Adaptive GNSS/INS Integration Based on Supervised Machine Learning Approach

Guohao Zhang and Li-Ta Hsu

Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong

Abstract. Since the usage of Unmanned Aerial Vehicle (UAV) for civil applications is increasing, the localization accuracy in urban becomes an important issue for safety. However, the GNSS localization solution suffers a large error by the multipath effect. Since the multipath effect is unable to be completely solved but to mitigate, the multi-sensor integrated localization method is a common method to reduce this error. This study develops an adaptive Kalman filter adjusting the weighting between GNSS and INS measurements for different circumstance, further to improve the integration performance. The adaptation is based on supervised machine learning model classification, predicting the GNSS conditions with measurement features. The principle component analysis (PCA) is employed to aid selecting major features and labeling data for machine learning. Then, the supervised machine learning model is trained base on the decision tree and random forest (RF) learning algorithm with real operation data covering most situations. To reduce the miss-classification error, the fuzzy logic algorithm is designed to avoid the classification result with rapid change. Besides, the process noise covariance is determined with Allan variance analysis. The localization performance of the proposed adaptive Kalman filter is compared with conventional Kalman filter and onboard localization solution provided by commercial fly controller. The results prove that the presented adaptive Kalman filter using random forest with fuzzy logic can achieve better GNSS condition classification outperforming other algorithms. The fuzzy logic in the proposed algorithm can mitigate jumping error causing by miss-classification. For urban areas, the overall localization result improves about 50% comparing with the onboard solutions. The maximum localization error can be reduced from 43.2 to 14.7 meters. The result verifies that the proposed adaptive Kalman filter can mitigate the localization error from multipath effect as well as achieving more accurate localization solution for UAV in urban areas.

Keywords: UAV, Adaptive Kalman Filter, Supervised Machine Learning, Decision Tree, Random Forest, Fuzzy Logic.

1 Introduction

Unmanned aerial vehicles (UAV) have been widely employed in military as military reconnaissance, and recently, UAV are increasingly used in civilian applications, such

as disaster searching and rescuing [1], package delivering [2] and scientific research [3]. Since UAV has the advantages of high maneuvering capability and auto operation without pilot control, those missions can achieve higher safety and convenient with UAV. While more UAV participate into civilian applications, the operating environment is required closer to civilians where closely near urban cities or even inside the urban. The navigation system of UAV is usually guided by Global Navigation Satellite System (GNSS) localization, by receiving different satellite signals and further process all the signals, the distance between receiver and satellite can be calculated, and the current position of UAV can be further determined. The performance of GNSS localization is affected by serval factors, including satellite clock/orbit bias, atmospheric delay, receiver thermal noise and multipath delays, by nowadays technologies as clock calibration, atmosphere model correction, the error of GNSS localization can be reduced to 1-2 meters in open area. However, when the operating environment is close to urban area with buildings nearby or surrounding, the signals from satellites will be reflected by the surface of buildings and suffer a signal delay that further cause positioning error, namely, the multipath delay [4]. The multipath delay is commonly occurred effect and the dominate error in the urban area for GNSS localization, the multipath effect can be mitigated with improved antenna and signal processing methods. However, there has no complete solution to eliminate this error, even more, when the receiver is operating in the deep urban with dense building construction, the multipath effect will be severe and even able to introduce enormous error on localization. As Fig. 1 shows, the GNSS localization solution has large error with regarding to the true trajectory, where some of the error even exceeds 50 meters. Comparing the space between each building in urban, this error level is dangerous UAV to operate automatic missions and has high risk crushing on buildings due to the wrong localization result. Hence, more effective method is required to reduce the multipath introduced positioning error, in order to achieve higher safety and reliability for UAV operation in urban area.

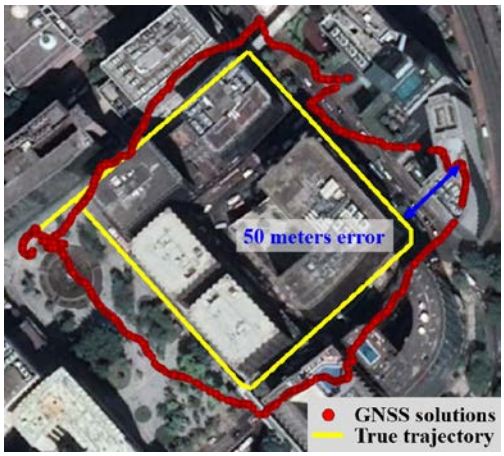

Fig. 1. Experimental UAV localization error in urban area, blue line as the true trajectory.

The common method to achieve better accuracy for UAV navigation positioning is to apply sensor fusion. For GNSS, the data receiving frequency is low and may subject to outages when process positioning, also the short-term noise is high, but it is stable during a long term and relatively accurate. On the other hand, the inertial navigation system (INS) has the characteristics to supply continuous data with high rate with low short-term noise, but suffering large accumulated error. Due to these complementary aspects between GNSS and INS, it is effective to integrate these navigation systems for positioning and achieve higher precision localization performance [5, 6]. Other high-class sensors also been used to improve the localization accuracy for UAV by recent studies, such as vision sensor [7, 8] or light detection and ranging (LiDAR) with simultaneous localization and mapping (SLAM) technology aiding the navigation system when suffering a large GNSS positioning error or large outage. However, these high-class sensors usually have large weight may exceeds UAV`s payload capability, and also expensive for widely using. Moreover, the high computation load will further cause large power consumption that limits the UAV operation duration. Then, another effective method to improve the localization performance in urban is to improve the conventional GNSS/INS integration adapt into urban environment.

The well-known Kalman filter [9, 10] is widely employed to integrate GNSS and INS with a tradeoff between two systems. Usually, the INS will be treated as prediction and GNSS as measurement, by comparing their localization result, the Kalman filter can update the position after each integration. The value of process noise covariance and measurement noise covariance can affect the Kalman gain, further control the weighting between prediction and measurement. For most application, the process noise covariance and measurement noise covariance are fixed values, then the weighting between INS and GNSS is static value. However, the operating environment is not static in urban area. During a single flight as shown in Fig. 2, the UAV will go through some area as A with dense buildings causing large GNSS error, it also will fly to some area as B, has better satellite view and achieve good GNSS positioning result. Then, constant covariance is unable to adapt all situations and may not able to achieve optimal performance, the adaptive Kalman filter (AKF) [11, 12] is required to adjust the weighting between GNSS and INS. In this study, we develop an adaptive Kalman filter to adjust the weighting between GNSS and INS, in order to achieve higher localization accuracy in urban area. Since the operating environment changing in urban scenario greatly influence the GNSS measurement rather than the INS measurement, the adaptation is based on the GNSS measurement condition to adjust the measurement noise covariance. The process noise covariance relating to INS is designed with an appropriate constant value. The GNSS also provides many other information while measuring position, such as position dilution of precision (PDOP), satellite numbers (nSat), etc. These measurements are highly relating to the GNSS measurement environment, and able to be used to estimate the measurement condition.

Since there are several information and each of them has different meaning, it is hard to directly develop a complete model denoting the relationship. By employing the machine learning algorithm, the relationship can be derived by a learning model trained

with enough data [13]. Without studying the detail relationship, the well-trained model is able to predict the condition with an acceptable accuracy. Among different kinds of machine learning algorithm, the proposed adaptive Kalman filter employs the supervised machine learning algorithm as the decision tree (DT) [14] and the random forest (RF) [15] for prediction. The random forest algorithm is able to correct the overfitting problem from decision tree (DT) algorithm, improving the accuracy. Here, the random forest is trained with different measurement condition data and able to predict the condition. However, the machine learning model still occur misclassification as the out of sample error, which may still influence the overall integration performance. During the operation, the GNSS condition related value is slightly change, then the rapid changing classification result is more likely the misclassification error. In this study, we further employed the fuzzy control algorithm to smooth the rapid changing classification result and mitigate the misclassification error [16]. Based on the prediction, it is able to determine the measurement noise covariance value and further adjust the weighting between GNSS and INS during integration. The proposed GNSS/INS integration algorithm using the machine learning based adaptive Kalman filter is able to classify the GNSS condition with sufficient accuracy and adjust the weight to adapt different circumstance in urban. The performance of DT, RF and RF with fuzzy logic are also compared in the experiment, where the RF with fuzzy logic achieve better classification accuracy. The proposed algorithm is able to achieve higher localization accuracy than the conventional Kalman filter and the onboard positioning result, which can increase the safety during UAV operating inside urban area.

