once this happens, they are updated in a simple and fast manner:

$$\boldsymbol{K}^{vsn}_k = \boldsymbol{M}_* \boldsymbol{P}_s \boldsymbol{H}^{vsn^*T}_s \left[\boldsymbol{H}^{vsn^*}_s \boldsymbol{P}_s \boldsymbol{H}^{vsn^*T}_s + \boldsymbol{U}^{vsn^*}_k \right]^{-1}, \tag{27a}$$

$$\delta\boldsymbol{P}_k = \boldsymbol{K}^{vsn}_k \boldsymbol{H}^{vsn^*}_s \boldsymbol{P}_s \boldsymbol{M}^T_*, \tag{27b}$$

$$\delta\hat{\boldsymbol{x}}_k = \boldsymbol{M}_* \boldsymbol{K}^{vsn}_k \left(\boldsymbol{z}^{vsn^*}_k - \boldsymbol{H}^{vsn^*}_s \hat{\boldsymbol{x}}_s \right). \tag{27c}$$

3.4 Flow of EKF-Larsen Processing

Fig. 3 shows an overview of the measurement processing procedures for the standard EKF and Larsen method. The image processing routines are started at time s, tracking feature points in new scenes; however, there is no available vision-based measurement until time $k = s + N$.

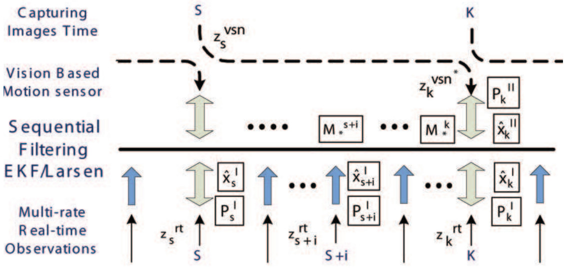


Figure 3: Flow of sequential EKF/Larsen processing.

Meanwhile, the multi-rate real-time measurements $\boldsymbol{z}^{rt}_{s+i}, 1 \leq i \leq N$ are fused through the EKF Eqs. (19b–21b) as they arrive, using $\boldsymbol{H}^{rt}_{s+i}$. This will produce the Kalman gain $\boldsymbol{K}^{rt}_{s+i}$, state estimates $\hat{\boldsymbol{x}}^I_{s+i}$

and covariance $\boldsymbol{P}^{I}_{s+i}$. Implementing Larsen approach requires only the state vector and covariance error at time s to be stored and the correction term $\boldsymbol{M}_{*}^{s+i}$ to be calculated during the delay period as

$$\boldsymbol{M}_{*}^{s+i} = \boldsymbol{M}_{*}^{s+i-1}\left(\boldsymbol{I} - \boldsymbol{K}^{rt}_{s+i}\boldsymbol{H}^{rt}_{s+i}\right)\boldsymbol{A}_{s+i-1}. \qquad (28)$$

At time k, when the vision-based measurements become available, Larsen equations are used to incorporate delayed measurement $\boldsymbol{z}_{k}^{vsn^{*}}$ in the estimation procedure. The Kalman gain $\boldsymbol{K}_{k}^{vsn}$ is calculated using Eq. (27a). Finally, visual measurements corrections $\delta\boldsymbol{P}_{k}$ and $\delta\hat{\boldsymbol{x}}_{k}$, obtained by Eqs. (27b–27c), are added to the covariance matrix and state vector of real-time measurements updates $\boldsymbol{P}_{k}^{I}$ and $\hat{\boldsymbol{x}}_{k}^{I}$, to obtain new quantities $\boldsymbol{P}_{k}^{II}$ and $\hat{\boldsymbol{x}}_{k}^{II}$.

4 SIMULATION EXPERIMENTS

A Matlab/Simulink simulator was developed, that includes a flight mechanics model of a small RUAV, models of inertial navigation sensors, magnetometer, GPS and their noise models. The simulator is used in conjunction with the OGRE graphics engine (Junker, 2006), for rendering a virtual environment scene and simulating the image acquisition process. All sensor measurements are simulated (see table 2) as the helicopter flies at an altitude of 2 m following a rectangular path at a constant speed of 2 m/sec within a small village, composed of houses and several other objects with realistic textures (see Fig. 4).

