Spectral Graph Filtering for Noisy Signals Using the Kalman filter

Ali K. Nawar Al-Attabi[†] and **Ali A. Tayeb**, Non-members

ABSTRACT

Noise is unwanted electrical or electromagnetic radiation that degrades the quality of the signal and the data. It can be difficult to denoise a signal that has been acquired in a noisy environment, but doing so may be necessary in a number of signal processing applications. This paper extends the issue of signal denoising from signals with regular structures, which are affected by noise, to signals with irregular structures by applying the graph signal processing (GSP) technique and a very well-known filter, the standard Kalman filter, after adjusting it. When the modified Kalman filter is compared to the standard Kalman filter, the modified one performs better in situations where there are uncertain observations and/or processing noise and shows the best results. Also, the modified Kalman filter showed a higher efficiency when we compared it with other filters for different types of noise, which are not only standard Gaussian noises but also uniform distribution noise across two intervals for uncertain observation noise.

Keywords: Graph signal processing, Graph Fourier transform, Signal denoising, Kalman filter

1. INTRODUCTION

Noise is undesired electrical or electromagnetic radiation that impairs signal and data quality. Noise may damage files and communications of all sorts, including text, programs, photos, audio, and telemetry, in both digital and analog systems. Denoising a signal acquired in a noisy environment is a significant challenge that might be essential in a variety of signal processing applications. One of the well-known techniques for signal denoising is the Kalman filter, which has seen a lot of improvements in recent years [1]. To have a high Kalman filter efficiency, the assumptions that the linear dynamic system model is accurately determined a priori and that the process and measurement noises are zero-mean, jointly standard Gaussian noises with known covariance matrices are crucial. However, in reality, these presumptions are seldom true, particularly when the noise covariance matrices of the process and measurement are incorrectly understood, which can cause the Kalman filter's performance to significantly degrade or even be rendered useless due to a convergence issue. The statistical characteristics of the process and measurement noises from extensive practical trials are often used to derive the covariance matrices of the process and measurement noises. Distinguishing between noise and measurement signals is very hard due to the intricacy of the noise's underlying process.

H-infinity filtering issues have gotten a lot of academic attention in order to address the modeling flaws and noise uncertainty [2] and [3]. In matrix Riccati-type equations with a Kalman filter, the solutions of H-infinity filters have a comparable structural shape. A Kalman filter with a noise covariance matrix that depends on the noise attenuation parameter can be used to generate an H-infinity filter. A reliable approach for estimating the state of the system with unknown noise statistics is H-infinity filtering. However, the choice of the noise attenuation affects the H-infinity filtering's convergence, accuracy, and computing cost [4]. The adaptive Kalman filter is primarily used to address the issue of uncertain or unreliable statistical features of process noise and/or measurement noise. The key concept is to adaptively assess the hazy statistical characteristics of the noise surrounding the nominal values. With the updated covariance matrices, the Kalman filter may be used. Adaptive fading Kalman filter (AFKF), multiple model adaptive estimation (MMAE), and innovation adaptive estimation (IAE) are the three main methodologies used by adaptive Kalman filtering technology. The fading factor is multiplied with a time-dependent variable to scale the noise covariance matrix [5]. The application often contains both a single fading component and numerous fading factors [6] and [7]. The MMAE method involves choosing the best state estimate from a bank of Kalman filters that are all running simultaneously. Potential issues with process load reduction and speed reduction have recently been explored [8]. To assess the efficacy of covariance estimation, the IAE approach builds an innovation sequence based on the characteristics of an ideal filter. An adaptive limited memory filter (ALMF), with unidentified noise statistics, was introduced in 1976. This led to the development of a few modified adaptive Kalman filter algorithms [9] and [10].

This paper suggests creating an efficient solution in order to cope with scenarios where process or measurement noises are manufactured by adding random

Manuscript received on May 25, 2022; revised on July 29, 2022; accepted on February 3, 2023. This paper was recommended by Associate Editor Piya Kovintavewat.

The author is with Electrical Engineering Department, College of Engineering, Wasit University, Kut, Iraq.

[†]Corresponding author: ali.khalaf@uowasit.edu.iq

Digital Object Identifier: 10.37936/ecti-eec.2023212.249818

noise to the real data based on the Kalman filter and GSP. The graph is a well-known combinatorial structure composed of vertices and edges [11]. Nowadays, signal processing on graphs is widely employed in the field of sensor networks because of the continuous streams of data generated by sensors with extremely high spatial and temporal resolution [12]. The most significant advancement was the introduction of a spectral framework for signal analysis; this is similar to how the Fourier Transform allows complicated signals from sensors to be decomposed into their fundamental frequencies. The spectral transform describes the link between graph signals and the underlying graph's spectral properties. Different authors propose regression methods [13], wavelet decompositions [14]–[17], filter banks on graphs [18] and [19], denoising [20], and compression techniques employing the Laplacian [21] as a result of processing data gathered by sensor networks with erratically arranged sensors. Some of these references take into account the distributed processing of data from sensor fields, while others investigate the multiresolution localized processing of signals on graphs by representing data using wavelet-like bases with varying "smoothness" or by defining transforms based on node neighborhoods. Additionally, signal processing settings have made extensive use of graph-based methodologies. For instance, a prominent image-dependent filtering technique may be understood from a graph viewpoint [22], where other writers depict pictures as graphs for segmentation [23]. Graph-based filtering, processing, and multiresolution representations may be created, and models used in computer graphics applications are frequently seen as graphs [24] and [25].

The main contributions of this paper include:

- This paper extends the challenge of signal denoising from signals with regular structure to signals with irregular structure, which are typically represented by graphs and the graph Fourier transform. After adding noise to the signal, we used a low-pass filter to modify the standard Kalman filter. Then, by adjusting the Kalman filter, a novel method has been presented for signal denoising on graphs.
- The effectiveness of the modified Kalman filter is shown in two different scenarios. In the first scenario, a comparison is made between the standard Kalman filter and the modified one when we have an uncertain process noise or an uncertain observation noise. Some of the standard Gaussian noises with zero mean and different standard deviations, which are important to get a high Kalman filter efficiency, are used. In the second scenario, we compare the modified Kalman filter to an inverse graph filter as well as an approximate iterative solution expressed by a standard graph filter, which are represented in [26]. We then applied this algorithm to measurement denoising for the same data and added the same types of noise, which are not only standard Gaussian noises but also two uniform distributions across two intervals for uncertain observation noise.

The structure of this essay is as follows: Section 2 outlines the background information for GSP and the standard Kalman filter. The methodology to address the problem is introduced in Section 3. To demonstrate the efficacy of the suggested method, Section 4 includes results and discussion of simulation examples. Section 6 contains the conclusions. The future work is represented in Section 7.

