Single Dimensional Generalized Kalman Filter

Zainullah Khan
Department of Electronic Engineering
Faculty of Information and Communication Technologies
BUITEMS
Quetta, Pakistan
zain.9496@gmail.com

Hairbyar Bugti
Department of Electronic Engineering
Faculty of Information and Communication Technologies
BUITEMS
Quetta, Pakistan
herrybugti@gmail.com

Ali Sheraz Bugti
Department of Electronic Engineering
Faculty of Information and Communication Technologies
BUITEMS
Quetta, Pakistan
alisherazbugti026@gmail.com

***Abstract*—Sensors are prone to environmental disturbances which introduces error in the sensor data and therefore the data becomes very unreliable and it cannot be used in precise electronic measurement systems. In this paper a software filter is applied that reduces this noise. The Kalman filter is an iterative filter that can reduce the error in the sensor data in real-time. These filters are regularly used in odometry and tracking applications, however, Kalman filters are very task specific. A generalized filter is proposed here that can be applied to multiple sensor types. Noisy data from temperature sensor, LDR and a flex sensor was acquired and the Kalman filtere was applied to it. when the raw sensor data was plotted side by side with the filtered data, the latter produced a smoother output waveform. The effect of different Kalman gain values was also tested on the sensors and it resulted in a delayed estimated signal.**

Keywords—Kalman Filter, Signal Filtering, Generalized Filter.

I. Introduction

Noise filtering and state estimation are imperative whenever a high end electronic system needs to be designed. Since the data received from a sensor does not represent the quantity being measured in its pure form and some elements of the data are affected by ambient noise, hence an error is introduced in the data. The error will either be systematic or random, where different factors like temperature, humidity, sensor placement and electromagnetic radiation will affect the reliability of the measured sensor readings [1]. Therefore, raw sensor data cannot be used when precision is required, hence the erroneous data from the sensor reading needs to be removed to obtain a single value output that should be as close to the measurand as possible. The need for the measured data to be accurate can be understood in applications where a system must operate on reliable data and any uncertainty may result in an unreliable system outcome, such as estimating the correct position of a self-driving car using GPS data. Therefore, in order to make the data more dependable we need a software filter which will nullify the effects of environmental disturbances post measurement.

Filtering and state estimation are the techniques employed to reduce the noise and uncertainty in sensor data. The state estimation element in a software filter predicts the value that the sensor might measure in the next step, the accuracy of the predictions depends on the amount of environmental information provided to the estimation algorithm [2]. Different state estimators take different approaches to remove errors, for example, a running average filter may take average of a fixed number of sampled points to give a single estimated output [3], a particle filter and a Kalman filter may take a probabilistic approach [2].

Modelling the sensor noise over multiple data points results is a Gaussian distribution curve [1]. Therefore, the error associated with an instrument can be represented by the standard deviation of that instrument over its Gaussian distribution curve [1]. Therefore, in this document standard deviation and error are used interchangeably.

The Kalman filter works on the principle of Bayes' Theorem, the state estimation in Kalman filtering depends on the prior knowledge of the system, alternatively stated, the likelihood of any new estimate depends on the probability of the previously occurred states. The Kalman filter performs very well on linear data [4]. It is an iterative filter that runs in real-time to estimate the state of the system in observation. The equations for the simplified single dimensional Kalman filter are derived in [5], [13] and from works of [12], [14] it can be deduced that the Kalman filter state estimation works on three simple equations.

The first equation calculates the Kalman gain which is the most important part of the filter, this gain will determine whether the system should depend more on the measured values or the estimated values:

$$\mathbf{K} = \frac{\boldsymbol{\sigma}_{\mathbf{E}}}{\boldsymbol{\sigma}_{\mathbf{E}} + \boldsymbol{\sigma}_{\mathbf{M}}} \tag{1}$$

Where $\mathbf{K}$ is the Kalman gain and $\boldsymbol{\sigma}_{\mathrm{M}}$ and $\boldsymbol{\sigma}_{E}$ are the measured and estimated errors, respectively. The measured error for any sensor is given by the manufacturer in the datasheet and the estimate is calculated in every iteration.

