UDC 681.5 (045)

[1]M. P. Mukhina,
[2] A. P. Prymak

GAUSSIAN PARTICLE FILTERING BLOCK OF NAVIGATION DATA

Educational and Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine
E-mails: [1]mukhina@nau.edu.ua, [2]primak.artem@gmail.com

Abstract**—** *A filtering block has been developed with the help of which navigation data, received from UAV is filtered. UAV motion over some territory with camera on board has been held and images are captured from it. Images were processed by blob method and we got some linear references, filtered it and compared with Google maps. As a result is received probabilities of state equation, in particular transient probabilities and its approximations according to the flight task, which allows to estimate UAV location, which showed that block is appropriate for use on UAV board and due to applying filtering, gives as correct results of location determining as possible.*

Index Terms—BLOB method; filtering block; Google maps; Gaussian particle filtering.

I. INTRODUCTION

Defining as precisely as possible the coordinates of the UAV on the measurement sequence generated by radar and electro-optical system is a central task of any navigation system. To solve it, significant number of algorithms is developed, based mainly on the well-known recursive algorithm for the Kalman filter, and effectively implemented on a computer. However we still can't regard this problem finally solved. This is due to many factors, and one of the most important is the nonlinearities of the motion models and measurement in many practical problems. Nonlinearity occurs for many reasons – because of the nonlinear connection of coordinate systems used in the equations of the observed object and the measurer, because of the nonlinear nature of the equations themselves. Nonlinear problems arise in the construction of adaptive systems, implemented by the inclusion of uncertain parameters in the estimated state vector. Ignoring nonlinearities and extreme simplification of the situation may significantly reduce the efficiency of coordinates and velocities estimation algorithms in the real systems. In practice, non-linear estimation algorithms are applied, but in general, limited to simple options such as extended Kalman filter.

At the same time more powerful algorithms exist but are rarely used because they require large computational cost. However, the rapid growth of computer technology opportunities during the past years enables us to use many of these algorithms in practice. So, it is needed to build such algorithm that could filter data with lower computational cost and higher accuracy. For this purpose Gaussian particle filter is developed.

II. PROBLEM STATEMENT

Nonlinear filtering problems arise in many fields including statistical signal processing, economics, statistics, biostatistics, and engineering such as communications, radar tracking, sonar ranging, target tracking, and satellite navigation. The problem consists of estimating a possibly dynamic state of a nonlinear stochastic system, based on a set of noisy observations. Many of these problems can be written in the form of the so-called dynamic state space (DSS) model. The DSS model represents the time-varying dynamics of an unobserved state variable x_n, as the distribution $p(x_n \mid x_{n-1})$, where n indicates time (or any other physical parameter). The observations y_n in the application are usually noisy and distorted versions of x_n. The distribution $p(y_n \mid x_{n-1})$ represents the observation equation conditioned on the unknown state variable x_n, which is to be estimated. Alternatively, the model can be written as

$$x_n = f\left(x_{n-1}, u_n\right), \tag{1}$$

$$y_n = h(x_n, v_n), \tag{2}$$

where $f(\ )$ and $h(\ )$ are some known functions, and u_n and v_n are random noise vectors of given distributions. The process equation represents a system evolving with time, where the system is represented by the hidden state, and the prior knowledge of the initial state is given by the probability distribution. Our aim is to learn more about the unknown state variables, given the observations as time evolves. We denote by $x_{0:n}$ and $y_{0:n}$ the signal and observations up to time n, respectively, i.e., $x_{0:n} \equiv \{x_0,\dots, x_n\}$ and $y_{0:n} \equiv \{y_0,\dots, y_n\}$. In a Bayesian context, our aim is to estimate recursively in time

– the filtering distribution $p(x_n \mid y_{0:n})$ at time n given all the observations up to time n;

– the predictive distribution $p(x_{n+1} \mid y_{0:n})$ at time given n all the observations up to time n.

From these distributions, an estimate of the state can be determined for any performance criterion suggested for the problem. The filtering distribution or the marginal posterior of the state at time n can be written as

$$p(x_n \mid y_{0:n}) = C_n p(x_n \mid y_{0:n-1}) p(y_n \mid x_n), \tag{3}$$

where C_n is the normalizing constant given by

$$C_n = \left(\int p(x_n \mid y_{0:n-1}) p(y_n \mid x_n) dx_n\right)^{-1}. \tag{4}$$

Furthermore, the predictive distribution can be expressed as

$$p(x_{n+1} \mid y_{0:n}) = \int p(x_{n+1} \mid x_n) p(x_n \mid y_{0:n}) dx_n. \tag{5}$$

The versatility of the filter can be improved if the restrictive assumption of additive Gaussian noise made in the EKF like filters is removed [1].