2. BACKGROUND INFORMATION

This section contains some introductions to graph theory, the graph Fourier transform (GFT), signal filtering on graphs, and the standard Kalman filter.

2.1 Graph Theory

A graph is a mathematical structure that illustrates the relationships between a collection of objects. Vertices, or nodes, represent the items, whereas edges, or links, reflect the relationships between the vertices. Thus, graphs can be used to describe the geometric structures of data domains in a wide variety of applications, including social networks, sensors, energy, and neural networks.

A graph is either undirected or directed. If there is no origin or destination node for each pair of connected nodes (i.e., the edge connects the nodes bidirectionally), the graph is undirected. On the other hand, in a directed graph, every edge starts at the first vertex of the ordered pair and points to the second vertex of the ordered pair. Undirected graphs are ideal for a wide variety of applications. Certain applications, however, necessitate a directed graph representation. Depending on the application, the number of vertices and edges may vary. Vertices represent people in social network graphs, and links between vertices exist if they have a social or other relationship [27]. Several instances of data processing on graphs are shown in Fig. 1.

A graph is defined as a set of finite vertices $V = \{v_i\}$, a set of finite edges $E = \{e_{ij}\}$, and an adjacency matrix W, which might be unweighted or weighted, and is formally represented by $G = \{V,\ E,\ W\}$. The weight matrix of a weighted graph is an $N \times N$ matrix. If an edge e_{ij} connects two vertices i and j, in a weighted graph W_{ij} represents the weight of this edge, and in an unweighted graph $W_{ij} = 1$. For both weighted and unweighted graphs, if E does not contain e_{ij} then $W_{ij} = 0$[11].

In sensor network graphs, for example, the weights indicate the correlation of the signal corresponding to those nodes and are inversely proportional to the actual distance between sensor nodes. Additionally, in GSP, there is another important matrix called the Laplacian matrix. This matrix is calculated using Eq. (1):

$$L = D - W \tag{1}$$

In Eq. (1), the diagonal degree matrix is denoted by D, and the vector of degrees is designated by d because it is made up of the diagonal elements of D. The degree of

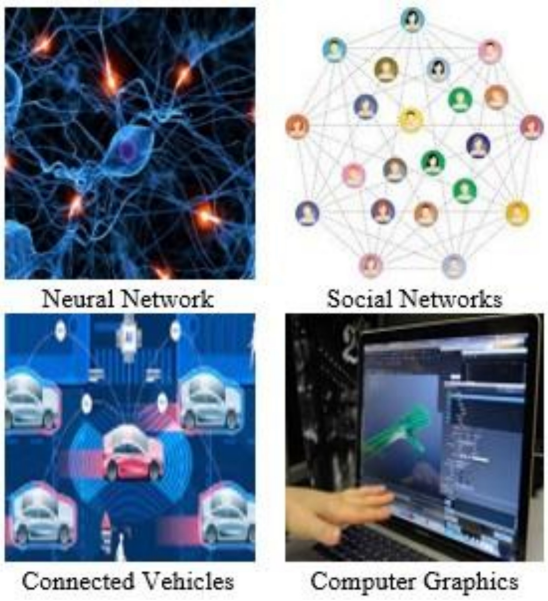
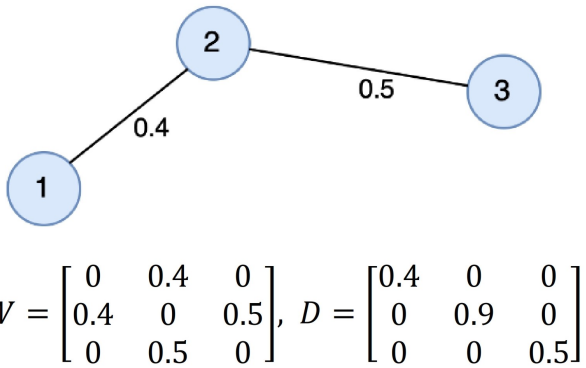


***Fig. 1**: Examples of graph-structured data from the real world. The signals are the values on the nodes.*



$$W = \begin{bmatrix} 0 & 0.4 & 0 \\ 0.4 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

***Fig. 2**: A graph with three nodes and two edges, and its representation using the weight matrix W and the degree matrix D.*

the relevant vertex is contained in each component of d since $d_j = \sum_{i \neq j} W_{ij}$ as illustrated in Fig. 2.

The Graph Laplacian contains a complete collection of orthonormal eigenvectors since it is a real symmetric matrix. The normalized Laplacian matrix can be evaluated from the Laplacian matrix using the degree matrix D using Eq. (2):

$$L_{norm} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{2}$$

The normalized Laplacian matrix, which is shown in Eq. (2), simplifies frequency analysis since it normalizes the frequency to the range from 0 to 2. However, note that L and L_{norm} are not similar matrices; hence, their eigenvalues and eigenvectors are not the same. Additionally, the choice of a suitable Laplacian matrix for a given graph-based problem is application-dependent.

Any undirected graph has a symmetric Laplacian matrix as shown in Eq. (3), which is positive definite. The eigenvalues will be real and non-negative, and the eigenvectors will be orthogonal. Singular value decomposition is a technique for separating values that are the same. Λ is a diagonal matrix having real eigenvalues greater or equal to zero, and the columns of U are the eigenvectors in agreement with the eigenvalues in the ascending order, i.e., $0 \leq \lambda_0 \leq \lambda_1 \ldots \leq \lambda_{N-1}$ and forms an orthonormal basis for $\mathbb{R}^N$ [28].

$$L = U \; \Lambda \; \mathrm{U}^{\mathrm{T}} \tag{3}$$

2.2 Graph Fourier Transform

In traditional signal processing, for signal analysis, the discrete Fourier transform (DFT) was a crucial tool. The discrete Fourier transform is a time-domain alternative basis representation for signals. The Fourier basis is a collection of basis vectors that decompose a given signal. It brings a set of remarkable features that could be used to analyze signals. The DFT representation of the signal is used in a number of existing signal processing algorithms for both time and image signals. This shows that for graph signals, a collection of basis vectors equivalent to the Fourier basis could be generated.

On graphs, analogous concepts of high and low frequencies can be expressed using basis vectors akin to the Fourier basis. A graph signal with a low frequency varies slowly in comparison to its neighbors, whereas a signal with a high frequency fluctuates dramatically with respect to its neighbors. In addition, the basis vectors must be invariant in terms of node order. Surprisingly, the Laplacian matrix's eigenvectors have a natural signal-frequency explanation, similar to sinusoids in the time domain. The existence of a diagonalizing orthonormal matrix is guaranteed by the spectral decomposition theorem [5].