The paper is organized as follows. Section II illustrates the system architecture of the proposed cooperative localization solution. In section III, the adaptive Kalman filter algorithm using different supervised machine learning is derived. The experiment setup and result are provided in Section IV. Finally, the conclusion is drawn in section V.

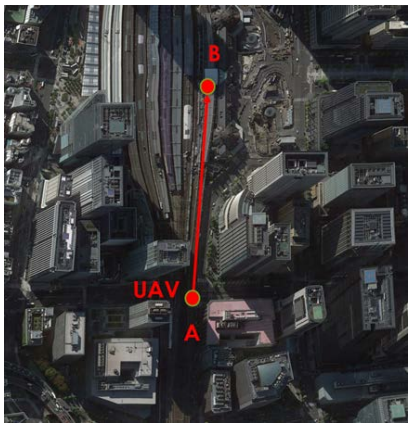

Fig. 2. Different operating environment for UAV.

2 Navigation System

The localization of UAV is usually based on the sensors onboard, such as the Global Navigation Satellite System (GNSS), Gyroscope, Accelerometers and others. Since the localization accuracy is highly related to the performance of UAV and safety operation, multiple sensors are commonly used and integrated to obtain more accurate integrated localization solution. The GNSS/INS integration is widely used in UAV for its reliability and system simplicity, the localization performance is accurate for UAV operation in open sky environment. In this study, we employed the loosely couple GNSS/INS integration algorithm for the UAV navigation system.

2.1 Global Navigation Satellite System

GNSS represents all the globally covered satellite navigation system that obtaining user`s position, velocity and time. By processing the signals transmitted from different satellites and ranging, GNSS can provide three-dimensional localization solution for the user receiver. Base on the principle that satellite signal transporting as the speed of light, the satellite-to-user range can be derived as following, namely the pseudorange.

$$\rho_j = (t_{sa,j} - t_{st,j})c \tag{1}$$

where $t_{sa,j}$ is the decoded signal transmission time and $t_{st,j}$ is the measured arrival time for the j^{th} satellite, c is the speed of light.

After having at least 4 satellite measurements and each satellite position from ephemeris, we are able to apply satellite ranging localization algorithm to calculate user`s position [17]. For over 4 measurements, the position difference between initial position estimation can be obtained by least square iteration:

$$\Delta\mathbf{x} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta\boldsymbol{\rho} \tag{2}$$

$\Delta\boldsymbol{\rho}$ is the pseudorange difference vector, $\mathbf{H}$ is the direction cosines matrix and $\Delta\mathbf{x}$ is the displacement vector which needs to be solved to determine the user position.

By correcting the initial position estimation with $\Delta\mathbf{x}$, the 3-dimensional absolute position of user receiver can be determined with regarding to earth.

Multipath Error

However, GNSS positioning performance is affected by several factors, including satellite clock/orbit bias, atmospheric delays, receiver thermal noise and multipath delays [18]. The measurement errors are originated from time delays since the error sources affect the transmitting time. The equation is given as following:

$$\delta t_D = \delta t_{atm} + \delta t_{noise} + \delta t_{sat} + \delta t_{mp} \tag{3}$$

The overall time offset δt_D is the summation of different signal delays, including the atmosphere errors δt_{atm}, the receiver thermal noise δt_{noise}, satellite clock and orbit bias δt_{sat} and the multipath offset δt_{mp}. There are several models are developed to mitigate or eliminate the errors above. The atmospheric delay is caused from the signal transporting through the ionosphere and troposphere layers, which can be eliminated by differential GPS (DGPS) [19]. In general, the receiver thermal noise in current device is less than the order of a decimeter, which is negligible compared to other errors. The satellite clock and orbit error can be decreased to a neglectable level via constellation update and synchronization corrections.

Besides the errors above, the multipath error is caused by receiving the reflected signals as shown in Fig. 3. The green trajectory represents the direct transmitted satellite signal and the red trajectory represents the satellite signal reflected by the surface of building. Due to the extra traveling distance from reflection, the signal experiences a transporting time error which further influences the correctness of the pseudorange measurement. The multipath effect is highly depending on the surrounding environment, hence DGPS cannot mitigate it. There are several methods to coarsely mitigate multipath effects, such as sophisticated discriminator design and hardware enhanced antennas [20]. However, there is still no complete solution to eliminate this effect. When the UAV operation area is in urban area with many high buildings surrounding, the multipath effect will be very severe. Thus, multipath is the dominant factor for GNSS positioning accuracy in our target application. To ensure the safety of UAV application in urban, it is important to reduce the positioning error due to surrounding buildings multipath effect. An effective method is to integrate with other sensors, compensate multipath error with other more reliable measurements.

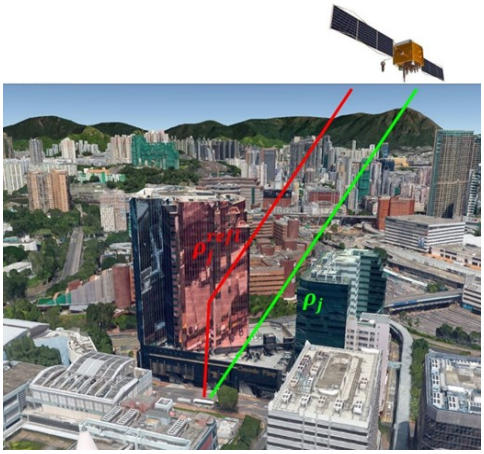

Fig. 3. Multipath reflected signal with extra traveling distance in urban area, green line is the direction transporting signal and red line is the reflected signal

2.2 Inertial Navigation System (INS)

Inertial Navigation System (INS) is a dead-reckoning navigation system, using the Inertial Measurement Units (IMU) measurement and navigation equations to update the current state of target, including attitude, velocity and position [9]. After given the initial state of target, the INS is able to update the attitude of vehicle by the angular measurement from IMU. The attitude will be further combined with specific force measurement from IMU to update the velocity. Finally, the position update can be achieved with velocity. The INS can update the state with a certain rate via iteration and navigate the vehicle during operation. Regarding to the ECEF coordinate frame, the velocity iteration is derived from the change rate of velocity as following:

$$\mathbf{v}_{eb}^{e}(+) \approx \mathbf{v}_{eb}^{e}(-) + (\boldsymbol{f}_{ib}^{e} + \boldsymbol{g}_{b}^{e} - 2\boldsymbol{\Omega}_{ie}^{e}\mathbf{v}_{eb}^{e}(-))\tau_i \tag{4}$$

Here, $\boldsymbol{f}_{ib}^{e}$ is the specific force derived above, $\boldsymbol{g}_{b}^{e}$ is the gravitation force acceleration at a specific location, $2\boldsymbol{\Omega}_{ie}^{e}\mathbf{v}_{eb}^{e}$ is the Coriolis acceleration. τ_i is the time interval. Symbol $(+)$ denotes current epoch and $(-)$ denotes the previous epoch. By assuming the velocity varies linearly over the integration interval, the current position $\mathbf{r}_{eb}^{e}(+)$ can be updated as:

$$\mathbf{r}_{eb}^{e}(+) = \mathbf{r}_{eb}^{e}(-) + (\mathbf{v}_{eb}^{e}(-) + \mathbf{v}_{eb}^{e}(+))\frac{\tau_i}{2} \tag{5}$$

After obtaining the navigation parameters from INS iteration, the estimation will be further integrated with the GNSS measurement data by Kalman filter.