The Kalman gain is used to estimate the current value of the sensor:

$$\mathbf{E_t} = \mathbf{E_{t-1}} + \mathbf{K}\,(\mathbf{M} - \mathbf{E_{t-1}}) \tag{2}$$

$\mathbf{E_t}$ is the estimate at time $\mathbf{t}$ and $\mathbf{E_{t-1}}$ is the previous estimate and $\mathbf{M}$ is the measured value.

The estimate error is calculated again so that it can be reused in the next iteration of the filter:

$$\boldsymbol{\sigma}_{\mathbf{E}} = (\mathbf{1} - \mathbf{K})\,\mathbf{E_{t-1}} \tag{3}$$

Fig. 1 shows the basic block diagram of a single dimensional Kalman Filter, the Kalman filter equations are

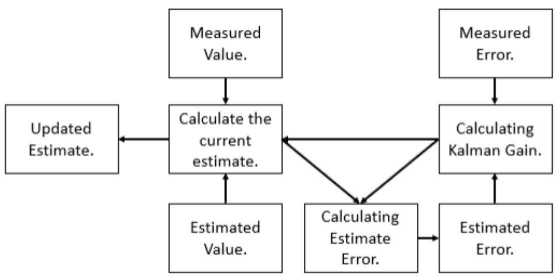

Fig. 1. Block diagram of single dimensional Kalman Filter.

978-1-5386-7939-5/18/$31.00 ©2018 IEEE

integrated into the block diagram. It can see that the Kalman gain is the most important part of the entire filtering process because it is used to calculate the next state and also the error in that state.

Fig. 2 shows that the Kalman filter weighs the probabilities of the estimated and the measured values and provides a new state estimate that is more reliable than the measure data and the previous estimated value. Kalman gain determines whether the new estimated value should depend more on the previous estimate or the measured value.

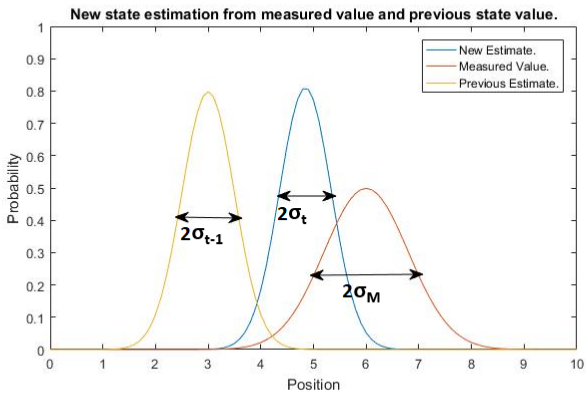

Fig. 2. The measured position of a moving object is shown at position 3 and the estimated value is shown at position 6. The error in the measurements are shown by σ_m, the error in the previous estimate is shown by σ_{t-1} and the error in the new estimate is shown by σ_t. The wider curve is more unreliable than the other curves.

Kalman filters have been researched for years and many variations of the basic filter have been introduced over the years. The importance of Kalman filter is evident from one of its earliest application in the Apollo missions and many other following space missions [15], such as [10] where Kalman filter was used to extract data from line-of-sight sensors and gyros to estimate the relative position and altitude of spacecrafts. The Global Navigation Satellite System that was operational in 1993 and it used the Kalman filter to track the position, velocity and trajectories of more than 24 stellites [15], [18]. Kalman filters have also been used for medical purposes in [16] to detect retinal blood vessels and in [17] to detect kidney transplant rejection. However, robotics and signal processing utilize Kalman filters more than any other field, such as in [6]–[8] the position of a robot is estimated by filtering and analyzing the data from optical range finders and cameras using the state space representation of Kalman filter, similarly, in [9] the position and speed of a brushless DC motor are estimated using the Kalman filter by measuring the current and voltage supplied to the motor. In [20] the data from a motion capture system is successfully mapped to a tele-operated robot where the speed and position of the operator's arms is estimated by the Kalman filter.

Kalman filters are computationally light [2], therefore, they are used to estimate the state of sensor networks as implemented in [11] and [19]. The wireless sensor network in [19] uses Kalman filter to send data that has been verified by multiple sensors, to save power. Single dimensional Kalman filters were used in [12] and [14] for surface temperature forecasting, this produced reliable temperature readings and the algorithm used could be run on any PC.