Numerous research papers, such as [11], [29], and [30], mention the Graph Fourier Transform defined in Equation (4): Additionally, Eq. (5) describes the inverse graph Fourier transform (IGFT).

$$\widehat{f}\left(\lambda_i\right) := \langle f, U_i \rangle = \sum_{j=1}^{N} f(j) U_i^*(j) \tag{4}$$

$$f\left(j\right) = \sum_{i=1}^{N} \widehat{f}\left(\lambda_i\right) U_i(j) \tag{5}$$

To compute the vector of GFT $f \in R^N$, which represents the measured signals at each graph vertex, the eigenvectors of the graph Laplacian have been employed. For time-domain signals, the GFT is similar to the traditional Fourier transform, which is represented by Eq. (6).

$$\widehat{f}\left(\xi\right) := \left\langle f, e^{2\pi i \xi t} \right\rangle = \int_R f(t) e^{2\pi i \xi t} dt \tag{6}$$

Assuming a node ordering, it is possible to associate

each element of the eigenvector with the appropriate graph element. For a connected graph, at each node, the Laplacian eigenvector, which corresponds to the eigenvalue, is constant. This means that the value remains constant across nodes, making it similar to a DC signal on a graph. The Laplacian eigenvectors of the graph that are linked to lower frequencies are the reason for the graph's slow variation.

When two vertices are joined by a thin edge with a tiny weight, the signal levels at those points are likely to be different. Conversely, eigenvectors with bigger eigenvalues vary more quickly when vertices are connected by a high-weight edge. In general, the eigenvectors and the eigenvalues represent the concept of frequency, so they represent the constant values and show the maximum variations in the signal on the graph.

2.3 Filtering a Signal on Graph

Filtering noisy signals on a graph is a basic application of GSP. In a classical filter, signal filtering is accomplished using convolution, which is identical to point-wise multiplication. Filtering a graph signal is done in the same way by multiplying it by a graph Fourier domain filter in the graph Fourier domain. The process of amplifying or attenuating the linear contributions of certain of the complex exponential components, which is an analogous representation of the input signal, is specified as a classic filter in the frequency domain. The following equation, Eq. (7), demonstrates it. However, the inverse of this operation may be written as shown in Eq. (8).

$$\hat{f}_o(\xi) = \hat{h}(\xi)\ \hat{f}_i(\xi) \tag{7}$$

$$f_o = \int_R^{\cdot} f_i(\tau)\,\hat{h}(\xi)\,e^{2\pi i\xi t}d\xi := (f_i * h)(t) \tag{8}$$

With a spectral representation of a graph in hand, a frequency filter may be constructed using Eq. (7). A graph spectral filter as described by Eq. (9) which can be written in matrix notation as shown in Eq. (10).

$$\hat{f}_o(\lambda_i) = \hat{f}_i\left(\lambda_i\right)\ \hat{h}(\lambda_i) \tag{9}$$

$$f_o = U g(\Lambda) U^* f_i \tag{10}$$

Here, $g(\Lambda)$ is a diagonal matrix with coefficients $g(\lambda)$ on its diagonal. Similarly, Eq. (11) defines the IGFT.

$$f_o(i) = \sum_{i=0}^{N-1} \hat{f}_i(\lambda_i)\hat{h}(\lambda_i)u_i(i) \tag{11}$$

In Eq. (11), f_o is the output signal of the filter. To define a graph filter, a continuous function $\hat{h} : R_+ \rightarrow R$ is used to find its discrete coefficients, at each eigenvalue, this function is calculated.: $\hat{h}(\lambda_i)$ when $i = 0, ..., N-1$, as illustrated in Fig. 3. Discrete versions of well-known continuous filtering methods can

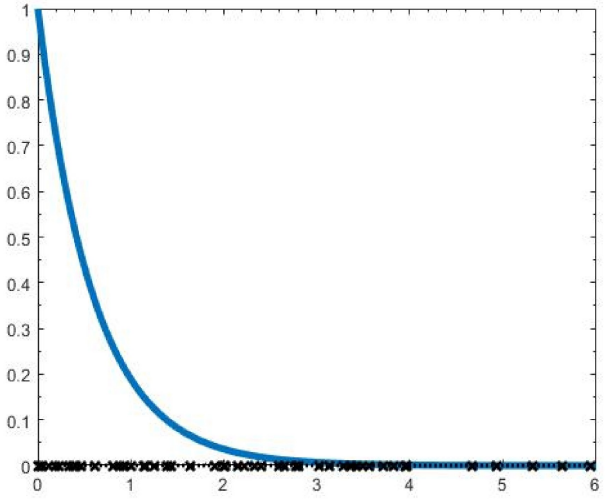


***Fig. 3**: Along the spectrum of the graph, a sample heat kernel filter $\hat{h}$ is plotted. The eigenvalues are represented by the black crosses, and the continuous filter is calculated to provide a discrete counterpart [32].*

be developed with simple spectral graph filters. These approaches for filtering include anisotropic diffusion, Gaussian smoothing, filtering of the total variation, bilateral filtering, and nonlocal means filtering. These filters emerge as options for denoising, inpainting, and super-resolution, which are ill-posed inverse problems [11].

Utilizing the frequency filter involves diagonalizing the Laplacian graph, which is an operation of $\mathcal{O}\left(N^3\right)$ time and $\mathcal{O}\left(N^2\right)$ space complexity. As a result, this filter is applicable only to graphs with fewer than a few thousand vertices.

Nevertheless, various effective methods for filtering signals in the vertex domain have been developed, which are summarized in [11], [31], and [17]. When the polynomial frequency filter in Eq. (9) is $\hat{h}\left(\lambda_i\right) = \sum_{k=0}^{K} a_k \lambda_l^k$ of degree K. In the frequency domain, a filter is equivalent to a filter in the vertex domain. As a result, Eq. (11) can be restructured in the following way, which is shown in Eq. (12).

$$\begin{aligned} f_o(i) &= \sum_{i=0}^{N-1} f_i(i) \sum_{k=0}^{K} a_k \sum_{l=0}^{N-1} u_l^*(i)\,\lambda_l^k u_l(j) \\ &= (a_0 I + a_1 L + \ldots + a_M L^M) f_i \end{aligned} \tag{12}$$

This method uses the Laplacian operator, which is analogous to multiplying the signal with the filter function in the spectral domain. Eq. (12) illustrates the ability to efficiently apply polynomial filters to a certain filter function by selecting the appropriate polynomial. For instance, Chebyshev polynomials are widely used due to their numerical constancy and the efficiency with which they can be assessed with only three recursive terms. [17].