2.3 Loosely Coupled Integration

There are various levels of GNSS/INS integration for navigation, depending on how corrections are made to the inertial navigation system, GNSS measurement types and how INS aiding GNSS with integration algorithm. Commonly, the architecture can be categorized as loosely coupled integration (LC), tightly coupled integration (TC) and ultra-tightly coupled integration (UTC). In this study, the loosely coupled integration is employed in ECEF frame for simplicity, it uses the GNSS position and velocity solution as the measurement to correct INS prediction result.

The state vector during integration includes the differences of attitude, velocity and position, also the biases of accelerometer and gyroscope are included, which makes the state vector as:

$$\mathbf{x}_{INS}^{e} = \begin{pmatrix} \delta\boldsymbol{\Psi}_{eb}^{e} \\ \delta\mathbf{v}_{eb}^{e} \\ \delta\mathbf{r}_{eb}^{e} \\ \mathbf{b}_{a} \\ \mathbf{b}_{g} \end{pmatrix} \tag{6}$$

The superscript e denotes the ECEF-frame, $\delta\boldsymbol{\Psi}_{eb}^{e}$ denotes the attitude error, $\delta\mathbf{v}_{eb}^{e}$ denotes the velocity error, $\delta\mathbf{r}_{eb}^{e}$ denotes the position error, $\mathbf{b}_a$ denotes the accelerometer biases and $\mathbf{b}_a$ denotes the gyroscope biases.

As for the assumption of Kalman filter that the time derivative of each state is a linear function of the other states and white noise sources, the dynamic model is constructed as:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}_s(t) \tag{7}$$

Where $\mathbf{x}(t)$ is the true state vector, $\mathbf{F}(t)$ is the system matrix, $\mathbf{G}(t)$ is the system noise distribution matrix and $\mathbf{w}_s(t)$ is the system noise vector, the system noise vector is assumed having a zero-mean Gaussian distribution. By applying the expected value to the dynamic model and first order approximation, the transition matrix $\boldsymbol{\Phi}_{k-1}$ for discrete Kalman filter can be derived as following:

$$\boldsymbol{\Phi}_{k-1} \approx \mathbf{I} + \mathbf{F}_{k-1}\tau_s \tag{8}$$

$$\mathbf{F}_{k-1}^{e} = \begin{pmatrix} -\boldsymbol{\Omega}_{ie}^{e} & 0_3 & 0_3 & 0_3 & \hat{\mathbf{C}}_b^e \\ \mathbf{F}_{21}^{e} & -2\boldsymbol{\Omega}_{ie}^{e} & \mathbf{F}_{23}^{e} & \hat{\mathbf{C}}_b^e & 0_3 \\ 0_3 & \mathbf{I}_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{pmatrix} \tag{9}$$

Where

$$\mathbf{F}_{21}^{e} = \left[-(\hat{\mathbf{C}}_b^e \hat{\mathbf{f}}_{ib}^{b}) \quad \wedge\right] \tag{10}$$

$$\mathbf{F}_{23}^{e} = \frac{2g_0(\widehat{L_b})}{r_{eS}^{e}(\widehat{L_b})} \frac{\hat{\mathbf{r}}_{eb}^{e}}{|\hat{\mathbf{r}}_{eb}^{e}|^2} \hat{\mathbf{r}}_{eb}^{e\ T} \tag{11}$$

$-\boldsymbol{\Omega}_{ie}^{e}$ is the Earth-rate introduced attitude differences term and $\hat{\mathbf{C}}_b^e$ is the gyroscope measurement bias term. $\mathbf{F}_{21}^{e}$ is the attitude error introduced velocity error term, $-2\boldsymbol{\Omega}_{ie}^{e}$ is the Coriolis error term, $\mathbf{F}_{23}^{e}$ is the gravitation variation term with g_0 as surface gravity and r_{eS}^{e} as geocentric radius and $\hat{\mathbf{C}}_b^e$ denotes the accelerometer biases.

The process noise covariance matrix $\mathbf{Q}_{k-1}$ is determined with the power spectral densities of the gyroscope random noise, gyroscope bias random walk, accelerometer random noise and accelerometer bias random walk respectively. By applying Allan variance analysis, the error characteristic of INS can be obtained and aid constructing the process noise covariance.

The initial error covariance matrix $\mathbf{P_0}$ is determined with the uncertainty of each variable in the state vector. For Kalman filter prediction phase, the propagation of state estimates $\hat{\mathbf{x}}_k^-$ are zero for all states by close-loop corrections. The process estimate covariance is derived as follows:

$$\mathbf{P}_k^- = \boldsymbol{\Phi}_{k-1}(\mathbf{P}_{k-1}^+ + \tfrac{1}{2}\mathbf{Q}_{k-1})\boldsymbol{\Phi}_{k-1}^{\mathrm{T}} + \tfrac{1}{2}\mathbf{Q}_{k-1} \tag{12}$$

During the measurement update phase, the measurement matrix $\mathbf{H}_k$ is determined neglecting the coupling of attitude error and gyroscope biases into measurement. The measurement noise covariance $\mathbf{R}_G$ can be constructed with the standard deviation of measurement noise per axis from position and velocity. Then, the Kalman gain can be derived as:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_G)^{-1} \tag{13}$$

The state estimates and state estimation error covariance matrix can be further updated as following

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \delta \boldsymbol{z}_{G,k}^- \tag{14}$$

$$\mathrm{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \tag{15}$$

Where $\delta \mathbf{z}_{G,k}^-$ is the difference of position and velocity between prediction and GNSS measurement. Finally, the close-loop correction is applied for the state vector of previous epoch and obtain the navigation solutions for the system.

3 Positioning Error Classification

In order the adjust the measurement noise covariance with different circumstance, it is important to classify the GNSS measurement condition first. Since Pixhawk is a well-known autopilot system collecting the GNSS information relates to its operating condition, we are able to use available GNSS features to classify the corresponding GNSS conditions. The adaptive tuning can base on difference GNSS conditions from classification.

3.1 Principle Component Analysis (PCA)

For the Pixhawk autopilot system, the GNSS measurement data containing not only position solution but also features relating to GNSS condition, which are transformed into its own format. The measurement features and corresponding meanings from Pixhawk GNSS log data are shown as Table 1. However, some of the features may have no relationship for classification and even influence the classification accuracy. Therefore, the PCA technique is employed to select major related features and aid classification.

Principle component analysis (PCA) is a statistical method to observe the main parameters from the dataset in the transformed dimension. By using the orthogonal transformation, a set of correlated parameters can be converted into a set of linear uncorrelated parameters, namely the principle components. The PCA can be employed to extract the main characteristic components of the data and have been widely used in dimensionality reduction for high dimensional data. The algorithm of PCA can be illustrate as Algorithm 1.

