

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

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

▶ **To cite this version:**

Yoko Watanabe, Augustin Manecy, Antal Hiba, Sho Nagai, Aoki Shin. Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures. AIAA Scitech 2019, Jan 2019, San Diego, CA, United States. pp.0113, 10.2514/6.2019-0113 . hal-02020366

HAL Id: hal-02020366
https://hal.science/hal-02020366

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‡
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-sate Kalman filter (ESKF) framework incooporating time-delayed measurements is established. The proposed delayed-measurement ESKF framework uses a fact that camera image acquisitions are triggered and notified without delay, and performs the back-propagation of the estimated state forward in time to prepare in advance for the future correction step when the corresponding 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, and flight evaluation results will be presented in the final paper.

I. Introduction

Despite an overall decrease tendency in both the number of fatal accidents and fatalities in commercial aircraft operations, 452 accidents still happened in the last five years (2012-2016) resulting more than 2000 onboard fatalities[1]. According to different statistical survey reports on commercial aircraft accidents, 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 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 developped and flight-validated on a fixed-wing UAV (Unmanned Aerial Vehicle). These new GN&C solutions will improve the robustness and self-adaptability of current aircraft flight systems, and hence will contribute to ease

*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

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 the VISION project.

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 addresse monocular vision-only navigaiton 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].

As the preliminary outcomes from this work, this paper proposes the state estimator design by establishiing an Error-State Kalman Filter (ESKF) framework[17][18] 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 and notified to the navigation system without delay, and performs the back-propagation of the estimation state forward in time. In order to do so, we can avoid the redundant KF correction process with already arrived measurements and the stoackage of all the data histories needed in 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 gives a brief description of the vision systems and the experimental UAV platform to bse used for future flight validation. Flight test results of open-loop state estimation performance of the proposed vision-integrated navigation system applied for the aircraft final approach will be expected in the final paper.

II. Sensor failure scenarios on final approach

This paper 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)[16]. 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 aporoach 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 the navigation sensors. In this work, the following sensor failures are considered to happen during the final approach phase, supposing that the accurate aircraft position

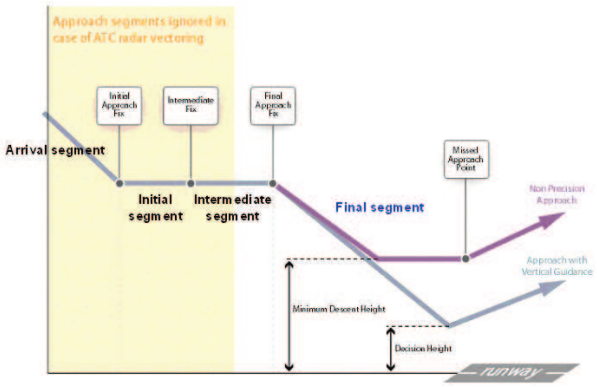


Fig. 1 Instrument Approach Procedure[16]

relative to runway is availabe on the initial approach segment for initializing the visual tracker.

- 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 itnerference with SBAS augmentation
- In the classical ILS approach procedure:
  - Lack of ILS signals
  - Misleading of ILS signals due to secondary lob

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

III. Vision-integrated navigation system design

The idea of this work is to design the vision-integrated navigaiton 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 (barometers, inclinometers etc.) 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)[17][18]. 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 [18] 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 [19] for the 6D pose estimation by fusion of IMU and other sensor measurements (such as GPS). See [19] for definition of the quaternion operators. Let $\boldsymbol{x}_t$ be a **true state** to be estimated.

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{X}_t^T & \boldsymbol{V}_t^T & \boldsymbol{b}_{a_t}^T & P_{0_t} & \boldsymbol{q}_t^T & \boldsymbol{b}_{\omega_t}^T \end{bmatrix} \tag{1}$$

where $\boldsymbol{X}_t$ and $\boldsymbol{V}_t$ are position and velocity vectors in a locally-fixed runway frame, $\boldsymbol{q}_t$ is a quaternion vector from the vehicle body frame to the runway frame, P_{0_t} be a pressure adjusted to a standard atomosphere at the sea level, $\boldsymbol{b}_{a_t}$ and $\boldsymbol{b}_{\omega_t}$ are acceleration and angular rate measurement bias in the vehicle body frame respectively. The runway frame is

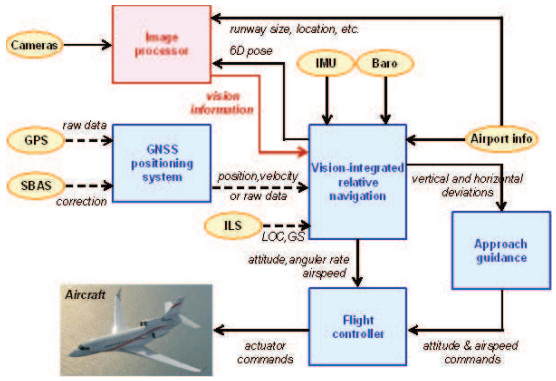


Fig. 2 Vision-integrated Navigation System

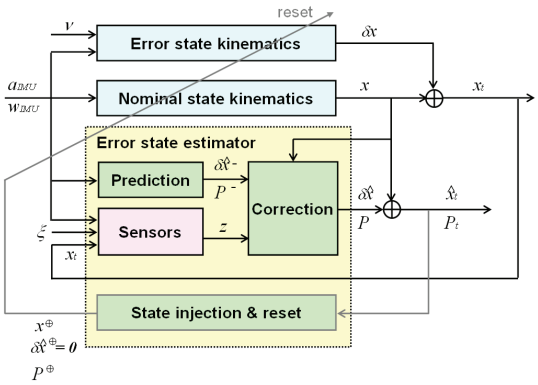


Fig. 3 Error State Kalman Filter Process

defined as a frame fixed to the runway with its origin at the threshold point, its X-axis aligns with the runway landing direction and Z-axis vertically downwards. Then its dynamics can be written by

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

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

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

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{X}^T & \boldsymbol{V}^T & \boldsymbol{b}_a^T & P_0 & \boldsymbol{q}^T & \boldsymbol{b}_\omega^T \end{bmatrix} \tag{3}$$

It evolves with the following deterministic dynamics.