The Gaussian particle filter approximates the filtering and predictive distributions by Gaussian densities using the particle filtering methodology. The basic idea of Monte Carlo methods is to represent a distribution $p(x_n)$ of a random variable x_n by a collection of samples (particles) from that distribution. M particles $X = \{x_n^{(1)}, \ldots, x_n^{(M)}\}$ from a so-called importance sampling distribution $\pi(x_n)$ (which satisfies certain conditions) are generated. The particles are then weighted as $\omega^{(j)} = (p(x_n^{(j)})) / (\pi(x_n^{(j)}))$. If $W = \{\omega^{(1)}, \ldots, \omega^{(M)}\}$, then the set $\{X, W\}$ represents samples from the posterior distribution $p(x_n)$. Monte Carlo integration suggests then that the estimate of

$$E_p(g(x_n)) = \int g(x_n) p(x_n) dx_n, \tag{6}$$

can be computed as:

$$\hat{E}_p(g(x_n)) = \frac{\sum_j \omega^{(j)} g(x_n^{(J)})}{\sum_j \omega^{(j)}}. \tag{7}$$

Using the Strong Law of Large Numbers, it can be shown that:

$$\hat{E}_p(g(x_n)) \to E_p(g(x_n)), \tag{8}$$

almost surely as $M \to \infty$. The posterior distribution can be approximated as:

$$p(x_n)dx_n = P(dx_n) \approx \frac{\sum_{j=1}^{M} \omega^{(j)} \delta_{x_n^{(j)}}(dx_n)}{\sum_{j=1}^{M} \omega^{(j)}}, \tag{9}$$

where $\delta_{x_n^{(j)}}(dx_n)$ is the Dirac delta function. In the sequel, the density of a Gaussian random variable x is written as

$$\mathcal{N}(x; \mu, \Sigma) = (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right), \tag{10}$$

where the m-dimensional vector μ is the mean, and the covariance is the positive definite matrix Σ. Assume that at time $n = 1$ we have $p(x_1 \mid y_0) = \mathcal{N}(x_1; \bar{\mu}_1, \bar{\Sigma}_1)$ where $\bar{\mu}_1$ and $\bar{\Sigma}_1$ are chosen based on prior information. As new measurements are received [2], measurement and time updates are performed to obtain the filtering and predictive distributions as discussed in the following sections.

III. Problem Solution

In this work it was performed a UAV flight over some region in Kyiv. A UAV has an on-board camera and standard set of measurement devices, such as magnetic compass, gyros, GPS, accelerometers and so on. A video was recorded from on-board camera with resolution 3840×2160 pixels and frequency 30 frames / s. It was selected a row of 20 frames, captured each second (each 30th frame). So it was simulated only 20 s. of the flight with constant velocity of ~ 9.7 m/s (Fig. 1).

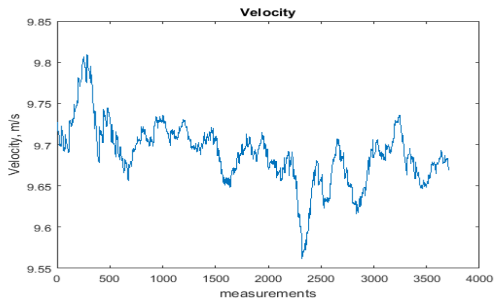

Fig. 1. Average velocity

On an image below it is represented the trajectory of UAV motion (Fig. 2).

Fig. 2. Trajectory of motion

In work it was chosen linear reference as the main feature to detect course. Linear reference is good for this purpose because of its linearity. A road was chosen as a linear reference [3], we can determine road heading in terms of the geographic coordinates. It was chosen a Google Earth app to find the heading of the road due to the presence a ruler, which can measure not only distance, but a heading as well. So, a part of trajectory with constant speed and a straight road on the captured images from camera was chosen. Illustration below represents the main road, selected as linear reference with heading of the road in respect to the North (Fig. 3).

Fig. 3. Heading of the main road

The heading of the road is 197.12 deg., the UAV moves exactly along the road and the arrow on the image above coincides with drone's course [4].

The bock developed during this work is quite simple. The simulation of block's operation was done in MATLAB R2016b. First of all it was needed to load flight data log, recorded by drone during flight it is an array, which has realization of 27 parameters.