Table 1. Pixhawk GNSS measurement output features

GPSTime	Time data with GPS week/second format
Fix	GPS fixed mode
EPH	Standard deviation of horizontal positioning error
EPV	Standard deviation of vertical positioning error
Lat	Latitude solution
Lon	Longitude solution
Alt	Altitude solution
VelN	Velocity along N-axis in NED frame
VelE	Velocity along E-axis in NED frame
VelD	Velocity along D-axis in NED frame
nSat	Number of satellites for positioning
N	GPS noise
J	GPS jamming

Algorithm 1: Principle Component Analysis (PCA)

STEP1:	Input the raw data with k sets and n parameters as $n \times k$ matrix $\mathbf{A}$
STEP2:	Apply zero mean normalization for each of the n^{th} parameters with k data
STEP3:	Calculate the covariance matrix $\mathbf{C} = \frac{1}{k}\mathbf{A}\mathbf{A}^{\mathrm{T}}$
STEP4:	Calculate the eigenvalue and corresponding eigenvector of the covariance matrix $\mathbf{C}$
STEP5:	Construct the transformation matrix $\mathbf{T}$ sorted by the eigenvector for the first η rows according to the corresponding eigenvalue.
STEP6:	The raw data matrix $\mathbf{A}$ converts to $\mathbf{B}$ which is reduced to η dimensions with uncorrelated parameters by $\mathbf{B} = \mathbf{T}\mathbf{A}$

For a raw data with k sets and n parameters requiring dimension reduction, all of the data can be applied zero mean normalization and constructed as matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \hat{a}_{1,1} & \hat{a}_{1,2} & \cdots & \hat{a}_{1,k} \\ \hat{a}_{2,1} & \hat{a}_{2,2} & \cdots & \hat{a}_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{n,1} & \hat{a}_{n,2} & \cdots & \hat{a}_{n,k} \end{bmatrix} \tag{16}$$

Where

$$\hat{a}_{n,k} = a_{n,k} - \frac{\sum_{i=1}^{k} a_{n,i}}{k} \tag{17}$$

$\hat{a}_{n,k}$ is the element in matrix $\mathbf{A}$ after zero mean normalization, $a_{n,k}$ is the element in raw data matrix. The covariance matrix of $\mathbf{A}$ can be obtained by

$$\mathbf{C} = \frac{1}{k}\mathbf{A}\mathbf{A}^{\mathrm{T}} \tag{18}$$

The diagonal elements in $\mathbf{C}$ are the variance for each parameter, and others are the covariance. In order to apply dimensional reduction on data, the information from the raw data should be retained as much as possible, which requires the data projection of the converted dimension to be separated from each other. It means achieving a maximum variance value after dimensional reduction. However, to include much information by different dimension, it is also required each dimension is not containing repetition of information. This can be illustrated as it is not correlated between different dimensions after the dimensional reduction, which means achieving zero covariance after the reduction. Thus, the transformation matrix $\mathbf{T}$ is required to diagonalize the transformed covariance matrix, and further obtained with symmetric matrix characteristics as following, e_n is each eigenvector from the covariance matrix $\mathbf{C}$.

$$\mathbf{T} = (\mathrm{e}_1 \quad \mathrm{e}_2 \quad \cdots \quad \mathrm{e}_n)^{\mathrm{T}} \tag{19}$$

Since the PCA technique is able to reduce the dimension of a multiple variables data and convert the data into more major dimensions for evaluation, the features of GNSS has been processed with PCA to analysis the relationship between positioning error [21]. The GNSS localization condition can be separated with several classes after applying the PCA and plotted into the principle component dimensions. By selecting first three principle component as the major three dimensions, the data after dimension transformation can be visualized as Fig. 4. The color of each point indicates the corresponding GNSS localization error with ground truth. According to the PCA result, the GNSS localization can be separated with 4 class for classification as Table 2. The PCA eigenvalue and eigenvector result is shown as Table 3, hence the major feature relating to GNSS positioning error level are EPH, EPV, nSat, N and J. By learning the physical meaning of these features and tested into supervised machine learning model, the feature of EPH, EPV and N are more related to the GNSS health condition, which will be further used into supervised machine learning features.

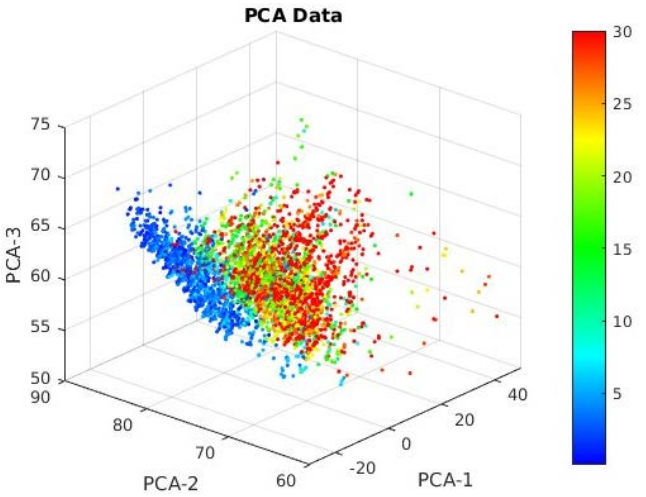

Fig. 4. The GNSS localization error distribution after PCA transformation from GNSS features, the color of each point indicates the corresponding localization error.

Table 2. Definition of GNSS condition classification

Class	Positioning error
Health (HL)	below 5 meters
Slightly shift (SS)	5 to 13 meters
Inaccurate (IA)	13 to 23 meters
Danger (DG)	over 23 meters

Table 3. PCA eigenvalue and eigenvector result

	PCA-1	PCA-2	PCA-3
EPH	0.023	0.214	0.242
EPV	0.067	0.488	0.718
VelN	0.001	0	0.029
VelE	0.003	0.011	0.021
VelD	0.004	0.012	0.053
nSat	0.038	0.331	0.208
N	0.184	0.754	0.613
J	0.976	0.193	0.051
Eigenvalue	399.0671	35.9236	9.9084

3.2 Supervised Machine Learning

For the GNSS condition classification, it is hard to derive a specific function between features and condition classes. Since the development of computer science, a feasible method is to employ machine learning technique. By training with enough amount and comprehensive data, the machine learning model can provide accurate decision and classification for GNSS condition by new coming features. The Machine Learning technique is commonly divided into three types, supervised learning, unsupervised learning and reinforcement learning. In supervised learning, the learner observes training examples with inputs and corresponding desire outputs, and then obtain a general decision rule in order to predict outputs from new inputs. For the efficiency, we employed the supervised machine learning technique to predict the GNSS positioning error level, further to aid adaptive tuning the measurement noise covariance value in integration process. In this study, the decision tree method and random forest supervised machine learning method are applied for their simplicity and relatively good accuracy.

Decision Tree

The decision tree learning technique have been widely used in supervised learning problems, it represents a model which takes the input containing attribute values as a vector and returns an output value, namely the decision. For a Boolean classification case as Fig. 5, all outputs are defined as positive or negative. The node is an arbitrary feature and the branch is the value to split the input examples. The split example with mixed output will be further treated as a node with another feature for splitting, until all the branches have pure separation with positive and negative. The major task is to explore an arrangement of feature nodes that split the examples well, the arrangement becomes a model further used to make prediction with new inputs.

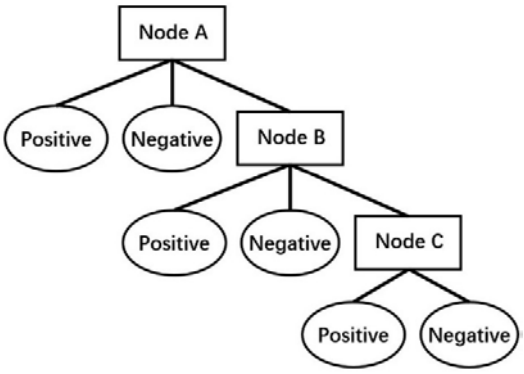

Fig. 5. Decision Tree model for positive and negative label data.

To determine the arrangement order of the nodes, the Shannon Entropy is employed to evaluate the information importance. The Shannon Entropy is the expected value of the

information containing in each of the message and is further used to calculate the information gain for determining the splitting nodes order. For GNSS positioning error level classification, the inputs are continuous value of different features, such as EPH, EPV, N etc., then the information gain of can be derived as:

$$I_k = E(S_k) - \sum_{i \in L,R} \frac{|S_k^i|}{|S_k|} E(S_k^i) \tag{20}$$

$$S_k = S_k^L \cup S_k^R \tag{21}$$

The k^{th} feature of GNSS measurement data for splitting examples is denoted as S_k, when examples are split by a specific value for a testing feature, the continuous input of that feature will be separated into two parts, left and right, denoted as S_k^L and S_k^R respectively. The function E denotes the Shannon Entropy.

