Trainable Kalman Filter Based on Recurrent Neural Network and its Application in Aviation Surveillance

Yanjun Jiang\textsuperscript{a} and Xiangyu Nong\textsuperscript{b}

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
\textsuperscript{a}jiangyanjun0718@bupt.edu.cn, \textsuperscript{b}2014211465@bupt.edu.cn

Abstract. The most commonly used algorithm for aviation surveillance is Kalman filter. The accuracy of Kalman filter is affected by the accuracy of its parameters. When the parameters may change with environment change, the accuracy of traditional linear Kalman filter will be affected, in severe cases, filtering divergence will occur. This paper proposes an aviation surveillance filtering model that treats Kalman filter as the kernel of Recurrent Neural Network, uses Back Propagation Neural Network to predict parameters of Kalman filter. Which let Kalman filter be trainable and have ability to estimate parameters dynamically. Moreover, actual radar measurement data is used for radar filtering experiment, and the experiment results show the feasibility of this model, and show that this model has better accuracy and adaptability than traditional Kalman filter.

1. Introduction
Since the 21st century, civil aviation has been developing rapidly. The increase in the number of civilian aircraft has brought great challenges to aviation safety. Existing aviation surveillance systems usually use secondary surveillance radar to monitor aircraft. However, while radar monitoring, the accuracy will be reduced by refraction of electromagnetic waves as waves propagate in the atmosphere\cite{1}. Therefore, filtering algorithm is usually used to filter radar measurement data to reduce the errors caused by environment or radar itself. An excellent filtering algorithm can get more accurate position estimation about aircraft, which has a great effect on safety of aviation surveillance system.

At present, aviation surveillance systems mainly use Kalman filter algorithm to filter data. Many researches have been done for Kalman filter. Wang et al. addressed the problem of poor robustness and adaptability of Kalman filtering in radar target tracking in complex environments, and studied a new radar target adaptive robust tracking algorithm\cite{2}. Gao et al. analysed the problems in the application of traditional Kalman filter in complex data environment, presented a synthesis estimation of target state component based on weighted linear track parameter estimation model of single radar\cite{3}, Yang et al proposed a cascaded Kalman filter algorithm\cite{4}.

Kalman filter is a recursive linear minimum variance estimation algorithm. The advantage of applying this filtering algorithm is that each time a new measurement value is added, only the estimation result of previous moment is needed to calculate the estimation value of current moment. There is no need to store measurement values at various times in the past, and as long as dynamic noise and measurement noise parameters used by this algorithm are accurate, Kalman filter is an optimal estimation, that is, mean value of estimation error is 0, and error variance is the smallest. However, when parameters are inaccurate, the filtering accuracy of Kalman filter will be affected by...
error accumulation, or even cause filtering divergence, and the result of filtering will deviate completely from true value. How to estimate parameters of Kalman filtering more accurately is an important factor for applying Kalman filter correctly.

Based on the structural similarity between Kalman filter and Recurrent Neural Network (RNN), a filtering algorithm that treats Kalman filter as core of RNN is proposed in this paper, so that Kalman filter can be used as RNN layer, using back propagation algorithm of RNN to train Kalman filter directly and learns the parameters of the Kalman filter. Enable Kalman filter algorithm to obtain real-time estimation of parameters, so as to improve the adaptability of the traditional Kalman filter algorithm in a dynamic environment, and reduce the workload of manual parameter adjustment.

2. Preliminaries

2.1. The introduction of BPNN
Back propagation neural network (BPNN) is the most widely used neural network, its structure is shown in Figure 1.

