International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:8, No:4, 2014

Navigation and Guidance System Architectures for Small Unmanned Aircraft Applications

Roberto Sabatini, Celia Bartel, Anish Kaharkar, Tesheen Shaid, Subramanian Ramasamy

***Abstract*—**Two multisensor system architectures for navigation and guidance of small Unmanned Aircraft (UA) are presented and compared. The main objective of our research is to design a compact, light and relatively inexpensive system capable of providing the required navigation performance in all phases of flight of small UA, with a special focus on precision approach and landing, where Vision Based Navigation (VBN) techniques can be fully exploited in a multisensor integrated architecture. Various existing techniques for VBN are compared and the Appearance-Based Navigation (ABN) approach is selected for implementation. Feature extraction and optical flow techniques are employed to estimate flight parameters such as roll angle, pitch angle, deviation from the runway centreline and body rates. Additionally, we address the possible synergies of VBN, Global Navigation Satellite System (GNSS) and MEMS-IMU (Micro-Electromechanical System Inertial Measurement Unit) sensors, and the use of Aircraft Dynamics Model (ADM) to provide additional information suitable to compensate for the shortcomings of VBN and MEMS-IMU sensors in high-dynamics attitude determination tasks. An Extended Kalman Filter (EKF) is developed to fuse the information provided by the different sensors and to provide estimates of position, velocity and attitude of the UA platform in real-time. The key mathematical models describing the two architectures i.e., VBN-IMU-GNSS (VIG) system and VIG-ADM (VIGA) system are introduced. The first architecture uses VBN and GNSS to augment the MEMS-IMU. The second mode also includes the ADM to provide augmentation of the attitude channel. Simulation of these two modes is carried out and the performances of the two schemes are compared in a small UA integration scheme (i.e., AEROSONDE UA platform) exploring a representative cross-section of this UA operational flight envelope, including high dynamics manoeuvres and CAT-I to CAT-III precision approach tasks. Simulation of the first system architecture (i.e., VIG system) shows that the integrated system can reach position, velocity and attitude accuracies compatible with the Required Navigation Performance (RNP) requirements. Simulation of the VIGA system also shows promising results since the achieved attitude accuracy is higher using the VBN-IMU-ADM than using VBN-IMU only. A comparison of VIG and VIGA system is also performed and it shows that the position and attitude accuracy of the proposed VIG and VIGA systems are both compatible with the RNP specified in the various UA flight phases, including precision approach down to CAT-II.

Roberto Sabatini is an Associate Professor in the School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, VIC 3053, Australia. (Corresponding author: phone: +61 (0) 3 9925 8015; www.rmit.edu.au e-mail: roberto.sabatini@rmit.edu.au)

Celia Bartel, Anish Kaharkar, and Tesheen Shaid were postgraduate students with the Department of Aerospace Engineering, Cranfield University, Cranfield, Bedford, MK 43 0AL, UK.

Subramanian Ramasamy is a doctoral research student in the School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, VIC 3053, Australia.

***Keywords*—**Global Navigation Satellite System (GNSS), Low-cost Navigation Sensors, MEMS Inertial Measurement Unit (IMU), Unmanned Aerial Vehicle, Vision Based Navigation.

I. Introduction

CIVIL and military applications of Unmanned Aircraft (UA) have much expanded in recent years as these vehicles provide cost-effective and safe alternatives to manned aircraft in several operational scenarios. In particular, small UA have the ability of performing tasks with higher manoeuvrability, longer endurance and, additionally pose less risk to human lives and nature. UA employ a variety of sensors, as well as multisensor data fusion algorithms, to provide autonomy to the platform in the accomplishment of mission- and safety-critical tasks. Technical requirements for air navigation systems primarily include accuracy, physical characteristics such as weight and volume, support requirements such as electrical power, and system integrity. One of the most important concepts is to use a multisensor integrated system to cope with the requirements of long/medium range navigation and landing. This would reduce cost, weight/volume and support requirements and, with the appropriate sensors and integration architecture, give increased accuracy and integrity of the overall system. The best candidates for such integration are indeed satellite navigation receivers and inertial sensors. In recent years, computer vision and Vision-Based Navigation (VBN) systems have started to be applied to UA. VBN can provide a self-contained autonomous navigation solution and can be used as an alternative (or an addition) to the traditional sensors including Global Navigation Satellite Systems (GNSS), Micro Electro Mechanical Systems (MEMS) based Inertial Measurement Units (IMUs) and GNSS/IMU integrated sensors. The required information to perform autonomous navigation can be obtained from cameras, which are compact and lightweight sensors. This is particularly attractive in UA platforms, where weight and volume are tightly constrained. A model-based approach is used to develop a system which processes image sequences from visual sensors fused with readings from GNSS/IMU to update a coarse, inaccurate 3D model of the surroundings [1]. Digital Elevation Models (DEM) were used to build the 3D model of the environment. Occupancy grid mapping was used in this study, in which the maps were divided into cells. Each cell had a probability value of an obstacle being present associated with it. Using this 'risk map' and the images from the visual sensors, the UA was able to update its stored virtual map. Shortest path optimization techniques based on the Djikstra algorithm and dynamic

programming were then used to perform obstacle avoidance and online computation of the shortest trajectory to the destination. A system which deals with vision-based SLAM using a trinocular stereo system was proposed in [2]. In this study, Scale-Invariant Feature Transform (SIFT) was used for tracking natural landmarks and to build the 3D maps. The algorithm built submaps from multiple frames which were then merged together. The SIFT features detected in the current frame were matched to the pre-built database map in order to obtain the location of the vehicle. With the development of deep space exploration, VBN has been used to provide an autonomous navigation system for rovers operating on the lunar surface and on Mars. The use of GNSS on the Moon is not possible as its signals cannot reach the lunar surface. Sonar and magnetic sensors cannot be used due to the absence of an atmosphere and magnetic field; while dead reckoning sensors such as IMU suffer from error accumulation. Stereo vision-based navigation has been used for navigating these space rovers because of its low power consumption and high reliability. The stereo vision system provides motion estimation for autonomous localization techniques and environmental information for obstacle avoidance and DEM construction. A stereo vision-based navigation system was implemented on the 2004 twin Mars Exploration Rovers, Spirit and Opportunity. They were equipped with three types of stereo cameras which allowed them to navigate the Martian surface for long distances. A stereo vision-based navigation system for lunar rovers was proposed in [3]. The system performed robust motion estimation and disparity estimation using stereo images. This enabled accurate lunar navigation, obstacle avoidance and DEM reconstruction. The position of the rover while descending was found using sequence images taken by a descent camera and the reading were integrated with measurements from a laser altimeter and DEM. Image processing was carried out using feature detection, tracking and stereo matching. Levenberg-Marquardt non-linear estimation was used for motion estimation and Weighted Zero Sum of Squared Differences gave the disparity estimation. A representation of the visual route taken by robots was used in appearance-based navigation [4]. This approach was called the View-Sequenced Route Representation (VSRR) and was a sequence of images memorized in the recording run along the required route. The visual route connected the initial position and destination via a set of images. This visual route was used for localization and guidance in the autonomous run. Pattern recognition was achieved by matching the features detected in the current view of the camera with the stored images. Criteria for image capture during the learning stage were given in the study. The visual route was learnt while the robot was manually guided along the required trajectory. A matching error between the previous stored image and current view was used to control the capture of the next key image. The current view was captured and saved as the next key image when a pre-set error threshold was exceeded. Localization was carried out at the start of the autonomous run by comparing the current view with the saved visual route images. The key image with the greatest similarity to the current view represented the start of the visual route. The location of the robot depended purely on the key image used and no assumption was made of its location in 3D space. During the autonomous run, the matching error between the current view and key images was monitored in order to identify which image should be used for guidance. The robot was controlled so as to move from one image location to another and finally to reach its destination. This 'teach-and-replay' approach was adopted in [5]-[8]. In the case of [5], [6], a single camera and natural landmarks were used to navigate a quadrotor UA along the visual route. The key images were considered as waypoints to be followed in the sensor space. Zero normalised cross correlation was used for feature matching between the current view and the key images. A control system using the dynamic model of the UA was developed. Its main task was to reduce the position error between the current view and key image to zero and to stabilize and control the UA. Vision algorithms to measure the attitude of a UA using the horizon and runway were presented in [9], [10]. The horizon is used by human pilots to control the pitch and roll of the aircraft while operating under visual flying rules. A similar concept is used by computer vision to provide an intuitive means of determining the attitude of an aircraft. This process is called Horizon-Based Attitude Estimation (HBAE). In [9], grayscale images were used for image processing. The horizon was assumed to be a straight line and appeared as an edge in the image. Texture energy method was used to detect it and this was used to compute the bank and pitch angle of the UA. The position of the UA with respect to the runway was found by computing the angles of the runway boundary lines. A Canny Edge detector was applied to part of the image below the horizon. The gradient of the edges was computed using the pixel coordinates, which gave a rough indication of where the UA was situated with respect to the runway. A similar approach to develop algorithms to compute the attitude and attitude rates is used in [10]. A Sobel edge detector was applied to each channel of the Red Green Blue (RGB) image. The three channels were then combined and Hough transform was used to detect the horizon. In this research, it was assumed that the camera frame and the body frames were coincidental and equations were developed so as to calculate the pitch and roll angle. The angular rates of the UA were derived using optical flow of the horizon. Optical flow gives us additional information of the states of the UA and is dependent on the angular rates, velocity and the distance of the detected features. During this research, it was observed that the image processing frontend was susceptible to false detection of the horizon if any other strong edges were present in the image. Therefore, an Extended Kalman Filter (EKF) was implemented to filter out these incorrect results. The performance of the algorithms was tested via test flights with a small UA and a Cessna 172. Results of the test flight with the UA showed an error in the calculated pitch and roll with standard deviations of 0.42 and 0.71 degrees respectively. Moving forward from these results, in our research we designed and tested a new VBN sensor specifically tailored for

approach/landing applications which, in addition to horizon detection and image-flow, also employed runway features extraction during the approach phase. MEMS-IMUs are low-cost and low-volume/weight sensors particularly well suited for small/medium size UA applications. However, their integration represents a challenge, which needs to be addressed either by finding improvements to the existing analytical methods or by developing novel algorithmic approaches that counterbalance the use of less accurate inertial sensors. In line with the above discussions, the main objective of our research is to develop a low-cost and low-weight/volume Navigation and Guidance System (NGS) based on VBN and other low-cost and low-weight/volume sensors, capable of providing the required level of performance in all flight phases of a small/medium size UA, with a special focus on precision approach and landing (i.e., the most demanding and potentially safety-critical flight phase), where VBN techniques can be fully exploited in a multisensor integrated architecture. The NGS is implemented using an existing controller that employs Fuzzy logic and Proportional-Integral-Differential (PID) technology.

II. VBN Sensor Design, Development and Test

As discussed above, VBN techniques use optical sensors (visual or infrared cameras) to extract visual features from images which are then used for localization in the surrounding environment. Cameras have evolved as attractive sensors as they help design economically viable systems with simpler hardware and software components. Computer vision has played an important role in the development of UA [11]. Considerable work has been made over the past decade in the area of vision-based techniques for navigation and control [9]. UA vision-based systems have been developed for various applications ranging from autonomous landing to obstacle avoidance. Other applications looked into the possible augmentation IMU and GNSS/IMU by using VBN measurements [12]. As discussed above, several VBN sensors and techniques have been developed. However, the vast majority of VBN sensor schemes fall into one of the following two categories [13]: Model-based Approach (MBA) and Appearance-based Approach (ABA). MBA uses feature tracking in images and creates a 3D model of the workspace in which robots or UA operates [14]. The 3D maps are created in an offline process using a priori information of the environment. Localisation is carried out using feature matching between the current view of the camera and the stored 3D model. The orientation of the robot is found from 3D-2D correspondence. MBA has been extensively researched in the past and is the most common technique currently implemented for VBN. However, the accuracy of this method is dependent on the features used for tracking, robustness of the feature descriptors and the algorithms used for matching and reconstruction. The reconstruction in turn relies on proper camera calibration and sensor noise. Knowledge of the surroundings so as to develop the 3D models is also required prior to implementation which may not be case in most situations. ABA algorithms eliminate the need for a metric model as they work directly in the sensor space. This approach utilizes the appearance of the whole scene in the image, contrary to MBA, which uses distinct objects such as landmarks or edges [4]. The environment is represented in the form of key images taken at various locations using the visual sensors. This continuous set of images describes the path to be followed by the robot. The images are captured while manually guiding the robot through the workspace. In this approach, localisation is carried out by finding the key image with the most similarity to the current view. The robot is controlled by either coding the action required to move from one key image to another or by a more robust approach using visual serving [8]-[15]. The ABA approach is relatively new and has gained active interest. The modelling of the surrounding using a set of key images is more straightforward to implement compared to 3D modelling. A major drawback of this method is its limited applicability. The robot assumes that the key image database for a particular workspace is already stored in its memory. Therefore, the key images need to be recaptured each time the robot moves to a new workspace. It is limited to work in the explored regions which have been visualised during the learning stage [16]. The ABA approach has a disadvantage in requiring a large amount of memory to store the images and is computationally more costly than MBA. However, due to improvements in computer technology, this technique has become a viable solution in many application areas. We selected the ABA approach for the design of our VBN sensor system.