The selection of the attribute, or the splitting node is to evaluate all the attributes with corresponding information gain, and then choose the attribute with highest information gain. By continuing explore and select the attribute with highest Information Gain, the node will keep splitting with branches until all examples are separated as pure output. Finally, the order of nodes is the training result for decision tree learning, which able to predict or classify the output for new input data. After training with GNSS measurement examples with positioning error label, a decision tree learning model can be obtained and employed to classify the positioning error by a given set of GNSS features.

Random Forest

The Random Forest technique is an ensemble learning method for classification, regression etc., it is processed with constructing multiple decision trees with a subset of features or examples and obtain a model for new data prediction. As for decision tree learning, it suffers from overfitting to the training set, which can be corrected by using random forest method.

For $b = 1$ to B decision trees and given training examples $X = x_1, \dots, x_n$ with corresponding output as $Y = y_1, \dots, y_n$, a bootstrap sample from X and Y as X_b and Y_b can be trained by a decision tree T_b. The training result of the subset of examples represents as a model of classification makes $Y_b = g_b(X_b)$. The final model of the random forest is the ensemble of trees as $\{T_b\}_1^B$, which made by processing average to all individual decision tree, and finally able to predict the class for the following new data input x'. The ensemble function can be derived as following:

$$\hat{g} = \frac{1}{B} \sum_{b=1}^{B} g_b(x') \tag{22}$$

The bootstrapping process improve the performance of model because the variance of the model is decreasing with it, but the bias is not increasing. Each random tree is determined by a random set of data which is highly uncorrelated, then by processing

the average of tress, the variance can be canceled and not sensitive to the noise in training sets. To determine the random forest size B, the cross-validation is able to evaluate the performance of model and found an optimal number of trees. cross-validation is to exclude a small subset of data during model training and use that subset to verify the model accuracy.

For GNSS positioning error level prediction, the random forest model is trained with several GNSS data with features as inputs and the corresponding positioning error level calculated with ground truth as the desire output. After obtaining the random forest model consisting of multiple decision trees, the new GNSS data feature will go through each tree in the forest and obtain a prediction individually for each tree. By averaging all the individual predictions of each tree, the mean value of prediction represents the final prediction from the given GNSS feature, which is able to be a continuous number between each defined class, makes the prediction closer to the true level of positioning error situation.

4 Adaptive Kalman Filter

For conventional Kalman filter of GNSS/INS integration, the state propagation is always based on fixed values of the process noise covariance matrix $\mathbf{Q}_{INS}$ and the measurement noise covariance matrix $\mathbf{R}_G$, further to determine the Kalman gain $\mathbf{K}_k$ as a static value. The invariable Kalman gain is leading to a static weighting between the estimation and measurement during propagation, regardless to all the circumstance. In most applications that the UAV operating in a known and open area, the environment for the sensors' characteristic can be treated as unchanged during operation, makes the static Kalman gain become effective. Due to the GNSS much outperforming than INS in open area, the conventional Kalman filter is always designed with high weighting for GNSS to correct INS.

However, when the UAV is used for applications close or even in the urban area, such as disaster rescuing or package delivering, it has to operation with a restricted environment with skyscrapers and obstacles. Even though the IMU has little dependency for the environment, the GNSS can be highly influenced with a harsh environment and introduce inevitable large error by multipath effect. Since the GNSS error can be quite larger than INS error, the conventional GNSS/INS integration make the navigation system correcting the result seriously based on the large error solution. Hence, the UAV may execute a wrong action with incorrect navigation and crash on nearby buildings. Then an innovative Kalman filter is required to automatically identify the environment and determine an appropriate Kalman gain between estimation and measurement from INS and GNSS, namely an adaptive Kalman filter (AKF).

4.1 System Architecture

Different from the conventional Kalman filter, the presented adaptive Kalman filter is required to identify the operation environment for GNSS measurement, and further base on the identification to tune different measurement noise covariance matrix $\mathbf{R}_G$ for integration. For this purpose, a supervised machine learning method is employed for the identification of environment, also the consistence residual and fuzzy logic are used to further avoid rapid correction and smooth changes. The architecture of the AKF is shown as Fig. 6. The IMU measurement as angular rate and specific force will be processed with navigation equation and obtain the attitude, velocity and position of the UAV, namely the estimate state. The measurement from GNSS as velocity and position are the measurement state for Kalman filter as conventional integration, however, the GNSS also output the values of serval features related to the operating environment. By collecting the data of features, a pre-trained machine learning model is employed to classify the operating environment with those features, the environment classification result will further process to obtain a particular measurement noise covariance matrix with fuzzy logic and used into the Kalman filter to achieve adaptive tuning.

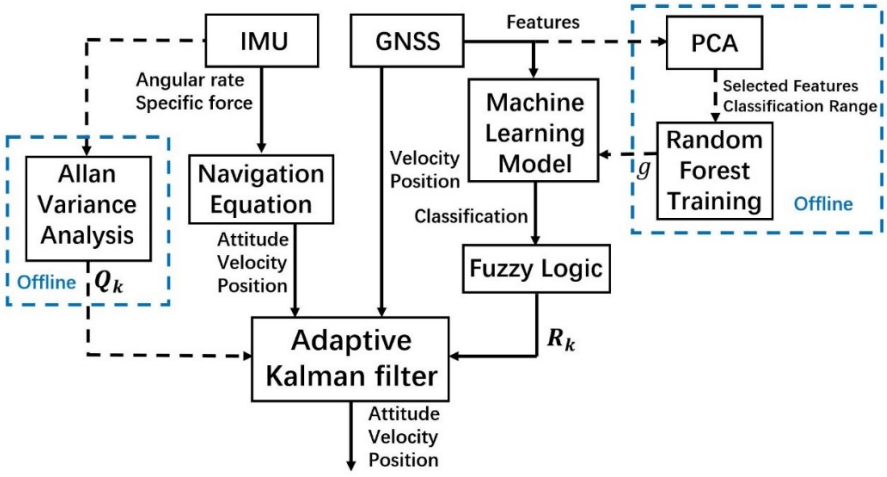

Fig. 6. The architecture for adaptive Kalman filter

4.2 Misclassification Mitigation

The fuzzy logic control can be process with steps, mainly including fuzzification interface, decision making logic and defuzzification interface. In this study, the fuzzy logic is employed to identify enormous change of GNSS condition from machine learning classification, which is abnormal for the actual GNSS operation. Hence, the last epoch GNSS condition estimation and current epoch GNSS condition classification are applied with fuzzy logic algorithm, the current epoch GNSS condition estimation is calculated based on these inputs as Fig. 7. By designing appropriate decision logic, the fuzzy logic technique can mitigate the abnormal error and smooth the output closer to

the true value. The fuzzification interface is to convert a crisp sensor value into a fuzzy singleton, the probabilistic number is converted into fuzzy numbers with the knowledge base. The fuzzification can be processed based on a linear distribution.

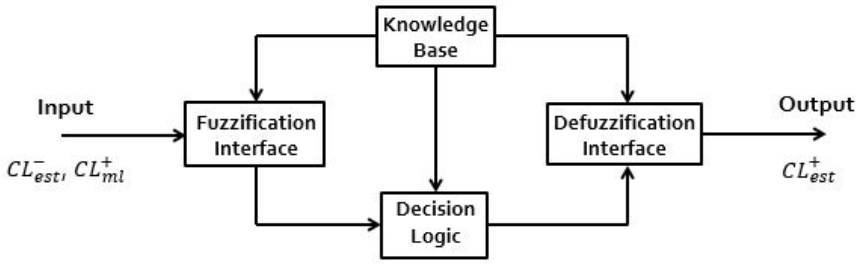

Fig. 7. Fuzzy logic fundamental procedure, CL^{-}_{est} is last epoch`s GNSS condition estimation, $\mathrm{CL}^{+}_{ml}$ is the GNSS condition classification result from machine learning model and $\mathrm{CL}^{+}_{est}$ is the current epoch`s GNSS condition estimation.