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

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

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

where $\delta\boldsymbol{\theta}$ is an error angle vector and $\boldsymbol{q}(\delta\boldsymbol{\theta}) = \begin{bmatrix} \cos\frac{\|\delta\boldsymbol{\theta}\|}{2} & \sin\frac{\|\delta\boldsymbol{\theta}\|}{2}\frac{\delta\boldsymbol{\theta}^T}{\|\delta\boldsymbol{\theta}\|} \end{bmatrix}^T$. Then the error state kinematics becomes

linear as below.

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

where $\mathcal{A} = [\boldsymbol{a}\times]$ 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

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

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

where $\boldsymbol{\nu}_j \sim \mathcal{N}(\mathbf{0},\ Q_j \simeq Q\boldsymbol{\nu}_{\Delta t_j})$ 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 $\boldsymbol{x}_t$.

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).

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

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

2. GNSS (Gloabl Navigation Satellite System)

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

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

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

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

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

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

$\boldsymbol{X}_{SAT_i}^{ECEF}$ is a known position of the i-th satellite in the ECEF (Earth-Centered Earth-Fixed) frame, c is a light speed, τ_t and τ_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 ionosphereic and tropospheric signal delays, multi-path effects, etc.]. σ_{UERE} (UERE stands for User Equivalent Range Error) is calculated and provided by the receiver. The receiver clock-bias τ_t can be modeled as a 2nd order random process such as

$$\dot{\boldsymbol{x}}_{\tau_t} = \begin{bmatrix} \dot{\tau}_t \\ \dot{\nu}_{\tau_t} \end{bmatrix} = \begin{bmatrix} \nu_{\tau_t} \\ 0 \end{bmatrix} + \begin{bmatrix} \nu_\tau \\ \nu_{\nu_\tau} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_{\tau_t} + \boldsymbol{\nu}_\tau = F_\tau \boldsymbol{x}_{\tau_t} + \boldsymbol{\nu}_\tau$$

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

$$\dot{\boldsymbol{x}}_\tau = F_\tau \boldsymbol{x}_\tau, \quad \boldsymbol{\delta}\dot{\boldsymbol{x}}_\tau = F_\tau \boldsymbol{\delta x}_\tau + \boldsymbol{\nu}_\tau \tag{9}$$

Then the measurement model becomes

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

3. Barometer

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

$$z_{BARO} = P_{0_t} \exp\left(\frac{gM}{R_0 T_0}(Z_t - Z_{THD})\right) + \xi_{BARO} = \tilde{h}_{BARO}(\boldsymbol{x}_t) + \xi_{BARO}$$

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

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 $\Delta\boldsymbol{X}_C$ and $\Delta\boldsymbol{q}_C$ be a known camera position and orientation in the vehicle body frame. Then the true camera pose in the runway frame is given by

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

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

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

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 $d_{p_t} = Bf_x/Z_{p_t}^C$.

In this work, the image processor is developped to detect the runway features shown in Figure 4 to calculate the 6D pose of the camera in the runway frame(see Section IV.B). $\boldsymbol{p}_{THD}$, $\boldsymbol{p}_{L/R}$ and $\boldsymbol{p}_V$ are the pixel-coordinates of the

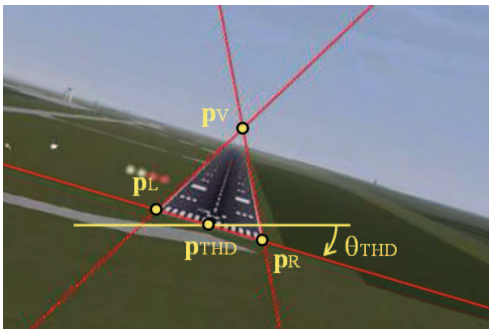


Fig. 4 Runway feature points to be detected on an image

threshold point, the runway left/right corners and the vanishing point of the two parallel sidelines respectively. The 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.

$$z_{VISION} = \tilde{\boldsymbol{h}}_{VISION}(\boldsymbol{x}_t) + \boldsymbol{\xi}_{VISION}$$

For the tight-coupling, the image processor raw outputs such as the feature point pixel-coordinates and the disparity measurement are used instead. Their measurement model $\tilde{\boldsymbol{h}}$ in function of the true state can be derived by the pin-hole camera model (10).

