



**HAL**  
open science

## Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures

Yoko Watanabe, Augustin Manecy, Antal Hiba, Sho Nagai, Shin Aoki

► **To cite this version:**

Yoko Watanabe, Augustin Manecy, Antal Hiba, Sho Nagai, Shin Aoki. Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures. AIAA Scitech 2019, Jan 2019, SAN DIEGO, United States. hal-02021142

**HAL Id: hal-02021142**

**<https://hal.science/hal-02021142>**

Submitted on 15 Feb 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures

Yoko Watanabe and Augustin Manecy\*  
*ONERA - The French Aerospace Laboratory, Toulouse, France*

Antal Hiba†  
*Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary*

Sho Nagai and Shin Aoki‡  
*RICOH Co. Ltd., Yokohama, Japan*

**This paper proposes a vision-integrated navigation system to guide an aircraft on the final glide path. It makes use of onboard vision systems which track runway features and estimate a 6D aircraft pose with respect to a runway to land. The proposed vision-integrated navigation system will allow an aircraft to continue the final approach procedure by maintaining the navigation precision in case of possible degradation or failure of ILS or GNSS/SBAS sensors. In order to handle a non-negligible delay of such vision-based measurements due to the image processing time, an error-state Kalman filter (ESKF) framework incorporating time-delayed measurements is established. The proposed delayed-measurement ESKF framework uses a fact that camera image acquisitions are triggered by a system and hence can be notified without delay. This enables the navigation filter to perform back-propagation of the estimated state forward in time to prepare for the future correction step at the time the measurement becomes available. The vision-integrated navigation system based on this framework was developed and its functionality is validated in simulations. Its estimation performance will be flight-evaluated with two different vision systems onboard a fixed-wing UAV experimental platform.**

## I. Introduction

According to different statistical survey reports on commercial aircraft accidents[1], nearly half of the fatal accidents happened during the final approach/landing flight phases. Therefore, enhancing airplane flight safety and autonomy level during such critical operation phases is an important key to the accident rate reduction, which is a common goal of the world civil aviation. In order to contribute towards this global goal of the aircraft accident rate reduction, a Europe-Japan collaborative research project called VISION\* (Validation of Integrated Safety-enhanced Intelligent flight cONtrol) has been launched in 2016. This 3-year VISION project has objectives of investigating, developing, and above all validating advanced aircraft Guidance, Navigation and Control (GN&C) solutions that can automatically detect and overcome some critical flight situations. VISION project tackles two different types of fault recovery scenarios focusing on the aircraft final approach phase; i) Flight control performance recovery from actuator or sensor failures (e.g. control jamming or authority deterioration, loss of airspeed information), and ii) Navigation and guidance performance recovery from sensor failures (e.g. degradation of GPS/SBAS or ILS) or flight path obstruction. For the first set of scenarios, different Fault Detection and Diagnosis/Fault Tolerant Control(FDD/FTC) approaches are proposed and flight-evaluated on a full-scale manned aircraft[2][3]. For the second set of scenarios, onboard vision-aided navigation and guidance approaches are developed and to be flight-validated on a fixed-wing UAV (Unmanned Aerial Vehicle) experimental platform. These new GN&C solutions will improve the robustness and self-adaptability of current aircraft flight systems, and hence will contribute to ease the pilot's task and stress in handling such anomalies. This paper presents the preliminary outcomes of the work on the vision-integrated navigation system in case of possible sensor failures, performed for the second scenario set in VISION project.

---

\*Research Engineer, Department of Information Processing and Systems

†Research Engineer, Computational Optical Sensing and Processing Laboratory

‡Engineer, Photonics Research& Development Center

\*[http://w3.onera.fr/h2020\\_vision](http://w3.onera.fr/h2020_vision)

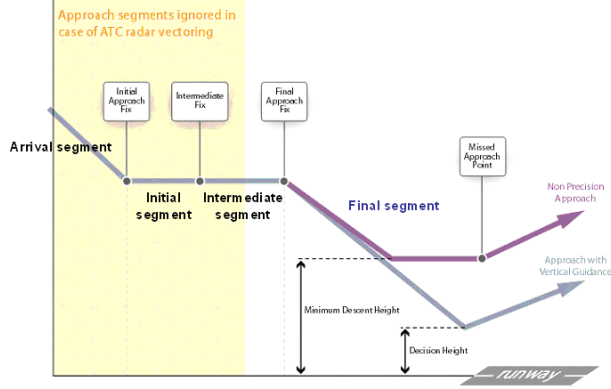
Recent large commercial airplanes are equipped with onboard cameras, which are used to augment pilot's situational awareness for obstacle clearance during taxiing. For example, A380 has fin-tip and belly cameras[4]. However, their usage is still very limited to cockpit display for pilot aide. The feasibility of using onboard vision as a new navigation sensor for making an autonomous airport approach has been studied in some past projects. In the EU-funded PEGASE project (2006-2009) coordinated by Dassault Aviation, the position- and image-based visual servoing algorithms for final approach guidance were proposed[5][6]. The French government-funded Visioland project (2013-2017) proposed an observer to estimate the aircraft position relative to the runway from image features[7]. The both projects benefitted from the participation of the aircraft manufacturers to evaluate the proposed algorithms in high-fidelity flight simulation framework with synthetic images. But no flight validation has been performed on a real aircraft. There are also many related work on vision-based autonomous landing of a fixed-wing UAVs by runway detection[8][9]. Most of them show validation results in simulation environment (often using FlightGear for image generation), and only few actually flight tested the navigation and guidance approaches in a closed-loop manner[10][11].

Many of the above listed work address monocular vision-only navigation approaches, and do not cope with the integration with other navigation sensors (INS, GPS or ILS). The onboard vision-aided inertial navigation has been one of the most intensively studied research topics in the UAV and robotics research community for decades. One can find in literature a wide variety of vision-based localization techniques by geo-referenced landmark detection[12], visual odometry and visual SLAM (Simultaneous Localization and Mapping) algorithms[13][14] applied to UAV navigation in GPS-denied urban or indoor environment. Unlike these UAV or robotics applications, the civil aviation application requires for the navigation system to provide an integrity monitoring function for possible sensor failure detection and exclusion. Our ultimate goal for this work of the VISION project is to provide a flight-validated vision-integration navigation system for the aircraft 6D pose estimation by multi-sensor fusion, augmented with an Aircraft Autonomous Integrity Monitoring (AAIM) function based on Multiple Solution Separation (MSS) method[15] or residual-based integrity monitoring[16][17].

As the preliminary outcomes from this work, this paper proposes the state estimator design by establishing an Error-State Kalman Filter (ESKF) framework[19][20] incorporating time-delayed measurements so that it can handle an inevitable non-negligible time-delay of the vision-based measurements due to the image processing time. The conventional way of handling the time-delayed measurements is to stock all the histories of the sensor measurements (including IMU) and the estimator outputs for a certain time horizon and re-runs the KF process from the time of the measurement to the current time as if the measurements arrived without delay. The proposed framework in this paper uses a fact that acquisitions of the camera images are triggered by a vision system and hence can be notified to the navigation system without delay. This fact enables us to perform the back-propagation of the estimation state forward in time. By doing so, we can avoid performing redundant KF correction processes with already arrived measurements. Moreover, it does not require to stock all the data histories unlike the conventional approach. The proposed approach has been implemented and its functionality was validated in simulations. But the work towards its flight validation with two different vision systems onboard, a stereo-vision system and a pair of monocular cameras with complementary specifications, is on its way. This paper also gives a brief description of the vision systems and the experimental UAV platform to be used for future flight validation.

## II. Sensor failure scenarios on final approach

The VISION project focuses on the final approach segment, from the Final Approach Fix (FAF) until reaching at a decision altitude/height (DA/H), in an instrument approach procedure of a civil airplane illustrated in Figure 1. The classical instrument approach procedure uses ILS (Instrument Landing System) which provides both lateral and vertical guidance on a stabilized continuous descent path (usually at a 3-degree glide slope). Recently, RNAV (Area Navigation) approach procedure, described by a series of waypoints, has become widely available by the use of GNSS (Global Navigation Satellite System)[18]. In particular, high performance approach guidance can be provided thanks to GNSS augmentation system such as SBAS (Satellite Based Augmentation System) which uses a differential GNSS technology and provides the integrity and correction information. The interest of using onboard vision in the both ILS and GNSS/SBAS approach procedures is to allow a pilot to continue the approach until DA instead of triggering a go-around or missed approach procedure in case of possible degradation or failure of such navigation sensors. In our project, the following sensor failures are considered to happen during the final approach phase, supposing that the accurate aircraft position relative to runway is available on the initial approach segment for initializing the visual tracker.



**Fig. 1 Instrument Approach Procedure[18]**

- In the GNSS/SBAS approach procedure:
  - Lack of SBAS signals with GPS signals fully available
  - Lack of SBAS and GNSS signals due to jamming
  - Degradation of GNSS signals due to a reduced number of tracked satellites with SBAS augmentation
  - Degradation of GNSS signals due to ionosphere interference with SBAS augmentation
- In the classical ILS approach procedure:
  - Lack of ILS signals
  - Misleading of ILS signals due to secondary lobe

These failures are emulated by using GNSS or ILS sensor error models developed in the project.

### III. Vision-integrated navigation system design

The idea of this work is to design the vision-integrated navigation system which uses information extracted from onboard camera images in order to maintain the SBAS-augmented GNSS localization or ILS precision in case of their possible degradation or failure so that the aircraft can still continue the precision approach procedure. Figure 2 overviews the onboard system architecture with the vision-integrated navigation system. It fuses the navigation sensor measurements (GNSS or ILS and vision) with IMU acceleration and rate gyro inputs and other sensor measurements (such as barometric pressure measurement) to estimate the aircraft 6D pose relative to a runway to land.

The estimator design proposed in this paper is based on the Error-State Kalman Filter (ESKF)[19][20]. In the ESKF framework, an estimation state vector is decomposed into a nominal state and an error state. The nominal state system does not include any uncertainty (model error nor noise) and hence the state is propagated deterministically. All the system uncertainties are included in the error state system, and a linear Kalman Filter (KF) is applied to estimate the error state. After each KF correction, the measurement-corrected estimated error state is injected to the nominal state and the estimated error state is reset to zero. This injection and reset operation makes the error state always small and operating close to the origin, and so the linearization validity of the error state system holds. It is claimed in [20] that the ESKF performance is more robust than Extended KF to different aircraft maneuvers (i.e. system nonlinearities). Figure 3 illustrates the ESKF process.

#### A. Aircraft kinematics model

We base on the formulation presented in [21] for the 6D pose estimation by fusion of IMU and other sensor measurements (such as GPS). See [21] for definition of the quaternion operators. Let  $\mathbf{x}_t$  be a **true state** to be estimated.