Numerous prior research works have emphasized the need to improve the accuracy of filtering a signal on a graph. For example, [31] proposed using the Lanczos method to present an accelerated algorithm that automatically adjusts to the Laplacian spectrum without computing it explicitly. Also, [17] presented an algorithm that relies on the graph total variance of noisy signals being normalized to generate a closed-form solution to the associated optimization problem by combining an iterative solution and an inverse graph filter represented by a conventional graph filter [26]. The low-pass filter is a common filter that is used to denoise a signal on a graph. This filter is used to reduce the magnitude of the high-frequency content of a signal when the noise has no effect on the frequency content of low-frequency signals. The frequency content is affected by the traditional low-pass filter of the signal $\hat{f_{in}}(\xi)$ is completely defined by its response, which is obtained by multiplying the Fourier transform coefficients of the input signal elementwise by the filter's frequency response. Likewise, after selecting a cut-off frequency or λ_{cut}, to attenuate the high-frequency components of the spectrum that correspond to large eigenvalues and satisfy the requirements, a low-pass graph filter should be built $\hat{h}\left(\lambda_m\right) \approx 0$. For instance, setting λ_{cut} equivalent to the bandwidth's median value denotes that half of all frequencies are equal to or less than λ_{cut} [30]. A low-pass filter is effective at denoising and recovering the true signal from a noisy measurement. In short, numerous existing denoising methods rely heavily on signal smoothing using low-pass filters [26].

There are more signal processing operations on the graph, such as signal translation, down-sampling, many signal filtering approaches, and signal denoising, in addition to the aforementioned procedures.

2.4 Kalman Filter

R.E. Kalman published a seminal study on a recursive solution to the discrete-data linear filtering problem in 1960 [33]. The Kalman filter is a mathematical method that guesses unknown variables using a series of measurements containing statistical noise and other imperfections. These are usually more accurate than estimates based on a single measurement. As a result, real-time applications such as signal processing and econometrics benefit from the Kalman filter. The Kalman filter requires only the present and previous states, so the amount of data stored is minimal. Numerous modern technical applications, such as vehicle guidance, navigation, and control, rely on the employment of a Kalman filter. Furthermore, the Kalman filter is a well-established technique for model-based fusion of sensor data, and it has a wide range of applications in time series analysis [34].

The Kalman filter is designed to address the challenge of estimating the state of a discrete-time controlled process that is governed by the linear stochastic difference equation [35]:

$$x_k^- = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{13}$$

and a measurement z is given by:

$$z_k = Hx_k + v_k \tag{14}$$

In Eqs. (13) and (14), x_k represents the state vector, and x_k^- represents the predicted state vector. The random variables w_{k-1} and v_k embody the process and measurement noise, and they are considered to be independent of one another. To ensure the Kalman filter operates optimally. w_{k-1} and v_k are expected to be white noise with normal probability distributions that are uncorrelated $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$, respectively. In the absence of a driving function or process noise, the transition matrix A relates the state x_k to the state x_{k+1} at time step $k+1$. Also, B and H relate an optional control input u_k and the observation z_k to the state, respectively. Both A and H are subject to change with each time step or measurement; however, they are usually assumed to be constant matrices.

Prediction and update are the two stages of the Kalman filter process. The prediction stage calculates the state based on previous instants, whereas the update stage adjusts the anticipated state based on fresh data [33]. The steps for the stage of prediction are:

- $x_k^- = Ax_{k-1} + Bu_{k-1}$
- $P_k^- = AP_{k-1} + A^T + Q$

The steps for the update stage are as follows:

- $z_k = Hx_{measured}$
- $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
- $x_k = x_k^- + K_k(z_k - Hx_k^-)$
- $P_k = (1 - K_k H)P_k^-$

Here, K_k stands for the Kalman gain in discrete time, I stands for the unit matrix, and matrices P_k^- and P_{k-1} are the covariance matrices of the state distribution before and after the filter Kalman update [35].

3. METHODOLOGY

Two scenarios have been used to prove the strength of the proposed method in this paper and can be described as follows: Scenario A uses data for taxi cabs in San Francisco. Scenario B uses temperature data that was used in a previous study for a filtering operation with simulated noise. As a result, Algorithm 1 can be used to refer to the modified Kalman filter method with uncertain process or observation noise covariance estimation.

Algorithm 1

1. Creating a Graph $G = V,\ E,\ W$.
2. Determining the diagonal degree matrix (D).
3. Locate and normalize the Laplacian graph's matrix (L and L_{norm}).
4. Determine the normalized Laplacian matrix's eigenvectors U and eigenvalues Λ.
5. Add random noise (w) to the vertex domain data (f). All noise sources are white Gaussian noises in the first scenario, while white Gaussian noises and uniform

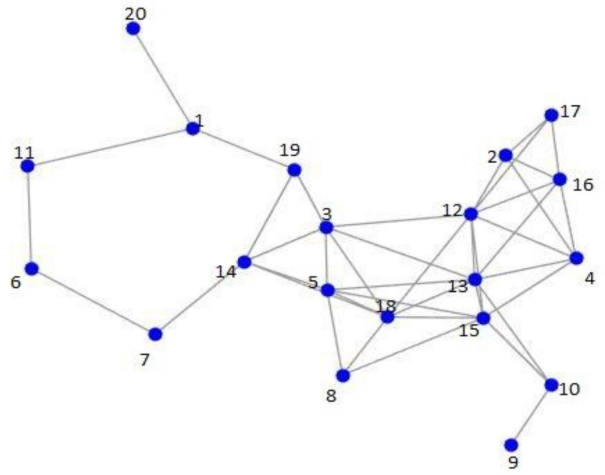


Fig. 4*: The blue circles represent the graph's vertices, and the gray connections represent the graph's edges.*

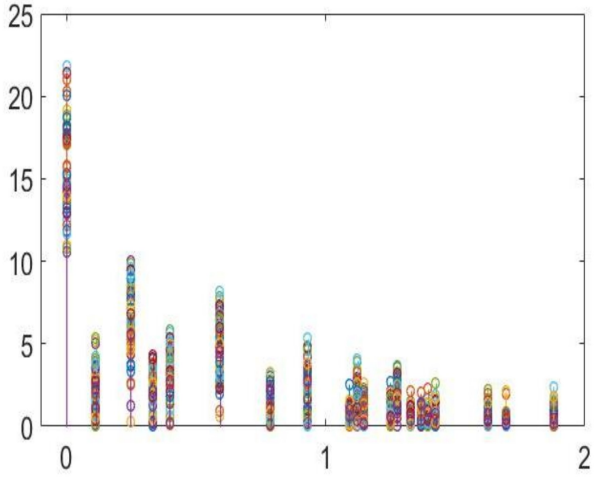


Fig. 5*: In this cumulative spectrum diagram, the magnitudes of the spectrum components are given on the y-axes, while the eigenvalues are seen on the x-axes.*

distribution noises across intervals are added in the second scenario.

