Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 1199-1212
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.5221

A new modification of Kalman filter algorithm

Ahmed Raad Radhi[a,*], Zainab Abdul Redha[a,b], Irtefaa A. Neamah[a]

[a]*Department of Mathematics, Faculty of Computer Science and Mathematics, University of Kufa, Iraq*
[b]*Ministry of Education, Najaf, Iraq.*

(Communicated by Madjid Eshaghi Gordji)

Abstract

This study is concerned with estimating random data and in the presence of noise, as we used the Kalman filter estimation method through the backpropagation algorithm to estimate these data. This is because modern estimation methods have become more important as they were in the past years due to the expansion of the field of science and technology and the increasing data Therefore, the interest became in estimation methods that solve the noise problems that occur in the data. The Kalman filter has become one of the most popular and most reliable estimators in case of data noise. This study tests the use of the Kalman filter and Back Propagation algorithm to estimate the data containing noise and compare the results with the proposed method on the same data. The data is generated randomly in the simulation study. The results showed that Kalman is more efficient in filtering noise from the data and giving a lower mean square error compared to the backpropagation algorithm, but the results of the proposed method outperformed the results of the Kalman filter and the backpropagation with the least possible error.

Keywords: Kalman Filter, Backpropagation Algorithms, Estimation, Machine learning.

1. Introduction

Machine learning play a fundamental role in recent researches in statistics as well as big data. Today, the Kalman family of condition assessment techniques, including the Kalman filter and its many variations, is the de facto basis for condition estimation. More than 6000 patents have been issued in the United States regarding applications or processes containing the Kalman filter. According to Google Scholar, the term "Kalman filter" is used in more than 100,000 scientific articles.

*Corresponding author
Email addresses: ahmedalshmirty@gmail.com (Ahmed Raad Radhi), zainaba.allahibi@student.uokufa.edu.iq (Zainab Abdul Redha), irtefaa.radhi@uokufa.edu.iq (Irtefaa A. Neamah)

Received: April 2021 *Accepted:* May 2021

Additionally, over 7,500 scholarly citations were recorded in Kalman's original article [9]. One of the most commonly used learning algorithms is back-propagation (BP-ANN) [13].The back-propagation algorithm is the main one for Feed-Forward Neural Networks training, very simpler in implementation and calculation compared to other techniques that are mathematically complicated [22].

Rudolf Emil Kalman was born on May 19, 1930, in Budapest, Hungary, It was 1958 that he discovered the Kalman filter for the first time. Kalman Filter (KF) is named after Rudolf E. Kalman, who published his popular paper "A New Approach to Linear Filtering and Prediction Problems" in 1960.

He's revolutionized the subject in 1960 and 1961 with his articles on the Kalman filter [12, 11]. In his first essay, Kalman demonstrated difficulties in case estimation and prediction. He suggested a framework to address these exemplary problems where the system model is linear and properly understood and the statistical properties of noise are achieved. If the system model is not linear, the unscented Kalman filter (UKF) was introduced in 1997 by S.J Julier and J.K Uhlmann. The Robust Kalman filter (RUK)was also introduced to accurately solve problems that could not be achieved on the system model. Then Adaptive Kalman Filter (AKF) was used to solve the problem that the statistical properties of noise and processing are not certain [17].

The Kalman filter is one of the most remarkable inventions of the twentieth century. It has been more than 50 years since Rudolf Kalman's first groundbreaking paper on state estimation began, which launched a major transition towards dynamic modeling of state space. This powerful tour in theory of mathematical systems, along with the two other groundbreaking articles, helped him secure a variety of important awards, including the Medal of Honor of IEEE in 1974, the Kyoto Prize in 1985, the Steele Prize in 1986, and the Charles Stark Prize. Draper Prize in 2008, and the U.S. Global Medal in Science in 2009. The Kalman filter is a popular tool to remove the measurement noise that gets in the data that is often retrieved from various sensors. The most popular application is data cleaning from the dimensions of accelerometers and gyroscopes, which can be found in smartphones and other modern devices, among other things. The Kalman filter is also an estimator that allows the estimation of parameters that are not measured directly by the sensors but are dependent on the measurement quantities [19]. For optimal linear filtering of static random processes, it can be said that the Kalman filter (KF) is the most popular algorithm that provides optimal estimates in the minimum mean squared error (MMSE) [27].of a linear state-space model with Gaussian operation and measurement noise [8]. This algorithm is based on the state-space representation of a linear process. This can be described by two equations, the first is called the state equation, which is a linear combination of a vector of unobserved state variables and some process noise, while the second is called the observation equation that is formed by two lines of relationship with state variables and random noise [14]. The Kalman filter is also an estimation of the minimum variability of dynamic systems and has attracted widespread interest with increasing demands for target tracking. [15].

Kalman filter whose efficiency is due to

1. Low-cost computational requirements.
2. Well-designed recursive properties.
3. Representing the optimum estimator for one-dimensional linear systems, assuming Gaussian error statistics and suitability for real-time application [1].

Due to its optimization, simplicity of implementation, and low computational complexity, the Kalman filter has been effectively implemented in many practical applications, such as target tracking, positioning, navigation, and signal processing [8] It is also commonly used in nearly every quantitative or technical field. In computing, for example, it also has common applications in mapping, guidance, robotics, radar, error detection, and computer vision. They are also used in applications including stabilization of video, recognition of voice, and vehicle control systems. In strictly quantitative domains, the Kalman filter algorithm plays an important role in the study of mathematical finance, time series, econometrics, system determination, and neural networks [13].