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 $\boldsymbol{\delta\hat{x}}_j^-$ 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.

$$\boldsymbol{\delta\hat{x}}_j^- = \Phi_j \boldsymbol{\delta\hat{x}}_{j-1} \tag{11}$$

$$P_j^- = \mathbb{E}\left[\boldsymbol{\delta\tilde{x}}_j^- \boldsymbol{\delta\tilde{x}}_j^{-T}\right] = \Phi_j P_{j-1} \Phi_j^T + Q_j \tag{12}$$

It should be noted that, in the ESKF framework, $\boldsymbol{\delta\hat{x}}_j^-$ is always zero as $\boldsymbol{\delta\hat{x}}_{j-1}$ 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.

$$z_k = \tilde{\boldsymbol{h}}(\boldsymbol{x}_t(t_k), \boldsymbol{a}_t(t_k), \boldsymbol{\omega}_t(t_k)) + \boldsymbol{\xi}_k \tag{13}$$

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

$$z_k \simeq \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k}) + H(\boldsymbol{x}_k, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})\boldsymbol{\delta x}_k + \boldsymbol{\xi}_k + D(\boldsymbol{x}_k, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})\boldsymbol{\nu}(t_k)$$

where

$$D(\boldsymbol{x}_t, \boldsymbol{a}_{IMU}, \boldsymbol{\omega}_{IMU}) = \left[\frac{\partial \tilde{\boldsymbol{h}}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{\omega}_t)}{\partial \boldsymbol{a}_t} R(\boldsymbol{q}_t) \quad O \quad \boldsymbol{0} \quad \frac{\partial \tilde{\boldsymbol{h}}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{\omega}_t)}{\partial \boldsymbol{\omega}_t} \quad O\right]\Bigg|_{\substack{\boldsymbol{a}_t = R(\boldsymbol{q}_t)(\boldsymbol{a}_{IMU} - \boldsymbol{b}_{a_t}) + \boldsymbol{g} \\ \boldsymbol{\omega}_t = \boldsymbol{\omega}_{IMU} - \boldsymbol{b}_{\omega_t}}}$$

$$H(\boldsymbol{x}_k, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k}) = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})}{\partial \boldsymbol{x}_t}\Bigg|_{\boldsymbol{x}_t = \boldsymbol{x}_k} \frac{\partial(\boldsymbol{x}_k \oplus \boldsymbol{\delta x})}{\partial \boldsymbol{\delta x}}$$

$$= \frac{\partial \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})}{\partial \boldsymbol{x}_t}\Bigg|_{\boldsymbol{x}_t = \boldsymbol{x}_k} \begin{bmatrix} R(\boldsymbol{q}_k) & O & O & \boldsymbol{0} & O & O \\ O & R(\boldsymbol{q}_k) & O & \boldsymbol{0} & O & O \\ O & O & I & \boldsymbol{0} & O & O \\ O & O & O & 1 & O & O \\ O & O & O & \boldsymbol{0} & \frac{1}{2}\begin{bmatrix} -\boldsymbol{q}_{v_k}^T \\ q_{w_k} I + [\boldsymbol{q}_{v_k}\times] \end{bmatrix} & O \\ O & O & O & \boldsymbol{0} & O & I \end{bmatrix}$$

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

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

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

$$\begin{aligned} \boldsymbol{\delta}\hat{\boldsymbol{x}}_k &= \boldsymbol{\delta}\hat{\boldsymbol{x}}_k^- + K_k(\bar{\boldsymbol{z}}_k - H_k \boldsymbol{\delta}\hat{\boldsymbol{x}}_k^-) = K_k(\boldsymbol{z}_k - \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{a}_{IMU_k}, \boldsymbol{\omega}_{IMU_k})) && (15) \\ P_k &= (I - K_k H_k) P_k^-, \quad K_k = P_k^- H_k^T (H_k P_k^- H_k^T + \bar{R}_k)^{-1} && (16) \end{aligned}$$

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 $\boldsymbol{\delta}\hat{\boldsymbol{x}}_k$ is injected to the nominal state by (5) and the error state is reset to zero, i.e., $\boldsymbol{x}_k^\oplus = \boldsymbol{x}_k \oplus \boldsymbol{\delta}\hat{\boldsymbol{x}}_k$ and $\boldsymbol{\delta}\hat{\boldsymbol{x}}_k^\oplus = \boldsymbol{0}$. As the true state does not change by the injection, the true error state will also change and hence

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

From the definition (5),

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

Therefore, the error covariance P_k should be also reset to

$$P_k^\oplus = \mathbb{E}\left[\boldsymbol{\delta}\tilde{\boldsymbol{x}}_k^\oplus \boldsymbol{\delta}\tilde{\boldsymbol{x}}_k^{\oplus T}\right] = \mathbb{E}\left[\boldsymbol{\delta x}_k^\oplus \boldsymbol{\delta x}_k^{\oplus T}\right] = G^\oplus(\boldsymbol{\delta}\hat{\boldsymbol{x}}_k) P_k G^{\oplus T}(\boldsymbol{\delta}\hat{\boldsymbol{x}}_k) \tag{18}$$

The ESKF procedure is reset with $\boldsymbol{x}_k^\oplus$, $\boldsymbol{\delta}\hat{\boldsymbol{x}}_k^\oplus = \boldsymbol{0}$ 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 $\boldsymbol{z}_{k_i}$ 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 $\boldsymbol{x}_j^{i-1}$ and $\boldsymbol{\delta x}_j^{i-1}$ respectively. The state injection and ESKF reset process gives following relation between the $(i-1)$-th and i-th states, where $\boldsymbol{\delta}\hat{\boldsymbol{x}}_{k_i}^{i-1}$ is the $(i-1)$-th error state updated at t_{k_i} with the i-th measurement.