6. Calculate the spectrum of the noisy signal ($\hat{x}$) for the noisy data x.
7. Adjust the standard Kalman filter by adjusting the H matrix to be a harmonic breakdown filter as given by Equation (25), and run it when the data is represented in the spectrum domain ($\hat{x}$). Then calculate the filtered signal (x) in the vertex domain using IGFT.
8. In the first scenario, compare the modified Kalman filter and the standard Kalman filter in the RMSE sense when known uncorrelated observation noise and different values of an uncorrelated process noise are used, and the same process is repeated with known process noise and unknown
observation noise.
9. In the second scenario, compare the modified Kalman filter with the exact graph total variation regularization (GTVR) and the graph filter (GF) that are discussed in [26] for different types of observation noise.

3.1 Scenario A

The graph was created using the coordinates of 20 taxi stands, with distances ranging from 0.5 km to 6 km between them, based on the locations of taxi stands in San Francisco [36]. Following the establishment of a threshold for excluding links with lower weights or longer lengths of more than 1.5 km, the graph is generated as shown in Fig. 4.

We looked at a weighted undirected graph with 20 vertices and the weight function $W : 20 \times 20 \rightarrow \mathbb{R}$. The weight of the edge linking the respective vertices is contained in each element of the weight matrix: $W_{i,j} : W\left(v_i, v_j\right)$, and it is formed according to the inverse of the distances between them. The weight is set to zero when there is no edge between two vertices. The matrix W is symmetric. The degree $d(i)$ is known as the summation of the weights of the incident edges: $d\,(i) = \sum_{j=1}^{20} W_{i,j}$; also, A matrix $D_{ii} = d\,(i)$. The Laplacian graph has been normalized. L_{norm} is presented as $L_{norm} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ since it is permanently symmetric and positive semi-definite.

A synthetic taxi proximity signal is created using data from 20 taxicabs [37]. We interpolated their locations with a one-minute time step on May 17, 2008, to ensure time-synchronization between 15:00:00 and 16:00:00 GMT. When the distance between the taxis and the stands was less than 1.5 km, the distance between them was estimated to count the number of taxis in the neighborhood of each stand.

Filtering procedures of a signal on a graph are common, as is smoothing the signal by passing it through low-pass filters. The magnitudes in absolute terms of the spectral components in the graph previously produced reveal that the data obtained from taxi stand networks already had the low-frequency characteristics seen in Fig. 5.

As shown in the next figure, Fig. 6 (a), the noisy signals (x) are definite by applying random noise in comparison to the original signals (f) in the vertex domain. The blue color represents the truth signal, whereas the red color represents the noisy signal. Eqs. (15), (16), (17), and (18) describe the spectral graph filtering procedure:

$$x = f + w \tag{15}$$

$$\hat{x}(\lambda_i) = \sum_{j=1}^{N} x(j) * U_i^*(j) \tag{16}$$

$$\widehat{f}_o(\lambda_i) = \widehat{h}(\lambda_i) * \hat{x}(\lambda_i) \tag{17}$$

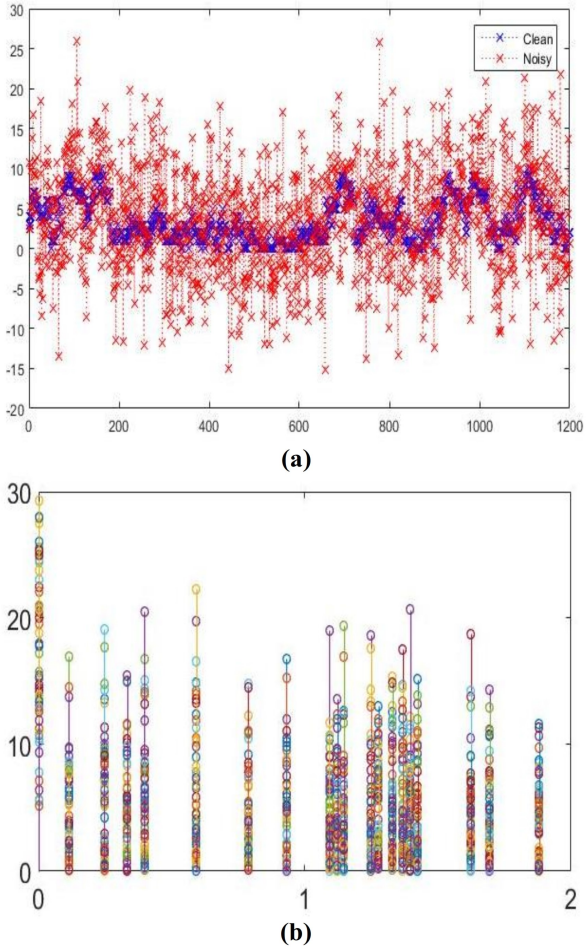


***Fig. 6**: The noisy and true signals, as well as the noisy spectrum, from the vertex domain. (a) f and x the noise is assumed by $N(0, 6^2)$ and by using 20 taxi stand positions. For 60 time-steps. (b) The noise spectral accumulation.*

$$f_o(j) = \sum_{i=1}^{N} \widehat{f}_o(\lambda_i) * U_i(j) \tag{18}$$

The noise, which is w, is produced using a normal distribution represented by $N(0, \sigma^2)$. As seen in Fig. 6 (b), the spectrum of the noisy signal is represented by $\hat{x}$. The spectral response for the i_{th} eigenvalue is represented by $\hat{h}(\lambda_i)$, and the filtered components in the graph spectral domain are represented by $\hat{f}_o(\lambda_i)$.

Fig. 6 shows that introducing Gaussian noise to the signals has a considerably greater impact on the high frequencies than on the low frequencies. As a result, a new filter has been developed based on the harmonic decline equation that is given in Eq. (19).

$$\widehat{h}(\lambda_i) = \frac{a}{\left(1 + \left(\frac{\lambda_i}{b}\right)\right)} \tag{19}$$

As illustrated in Fig. 7 (a), $\hat{h}(\lambda_i)$ is chosen to be utilized as a low-pass filter, with the black crosses representing