A. Learning Stage

The first step required for appearance based navigation is the learning stage. During this stage, a video is recorded using the on-board camera while guiding the aircraft manually during the landing phase. The recorded video is composed of a series of frames which form the visual route for landing. This series of frames is essentially a set of images connecting the initial and target location images. The key frames are first sampled and the selected images are stored in the memory to be used for guidance during autonomous landing of the aircraft. During the learning stage, the UA is flown manually meeting the Required Navigation Performance (RNP) requirements of precision approach and landing. If available, Instrument Landing System (ILS) can also be used for guidance. It should be noted that the visual route captured while landing on a runway, can only be used for that particular runway. If the UA needs to land at multiple runways according to its mission, the visual route for all the runways is required to be stored in the memory. The following two methods can be employed for image capture during the learning stage.

- Method 1: Frames are captured from the video input at fixed time intervals. The key frames are selected manually in this case.
- Method 2: Frames are captured using a matching difference threshold [4]. This matching difference threshold is defined in number of pixels and can be

obtained by tracking the features in the current view and the previously stored key image. The key images can then be selected based on the threshold and stored in the memory.

The algorithm starts by taking an image at the starting point. Let this image be captured at location i. As the aircraft moves forward, the difference between the current view (V) and the image M_i increases. This difference keeps increasing until it reaches the set threshold value (x). At this point, a new image $M_{i+1}$is taken (replacing the previous image M_i) and the process is repeated until the aircraft reaches its destination. The image capture process is summarised in Fig. 1.

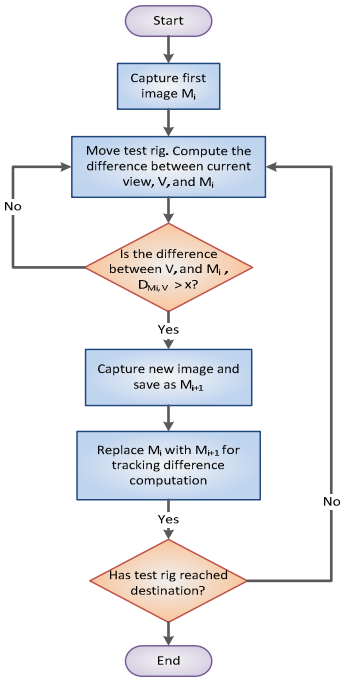

Fig. 1 Image capture process

B. Localisation

Localisation is a process which determines the current location of the aircraft at the start of autonomous run. This process identifies the key image which is the closest match to the current view. The current view of the aircraft is compared with a certain number of images, preferably the ones at the start of the visual route. The key image with the least matching difference is considered to be the start of the visual route to be followed by the UA. At the start of the autonomous run, the UA is approximately at the starting position of the visual route. The current view, captured from the on-board camera is compared with a set of images (stored previously in the memory during the learning stage) in order to find the location of aircraft with respect to the visual route. The key image with the least matching difference with the current view is considered to be the location of the UA and marks the start of the visual route to be followed. The localisation process is summarised in Fig. 2.

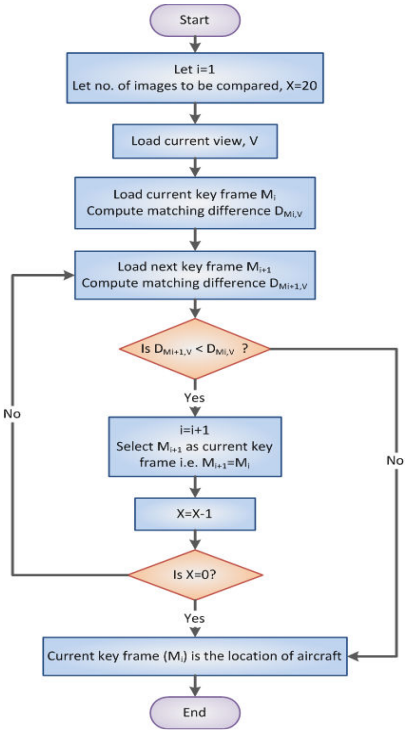

Fig. 2 Localisation process

In this example, the number of images to be compared (X) is taken as 20. First, the algorithm loads the current view (V) and the first key frame (M_i). Then the difference between the current view and the current key frame is computed. The algorithm then loads the next key frame M_{i+1} and again computes the difference with the current view. If this difference is less than the previous difference, M_{i+1} replaces M_i, and the process is repeated again. Otherwise, M_i is considered as the current location of the aircraft.

C. Autonomous Run

During the autonomous run phase, the UA follows the visual route (previously stored in memory during the learning stage) from the image identified as the current location of the aircraft during localisation. The set of key images stored as the visual route can be considered as the target waypoints for the aircraft in sensor space. The current view is compared to the key images so as to perform visual servoing. The approach followed to identify the key image to be used for visual servoing, is describes as follows. Let M_j be the current key frame, i.e. image with the least matching difference with the current view. During the autonomous run, the current key image and the next key image (M_{j+1}) are loaded. The matching differences of the current view V with M_j and M_{j+1} (which are $D_{Mj,V}$ and $D_{Mj+1,V}$ respectively) are tracked. When the matching difference $D_{Mj,V}$ exceeds $D_{Mj+1,V}$, M_{j+1} is taken as the current key image replacing M_j and the next key image is loaded as M_{j+1}. This process keeps repeating until the aircraft reaches its destination, that is the final key frame. Fig. 3 summarises the process of autonomous run in the form of a

flow chart while the change in matching difference for different key frames during autonomous run is presented in Fig. 4.

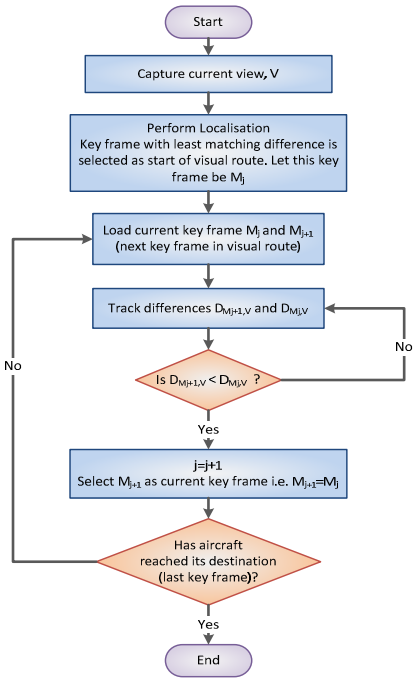

Fig. 3 Autonomous run

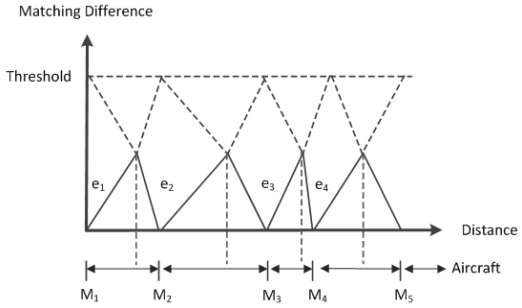

Fig. 4 Matching difference/key frame selection process

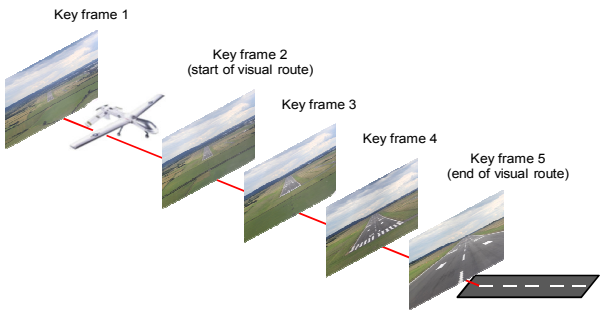

Fig. 5 VBN process

The proposed VBN process is depicted in Fig. 5. The key frames represent the visual route the UA requires to follow. The figure shows that the key frame 2 is identified as the starting point of the visual route during the localisation process.

The onboard computer tracks the matching difference between current view and the second and third key frames until the difference for key frame 2 and the current view exceeds the difference of key frame 3 and the current view. At this stage, key image 3 is used to control the UA and the matching differences between key frames 3, 4 and the current view are monitored. This process is repeated until the UA reaches its destination. To capture the outside view, a monochrome Flea camera from Point Grey Research was used. The main specification of the camera and lenses are listed in Table I. This camera was also used in a previous study on stereo vision [17] and was selected for this project. The Flea camera is a pinhole Charged Coupled Device (CCD) camera with a maximum image resolution of 1024 x 768 pixels. It is capable of recording videos at a maximum rate of 30 fps. An IEEE 1394 connection was used to interface the camera and computer with a data transfer speed of 400 Mbps.

D. Image Processing Module

The Image Processing Module (IPM) of the VBN system detects horizon and runway centreline from the images and computes the aircraft attitude, body rates and deviation from the runway centreline. Fig. 6 shows the functional architecture of the IPM. As a first step, the size of the image is reduced from 1024x768 pixels to 512x384 pixels. After some trials, it was found that this size reduction speeds up the processing without significantly affecting the features detection process. The features such as the horizon and the runway centreline are extracted from the images for attitude computation. The horizon is detected in the image by using Canny edge detector while the runway centreline is identified with the help of Hough Transform. The features are extracted from both, the current viewand the current key frame. The roll and pitch are computed from the detected horizon while the runway centreline in used to compute the deviation of aircraft from the runway centreline. Then the roll and pitch difference are computed between the current view and the current key frame. Optical flow is determined for all the points on the detected horizon line in the images. The aircraft body rates are then computed based on the optical flow values. The IPM provides the aircraft attitude, body rates, pitch and roll differences between current view and key frame, and deviation from the runway centreline.

TABLE I
POINT GREY FLEA AND LENSES SPECIFICATIONS

Sensor type	Sony ICX204AQ/AL 1/3" CCD sensor
Scan type	Progressive
Resolution	1024x768 BW
Format	8-bit or 16-bit, 12-bit AtoD
Pixel size	4.65µm x 4.65µm
Frame rates	1.875, 3.75, 7.5, 15, 30fps
Video output signal	8 bits per pixel / 12 bits per pixel digital data
Interfaces	6-pin IEEE-1394 for camera control and video data transmission 4 general purpose digital input/output pins
Voltage requirements	8-32V
Power consumption	< 3W
Gain	Automatic/Manual modes at 0.035dB resolution (0 to 24dB)
Shutter	Automatic/Manual/Extended Shutter modes (20µs to66ms @ 15Hz)
Trigger modes	DCAM v1.31 Trigger Modes 0, 1 and 3
SNR	50dB or better at minimum gain
Camera dimensions (no lenses)	30mm x 31mm x 29mm
Mass	60g without optics
Operating temperature	0° to 45°C
Focal length	3.5- 8.0 mm
Max CCD format	1/3"
Aperture	F1.4 – 16 (closed) - Manual control
Maximum Field of View (FOV)	Horizontal: 77.6°/Vertical: 57.6°
Min working distance	0.4m
Lenses dimensions	34.0 mm diameter x 43.5 mm length

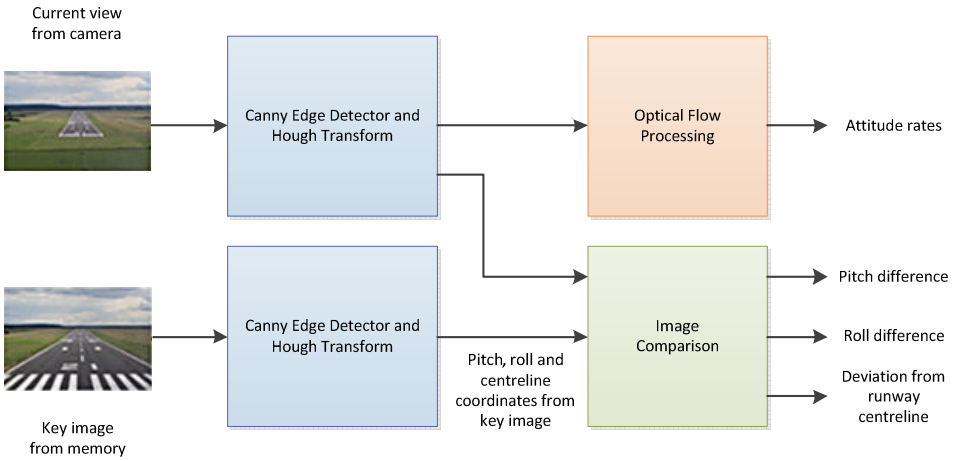

Fig. 6 Functional architecture of the IPM

The attitude of the aircraft is computed based on the detected horizon and the runway. The algorithm calculates the pitch and roll of the aircraft using the horizon information while aircraft deviation from the runway centreline is computed using the location of runway centreline in the current image. The detailed processing performed by the IPM is illustrated in Fig. 7.

E. Mathematical Model

Fig. 8 shows the relationship between the body (aircraft) frame (O_b, X_b, Y_b, Z_b), camera frame (O_c, X_c, Y_c, Z_c) and the Earth frame coordinates (O_w, X_w, Y_w, Z_w).