All the above mentioned Kalman filter implementations calculate multiple features at a time and they are very task specific, therefore, the same filtering algorithm cannot be used for a different system. These implementations are also very complex and need plenty of system information which makes them very hard to understand and implement. The filters mentioned in [12] and [14] were, although easy to implement and computationally light, very task specific and could not be used for any other sensor without making major changes to the equation.

The proposed solution is to use the Kalman filter represented in equations stated above. This variation of the Kalman filter only works for estimating the value that the sensor will measure in the next time instant and the same algorithm can be applied to multiple sensor types to filter their data. Hence this paper attempts to use a single set of equations and applies it to a variety of sensors to prove that a single algorithm can work for a broad spectrum of sensors without changing any of the elements of the filter.

II. Methodology

Controlled experiments were performed on a temperature sensor, a flex sensor and an LDR. Fig. 3. shows the block diagram of the experiment, it can be seen that the experiment is conducted by acquiring the data from the sensor using an Arduino, the data is then sent to MATLAB and the Kalman Filter is used to filter the raw data, then both, the filtered data and the unfiltered data, are plotted side by side for comparison.

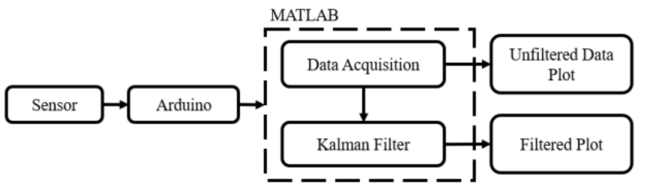

Fig. 3. Block diagram of the experiment, the setup remains the same for all of the sensor types.

A. Flex Sensor

Flex sensors is a strain sensor that changes its resistance when it is bent. The flex sensor was bent by a servo motor that rotated from 0 degrees to 180 degrees in 1.8 seconds and then the servo returned to its normal position again and repeated the same process multiple times. The sensor formed a voltage divider circuit with a fixed resistor and the voltage drop across the sensor was measured using the Arduino's Analog to Digital Converter (ADC). Once the data was acquired, the Kalman filter was applied to it and the data from the Kalman equations was plotted with the original noisy data for comparison. The setup used in this experiment is shown in Fig. 4.

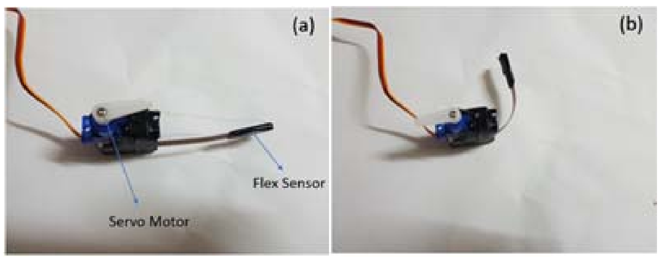

Fig. 4. (a), The flex sensor is relaxed. (b), The servo has rotated 180 degrees and the sensor has been deformed.

B. Light Dependent Resistor

The Light Dependent Resistor (LDR) was enclosed inside a straw at one end and at the other end of the straw a White LED was connected in such a way that the LDR directly faced the LED as shown in Fig. 5. The straw was covered in black tape to minimize the effects of ambient light on the experiment setup. The brightness of the LED was increased from low to high using the Pulse Width Modulation (PWM) technique and at the same time the data from the LDR was acquired by through the ADC of the Arduino.

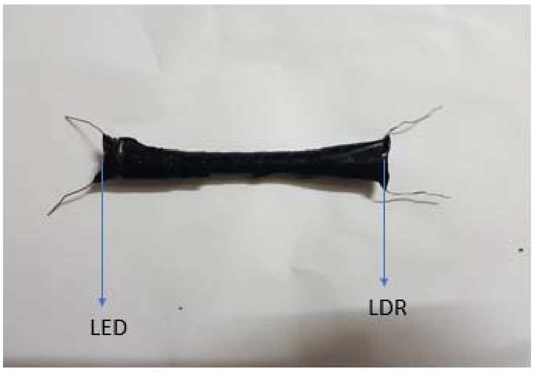

Fig. 5. The LDR and the LED enclosed inside a straw covered with black tape.