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

1. Nominal state propagation and Error state prediction

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

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

$$\delta\boldsymbol{x}_{k_i}^{i-1} = \delta\tilde{\boldsymbol{x}}_{k_i}^{i-1-} = \Phi_{k_i k_{i-1}}^{i-1} \delta\boldsymbol{x}_{k_{i-1}}^{i-1} + \sum_{j=k_{i-1}+1}^{k_i} \Phi_{k_i j}^{i-1} \boldsymbol{\nu}_j = \Phi_{k_i k_{i-1}}^{i-1} \delta\boldsymbol{x}_{k_{i-1}}^{i-1} + \bar{\boldsymbol{\nu}}_{k_{i-1}} \tag{20}$$

$$P_{k_i}^{i-1-} = \Phi_{k_i k_{i-1}}^{i-1} P_{k_{i-1}}^{i-1} \Phi_{k_i k_{i-1}}^{i-1^T} + \sum_{j=k_{i-1}+1}^{k_i} \Phi_{k_i j}^{i-1} Q_j \Phi_{k_i j}^{i-1^T} = \Phi_{k_i k_{i-1}}^{i-1} P_{k_{i-1}}^{i-1} \Phi_{k_i k_{i-1}}^{i-1^T} + \bar{Q}_{k_{i-1}} \tag{21}$$

2. Error state correction with delayed measurement

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

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

where $\boldsymbol{x}_{m_i}^{l_i}$ and $\delta\boldsymbol{x}_{m_i}^{l_i}$ 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

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

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

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

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

The error covariance can be obtained as follows.

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

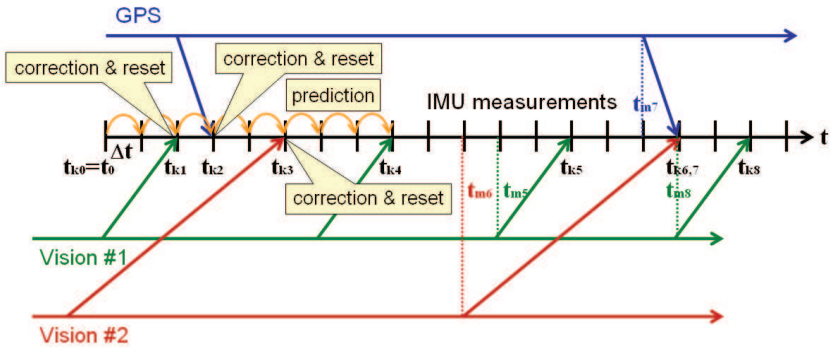


Fig. 5 ESKF process timeline with delayed measurements

where $P^{+}_{k_i m_i} = \mathbb{E}\left[\delta\tilde{\boldsymbol{x}}^{i-1^-}_{k_i}\tilde{\boldsymbol{x}}^{l_i{}^+T}_{m_i}\right]$ and $P^{l_i+}_{m_i} = \mathbb{E}\left[\delta\tilde{\boldsymbol{x}}^{l_i+}_{m_i}\delta\tilde{\boldsymbol{x}}^{l_i^+T}_{m_i}\right]$. As done in [20], the Kalman gain K_i is chosen so that the trace of $P^{i-1}_{k_i}$ is minimized.

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

Then the error covariance P_k becomes

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

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

3. Back-propagation

Now we derive $\delta\hat{\boldsymbol{x}}^{l_i+}_{m_i}$ by back-propagation of the current estimated error state $\delta\hat{\boldsymbol{x}}^{i-1^-}_{k_i}$. 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.

$$\Gamma_{k_n k_l} = \begin{cases} \prod_{j=l}^{n-1} \Phi^{j}_{k_{j+1}k_j} G^{\oplus}_{k_j} = \Phi^{n-1}_{k_n k_{n-1}} G^{\oplus}_{k_{n-1}} \cdots \Phi^{l}_{k_{l+1}k_l} G^{\oplus}_{k_l} & , \text{ n>l} \\ I & , \text{ n=l} \\ \Gamma^{-1}_{k_l k_n} & , \text{ n<l} \end{cases} \tag{27}$$

Then, the back-propagated error state estimate becomes

$$\delta\hat{\boldsymbol{x}}^{l_i+}_{m_i} = \Phi^{l_i^{-1}}_{k_{l_i+1}m_i}\delta\hat{\boldsymbol{x}}^{l_i^+}_{k_{l_i+1}} = \Phi^{l_i^{-1}}_{k_{l_i+1}m_i}\sum_{n=l_i+1}^{i-1}\Gamma^{-1}_{k_n k_{l_i+1}}\delta\hat{\boldsymbol{x}}^{n-1}_{k_n} \tag{28}$$

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

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

The back-propagated estimation error can be derived as

$$\delta\tilde{\boldsymbol{x}}^{l_i+}_{m_i} = \Phi^{l_i^{-1}}_{k_{l_i+1}m_i}\Gamma^{-1}_{k_i k_{l_i+1}}\left(\delta\tilde{\boldsymbol{x}}^{i-1-}_{k_i} - \Delta\boldsymbol{\nu}^{i}_{k_i}\right) \tag{30}$$

where

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

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

$$P^{l_i+}_{m_i} = \Phi^{l_i^{-1}}_{k_{l_i+1}m_i}\Gamma^{-1}_{k_i k_{l_i+1}}\left(P^{i-1^-}_{k_i} - \Delta Q^{i}_{k_i} + \delta Q^{i}_{k_i} + \delta Q^{iT}_{k_i}\right)\Gamma^{-T}_{k_i k_{l_i+1}}\Phi^{l_i^{-T}}_{k_{l_i+1}m_i} \tag{31}$$

$$P^{+}_{k_i m_i} = \left(P^{i-1^-}_{k_i} - \Delta Q^{i}_{k_i} + \delta Q^{i}_{k_i}\right)\Gamma^{-T}_{k_i k_{l_i+1}}\Phi^{l_i^{-T}}_{k_{l_i+1}m_i} \tag{32}$$