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{X}_t^T & \mathbf{V}_t^T & \mathbf{b}_{a_t}^T & P_{0_t} & \mathbf{q}_t^T & \mathbf{b}_{\omega_t}^T \end{bmatrix} \quad (1)$$

where  $\mathbf{X}_t$  and  $\mathbf{V}_t$  are position and velocity vectors in a locally-fixed runway frame,  $\mathbf{q}_t$  is a quaternion vector from the vehicle body frame to the runway frame,  $P_{0_t}$  be a pressure adjusted to a standard atmosphere at the sea level,  $\mathbf{b}_{a_t}$  and  $\mathbf{b}_{\omega_t}$  are acceleration and angular rate measurement bias in the vehicle body frame respectively. The runway frame is defined as a frame fixed to the runway with its origin at the threshold point, its X-axis aligns with the runway landing

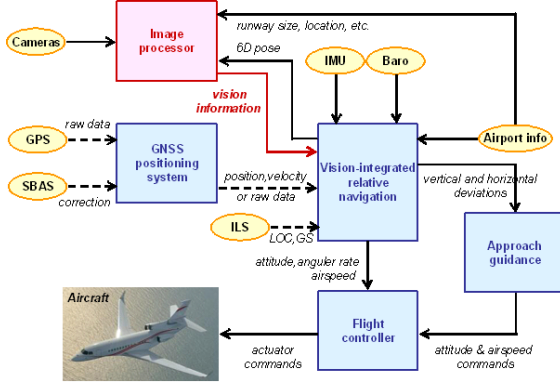


Fig. 2 Vision-integrated Navigation System

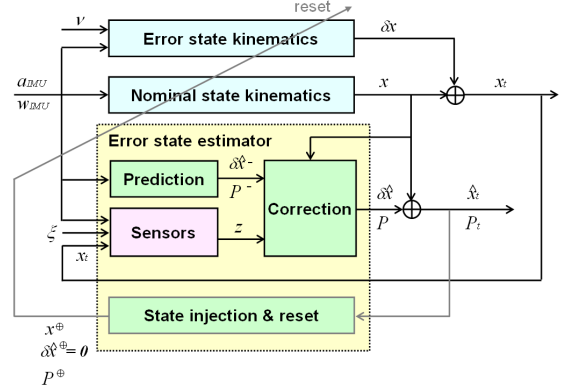


Fig. 3 Error State Kalman Filter Process

direction and Z-axis vertically downwards. Then its dynamics can be written by

$$\dot{\mathbf{x}}_t = \begin{bmatrix} \dot{X}_t \\ \dot{V}_t \\ \dot{\mathbf{b}}_{a_t} \\ \dot{P}_{0_t} \\ \dot{\mathbf{q}}_t \\ \dot{\mathbf{b}}_{\omega_t} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_t \\ \mathbf{a}_t \\ \mathbf{v}_{b_a} \\ \mathbf{v}_{P_0} \\ \frac{1}{2} \mathbf{q}_t \otimes \boldsymbol{\omega}_t \\ \mathbf{v}_{b_\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_t \\ R(\mathbf{q}_t)(\mathbf{a}_{IMU} - \mathbf{b}_{a_t} - \mathbf{v}_a) + \mathbf{g} \\ \mathbf{v}_{b_a} \\ \mathbf{v}_{b_b} \\ \frac{1}{2} \mathbf{q}_t \otimes (\boldsymbol{\omega}_{IMU} - \mathbf{b}_{\omega_t} - \mathbf{v}_\omega) \\ \mathbf{v}_{b_\omega} \end{bmatrix} \quad (2)$$

where  $\mathbf{a}_t$  and  $\boldsymbol{\omega}_t$  are the true acceleration and angular rate,  $\mathbf{a}_{IMU}$  and  $\boldsymbol{\omega}_{IMU}$  are the IMU measurements,  $\mathbf{g} = [0 \ 0 \ g]^T$  is a gravity vector in the runway frame, and  $\mathbf{v}_*$  represents a zero-mean Gaussian noise.

Let  $\mathbf{x}$  be a **nominal state** vector which is defined as follows.

$$\mathbf{x} = \begin{bmatrix} X^T & V^T & \mathbf{b}_a^T & P_0 & \mathbf{q}^T & \mathbf{b}_\omega^T \end{bmatrix} \quad (3)$$

It evolves with the following deterministic dynamics.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{X} \\ \dot{V} \\ \dot{\mathbf{b}}_a \\ \dot{P}_0 \\ \dot{\mathbf{q}} \\ \dot{\mathbf{b}}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{V} \\ R(\mathbf{q})(\mathbf{a}_{IMU} - \mathbf{b}_a) + \mathbf{g} \\ \mathbf{0} \\ 0 \\ \frac{1}{2} \mathbf{q} \otimes (\boldsymbol{\omega}_{IMU} - \mathbf{b}_\omega) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{V} \\ R(\mathbf{q})\mathbf{a} + \mathbf{g} \\ \mathbf{0} \\ 0 \\ \frac{1}{2} \mathbf{q} \otimes \boldsymbol{\omega} \\ \mathbf{0} \end{bmatrix} \quad (4)$$

where defined  $\mathbf{a} = \mathbf{a}_{IMU} - \mathbf{b}_a$  and  $\boldsymbol{\omega} = \boldsymbol{\omega}_{IMU} - \mathbf{b}_\omega$ . The true state  $\mathbf{x}_t$  can be decomposed by the nominal state  $\mathbf{x}$  and the **error state**  $\delta\mathbf{x}$ .

$$\mathbf{x}_t = \begin{bmatrix} X_t \\ V_t \\ \mathbf{b}_{a_t} \\ P_{0_t} \\ \mathbf{q}_t \\ \mathbf{b}_{\omega_t} \end{bmatrix} = \begin{bmatrix} X + R(\mathbf{q})\delta X \\ V + R(\mathbf{q})\delta V \\ \mathbf{b}_a + \delta\mathbf{b}_a \\ P_0 + \delta P_0 \\ \mathbf{q} \otimes \mathbf{q}(\delta\boldsymbol{\theta}) \\ \mathbf{b}_\omega + \delta\mathbf{b}_\omega \end{bmatrix} = \mathbf{x} \oplus \delta\mathbf{x}, \quad \delta\mathbf{x} = \begin{bmatrix} \delta X \\ \delta V \\ \delta\mathbf{b}_a \\ \delta P_0 \\ \delta\boldsymbol{\theta} \\ \delta\mathbf{b}_\omega \end{bmatrix} \quad (5)$$

where  $\delta\boldsymbol{\theta}$  is an error angle vector and  $\mathbf{q}(\delta\boldsymbol{\theta}) = \left[ \cos \frac{\|\delta\boldsymbol{\theta}\|}{2} \quad \sin \frac{\|\delta\boldsymbol{\theta}\|}{2} \frac{\delta\boldsymbol{\theta}^T}{\|\delta\boldsymbol{\theta}\|} \right]^T$ . Then the error state kinematics becomes

linear as below.

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} -\Omega & I & O & \mathbf{0} & O & O \\ O & -\Omega & -I & \mathbf{0} & -\mathcal{A} & O \\ O & O & O & \mathbf{0} & O & O \\ O & O & O & 0 & O & O \\ O & O & O & \mathbf{0} & -\Omega & -I \\ O & O & O & \mathbf{0} & O & O \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{v}_a \\ \mathbf{v}_{b_a} \\ \mathbf{v}_{P_0} \\ -\mathbf{v}_\omega \\ \mathbf{v}_{b_\omega} \end{bmatrix} = F(\mathbf{x}, \mathbf{a}_{IMU}, \boldsymbol{\omega}_{IMU})\delta\mathbf{x} + \mathbf{v} \quad (6)$$

where  $\mathcal{A} = [\mathbf{a}\times]$  and  $\Omega = [\boldsymbol{\omega}\times]$  are the cross-product skew-symmetric matrices.

The nominal state and error state kinematics (4) and (6) are discretized as

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{X}_j \\ \mathbf{V}_j \\ \mathbf{b}_{a_j} \\ P_{0_j} \\ \mathbf{q}_j \\ \mathbf{b}_{\omega_j} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{j-1} + \mathbf{V}_{j-1}\Delta t_j + \frac{1}{2}(R(\mathbf{q}_{j-1})\mathbf{a}_j + \mathbf{g})\Delta t_j^2 \\ \mathbf{V}_{j-1} + (R(\mathbf{q}_{j-1})\mathbf{a}_j + \mathbf{g})\Delta t_j \\ \mathbf{b}_{a_{j-1}} \\ P_{0_{j-1}} \\ \mathbf{q}_{j-1} \otimes \mathbf{q}(\boldsymbol{\omega}_j\Delta t_j) \\ \mathbf{b}_{\omega_{j-1}} \end{bmatrix} \quad (7)$$

$$\delta\mathbf{x}_j = e^{F(\mathbf{x}_j, \mathbf{a}_{IMU_j}, \boldsymbol{\omega}_{IMU_j})\Delta t_j} \delta\mathbf{x}_{j-1} + \mathbf{v}_j = e^{F_j\Delta t_j} \delta\mathbf{x}_{j-1} + \mathbf{v}_j = \Phi_j \delta\mathbf{x}_{j-1} + \mathbf{v}_j \quad (8)$$

where  $\mathbf{v}_j \sim \mathcal{N}(\mathbf{0}, Q_j \approx Q_{\mathbf{v}\Delta t_j})$  is a discretized Gaussian noise.

## B. Sensor measurement models

The main sensors used for the aircraft relative state estimation are GNSS or ILS, barometer and vision sensors (but we can add other sensors such as magnetometer or GNSS-headings, inclinometers, radio-altimeter, etc.). Their measurements can be modeled as a function of the true state  $\mathbf{x}_t$ .

### 1. ILS (Instrument Landing System)

ILS is composed of the lateral localizer (LOC) and the vertical glide slope (GS), which provides final approach guidance information, that is, the deviations of the aircraft flight path from the desired glide course (normally  $\gamma_{GS} = 3^\circ$  glide path). The deviations from the desired glide path can be measured by taking a difference in amplitude of the two lobes of signals modulated at 90 Hz and 150 Hz, called DDM (Difference in depth of modulation).

$$\begin{aligned} z_{ILS} &= \begin{bmatrix} DDM_{LOC} \\ DDM_{GS} \end{bmatrix} = \begin{bmatrix} K_{LOC}\Delta\theta_{LOC} \\ K_{GS}\Delta\theta_{GS} \end{bmatrix} + \boldsymbol{\xi}_{ILS} = \begin{bmatrix} K_{LOC} \arctan \frac{Y_t}{\frac{TCH}{\tan\gamma_{GS}} - X_t} \\ K_{GS} \left( \arctan \frac{-Z_t}{\frac{TCH}{\tan\gamma_{GS}} - X_t} - \gamma_{GS} \right) \end{bmatrix} + \boldsymbol{\xi}_{ILS} \\ &= \tilde{\mathbf{h}}_{ILS}(\mathbf{X}_t) + \boldsymbol{\xi}_{ILS} \end{aligned}$$

where  $TCH$  is a height at the threshold point.  $K_{LOC}$  and  $K_{GS}$  are known angular displacement sensitivity.