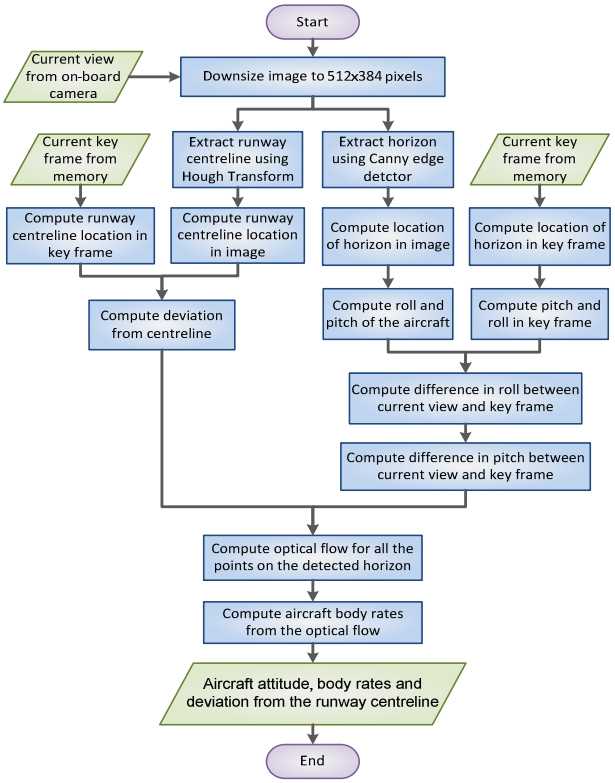

Fig. 7 Image processing module flowchart

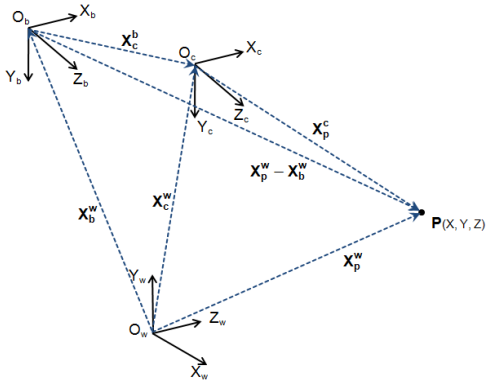

Fig. 8 Coordinate system

The position of a 3D space point P in Earth coordinates is represented by a vector X_p^w with components x_p, y_p and z_p in the Earth frame. The position of aircraft centre with respect to the Earth coordinates is represented by the vector X_b^w with components x_b, y_b and z_b in the Earth frame. The vector X_p^c represents the position of the point P with respect to the camera frame with components x_{cp}, y_{cp} and z_{cp} in the camera frame. The position of centre of camera lens with respect to the body frame is represented by the vector X_c^b. The vector X_c^w represents the position of lens centre with respect to the ground frame with components x_c, y_c and z_c in the ground frame. The position of point P with respect to body frame with components in the Earth frame can be computed as X_p^w-X_b^w. For representation purposes, we adopt the following notation; C as cos and S as sin The transformation matrix from Earth frame to body frame C_w^b is obtained in terms of the yaw ψ, pitch θ, and roll angle ϕ as:

$$C_w^b = \begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ -C\phi S\psi + S\phi S\theta C\psi & C\phi C\psi + S\phi S\theta S\psi & S\phi C\theta \\ S\phi S\psi + C\phi S\theta C\psi & -S\phi C\psi + C\phi S\theta S\psi & C\phi C\theta \end{bmatrix} \quad (1)$$

The position of point P with respect to the aircraft's body with components in the body frame can be obtained as:

$$X_p^b = C_w^b\left(X_p^w - X_b^w\right) \quad (2)$$

The position of point P with respect to the camera frame can also be found in a similar way as:

$$X_p^c = C_b^c\left(X_p^b - X_c^b\right) = C_b^c C_w^b\left(X_p^w - X_b^w\right) - C_b^c X_c^b \quad (3)$$

where C_b^c is the constant transformation matrix from the body frame to the camera frame. With the assumption that the camera is fixed with respect to the body and the angle from the camera optical axis to the longitudinal axis of the aircraft is fixed value, $-C_b^c X_c^b$ is a known constant vector with components k_x, k_y and k_z in the camera frame. In this case, the velocity and rotation rate of aircraft are the same as those of the camera. Thus, the position and attitude of the aircraft can be easily computed according to those of the camera as:

$$X_b^w = X_c^w - C_c^w C_b^c X_c^b = X_c^w - C_c^w[k_x, k_y, k_z]^T \quad (4)$$

$$\phi = \phi_c \quad (5)$$

$$\theta = \theta_c - \theta_0 \quad (6)$$

$$\psi = \psi_c \quad (7)$$

where ϕ_c is the roll, θ_c is the pitch, ψ_c is the yaw and θ_0 is the angle of incidence of the camera. The transformation matrix from camera frame to the ground frame, represented by C_c^w, is obtained from:

$$C_w^b = \begin{bmatrix} C\theta_c C\psi_c & -C\phi_c S\psi_c + S\phi_c S\theta_c C\psi_c & S\phi_c S\psi_c + C\phi_c S\theta_c C\psi_c \\ C\theta_c S\psi_c & C\phi_c C\psi_c + S\phi_c S\theta_c S\psi_c & -S\phi_c C\psi_c + C\phi_c S\theta_c S\psi_c \\ -S\theta_c & S\phi_c C\theta_c & C\phi_c C\theta_c \end{bmatrix} \quad (8)$$

$$C_w^c = [C_c^w]^T \quad (9)$$

Equation (9) represents the transformation matrix from the Earth frame to the camera frame coordinates [2]. From now onwards, only the state estimates of the camera are considered. The position of 3D point P with respect to camera frame is given by:

$$X_p^c = C_w^c\left(X_p^w - X_c^w\right) \quad (10)$$

with components in the camera frame given by:

$$\begin{cases} X = (-\mathrm{C}\phi_c \mathrm{S}\psi_c + \mathrm{S}\phi_c \mathrm{S}\theta_c \mathrm{C}\psi_c)(x_p - x_c) \\ \quad +(\mathrm{S}\phi_c \mathrm{C}\theta_c)(z_p - z_c) \\ \quad +(\mathrm{C}\phi_c \mathrm{C}\psi_c + \mathrm{S}\phi_c \mathrm{S}\theta_c \mathrm{S}\psi_c)(y_p - y_c) \\ Y = (\mathrm{S}\phi_c \mathrm{S}\psi_c + \mathrm{C}\phi_c \mathrm{S}\theta_c \mathrm{C}\psi_c)(x_p - x_c) \\ \quad +(\mathrm{C}\phi_c \mathrm{C}\theta_c)(z_p - z_c) \\ \quad +(-\mathrm{S}\phi_c \mathrm{C}\psi_c + \mathrm{C}\phi_c \mathrm{S}\theta_c \mathrm{S}\psi_c)(y_p - y_c) \\ Z = (\mathrm{C}\theta_c \mathrm{C}\psi_c)(x_p - x_c) + (\mathrm{C}\theta_c \mathrm{S}\psi_c)(z_p - z_c) \\ \quad +(-\mathrm{S}\theta_c)(y_p - y_c) \end{cases} \quad (11)$$

Then, the coordinates (u, v) of P in the image plane is obtained from:

$$u = \frac{fX}{Z} \quad (12)$$

$$v = \frac{fY}{Z} \quad (13)$$

Using the coordinate previously defined, the point P is assumed to be located on the detected horizon line. As the Earth's surface is approximated by a plane, a normal vector to the plane, n_w is described as:

$$n_w = [0 \quad 1 \quad 0]^T \quad (14)$$

If the horizon line is described by a point X_p^w and a direction vector $l_w$tangential to the line of horizon visible to the image plane, then:

$$X_p^w = [x \quad 0 \quad d]^T \quad (15)$$

$$l_w = [1 \quad 0 \quad 0]^T \quad (16)$$

where x is an arbitrary point along x-axis and d is the distance to horizon along z-axis. If the camera is assumed to be placed directly above the origin of the ground frame, the position of camera X_c^w is described as:

$$X_c^w = [0 \quad h \quad 0]^T \quad (17)$$

Then, a point on horizon may be expressed as:

$$X_p^w = X_p^c + X_c^w \quad (18)$$

The horizon projection on the image plane can be described by the point p and a direction vector m as:

$$m = \begin{bmatrix} m_x & m_y & 0 \end{bmatrix}^T \quad (19)$$

$$p = [u \quad v \quad f]^T \quad (20)$$

where (m_y/m_x) gives the gradient of the horizon line. As the position of the horizon X_p^w lies on the surface of the ground plane, therefore:

$$n_w \cdot X_p^w = 0 \quad (21)$$

Substituting X_p^w gives:

$$n_w \cdot (X_p^c + X_c^w) = 0 \quad (22)$$

The direction vector of the horizon line l_w lies on the plane and is therefore orthogonal to the normal vector. Therefore:

$$n_w \cdot l_w = 0 \quad (23)$$

Equations (22) and (23) are in form known as line plane correspondence problem. Recalling the equations for a projective perspective camera:

$$\begin{bmatrix} u \\ v \\ f \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (24)$$

Substituting (24) into (22), roll angle can be derived as [6]:

$$\phi = \tan^{-1}\left(\frac{-m_y}{m_x}\right) \quad (25)$$

which is an intuitive result that roll angle is dependent on the gradient of the horizon line on the image plane. Similarly, it can be shown that substituting (20) into (22), the pitch angle can be derived as [6]:

$$\theta = \tan^{-1}\left(\pm\frac{hf + ud\sin\phi + vd\cos\phi}{df - uh\sin\phi - vh\cos\phi}\right) \quad (26)$$

If the distance to the horizon is much greater than the height of the aircraft (i.e., $d \gg h$), the expression for pitch reduces to the following:

$$\theta = \tan^{-1}\left(\pm\frac{u\sin\phi + v\cos\phi}{f}\right) \quad (27)$$

which shows the pitch dependency on roll angle, focal length and position of the horizon in the image plane. Optical flow depends on the velocity of the platform, angular rates of the platform and the distance to any features observed [6]. Differentiating (24) we obtain:

$$\dot{u} = \frac{f(\dot{X}Z - \dot{Z}X)}{Z^2} = \frac{f}{Z}\left(\dot{X} - \frac{X}{Z}\dot{Z}\right) \quad (28)$$

$$\dot{v} = \frac{f(\dot{Y}Z - \dot{Z}Y)}{Z^2} = \frac{f}{Z}\left(\dot{Y} - \frac{Y}{Z}\dot{Z}\right) \quad (29)$$

Substituting (28) into the time derivative of (18) yields the classical optical flow equations [6]:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} 1 & 0 & \frac{-u}{f} \\ 0 & 1 & \frac{-v}{f} \end{bmatrix}\begin{bmatrix} \dot{r}_x^w \\ \dot{r}_y^w \\ \dot{r}_z^w \end{bmatrix} + \begin{bmatrix} \frac{uv}{f} & -\left(f + \frac{u^2}{f}\right) & v \\ \left(f + \frac{v^2}{f}\right) & -\frac{uv}{f} & -u \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (30)$$

If the observed point lies on the horizon, then Z will be large and the translational component will be negligible. In this case, (30) reduces to:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix}=\begin{bmatrix}\frac{uv}{f} & -(f+\frac{u^2}{f}) & v\\ (f+\frac{v^2}{f}) & -\frac{uv}{f} & -u\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \quad (31)$$

To minimize the effect of errors, a Kalman filter is employed. The state vector consists of the roll angle, pitch angle and body rates of the aircraft. It is assumed that the motion model of the aircraft is disturbed by uncorrelated zero-mean Gaussian noise. The EKF measurement model is defined as:

$$z_k = H_k * x_k + v_k \quad (32)$$

where:
z_k is the measurement vector
H_k is the design matrix
x_k is the state vector
v_k is the measurement noise
k is the k[th] epoch of time, t_k

$$x_{k+1} = \Phi_k * x_k + G_k * w_k \quad (33)$$

where:
x_{k+1} is thestate vector at epoch $k+1$
Φ_k is the state transition matrix from epoch k to $k+1$
G_k is the shaping matrix
w_k is the process noise

The EKF comprises of prediction and update algorithms. The prediction algorithm estimates the state vector and computes the corresponding covariance matrix P_k from the current epoch to the next one using the state transition matrix characterizing the process model described by:

$$P_{k+1}^- = \Phi_{k+1} P_k^+ \Phi_{k+1}^T + Q_k \quad (34)$$

where P_{k+1}^- represents a predicted value computed by the prediction equations and P_k^+ refers to updated values obtained after the correction equations. The process noise at a certain epoch k is characterized by a covariance matrix, Q_k. The updating equations correct the predicted state vector and the corresponding covariance matrix using the measurement model as follows:

$$x_{k+1}^+ = K_{k+1} u_{k+1} \quad (35)$$

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} H_{k+1} P_{k+1}^- \quad (36)$$

where $K_{k+1}$is the Kalman gain matrix at epoch, $k+1$ and u_{k+1} is the innovation vector at epoch, $k+1$. The innovation vector represents the difference between the current measurement and the predicted measurement and can be described as:

$$u_{k+1} = z_{k+1} - H_{k+1} x_{k+1}^- \quad (37)$$

The Kalman gain is used to quantify the influence of new information present in the innovation vector on the estimation of the state vector and can be considered as a weight factor. It is basically equal to the ratio of the uncertainty on the current measurement and the uncertainty on the predicted one. This gain is defined by:

$$K_{k+1} = P_{k+1}^- H_{k+1}^T [H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1}]^{-1} \quad (38)$$

where $R_{k+1}$is the measurement noise covariance matrix. For the process model defined here, the state vector of the system composed of error in position, δr^n, velocity, δv^n and attitude, ϵ^n is described by:

$$\times = \begin{bmatrix}\delta r^n\\ \delta v^n\\ \epsilon^n\end{bmatrix} \quad (39)$$