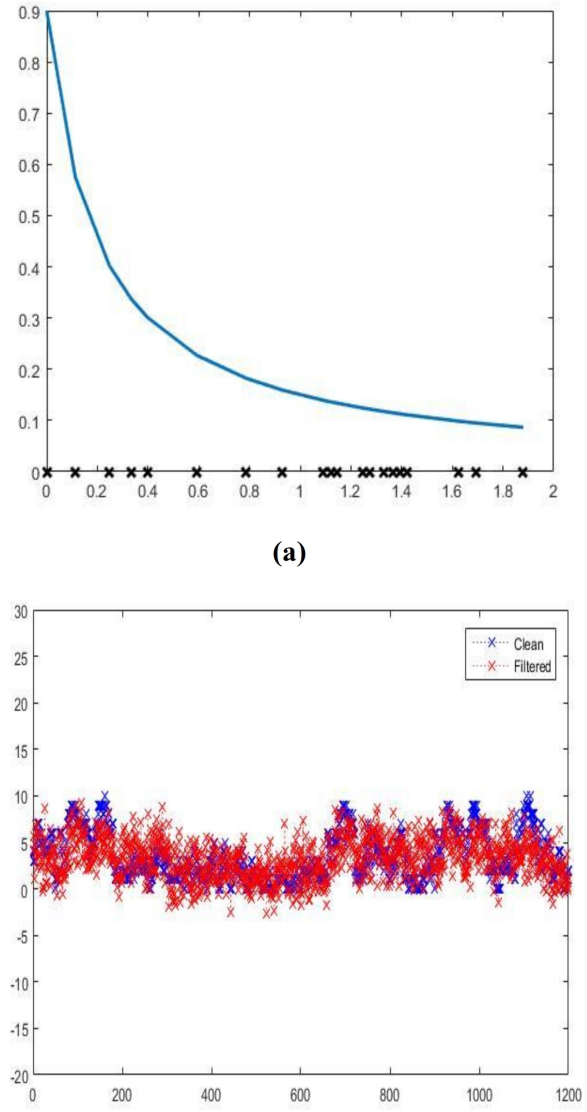


***Fig. 7**: When ($a = 1$ and $b = 0.2$), the properties of the new filter were used to compare the denoised signals to the true signals over 60 time-steps. Using the graph's eigenvalues, (a) $\hat{h}$ is determined. (b) When the noise is given by $N(0, 6^2)$, the filtered signals are compared to the clean signals.*

the eigenvalues and evaluation points for the filter.

Because it reduces the statistical variance matrix of the state error, the Kalman filter is used to enhance the filtering process [34]. For a well-modeled linear system, where the noise sources are all white Gaussian noise and the noise covariance matrices are adequately described, the Kalman filter's optimality can be achieved [38]. Both the discretized system state and measurement equations are represented by the equations below in the Kalman filter algorithm:

- $x_k^- = Ax_{k-1} + w_{k-1}$
- $z_k = Hx_k + v_k$

The noise covariance matrices $w_(k-1)$ and v_k

equivalent to $E[w_{k-1}w_{k-1}^T] = Q_{k-1}$ and $E[v_k v_k^T] = R_k$, respectively. A and H are both identity matrices, and the magnitudes are treated as the state vector (x_k) of the noisy spectral components. Eqs. (20) and (21) are used to describe the prediction stage:

$$x_k^- = Ax_{k-1} \tag{20}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{21}$$

The state is assigned ahead of time in Eq. (20), whereas the error covariance is assigned ahead of time in Eq. (21). Eqs. (22), (23), and (24) describe the update stage:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{22}$$

$$x_k = x_k^- + K_k(z_k - Hx_k^-) \tag{23}$$

$$P_k = (I - K_k H)P_k^- \tag{24}$$

The Kalman gain, updating the predicted state with the observation z_k, and updating the error covariance are all represented by the above equations.

On the other hand, the scale of the detected signal may differ from the scale of the condition being tracked. If we can evaluate the properties of these spectral components, we can make changes to the components of the H matrix. As a result, using Eq. (19), the H matrix is adjusted to operate as a harmonic breakdown filter. Updating the H matrix according to Eq. (25) is related to developing a Kalman filter that is adaptable by combining processing power from two filters because the equivalent of filtering a signal in the spectral domain is point-wise multiplication.

$$H = \begin{cases} H(i,j) = \widehat{h}(\lambda_i) = \frac{1}{1+\left(\frac{\lambda_i}{0.2}\right)} & i = j \\ H(i,j) = 0 & i \neq j \end{cases} \tag{25}$$

Furthermore, both the process noise covariance matrix (Q) and the observation noise matrix (R) need to be adjusted for Kalman filtering to function properly. It's sometimes difficult to find clear formulations for big time-varying system applications [39], [40], and [41]. The standard Kalman filter and the adaptive Kalman filter with a given observation noise and an unknown process noise are applied to the previous spectral graph. The same process is performed with an undefined amount of observation noise and a defined amount of process noise. The parameters are presented in terms of the root mean square error (RMSE) between the filtered and noiseless signals as provided by Eq. (26). The filtered signal is in the domain of the vertex and is a signal that has been filtered using IGFT.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f - f_o)^2} \tag{26}$$

***Table 1**: The average RMSE of Denoised Observations across 60-time steps is indicated, with the average RMSE between Noisy Measures and Noiseless Signals being 5.959 and Observation Noise being N(0,36).*

Observation Noise	Standard. Kalman Filter	Adaptive. Kalman Filter
Zero R	5.959	5.976
N(0,1).	4.032	2.113
N(0,5),	2.875	1.662
N(0,10).	2.510	1.554
N(0,50).	1.930	1.486

Random noise was applied to the original data to produce the noisy measurements. To simulate noise, four different distributions are used: They are all Gaussian distributions with a zero mean and standard deviations of 1, 5, 10, and 50, respectively.

3.2 Scenario B

We used a one-year daily temperature data set gathered from 150 climate stations located in the United States to demonstrate the robustness of the adaptive Kalman filter. The readings from all stations on each day are represented by the signal of a graph with a length of N = 150, ranging from (-50 to 120) degrees Fahrenheit. The geodesic distances between stations are used to create the graph that represents this dataset. With nodes representing stations, we utilize an eight -neighbor graph. Each node is linked to the eight nodes that are geographically closest to it. Following are the edge weights: $A_{n,m} = P_{n,m} / \sum_n P_{n,m}$,

$$P_{(n,m)} = \exp\left(-\frac{d_{n,m}}{\sum_{n,m} d_{n,m}/N^2}\right), 0 \leq n, m < N, \tag{27}$$

where $d_{n,m}$ represents the geodesic distance between stations of the climate n_{th} and m_{th}.

As seen in prior work [26], the noisy measurements were obtained by applying random noise to the actual data. Four distinct distributions are used to mimic noise: A Gaussian distribution with a zero mean and standard deviation equals 5, and a Gaussian distribution with a zero mean and standard deviation equals 10, indicated respectively; also, two uniform distributions across the intervals [-5, 5] and [-10, 10] are represented as U ([-5, 5]) and U ([-10, 10]), respectively.

4. RESULTS AND DISCUSSION

4.1 Scenario A

The following table, Table 1 shows the average RMSE of 60 sets of a graph of sensor signals at 20 taxi stand networks with various levels of observation noise and process noise that are uncorrelated.