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

$$\Delta Q^{i}_{k_n} = \mathbb{E}\left[\Delta\boldsymbol{\nu}^{i}_{k_n}\Delta\boldsymbol{\nu}^{iT}_{k_n}\right]$$

$$\delta Q^{i}_{k_n} = -\mathbb{E}\left[(\delta\tilde{\boldsymbol{x}}^{n-1-}_{k_n} - \Delta\boldsymbol{\nu}^{i}_{k_n})\Delta\boldsymbol{\nu}^{iT}_{k_n}\right] = -\mathbb{E}\left[\delta\tilde{\boldsymbol{x}}^{n-1-}_{k_n}\Delta\boldsymbol{\nu}^{iT}_{k_n}\right] + \Delta Q^{i}_{k_n}$$

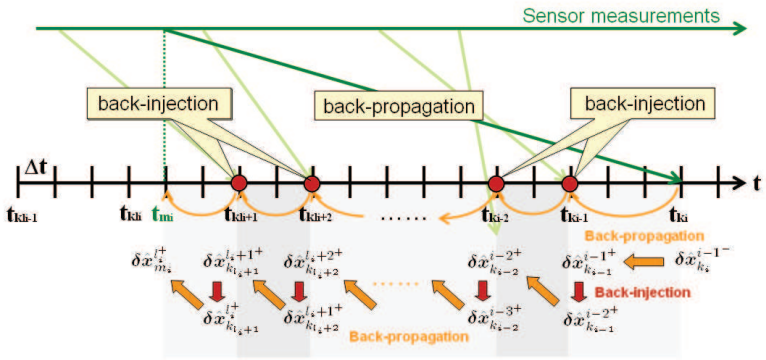


Fig. 6 Back-propagation process timeline with delayed measurements

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

$$\Delta Q^i_{k_n} = \Gamma_{k_n k_{n-1}} \Delta Q^i_{k_{n-1}} \Gamma^T_{k_n k_{n-1}} + \bar{Q}_{k_{n-1}} \tag{33}$$

$$\delta Q^i_{k_n} = \Gamma_{k_n k_{n-1}} \left(\delta Q^i_{k_{n-1}} + F^i_{k_{n-1}} \right) \Gamma^T_{k_n k_{n-1}} \tag{34}$$

where $F^i_{k_{n-1}} = \mathbb{E}\left[\delta\hat{\boldsymbol{x}}^{n-2}_{k_{n-1}} \Delta\boldsymbol{\nu}^{iT}_{k_{n-1}} \right]$. Define $C_n = K_n H_n \Phi^{l_n^{-1}}_{k_{l_n+1} m_n} = K_n \bar{H}_n$. Then $F^i_{k_{n-1}}$ can be obtained as follows.

$$F^i_{k_{n-1}} = C_{n-1} \Gamma^{-1}_{k_{n-1} k_{l_{n-1}+1}} (E^i_{k_{n-1}} - \delta Q^i_{k_{n-1}}) + K_{n-1} D_{n-1} \mathbb{E}\left[\boldsymbol{\nu}_{m_{n-1}} \Delta\boldsymbol{\nu}^{iT}_{k_{n-1}} \right]$$

where

$$E^i_{k_{n-1}} = \mathbb{E}\left[(\Delta\boldsymbol{\nu}^i_{k_{n-1}} - \Delta\boldsymbol{\nu}^{n-1}_{k_{n-1}}) \Delta\boldsymbol{\nu}^{iT}_{k_{n-1}} \right] = \begin{cases} 0 & ,\mathrm{m}_i \geq m_{n-1} \\ \Delta Q^i_{k_{n-1}} - \Delta Q^{n-1}_{k_{n-1}} & ,\mathrm{m}_i < m_{n-1} \end{cases}$$

$$\mathbb{E}\left[\boldsymbol{\nu}_{m_{n-1}} \Delta\boldsymbol{\nu}^{iT}_{k_{n-1}} \right] = \begin{cases} 0 & ,\mathrm{m}_i \geq m_{n-1} \\ Q_{m_{n-1}} \Phi^{l_{n-1} T}_{k_{l_{n-1}+1} m_{n-1}} \Gamma^T_{k_{n-1} k_{l_{n-1}+1}} & ,\mathrm{m}_i < m_{n-1} \end{cases}$$

Therefore, for each measurement, we keep updating the matrices $\Delta Q^i_{k_n}$, $\delta Q^i_{k_n}$ 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 $t = t_{m_i}$, $\boldsymbol{h}_i$, H_i and D_i can be calculated by using the nominal state $\boldsymbol{x}^{l_i}_{m_i}$. At t_{l_i+1}, we save $\bar{H}_i = H_i \Phi^{l_i^{-1}}_{k_{l_i+1} m_i}$. After each EKF update at t_{k_n}, we track

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

where $\delta\hat{\boldsymbol{x}}^{l_i^{l_i+}}_{k_{l_i+1}} = \boldsymbol{0}$. Then $\delta\hat{\boldsymbol{x}}^{l_i^{+}}_{k_{l_i+1}} = \delta\hat{\boldsymbol{x}}^{l_i^{(i-1)+}}_{k_{l_i+1}}$ 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.