C. Temperature Sensor

The temperature sensor was connected in a voltage divider circuit and then connected to the Arduino, as shown in Fig. 6. The metal body of the sensor was attached to a metallic container and the container was filled with hot water to increase the temperature.

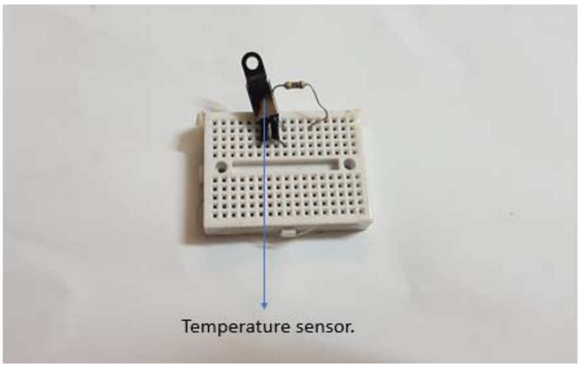

Fig. 6. Temperature sensor connected in a voltage divider configuration.

D. Kalman Filter Values

The same Kalman filter algorithm was used for all of the sensors. The measured error is the only value that is changed when a different sensor is connected to the Arduino. If the Kalman gain is close to unity, then the measured data has less error and the sensor is producing accurate readings. If the Kalman gain is close to zero, then the measured data is erroneous and the estimates are more accurate.

III. RESULTS AND DISCUSSIONS

The experiments were performed on the sensors and the resulting data was plotted in real time on MATLAB. The filtered data and the raw data were compared side by side after the experiment was complete.

Fig. 7 shows the data from the flex sensor as it was continuously bent and relaxed by a servo motor. The raw sensor data was plotted on MATLAB. It can be observed that the sensor data is very noisy and it fluctuates whenever it is about to transition from the flexed state to a relaxed state and vice versa. When the Kalman filter is applied to the raw data, the fluctuations become less frequent and an almost periodic signal is produced. It can be seen that the output graph of the filtered data is delayed. This short delay shows that the sensor data is given less preference and therefore the resulting data depends more on the estimate, and the estimate calculation always lags the sensor data by a short time, hence the time delay. When the experiment started, the filtered data spiked because the initial estimate was very high and then it returned back to normal, this shows that regardless of the initial value of the estimate, the Kalman filter will always find the accurate estimates after a few iterations.

The results for an LDR based light sensor are shown in Fig. 8. The voltage drop across the LDR was plotted while constantly increasing the LED brightness using PWM. The raw sensor data shows that the ADC values increase from zero and keep increasing till 170. The raw data fluctuates constantly, these fluctuations result in a very rough data. After the Kalman filter was applied to the data, the fluctuations in the sensor readings were reduced considerably. The filtered data still fluctuates a little, this is because the resulting data depends more on the measured values than on the estimates. The curve for the filtered data takes some time to rise, this time delay actually shows that the estimated error and the estimated values are being calculated on every iteration and it takes a little while for the filtered plot to reach a correct estimated value.

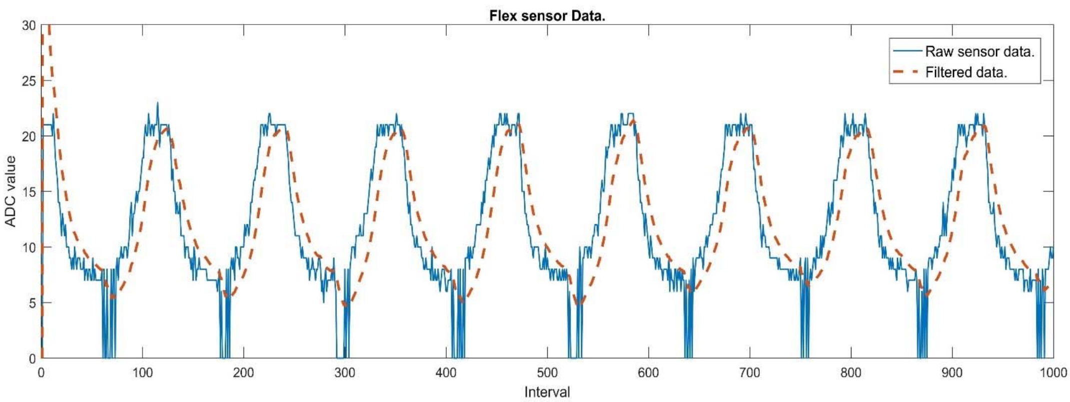

Fig. 7. Comparison between the filtered data and the raw sensor data.