The dynamic matrix of the system is expressed by:

$$F = \begin{bmatrix} F_{rr} & F_{rv} & 0\\ F_{vr} & F_{vv} & (f^n\times)\\ F_{er} & F_{ev} & -(\delta\omega_{in}^n\times)\end{bmatrix} \quad (40)$$

where:

$$(f^n\times) = \begin{bmatrix} 0 & -e_D & e_E\\ e_D & 0 & -e_N\\ -e_E & e_N & 0\end{bmatrix} \quad (41)$$

where e_D is the yaw error, e_N is the roll error, f^n is the specific force transformation matrix from the inertial frame to the navigation frame. ω_i^n is the angular rate transformation matrix from inertial frame to navigation frame defined w.r.t latitude, ϕ, altitude, h, velocity in east direction, v_E, velocity in north direction, v_N, angular rate error, ω_e, the radius of Earth is $R = \sqrt{r_M + r_T}$ and is given by:

$$-(\delta\omega_i^n\times) = [W \;\; Y \;\; Z] \quad (42)$$

where the matrixes W, Y and Z are given by:

$$W = \begin{bmatrix} 0\\ -\omega_e\, sin(\phi) - \frac{v_E * tan\phi}{r_T+h}\\ \frac{-v_N}{r_M+h}\end{bmatrix} \; Y = \begin{bmatrix}\omega_e\, sin(\phi) + \frac{v_E * tan\phi}{r_T+h}\\ 0\\ -\omega_e\, cos(\phi) - \frac{v_E}{r_T+h}\end{bmatrix} \; Z = \begin{bmatrix}\frac{v_N}{r_M+h}\\ \omega_e\, cos(\phi) + \frac{v_E}{r_T+h}\\ 0\end{bmatrix} \quad (43)$$

The augmented process model is obtained as follows:

$$x(t)=\begin{bmatrix}\dot{\delta r^n}\\ \dot{\delta v^n}\\ \dot{\epsilon^n}\\ \dot{\delta b_a}\\ \dot{\delta b_g}\\ \dot{\delta S_a}\\ \dot{\delta S_g}\end{bmatrix}=\begin{bmatrix} F_{rr} & F_{rv} & 0_3 & 0_3 & 0_3 & 0_3 & 0_3\\ F_{vr} & F_{vv} & (f^n\times) & C_b^n & 0_3 & C_b^nF^b & 0_3\\ F_{er} & F_{ev} & -(\delta\omega_{in}^n\times) & 0_3 & -C_b^n & 0_3 & C_b^nW^b\\ 0_3 & 0_3 & 0_3 & D_{ba} & 0_3 & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3 & D_{bg} & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & D_{Sa} & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & D_{Sg}\end{bmatrix}\begin{bmatrix}\delta r^n\\ \delta v^n\\ \epsilon^n\\ \delta b_a\\ \delta b_g\\ \delta S_a\\ \delta S_g\end{bmatrix}+\begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3\\ C_b^n & 0_3 & 0_3 & 0_3 & 0_3 & 0_3\\ 0_3 & -C_b^n & 0_3 & 0_3 & 0_3 & 0_3\\ 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3\end{bmatrix}\begin{bmatrix}\eta_a\\ \eta_g\\ \eta_{ba}\\ \eta_{bg}\\ \eta_{Sa}\\ \eta_{Sg}\end{bmatrix} \tag{44}$$

where:

$$D_{ba}=\begin{bmatrix}-1/\tau_{bax} & 0 & 0\\ 0 & -1/\tau_{bay} & 0\\ 0 & 0 & -1/\tau_{baz}\end{bmatrix}\qquad D_{bg}=\begin{bmatrix}-1/\tau_{bgx} & 0 & 0\\ 0 & -1/\tau_{bgy} & 0\\ 0 & 0 & -1/\tau_{bgz}\end{bmatrix} \tag{45}$$

$$D_{Sa}=\begin{bmatrix}-1/\tau_{Sax} & 0 & 0\\ 0 & -1/\tau_{Say} & 0\\ 0 & 0 & -1/\tau_{Saz}\end{bmatrix}\qquad D_{Sg}=\begin{bmatrix}-1/\tau_{Sgx} & 0 & 0\\ 0 & -1/\tau_{Sgy} & 0\\ 0 & 0 & -1/\tau_{Sgz}\end{bmatrix} \tag{46}$$

$$F^b=\begin{bmatrix} f_x & 0 & 0\\ 0 & f_y & 0\\ 0 & 0 & f_z\end{bmatrix}\qquad W^b=\begin{bmatrix}\omega_x & 0 & 0\\ 0 & \omega_y & 0\\ 0 & 0 & \omega_z\end{bmatrix} \tag{47}$$

where $\delta b_a$and δb_g are the accelerometers and gyroscopes biases. δS_a and δS_g are the scale factors of accelerometers and gyroscopes. τ_{ba} and τ_{bg} are the correlation times for accelerometers and gyroscopes. τ_{Sa} and τ_{Sg} are the correlation times for accelerometers and gyroscopes scale factors. C_b^n is the transformation matrix from the body frame to the navigation frame. The covariance matrix of the model is given by:

$$Q=\begin{bmatrix} q_a & 0 & 0 & 0 & 0 & 0\\ 0 & q_g & 0 & 0 & 0 & 0\\ 0 & 0 & q_{ba} & 0 & 0 & 0\\ 0 & 0 & 0 & q_{bg} & 0 & 0\\ 0 & 0 & 0 & 0 & q_{Sa} & 0\\ 0 & 0 & 0 & 0 & 0 & q_{Sg}\end{bmatrix} \tag{48}$$

The discrete process noise matrix is defined as follows:

$$Q_k\sim G(t_k)Q(t_k)G^T(t_k) \tag{49}$$

When the position errors are expressed in radians (latitude and longitude), the values are so meager that they can lead to numerical instabilities in the Kalman filter [18]. Then it is preferable to express them in NED coordinates [19]. For this, the position terms in the state vector $x(t)$ in (2) are transformed. Defining x' as the new state error vector, after being transformed by the matrix *T*, we obtain:

$$x'=Tx \tag{50}$$

$$x=T^{-1}x' \tag{51}$$

Substituting (50) and (51) in (33), we obtain:

$$\dot{x'}=\dot{T}x+T\dot{x} \tag{52}$$

$$\dot{x'}=\dot{T}T^{-1}x'+TFx=\left(\dot{T}T^{-1}+TFT^{-1}\right)x'+TGw \tag{53}$$

$$\dot{x'}=F'x'+G'w \tag{54}$$

where F' and G' are the new dynamics and shaping matrices defined by:

$$F'=\dot{T}T^{-1}+TFT^{-1}\approx TFT^{-1} \tag{55}$$

$$G'=T\,G \tag{56}$$

Herein, the state error vector x with position error δr^n expressed in terms of latitude and longitude (radians) is transformed into x' with position error δP^n expressed in meters as:

$$x=\begin{bmatrix}\delta r^n\\ \delta v^n\\ \epsilon^n\\ \delta b_a\\ \delta b_g\\ \delta S_a\\ \delta S_g\end{bmatrix};\qquad x'=\begin{bmatrix}\delta \mathrm{P^n}\\ (\delta \mathrm{v^n})'\\ (\epsilon^{\mathrm{n}})'\\ (\delta \mathrm{b_a})'\\ (\delta \mathrm{b_g})'\\ (\delta \mathrm{S_a})'\\ (\delta \mathrm{S_g})'\end{bmatrix} \tag{57}$$

where $\delta \mathrm{P^n}$ is the position error state, $\delta \mathrm{v^n}$ is the velocity error and ϵ^{n} is the attitude error term and

$$T=\begin{bmatrix}A & 0_{18*18}\\ 0_{18*18} & I_{18}\end{bmatrix} \tag{58}$$

$$A=\begin{bmatrix} r_M+h & 0 & 0\\ 0 & (r_T+h)cos\,(\varphi) & 0\\ 0 & 0 & -1\end{bmatrix} \tag{59}$$

If the body rates are assumed to be approximately constant during the sampling interval Δt and first order integration is applied, then the state transition equations are as follows:

$$\begin{bmatrix} \phi(k+1) \\ \theta(k+1) \\ \omega_x(k+1) \\ \omega_y(k+1) \\ \omega_z(k+1) \end{bmatrix} = \begin{bmatrix} \phi(k) + \Delta t(\dot{\phi}(k)) \\ \theta(k) + \Delta t(\dot{\theta}(k)) \\ \omega_x(k) \\ \omega_y(k) \\ \omega_z(k) \end{bmatrix} + \begin{bmatrix} \eta_\phi(k) \\ \eta_\theta(k) \\ \eta_{\omega_x}(k) \\ \eta_{\omega_y}(k) \\ \eta_{\omega_z}(k) \end{bmatrix} \quad (60)$$

where:

$$\dot{\phi}(k) = (\omega_x(k)\sin(\phi(k)) + \omega_y(k)\cos(\phi(k))\tan(\phi(k)) + \omega_z(k) \quad (61)$$

$$\dot{\theta}(k) = \omega_x(k)\cos(\phi(k)) - \omega_y(k)\sin(\phi(k)) \quad (62)$$

The measurement equations are comprised of direct observations of the pitch and the roll from the horizon and i optical flow observations on the detected horizon line. Therefore, the length of the measurement vector $z(k)$ is $2(i+1)$. The relation of measurement vector and the states is represented by following linear equations:

$$\begin{bmatrix} \phi(k) \\ \theta(k) \\ \dot{u}_1(k) \\ \dot{v}_1(k) \\ \dot{u}_2(k) \\ \dot{v}_2(k) \\ \cdots \\ \dot{u}_i(k) \\ \dot{v}_i(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{u_1 v_1}{f} & -\left(f + \frac{{u_1}^2}{f}\right) & v_1 \\ 0 & 0 & \left(f + \frac{{v_1}^2}{f}\right) & \frac{-u_1 v_1}{f} & -u_1 \\ 0 & 0 & \frac{u_2 v_2}{f} & -\left(f + \frac{{u_2}^2}{f}\right) & v_2 \\ 0 & 0 & \left(f + \frac{{v_2}^2}{f}\right) & \frac{-u_2 v_2}{f} & -u_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \frac{u_i v_i}{f} & -\left(f + \frac{{u_i}^2}{f}\right) & v_i \\ 0 & 0 & \left(f + \frac{{v_i}^2}{f}\right) & \frac{-u_i v_i}{f} & -u_i \end{bmatrix} \times \begin{bmatrix} \phi(k) \\ \theta(k) \\ \omega_x(k) \\ \omega_y(k) \\ \omega_z(k) \end{bmatrix} \quad (63)$$

F. VBN Sensor Performance

Based on various laboratory, ground and flight test activities with small aircraft and UA platforms, the performance of the VBN sensor were evaluated. Fig. 9 shows a sample image used for testing the VBN sensor algorithms and the results of the corresponding horizon detection process for attitude estimation purposes. The algorithm detects the horizon and the runway centreline from the images. The horizon is detected in the image by using Canny edge detector with a threshold of 0.9 and standard deviation of 50. In this experiment, the values of the threshold and the standard deviation were selected by hit-and-trial method. The resulting image after applying the Canny edge detector is a binary image. The algorithm assigns value '1' to the pixels detected as horizon while the rest of the pixels in the image are assigned value '0'. From this test image, the computed roll angle is 1.26° and the pitch angle is -10.17°. To detect the runway in the image, kernel filter and Hough Transform are employed. The runway detected from the same test image is shown in Fig. 10. For this image, the location of the runway centreline was computed in pixels as 261. The features were extracted from both the current viewand the current key frame. After the pitch, roll and centreline values were determined, the roll/pitch differences and the deviation from centreline are computed between the current view and the current key frame.