![Figure 1. Structure of BPNN.](image)

Back propagation neural network is composed of multi-layer perceptron neurons. Each neuron accepts the output vector from previous layer as its input, and associates a weight for each input component. Each neuron calculates the weighted sum of all input vector components, and applies a nonlinear function to the sum to get output. The neural network fully connects the output of each layer to all neurons of next layer, and the output of the last layer is the output of entire neural network. Input and output of each layer of neurons satisfy is as follow:

\[ y = \varphi(w^T \cdot x + b) \]  

BPNN is widely used because of its excellent generalization ability and simple principle, it’s also been applied in aviation surveillance. Xia and Peng aimed at the low recognition rate of template matching in aircraft target recognition, proposed an aircraft target recognition algorithm based on multi-layer BP neural network[5]. Qian et al. aimed at the problem of target track prediction in hotspot area, proposed an aircraft target track prediction model based on BP neural network[6].

2.2. The introduction of RNN
RNN is a neural network used to analyse time series data. Unlike BPNN, data from RNN does not just flow from input layer to output layer. Each layer of the network has a connection pointing to itself. The simplest RNN neuron is shown in Figure 2. Neurons of RNN receive not only external input at each moment, but also their own output at previous moment. The output at current moment is generated by these two inputs through calculation. If expand a neuron in timestep, its expanding structure looks like Figure 2(b).
At each time step $t$, this cyclic neuron receives an external input $x_t$, and its own output $y_{t-1}$ of previous moment. The output of a single neuron is as follow.

$$y_t = \varphi(x_t \cdot w_x + y_{t-1} \cdot w_y + b)$$  \tag{2}$$

In 1989, Ronald Williams and David Zipser proposed Real-Time Recurrent Learning (RTRL)\cite{7} for Recurrent Neural Networks, and then Paul Werbos proposed the back propagation of Recurrent Neural Networks over time (BP Through Time, BPTT)\cite{8} in 1990. These training algorithms make RNN become practical.

Long-term short-term memory unit (LSTM) is an improvement of basic unit of RNN. It was first proposed by S.H. and J.S. in 1997\cite{9}. It is used to solve the gradient disappearance and gradient explosion phenomenon that occurs in traditional RNN learning sequence. The structure of the long-term and short-term memory unit is shown in Figure 3.

If regard LSTM unit as a black box, it is similar to the basic form of RNN neurons from outside, but LSTM unit will perform better than basic unit, converge faster, and be able to perceive long-term dependence of data. Hu et al. proposed a method for vessel trajectory prediction based on RNN\cite{10}, and Quan et al. proposed a prediction method combining ship trajectory automatic identification system (AIS) data and LSTM\cite{11}.

2.3. Experiment data and preprocessing
The source of experiment data for this paper is message that complies with the ASTERIX standard\cite{12}. The ASTERIX standard is a presentation layer protocol developed for transmission and exchanging of aerial surveillance data. The categories of messages used in this paper are 21 and 48. Among them, the message of category 21 represents the ADSB measurement data of flight target, and the type 48 is the
radar measurement data of the flight target. These two messages will be referred to as ADSB message and radar message for short in the follow-up content of this paper. Among all data items parsed from these two messages, the fields mainly used in this article are shown in Table 1 and Table 2.

Table 1. Data items of category 48 message.

| Data Item   | Name                                      |
|-------------|-------------------------------------------|
| I048/140    | Time of Day                               |
| I048/040    | Measured Position in Polar Co-ordinates   |
| I048/090    | Flight Level in Binary Representation      |
| I048/240    | Aircraft Identification                   |
| I048/200    | Calculated Track Velocity in Polar Co-ordinates |

Table 2. Data items of category 21 message.

| Data Item   | Name                                      |
|-------------|-------------------------------------------|
| I021/071    | Time of Applicability for Position         |
| I021/130    | Position in WGS-84 Co-ordinates           |
| I021/145    | Flight Level                              |
| I021/170    | Target Identification                     |
| I021/160    | Airborne Ground Vector                    |

Data items I048/140, I048/090, I048/240, I048/200 of category 48 message correspond to data items I021/071, I021/145, I021/170, I021/160 of category 21 message. The aircraft position information parsed in radar message is radar local polar coordinates, however the position information in ADSB message is WGS.84 latitude and longitude coordinates, the coordinate system of two messages is different. On the other hand, the unit of each data item is not uniform. So before experiment, these two messages should be pre-processed. Units of correspond data items will be converted uniformly, and the coordinate system should be unified, polar coordinates of radar message will be projected to WGS.84 latitude and longitude coordinates, and the latitude and longitude coordinates of both messages are projected to terrestrial coordinate system by using Gauss-Kruger projection.