A typical structure of a neural network consists of an “input” layer, one or more “hidden” layers, and an “output” layer. The amount of neurons in the layer and the number of layers depend greatly on the complexity of the system studied. It is also important to define Optimal network design [2]. Neural networks come in two classes:

1. feed-forward networks.
2. recurrent (or feedback) networks.

The feed-forward network. It’s a non-recurring network that includes inputs, hidden layers, and output; signals may only propagate in one direction [22].

The Backpropagation algorithm quickly learns by calculating interlaced updates using feedback connections to provide error signals.

The BackPropagation Neural Network (BPNN) was suggested by Rumelhart and McClelland in 1986. It is a multilayer feedforward network trained by the backpropagation algorithm error and it is the most commonly utilized neural network model at present. The BPNN consists of input layer, secret layer, and output layer, and each layer includes multiple neurons. The neurons in the same layer are not connected with each other, while the neurons in the neighboring layers are connected with each other. Trained backpropagation networks are at the core of the latest machine learning successes, including the very latest in speech and image recognition as well as language translation [16], [26].

In several real-world contexts, the elucidation of the training this algorithm by Rumelhart (1986) was the crucial move to make neural networks functional. It is a systematic method for multiple training (three or more) layered artificial neural networks. However, the BP algorithm is a critical to the advancement of neural networks due to the limitations of one-layer and two-layer networks. Indeed, back propagation played a critically important role in the re-emergence of the neural network field in the mid-1980s. Today, 80% of all applications are estimated to use this BP algorithm in one form or the other [4]. By the backpropagation algorithm [23] computed the images of a simple object digitally from the phase and intensity data measured through a large numerical aperture located at an interval of half the aperture width of the object. The images were produced for distinct object directions, and doubling the transparency of the images improved the resolution. The experimental results of using the reverse error propagation algorithm in a simple and carefully selected problem have been presented by the researchers, [3] . The effect of changing network architecture, the number of hidden units, training set size, and initial weight values were studied. Many solution analysis methods have been shown for such a simple problem. Explanations are provided for observed behavior that may provide insights that apply to a range of problems.

This article use the ability of back propagation algorithm in estimation without noise and combine these features in Kalman filter to introduced a new algorithm better than both of them . The suggested algorithm is applied in simulation study to compare the numerical results with different sample size and different rates of noise.

2. Optimum estimates

The Kalman filter is known as an optimal estimator for linear dynamic systems with white process and measurement noise. Before deriving the Kalman filter, it is helpful to review some basic concepts to an optimal estimate. Generalizing the theory to random variables is easy. Assume we get what can be observed

$$y_k = x_k + v_k.$$

x_k : defined as unknown signal

v_k : refer to component of additive noise

If $\widehat{x}_k$ defined as the a signal posteriori estimate x_k , given the observations $y_1, y_2, \ldots,\ y_k$.

Generally, $\widehat{x}_k$ estimation differs from the unknown term signal x_k .

It is necessary for incorrect estimates to have a cost function in order to derive this estimate ideally. The cost function must meet the following requests:

- The cost function is nonnegative.
- The cost function is the non decreasing of the estimation error $\widetilde{x}_k$ defined by

$$\widetilde{x}_k = x_k - \widehat{x}_k.$$

These two requests are met by a mean square error (MSE) defined by

$$\begin{aligned} J_k &= E[(x_k - \widehat{x}_k)^2] \\ &= E\left[\widetilde{x}_k^2\right], \end{aligned}$$

Where E is the expected value operator.

The cost function dependent J_k on time k emphasizes the nonstationary nature of the recursive process of estimation. The performance cost functions are defined as least mean squared estimation error and the dynamic systems are defined as linear. To compute the derivative of an optimal value for the estimate $\widehat{x}_k$, two theories can be used from the stochastic process theory.

3. Kalman filter estimation

In control theory and statistics, Kalman filtering, also known as linear-quadratic estimation (LQE) since it minimizes the quadratic function of estimate error for a linear dynamic system with white measurement and disturbance noise [17]. Uses of a sequence of measurements observed over time, containing statistical noise and other inaccuracies, and generates estimates of unknown variables that appear to be more reliable than those based on a single measurement alone are by estimating a joint probability distribution over the variables for each period [17]. In certain respects, the filter is very powerful: it supports estimates of past, present, and even future states [25].

Let's now think of the "filter" part. All filters have the same goal: to achieve something when nothing else can. The coffee filter is an example that many people can relate to this portafilter will cause liquid to flow through it, leaving behind strong coffee beans. You might also consider a low-pass filter that allows lower frequencies to pass while attenuating higher frequencies. The Kalman filter also functions as a filter, but its operation is more complicated and difficult to understand. The Kalman filter takes the information with the knowledge that it is error, ambiguous, or noisy.

3.1. The discrete-time Kalman filter

Kalman filter (KF) is a collection of the mathematical equations which are, in a way that reduces the mean squared error, provides an effective (iterative) computational method for estimating the state of a process [25] We chose to adopt the original Kalman paper for derivation, which is not only elegant but informative as well. We have studied a discrete linear dynamics system of time. The state or state vector simply, denoted by x_k, the subscript k indicates a separate time the state is the least amount of data on past behavior of the system needed to predict future behavior. The status of x_k is generally unknown. We use a set of observed data, denoted by the vector y_k , to estimate it [7].