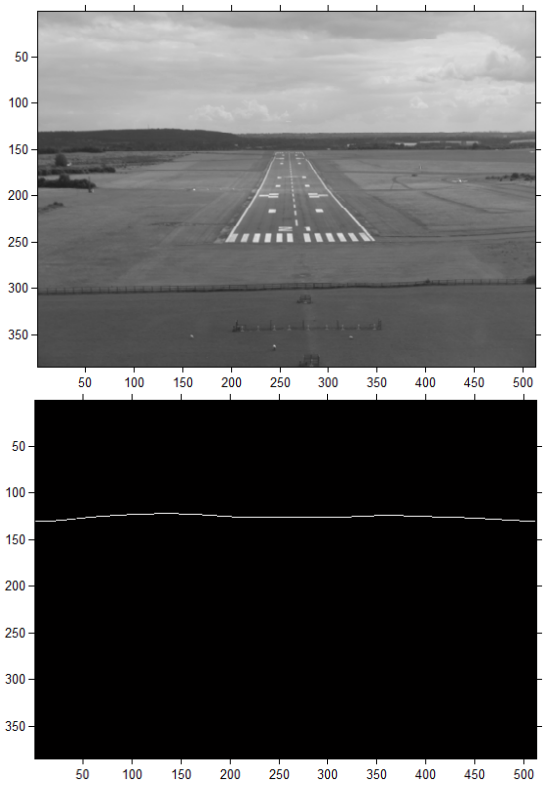

Fig. 9 Horizon detected from the test image (landing phase)

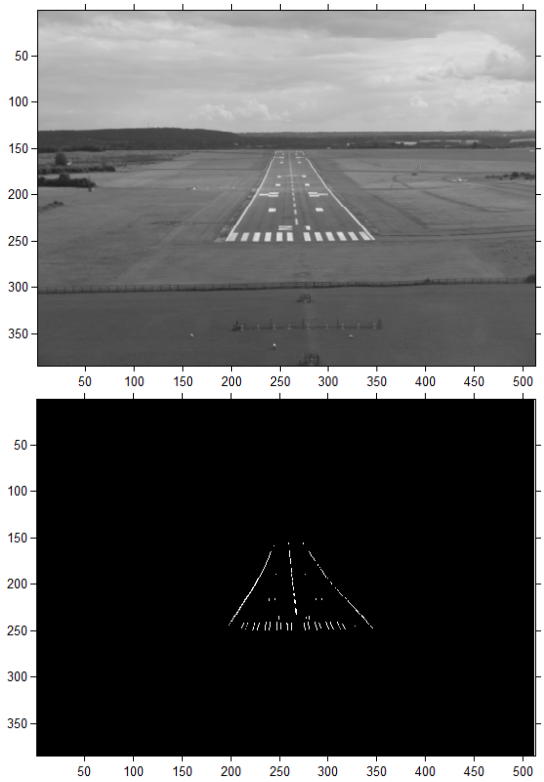

Fig. 10 Runway detected in the test image

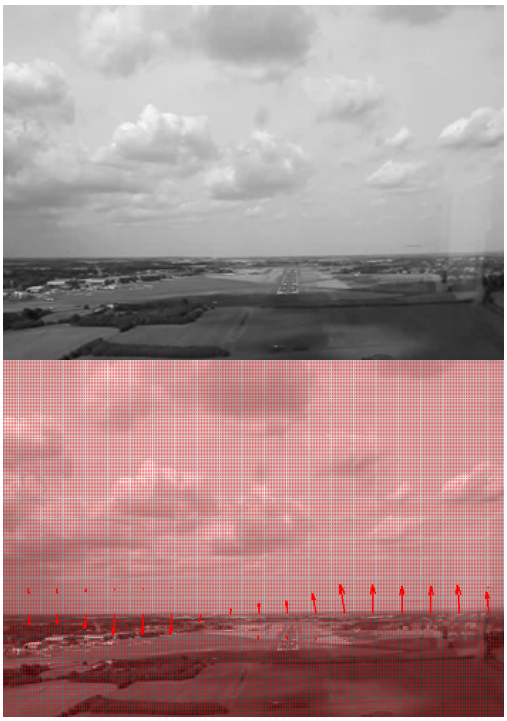

Fig. 11 Received image from camera (top) and optical flow computed for the detected horizon in the image (bottom)

The algorithm also computes the optical flow for all the points on the detected horizon line in the images. The optical flow is determined based on the displacement of points in two consecutive frames of a video. The algorithm takes two consecutive frames at a time and determines the motion for each point on the horizon. These optical flow values are used to compute the body rates of the aircraft. An example of the optical flow calculation is shown in Fig. 11, where the original image (from the camera) is shown on the top and the image on the bottom shows the optical flow vectors (in red) computed for the detected horizon line. The vectors are magnified by a factor of 20. Since the vectors on the right half of the horizon line are pointing upwards and the vectors on the left halfare pointing downwards, the aircraft is performing roll motion (clockwise direction). The real-time performance of the IPM algorithms were evaluated using a combination of experimental data (from the VBN camera) collected in flight and IPM simulation/data analysis performed on the ground using MATLAB^{TM}. The algorithm processed the video frame by frame and extracted horizon and the runway from each frame. The roll and pitch of the aircraft were computed based on the horizon detected in each frame.

The algorithm also identified the location of runway centreline in each frame which was further used to calculate the deviation of the aircraft from the runway centreline. Kalman filter was employed to reduce the effect of errors in the measurements. The roll and roll-rate results obtained for 800 frames are shown in Fig. 12.

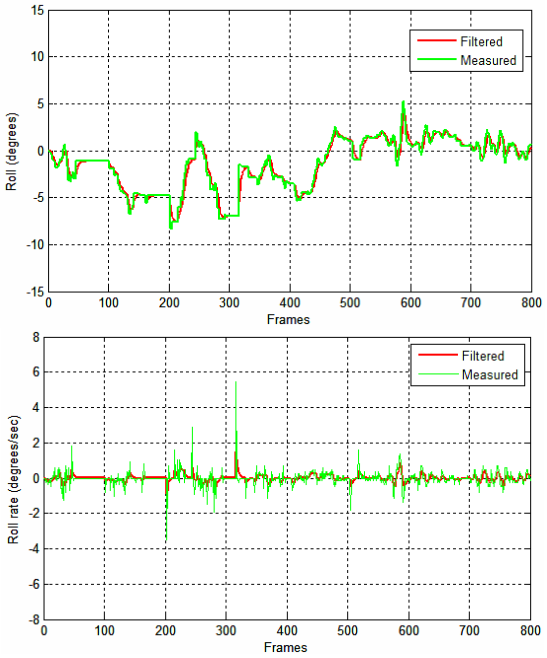

Fig. 12 Roll and roll-rate computed from the test video

TABLE II
VBN ATTITUDE AND ANGULAR RATES ERRORS PARAMETERS

VBN measured parameters	Mean	Standard deviation
Roll angle	0.22°	0.02°
Pitch angle	-0.32°	0.06°
Yaw angle (centreline deviation)	0.64°	0.02°
Roll rate	0.33°/s	0.78 °/s
Pitch rate	-0.43°/s	0.75°/s
Yaw rate	1.86°/s	2.53°/s

Similarly, Fig. 13 depicts the results for pitch and pitch-rate. The computed location of centreline (pixels) and the centreline drift rate (pixels per second) are shown in Fig. 14. Although the test activities were carried out in a limited portion of the UA operational flight envelopes, some preliminary error analysis was performed comparing the performance of the VBN sensor and IMU. The mean and standard deviation of the VBN attitude and attitude-rate measurements are listed in Table II.

The performance of the VBN sensor is strongly dependent on the characteristics of the employed camera. The developed algorithms are unable to determine the attitude of the aircraft in case of absence of horizon in the image. Similarly, the deviation of the aircraft from the runway centreline cannot be computed in the absence of runway in the image. The most severe physical constrain is imposed by the Field of View (FOV) of the camera. The maximum vertical and horizontal FOVs of the Flea Camera are 57.6° and 77.6° respectively. Due to this limitation, the VBN sensor can compute a minimum pitch angle of -28.8° and a maximum of +28.8°. Additionally, environmental factors such as fog, night/low-light conditions or rain also affect the horizon/runway visibility and degrade the performance of the VBN system.

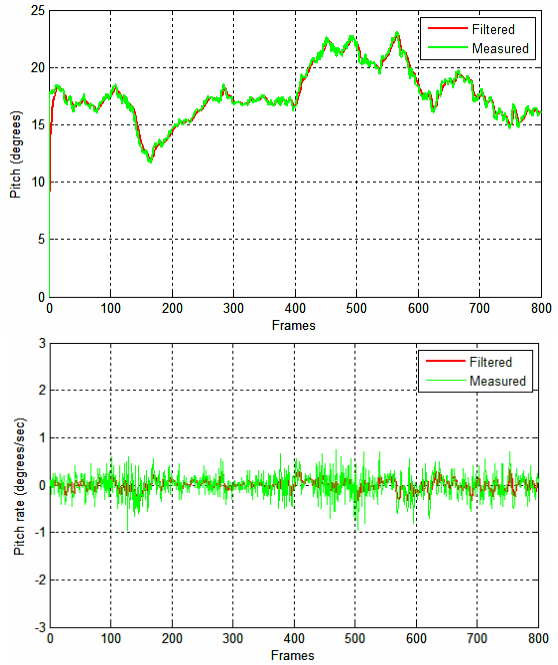

Fig. 13 Pitch and pitch-rate computed from the test video

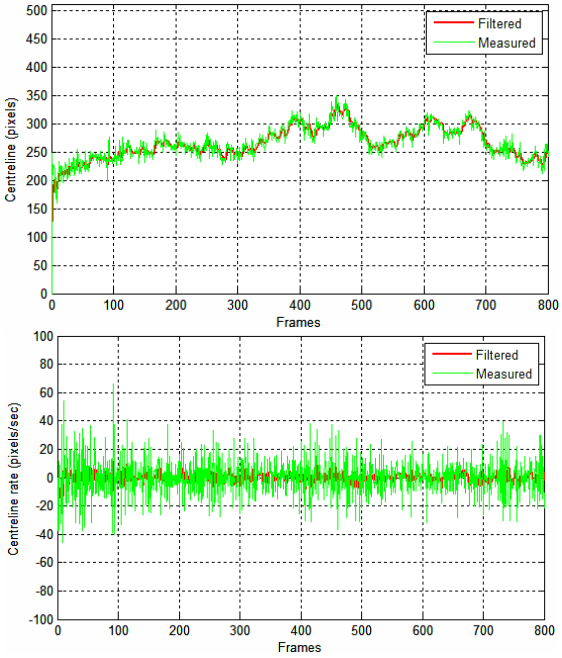

Fig. 14 Centreline and rate of change of centreline location from the test video

III. Integration of Candidate Sensors

There are a number of limitations and challenges associated to the employment of VBN sensors in UA platforms. VBN is best exploited at low altitudes, where sufficient features can be extracted from the surrounding. The FOV of the camera is limited and, due to payload limitations, it is often impractical to install multiple cameras. When multiple cameras are installed, additional processing is required for data exploitation. In this case also stereo vision techniques can be implemented. Wind and turbulence disturbances must be modelled and accounted for in the VBN processing. Additionally the performance of VBN can be very poor in low-visibility conditions (performance enhancement can be achieved employing infrared sensors as well). However, despite these limitations and challenges, VBN is a promising technology for small-to-medium size UA navigation and guidance applications, especially when integrated with other low-cost and low-weight/volume sensors currently available. In our research, we developed an integrated NGS approach employing two state-of-the-art physical sensors: MEMS-based IMU and GNSS, as well as augmentation from Aircraft Dynamics Model (ADM) in specific flight phases.

A. GNSS and MEMS-IMU Sensors Characteristics

GNSS can provide high-accuracy position and velocity data using pseudorange, carrier phase, Doppler observables or various combinations of these measurements. Additionally, using multiple antennae suitably positioned in the aircraft, GNSS can also provide attitude data. Additional research is currently being conducted on GNSS Carrier Phase Measurements (CFM) for attitude estimation. Table III lists the position and velocity of state-of-the-art GNSS receivers. Position error parameters are from [20] and velocity error parameters are from [21], in which an improved time differencing carrier phase velocity estimation method was adopted. Typically, GNSS position and velocity measurements are provided at a rate of 1 Hz.

TABLE III
GNSS Position and Velocity Errors

Errors	Mean	Standard deviation
North position error	-0.4 m	1.79 m
East position error	0.5 m	1.82 m
Down position error	0.17 m	3.11 m
North velocity error	0 mm/s	3.8 mm/s
East velocity error	0 mm/s	2.9 mm/s
Down velocity error	2.9 mm/s	6.7 mm/s

The IMU include 3-axis gyroscopes, measuring the roll, pitch and yaw rates of the aircraft around the body-axis. They also comprise 3-axis accelerometers determining the specific forces in the inertial reference frame. In our research, we considered a strap-down IMU employing low-cost MEMS IMUs. MEMS-based IMUs are low-cost and low-weight/volume devices that represent an attractive alternative to high-cost traditional IMU sensors, especially for general aviation or small UA applications. Additionally, MEMS sensors do not necessitate high power and the level of maintenance required is far lower than for high-end IMU sensors [22]. The main drawback of these sensors is the relatively poor level of accuracy of the measurements that they provide. In our research, MEMS IMU errors are modeled as White Noise (WN) or as Gauss-Markov (GM) processes [23],

[24]. Table IV lists the MEMS IMU error parameters considered in our research.