$$K_i = \left(P^{i-1^-}_{k_i} - \Delta Q^i_{k_i} + \delta Q^i_{k_i} \right) \tilde{H}^T_i \left(\tilde{H}_i \left(P^{i-1^-}_{k_i} - \Delta Q^i_{k_i} + \delta Q^i_{k_i} + \delta Q^{iT}_{k_i} \right) \tilde{H}^T_i + \bar{R}_{k_i} \right)^{-1} \tag{35}$$

$$P^{i-1}_{k_i} = P^{i-1^-}_{k_i} - K_i \tilde{H}_i \left(P^{i-1^-}_{k_i} - \Delta Q^i_{k_i} + \delta Q^{iT}_{k_i} \right) \tag{36}$$

where $\tilde{H}_i = \bar{H}_i \Gamma^{-1}_{k_i k_{l_i+1}}$ and recall $\bar{R}_{k_i} = R_{k_i} + D_i Q_\nu D^T_i$.

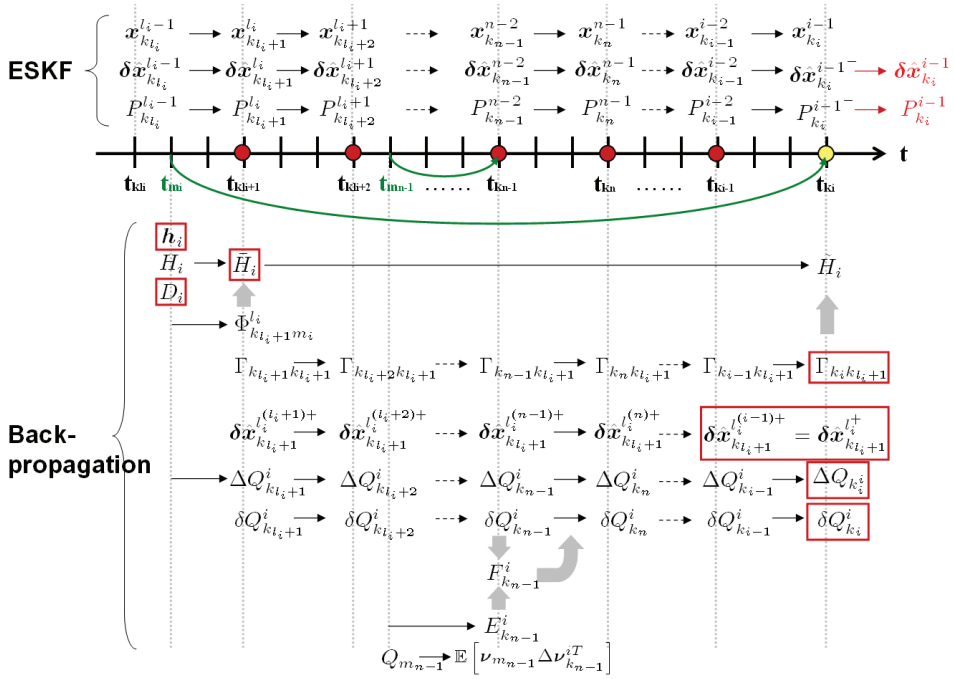


Fig. 7 ESKF and Back-propagation process

4. Another method to handle the delayed measurement in ESKF

A conventioal 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.

5. Functionality validation in simulations

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, it has been implmented in the 6DoF aircraft simulator (of the K50 experimental platform, described later in Section IV.A) with basic flight controller, an approach guidance law and the navigation sensor models. It includes a GPS/SBAS simulation model[†] of standard 24 satellite constellation with different failure modes defind in Section II. It 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. A stereo-vision sensor is also simulated by adding theoretical errors on the image-detection position of the runway feature points. The camera frequency is set at 10 Hz, and the time delay of 80 msec due to the image processing time is simulated.

The vision-integrated navigation filters with different configurations (loose/tight couplings of GNSSS 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 and velocity estimation performances with and without using the vision-based

[†]provided by a VISION project partner, Electoronic Navigation Reserach Institute, Tokyo, Japan.

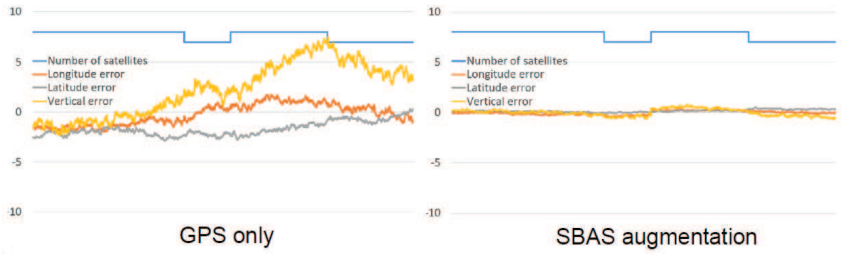


Fig. 8 GNSS localization error models with (right) and without (left) SBAS augmentation