Since the fuzzy logic is designing to employ previous epoch classification into current epoch GNSS positioning error level classification, the data base for determination is designed as Fig. 8 with 4 classification sets from PCA. The four classes are defined as 1) health (HL); 2) slightly shift (SS); 3) inaccurate (IA); 4) danger (DG) for the error level of previous epoch estimation, current epoch measurement and the current epoch estimation.

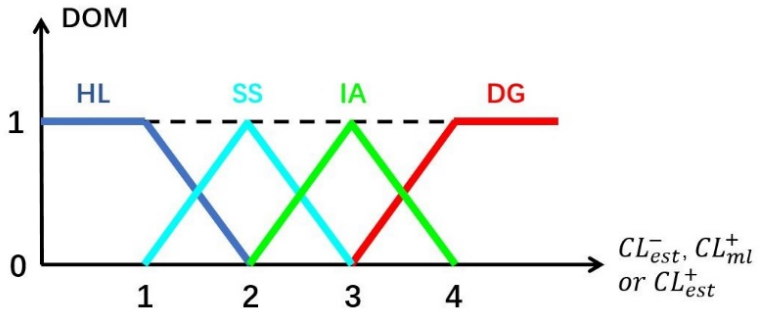

Fig. 8. Fuzzy logic data base, DOM is the degree of membership.

Table 4. The fuzzy logic rule base

CL_{est}^{-} \ CL_{ml}^{+}	HL	SS	IA	DG
HL	HL	HL	SS	SS
SS	HL	SS	SS	IA
IA	SS	IA	IA	DG
DG	IA	IA	DG	DG

To determine the classification by both previous epoch and current epoch, a rule base is designed as Table 4, the CL_{est}^{+} can be calculated with the experience or expectation. For the CL_{est}^{-} and CL_{ml}^{+} are continuous values transferred as a fraction of different class, each of the combination of the two epochs` condition is able to determine a class. The determination is based on the fraction for last epoch estimation and current epoch prediction, which is the degree of fulfillment of the rule promise, derived as:

$$\alpha_k = \mu_k(CL_{est}^{-}) \wedge \mu_k(CL_{ml}^{+}) \tag{23}$$

Where μ_k denotes the degree of membership for each input with rule k. The ensemble of all possible combinations is the distribution of each situation, the defuzzification process is to calculate the output value from the distribution by following equation:

$$CL_{est}^{+} = \frac{\alpha_1 s_1 + \alpha_2 s_2 + \alpha_3 s_3 + \alpha_4 s_4}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4} \tag{24}$$

Where s denotes the value of different classes, the CL_{est}^{+} is a weighted average output estimation result. Hence, the fuzzy logic technique can correct the machine learning misclassification error via the designed rule base.

4.3 Adaptive Tuning

For Kalman filter, the measurement noise covariance R determines the confidence coefficient between GNSS and INS, which will further influence the integrated localization performance. A relatively small R make the system more rely on the GNSS measurement, while a large R value makes the system neglect GNSS measurement and more rely on INS result. A fine-tuned or appropriate R value can help to improve the performance of localization, hence an adaptive algorithm for determination of R is developed. By testing the integration algorithm with fixed value, an appropriate R value for different situation as HL, SS, IA, DG from fuzzy logic can be found. Since the output classification for GNSS positioning error level is a continuous value, the value in between is based on linear distribution from 1 to 3.5. The

level over a threshold of 3.5 is considered as a severe error case, which sets an enormous R to neglect the GNSS measurement for localization. The relationship between fuzzy logic classified value CL_{est}^{+} and R is shown as Fig. 9.

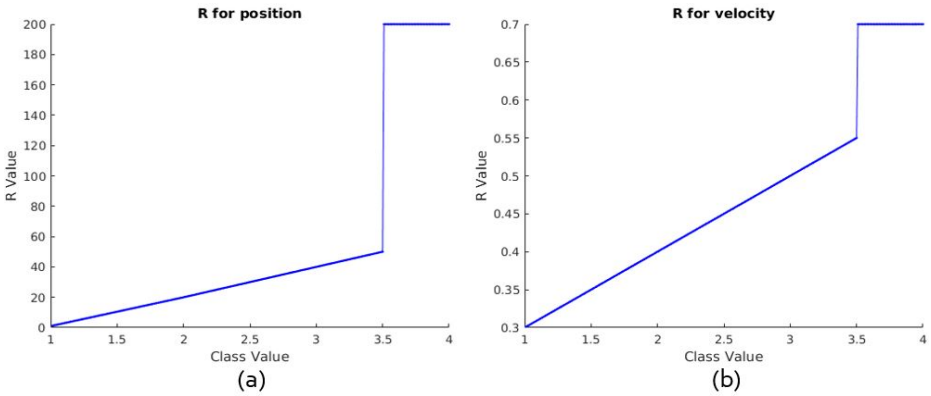

Fig. 9. The measurement noise covariance matrix R value for (a) position and (b) velocity in adaptive GNSS/INS integration with regarding to supervised machine learning prediction class

5 Experiment Result

5.1 Experiment Setup

In this study, the UAV platform is a 99 x 99 mm size quadrotor UAV. The autopilot hardware is using the Pixhawk 2 and the GNSS receiver is using the Ublox NEO-M8N GNSS Modular as Fig. 10. During a flight, the autopilot system will store all the flight information as log file, including the IMU raw measurement data, GNSS measurement data and other sensors. The verification of the proposed adaptive Kalman filter performance is to post-process the raw data from the log file with corresponding time. The raw collected data as IMU data and GNSS positioning data will be processed as the input for Kalman filter, the corresponding GNSS feature data is processed as the input for supervised machine learning prediction. The autopilot system also equips with an inboard GNSS/INS integration algorithm, which can be compared with the proposed localization algorithm regarding to the ground truth for evaluation.

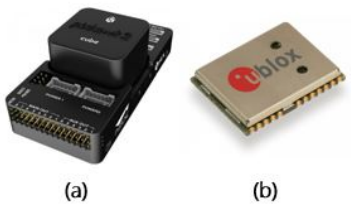

Fig. 10. (a) Pixhawk 2 (Autopilot hardware); (b) Ublox NEO-M8N GNSS modular.

The experiment is setup in the Tsim Sha Tsui, Hong Kong where has a great amount of skyscrapers, in order to evaluate the performance of the proposed adaptive Kalman filter for dense urban area. Since the UAV is banned for real flight in the urban area in Hong Kong, the UAV is lifted by human and moved following a designed trajectory. To verify the performance, one trajectory with open sky scenarios and one trajectory with building surroundings are designed.

5.2 Open Sky Localization Result

To evaluate the performance of the proposed adaptive Kalman filter for normal operation, an open sky GNSS operation scenario is designed to verify the AKF is able to maintain the precise localization performance with accurate GNSS measurement. The designed trajectory is shown as Fig. 11 with the key locations shown in Table 5. The operation environment is clean without building surrounding, which has nearly noun multipath effect. The trajectory is from START to END marker as a straight line.

Table 5. Key location for open sky localization scenario

Key Location	START	END
Latitude	22.297977^{o}	22.297457^{o}
Longitude	114.179264^{o}	114.178622^{o}

Fig. 11. Experiment trajectory for open sky localization scenario.

5.3 Urban localization Result

The experiment for evaluating the performance of proposed adaptive Kalman filter localization in urban is designed as Fig. 12 with key locations as Table 6. The trajectory is begin from START and straight to TURNING point, then the platform will experienced an 180^{o} turning and following the same route to the END. The buildings are distributing along two sides of the trajectory.