Below it is represented 1st image of the trajectory with a parallel road. On the left it is shown an image, captured from camera and on the left it is shown the same image, but it was binarized by the program in order to detect BLOBs (Fig. 4).

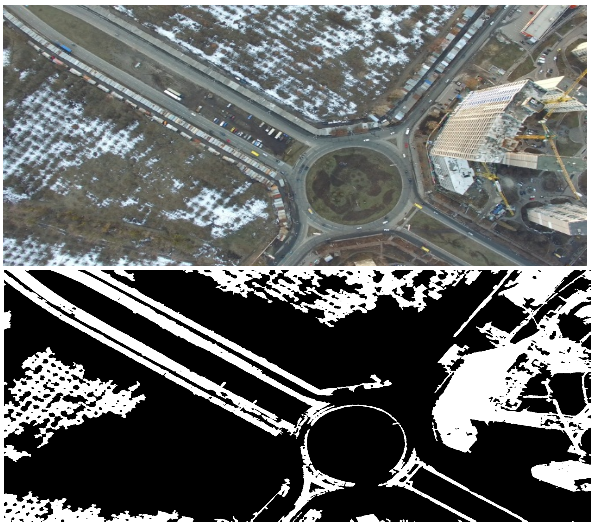

Fig. 4 Normal and binary images

After detecting blobs it is necessary to choose only linear objects, in our case it is a road, so we should define blob with parameters, which fit our criteria (Figs 5 and 6).

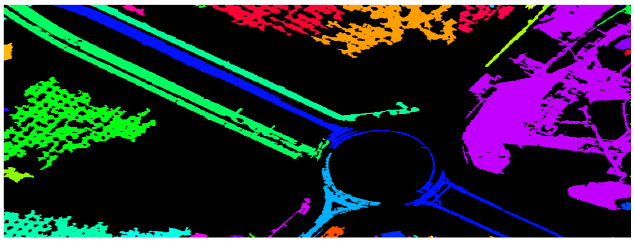

Fig. 5. All detected blobs

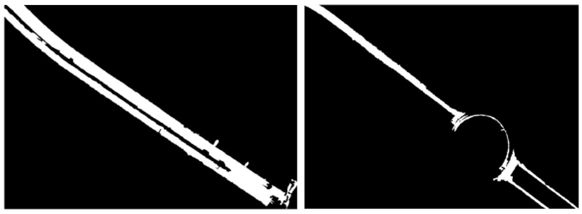

Fig. 6. Detected blobs with satisfactory parameters

Images above prove that the method works and detects objects, which can be used as linear references. Each blob has its own orientation, area, centroid and eccentricity, these parameters helps to select needed objects and to dismiss any other objects, that don't look like linear reference.

Matlab function *regionprops* can show properties of each object on a binary image. Objects 1 and 13 are fit our purpose, Area is quite big (large object), Eccentricity is very close to 1, it means, that it is almost straight line (linear reference). We will use Orientation, because this work is made to calculate and filter course of the drone (Fig. 7).

Fields	Area	Centroid	Eccentricity	Orientation	FilledImage
1	279449	[932.7447 7...	0.9973	-36.0310	1454x1992 logi...
2	367	[7.2698 130....	0.7690	89.5595	34x19 logical
3	7019	[24.8047 43...	0.9444	86.7799	161x56 logical
4	671	[12.9613 97...	0.6369	-41.7778	35x29 logical
5	280067	[391.2740 1....	0.6696	4.5772	867x936 logical
6	38	[2.1316 1.26...	0.9716	81.2226	18x5 logical
7	4	[1 1.5025e+...	0.9682	90	[1; 1; 1; 1]
8	12340	[73.6837 1.5...	0.8006	-22.3383	149x183 logical
9	2	[1 1.6325e+...	0.8660	90	[1; 1]
10	84	[3.1786 1.90...	0.9213	89.7432	17x6 logical
11	91284	[282.6024 2....	0.9528	-7.5267	256x672 logical
12	337	[159.0237 6....	0.8665	-0.5490	14x31 logical
13	239755	[1.7347e+03...	0.9975	-37.5314	2160x3136 logi...

Fig. 7 Parameters of blobs

There is an issue with found Orientation, it is an object's orientation in terms of the image, but not in terms of geographic coordinates, so this parameter was re-calculated to geographic coordinates. The heading of each blob was calculated for all 20 points and the next image illustrates calculated drone's course (blobs' heading) and course, measured by magnetic compasses (Fig. 8).