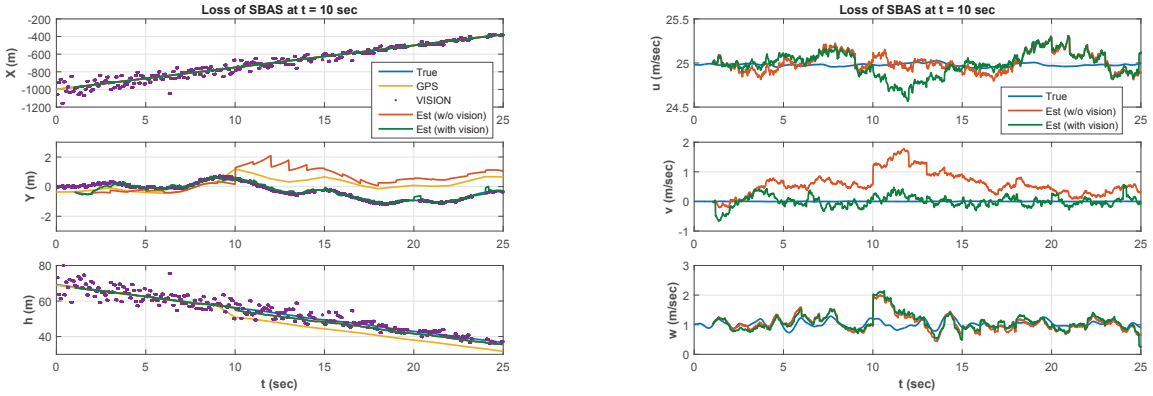


Fig. 9 Example of open-loop simulation test results: (left) position and (right) velocity estimation performances with (in green) and without (in red) using the vision information. Loss of SBAS correction signal is simulated at $t = 10$ (sec)

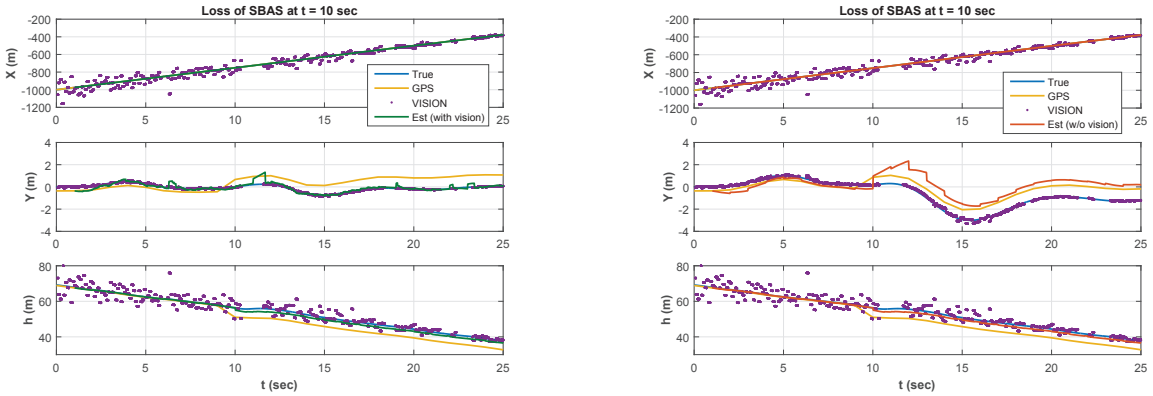


Fig. 10 Example of closed-loop simulation test results: (left) with and (right) without using the vision information. Loss of SBAS correction signal is simulated at $t = 10$ (sec)

measurement. In this simulation, a failure of loss of SBAS correction signals which degrades the GNSS localization precision is simulated to happen at $t = 10$ (sec). This uses the GNSS-tight coupled navigation filter which uses the GNSS pseudo-range measurements. The estimation performance difference can be seen in the lateral deviation of Y position. Without the additional vision information, a jump in the position estimation occurs at $t = 10$ (sec) when the failure occurred and the estimation bias remains. This jump in the GNSS position measurement will induce also the jumps in the velocity and acceleration bias estimates and they take time to be re-converged. The performance in the altitude estimation are nearly the same thanks to the barometer 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 deviated Y position estimation at the runway centerline ($Y = 0$). It results in the centerline alignment error of the closed-loop trajectory. This miss-alignement can be avoided when using the vision information in the case of the SBAS loss failure (the left figure). These are the results of the GNSS-tight coupling navigation filter, but the similar performances were obtained with the loose-coupled filter.

A problem observed in this simulation tests with the vision-based navigation is that the navigaiton solution is rapidly attracted to the degraded GNSS measurements once the runway goes beyond the camera fielf of view (FoV). This is seen in the Y position estimate at around $t = 12$ (sec) on the left figure of Figure 10. This simulation uses the narrow FoV stereo-vision sensor model from Section IV.B.1, and hence sometimes it looses 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 to apply the Aircraft Autonomous Integrity Monitoring (AAIM)-lile algorithm based on Multiple Solution Separation (MSS) method[15]. A possible way to estimate and compensate this kind of sudden jumps in the measurement will be also explored in future.

IV. Towards flight validation

As stated in the Introduction, 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. The two vision systems have been integrated on the experimental UAV platform and we have started the first flight test campaigns for the image data recording. The open-loop flight test results of the proposed vision-integrated navigation system will be expected in the final paper.

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 (Figure11). 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 onbaord systems (avionics, payload computers, cameras, etc.). The K50-Advanced is equipped with the ONERA in-house flight avioncis 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 program codes. The two onboard vision systems send the image processor outputs to this payload computer in which the visio-based navigation system algorithm will be implemented for its flight validation.



Fig. 11 K50 experimental UAV platform



Fig. 12 RICOH stereo-vision system on the K50 platform

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 prupose 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 onbord the K50 aircraft in this work.