The best results from the Kalman filter were obtained from the temperature sensor. Since it had a small resolution it was more prone to errors than other sensors that were tested. Fig. 9 shows the temperature sensor data. The sensor readings constantly fluctuate even when there is no apparent change in the external temperature. The raw sensor data is very noisy. When the Kalman filter is applied to the sensor data, the filter takes time to reach accurate estimate values, like it has been observed in the previous experiments. When the Kalman filter stabilizes, it can be seen that the filtered data produces a very smooth graph and it follows the raw data perfectly. Even when the temperature is increased, the filter follows the temperature changes perfectly and the resulting curve is much smoother than the raw data. Therefore, it can be used for reliable temperature readings.

From the result of the previously seen sensors it is evident that the filtered data depends on the error in the data and the error in the estimates and based on these errors the Kalman gain decides which data to favor more in the next iteration. Fig. 10 shows that the effect of different Kalman gain values

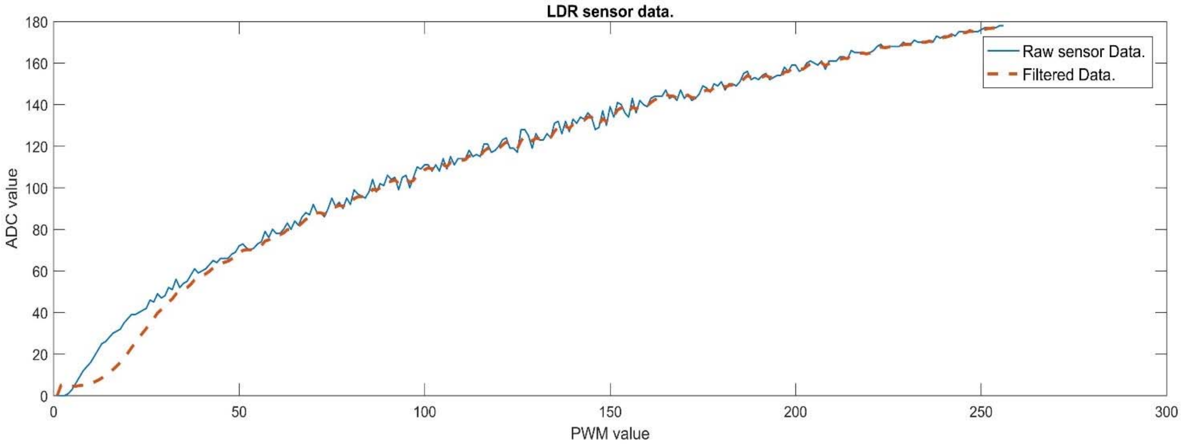

Fig. 8. Raw LDR data compared with filtered data.