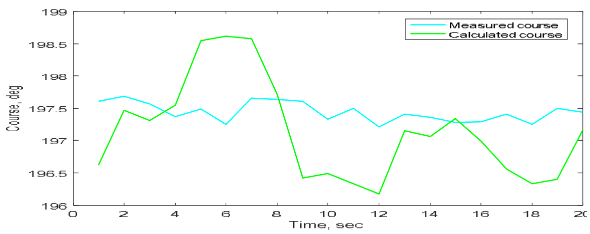

Fig. 8. Calculated vs. measured course

As it is clearly seen the calculated heading is not accurate in respect to measured data, so it should be filtered. Here Gaussian Particle Filtering (GPF) approach is proposed to be used. So, on the next step the program filters calculated data by GPF, a priori knowledge of true course obtained from a magnetic compass, so GPF in this case is one of the best solutions. Here it is on the illustration below we may see, that particle filter approximated the data well and on the next steps results will be more and more accurate [5].

The only thing that remains unknown yet is the error of estimation, it is easy to compute, so estimation error looks like (Figures 9 – 11).

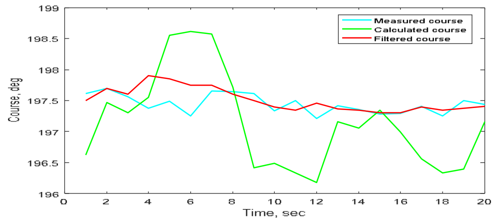

Fig. 9. Filtered course

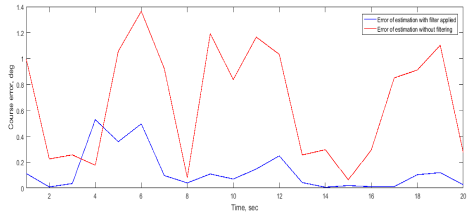

Fig. 10. Error of estimation

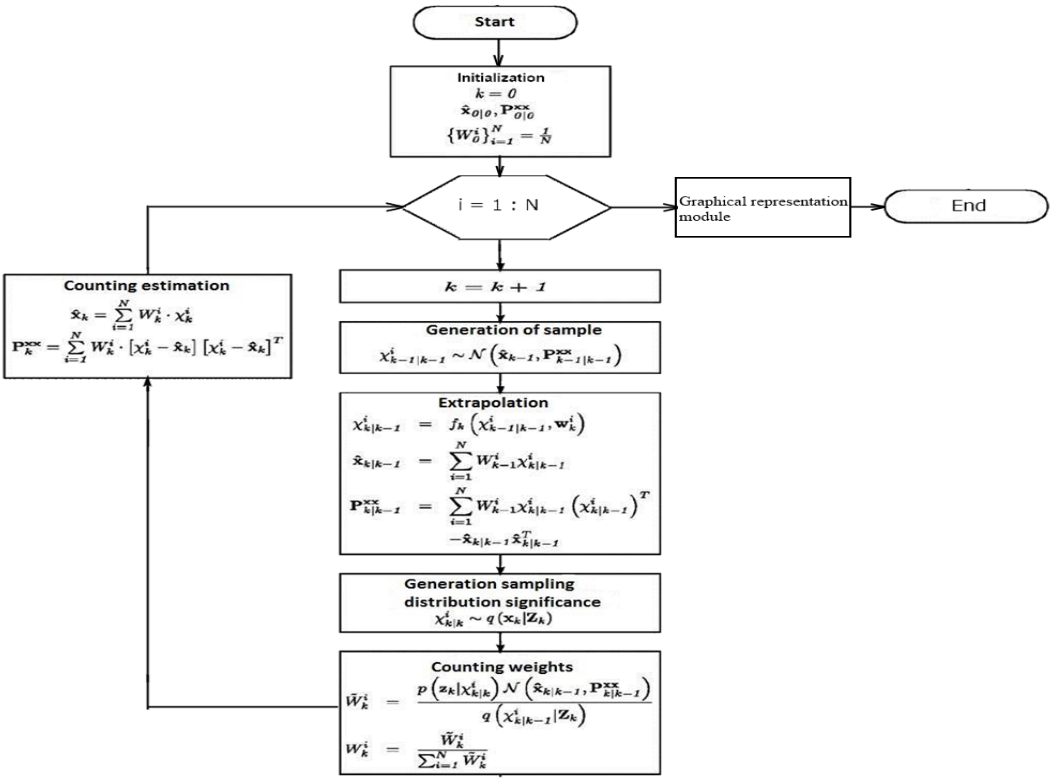

Fig. 11. Block diagram of Gaussian particle filter algorithm

IV. CONCLUSIONS

A filtering block was designed to capture images from on-board camera and find Binary Large Objects, which are the realization of linear references. A great work was made on getting and computing parameters of objects, which were detected on an image. Filtering algorithm approximated data, received from camera, taking into account a priori knowledge of a course, received from a navigation log.