1. Stereo vision system

The first vision system is a stereo-vision system, developped by RICOH Co. Ltd., hanged under a belly of the aircraft. Stereo-based methods have the advantage of eliminating the scale ambiguities that are inherent in monocular systems. The first prototype of the stereo-vision system (Figure 12) uses a pair of Blackfly 1920x1200 GigE colored cameras with a narrow field of view of 18x11 degrees for the detection precision. The estimation of relative position to the runway uses the image position and the disparity information of the center line and the threshold of the runway. The algorithm applies edge filters by each pixel for extracting the region which includes the center line (a vertical line on an image) and the threshold line (horizontal line on an image). Then the model fitting method like RANSAC (Random sample consensus) is applied on the extracted feature pixels in order to detect the lines robustly to the noises. Once the image position and disparity of both detected lines are validated to be consistent with the predicted relative pose estimation from the navigaiton filter, then it proceeds to calculate the navigation data.

Two manual flights of K50 have been made so far with this stereo-vision system onbaord and the image and flight test data were collected. Figure 13 shows examples of the depth map calculated from the real stereo-images of the runway. For the preliminary validation, the 6D camera pose was estimated from the stereo-calculated disparity on the recorded image sequence (with manually detected feature points due to a lack of the centerline marking on the tested site), and compared to the GPS-position and INS-attitude data. The comparison results shown in Figure 14 give an idea on the image-based 6D pose estimation accuracy (when the runway features are well detected).

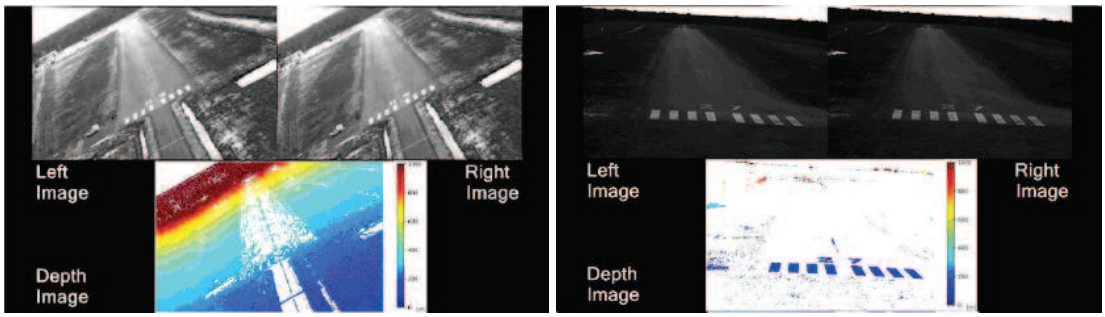


Fig. 13 Examples of depth images on final approach calculated by the stereo-vision systems

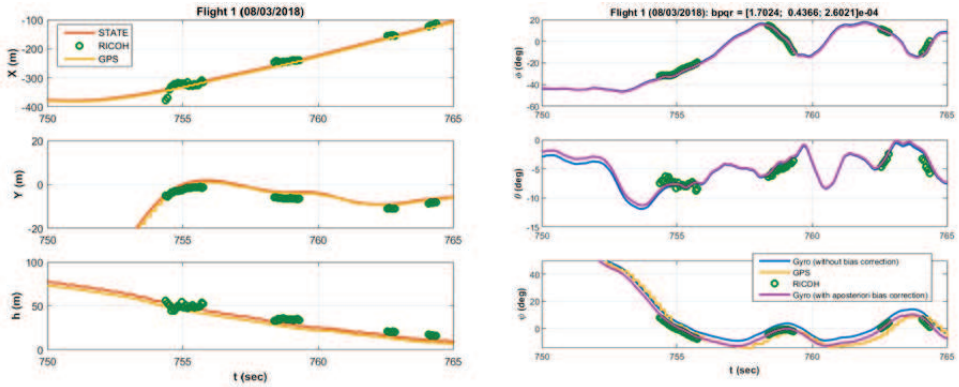


Fig. 14 6D camera pose estimation on the stereo-image data recorded on th K50: (left) position estimation vs. GPS data, (right) attitude estimation vs. INS data

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 filed 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 [21] 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 than 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.

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 [22] 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 [23], and a line-based method [24] where the authors assume that the detection of lines is more robust than that of points. Instead of the 6D pose the visual features can be also used directly in the navigation (tight-coupling). We investigate both possibilities.



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

V. Conclusion

This paper presented the vision-integrated navigation system for the aircraft final approach based on the ESKF framework which incooporates 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 its functionality to maintain the localization precision in case of sensor degradation so that the aircraft can continue the approach procedure was validated.

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 developped and integrated on the K50 UAV experiment platform. First flight test campaign with these vision systems have started for image data recording, and the development and evaluation of the image processing algorithms for runway detection is ongoing. The proposed navigation system will be implemented onboard the K50 shortly. We plan to include in the

final paper some open-loop flight test results of the vision-based aircraft pose estimation relative to the runway during the final approach phase.

In parallel, even in the first simulation validation, we see a strong need of the fault detection and exclusion function to provide the non-deviated navigation solution in case of possible sensor failures. Future perspectives include the development of an AAIM function integrated in the navigation system.

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

[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. Pflimlin, 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 Cnference 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] EuroControl, RNAV Approaches, 2012.

[17] N. Trawny and S.I. Roumeliotis, Indirect Kalman Filter for 3D Attitude Estimation, Technical Report 2005-002, University of Minnesota, 2005.

[18] 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.

[19] J. Sola, Quaternion kinematics for the error-state Kalman filter, CNRS/LAAS Technical notes, 2017.

[20] 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.

[21] 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.

[22] F. Li, D.Q. Tang and N. Shen, Vision-based Pose Estimation of UAV from Line Correspondences, Elsevier Procedia Engineering, vol.15, 2011.

[23] 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.

[24] 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.