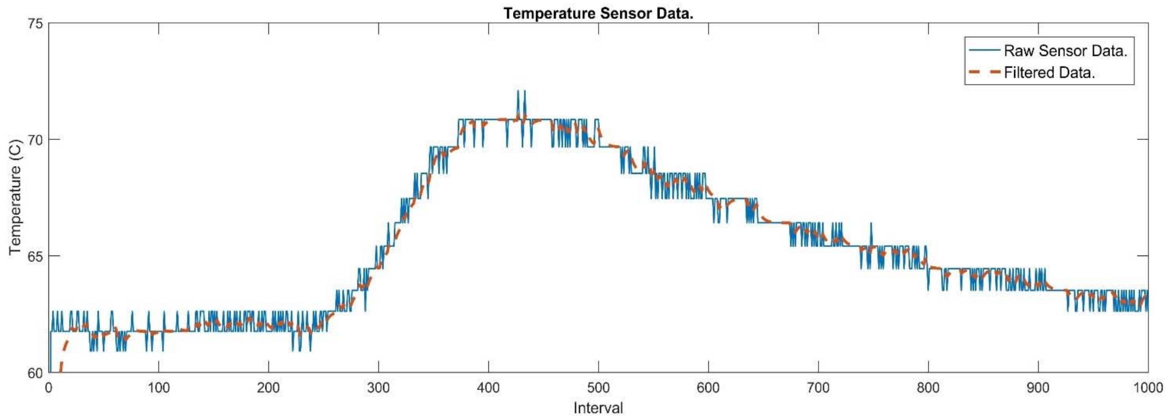

Fig. 9. Raw Temperature sensor data compared with Filtered data.

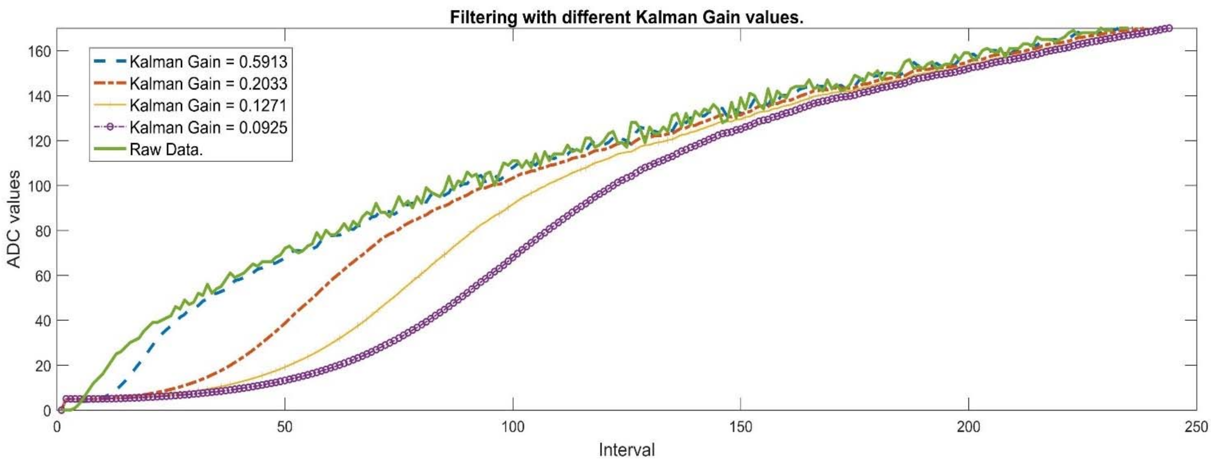

Fig. 10. Kalman filter with different Kalman gain values.

on the filtered LDR sensor readings. The LDR was chosen because it showed the effect of different changing Kalman gains better than the other sensors. it can be seen that when the Kalman gain is close to one, the curve for the filtered data is closer to raw data curve and there is not much difference between the filtered data and the sensor readings. As the Kalman gain decreases, the filtered data favors the estimates more and the filtered curve starts to drift away from the sensor reading. The filtered data becomes very unreliable as the Kalman gain increases because for this particular sensor there is no significant noise in the raw sensor data, therefore the estimates are more erroneous than the sensor data and the Kalman filter needs to favor the sensor data more.

IV. Conclusion

Noisy data can cause a drastic change in results, if supplied to a system the noisy data can mislead the system and the result will be unpredictable outcomes from the system. In this paper the Kalman filter was successfully implemented to three different sensors and it significantly improved the quality of the sensor data. This variation of the Kalman filter can easily be applied to multiple sensor types therefore it generalizes very well. The effect of different Kalman gain values was also analyzed and it was concluded that if the Kalman gain is small, the resulting data will lag the sensor reading and if the gain is too large the data will suffer from the noise received from the sensor. Therefore, before applying the Kalman filter to a sensor, it should be known whether the sensor data should be favored more or the estimated values. Hence in the end it can be deduced that a single algorithm can be used to filter the data of a broad spectrum of instruments without changing an of the elements of the algorithm.