3.2. Kalman filter algorithm summary

The algorithm of Kalman filter consists of two main steps: predicting and updating. Note that the concepts " prediction " and "update" are often referred to as "propagation" and "correction" respectively. The algorithm of the Kalman filter is summarized as follows:

- Linear Process and Measurement Models
  Process Equation: $x_k = A_{k-1}x_{k-1} + w_{k-1}$
  Measurement Equation: $y_k = H_k x_k + v_k$
- Prediction step:
  1. Predicted state estimate $\quad \widehat{x}_k^- = A_{k-1}\widehat{x}_{k-1}$
  2. Predicted error covariance $\quad P_k^- = A_{k-1}P_{k-1}A_{k-1}^T + Q_{k-1}$
- Update step:
  1. Kalman gain $\quad K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R_k\right]^{-1}.$
  2. Updated state estimate $\quad \widehat{x}_k = \widehat{x}_k^- + K_k\left[y_k - H_k\widehat{x}_k^-\right].$
  3. Updated error covariance $\quad P_k = (I - K_k H_k)\, P_k^-.$

In the equations mentioned above, the predicted state estimate is evolved from the updated previous updated state estimate. P is called state error covariance. It encrypts the error covariance that the filter think the estimate error has. Note that the covariance of a random variable x is defined as $\operatorname{cov}(x) = \mathbb{E}\left[(x-\widehat{x})(x-\widehat{x})^T\right]^T$ where $\mathbb{E}$ denotes the predicted (mean) value of the statement. We will observe that the error covariance increases at the prediction stage due to the summation with Q, which means that the filter is more unsure about the state estimation after the prediction stage. During the update stage, the measurement residual $\widetilde{y}_k$ is calculated first. The measurement residual, also known as creativity, is the difference between the actual measurement y_k and the estimated $H\widehat{x}_k^-$ .The filter estimates the current measurement by multiplying the predicted state by the measurement matrix. to provide the correction, $K_k\widetilde{y}_\mathrm{k}$, to the predicted estimate $\widehat{x}_k^-$. we use the residual, $\widetilde{y}_k$, Multiplied by the Kalman gain, K_k . After having the updated estimates, the Kalman filter calculates a new error covariance "Updated", P_k , which will be used for the next time stage. To implement the Kalman filter, we need an initialization step. As initial values, We need the initial guess of the state estimation, $\widehat{x}_0$, as well as the initial guess of the error covariance matrix, P_0.

Along with Q and R, $\widehat{x}_0$ and P_0 play an important role in achieving the desired results. There is a rule of thumb called "initial ignorance," which implies that, for faster convergence, User must choose a large P_0 . Finally, after initialization of estimates, one can achieve the implementation of a Kalman filter by implementing the prediction and update phases for each time stage, k = *1,2,3,...,* [6].

4. Back propagation algorithm

The back propagation algorithm (BP) is a fast and simple iterative process that typically performs well, even with complex data. Unlike other learning algorithms (such as Bayesian learning), back propagation has strong computational properties particularly when large-scale data is presented. Training for back propagation is like training on other neural networks. During the training phase with back propagation, the input data is sent repeatedly to the neural network, and for each iteration of the training process, each neural network output presentation is compared to the result required to calculate the error. The error (backpropagation) is then passed onto the neural network and used to adjust the weights so that the error decreases with each iteration. As a result, the neural model gets closer and closer to producing the desired output [21], [18].

The structure of the three-layer BP neural network $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}^T$ is the input vector, $x_0 = 1$ is used to import the hidden layer's threshold; $V = \{v_1, v_2, \ldots, v_i, \ldots, v_n\}^T$ It is the output vector for the hidden layer , $v_0 = 1$ is used to import the output layer's threshold; $O = \{o_1, o_2, \ldots, o_i, \ldots, o_n\}^T$ is the output vector of output layer; and $d = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}^T$ Is the expected output vector [10].

Artificial neural networks j and output O_j can be formulated in mathematical form as follows:

$$h_j = \sum_{i=1}^{n} w_{\mathrm{ji}} x_i.$$

The bias (threshold) is sometimes designated as a weight coming from a unitary valued input and denoted as w_0. As a result, the neuron's final production is given by the following equation,

$$o_j = f\left(h_j + \theta_j\right) = f\left(\sum_{i=1}^{n} w_{\mathrm{ji}} x_i + \theta_j\right).$$

Where

h_j : is the hidden layer.

o_j: is the output layer,

θ_j : the bias factor,

f : the activation function [24, 20].

Input to the hidden layer $h_{i1}$is given by Eq. (4.1) where x_{i0} represents the data for input layer

$$h_{i1} = \sum_{i_0=1}^{n0} w_{i_0 i_1} x_{i0} \tag{4.1}$$

The answer of the hidden layer $v_{i1}$is seen in Eq. (4.2) by sigmoid activation function;

$$v_{i1} = \frac{1}{1 + e^{-h_{i1}}} \tag{4.2}$$

where N_{i2} is the output to the output layer and given as Eq. (4.3);

$$N_{i2} = \sum_{i_1=1}^{n1} w_{i_1 i_2} v_{i1} \tag{4.3}$$

The answer of the output layer $O_{i2}$is shown as Eq. (4.4) by sigmoid activation function;

$$O_{i2} = \frac{1}{1 + e^{-N_{i2}}} \tag{4.4}$$

In order to minimize the error, those weights should be updated. Mean square error is determined by using defined relation as Eq. (4.5);

$$Error = \frac{1}{2} \sum_{i_2=1}^{n2} \left(O_{i_2}^{d} - O_{i2}\right)^2 \tag{4.5}$$

Where the desired output is $O_{i_2}^{d}$ and ANN model output is O_{i2}. The equation which use to get the weight updated is given by the standard gradient descent approach as Eq. (4.6);

$$w_{i1i_2}\,(\text{new}) = w_{i1i_2}\,(\text{old}) - \eta \frac{\partial E}{\partial w_{i2} i_1} \tag{4.6}$$

where w_{i1i_2} are weights shared between output layer neurons and hidden layer neurons, η is the learning rate and E is the mean square error. This equation can also be written as Eq. (4.7);

$$w_{i1i_2}\,(\text{new}) = w_{i1i_2}\,(\text{old}) + \eta \delta_{\text{Oi2}}\ v_{i1} \tag{4.7}$$