### 2. GNSS (Global Navigation Satellite System)

Let  $\Delta\mathbf{X}_{GNSS}$  denote a known GNSS receiver position in the vehicle body frame. Then, the true GNSS receiver position and velocity are

$$\begin{bmatrix} \mathbf{X}_{GNSS_t} \\ \mathbf{V}_{GNSS_t} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_t + R(\mathbf{q}_t)\Delta\mathbf{X}_{GNSS} \\ \mathbf{V}_t + R(\mathbf{q}_t)[\boldsymbol{\omega}_t\times]\Delta\mathbf{X}_{GNSS} \end{bmatrix} = \tilde{\mathbf{h}}_{GNSS}(\mathbf{x}_t, \boldsymbol{\omega}_t)$$

For the GNSS loose-coupling, the position and velocity measurements are used and hence

$$z_{GNSS_k} = \tilde{\mathbf{h}}_{GNSS}(\mathbf{x}_{t_k}, \boldsymbol{\omega}_{t_k}) + \boldsymbol{\xi}_{GNSS_k}$$

For the GNSS tight-coupling, a set of pseudo-ranges to visible satellites is used as a measurement. The pseudo-range to the  $i$ -th satellite is modeled as

$$\rho_i = \|\mathbf{X}_{GNSS_i}^{ECEF} - \mathbf{X}_{SAT_i}^{ECEF}\| + c(\tau_t - \tau_i) + \xi_{\rho_i} = d_{i_t}(\mathbf{x}_t) + c(\tau_t - \hat{\tau}_i) + \xi_{\rho_i}$$

$\mathbf{X}_{SAT_i}^{ECEF}$  is a known position of the  $i$ -th satellite in the ECEF (Earth-Centered Earth-Fixed) frame,  $c$  is a light speed,  $\tau_t$  and  $\tau_i$  are the clock bias of the receiver and the  $i$ -th satellite respectively.  $\xi_{\rho_i} \sim \mathcal{N}(0, \sigma_{UERE}^2)$  is a zero-mean Gaussian noise which includes errors from different sources such as the satellite clock and ephemeris errors, compensation errors in ionospheric and tropospheric signal delays, multi-path effects, etc.  $\sigma_{UERE}$  (UERE stands for User Equivalent Range Error) is calculated and provided by the receiver. The receiver clock-bias  $\tau_t$  can be modeled as a 2nd order random process such as

$$\dot{\mathbf{x}}_{\tau_t} = \begin{bmatrix} \dot{\tau}_t \\ \dot{v}_{\tau_t} \end{bmatrix} = \begin{bmatrix} v_{\tau_t} \\ 0 \end{bmatrix} + \begin{bmatrix} v_{\tau} \\ v_{v_{\tau}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}_{\tau_t} + \mathbf{v}_{\tau} = F_{\tau} \mathbf{x}_{\tau_t} + \mathbf{v}_{\tau}$$

This state can be decomposed of the nominal and error states,  $\mathbf{x}_{\tau_t} = \mathbf{x}_{\tau} + \delta \mathbf{x}_{\tau}$ , and has the following kinematics. The estimation state should be augmented with these for the tight-coupling case.

$$\dot{\mathbf{x}}_{\tau} = F_{\tau} \mathbf{x}_{\tau}, \quad \delta \dot{\mathbf{x}}_{\tau} = F_{\tau} \delta \mathbf{x}_{\tau} + \mathbf{v}_{\tau} \quad (9)$$

Then the measurement model becomes

$$\mathbf{z}_{\rho} = \begin{bmatrix} \rho_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \|\mathbf{X}_{GNSS_i}^{ECEF} - \mathbf{X}_{SAT_i}^{ECEF}\| + c(\tau_t - \tau_i) \\ \vdots \end{bmatrix} + \begin{bmatrix} \xi_{\rho_i} \\ \vdots \end{bmatrix} = \tilde{\mathbf{h}}_{\rho}(\mathbf{x}_{t_k}) + \xi_{\rho_k}$$

### 3. Barometer

A barometer gives an atmospheric pressure measurement which is a function of the sensor altitude.

$$z_{BARO} = P_0 \exp\left(\frac{gM}{R_0 T_0}(Z_t - Z_{THD})\right) + \xi_{BARO} = \tilde{\mathbf{h}}_{BARO}(\mathbf{x}_t) + \xi_{BARO}$$

where  $Z_{THD}$  is the runway threshold elevation.  $M$ ,  $R_0$  and  $T_0$  are the known constants of the standard atmosphere.

### 4. Vision sensor

The vision measurement can be 2D such as pixel-coordinates of a point-of-interest or 3D such as relative position with respect to a point-of-interest in the camera frame. Let  $\Delta \mathbf{X}_C$  and  $\Delta \mathbf{q}_C$  be a known camera position and orientation in the vehicle body frame. Then the true camera pose in the runway frame is given by

$$\begin{bmatrix} \mathbf{X}_{C_t} \\ \mathbf{q}_{C_t} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_t + R(\mathbf{q}_t) \Delta \mathbf{X}_C \\ \mathbf{q}_t \otimes \Delta \mathbf{q}_C \end{bmatrix} = \tilde{\mathbf{h}}_{VISION}(\mathbf{x}_t)$$

A point at  $\mathbf{X}_p$  in the runway frame has a position expressed in the camera frame at  $\mathbf{X}_{p_t}^C = R(\mathbf{q}_{C_t}^*)(\mathbf{X}_p - \mathbf{X}_{C_t})$ . Then the pixel coordinates  $\mathbf{p}$  of this point is defined by

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \frac{\mathbf{X}_{p_t}^C}{Z_{p_t}^C} = C \frac{\mathbf{X}_{p_t}^C}{Z_{p_t}^C} \quad (10)$$

where  $C$  is a known camera matrix with focal length  $f_{(x,y)}$  and image center pixels  $c_{(x,y)}$ .  $Z_{p_t}^C$  is an image depth, a distance to the point-of-interest along the camera optical axis. A camera disparity for a stereo vision system with baseline  $B$  is measured as  $d_{p_t} = B f_x / Z_{p_t}^C$ .

In this work, the image processor is developed to detect the runway features shown in Figure 4 to calculate the 6D pose of the camera in the runway frame (see Section V.B).  $\mathbf{p}_{THD}$ ,  $\mathbf{p}_{L/R}$  and  $\mathbf{p}_V$  are the pixel-coordinates of the threshold point, the runway left/right corners and the vanishing point of the two parallel sidelines respectively. The

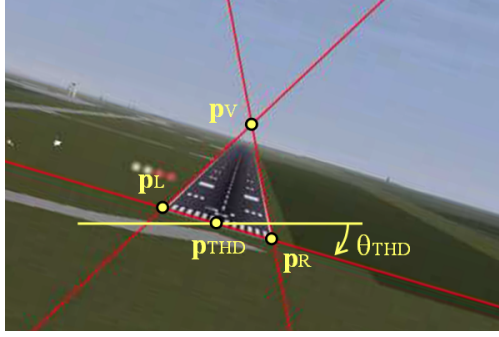


Fig. 4 Runway feature points to be detected on an image

monocular vision system could use an apriori knowledge on the runway width to resolve the scale ambiguity. For the vision loose-coupling, we use directly the image-based 6D pose estimation result as a measurement.

$$z_{VISION} = \tilde{h}_{VISION}(x_t) + \xi_{VISION}$$

### C. Error-State Kalman Filter process

As stated earlier, a linear Kalman filter is applied to the error state estimation from the available sensor measurements. Let  $\delta\hat{x}_j^-$  and  $P_j^-$  be the predicted estimate and its error covariance. (8) gives a process model, and so the prediction step in Figure 3 is performed at each IMU measurements update as follows.

$$\delta\hat{x}_j^- = \Phi_j \delta\hat{x}_{j-1} \quad (11)$$

$$P_j^- = \mathbb{E} [\delta\hat{x}_j^- \delta\hat{x}_j^{-T}] = \Phi_j P_{j-1} \Phi_j^T + Q_j \quad (12)$$

It should be noted that, in the ESKF framework,  $\delta\hat{x}_j^-$  is always zero as  $\delta\hat{x}_{j-1}$  is reset to zero after each correction step. Hence, only (12) needs to be implemented.

Suppose that a sensor measurement  $z_k$  is available at  $t = t_k$  without any delay.

$$z_k = \tilde{h}(x_t(t_k), a_t(t_k), \omega_t(t_k)) + \xi_k \quad (13)$$

where  $\xi_k \sim \mathcal{N}(\mathbf{0}, R_k)$  is the zero-mean Gaussian measurement noise. Then (13) can be expanded as follows.

$$z_k \approx h(x_k, a_{IMU_k}, \omega_{IMU_k}) + H(x_k, a_{IMU_k}, \omega_{IMU_k}) \delta x_k + \xi_k + D(x_k, a_{IMU_k}, \omega_{IMU_k}) v(t_k)$$

where

$$\begin{aligned} D(x_t, a_{IMU}, \omega_{IMU}) &= \left[ \frac{\partial \tilde{h}(x_t, a_t, \omega_t)}{\partial a_t} R(q_t) \quad O \quad \mathbf{0} \quad \frac{\partial \tilde{h}(x_t, a_t, \omega_t)}{\partial \omega_t} \quad O \right] \Bigg|_{\substack{a_t = R(q_t)(a_{IMU} - b_{a_t}) + g \\ \omega_t = \omega_{IMU} - b_{\omega_t}}} \\ H(x_k, a_{IMU_k}, \omega_{IMU_k}) &= \frac{\partial h(x_t, a_{IMU_k}, \omega_{IMU_k})}{\partial x_t} \Bigg|_{x_t=x_k} \frac{\partial (x_k \oplus \delta x)}{\partial \delta x} \\ &= \frac{\partial h(x_t, a_{IMU_k}, \omega_{IMU_k})}{\partial x_t} \Bigg|_{x_t=x_k} \begin{bmatrix} R(q_k) & O & O & \mathbf{0} & O & O \\ O & R(q_k) & O & \mathbf{0} & O & O \\ O & O & I & \mathbf{0} & O & O \\ O & O & O & 1 & O & O \\ O & O & O & \mathbf{0} & \frac{1}{2} \begin{bmatrix} -q_{v_k}^T \\ q_{w_k} I + [q_{v_k} \times] \end{bmatrix} & O \\ O & O & O & \mathbf{0} & O & I \end{bmatrix} \end{bmatrix} \end{aligned}$$



with  $\mathbf{q}_k = \begin{bmatrix} q_{w_k} & \mathbf{q}_{v_k}^T \end{bmatrix}^T$ . Then, the measurement  $\mathbf{z}_k$  can be transformed to a linear measurement to the error state.

$$\bar{\mathbf{z}}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k}) = H_k \delta \mathbf{x}_k + \boldsymbol{\xi}_k + D_k \mathbf{v}(t_k) = H_k \delta \mathbf{x}_k + \bar{\boldsymbol{\xi}}_k \quad (14)$$

where  $H_k = H(\mathbf{x}_k, \mathbf{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})$  and  $D_k = D(\mathbf{x}_k, \mathbf{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})$ . The augmented measurement error follows  $\bar{\boldsymbol{\xi}}_k \sim \mathcal{N}(\mathbf{0}, \bar{R}_k = R_k + D_k Q_v D_k^T)$ . Now a standard linear Kalman filter is applied to update the predicted error state  $\delta \hat{\mathbf{x}}_k^-$  and its error covariance  $P_k^-$  using this measurement  $\bar{\mathbf{z}}_k$ . Since  $\delta \hat{\mathbf{x}}_k^- = \mathbf{0}$ ,

$$\delta \hat{\mathbf{x}}_k = \delta \hat{\mathbf{x}}_k^- + K_k (\bar{\mathbf{z}}_k - H_k \delta \hat{\mathbf{x}}_k^-) = K_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})) \quad (15)$$

$$P_k = (I - K_k H_k) P_k^-, \quad K_k = P_k^- H_k^T (H_k P_k^- H_k^T + \bar{R}_k)^{-1} \quad (16)$$

In the ESKF process, after this KF measurement correction step, the error state injection and reset process is performed. The corrected non-null error state  $\delta \hat{\mathbf{x}}_k$  is injected to the nominal state by (5) and the error state is reset to zero, i.e.,  $\mathbf{x}_k^\oplus = \mathbf{x}_k \oplus \delta \hat{\mathbf{x}}_k$  and  $\delta \hat{\mathbf{x}}_k^\oplus = \mathbf{0}$ . As the true state does not change by the injection, the true error state will also change and hence

$$\begin{aligned} \mathbf{x}_{t_k} &= \mathbf{x}_k \oplus \delta \mathbf{x}_k = \mathbf{x}_k \oplus (\delta \hat{\mathbf{x}}_k + \delta \tilde{\mathbf{x}}_k) \\ &= \mathbf{x}_k^\oplus \oplus \delta \mathbf{x}_k^\oplus = (\mathbf{x}_k \oplus \delta \hat{\mathbf{x}}_k) \oplus \delta \tilde{\mathbf{x}}_k^\oplus \end{aligned}$$