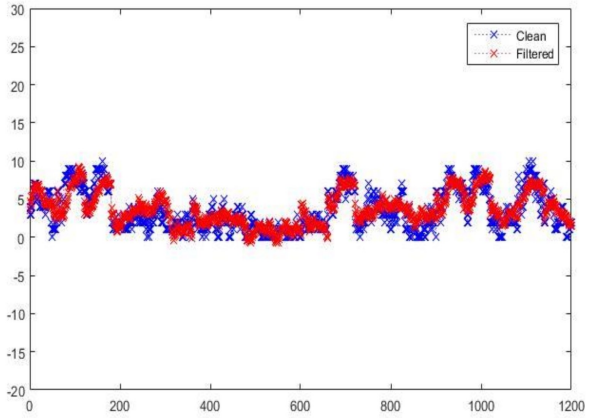


***Fig. 8**: The adapted KF has been used to compare the actual signals to the denoised signals, with $N(0, 6^2)$ representing Gaussian noise.*

***Table 2**: Denoised Measurement Average RMSE for 60 Time Steps that used a Graph of 20 Taxi Stand Sites, At which Average RMSE Between Noisy Readings and Noiseless Signals is 5.959; Process Noise is N(0,1).*

The Process Noise's Covariance	Standard. Kalman Filter	Adaptive. Kalman Filter
Zero Q	1.769	1.586
N(0,1).	2.023	1.475
N(0,5),	2.642	1.497
N(0,10).	3.045	1.594
N(0,50).	4.237	2.117

However, Table 2 shows the average RMSE of 60 sets of graph signals seen on sensors in 20 taxi-cab stand locations using an uncorrelated process noise that is well-known and varied in its noise of observations that are uncorrelated values.

As shown in Table 1, the minimum values of the average RMSE are detected using an adjusted Kalman filter. Fig. 8 illustrates the truth signals and the filtered signals.

By using the same process noise and observation noise, this approach highlights the advantages of employing the adaptive Kalman filter over the harmonic decline filter and the standard Kalman filter. In addition to the adaptive KF method's robustness, there are minor changes in the average RMSE for ambiguous process noise (Q) or ambiguous observation noise (R), as demonstrated in Tables 1 and 2, when compared to a conventional Kalman filter. If the amount of noise polluting the measured signals or the changing characteristics of the status of the system over time are uncertain, an adaptive Kalman filter can be applied.

4.2 Scenario B

We have used a modified Kalman filter using the harmonic decline equation to put up the H matrix as provided by Eq. (19) after calculating the Graph Fourier Transform for the noisy data. The result is calculated in the sense of RMSE, as shown in Table 3.

***Table 3**: The Average of RMSE Measurements Between Noiseless and Denoised Signals by Using the Adaptive Kalman Filter.*

Different Types of the Observation noise					
The noise	Without Filtering	$N(0, 5^2)$	$N(0, 10^2)$	$N(0, 3^2)$	$N(0, 5.74^2)$
$N(0, 5^2)$	4.974	**3.191**	3.786	**3.166**	**3.263**
$N(0, 10^2)$	9.974	**4.905**	**4.649**	5.636	**4.782**
$U([-5, 5])$	2.872	**2.668**	3.557	**2.299**	2.815
$U([-10, 10])$	5.773	**3.441**	3.897	**3.550**	**3.480**

The performance of the modified Kalman filter in signal denoising is excellent, as shown in Table 3. Temperature errors as minor as 2.299 degrees Fahrenheit were recorded. Furthermore, for various types of observation noise, the majority of the results in bold are better while the others are equivalent to those in the prior study listed above [26]. As a result, there is no need to tune the filter, which can be difficult in some applications.

5. CONCLUSION

Filtering a signal obtained in a noisy environment is a difficult task that may be required in a range of signal processing applications. In this paper, the concept of graph signal processing and filtering a signal on a graph by the Kalman filter and the modified Kalman filter are described. Two types of real data, taxi stand data and weather station data, have been utilized to prove the effectiveness of the proposed method by using the modified Kalman filter. The significant hurdles in processing these data are highlighted, particularly when these data are collected via noisy measurements and cases of uncertain observations and/or processing noise, for which their estimation is a prevalent difficulty in the standard Kalman filter. The harmonic decline equation is used as a low-pass filter to adjust the H matrix in the standard Kalman filter and denoise the data in the spectral domain. A comparison between the standard Kalman filter and the adaptive Kalman filter is presented in two scenarios. The adaptive Kalman filter, as seen above, produces the best results. Moreover, the proposed method improves the filtering operation when compared to an inverse graph filter as well as an approximate iterative solution expressed by a standard graph filter, which are represented in the previous study. The uncertainty of observations and/or processing noise and different types of noise are considered in both scenarios.

6. THE FUTURE WORK

Due to a lack of data (i.e., experiments involving gathering real data are usually very time-consuming), many other tests and studies have been postponed in

the future. Future work concerns a deeper analysis of using the GSP technique with the developed types of the standard Kalman filter, such as the extended Kalman filter (EKF), the unscented Kalman filter (UKF), and the cuvetted Kalman filter (CKF), that are used for nonlinear systems in different applications.

ACKNOWLEDGMENTS

Mr. Siheng Chen is thanked for his help and for submitting the data to the authors.

REFERENCES

[1] C. Urrea and R. Agramonte, "Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation," *Journal of Sensors*, 2021, pp. 1–21, Sep. 2021.

[2] J. Chi, C. Qian, P. Zhang, W. Xiao, and L. Xie, "A novel ELM based adaptive Kalman filter tracking algorithm," *Neurocomputing*, Vol. 128, pp. 42-49, Mar. 2014.

[3] Y. Fang, B. Jiang, X. Lv, and Z. Mao, "Research on Fault Estimation and Fault-tolerant Control of Hypersonic Aircraft Based on Adaptive Observer," in *2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, Xiamen, China, 2019, pp. 541-546.

[4] H. Poveda, E. Grivef, G. Ferré, and N. Christov, "Kalman vs H∞ filter in terms of convergence and accuracy: Application to carrier frequency offset estimation," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, 2012, pp. 121-125.

[5] C. Hajiyev and H. E. Soken, *Fault Tolerant Attitude Estimation for Small Satellites*. Boca Raton, USA: CRC Press eBooks, 2020.

[6] S. Guo, L. Chang, Y. Li, and Y. Sun, "Robust fading cubature Kalman filter and its application in initial alignment of SINS," *Optik*, vol. 202, no. 163593, 2020.

[7] H. E. Soken and C. Hajiyev, "Adaptive Fading UKF with Q-Adaptation: Application to Picosatellite Attitude Estimation," *Journal of Aerospace Engineering*, vol. 26, no. 3, pp. 628–636, Jul. 2013.

[8] P. D. Hanlon and P. S. Maybeck, "Multiple-model adaptive estimation using a residual correlation Kalman filter bank," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 4, Tampa, FL, USA, 1998, pp. 4494–4495.

[9] S. Pourdehi, A. Azami, and F. Shabaninia, "Fuzzy Kalman-type filter for interval fractional-order systems with finite-step auto-correlated process noises," *Neurocomputing*, vol. 159, pp. 44–49, Jul. 2015.