The radar record and ADSB record of aircraft state after data processing can be expressed as below, where x and y represent the horizontal coordinate of aircraft in terrestrial coordinate system, t is the Unix timestamp, α and β are the components of aircraft ground instantaneous speed on x and y axis. In follow-up experiments, ADSB record will be taken as the real state of aircraft, and radar record will be taken as the measurement state of aircraft, since ADSB record is more accurate.

\[ r_t = [t, x, y, \alpha, \beta] \] (3)

\[ a_t = [t, x, y, \alpha, \beta] \] (4)

3. Trainable Kalman filter algorithm based on RNN

3.1. Kalman filter structure

In this paper, we choose to use linear Kalman filter with white noise, to filter the radar record. The general system state equation of the general linear system with white noise is as follows:

\[ x_k = \phi_{k|k-1} x_{k-1} + B_{k-1} u_{k-1} + \Gamma_{k-1} w_{k-1} \] (5)

\[ z_k = H_k x_k + y_k + v_k \] (6)

Equation (5) represent the motion of system, where \( x_k \) is state vector at time step k, \( \phi_{k|k-1} \) is state transition matrix from time step k-1 to time step k, \( \{ u_k \} \) is a non-random vector sequence, it can be understood as external control term or mean value of motion equation error. \( \{ w_k \} \) is a zero-mean white noise vector sequence. \( B_k \) and \( \Gamma_k \) are conversion matrices, they convert \( u_k \) and \( w_k \) to vectors of same dimension as \( x_k \) respectively. In Kalman filter motion model, the motion state at time step k is
obtained from the state at time state $k-1$, through state transition matrix and adding random errors. Equation (6) represent the motion of system, $z_k$ is the measurement vector of state vector $x_k$ at time step $k$, $H_k$ is measurement matrix converts state vector $x_k$ into the measurement vector, $\{y_k\}$ is mean value of the measurement error of each time, it is non-random sequence also, like $\{u_k\}$. And $\{v_k\}$ is a zero-mean white noise sequence. The linear filter recursion equation at each moment is as follows. Where equation (7) and (12) is state prediction equation and others are variance prediction equation.

$$\hat{x}_{k/\rightarrow^{-1}} = \phi_{k,k-1} \hat{x}_{k-1} + B_{k-1} u_{k-1} + J_{k-1} (z_{k-1} - y_{k-1} - H_{k-1} \hat{x}_{k-1})$$

$$P_{k/\rightarrow^{-1}} = (\phi_{k,k-1} - J_{k-1} H_{k-1}) P_{k-1} (\phi_{k,k-1} - J_{k-1} H_{k-1})^T + \Gamma_{k-1} Q_{k-1} \Gamma_{k-1}^T - J_{k-1} R_{k-1} J_{k-1}^T$$

$$J_k = \Gamma_{k-1} S_{k-1} R_{k-1}^{-1}$$

$$K_k = P_{k/\rightarrow^{-1}} H_k (H_k P_{k/\rightarrow^{-1}} H_k^T + R_k)^{-1}$$

$$P_k = (I - K_k H_k) P_{k/\rightarrow^{-1}}$$

$$\hat{x}_k = \hat{x}_{k/\rightarrow^{-1}} + K_k (z_k - y_k - H_k \hat{x}_{k/\rightarrow^{-1}})$$

In the experimental scenario of this paper, we use the ADSB data of aircraft as actual position of aircraft, and the radar data as the measurement position of the aircraft. According to the pre-processed results in Section 2, the state vector of aircraft at time $k$ is $x_k = [x_k, y_k, \alpha_k, \beta_k]$, and measurement vector at time $k$ is $z_k = [x_k, y_k, \alpha_k, \beta_k]$. The measurement vector $x_k$ and estimated state vector $\hat{x}_k$ are the same, and the meanings of each corresponding components are the same, so the measurement matrix $H_k$ is identity matrix at any time.