Where $\delta_{O_{i2}} = \left(O_{i_2}^{d} - O_{i2}\right) O_{i2}\left(1 - O_{i2}\right)$ called error backpropagated through weights from output layer neurons to hidden layer neurons. Answer of hidden layer neurons is written as v_{i1}. Same process is practiced to update the weights between secret layer neurons and input layer neurons. Associated gradient descent approach is shown as Eq. (4.8)

$$w_{i0i_1}\,(\text{new}) = w_{i0i_1}\,(\text{old}) - \eta \frac{\partial E}{\partial w_{i1} i_0} \tag{4.8}$$

where w_{i0i_1} are weights shared between hidden layer neurons and input layer neurons. This equation can also be written as Eq. (4.9);

$$w_{i0i_1}\,(\text{new}) = w_{i0i_1}\,(\text{old}) + \eta \delta_{\text{hi1}}\ x_{i0}. \tag{4.9}$$

Where $\delta_{hi1} = v_{i1}\left(1 - v_{i1}\right) \sum_{i_2=1}^{n_2} \delta_{\text{Oi2}} w_{i1i_2}$ called error back propagated through weights from hidden layer neurons to input layer neurons.

If a situation happens where the input becomes zero during the period of iteration, when multiplied by weights, it will give no value. Therefore, in order to achieve a non-zero value, a prejudice is added. The prejudice valueneed not be zero [5].

5. Proposed algorithm

The proposed algorithm is a modification of the Kalman filter using the back propagation algorithm.

The Kalman filter, as it is known, works in noise and maintains its stability in the estimation calculation and depends on the most important feature is Kalman filter, it is the update measures and update state per step k.

While the back propagation algorithm is good at estimation even at large sample sizes, provided that there is no noise. It will be less effective as the noise increases.

Therefore, combining the updating features of the Kalman filter with the layers feature of the back propagation algorithm, we will obtain a modified algorithm that combines the advantages of two algorithms. The updating as the next:

1. Updating the true state of the Kalman filter by relying on the first layer in the back propagation (multiplying the real state in the first layer). We obtain the following equation:

$$\widehat{s}_k = \widehat{s_{k-1}} + K_k\left[z_k - \widehat{z}_k\right]\sum_{i=1}^{n} w_{\mathrm{ij}} x_i \qquad i = 1, \ldots, \; n$$

   Where the first layer is $h_i = \sum_{i=1}^{n} w_{\mathrm{ij}} x_i$,

2. Then updating the measurement of the Kalman filter based on the second layer of back propagation (multiplying the measurement in the second layer). We obtain
   The following equation:

$$\widehat{z}_k = H_k \widehat{s_k} \sum_{j=1}^{n} w_{\mathrm{ij}} x_i \qquad j = 1, \ldots, \; n$$

Where, $h_j = \sum_{i=1}^{n} w_{\mathrm{ji}} x_i$, the second layer.

Proposed Algorithm Steps

Step 1: Read process noise and measurement noise.
Step 2: Generate random data.
Step 3: Define measurements and Kalman gain.
Step 4: First measurement= weights of BP + square root of process noise $\star$ data
Step 5: Estimating of posterior state.
Step 6: Estimating of posterior error covariance.
Step 7: For $i : N$
do updating of true state
data= data+ first estimate $\star$ first layer of BP.
Step 8: Updating measurement = first measurement + square root of measurement noise $\star$ second layer.
Step 9: Compute new Kalman posterior.
Step 10: Compute MSE.

This proposed method will be implemented and applied in a simulation study in the next section.

6. Simulation and results

Simulation is performed in MATLAB in a Simulink environment. Digital simulation by generating random data from samples of different sizes (n=30,50,100,250, 500,1000,5000), with generate a process noise covariance matrix Q and a measurement noise covariance matrix R. Where this data was used in three methods: Kalman filter, back propagation, and the proposed method, which is a new method in the field of statistical estimation, which has proven its accuracy and efficiency through the results of the mean square error.
Figure 1 below, explains how this simulation is and then explains the strategy steps for this simulation.

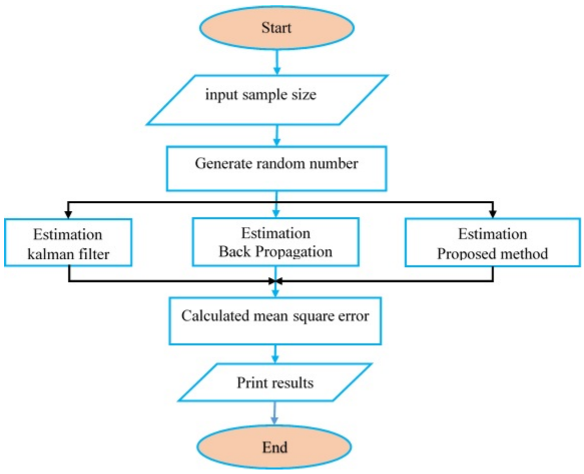


Figure 1: Strategy steps for a simulation study

We simulate the KF, BP, and KFBP algorithm for each of the seven different sample sizes mentioned in the above paragraph. We compare the proposed method (KFBP) with both KF and BP by MSE.

Table 1: Mean square error of estimation when $Q = 0.001, R = 0.02$.