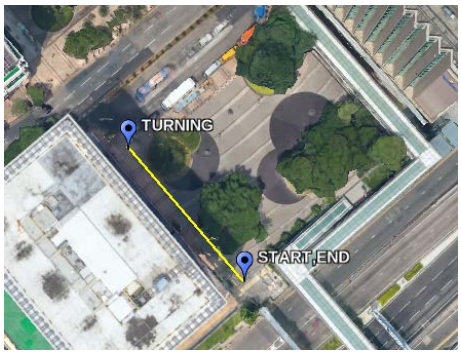

Fig. 12. Experiment trajectory for urban localization scenario.

Table 6. Key location for urban localization scenario

Key Location	START, END	TURNING
Latitude	22.298304°	22.298659°
Longitude	114.178896°	114.178554°

For the urban localization scenario, the localization result is shown as different trajectories for different method (Fig. 13), including the ground truth (Truth), the localization result from Pixhawk original integration (PX4), the localization with a tuned fixed R value (FR), the localization result from AKF with decision tree learning model (DT), the localization result from AKF with random forest model (RF) and the localization result from random forest and fuzzy logic (RFFL). Comparing with the ground truth, the RF and RFFL method achieve better localization trajectory.

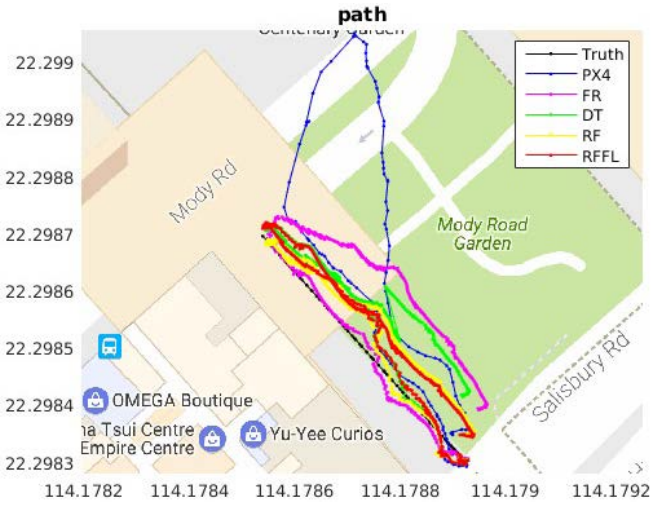

Fig. 13. Localization resulting trajectories for urban localization scenario.

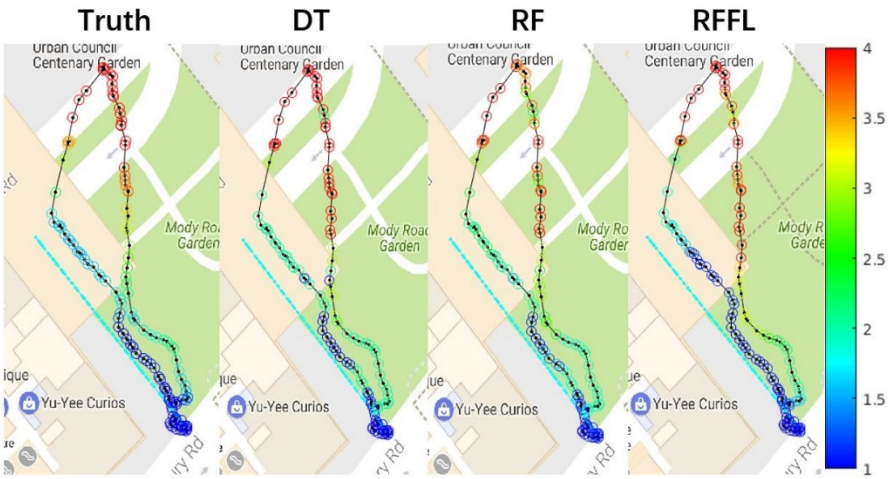

Fig. 14. GNSS positioning error level classification result on GNSS trajectory.

The classification result of supervised machine learning model is shown as Fig. 14, comparing with the truth classification, decision tree, random forest and random forest with fuzzy logic. The classification distribution along GNSS epoch is also compared with Fig. 15. The result shows the DT model only output the fixed number for the GNSS positioning error level, while the random forest is able to predict a continuous number closer to the truth. The fuzzy logic can mitigate the large sudden change of the GNSS error level, making the result more satisfied to the real distribution of GNSS condition. The proposed RFFL model can achieve a better classification result than all above.

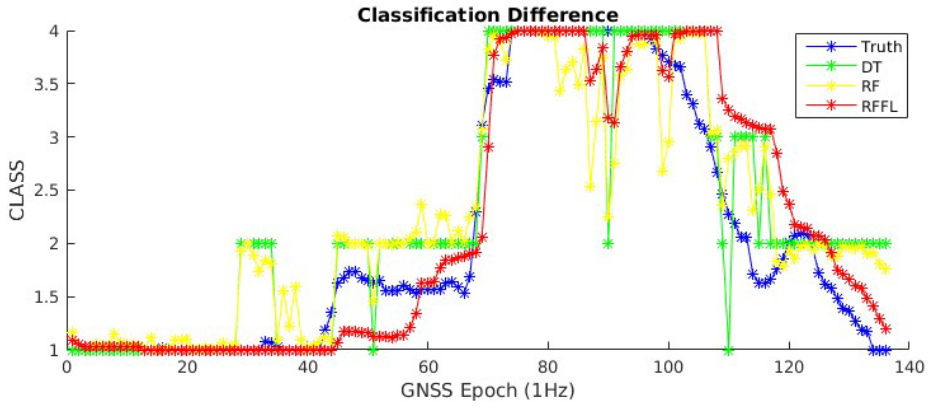

Fig. 15. GNSS positioning error level classification distribution during operation.

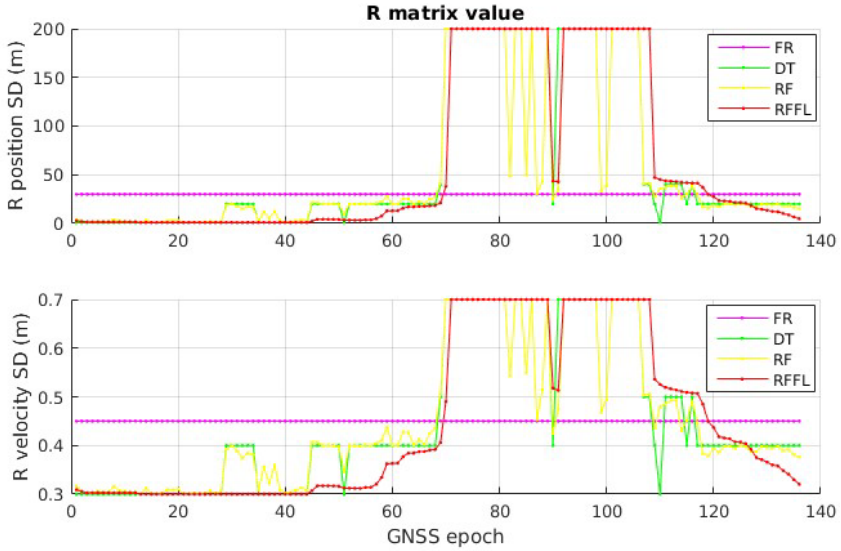

Fig. 16. The measurement noise covariance matrix R value distribution during operation for position and velocity.

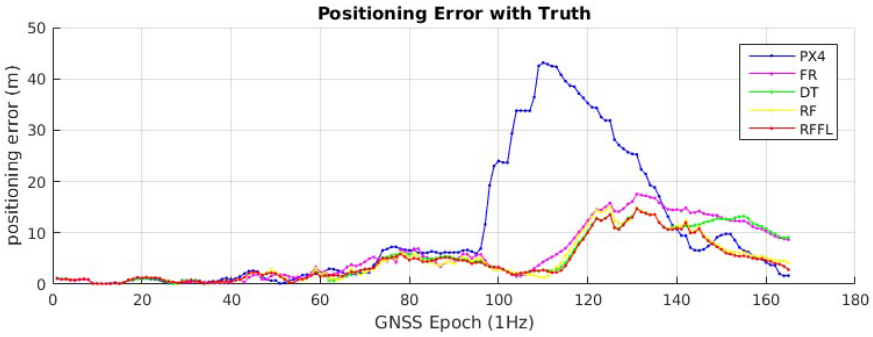

Fig. 17. The localization error distribution for different algorithm in urban localization scenario.