From the definition (5),

$$\delta \tilde{\mathbf{x}}_k^\oplus = \begin{bmatrix} I - [\delta \hat{\boldsymbol{\theta}}_k \times] & O & O & \mathbf{0} & O & O \\ O & I - [\delta \hat{\boldsymbol{\theta}}_k \times] & O & \mathbf{0} & O & O \\ O & O & I & \mathbf{0} & O & O \\ O & O & O & 1 & O & O \\ O & O & O & \mathbf{0} & I - \frac{1}{2} [\delta \hat{\boldsymbol{\theta}}_k \times] & O \\ O & O & O & \mathbf{0} & O & I \end{bmatrix} \delta \tilde{\mathbf{x}}_k = G^\oplus(\delta \hat{\mathbf{x}}_k) \delta \tilde{\mathbf{x}}_k \quad (17)$$

Therefore, the error covariance  $P_k$  should be also reset to

$$P_k^\oplus = \mathbb{E} [\delta \tilde{\mathbf{x}}_k^\oplus \delta \tilde{\mathbf{x}}_k^{\oplus T}] = \mathbb{E} [\delta \mathbf{x}_k^\oplus \delta \mathbf{x}_k^{\oplus T}] = G^\oplus(\delta \hat{\mathbf{x}}_k) P_k G^{\oplus T}(\delta \hat{\mathbf{x}}_k) \quad (18)$$

The ESKF procedure is reset with  $\mathbf{x}_k^\oplus$ ,  $\delta \hat{\mathbf{x}}_k^\oplus = \mathbf{0}$  and  $P_k^\oplus$  at  $t_k$  and start the new propagation-correction-reset cycle until the next measurement will be obtained.

## D. Error-State Kalman Filter process with time-delayed measurements

The vision-based measurement normally arrives with a non-negligible time delay due to the image processing time, and this time-delay should be taken into account in the estimator design. In general, acquisition of an image (or a pair of images for stereo vision) is triggered by a system and hence the time of the image capture is known and can be notified before the measurement arrives. By using this fact, this paper establishes a framework of the ESKF filter with time-delayed measurements.

Figure 5 shows the ESKF process timeline with time-delayed measurement. In more general way, let  $t_{k_i}$  be the time when the  $i$ -th measurement  $\mathbf{z}_{k_i}$  becomes available. The time interval  $k_{i-1} < j \leq k_i$  between the two successive measurements corresponds to the  $(i-1)$ -th cycle of ESKF propagation-correction-reset. During this cycle, the nominal state propagation (7) and the error state prediction step (11, 12) are repeated by using the IMU measurements. Then, the estimated error state will be corrected at  $t_{k_i}$  by using the  $i$ -th measurement, and it will be injected to the nominal state. Then the ESKF is reset for the next cycle. Let us denote the nominal state and error state of the  $(i-1)$ -th cycle at the time  $t_j$  by  $\mathbf{x}_j^{i-1}$  and  $\delta \mathbf{x}_j^{i-1}$  respectively. The state injection and ESKF reset process gives following relation between the  $(i-1)$ -th and  $i$ -th states, where  $\delta \hat{\mathbf{x}}_{k_i}^{i-1}$  is the  $(i-1)$ -th error state updated at  $t_{k_i}$  with the  $i$ -th measurement.

$$\begin{aligned} \mathbf{x}_{k_i}^i &= \mathbf{x}_{k_i}^\oplus = \mathbf{x}_{k_i}^{i-1} \oplus \delta \hat{\mathbf{x}}_{k_i}^{i-1} \\ \delta \hat{\mathbf{x}}_{k_i}^i &= \delta \hat{\mathbf{x}}_{k_i}^\oplus = \mathbf{0} \\ \delta \mathbf{x}_{k_i}^i &= \delta \tilde{\mathbf{x}}_{k_i}^i = \delta \tilde{\mathbf{x}}_{k_i}^\oplus = G^\oplus(\delta \hat{\mathbf{x}}_{k_i}^{i-1}) \delta \tilde{\mathbf{x}}_{k_i}^{i-1} = G_{k_i}^\oplus \delta \tilde{\mathbf{x}}_{k_i}^{i-1} \\ P_{k_i}^i &= P_{k_i}^\oplus = G_{k_i}^\oplus P_{k_i}^{i-1} G_{k_i}^{\oplus T} \end{aligned}$$

### 1. Nominal state propagation and Error state prediction

Starting from  $\mathbf{x}_{k_{i-1}}^{i-1}$ , the  $(i-1)$ -th nominal state  $\mathbf{x}_j^{i-1}$  is propagated at each time step  $t_j$  up to  $t_{k_i}$  by using (7). Note that  $\mathbf{a}_j^{i-1} = \mathbf{a}_{IMU_j} - \mathbf{b}_{\mathbf{a}_{k_{i-1}}}^{i-1}$  and  $\boldsymbol{\omega}_j^{i-1} = \boldsymbol{\omega}_{IMU_j} - \mathbf{b}_{\boldsymbol{\omega}_{k_{i-1}}}^{i-1}$ . Likewise, the estimated error state will be propagated up to  $t_{k_i}$ .

$$\delta \hat{\mathbf{x}}_{k_i}^{i-1-} = \left[ \prod_{j=k_{i-1}+1}^{k_i} \Phi_j^{i-1} \right] \delta \hat{\mathbf{x}}_{k_{i-1}}^{i-1} = \Phi_{k_i k_{i-1}}^{i-1} \delta \hat{\mathbf{x}}_{k_{i-1}}^{i-1} = \mathbf{0} \quad (19)$$

$$\delta \mathbf{x}_{k_i}^{i-1} = \delta \tilde{\mathbf{x}}_{k_i}^{i-1-} = \Phi_{k_i k_{i-1}}^{i-1} \delta \mathbf{x}_{k_{i-1}}^{i-1} + \sum_{j=k_{i-1}+1}^{k_i} \Phi_{k_i j}^{i-1} \mathbf{v}_j = \Phi_{k_i k_{i-1}}^{i-1} \delta \mathbf{x}_{k_{i-1}}^{i-1} + \bar{\mathbf{v}}_{k_i-1} \quad (20)$$

$$\mathbf{P}_{k_i}^{i-1-} = \Phi_{k_i k_{i-1}}^{i-1} \mathbf{P}_{k_{i-1}}^{i-1} \Phi_{k_i k_{i-1}}^{i-1T} + \sum_{j=k_{i-1}+1}^{k_i} \Phi_{k_i j}^{i-1} \mathbf{Q}_j \Phi_{k_i j}^{i-1T} = \Phi_{k_i k_{i-1}}^{i-1} \mathbf{P}_{k_{i-1}}^{i-1} \Phi_{k_i k_{i-1}}^{i-1T} + \bar{\mathbf{Q}}_{k_i-1} \quad (21)$$

### 2. Error state correction with delayed measurement

At time  $t_{k_i}$ , the  $i$ -th measurement  $\mathbf{z}_{k_i}$  becomes available. It may have a measurement delay and includes an information on the true state at  $t_{m_i} \leq t_{k_i}$  (See Figure 5). Suppose that the time  $t_{m_i}$  is in the  $l_i$ -th cycle of the ESKF estimator, i.e.,  $k_{l_i} \leq m_i < k_{l_i+1}$ . Similarly to (14), the  $i$ -th measurement is given by

$$\begin{aligned} \mathbf{z}_{k_i} &= \mathbf{h}_i(\mathbf{x}_t(t_{m_i}), \mathbf{a}_{IMU_{m_i}}, \boldsymbol{\omega}_{IMU_{m_i}}) + \bar{\boldsymbol{\xi}}_{k_i} \\ &\simeq \mathbf{h}_i(\mathbf{x}_{m_i}^{l_i}, \mathbf{a}_{IMU_{m_i}}, \boldsymbol{\omega}_{IMU_{m_i}}) + H_i(\mathbf{x}_{m_i}^{l_i}, \mathbf{a}_{IMU_{m_i}}, \boldsymbol{\omega}_{IMU_{m_i}}) \delta \mathbf{x}_{m_i}^{l_i} + \bar{\boldsymbol{\xi}}_{k_i} \end{aligned}$$

where  $\mathbf{x}_{m_i}^{l_i}$  and  $\delta \mathbf{x}_{m_i}^{l_i}$  are the  $l_i$ -th nominal and error states at time  $t_{m_i}$ . Then, the measurement residual to be used in the ESKF becomes

$$\begin{aligned} \Delta \mathbf{z}_{k_i} &= \mathbf{z}_{k_i} - \mathbf{h}_i(\mathbf{x}_{m_i}^{l_i}, \mathbf{a}_{IMU_{m_i}}, \boldsymbol{\omega}_{IMU_{m_i}}) - H_i(\mathbf{x}_{m_i}^{l_i}, \mathbf{a}_{IMU_{m_i}}, \boldsymbol{\omega}_{IMU_{m_i}}) \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} \\ &= H_i \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} + \bar{\boldsymbol{\xi}}_{k_i} = H_i \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} + \boldsymbol{\xi}_{k_i} + D_i \mathbf{v}_{m_i} \end{aligned} \quad (22)$$

where  $\delta \tilde{\mathbf{x}}_{m_i}^{l_i+}$  is the back-propagated  $l_i$ -th error state at time  $t_{m_i}$ , calculated from the latest estimate  $\delta \hat{\mathbf{x}}_{k_i}^{i-1-} (= \mathbf{0})$ .  $\delta \tilde{\mathbf{x}}_{m_i}^{l_i+}$  is its error. The Kalman filter correction is applied with (22) in order to update the current  $(i-1)$ -th error state estimate.

$$\delta \hat{\mathbf{x}}_{k_i}^{i-1} = \delta \hat{\mathbf{x}}_{k_i}^{i-1-} + K_i \Delta \mathbf{z}_{k_i} = \delta \hat{\mathbf{x}}_{k_i}^{i-1-} + K_i (H_i \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} + \bar{\boldsymbol{\xi}}_{k_i}) \quad (23)$$

$$\delta \tilde{\mathbf{x}}_{k_i}^{i-1} = \delta \tilde{\mathbf{x}}_{k_i}^{i-1-} - K_i (H_i \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} + \bar{\boldsymbol{\xi}}_{k_i}) \quad (24)$$

The error covariance can be obtained as follows.

$$\mathbf{P}_{k_i}^{i-1} = \mathbb{E} \left[ \delta \tilde{\mathbf{x}}_{k_i}^{i-1} \delta \tilde{\mathbf{x}}_{k_i}^{i-1T} \right] = \mathbf{P}_{k_i}^{i-1-} - K_i H_i \mathbf{P}_{k_i m_i}^{+T} - \mathbf{P}_{k_i m_i}^+ H_i^T K_i^T + K_i (H_i \mathbf{P}_{m_i}^{l_i+} H_i^T + \bar{\mathbf{R}}_{k_i}) K_i^T$$

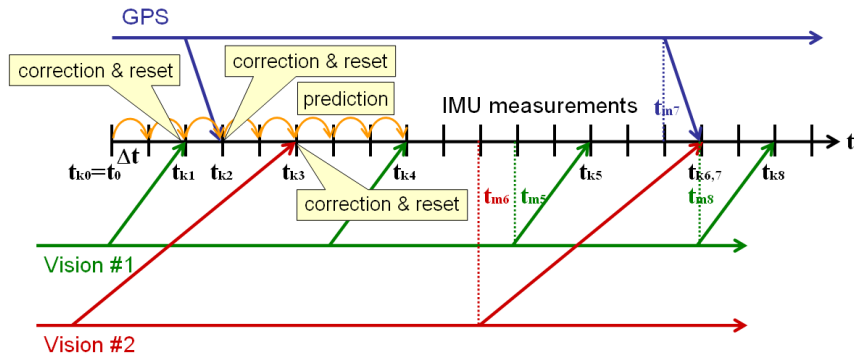


Fig. 5 ESKF process timeline with delayed measurements

where  $P_{k_i m_i}^+ = \mathbb{E} \left[ \delta \tilde{\mathbf{x}}_{k_i}^{i-1-} \tilde{\mathbf{x}}_{m_i}^{l_i+T} \right]$  and  $P_{m_i}^{l_i+} = \mathbb{E} \left[ \delta \tilde{\mathbf{x}}_{m_i}^{l_i+} \delta \tilde{\mathbf{x}}_{m_i}^{l_i+T} \right]$ . As done in [22], the Kalman gain  $K_i$  is chosen so that the trace of  $P_{k_i}^{i-1}$  is minimized.

$$K_i = P_{k_i m_i}^+ H_i^T (H_i P_{m_i}^{l_i+} H_i^T + \bar{R}_{k_i})^{-1} \quad (25)$$

Then the error covariance  $P_k$  becomes

$$P_{k_i}^{i-1} = P_{k_i}^{i-1-} - K_i H_i P_{k_i m_i}^{+T} = P_{k_i}^{i-1-} - P_{k_i m_i}^+ H_i^T (H_i P_{m_i}^{l_i+} H_i^T + \bar{R}_{k_i})^{-1} H_i P_{k_i m_i}^{+T} \quad (26)$$

So what we need to derive is the covariance and correlation matrices  $P_{k_i m_i}^+$  and  $P_{m_i}^{l_i+}$ .