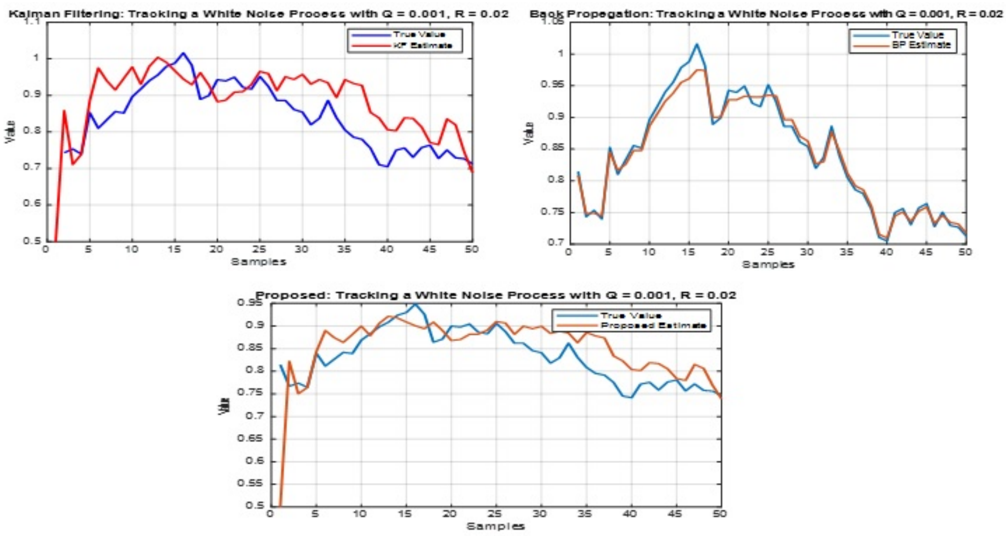
Simple Size	MSE		
	KF	BP	Proposed
30	0.0074	0.0002	0.0046
50	0.0079	0.0001	0.0037
100	0.0086	0.0002	0.0034
250	0.0062	0.0001	0.0021
500	0.0047	0.0001	0.0016
1000	0.0039	0.2018	0.0036
5000	0.0038	0.6857	0.0010

From the above table, we note that the proposed method is less than the mean square error (MSE) compared to other methods and for all sample sizes. We also note the following points:

1. The mean square error of a Kalman filter decreases as the sample size increases. This means that this filter has the ability to reduce the effects of random noise.
2. The back propagation algorithm deteriorates as the sample size increases and therefore the mean square error is increased due to the increase in noise.

3. The proposed algorithm remains the best, most accurate, and efficient in all samples, whether the sample size is increased or decreased, it has the mean square error as low as possible despite the presence of random noise and deterioration of the reverse propagation algorithm.

The following Fgures 2,3, shows the behavior of random data estimation in the Kalman filter, the back propagation algorithm, and the proposed algorithm. The data are represented by the real and estimated values of some sample sizes 50,500,1000. The above three figures also show the comparison between the algorithms in the sample size (50), as well as the samples (500, 1000), respectively. There is a clear difference between the real data and the speculative data in the sample size (1000) using the BP method while we do not observe this in KF and the proposed algorithm.



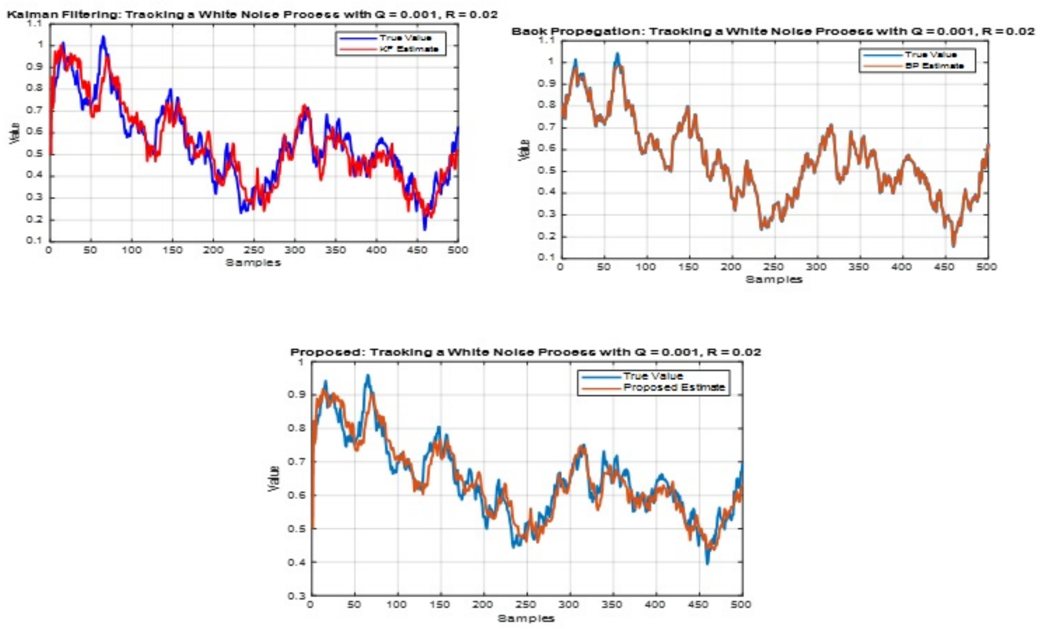


Figure 2: Simulation Results for KF, BP and proposed when sample size $= 500, Q = 0.001, R = 0.02$