Acknowledgment

The authors would like to acknowledge Balochistan University of Information Technology, Engineering and Management Sciences (BUITEMS), Quetta, Pakistan for the encouraging and enriching research environment provided. We would also like to thank Michel van Biezen for providing detailed lectures about Kalman filter that helped us in our study.

References

[1] A. S. Morris and R. Langari, Measurement and Instrumentation Theory and Application, Second. Joe Hayton, 2016.

[2] M. Grewal and A. Andrews, Kalman filtering: theory and practice using MATLAB, vol. 5. 2011.

[3] S. W. Smith, “Moving Average Filters,” Digit. Signal Process., pp. 277–284, 2003.

[4] J. K. Uhlmann and S. J. Julier, “A New Extension of the Kalman Filter to Nonlinear Systems,” in AeroSense: The 11th Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, 1997.

[5] M. B. Priestley, Spectral analysis and time series. London: Academic Press, 1981.

[6] S. Se, D. Lowe, and J. Little, “Vision-based mobile robot localization and mapping using scale-invariant features,” Proc. 2001 ICRA. IEEE Int. Conf. Robot. Autom. (Cat. No.01CH37164), vol. 2, pp. 2051–2058, 2001.

[7] F. Chenavier and J. L. Crowley, “Position estimation for a mobile robot using vision and odometry,” Robot. Autom. 1992. Proceedings., 1992 IEEE Int. Conf., pp. 2588–2593, 1992.

[8] D. B. Westmore and W. J. Wilson, “Direct dynamic control of a robot using an end-point mounted camera\nand Kalman filter position estimation,” Proceedings. 1991 IEEE Int. Conf. Robot. Autom., no. April, pp. 2376–2384, 1991.

[9] B. Terzic and M. Jadric, “Design and implementation of the extended Kalman filter for the speed and rotor position estimation of brushless DC motor,” IEEE Trans. Ind. Electron., vol. 48, no. 6, pp. 1065–1073, 2001.

[10] L. Zhang, H. Yang, H. Lu, S. Zhang, H. Cai, and S. Qian, “Cubature Kalman filtering for relative spacecraft attitude and position estimation,” Acta Astronaut., vol. 105, no. 1, pp. 254–264, 2014.

[11] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” Proc. IEEE Conf. Decis. Control, pp. 5492–5498, 2007.

[12] G. Galanis and M. Anadranistakis, “A one-dimensional Kalman filter for the correction of near surface temperature forecasts,” vol. 441, pp. 437–441, 2002

[13] A. G. (ed), Applied optimal estimation. MIT Press, 1974.

[14] M. Homleid, “Diurnal Corrections of Short-Term Surface Temperature Forecasts Using the Kalman Filter,” Weather and Forecasting, vol. 10, no. 4, pp. 689–707, 1995.

[15] M. S. Grewal and A. P. Andrews, “Applications of Kalman Filtering in Aerospace 1960 to the Present,” IEEE Control Syst., vol. 30, no. 3, pp. 69–78, 2010.

[16] O. Chutatape, Liu Zheng, and S. M. Krishnan, “Retinal blood vessel detection and tracking by matched Gaussian and Kalman filters,” Proc. 20th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Vol.20 Biomed. Eng. Towar. Year 2000 Beyond (Cat. No.98CH36286), vol. 6, no. 6, pp. 3144–3149, 1998.

[17] K. Gordon, “The multi-state Kalman Filter in medical monitoring,” Comput. Methods Programs Biomed., vol. 23, no. 2, pp. 147–154, 1986.

[18] M. S. Grewal, A. P. Andrews, and C. Bartone, Global navigation satellite systems, inertial navigation, and integration. Hoboken: John Wiley & Sons, 2013.

[19] G. Wang, N. Li, and Y. Zhang, “Diffusion distributed Kalman filter over sensor networks without exchanging raw measurements,” *Signal Processing*, vol. 132, pp. 1–7, 2017.

[20] C. Li, C. Yang, J. Wan, A. S. Annamalai, and A. Cangelosi, “Teleoperation control of Baxter robot using Kalman filter-based sensor fusion,” *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 156–167, 2017.