### 3. Back-propagation

Now we derive  $\delta \hat{\mathbf{x}}_{m_i}^{l_i+}$  by back-propagation of the current estimated error state  $\delta \hat{\mathbf{x}}_{k_i}^{i-1-}$ . Figure 6 illustrates this process. The back-propagation and back-injection operations are continued from the current time  $t_{k_i}$  back to the time of the measurement  $t_{m_i}$ . Define the following transition matrix which includes both the propagation and the injection.

$$\Gamma_{k_n k_l} = \begin{cases} \prod_{j=l}^{n-1} \Phi_{k_{j+1} k_j}^j G_{k_j}^\oplus = \Phi_{k_n k_{n-1}}^{n-1} G_{k_{n-1}}^\oplus \cdots \Phi_{k_{l+1} k_l}^l G_{k_l}^\oplus & , n>l \\ I & , n=l \\ \Gamma_{k_l k_n}^{-1} & , n<l \end{cases} \quad (27)$$

Then, the back-propagated error state estimate becomes

$$\delta \hat{\mathbf{x}}_{m_i}^{l_i+} = \Phi_{k_{l_i+1} m_i}^{l_i-1} \delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+} = \Phi_{k_{l_i+1} m_i}^{l_i-1} \sum_{n=l_i+1}^{i-1} \Gamma_{k_n k_{l_i+1}}^{-1} \delta \hat{\mathbf{x}}_{k_n}^{n-1} \quad (28)$$

where  $\delta \hat{\mathbf{x}}_{k_n}^{n-1}$  is the estimated error state updated at  $t_{k_n}$ , given by

$$\delta \hat{\mathbf{x}}_{k_n}^{n-1} = \delta \hat{\mathbf{x}}_{k_n}^{n-1-} + K_n (H_n \delta \hat{\mathbf{x}}_{m_n}^{l_n+} + \bar{\xi}_{k_n}) = K_n (H_n \delta \hat{\mathbf{x}}_{m_n}^{l_n+} + \bar{\xi}_{k_n}) \quad (29)$$

The back-propagated estimation error can be derived as

$$\delta \tilde{\mathbf{x}}_{m_i}^{l_i+} = \Phi_{k_{l_i+1} m_i}^{l_i-1} \Gamma_{k_i k_{l_i+1}}^{-1} \left( \delta \tilde{\mathbf{x}}_{k_i}^{i-1-} - \Delta \mathbf{v}_{k_i}^i \right) \quad (30)$$

where

$$\Delta \mathbf{v}_{k_n}^i = \begin{cases} \Gamma_{k_n k_{l_i+1}} \sum_{j=m_i+1}^{k_{l_i+1}} \Phi_{k_{l_i+1} j}^{l_i} \mathbf{v}_j & , n=l_i+1 \\ \sum_{p=l_i+2}^n \Gamma_{k_n k_p} \bar{\mathbf{v}}_{k_{p-1}} + \Gamma_{k_n k_{l_i+1}} \sum_{j=m_i+1}^{k_{l_i+1}} \Phi_{k_{l_i+1} j}^{l_i} \mathbf{v}_j = \Gamma_{k_n k_{n-1}} \Delta \mathbf{v}_{k_{n-1}}^i + \bar{\mathbf{v}}_{k_{n-1}} & , n \geq l_i+2 \end{cases}$$

Now the covariance matrices  $P_{k_i m_i}^+$  and  $P_{m_i}^{l_i+}$  will be derived. From (30), we obtain the following expressions.

$$P_{m_i}^{l_i+} = \Phi_{k_{l_i+1} m_i}^{l_i-1} \Gamma_{k_i k_{l_i+1}}^{-1} \left( P_{k_i}^{i-1-} - \Delta Q_{k_i}^i + \delta Q_{k_i}^i + \delta Q_{k_i}^{iT} \right) \Gamma_{k_i k_{l_i+1}}^{-T} \Phi_{k_{l_i+1} m_i}^{l_i-T} \quad (31)$$

$$P_{k_i m_i}^+ = \left( P_{k_i}^{i-1-} - \Delta Q_{k_i}^i + \delta Q_{k_i}^i \right) \Gamma_{k_i k_{l_i+1}}^{-T} \Phi_{k_{l_i+1} m_i}^{l_i-T} \quad (32)$$

where  $P_{k_i}^{i-1-}$  is the current predicted error covariance, and  $\Delta Q_{k_n}^i$  and  $\delta Q_{k_n}^i$  are defined as follows.

$$\begin{aligned} \Delta Q_{k_n}^i &= \mathbb{E} \left[ \Delta \mathbf{v}_{k_n}^i \Delta \mathbf{v}_{k_n}^{iT} \right] \\ \delta Q_{k_n}^i &= -\mathbb{E} \left[ (\delta \tilde{\mathbf{x}}_{k_n}^{n-1-} - \Delta \mathbf{v}_{k_n}^i) \Delta \mathbf{v}_{k_n}^{iT} \right] = -\mathbb{E} \left[ \delta \tilde{\mathbf{x}}_{k_n}^{n-1-} \Delta \mathbf{v}_{k_n}^{iT} \right] + \Delta Q_{k_n}^i \end{aligned}$$

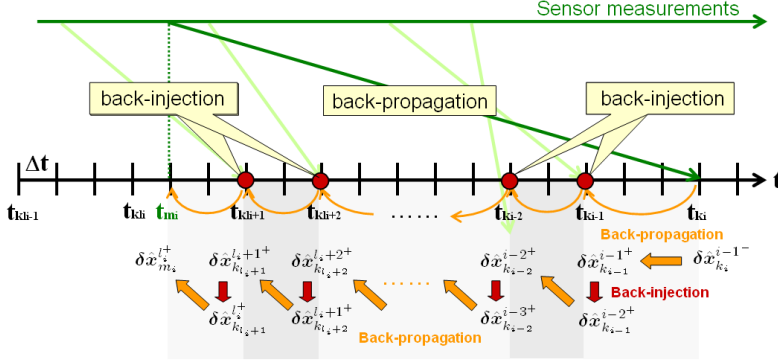


Fig. 6 Back-propagation process timeline with delayed measurements

Starting from  $\Delta Q_{k_{l_i+1}}^i = \sum_{j=m_i+1}^{k_{l_i+1}} \Phi_{k_{l_i+1},j}^{l_i} Q_j \Phi_{k_{l_i+1},j}^{l_i T}$  and  $\delta Q_{k_{l_i+1}}^i = O$ ,  $\Delta Q_{k_n}^i$  and  $\delta Q_{k_n}^i$  can be iteratively calculated up to  $n = i$  by

$$\Delta Q_{k_n}^i = \Gamma_{k_n k_{n-1}} \Delta Q_{k_{n-1}}^i \Gamma_{k_n k_{n-1}}^T + \bar{Q}_{k_{n-1}} \quad (33)$$

$$\delta Q_{k_n}^i = \Gamma_{k_n k_{n-1}} \left( \delta Q_{k_{n-1}}^i + F_{k_{n-1}}^i \right) \Gamma_{k_n k_{n-1}}^T \quad (34)$$

where  $F_{k_{n-1}}^i = \mathbb{E} \left[ \delta \hat{x}_{k_{n-1}}^{n-2} \Delta \mathbf{v}_{k_{n-1}}^{iT} \right]$ . Define  $C_n = K_n H_n \Phi_{k_{l_i+1}, m_n}^{l_n} = K_n \bar{H}_n$ . Then  $F_{k_{n-1}}^i$  can be obtained as follows.

$$F_{k_{n-1}}^i = C_{n-1} \Gamma_{k_{n-1} k_{n-1+1}}^{-1} \left( E_{k_{n-1}}^i - \delta Q_{k_{n-1}}^i \right) + K_{n-1} D_{n-1} \mathbb{E} \left[ \mathbf{v}_{m_{n-1}} \Delta \mathbf{v}_{k_{n-1}}^{iT} \right]$$

where

$$E_{k_{n-1}}^i = \mathbb{E} \left[ \left( \Delta \mathbf{v}_{k_{n-1}}^i - \Delta \mathbf{v}_{k_{n-1}}^{n-1} \right) \Delta \mathbf{v}_{k_{n-1}}^{iT} \right] = \begin{cases} 0 & , m_i \geq m_{n-1} \\ \Delta Q_{k_{n-1}}^i - \Delta Q_{k_{n-1}}^{n-1} & , m_i < m_{n-1} \end{cases}$$

$$\mathbb{E} \left[ \mathbf{v}_{m_{n-1}} \Delta \mathbf{v}_{k_{n-1}}^{iT} \right] = \begin{cases} 0 & , m_i \geq m_{n-1} \\ Q_{m_{n-1}} \Phi_{k_{l_i+1}, m_{n-1}}^{l_{n-1} T} \Gamma_{k_{n-1} k_{l_i+1}}^T & , m_i < m_{n-1} \end{cases}$$

Therefore, for each measurement, we keep updating the matrices  $\Delta Q_{k_n}^i$ ,  $\delta Q_{k_n}^i$  and  $\Gamma_{k_n k_{l_i+1}}$  until  $n = i$  when the measurement becomes available. Then the covariance matrices (31, 32) can be calculated, and used in the Kalman filter update process (25, 26).