TABLE IV
MEMS-INS ERROR PARAMETERS

IMU error parameters	Error models
p gyro noise	WN (0.53 °/s)
q gyro noise	WN (0.45 °/s)
r gyro noise	WN (0.44 °/s)
x accelerometer noise	WN (0.013 m/s^2)
y accelerometer noise	WN (0.018 m/s^2)
z accelerometer noise	WN (0.010 m/s^2)
p gyrobias	GM (0.0552 °/s, 300 s)
q gyrobias	GM (0.0552 °/s, 300 s)
r gyrobias	GM (0.0552 °/s, 300 s)
x accelerometerbias	GM (0.0124 m/s^2, 300 s)
y accelerometer bias	GM (0.0124 m/s^2, 300 s)
z accelerometer bias	GM (0.0124 m/s^2, 300 s)
p gyroscale factor	GM (10000 PPM, 18000 s)
q gyroscale factor	GM (10000 PPM, 18000 s)
r gyroscale factor	GM (10000 PPM, 18000 s)
x accelerometerscale factor	GM (10000 PPM, 18000 s
y accelerometer scale factor	GM (10000 PPM, 18000 s)
z accelerometer scale factor	GM (10000 PPM, 18000 s)

B. ADM Virtual Sensor Characteristics

The ADM Virtual Sensor is essentially a Knowledge-Based Module (KBM) used to augment the navigation state vector by predicting the UA flight dynamics (aircraft trajectory and attitude motion). The ADM can employ either a 6-Degree of Freedom (6-DOF) or a 3-DOF variable mass model with suitable constraints applied in the different phases of the UA flight. The input data required to run these models are made available from aircraft physical sensors (i.e., aircraft data network stream) and form ad-hoc databases. Additionally, for the 3-DOF case, an automatic manoeuvre recognition module is implemented to model the transitions between the different UA flight phases. Typical ADM error parameters are listed in Table V [23], [24]. Table VI lists the associated error statistics obtained in a wide range of dynamics conditions for 20 seconds runtime.

TABLE V
ADM ERROR PARAMETERS

ADM error parameters	Error models
Coefficients (on all except the flap coefficients)	GM(10%,120s)
Control input	WN(0.02°) aileron, rudder, elevator
Center of gravity error [x,y,z]	Constant [0.001, 0.001, 0.001]m
Mass error	2% of true
Moment of inertia error [J_x,J_y,J_z,J_{xz}]	2% of true
Thrust error	Force, 5% of true, Moment 5% of true
Gravity error 1σ	36 μg
Air density error	5% of true
Speed of sound error	5% of true

TABLE VI
ADM ERROR STATISTICS

Error	Mean	Standard deviation
North velocity error	4.48E-3m/s	3.08E-2m/s
East velocity error	-3.73E-2m/s	1.58E-1m/s
Down velocity error	-4.62E-2m/s	5.03E-2m/s
Roll error	4.68E-5°	7.33E-3°
Pitch error	3.87E-3°	2.41E-3°
Yaw error	-1.59E-3°	7.04E-3°

IV. MULTISENSOR SYSTEM DESIGN AND SIMULATION

The data provided by all sensors are blended using suitable data fusion algorithms. Due to the non-linearity of the sensor models, an EKF was developed to fuse the information provided by the different sensors and to provide estimates of position, velocity and attitude of the platform in real-time. Two different integrated navigation system architectures were defined, including VBN-IMU-GNSS (VIG) and VIG-ADM (VIGA). The VIG architecture uses VBN at 20 Hz and GNSS at 1 Hz to augment the MEMS-IMU running at 100 Hz. The VIGA architecture includes the ADM (computations performed at 100 Hz) to provide attitude channel augmentation. The corresponding VIG and VIGA integrated navigation modes were simulated using MATLAB^{TM} covering all relevant flight phases of the AEROSONDEUA (straight climb, straight-and-level flight, straight turning, turning descend/climb, straight descent, etc.). The navigation system outputs were fed to a hybrid Fuzzy-logic/PID controller [25]-[28]. Our previous research activities [25]-[35] highlight the various sensor choices, data fusion methods and the overall implementation of the various NGS schemes.

A. VIG and VIGA Architectures

The VIG architecture is illustrated in Fig. 15. The sensor measurements are processed by a sensor processing and data sorting block. The data sorting algorithm is based on Boolean decision logic, which accepts 0 and 1 as input states and allows automatic selection of the sensor data based on pre-defined priority criteria. The sorted data is then fed to an EKF to obtain the best estimate values. The IMU position and velocity provided by the navigation processor are compared to the GNSS position and velocity to form the measurement input of the data fusion block containing the EKF. A similar process is also applied to the IMU and VBN attitude angles, whose differences are incorporated in the EKF measurement vector. The EKF provides estimates of the Position, Velocity and Attitude (PVA) errors, which are then removed from the sensor measurements to obtain the corrected PVA states. The corrected PVA and estimates of accelerometer and gyroscope biases are also used to update the IMU raw measurements. The VIGA architecture is illustrated in Fig. 16. As in the case of VIG system, the IMU position and velocity provided by the navigation processor are compared to the GNSS data to form the measurement input of EKF. Additionally, in this case, the attitude data provided by the ADM and the IMU are compared to feed the EKF at 100 Hz, and the attitude data provided by the VBN sensors (VBS) and IMU are compared at 20 Hz and

input to the EKF. The EKF provides estimations of PVA errors, which are removed from the IMU measurements to obtain the corrected PVA states. The corrected PVA and estimates of accelerometer and gyroscope biases are used to update IMU raw measurements and the corrected attitude date is fed back to the ADM.

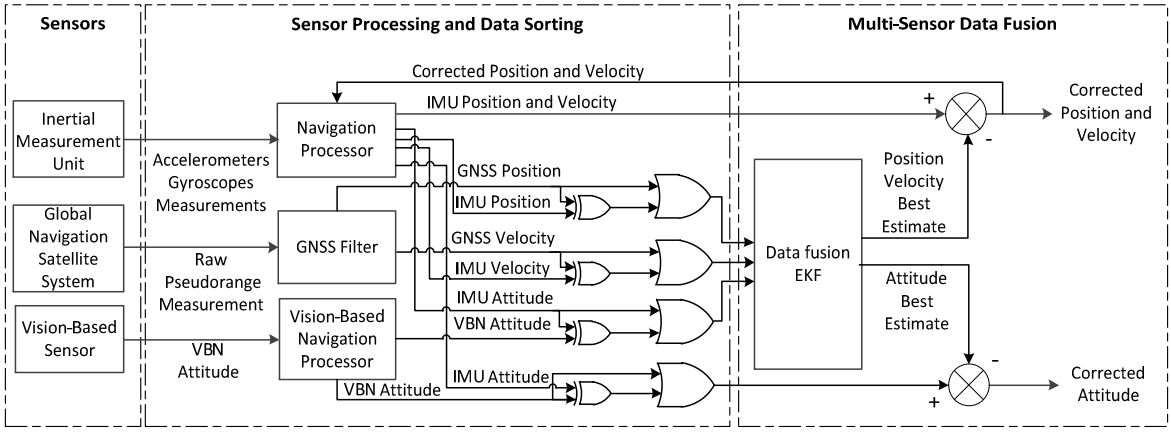

Fig. 15 VIG architecture

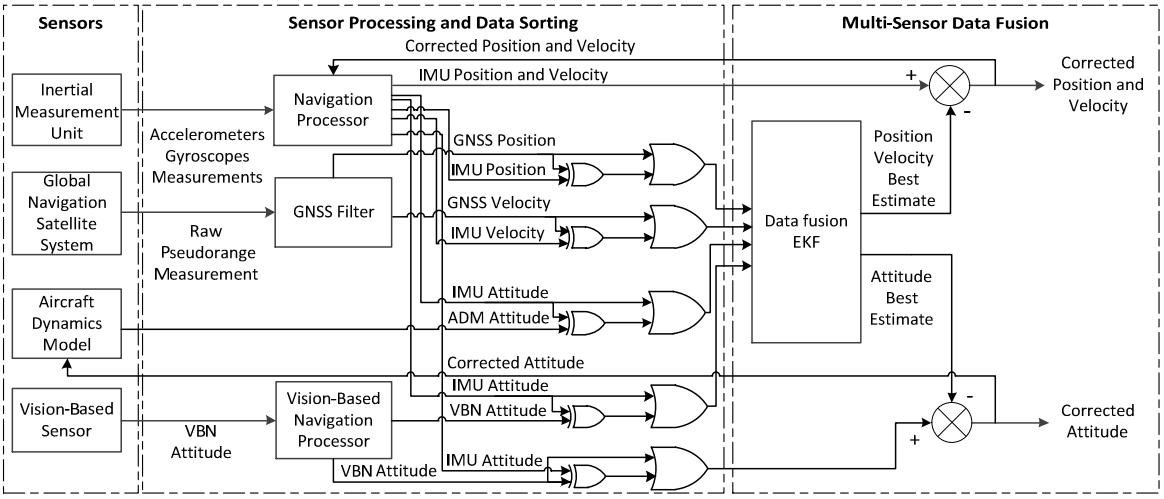

Fig. 16 VIGA architecture

B. VIG and VIGA Simulation

Both the VIG and VIGA multisensor architectures were tested by simulation in an appropriate sequence of flight manoeuvres representative of the AEROSONDE UA operational flight envelope. The duration of the simulation is 600 seconds. The 3D trajectory plot of UA flight profile is shown in Fig. 17. The list of the different simulated flight manoeuvres and associated control inputs is provided in Table VII. The VIG position error time histories (east, north and down) are shown in Fig. 18. For comparison, also the GNSS position errors (unfiltered) are shown. Table VIII presents the VIG position error statistics associated to each flight phase.

TABLE VII
FLIGHT MANOEUVRES AND CONTROL INPUTS

Flight maneuver	Required roll (°)	Required pitch (°)	Time (s)
Straight climb (take off)	0	10	50
Leftturningclimb	5	10	50
Left turning climb (helix)	4	2	50
Level left turn	10	8	100
Straight and level	0	3.5	100
Level right turn (helix)	12	-5	130
Level right turn	3	-4	50
Left turning descent	6	-3	50
Straight descent	0	-2	20