[10] B. Xu, P. Zhang, H. Wen, and X. Wu, "Stochastic stability and performance analysis of Cubature Kalman Filter," *Neurocomputing*, vol. 186, pp. 218–227, Apr. 2016.

[11] N. Vaishnav and A. Tatu, "Signal processing on graphs: Structure preserving maps," *IET Signal Processing*, vol. 13, no. 1, pp. 77–85, Feb. 2019.

[12] A. Loukas. "Distributed Graph Filters," PhD thesis, TU Delft, 2015.

[13] L. Rossi, B. Krishnamachari, and C. J. Kuo, "Energy Efficient Data Collection via Supervised In-Network Classification of Sensor Data," in *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Washington, DC, USA, 2016, pp. 33-42.

[14] K. Ahmed and M. A. Gregory, *Wireless Sensor Networks*. Boca Raton, FL, USA: CRC Press, 2016, pp. 33–60.

[15] G. Nikita and K. Seethalekshmi, "Artificial neural network and synchrosqueezing wavelet transform based control of power quality events in distributed system integrated with distributed generation sources," *International Transactions on Electrical Energy Systems*, vol. 31, no. 10, pp. 1-20, May 2020.

[16] X. wang, H. Qi and S. Beck, *Distributed Multitarget Detection in Sensor Networks*. New York, USA: Chapman and Hall/CRC, 2012, pp. 291-305.

[17] J. Huang and S. Li, "On the spectral characterizations of graphs," *Discussiones Mathematicae Graph Theory*, vol.37, no. 3, pp. 729-744, Apr. 2013.

[18] A. Sakiyama, K. Watanabe, Y. Tanaka, and A. Ortega, "Two-Channel Critically Sampled Graph Filter Banks With Spectral Domain Sampling," *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1447-1460, Mar. 2019.

[19] S. K. Narang and A. Ortega, "Perfect Reconstruction Two-Channel Wavelet Filter Banks for Graph Structured Data," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786-2799, Jun. 2012.

[20] L. Teng and H. Li, "CSDK: A Chi-square distribution-Kernel method for image de-noising under the Internet of things big data environment," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, May 2019.

[21] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional Neural Network Architectures for Signals Supported on Graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034-1049, Feb. 2019.

[22] Y. V. Galperin, *An Image Processing Tour of College Mathematics*. Boca Raton, FL, USA: CRC Press, 2021, pp. 103-150.

[23] C. Wang, X. Lin and C. Chen, "Gravel Image Auto-Segmentation Based on an Improved Normalized Cuts Algorithm," *Journal of Applied Mathematics and Physics*, vol. 7, no. 3, pp. 603–610, Jan. 2019.

[24] S. Hachicha, I. Sayahi, A. Elkefi, C.B. Amar, and M. Zaied, "GPU-Based Blind Watermarking Scheme for 3D Multiresolution Meshes Using Unlifted Butterfly Wavelet Transformation," *Circuits Systems and Signal Processing*, vol. 39, no. 3, pp. 1533–1560, Mar. 2019.

[25] K. Zhou, H. Bao, and J. Shi, "3D surface filtering using spherical harmonics," *Computer Aided Design*, vol. 36, no. 4, pp. 363–375, Feb. 2004.
[26] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, USA, 2014, pp. 872-876.
[27] C. Bodnar, C. Cangea, and P. Liò, "Deep Graph Mapper: Seeing Graphs Through the Neural Lens," *Frontiers in Big Data*, vol. 4, Jun. 2021.
[28] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, Mar. 2016.
[29] A. Sandryhaila and J. M. F. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80-90, Sep. 2014.
[30] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042-3054, Jun. 2014.
[31] A. Susnjara, N. Perraudin, D. Kressner, and P. Vandergheynst, "Accelerated filtering on graphs using lanczos method," *arXiv*, preprint arXiv:1509.04537, Sep. 2015.
[32] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond. "GSPBOX: A toolbox for signal processing on graphs," *arXiv*, arXiv:1408.5781, Aug. 2014.
[33] Z. Banjac, Ž. Durović, and B. Kovačević, "Approximate Kalman Filtering Using Robustified Dynamic Stochastic Approximation Method," in *2018 26th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, 2018, pp. 1-4.
[34] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and practice using MATLAB*. California, USA: Wiley-IEEE Press, 2008.
[35] Y. Kim and H. Bang, "Introduction to Kalman Filter and Its Applications," in *Introduction and Implementations of the Kalman Filter*. IntechOpen, May 2019.
[36] "Best places/streets to find a taxi-cab in San Francisco," https://www.google.com/maps/d/viewer?mid=18s-MrsEUD_X1-x5yn0MlEPhj0Uso&hl=en_US (accessed Nov. 1, 2019).
[37] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility," *CRAWDAD wireless network data archive*, Feb. 2009. Accessed: Nov. 1, 2016. [Online]. Available: http://crawdad.org/epfl/mobility/20090224/.
[38] C. K. chui and G. Chen, *Kalman Filtering with Real-Time Applications*. Berlin, Germany: Springer, 2009. pp. 164-177.
[39] G. Carpinelli, D. Proto, P. Caramia and L. Alfieri, "On the comparison between Ensemble Kalman Filter and Kalman Filter for the dynamic harmonic state estimation in a hybrid micro grid," in *2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, Capri, Italy, 2016, pp. 300-307.
[40] H. Ma, L. Yan, Y. Xia, and M. Fu, *Kalman Filter with Recursive Process Noise Covariance Estimation*. Berlin, Germany: Springer, 2019. pp. 21-49.
[41] Y. Bulut and O. Bayat, "Kalman Filtering with Model Uncertainties," *Topics in Model Analysis I*. New York, USA: Springer, 2012. vol. 5, pp. 447-455.

Ali K. Nawar Al-Attabi Received the bachelor's degree in electrical engineering from Al-Mustansiria University, Baghdad, Iraq in 2009. and the Master's degree in electrical engineering from California State University, Fullerton in the USA in 2017. Since 2019, he joined the department of electrical engineering at the University of Wasit, Iraq as a lecturer. His main research interests include IoT, Signal Processing, Artificial Intelligence, and Communication.

Ali Asaad Taib Al Received his B.Sc. in Electrical Engineering/Electronics from Amirkabir University of Technology in 1996, his M.Sc. in Computer Engineering from the same institution in 2000, and his Ph.D. in the fields of deep learning and computer vision from the University of Central Florida in the United States in 2020. He joined the academic staff of Wasit University, Kut, Iraq, from 2005 to 2013 as an assistant lecturer, and from 2020 until present as a lecturer. His main research interests include Machine learning / Deep learning, Computer vision, Large scale data retrieval, and Quantum computing.