When the time of the measurement  $t_{m_i}$  is known (by an image trigger, for example), we can process this back-propagation forward in time. Figure 7 summarizes this ESKF and "forward" back-propagation process for the  $i$ -th measurement. At  $t = t_{m_i}$ ,  $\mathbf{h}_i$ ,  $H_i$  and  $D_i$  can be calculated by using the nominal state  $\mathbf{x}_{m_i}^{l_i}$ . At  $t_{l_i+1}$ , we save  $\bar{H}_i = H_i \Phi_{k_{l_i+1}, m_i}^{l_i}$ . After each EKF update at  $t_{k_n}$ , we track

$$\delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+} = \sum_{p=l_i+1}^n \Gamma_{k_p k_{l_i+1}}^{-1} \delta \hat{\mathbf{x}}_{k_p}^{p-1} = \delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+(n-1)+} + \Gamma_{k_n k_{l_i+1}}^{-1} \delta \hat{\mathbf{x}}_{k_n}^{n-1}, \quad n \geq l_i + 1$$

where  $\delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+} = \mathbf{0}$ . Then  $\delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+(i-1)+} = \delta \hat{\mathbf{x}}_{k_{l_i+1}}^{l_i+(i-1)+}$  is already available at  $t_{k_i}$  when the measurement arrives. From (31, 32) and (25, 26), the Kalman filter update process can be re-written as follows.

$$K_i = \left( P_{k_i}^{i-1-} - \Delta Q_{k_i}^i + \delta Q_{k_i}^i \right) \tilde{H}_i^T \left( \tilde{H}_i \left( P_{k_i}^{i-1-} - \Delta Q_{k_i}^i + \delta Q_{k_i}^i + \delta Q_{k_i}^{iT} \right) \tilde{H}_i^T + \bar{R}_{k_i} \right)^{-1} \quad (35)$$

$$P_{k_i}^{i-1} = P_{k_i}^{i-1-} - K_i \tilde{H}_i \left( P_{k_i}^{i-1-} - \Delta Q_{k_i}^i + \delta Q_{k_i}^{iT} \right) \quad (36)$$

where  $\tilde{H}_i = \bar{H}_i \Gamma_{k_i k_{l_i+1}}^{-1}$  and recall  $\bar{R}_{k_i} = R_{k_i} + D_i Q_v D_i^T$ .

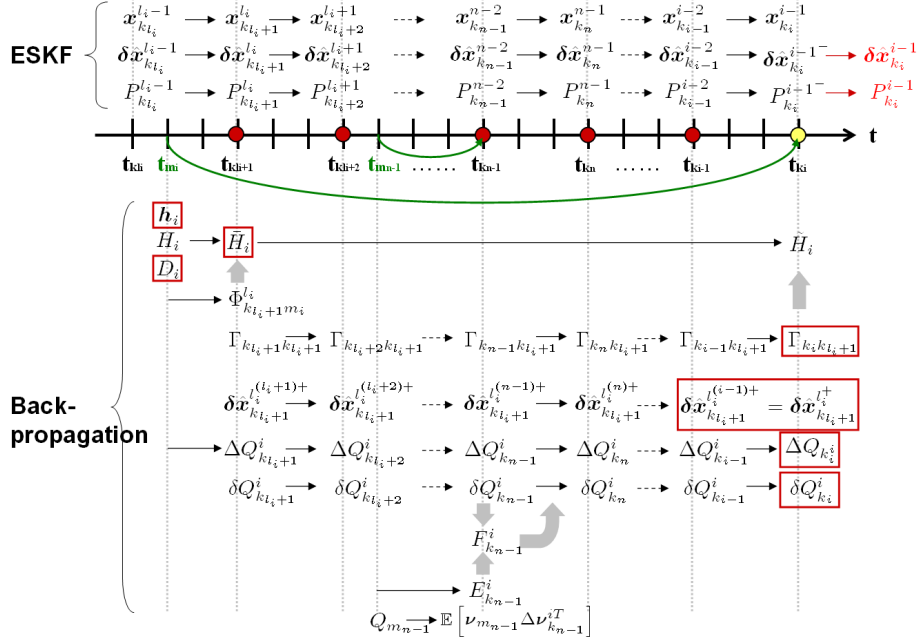


Fig. 7 ESKF and Back-propagation process

#### 4. Another method to handle the delayed measurement in ESKF

A conventional way to handle the delayed measurement in the ESKF is to save all the history of the sensor measurements (including IMU) from  $t_{m_i}$  and  $t_{k_i}$ , and re-runs the KF process from  $t_{m_i}$  to the current time as if the measurements arrived without delay. This method is often adopted in the robotics applications, and is applicable even when the time of measurement is not known until the measurement actually arrives. The drawbacks of this approach are the use of memory for recording all the estimation state and measurement data history, and the redundant computations corresponding to the KF correction steps with already arrived measurements between  $t_{m_i}$  and  $t_{k_i}$ . The proposed delayed-measurement ESKF framework has an advantage of eliminating the redundant computation, distributing the additional computation load at each time step but not at once only when the measurement arrives, and reducing the need of memory capacity.

## IV. Functionality validation in simulations

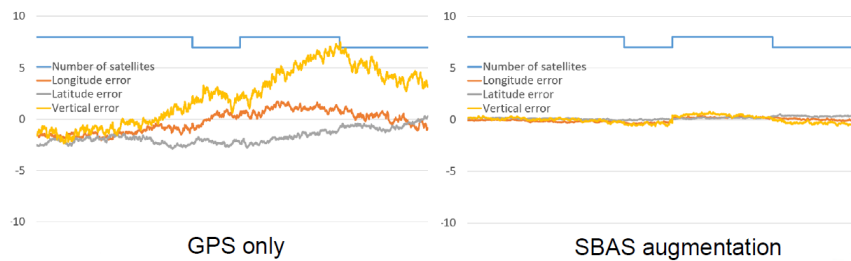
### A. Aircraft simulation framework

The proposed delayed-measurement ESKF framework is applied to design the vision/GNSS navigation system for final approach of an aircraft. For the functionality validation purpose, the navigation system has been implemented in the 6DoF aircraft simulator (of the K50 experimental platform, described later in Section V.A) realized in Matlab/Simulink. It simulates the aircraft dynamics model with basic flight controller, an approach guidance law and the navigation sensor models. It also includes a GPS/SBAS simulation model<sup>†</sup> of standard 24 satellite constellation with different failure modes defined in Section II. This GPS sensor model simulates pseudo-distance measurements to each visible satellite by adding errors in function of selected failure (or nominal) mode, and solves for the position in the geodetic coordinates out of them. For example, Figure 8 compares errors in the simulated GNSS localization solutions with and without SBAS augmentation. Two different and independent vision systems are also simulated; Stereo-vision system and a monocular-camera system. These simulation models add theoretical errors on the image-detection position of the runway feature points, and solve for the camera pose relative to a runway from them. Section V.B describes more these vision systems and models. The camera frequency is set at 10 Hz with the time delay of 80 msec for the stereo-vision system, and 12.5 Hz with 60 msec delay for the monocular-vision system.

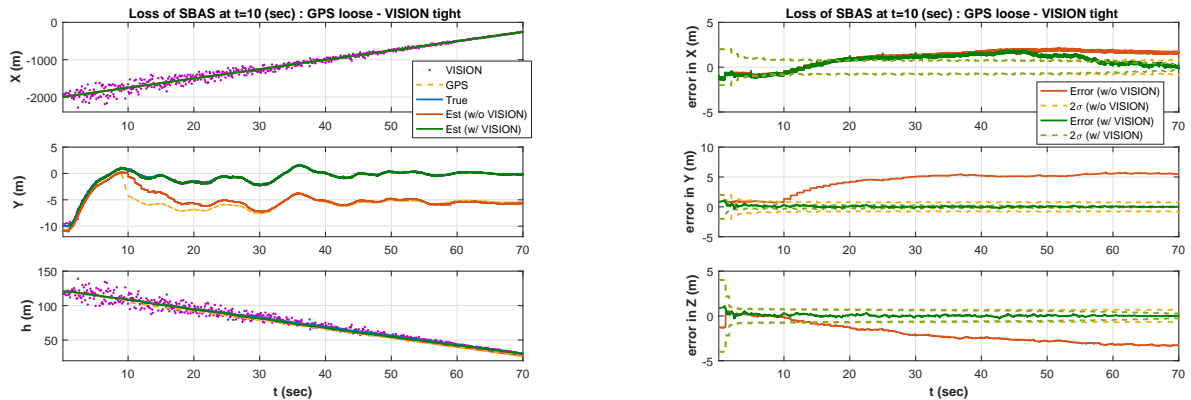
<sup>†</sup> developed within the VISION project by ENRI (Electronic Navigation Research Institute, Tokyo, Japan).

## B. Vision/GNSS integrated navigation system performance

The proposed vision-integrated navigation filter design in different configurations (loose/tight couplings of GNSS and vision, 6D pose estimation and 3D position estimation) were implemented in the simulation to fuse all the available sensor data with simulated GNSS localization degradation. Figure 9 shows an example of the open-loop simulation test and compares the relative position estimation performances with and without integrating the vision-based measurements. In this simulation, GNSS localization precision degraded due to loss of SBAS correction signals at  $t = 10$  (sec). The example uses the navigation filter configuration of GNSS-loose/Vision-tight couplings. The estimation performance difference can be seen in the estimation error profiles (the right figure of Figure 9). The estimation bias of up to 5 meters due to SBAS loss can be removed by using the vision-based measurements. Figure 10 shows the same example but when closing the approach guidance and flight control loop with the resulting navigation solution with and without vision. When not using vision (the right figure of Figure 10), the approach guidance law will make the aircraft lateral maneuver for aligning the biased  $Y$  position estimation at the runway centerline ( $Y = 0$ ). It results in the centerline alignment error of the closed-loop trajectory. This miss-alignment can be avoided when using the vision information in the case of the SBAS loss failure (the left figure).



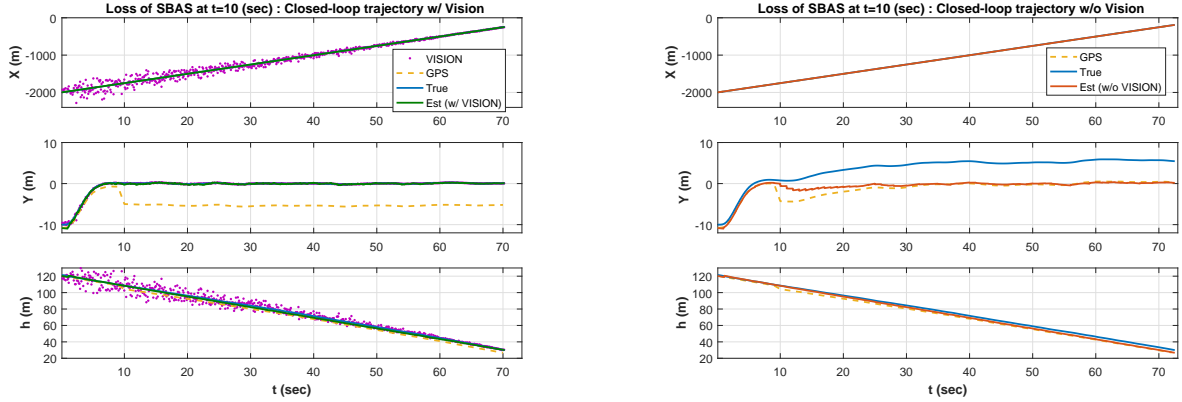
**Fig. 8** GNSS localization error models with (right) and without (left) SBAS augmentation



**Fig. 9** Example of open-loop simulation test results: (left) position estimation and (right) its error with (in green) and without (in red) using the vision information. Loss of SBAS correction signal is simulated at  $t = 10$  (sec)

## C. Fault detection and exclusion

One issue observed in these simulation tests with the vision-integrated navigation is that the navigation solution is rapidly attracted to the degraded GNSS measurements once the runway goes beyond the camera field of view (FoV). As described in the next section, this simulation uses the narrow FoV stereo-vision sensor model, and hence sometimes it loses the runway from the images. This problem can be solved by augmenting the navigation system with the integrity monitoring function which detects and excludes the erroneous measurements. It is planned in the future work to apply the Aircraft Autonomous Integrity Monitoring (AAIM) algorithm based on residual-based or Solution Separation-based approaches[15][16][17]. The AAIM algorithm is to detect and possibly exclude sensor faults from