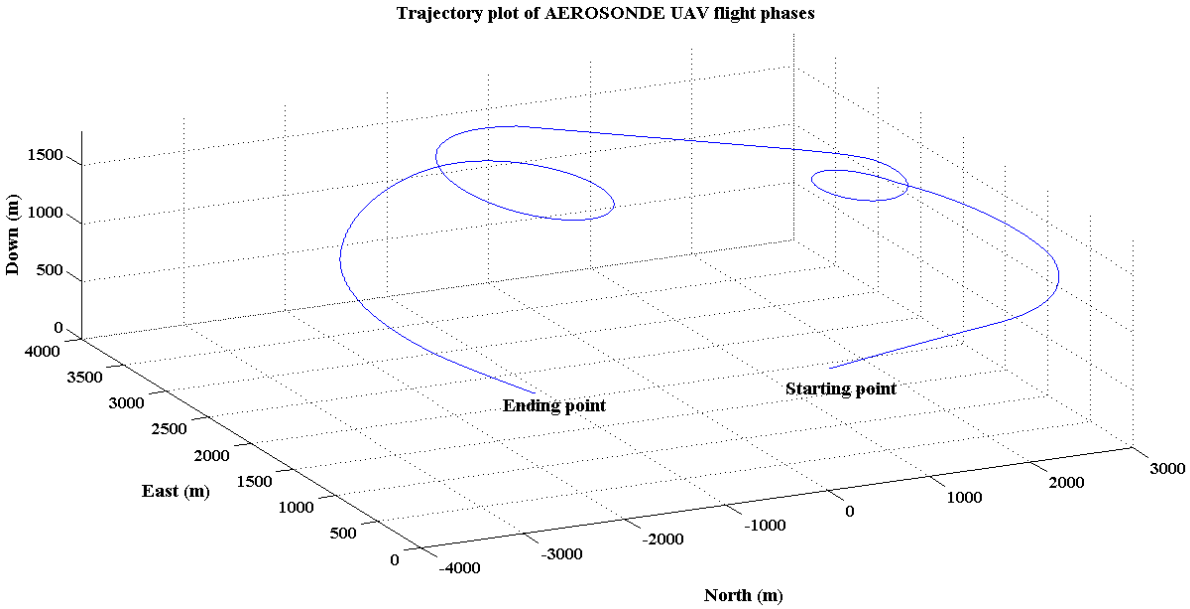

Fig. 17 3D Trajectory plot of UA flight profile

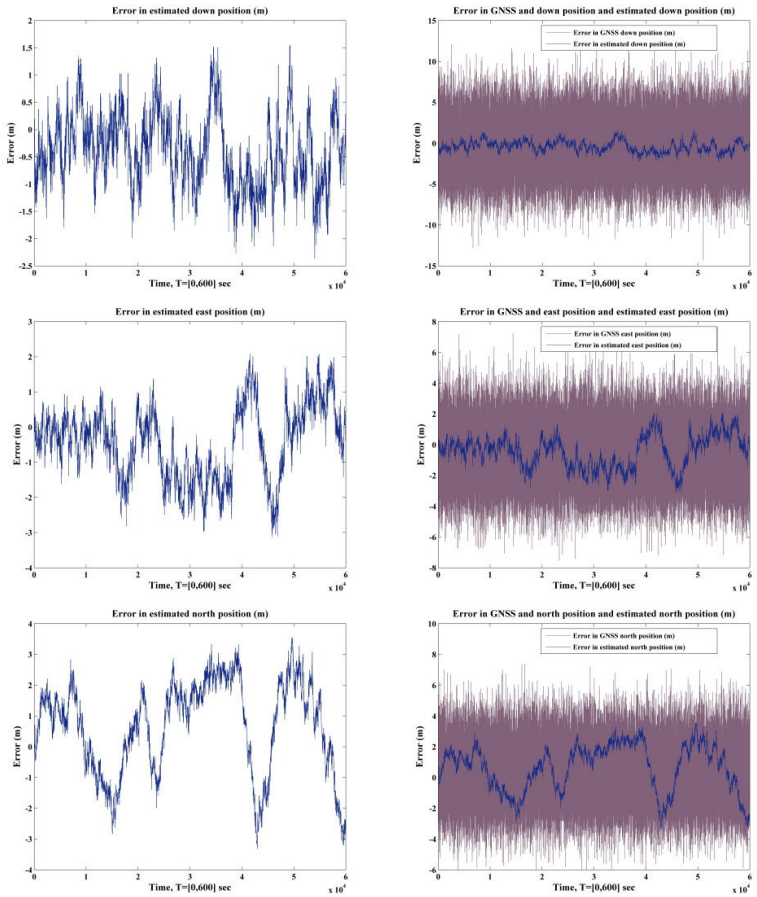

Fig. 18 VIG position error time histories

TABLE VIII
VIG POSITION ERROR STATISTICS

Flight maneuver	North position		East position		Down position	
	Mean (m)	σ (m)	Mean (m)	σ (m)	Mean (m)	σ (m)
Straight climb (take off)	1.1082	0.5897	-0.1871	0.3949	-0.4323	0.4526
Left turning climb	1.0531	0.7401	-0.2290	0.5289	-0.0612	0.6278
Left turning climb (helix)	-1.0110	0.5750	-0.1447	0.4886	-0.2855	0.4418
Level left turn	-0.1701	1.1080	-0.6219	0.8439	-0.1369	0.6026
Straight and level	1.7413	0.5327	-1.5381	0.5442	-0.3599	0.6775
Level right turn (helix)	0.9409	1.8376	-0.7228	1.3230	-0.7846	0.6752
Level right turn	2.1041	0.6436	0.2312	0.5950	-0.3845	0.7172
Left turning descent	0.5003	1.1802	0.9372	0.6482	-0.7820	0.7348
Straight descent	-2.3483	2.2365	-0.1956	0.4206	-0.3130	0.7518

The VIG velocity error time histories are obtained and it is deducted that GNSS is the dominating sensor for velocity computations but a significant improvement is obtained with the VIG system on the accuracy of the vertical data. Table IX shows the velocity error statistics associated to each flight phase. The attitude error time histories of the VIG system are shown in Fig. 19. Table X presents the associated attitude error statistics.

As discussed above, the ADM data were used in the VIGA architecture to update the attitude channel (the position and velocity channels are derived from the VIG system). Therefore, only the attitude error statistics of the VIGA system are presented here. The time histories of the VIGA attitude errors are shown in Fig. 20 and compared with the corresponding VIG attitude errors in Fig. 21. Table XI presents the VIGA attitude error statistics.

TABLE IX
VIG VELOCITY ERROR STATISTICS

Flight maneuver	North velocity		East velocity		Down velocity	
	Mean(m/s)	σ(m/s)	Mean(m/s)	σ(m/s)	Mean(m/s)	σ (m/s)
Straight climb (take off)	0.0012	0.0602	-0.1096	0.1000	0.0124	0.0844
Left turning climb	-0.0088	0.0588	-0.0073	0.0354	-0.0072	0.0423
Left turning climb (helix)	-0.0133	0.0247	-0.0011	0.0498	-0.0018	0.0008
Level left turn	-0.0061	0.0416	-0.0013	0.0433	-0.0022	0.0034
Straight and level	0.0162	0.0313	0.0803	0.0350	0.0002	0.0007
Level right turn (helix)	-0.9276	0.0333	0.0088	0.0370	-0.0006	0.0148
Level right turn	0.0066	0.0533	-0.0674	0.0435	-0.0110	0.0462
Left turning descent	-0.0018	0.0904	0.0033	0.0797	-0.0009	0.0361
Straight descent	0.0212	0.0697	0.1302	0.1665	0.0012	0.0146

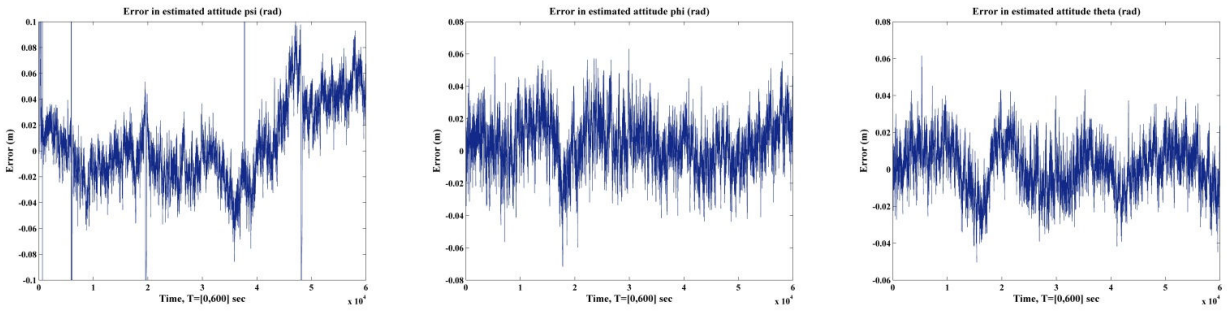

Fig. 19 VIG attitude error time histories

TABLE X
VIG ATTITUDE ERROR STATISTICS

Flight maneuver	Roll (Phi)		Pitch (Theta)		Yaw (Psi)	
	Mean (°)	σ (°)	Mean (°)	σ (°)	Mean (°)	σ (°)
Straight climb (take off)	0.0078	0.0037	0.0597	0.0045	0.0512	0.0102
Left turning climb	-0.0678	0.0502	0.1494	0.0032	-0.0130	0.0104
Left turning climb (helix)	0.0198	0.0421	0.0607	0.0104	-0.0116	0.0133
Level left turn	0.0052	0.0014	-0.0323	0.0167	-0.0323	0.0162
Straight and level	0.0672	0.0531	-0.0809	0.0045	-0.0175	0.0109
Level right turn (helix)	-0.0034	0.0082	-0.0297	0.0145	0.0016	0.0307
Level right turn	0.0102	0.0352	0.0776	0.0067	0.0042	0.0093
Left turning descent	0.0032	0.0058	0.0563	0.0042	0.0529	0.0098
Straight descent	0.0149	0.0128	-0.1107	0.0031	0.0568	0.0106

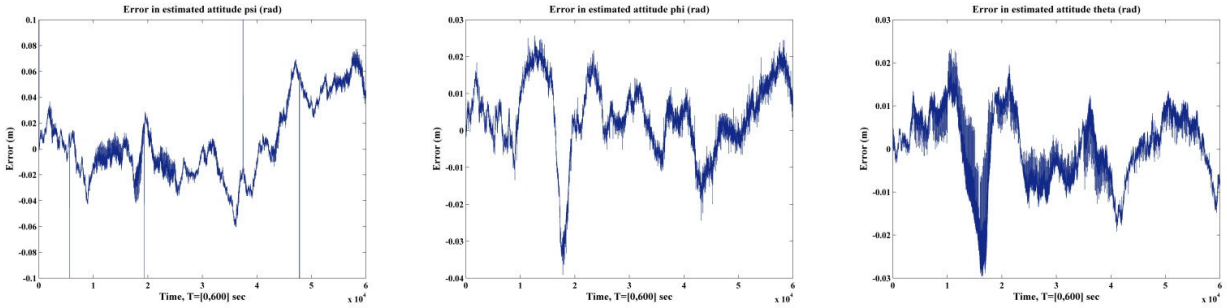

Fig. 20 VIGA attitude error time histories

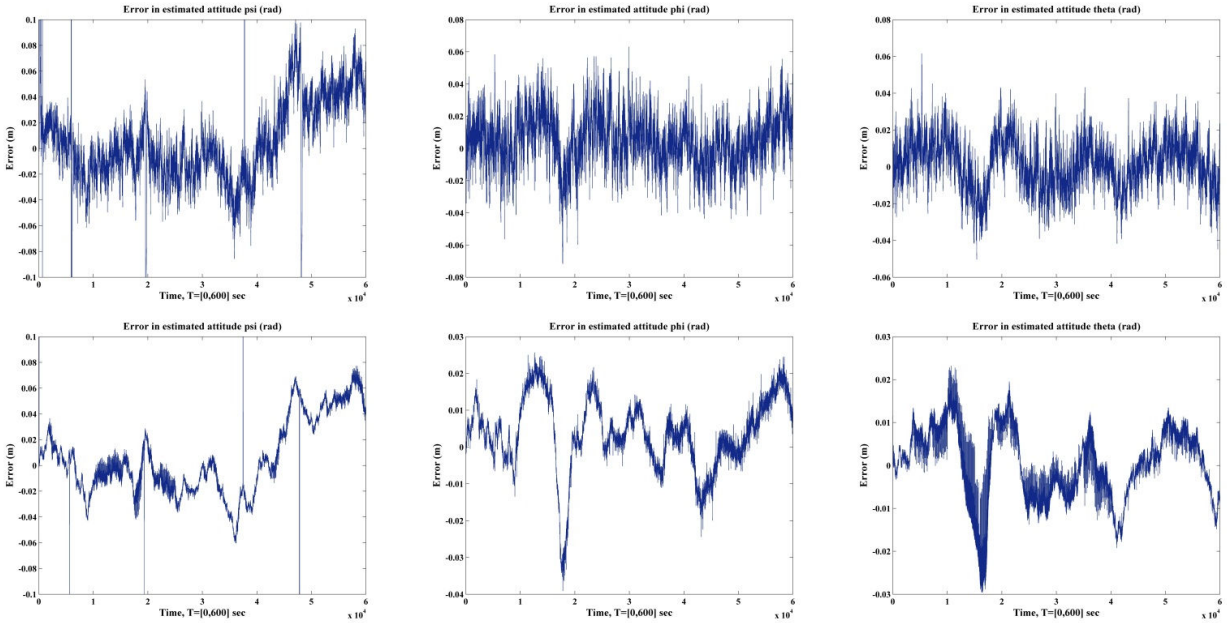

Fig. 21 Comparison of VIG (top) and VIGA (bottom) attitude errors

TABLE XI
VIGA ATTITUDE ERROR STATISTICS

Flight maneuver	Roll (Phi)		Pitch (Theta)		Yaw (Psi)	
	Mean (°)	σ (°)	Mean (°)	σ (°)	Mean (°)	σ (°)
Straight climb (take off)	0.0054	0.0039	0.0034	0.0037	0.0114	0.0083
Left turning climb	-0.0560	0.0461	0.0832	0.0028	-0.0130	0.0156
Left turning climb (helix)	0.0171	0.0307	0.0285	0.0099	-0.0072	0.0049
Level left turn	0.0046	0.0014	-0.0496	0.0132	-0.0073	0.0013
Straight and level	0.0350	0.0043	-0.0600	0.0036	-0.0163	0.0111
Level right turn (helix)	-0.0026	0.0068	-0.0266	0.0065	0.0002	0.0345
Level right turn	0.0012	0.0347	0.0710	0.0030	0.0040	0.0075
Left turning descent	0.0012	0.0069	0.0424	0.0038	0.0525	0.0093
Straight descent	0.0143	0.0036	-0.0893	0.0029	0.0569	0.0096

During the initial VIGA simulation runs it was evidenced that the ADM data cannot be used without being reinitialised regularly. For the AEROSONDE UA manoeuvres listed in Table VII, it was found that the optimal period between ADM reinitialisation was in the order of 20 seconds. Converting the data in Tables X and XI to the corresponding RMS (95%) values, we obtain the error data in Tables XII and XIII. Comparing the two tables, it is evident that the ADM virtual sensor contributes to a moderate reduction of the overall attitude error budget in all relevant flight phases.

TABLE XII
VIG ATTITUDE RMS-95% ERRORS

Flight maneuver	RMS-95% Phi (°)	RMS-95% Theta (°)	RMS-95% Psi (°)
Straight climb (take off)	0.0172	0.1197	0.1044
Left turning climb	0.1687	0.2988	0.0332
Left turning climb (helix)	0.0930	0.1231	0.0253
Level left turn	0.0132	0.0727	0.0722
Straight and level	0.0712	0.1620	0.0412
Level right turn (helix)	0.0177	0.0661	0.0614
Level right turn	0.0732	0.1557	0.0186
Left turning descent	0.0132	0.1129	0.1076
Straight descent	0.0392	0.2214	0.1155

TABLE XIII
VIGA ATTITUDE RMS-95% ERRORS

Flight maneuver	RMS-95% Phi (°)	RMS-95% Theta (°)	RMS-95% Psi (°)
Straight climb (take off)	0.0133	0.0100	0.0282
Left turning climb	0.1450	0.1664	0.0312
Left turning climb (helix)	0.0701	0.0603	0.0174
Level left turn	0.0096	0.1026	0.0148
Straight and level	0.0705	0.1202	0.0394
Level right turn (helix)	0.0145	0.0547	0.0690
Level right turn	0.0694	0.1421	0.0170
Left turning descent	0.1400	0.0851	0.1066
Straight descent	0.0294	0.1787	0.1154

To conclude the simulation data analysis, Table XIV shows a comparison of the VIG/VIGA horizontal and vertical accuracy (RMS-95%) with the required accuracy levels for precision approach [36], [37]. Both the VIG and VIGA performances are in line with CAT II precision approach requirements.