Table 2: Sensors and vibration noise levels.

Sensors	Noise Level
Gyro	50 deg/s
Accelerometer	0.5 m/s^2
Magnetometer	$1 * 10^{-4}$ Gauss
GPS	2 m

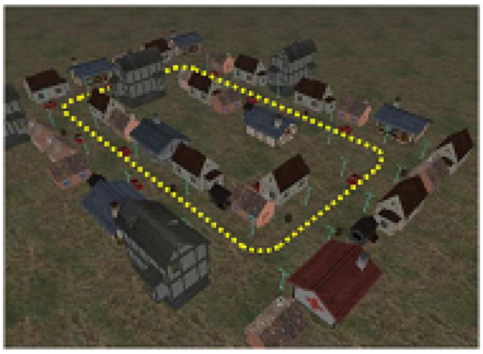


Figure 4: View of simulated small village environment and flight trajectory.

Navigation measurements are provided at a rate of 100 Hz, while stereo images at the rate of 2 Hz. The GPS, available at a rate of 1 Hz, is turned off after 5 sec in the flight, to clarify the affects of the visual measurement delay. The state estimates are obtained by four parallel data fusion processes: classic EKF with non-delayed measurements, classic EKF with delayed measurements, recalculation and EKF/Larsen methods in the presence of delay.

Fig. 5(a) shows the effects of delay on the EKF estimates, presenting a comparison of positions obtained by classic EKF, fed with delayed and non-delayed visual measurements. Fig. 5(b) shows position estimates obtained by the two methods of recalculation and sequential EKF/Larsen in the presence of delayed visual measurements, in comparison with the ideal classic EKF (without delay). Results clearly show the negative effects of delay on the standard EKF estimation, which are optimally compensated with the recalculation filter and the sequential filtering by EKF/Larsen.

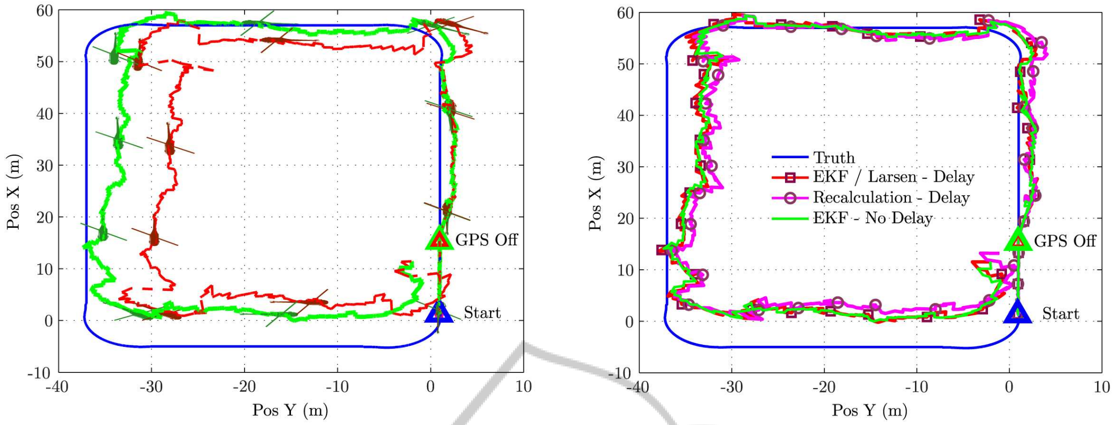
4.1 Monte Carlo Simulation

A Monte Carlo simulation is used here for considering the affects of random variation on the performance of the approaches, as well as evaluating the computational time burden of each method. The analysis consisted of 70 runs, which is the number of simulations that were necessary in this case to bring the statistics to convergence. For each simulation run, measurements and stereo images are generated for the 100 sec maneuver described above, each with randomly generated errors due to noise and random walks.

The average error in the position, velocity and attitude estimates are shown in Figs. (6–8), using the four implementations of the vision-augmented data fusion procedures explained above. The average estimation errors for each approach obtained by Monte Carlo Simulation are reported in table 3.