The corresponding measurement noise covariance matrix value R for position and velocity are shown as Fig. 16 respectively, the FR condition denotes the integration using a fixed R value in order to maintain a specific confident coefficient between GNSS and INS as conventional Kalman filter, while DT, RF and RFFL are using supervised machine learning algorithm to adaptively adjust R value. From the result, FR maintains a specific value while other algorithms adjust the R with a small value for good GNSS condition and a large value for bad GNSS condition respectively.

The localization error distribution with respect to the ground truth during operation is shown as Fig. 17, the mean error and STD are shown in Table 7. The original Pixhawk localization result experienced a large localization error exceeds 40 meters, having a mean value as 10.2 m and STD as 12.7 m, which is dangerous for UAV operation in urban area. The manually tuned R value can decrease the localization error highly, but still not appropriate for all the conditions during operation. The DT model use supervised machine learning algorithm to adjust the R value adaptive to the GNSS measurement condition and achieve lower positioning error and RF model improves the classification accuracy of DT model. The proposed RFFL adaptive Kalman filter algorithm is able to achieve better GNSS measurement condition prediction with its characteristics and mitigate the large sudden change for GNSS error level, it also helps to mitigate the mis-classification introduced error during operation. The proposed algorithm sustains the localization error under 15 meters with mean value as 4.2 m and STD as 3.9 m, which improves the conventional localization method over twice, achieves a safer UAV operation in urban area.

Table 7. Localization error mean value and STD for urban localization B

	PX4	FR	DT	RF	RFFL
Mean Localization Error (m)	10.2	5.8	5.1	4.4	4.2
STD (m)	12.7	5.4	4.7	4.2	3.9

6 Conclusion

In this study, the conventional Kalman filter for UAV localization is improved to obtain a better localization solution for operations in urban area, where has a large multipath error on GNSS positioning solution and may crash on obstacles with the wrong position. The Adaptive Kalman Filter is developed to automatically adjust the measurement noise covariance with the GNSS measurement features, to mitigate the GNSS large localization error by multipath effect. The Allan Variance Analysis technique is employed to study the characteristics of the IMU sensors on experiment platform, obtaining the random walks and bias instability values for sensors, further adjust the process noise covariance matrix in the proposed adaptive Kalman filter as power spectral density. To evaluate the GNSS measurement condition, the supervised machine learning algorithm is used to predict the GNSS condition with a pre-defined error evaluation level. The supervised machine learning in this study is including the decision tree learner and random forest, each of the model is trained with 5 days' data covered almost all possible situations in the experiment area as TST, Hong Kong. The supervised machine learning models can classify most of the GNSS measurement condition correctly. Then, the fuzzy logic is further developed to mitigate the large sudden change of GNSS measurement condition from machine learning models, by considering the current epoch and last epoch classification result, the final classification

result can be smooth to mitigate the sudden change and large classification error, which can improve the accuracy for evaluating the current GNSS measurement condition for GNSS/INS integration. The classification result from fuzzy logic is further linear corresponding to the measurement noise covariance into the proposed adaptive Kalman filter, with a threshold to input a great R value to estimate the position regardless to the GNSS measurement and mainly based on the INS estimation. By cooperating an appropriate process noise covariance matrix and an adaptive measurement noise covariance matrix from GNSS feature, the proposed integration algorithm is able to adjust the confident coefficient between GNSS and INS from learning the feature of measurement accuracy, improving the localization solution in urban area. From the experiment results, the proposed adaptive Kalman filter for GNSS/INS integration of UAV platform can effectively identify the GNSS error conditions with multipath effect, and improve the localization result for UAV, which help the UAV applications ensure the safety for urban operations.

However, the presented GNSS/INS integration algorithm with adaptive Kalman filter still has the following drawbacks: 1) the supervised machine learning model still occurs some mis-classification cases and may influence the localization accuracy of adaptive Kalman filter; 2) the proposed algorithm is still based on post-processing rather than real-time operation. Hence, the future works are including improving the supervised machine learning algorithm to achieve lower out of sample error or develop with other improved machine learning algorithm for more accurate classification result. The proposed algorithm is also able to be developed into real-time localization with UAV platform and verify the performance with on-board operation.

References

1. Erdelj, M. and E. Natalizio. UAV-assisted disaster management: Applications and open issues. in 2016 International Conference on Computing, Networking and Communications (ICNC). 2016.
2. Cistone, J., Next Century Aerospace Traffic Management: The Sky is No Longer the Limit. Journal of Aircraft, 2004. **41**(1): p. 36-42.
3. Marris, E., FLY, AND BRING ME DATA. Nature, 2013. **498**(7453): p. 156-158.
4. Hsu, L.T., et al., Multiple Faulty GNSS Measurement Exclusion Based on Consistency Check in Urban Canyons. IEEE Sensors Journal, 2017. **17**(6): p. 1909-1917.
5. Chiang, K.-W., T. Duong, and J.-K. Liao, The Performance Analysis of a Real-Time Integrated INS/GPS Vehicle Navigation System with Abnormal GPS Measurement Elimination. Sensors, 2013. **13**(8): p. 10599.
6. Wendel, J., et al., An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter. Aerospace Science and Technology, 2006. **10**(6): p. 527-533.
7. Park, J., et al., Vision-based SLAM system for small UAVs in GPS-denied environments. Journal of Aerospace Engineering, 2011. **25**(4): p. 519-529.
8. Bryson, M. and S. Sukkarieh, Building a Robust Implementation of Bearing‐only Inertial SLAM for a UAV. Journal of Field Robotics, 2007. **24**(1‐2): p. 113-143.
9. Groves, P.D., Principles of GNSS, inertial, and multisensor integrated navigation systems. 2013: Artech house.

10. Chowdhary, G. and R. Jategaonkar, Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter. Aerospace science and technology, 2010. **14**(2): p. 106-117.
11. Almagbile, A., J. Wang, and W. Ding, Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration. Journal of Global Positioning Systems, 2010. **9**(1): p. 33-40.
12. Hajiyev, C. and H.E. Soken, Robust Adaptive Kalman Filter for estimation of UAV dynamics in the presence of sensor/actuator faults. Aerospace Science and Technology, 2013. **28**(1): p. 376-383.
13. Russell Stuart, J. and P. Norvig, Artificial intelligence : a modern approach. 3rd ed. 2010, Upper Saddle River, N.J: Prentice Hall.
14. Quinlan, J.R., Induction of decision trees. Machine Learning, 1986. **1**(1): p. 81-106.
15. Ho, T.K. Random decision forests. in Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. 1995. IEEE.
16. Jantzen, J., Foundations of fuzzy control : a practical approach. 2013, Wiley: Chichester, West Sussex, United Kingdom.
17. Kaplan, E. and C. Hegarty, Understanding GPS: principles and applications. 2005: Artech house.
18. Misra, P. and P. Enge, Global Positioning System: Signals, Measurements and Performance Second Edition. Massachusetts: Ganga-Jamuna Press, 2006.
19. Parkinson, B.W. and P.K. Enge, Differential gps. Global Positioning System: Theory and applications., 1996. **2**: p. 3-50.
20. Veitsel, V.A., A.V. Zhdanov, and M.I. Zhodzishsky, The Mitigation of Multipath Errors by Strobe Correlators in GPS/GLONASS Receivers. GPS Solutions, 1998. **2**(2): p. 38-45.
21. Drawil, N.M., H.M. Amar, and O.A. Basir, GPS Localization Accuracy Classification: A Context-Based Approach. IEEE Transactions on Intelligent Transportation Systems, 2013. **14**(1): p. 262-273.