**Fig. 10** Example of closed-loop approach trajectory: (left) with and (right) without using the vision information. Loss of SBAS correction signal is simulated at  $t = 10$  (sec)

the navigation solution calculation, and also to provide the protection levels. As a preliminary work to this, in this paper, a simple snap-shot residual-based fault detection and exclusion method was tested. Upon each GNSS measurement update in the ESKF, the following normalized innovation residual is used as a detector.

$$q = \Delta z_{k_i}^T (H_i P_{m_i}^{l_i} + \bar{R}_{k_i})^{-1} \Delta z_{k_i}$$

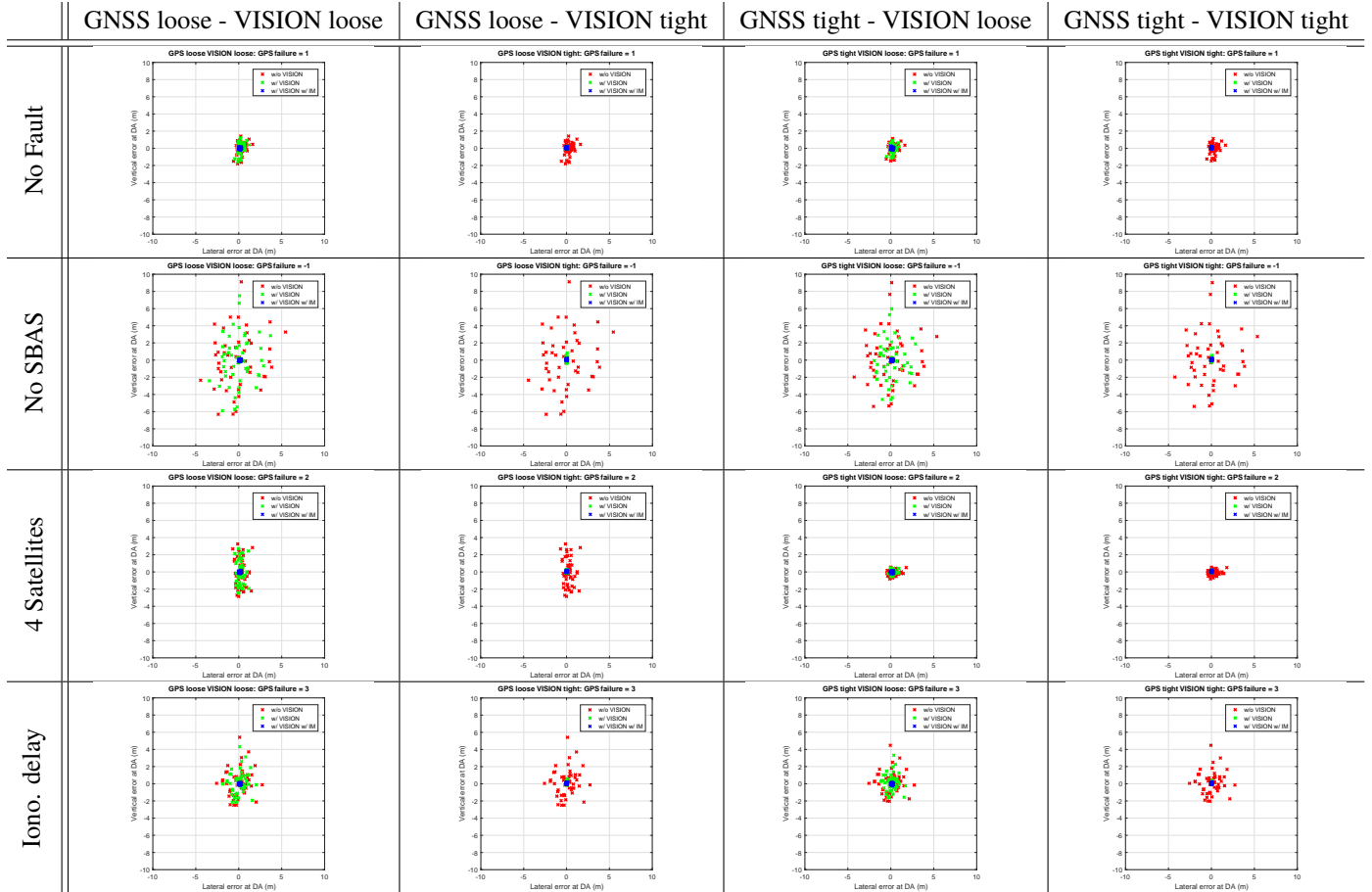
When assuming no sensor fault, this detector follows the  $\chi^2$ -distribution with the degree of freedom of the dimension of the measurement vector (i.e., three for the GNSS loose coupling, and the number of visible satellites for the GNSS tight coupling). The detection threshold value  $T$  is determined by the maximum allowable false alarm probability  $P_{cr}$ , such that,

$$Pr(q > T | \text{fault free}) Pr(\text{fault free}) \leq Pr(q > T | \text{fault free}) = P_{cr}$$

A simple logic of excluding the GNSS position or pseudo-range measurements from the navigation filter when  $q > T$  is applied.

#### D. Closed-loop simulation results

In order to analyze the navigation performance, 50 closed-loop simulations were run for each of the four navigation filter configurations (GNSS tight-/loose-, VISION tight-/loose-couplings) for different GNSS failure modes. Figure 11 compares the results of tracking errors from the 3-degree final approach path when the aircraft reaches at the decision altitude (DA = 30 m). The results when not using the vision information are plotted in red, those when using the vision are in green, and those when using the vision and applying the fault exclusion logic explained in the previous subsection are in blue. The fault cases are; no fault, loss of SBAS correction signal, reduced number of visible satellites to four, and the ionospheric interference which induces large error in the GNSS pseudo-distance measurements. From the figure, we can see how the GNSS fault degrade the final approach precision by comparing the first row results with the others. It also shows that the vision-based measurements aid the navigation system to limit such precision degradation in case of failures. With the vision system error models used in the simulation, the tight-coupling configuration fuses more effectively the vision-based measurements with GNSS. When loose-coupling the vision measurements, the navigation system still relies more on the GNSS measurements than on the vision measurements even after the failure occurs. This is due to a fact that the vision-based localization precision depends on a distance to the runway. The vision-based position measurement error is unbiased but is still much larger than the bias in the GNSS measurement, and the navigation filter results in relying more on the erroneous GNSS measurement. Then the navigation solution is attracted and converged to the biased GNSS position, and will have more difficulty in removing the bias. The fault detection and exclusion function works effectively in such cases. Difference in the navigation performance between the GNSS tight- and loose-couplings can be seen in the case of the reduced number of visible satellites (the third row results of Figure 11). The GNSS tight-coupling takes directly the pseudo-range measurements and avoids to introduce the measurement bias in the estimation process.



**Fig. 11** Closed-loop simulation results of the approach precision at DA without (in red), with (in green) the vision measurements and with the fault exclusion (in blue): Comparison for the four different configurations of the navigation filter, for the four different failure modes simulated at  $t = 10$  (sec).

## V. Towards flight validation

As stated in the Introduction section, the final goal of this work in the VISION project is to evaluate the proposed vision-integrated navigation system onboard a real aircraft with real vision sensors and image processors. This section describes the experimental fixed-wing UAV platform and the two onboard vision systems integrated on it.

### A. K50 fixed-wing UAV platform

An experimental platform used for the flight validation of the proposed vision-integrated navigation system is a fixed-wing UAV called K50-Advanced (Figure12). It has a fuselage length of 3 m, wingspan of 4 m and maximum take-off weight of 60 kg. This platform is manufactured within the VISION project by a Spanish company USOL, and is featured with its high payload capacity of 100 L and 20 kg that is suitable for flight experiments with different onboard systems (avionics, payload computers, cameras, etc.). The K50-Advanced is equipped with the ONERA in-house flight avionics with the basic navigation sensors (including GPS/RTK for the reference), and with a payload computer which serves for the interface between the avionics and other external payload systems and sensors as well as for hosting the experimental programs. The two onboard vision systems send the image processor outputs to this payload computer in which the visio-based navigation system algorithm proposed in this paper will be implemented.

### B. Vision systems

Runways have specific feature points (as already shown in Figure 4) possibly with known world coordinates, which can be utilized for the navigation purpose during final approach. The main challenges of the image-based runway



feature detection are visibility, calibration, accuracy and the refresh rate which is bounded by the speed of on-board image processing. The two different vision systems are integrated onboard the K50 aircraft in this work.

### 1. Stereo vision system

The first vision system was a stereo-vision system, developed by RICOH Co., Ltd., and hung under the belly of the aircraft (Figure 13). Stereo-based methods have the advantage of eliminating the scale ambiguities that are inherent in monocular systems. The prototype of the stereo-vision system used a pair of monochrome image sensors of  $4096 \times 3000$  pixel resolution with a global shutter at 15 Hz. Two identical monochrome image sensors were positioned parallel to each other to obtain a right and a left image. The distance between the two image sensors' optical axes was 400 mm. Each image sensor in this stereo-vision system had a narrow field of view of  $22.8^\circ \times 16.8^\circ$ . The onboard stereo-vision system issued synchronized triggers to the two image sensors and recorded images from each. This stereo-vision system computed a depth map using an FPGA logic component to compare the right and left images. Next, the system estimated the relative position and pose using the stereo camera monochrome images and depth map. This stereo-vision system was driven by a dedicated FPGA-based system that was specifically developed for this purpose. This system was connected to the ONERA avionics system using RS232 connections for the avionics flight data, for the RICOH sensor's data and PPS, with a UDP connection for the images. Figure 14 shows an example of the stereo images and the depth image captured onboard the K50 UAV during its final approach at ONERA Septfond flight test field using this stereo-vision system.



Fig. 12 K50 experimental UAV platform

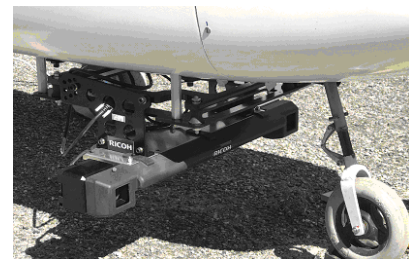


Fig. 13 RICOH stereo-vision system on the K50 platform

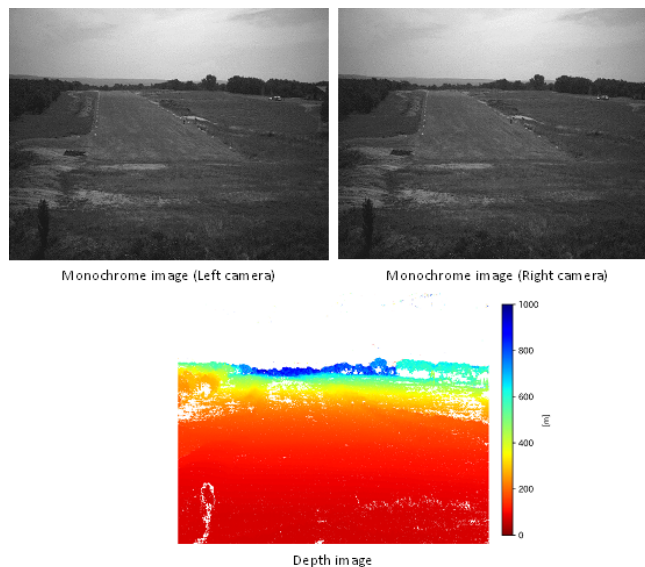


Fig. 14 Examples of depth images on final approach calculated by the RICOH stereo-vision systems

Stereo-vision systems estimate the camera position and attitude from each captured image. The stereo camera can measure 3D positions of objects in a camera-based coordinate system. Therefore, by measuring the 3D position of runway markings (center line and threshold line) in captured images, a stereo camera can directly measure not only the origin position of the runway frame (i.e., the threshold point) but also its X- and Y-axis directions.