Considering random noise variations, the recalculation and EKF/Larsen methods show a good performance.

In fact, the average errors of these methods is very close to that obtained by the classic EKF with no delay on the visual measurements. However, the processing time of the filter recalculation increases twofold, as shown by table 3, implying a considerable additional computational burden. On the other hand, the EKF/Larsen approach does not significantly affect the processing time of the filter, and therefore conjugates high quality estimation and low computing time.



(a) Effects of delay on classic EKF estimate

(b) Performance of delayed fusion methods

Figure 5: Comparison of position estimates in the X-Y plane. (a) EKF with (dark-line) and without (light-line) delay on visual measurements; (b) Recalculation and EKF/Larsen methods in the presence of delayed visual measurements.

5 CONCLUSIONS

In this work, a previously documented VA-INS was extended by implementing various approaches to handle feature tracking delays in a multi-rate multi-sensor data fusion process. Simulation experiments were used together with Monte Carlo analyses to assess the estimation error and the computational burden of the methods.

The paper shows that delay caused by image processing, if not properly handled in the state estimator, can lead to an appreciable performance degradation. Furthermore, recalculation and sequential EKF/Larsen restore the estimate accuracy in the presence of delay, while Alexander method is not a suitable solution in this case because of tracking uncertainties. Finally, the results of the paper indicate that recalculation implies a significant computational burden, while Larsen method is as expensive as the standard EKF.

This study concluded that Larsen method, for the present application, provides estimates that have the same quality and computational cost of the non-delayed case.

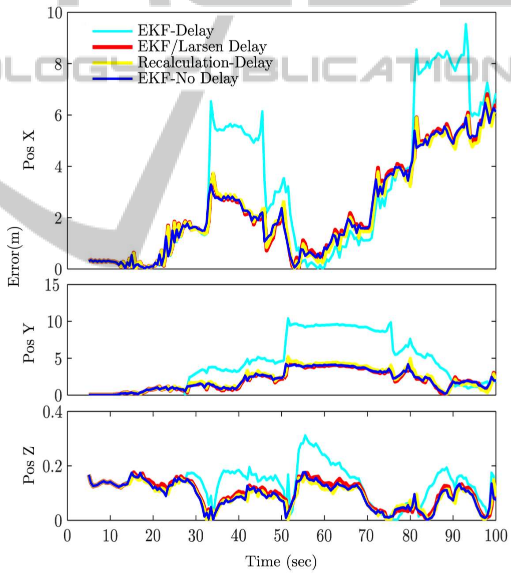


Figure 6: Position estimates error in X-Y-Z directions.

Table 3: Monte Carlo simulation results.

Method	Position RMSE	Velocity RMSE	Attitude RMSE	Filter Burden
	(m)	(m/s)	(deg)	(sec)
EKF No Delay	3.7355	0.8173	3.4061	0.0111
EKF Delay	6.8368	1.1069	3.9330	0.0111
Recalculation Delay	3.8156	0.8477	3.4397	0.0208
EKF/Larsen Delay	3.7690	0.8457	3.4336	0.0111

ACKNOWLEDGEMENTS

The authors would like to thank Mehmet Suat Kay for his help in the development of the feature tracking code.

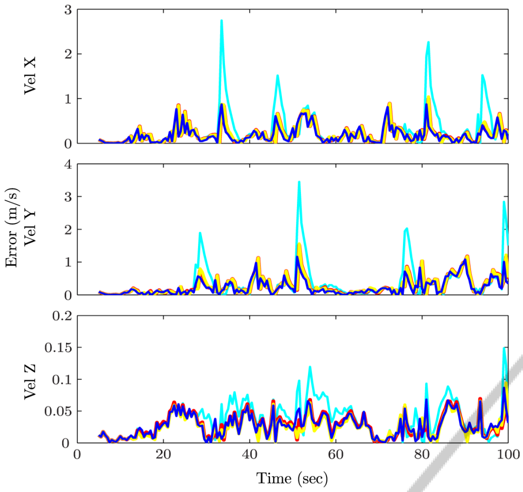


Figure 7: Velocity estimate errors in X-Y-Z directions.