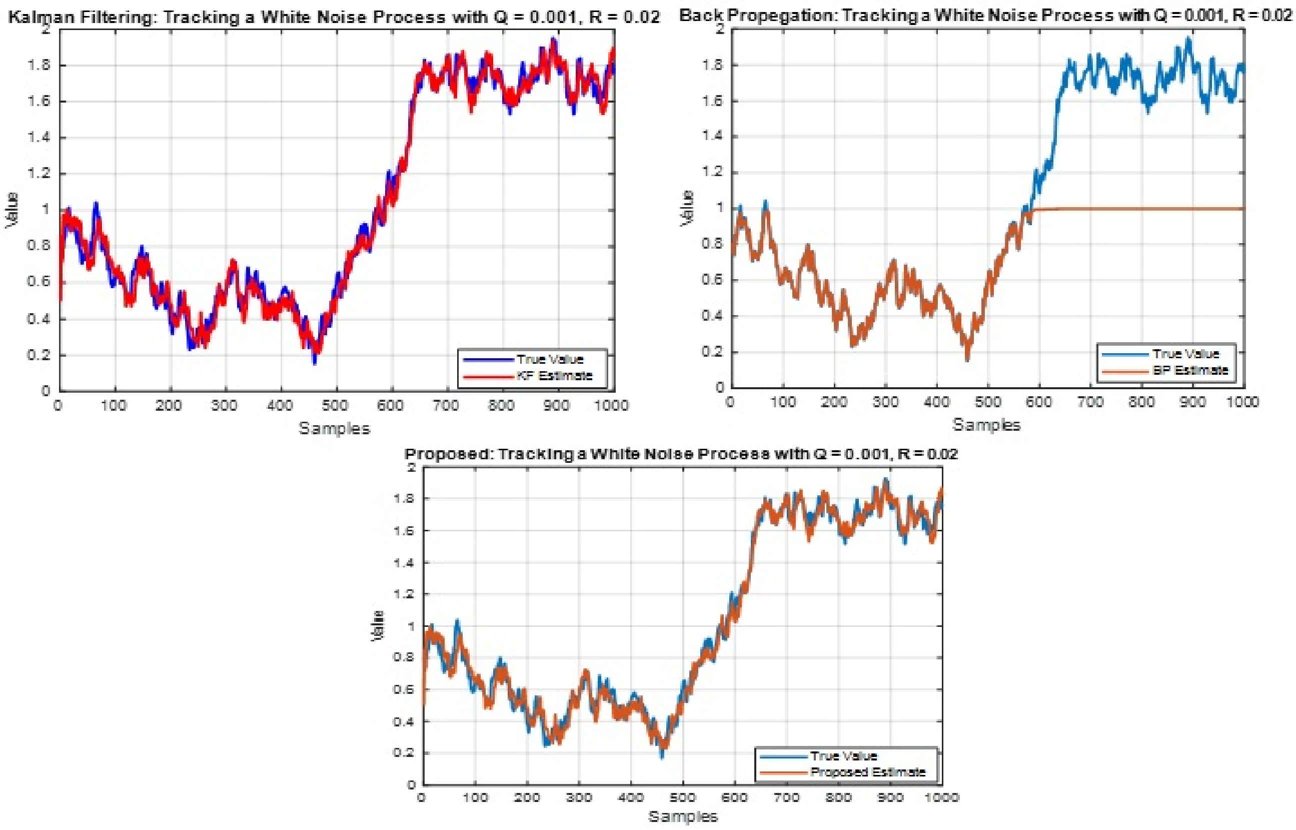


Figure 3: Simulation Results for KF, BP and proposed when sample size $= 1000, Q = 0.001, R = 0.02$

Table 2: Mean square error of estimation when $Q = 0.02, R = 0.001$.

Simple Size	MSE		
	KF	BP	Proposed
30	0.0047	0.0983	0.0042
50	0.0033	0.0611	0.0024
100	0.0021	0.0759	0.0013
250	0.0013	0.4066	0.0013
500	0.0011	0.6195	0.0005
1000	0.0010	6.1153	0.0001
5000	0.0009	19.8327	0.0009

From the above table, we note that the proposed method is less than the mean square error (MSE) compared to other methods and for all sample sizes. We also note the following points:

1. The mean square error of a Kalman filter decreases as the sample size increases. This means that this filter has the ability to reduce the effects of random noise.
2. The back propagation algorithm deteriorates as the sample size increases and therefore the mean square error is increased due to the increase in noise.

3. The proposed algorithm remains the best, most accurate, and efficient in all samples, whether the sample size is increased or decreased, it has the mean square error as low as possible despite the presence of random noise and deterioration of the reverse propagation algorithm.