Because two radar measurement data at adjacent times are separated by about 4 seconds, we can approximate that the aircraft does uniform linear motion with speed of previous moment between two adjacent times, so the state transition matrix is follow, where $\Delta t_{k,k-1}$ is time difference between two moments.

$$\phi_{k,k-1} = \begin{bmatrix} 1 & 0 & \Delta t_{k,k-1} & 0 \\ 0 & 1 & 0 & \Delta t_{k,k-1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To make up for error caused by the linear model itself, let the components of vector $u_k$ be average error of components of $x_k$ at time $k$, components of $w_k$ are random noise of components of $x_k$, and components of $w_k$ are mutually independent. Therefore, matrices $B_k$ and $\Gamma_k$ are both identity matrices. Systematic error variance $Q_k$ is the covariance matrix of $w_k$, and its diagonal element is variance of each component of $w_k$. Vector $y_k$ is the fixed error of radar observation at moment $k$, $v_k$ is random noise at time $k$, and components of $v_k$ are independent of each other, like $w_k$. Measurement error variance $R_k$ is the covariance matrix of $v_k$.

In order to reduce the complexity of model, we also make following assumptions: $w_k$ and $v_k$ are not related, that is, the covariance matrix $S_k$ of motion noise and measurement noise is 0. Equations (6), (7), (8) above will be simplified after these assumptions to equations (14), (15).

$$\hat{x}_{k/\rightarrow^{-1}} = \phi_{k,k-1} \hat{x}_{k-1} + u_{k-1}$$

$$P_{k/\rightarrow^{-1}} = \phi_{k,k-1} P_{k-1} \phi_{k,k-1}^T + Q_{k-1}$$

The algorithm structure of Kalman filter is shown in Figure 4. Comparing the structure of Kalman filter and LSTM (Section 2.2), it can be seen that there are many similarities between their structures. Firstly, they receive input vector at every time step. In Kalman filter, it is measurement vector $z_k$, There is an output at every time step, namely the predicted state vector $\hat{x}_k$. Both of them have two states that propagate between time steps. In Kalman filtering, these two states are $x_k$ and $p_k$. Therefore,
Kalman filter also has structure like neuron of RNN, treating Kalman filter as the core of RNN neuron and train it with by BPTT algorithm is possible.

3.2. Error prediction base on BP neural network
It is shown from Section 3.1 that in the filtering process, parameters that need to be dynamically estimated are $u_k, y_k, Q_k$ and $R_k$. In order to let them trainable, for each parameter vector or matrix, a group of BPNN is used for prediction, call it BPG, for example $BPG_u$. The structure of network is shown in Figure 5.

![Figure 4. Structure of Kalman filter.](image)

![Figure 5. Structure of BPG.](image)

Each BPNN predict one component of output, for example $BP_1$ represent the first component. Since $u_k, Q_k$ related to $x_k$ and $y_k$, $R_k$ related to $z_k$, input $i$ for $BPG_u$ and $BPG_Q$ is $x_k$, for $BPG_y$ and $BPG_R$ is $z_k$. If output should be a vector, BPG simply joins output values of all BPNNs together to build a vector, otherwise BPG builds a matrix, treats outputs as diagonal elements. After using BPG to predict parameters, final model structure is shown in Figure 6.

![Figure 6. Trainable Kalman filter model.](image)

4. Experiment and analysis

4.1. Experiment prepare
In order to verify the performance of trainable Kalman filter in solving radar tracking problem, this paper design experiment to compare traditional Kalman filter and trainable Kalman filter. For training,
select radar data and ADSB data of the same aircraft for 3 days, use first two days data as training set, and the third day data as testing set. To build training set, the first one of any two adjacent data is used as a training sample and the second one is used as a label to simulate the process of state transition. The sample format of the obtained training set is $[\Delta t, x, y, \alpha, \beta]$, and the format of label is $[x, y, \alpha, \beta]$. Thus, there are 70000 samples in training set.