The error model of the stereo-vision system used in the simulation (Section IV) is established by adding errors to image position and disparity at the four points on the runway; the left and right corners, the threshold point and the runway end point (the two endpoints of the centerline). The stereo measurement error has many sources, such as image sensor noise, camera tilt, and the line detection process. Each type of error has distinct characteristics. For example, the sensor noise-based error independently manifests at each feature point, whereas the camera tilt moves all feature points in nearly the same direction and distance. The localization error also varies based on the stereo measurement error characteristics. For example, if the positions of all feature points on the runway move constantly, then the image position of the runway origin will vary, but the direction of the lines will remain unchanged. In this case, the pitch and yaw angles of the localization results change, but the roll angle variation is small. In contrast, if disparity errors at the two points of the center line differ from each other, then the estimation of the X-axis direction varies; hence, pitch and height are heavily impacted.

To precisely represent the localization error, we assumed two components of the stereo measurement error.

- Constant component: constant for all points in one frame.
- Independent component: varies based on the location and frame.

Based on previously developed stereo camera experience at RICOH, we set the target error as zero mean normal distribution with standard deviation, as stated in Table. 1. In the simulation model, we generated pseudo-stereo measurements by adding these two normal distribution errors to the ideal stereo measurements calculated from the ideal camera position and attitude. Then, we simulated localization with error, by recalculating the camera position and attitude from these pseudo-stereo measurements.

**Table 1 Target error standard deviation**

	Constant [pix]	Independent [pix]
position (u, v)	0	0.158
disparity (d)	0.333	0.08

To validate this error model, we evaluated the stereo camera disparity accuracy. On the ground, we set a stereo camera on a rotary stage. We captured stereo images of objects at a distance of 400 m from the camera. We evaluated disparity errors at numerous image positions. The average disparity error in neighboring image regions was regarded as a constant component. The independent component was calculated from the deviation of the difference in disparity errors between two neighboring points. The standard deviations of measured constant and independent components of a prototype camera are shown in Table. 2 (with the assumed target). The measured disparity errors are smaller than the assumed target. This result shows that the proposed error model is feasible with respect to disparity. In future work, we will validate a comprehensive localization error model, including factors that have not yet been considered, such as image position error and mechanical vibration.

**Table 2 Measured disparity error standard deviation**

	Constant [pix]	Independent [pix]
position (u, v)	0.144	0.333
disparity (d)	0.055	0.08

## 2. Monocular vision system

The second vision system is a pair of monocular cameras with complementary specifications proposed by Hungarian Academy of Sciences. Two Basler GigE 2048x1536 colored cameras are integrated on the K50 UAV (Figure 15); one with wide 33.5 deg field of view (FOV) and one with narrow 15 deg FOV. It is beneficial, because narrow view angle with similar camera resolution yields higher accuracy, however, it is possible that the narrow angle camera temporarily does not see the runway. Each camera has a dedicated Nvidia Tegra X1 module which consists an NVIDIA Maxwell

GPU with 256 NVIDIA CUDA Cores, Quad-core ARM Cortex-A57 MPCore Processor and 4 GB LPDDR4 Memory. We applied a Crucial 250 GB mSATA SSD for video data recording.

The on-board processing of images consists the image acquisition and storage, feature detection and tracking, and navigation data extraction from features. In [23] the authors introduced a simple image processing approach for FlightGear simulator imagery. The algorithm segments the areas covered by concrete, and performs a masked adaptive threshold to get the markings on the concrete. The pair of sidelines are detected in the Hough space of the markings binary map, and then the threshold line is detected. All the three lines are fine-tuned by an oriented-mean method. The main detected features are the corner points and the vanishing point of the side lines. Figure 16 shows an example result of the image processing performed on an image recorded onboard the K50 UAV on final approach, where a simple CNN classifier finds candidate regions corresponding to the runway markers.

Based on the visual features of the runway detected on the image and the known width of the runway (and assuming parallel sides), the 6D pose can be determined. In our research we examine three possible methods. The 3-point method [24] utilizes the corner points of the runway threshold and the vanishing point of the two side lines. A general iterative method for  $n \geq 3$  points with known world coordinates [25], and a line-based method [26] where the authors assume that the detection of lines is more robust than that of points.



Fig. 15 Payload computers and the monocular cameras (under wings) on the K50 UAV.

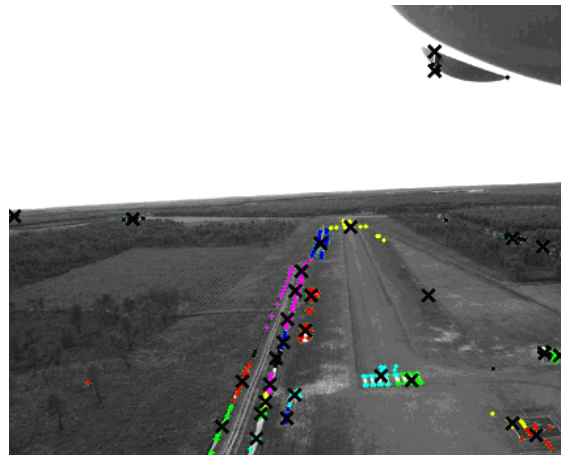


Fig. 16 An example of the runway feature detection: Candidate regions of the runway threshold marker.

## **VI. Conclusion**

This paper presented the vision-integrated navigation system for the aircraft final approach based on the ESKF framework which incorporates the time delay of the vision-based measurements. It benefits from the image trigger notice from the vision system to make the forward back-propagation to handle the time-delay directly in the KF process. The proposed system has been implemented in the simulation with the sensor models (including GNSS/SBAS failure models, and two vision systems), and its closed-loop performance to maintain the localization and final approach path tracking precision in case of GNSS sensor degradation was evaluated.

This is the work performed as a part of the H2020 VISION project, and the final goal of the project is to provide the flight-validated vision-based navigation solution augmented with an integrity monitoring function. Towards the flight validation, the two different vision systems, stereo- and a pair of monocular-vision systems, have been developed and integrated on the K50 UAV experiment platform. Our future work includes the implementation of the proposed vision-integrated navigation system onboard the K50, and making the closed-loop final approach guidance by using its localization solution. Also, we will augment the navigation system with the integrity monitoring function for fault detection and exclusion purpose and for calculating the protection level which is required in civil aviation applications.

## **Acknowledgments**

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 690811 and the Japan New Energy and Industrial Technology Development Organization under grant agreement No. 062800, as a part of the EU/Japan joint research project entitled "'Validation of Integrated Safety-enhanced Intelligent flight cONtrol (VISION).'"

## References

- [1] International Civil Aviation Organization (ICAO), Safety report, 2017 Edition. [https://www.icao.int/safety/Documents/ICAO\\_SR\\_2017\\_18072017.pdf](https://www.icao.int/safety/Documents/ICAO_SR_2017_18072017.pdf)
- [2] G. Hardier, G. Ferreres and M. Sato, Design and Flight Testing of an Adaptive Gain-Scheduled Controller using On-line Model Estimation, IEEE Conference on Control Technology and Application, 2018.
- [3] L. Chen, H. Alwi, C. Edwards and M. Sato, Evaluation of a Sliding Mode Fault Tolerant Controller on the MuPAL-Alpha Research Aircraft, IEEE Conference on Control Technology and Application, 2018.
- [4] P. Chandler (A380 Project Pilot), Airbus Innovation in the Cockpit, Farnborough Briefing, 2004.
- [5] T. Goncalves, J. Azinheira and P. Rives, Homographie-based Visual Servoing of an Aircraft for Automatic Approach and Landing, IEEE International Conference on Robotics and Automation, 2010.
- [6] L. Coutard, F. Chaumette and J.M. Pfimlin, Automatic Landing on Aircraft Carrier by Visual Servoing, IEEE International Conference on Intelligent Robots and Systems, 2011.
- [7] V. Gibert, L. Burlion, A. Chriette, J. Boada and F. Plestan, Nonlinear Observers in Vision System: Application to Civil Aircraft Landing, IEEE European Control Conference, 2015.
- [8] A. Miller, M. Shah and D. Harper, Landing a UAV on a Runway Using Image Registration, IEEE International Conference on Robotics and Automation, 2008.
- [9] P.A. Marianandam and D. Ghose, Vision-based Alignment to Runway during Approach for Landing of Fixed Wing UAVs, International Conference on Advances in Control and Optimization of Dynamical Systems, 2014.
- [10] P. Williams and M. Crump, Intelligent Landing System for Landing UAVs at Unsurveyed Airfields, International Congress of the Aeronautical Sciences, 2012.
- [11] M. Lailacker, K. Kondak, M. Schwarzbach and T. Muskardin, Vision Aided Automatic Landing System for Fixed Wing UAV, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.
- [12] G. Conte and P. Doherty, Vision-based unmanned aerial vehicle navigation using geo-referenced information, Journal on Advances in Signal Processing - Special issue on signal processing advances in robots and autonomy, 09(10), 2009.
- [13] R. Strydom, S. Thurrowgood and M.V. Srinivasan, Visual Odometry: Autonomous UAV Navigation using Optic Flow and Stereo, Australian Conference on Robotics and Automation, 2014.
- [14] S. Weiss, M.W. Achtelik, . Lynen, M.C. Achtelik, L. Kneip, M. Chli and R. Siegwart, Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium, Journal of Field of Robotics, 30(5), 2013.
- [15] M.A. Brenner, Navigation System with Solution Separation Apparatus for Detecting Accuracy Failures, US Patent 5760737 A, 1998.
- [16] M. Joerger, F.C.Chan and B.Pervan, Solution Separation versus Residual-based RAIM, Journal of The Institute of Navigation, Vol.61, No.4, 2014.
- [17] M. Joerger and B.Pervan, Kalman Filter-based Integrity Monitoring against Sensor Faults, AIAA Journal of Guidance, Control and Dynamics, Vol.36, No.2, 2013.
- [18] EuroControl, RNAV Approaches, 2012.
- [19] N. Trawny and S.I. Roumeliotis, Indirect Kalman Filter for 3D Attitude Estimation, Technical Report 2005-002, University of Minnesota, 2005.
- [20] V. Madyastha, V. Ravindra, S. Mallikarjunan and A. Goyal. Extended Kalman Filters vs. Error State Kalman Filter for Aircraft Attitude Estimation, AIAA Guidance, Navigation and Control Conference, 2011.
- [21] J. Sola, Quaternion kinematics for the error-state Kalman filter, CNRS/LAAS Technical notes, 2017.
- [22] T.D. Larsen, N.A. Andersen, O. Ravn and N.K. Poulsen, Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter, IEEE Conference on Decision and Control, 1998.
- [23] A. Hiba, A. Szabo, T. Zsedrovits, P. Bauer and A. Zarandy, Navigation Data Extraction from Monocular Camera Images during Final Approach, IEEE International Conference on Unmanned Aircraft Systems, 2018.

- [24] F. Li, D.Q. Tang and N. Shen, Vision-based Pose Estimation of UAV from Line Correspondences, Elsevier Procedia Engineering, vol.15, 2011.
- [25] C.P. Lu, G.D. Hager and E. Mjolsness, Fast and Globally Convergent Pose Estimation from Video Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6), 2000.
- [26] X. Zhang, K. Wang, Z. Zhang and Q. Yu, A New Line-based Orthogonal Iteration Pose Estimation Algorithm, IEEE International Conference on Information Engineering and Computer Science, 2009.