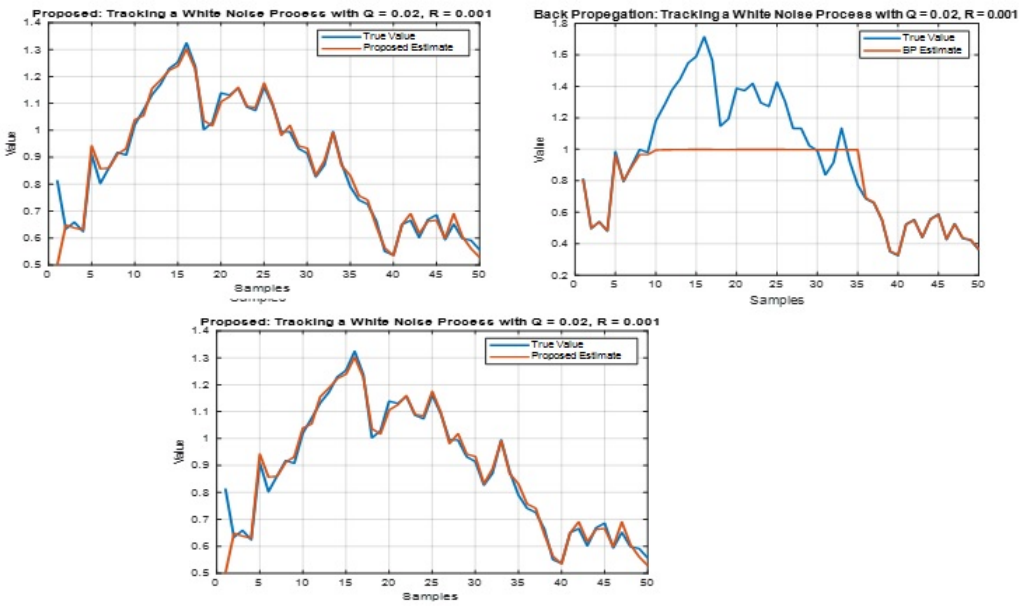


Figure 4: Simulation Results for KF, BP and proposed when sample size 50, $Q = 0.02, R = 0.001$

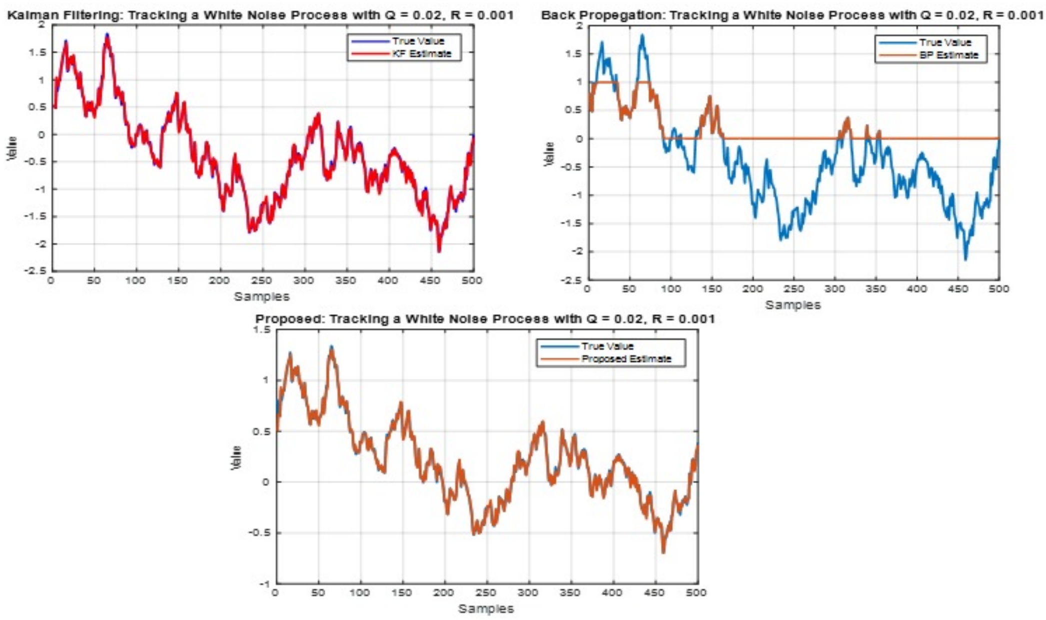


Figure 5: Simulation Results for KF, BP and proposed when sample size 500, $Q = 0.02, R = 0.001$

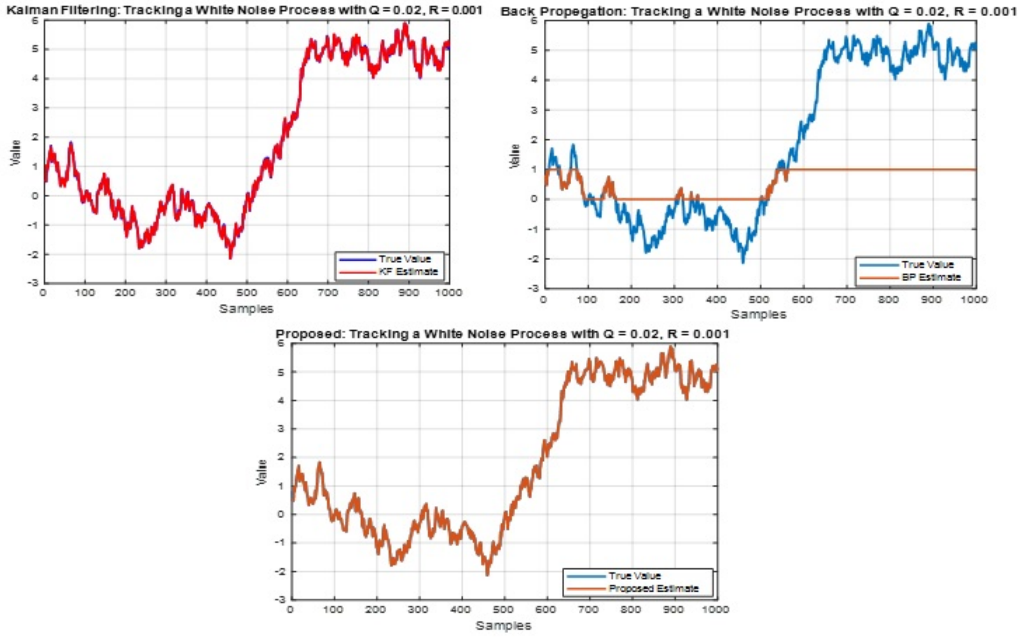


Figure 6: Simulation Results for KF, BP and proposed when sample size 1000, $Q = 0.02, R = 0.001$

The above three figures show the comparison between the algorithms in the sample size (50), as well as the samples (500, 1000), respectively. There is a clear difference between the real data and the speculative data in all sample sizes reported using the BP method while we do not observe this in the KF and the proposed algorithm.