4.2. Result comparison and analysis

Compare coordinates of radar measurement data, coordinates of ADSB measurement data, filtering result of traditional Kalman filter, and result of trainable Kalman filter. Part of tracks comparison is shown in Figure 7.

![Figure 7](image)

**Figure 7.** Tracks of aircraft.

Intuitively, the filtering accuracy of trainable Kalman filter is higher than traditional Kalman filter, and its filtering result is closer to coordinates of ADSB data.

![Figure 8](image)

**Figure 8.** Error chart.

Compared with the filtering error in Figure 8, traditional Kalman filter has better filtering effect in early stage of filtering, and its result is smoother. However, while system parameters changing with the changing of aircraft position, traditional Kalman filter results in accumulation of errors due to parameter errors, which ultimately results in filter divergence. The error of trainable Kalman filter is less stable at initial stage. However, since it can dynamically estimate the system parameters during the filtering process, it has better adaptability to dynamic environment and can achieve a lower average error value after filtering Close to 0, and reduce the traditional random noise.
5. Conclusions
This paper presents a model for filtering aviation surveillance data. This model uses Kalman filter as the core of the RNN, and uses the BPTT training algorithm to train the parameters of Kalman filter, which improves the adaptability of Kalman filter in a dynamic environment. A cascade of Kalman filters is possible benefit from this improvement, each Kalman filter uses the output of the previous Kalman filter as its input, to achieve a better filtering effect. Compared with traditional Kalman filter, this model has higher accuracy and higher filtering stability. It should be notice that Kalman filtering is too complicated compared to ordinary RNN neurons, the model itself is prone to the problem of overfitting. More research is needed, such as choosing other training methods to alleviate the problem of overfitting.

References
[1] WANG Bo. A Method for Improving Ranging Accuracy of a Secondary Surveillance Radar [J]. Telecommunication Engineering, 2019, 59(09): 1031-1036.
[2] WANG Fei, SHI Jian-tao. A Radar Target Tracking Algorithm Based on Adaptive Filtering [J]. Modern Radar, 2019, 41(10): 35-38.
[3] GAO Xiao, FANG Wei-hua, DONG Guang-bo. Linear-track-based Synthesis Estimation of Target State Component in Single Radar Configuration [J]. Modern Radar, 2019, 41(09): 55-59.
[4] Yang Li, Can Liang, Man Lu, et al. Cascaded Kalman filter for target tracking in automotive radar. 2019, 2019(19): 6264-6267.
[5] XIA Hai-qin, Peng Zhang-you. Aircraft target recognition based on BP neural network[J]. Electronic Measurement Technology, 2019, 42(14): 52-57.
[6] QIAN Kui, ZHOU Ying, YANG Liu-jing, XIE Rong-ping, HE Xi-dian. Aircraft Target Track Prediction Model Based on BP Neural Network[J]. Command Information System and Technology, 2017, 8(03): 54-58.
[7] Ronald J. Williams, David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. 1989, 1(2): 270-280.
[8] Werbos, P. J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
[9] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[10] HU Yu-ke, XIA Wei, HU Xiao-xuan, SUB Hai-quan, WANG Yun-hui. Vessel trajectory prediction based on recurrent neural network[J]. Systems Engineering and Electronics, 2020, 42(04): 871-877.
[11] QUAN Bo, YANG Bo-chen, HU Ke-qi, GUO Chen-xuan, LI Qiao-qin. Prediction Model of Ship Trajectory Based on LSTM[J]. Computer Science, 2018, 45(S2): 126-131.
[12] LIU Yun-feng, LIAO Ying-ting, LIU Shu-bo. Analysis and Implementation of Asterix Cat 021 Data Format Analysis Based on Python[J]. Science and Technology & Innovation, 2019(14): 46-47.