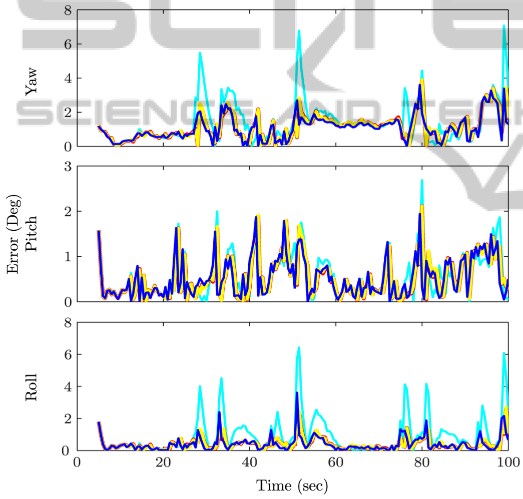


Figure 8: Attitude estimate errors in X-Y-Z directions.

REFERENCES

Alexander, H. L. (1991). State estimation for distributed systems with sensing delay. SPIE, 1470(1), 103-111. doi:10.1117/12.44843.

Bonin-Fontand, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3), 263-296.

Bottasso, C. L. and Leonello, D. (2009). Vision-augmented inertial navigation by sensor fusion for an autonomous rotorcraft vehicle. In *AHS International Specialists Meeting on Unmanned Rotorcraft*. 324-334.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *11th European Conference on Computer Vision*. LNCS Springer, 6314(3), 778-792.

Dalgleish, F. R., Tetlow, J. W., and Allwood, R. L. (2005). Vision-based navigation of unmanned underwater vehicles : a survey. part 2: Vision-basedstation-keeping and positioning. In *IMAREST Proceedings, Part B: Journal of Marine Design and Operations*. 8, 13-19.

Goedeme, T., Nuttin, M., Tuytelaars, T., and Gool, L. V. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3), 219-236.

Gopalakrishnan, A., Kaisare, N., and Narasimhan, S. (2011). Incorporating delayed and infrequent measurements in extended kalman filter based nonlinear state estimation. *Journal of Process Control*, 21(1), 119-129.

Jianbo, S. and Tomasi, C. (1994). Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 593-600.

Jones, E. S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4), 407-430.

Junker, G. (2006). *Pro OGRE 3D Programming*. Springer-Verlag, New York.

Larsen, T. D., Andersen, N. A., Ravn, O., and Poulsen, N. (1998). Incorporation of time delayed measurements in a discrete-time kalman filter. In *37th IEEE Conference on Decision and Control*. IEEE, 3972-3977.

Liu, Y. C. and Dai, Q. H. (2010). Vision aided unmanned aerial vehicle autonomy : An overview. In *3th International Congress on Image and Signal Processing*. IEEE, 417-421.

Mourikis, A. I. and Roumeliotis, S. I. (2007). A multi-state constraint kalman filter for vision-aided inertial navigation. In *IEEE International Conference on Robotics and Automation*. IEEE, 3565-3572.

Nister, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3-20.

Pornsarayouth, S. and Wongsaisuwan, M. (2009). Sensor fusion of delay and non-delay signal using kalman filter with moving covariance. In *ROBIO'09, IEEE International Conference on Robotics and Biomimetics*. 2045-2049.

Qian, G., Chellappa, R., and Zheng, Q. (2001). Robust structure from motion estimation using inertial data. *Journal of the Optical Society of America*, 18(12), 2982-2997.

Roumeliotis, S. I., Johnson, A. E., and Montgomery, J. F. (2002). Augmenting inertial navigation with image-based motion estimation. In *ICRA'02, IEEE International Conference on Robotics and Automation*. IEEE, 4326-4333.

Tatiraju, S., Soroush, S., and Ogunnaike, B. A. (1999). Multirate nonlinear state estimation with application to a polymerization reactor. *AIChE Journal*, 45(4), 769-780.

Veth, M. J., Raquet, J. F., and Pachter, M. (2006). Stochastic constraints for efficient image correspondence search. *Journal of IEEE Transactions on Aerospace Electronic Systems*, 42(3), 973-982.