7. Conclusion

Modern researches are concerned with employing the recent algorithms and its capabilities and advantages in developing the procedures of statistical methods.

The main point of this study was to improve the working of the Kalman filter algorithm based on the strengths of the back propagation algorithm.

Through the results that we obtained in this study, it can be concluded that the back propagation algorithm is inefficient in large sample sizes when increasing the noise ratio. As for the Kalman filter, it works optimaly in sample sizes despite the increase in the rate of noise in the data.

The strength of the proposed method lies in conjuncting the features of the back propagation propagation algorithm and its ability to train the data using layers, while the strength of the Kalman filter lies in the optimal estimation in the noised data. Therefore, the proposed method gave the best results in all sample sizes and with different noise ratios.

References

[1] B. Alsadik, *Adjustment Models in 3d Geomatics and Computational Geophysics with Matlab Examples*, Elsevier Inc. 2019.

[2] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hamp and J. Havel, *Artificial neural networks in medical diagnosis*, J. Appl. Biomed. 11(2) (2013) 47–58.
[3] M. D. Bedworth and J. S. Bridle, *Experiments With the Back Propagation Algorithm a Systematic Look at a Small Problem*, Royal Singals And Radar Establishment (RSRE), 1987.
[4] N. Bisoyi, H. Gupta, N. P. Padhy and G. J. Chakrapani, *Prediction of daily sediment discharge using a back propagation neural network training algorithm: A case study of the Narmada River, India*, Int. J. Sediment Res. 34(2) (2018) 125–135.
[5] L. Das, N. Kumar, R. S. Lather and P. Bhatia, *Emerging Trends in Mechanical Engineering: Select Proceedings of ICETMIE*, Springer, 2021.
[6] F. Govaers, *Introduction and Implementations of the Kalman Filter*, InTechOpen, 2019.
[7] S. Haykin, *Kalman Filtering and Neural Networks*, Wiley-Interscience, 2001.
[8] Y. Huang, Y. Zhang, Z. Wu, N. Li and J. Chambers, *A novel robust student's t-based Kalman filter*, IEEE Trans. Aerospace Elect. Syst. 53(3) (2017) 1545–1554.
[9] J. Humpherys, P. Redd and J. West, *A fresh look at the kalman filter*, Siam Review, 54(4) (2012) 801-–823.
[10] D. Jin and S. Lin, *Advances in Computer Science and Information Engineering*, Springer-Verlag, Berlin Heidelberg, 2012.
[11] R. Kalman, *A new approach to linear filtering and prediction problems*, J. Basic Eng. 82(1) (1960) 35–45.
[12] R. Kleinbauer, *Kalman Filtering Implementation with Matlab*, Universität Stuttgart, 2004.
[13] M. Kianpour, E. Mohammadinasab and T. M. Isfahani, *Comparison between genetic algorithm-multiple linear regression and back-propagation-artificial neural network methods for predicting the LD50 of organo (phosphate and thiophosphate) compounds*, Journal of the Chinese Chemical Soc. 67(8) (2020) 1356–1366.
[14] B. Lagos-Álvarez, L. Padilla, J. Mateu and G. Ferreira, *A Kalman filter method for estimation and prediction of space–time data with an autoregressive structure*, J. Stat. Plann. Inf. 203 (2019) 117–130.
[15] Q. Li, R. Li, K. Ji and W. Dai, *Kalman filter and its application*, Int. Conf. Intel. Networks Intel. Syst. (2015), doi: 10.1109/ICINIS.2015.35.
[16] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman and G. Hinton, *Backpropagation and the brain*, Nature Reviews Neurosci.21 (2020) 335—346.
[17] H. Ma, L. Yan, Y. Xia and M. Fu, *Kalman Filtering and Information Fusion*, Springer Singapore, 2020.
[18] N. M. Nawi, N. M. Zaidi, N. A. Hamid, M. Z. Rehman, A. A. Ramli and S. Kasim, *Optimal parameter selection using three-term back propagation algorithm for data classification*, Int. J. Adv. Sci. Engin. Inf. Tech. 7(4-2) (2017) 1528–1534.
[19] B. Nalepa and A. Gwiazda, *Kalman filter estimation of angular acceleration*, IOP Conf. Series: Mat. Sci. Engin. 2020.
[20] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*, Spie, 2005.
[21] S. Setti and A. Wanto, *Analysis of backpropagation algorithm in predicting the most number of internet users in the world*, J. Online Inf. 3(2) (2018) 110–115.
[22] M. Sornam and M. P. Devi, *A survey on back propagation neural network*, Int. J. Commun. Network. Syst. 5(1) (2016) 70–74.
[23] G. Tricoles, E. Rope and O. Yue, *Enhancement of short-range microwave images produced by backward propagation*, Soc. Photo-Optical Instrum. Engin. 1977 (1977), https://doi.org/10.1117/12.955670.
[24] B. Ul Islam, A. Mukhtar, S. Saqib, A. Mahmood, S. Rafiq, A. Hameed, M. Saad Khan, K. Hamid, S. Ullah, A. Al-Sehemi and M. Ibrahim, *Thermal conductivity of multiwalled carbon nanotubes-kapok seed oil-based nanofluid*, Chemical Engin. Tech. 48(8) (2020) 1638–1647.
[25] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, TR, 2006.
[26] X. Zhang, X. Chen and J. Li, *Improving dam seepage Prediction using back-propagation neural network and genetic algorithm*, Math. Prob. Engin. 2020 (2020) Article ID 1404295.
[27] S. Zhao and B. Huang, *Trial-and-error or avoiding a guess? initialization of the Kalman filter*, Autom. 121 (2020) 109184.