TABLE XIV
VIG AND VIGA POSITION ERROR STATISTICS

Category of approach	Horizontal Accuracy (m) 2D RMS-95%			Vertical Accuracy (m) RMS-95% Down		
	Required	VIG	VIGA	Required	VIG	VIGA
CAT I	16			4		
CAT II	6.9	6.2	5.9	2	1.6	1.6
CAT III	4.1			2		

V. CONCLUSIONS

Our research efforts have addressed the development of a low-cost and low-weight/volume integrated Navigation and Guidance System (NGS) for small/medium size UA applications. As a first step, we designed and tested a VBN sensor employing appearance-based techniques and specifically tailored for UA low-level flight, including precision approach and landing operations. In addition to horizon detection and image-flow, the VBN sensor also employed runway features extraction during the approach phase. Various candidates were considered for integration with the VBN sensor and, as a result, GNSS and MEMS-IMUs, with attitude augmentation from ADM were finally selected. The multisensor integration was accomplished with an EKF. The attitude/attitude-rate accuracies obtained with the VBN sensor were evaluated by a combination of laboratory, ground and flight test activities. The results were satisfactory in low-level flight and during the approach and landing phases of a UA flight. However, the VBN sensor performance was strongly dependent on the characteristics of the employed camera. The algorithms developed are unable to determine the attitude of the aircraft in case of absence of horizon in the image. Similarly, the deviation of the aircraft from the runway centreline cannot be computed in the absence of runway in the image. The most severe physical constrain is imposed by the angular FOV of the camera. The maximum vertical and horizontal FOVs of the employed camera are 57.6° and 77.6° respectively. Due to this limitation, the VBN sensor can compute a minimum pitch angle of -28.8° and a maximum of +28.8°. Two NGS architectures, namely VBN-IMU-GNSS (VIG) and VIG-ADM (VIGA), were proposed and the AEROSONDE UA was used as a test-bed for the simulation test cases. Simulation of the VIG integrated navigation mode showed that this integration scheme can achieve horizontal/vertical position accuracies in line with CAT-II precision approach requirements, with a significant improvement compared to stand-alone GNSS. An improvement was also observed in the accuracy of the vertical velocity data. Additionally, simulation of the VIGA navigation mode showed promising results since, in most cases, the attitude accuracy is higher using the ADM-VBN-IMU than using VBN-IMU only. However, due to rapid divergence of the ADM virtual sensor, there is a need for a frequent re-initialisation of the ADM data module, which is strongly dependent on the UA flight dynamics and the specific manoeuvres/flight-phase transitions performed. In the considered portion of the UA operational flight envelope, the required re-initialisation interval was approximately 20 seconds. Results show that there is a moderate accuracy improvement with the use of ADM in high dynamics conditions. Compared to the VIG system, the VIGA shows accuracy improvements in all three attitude angles. Current research activities are focusing on adopting the proposed integrated architectures for other UA platforms such as the JAVELIN UA [38]. GNSS multipath and antenna obscuration problems are being also investigated and adequate algorithms are being developed in order to cope with these effects during high dynamic manoeuvres. To cope with the issue of frequent ADM re-initialisation, the original ADM is being modified to take into account specific manoeuvre constraints and the transition states between various manoeuvres are being carefully modelled leading to a higher stability time. Additionally, the potential benefits attainable from the adoption of an Unscented Kalman Filter (UKF) to replace the EKF and/or to pre-filter the ADM data are being investigated. Finally, Avionics Based Integrity Augmentation (ABIA) and integrity monitoring functionalities for UA are being researched in order to improve the overall system performance in mission- and safety-critical applications [39]-[42]. As a result of these further developments, the performance of the multisensory integrated NGS will be significantly enhanced in terms of data accuracy, continuity and integrity to fulfill present and likely future RNP requirements for a variety of small UA applications.

REFERENCES

[1] B. Sinopoli, M. Micheli, G. Donato, and T. J. Koo, "Vision based navigation for unmanned aerial vehicles," in *Proc. International Conf. of Robotics & Automation*, vol. 2, 2001, pp. 1757 – 1764.
[2] S. Se, D. G. Lowe, and J. J. Little, "Vision based global localisation and mapping," *IEEE Trans. on Robotics*, vol. 21, no.3, pp. 364-375, June 2005.
[3] P. Cui and F. Yue, "Stereo vision-based autonomous navigation for lunar rovers," *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 79, no. 4, pp. 398-405, 2007.
[4] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue, "View-based approach to robot navigation," in *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, vol. 3, Japan, Nov. 2000, pp. 1702-1708.
[5] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, "Visual navigation of a quadrotor aerial vehicle," in *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, Oct. 2009, pp. 5315-5320.
[6] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, no. 7, pp. 789-799, July 2010.
[7] Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *IEEE Trans. on Robotics*, vol. 25, no. 3, pp. 749-754, June 2009.
[8] A. Remazeilles, and F. Chaumette, "Image-based robot navigation from an image memory," *Journal of Robotics and Autonomous Systems*, vol. 55, no. 4, 2007.
[9] L. Xinhua and Y. Cao, "Research on the application of the vision-based autonomous navigation to the landing of the UAV," in *Proc. Fifth International Symposium on Instrumentation and Control Technology*, vol. 5253, 2003, pp. 385-388.
[10] D. Dusha, L. Mejias, and R. Walker, "Fixed-wing attitude estimation using temporal tracking of the horizon and optical flow," *Journal of Field Robotics*, vol. 28, no. 3, pp. 355-372, 2011.
[11] M. A. Olivares-Mendez, I. F. Mondragon, P. Campoy, and C. Martinez, "Fuzzy controller for UAV-landing task using 3D position visual estimation," in *Proc. IEEE International Conf. on Fuzzy Systems*, 2010.
[12] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting Inertial Navigation with image-basedestimation," in *Proc. International Conf. of Robotics & Automation,* 2002, pp. 4326-4333.
[13] G. N. Desouza and A. C. Kak, "Vision for mobile robot navigation: a survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237 – 267, Feb. 2002.
[14] D. Santosh, S. Achar, and C. V. Jawahar, "Autonomous image-based exploration for mobile robot navigation," in *Proc. International Conf. of Robotics & Automation*, 2008, pp. 2717 – 2722.
[15] P. Rives and J. R. Azinheira, "Visual auto-landing of an autonomous aircraft," INRIA, no. 4606, 2002.
[16] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *Proc. International Conf. of Robotics & Automation*, 2005, pp. 3354-3359.
[17] G. Rangasamy, "Image sensor fusion algorithms for obstacle detection, location and avoidance for autonomous navigation of UAVs," M.Sc. Thesis, School of Engineering, Cranfield University, 2010.
[18] E. H. Shin, "Estimation technics for low-cost inertial navigation," PhD Thesis, University of Calgary, Alberta, Canada, 2005.
[19] A. Angrisano, "GNSS/INS Integration Methods," PhD Thesis, Department of Applied Sciences, Parthenope University of Naples, Italy, 2010.
[20] C. Tiberius, Standard positioning service, "Handheld GPS receiver accuracy," GPS World, 2003.
[21] W. Ding and J. Wang, "Precise Velocity Estimation with a Stand-Alone GPS receiver," *The Journal of Navigation*, vol. 69, no. 2, pp. 311-325, 2011.
[22] D. Titterton and J. Weston, "Strap down Inertial Navigation Technology," (2nd Edition), *The Institution of Electrical Engineers*, 2004.
[23] S. Troy, "Investigation of MEMS Inertial Sensors and Aircraft Dynamic Models in Global Positioning System Integrity Monitoring for Approaches with Vertical Guidance," PhD thesis, Queensland University of Technology, School of Engineering, 2009.
[24] S. Godha, "Performance Evaluation of Low Cost MEMS-Based IMU Integrated With GPS for Land Vehicle Navigation Application," UCGE Report No. 20239, University of Calgary, Department of Geomatics Engineering, Alberta, Canada, 2006.
[25] R. Sabatini, C. Bartel, A. Kaharkar, T. Shaid, H. Jia, and D. Zammit-Mangion , "Design and Integration of Vision-based Navigation Sensors for Unmanned Aerial Vehicles Navigation and Guidance," in *Proc. SPIE Photonics Europe Conf.*, Brussels, Belgium, 2012.
[26] R. Sabatini, L. RodríguezSalazar, A. Kaharkar, C. Bartel, and T. Shaid, "GNSS Data Processing for Attitude Determination and Control of Unmanned Aerial and Space Vehicles," in *Proc. European Navigation Conf.*, Gdansk (Poland), April 2012.
[27] R. Sabatini, L. RodríguezSalazar, A. Kaharkar, C. Bartel, and T. Shaid, "Low-Cost Vision Sensors and Integrated Systems for Unmanned Aerial Vehicle Navigation and Guidance," *ARPN Journal of Systems and Software*, ISSN: 2222-9833, vol. 2, no. 11, pp. 323-349, 2013.
[28] R. Sabatini, L. RodríguezSalazar, A. Kaharkar, C. Bartel, T. Shaid, D. Zammit-Mangion, and H. Jia, "Low-Cost Navigation and Guidance Systems for Unmanned Aerial Vehicles – Part 1: Vision-Based and Integrated Sensors," *Annual of Navigation Journal*, vol. 19, pp. 71-98, 2012.
[29] R. Sabatini, L. RodríguezSalazar, A. Kaharkar, C. Bartel, and T. Shaid, "Carrier-Phase GNSS Attitude Determination and Control System for Unmanned Aerial Vehicle Applications,"*ARPN Journal of Systems and Software*, ISSN: 2222-9833, vol. 2, no. 11, pp.297-322, 2012.
[30] R. Sabatini, C. Bartel, A. Kaharkar, T. Shaid, D. Zammit-Mangion, and H. Jia, "Vision Based Sensors and Multisensor Systems for Unmanned Aerial Vehicles Navigation and Guidance," in *Proc. European Navigation Conf.*, Gdansk, Poland, 2012.
[31] R. Sabatini, L. RodríguezSalazar, A. Kaharkar, C. Bartel, and T. Shaid, "Satellite Navigation Data Processing for Attitude Determination and Control of Unmanned Air Vehicles," in *Proc. European Navigation Conf.*, Gdansk, Poland, 2012.
[32] R. Sabatini, C. Bartel, A. Kaharkar, T. Shaid, L. RodríguezSalazar, and D. Zammit-Mangion, "Low-Cost Navigation and Guidance Systems for Unmanned Aerial Vehicles – Part 2: Attitude Determination and Control," *Annual of Navigation*, vol. 20, pp. 97-126, 2013.
[33] R. Sabatini, S. Ramasamy, A. Gardi, and L. RodríguezSalazar, "Low-cost Sensors Data Fusion for Small Size Unmanned Aerial Vehicles Navigation and Guidance,"*International Journal of Unmanned Systems Engineering*, vol. 1, no. 3, pp. 16-47, 2013.
[34] R. Sabatini, A. Kaharkar, C. Bartel, and T. Shaid, "Carrier-phase GNSS Attitude Determination and Control for Small UA Applications,"*Journal of Aeronautics and Aerospace Engineering*, vol. 2, no. 4, 2013.
[35] R. Sabatini, C. Bartel, A. Kaharkar, T. Shaid, and S. Ramasamy, "A Novel Low-cost Navigation and Guidance System for Small Unmanned Aircraft Applications," in *Proc. WASET International Conf. on Aeronautical and Astronautical Engineering (ICAAE 2013)*, Melbourne, Australia, 2013.
[36] ICAO - Annex 10 to the Convention on International Civil Aviation, "Aeronautical Telecommunications - Volume 1: Radio Navigation Aids," Edition 6, July 2006.
[37] CAA Safety Regulation Group Paper 2003/09, "GPS Integrity and Potential Impact on Aviation Safety," 2003.
[38] RMIT University, "Sky's the limit,"2013, Available online at: http://rmit.com.au/browse;ID=wcga2pa6sovqz. [Accessed 9th April, 2014].
[39] R. Sabatini, T. Moore, and C. Hill, "A Novel GNSS Integrity Augmentation System for Civil and Military Aircraft," *International Journal of Mechanical, Industrial Science and Engineering*, vol. 7, no. 12, pp. 1433-1449. International Science Index 84, 2013.
[40] R. Sabatini, T. Moore, and C. Hill, "A New Avionics Based GNSS Integrity Augmentation System: Part 2 – Integrity Flags," *Journal of Navigation*, vol. 66, no. 4, pp. 511-522, 2013.
[41] R. Sabatini, T. Moore, and C. Hill, "A New Avionics Based GNSS Integrity Augmentation System: Part 1 – Fundamentals," *Journal of Navigation*, vol. 66, no. 3, pp. 363-383, 2013.
[42] R. Sabatini, T. Moore, and C. Hill, "Avionics Based GNSS Integrity Augmentation for Mission- and Safety-Critical Applications," in *Proc. 25th International Technical Meeting of the Satellite Division of the Institute of Navigation: ION GNSS-2012*, Nashville, Tennessee, September 2012.