Finally the proposed algorithm of Gaussian particle filter has got its realization in MATLAB developing tool and successfully tested and results were estimated. This algorithm can be implemented in UAVs as an efficient tool for trajectory and course tracking.

REFERENCES

[1] Jayesh H. Kotecha and Petar M. Djuric, Gaussian Particle Filtering, 2003. – T.P. – pp. 2592–2601.

[2] Sebastian Thrun, Dieter Fox, Wolfram Burgard, Probabilistic Robotics, 1999-2000, 477 p.

[3] Francisco José Curado Mendes Teixeira, Terrain-Aided Navigation and Geophysical Navigation of Autonomous Underwater Vehicles, 2007, 131 p.

[4] N.K. Filyashkin, V.A. Rogozhin, A.V. Skripets, T.I. Lukinova, Inertial-satellite navigation systems: Manual, 2007, 340 p. (in Ukrainian)

[5] Niclas Bergman, Recursive Bayesian. Estimation Navigation and Tracking Applications, 1999, 205 p.

Received October 11, 2016

Mukhina Maryna. Doctor of Engineering Science. Associate Professor.
Institute of Information Diagnostic Systems, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine (2002).
Research area: images processing, navigation and control.
Publications: 40.
E-mail: m_mukhina@inbox.ru

Prymak Artem. Post-graduate student.
National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, Master (2016).
Research interests: adaprtive control systems.
Publications: 4.
E-mail: primak.artem@gmail.com

М. П. Мухіна, А. П. Примак. Блок гауссової фільтрації частинок навігаційних даних

Пропонується розроблений блок гаусівської точкової фільтрації для навігаційних даних БПЛА. В ході руху БПЛА були захоплені знімки рельєфу місцевості камерою. Дані захоплені зображення були оброблені BLOB методом. Результат обробки дозволив виявити відфільтровані характерні точки і порівняти з зображеннями, отриманих з Google карт. За результатами отриманих рівнянь імовірнісних станів і їх апроксимація та подальшого дослідження можна зробити висновок про те, що за рахунок застосування пропонованого блоку фільтрації по польотному завданню можна точно визначити місце розташування БПЛА.

Ключові слова: Метод BLOB; блок фільтрації; Google карти; гаусівська фільтрація частинок.

Мухіна Марина Петрівна. Доктор технічних наук. Доцент.
Інститут інформаційно-діагностичних систем, Національний авіаційний універститет, Київ, Україна.
Освіта: Національний авіаційний універститет, Київ, Україна, (2002).
Напрямок наукової діяльності: обробка зображень, навігація та керування.
Кількість публікацій: 40.
E-mail: m_mukhina@inbox.ru

Примак Артем Петрович. Аспірант.
Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, Магістр (2016).
Напрямок наукової діяльності: адаптивні системи управління.
Кількість публікацій: 4.
E-mail: primak.artem@gmail.com

М. П. Мухина, А. П. Примак. Блок гауссовой фильтрации частиц навигационных данных

Предлагается разработанный блок гауссовской точечнойфильтрации для навигационных данных БПЛА. В ходе движения БПЛА были захваченные снимки рельефа местности камерой. Данные захваченные изображения были обработаны BLOB методом. Результат обработки позволил выявить отфильтрованные характерные точки и сравнить с изображениями, полученных из Google карт. По результатам полученных уравнений вероятностныхсостояний и их аппроксимация и дальнейшего исследования можно сделать вывод о том, что за счет применения предлагаемого блока фильтрации по полетному заданию можно точно определить месторасположения

Ключевые слова: Метод BLOB; блок фильтрации; Google карты, гауссова фильтрация частиц.

Мухина Марина Петровна. Доктор технический наук, доцент.
Институт информационно-диагностических систем, Национальный Авиационный Университет, Киев,Украина.
Образование: Национальный авиационный универститет, Киев, Украина, (2002).
Направление научной деятельности: обработка изображений, навигация и управление.
Количество публикаций: 40.
E-mail: m_mukhina@inbox.ru

Примак Артем Петрович. Аспирант.
Национальный авиационный университет, Киев, Украина.
Образование: Национальный авиационный университет, Киев, Украина, Магистр (2016).
Направление научной деятельности: адаптивные системы управления.
Количество публикаций: 4.
E-mail: primak.artem@gmail.com