# Intelligent Estimation Strategies Applied to a Flight Surface Actuator

Elyse Hill
School of Engineering
University of Guelph
Guelph, Ontario, Canada
ehill06@uoguelph.ca

Andrew S. Lee
School of Engineering
University of Guelph
Guelph, Ontario, Canada
alee32@uoguelph.ca

S. Andrew Gadsden
School of Engineering
University of Guelph
Guelph, Ontario, Canada
gadsden@uoguelph.ca

Mohammad Al-Shabi
Department of Mechanical Engineering
University of Sharjah
Sharjah, United Arab Emirates
malshabi@sharjah.ac.ae

***Abstract*—The Kalman filter (KF) has drastically changed and formed the field of state and parameter estimation theory and has impacted a number of applications: spacecraft, GPS, fault detection and diagnosis, stock market analysis, cell phones, autonomous vehicles, to name only a few. A statistically optimal solution for known linear systems is provided by the KF, in the presence of Gaussian white noise. However, the optimality of the KF affects numerical stability and robustness. A number of linear and nonlinear forms of the KF have been introduced to overcome numerical, stability, and nonlinearity issues. In recent years, intelligent or cognitive-based KFs have been proposed. Intelligent filters generally include adaptive gains and feedback for improved estimation accuracy and robustness. These types of filters are typically more robustness to modeling uncertainties and disturbances. This paper provides a comparison of two popular KF methods: fuzzy-based and machine learning-based. These strategies are applied on a flight surface system and the estimation results are compared and discussed. Future trends in intelligent estimation theory are also considered.**

***Keywords—estimation theory; Kalman filter; fuzzy logic; artificial neural networks; flight surface actuator***

## I. Introduction

Estimation theory is a subfield of signal processing and statistics, and often finds applications in mechanical and electrical engineering. Estimation strategies are used to find state or parameter values of interest, which affects system output. These estimates are used typically in the presence of uncertain or inaccurate measurements. Estimation strategies can be used for a number of purposes: air traffic controllers, statistical inference, studies of planet orbits, message retrieval from signals, signal and image processing, and in control systems [1]. Successful system control depends on the knowledge of the parameters or states of interest. Consider a linear system, where the states of interest are position, velocity, and acceleration (i.e., kinematic states). The state dynamics may be represented mathematically by a linear state space representation as per the following:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1.1}$$

where $x_k$: system states, $A$: linear system matrix, $B$: input gain matrix, $u_k$: system input, and $w_k$: system noise. Elements from the state vector need to be measured in order to understand the system behaviour. Sensors placed within the system are used to measure the states. As per the following linear measurement equation, a relationship exists between the states and the measurements:

$$z_{k+1} = Cx_{k+1} + v_{k+1} \tag{1.2}$$

where $z_k$: measurements, $C$: linear measurement matrix, and $v_k$: measurement noise in the sensors. Note that the noise (system and measurement) are defined statistically as white Gaussian noise (zero mean and covariance's $Q_k$ and $R_k$), respectively as per the following:

$$p(w_k) \sim \mathcal{N}(0, Q_k) \tag{1.3}$$

$$p(v_k) \sim \mathcal{N}(0, R_k) \tag{1.4}$$

A filter is used in engineering to extract knowledge of the true states, and to form state estimates. The system measurements typically contain noise. Since estimation strategies remove unwanted noise from signals, the name 'filter' is used. As per equations (1.1) and (1.2), the linear system and measurement dynamics are model based. Estimation strategies may also be applied to nonlinear systems and measurements:

$$x_{k+1} = f(x_k, u_k) + w_k \tag{1.5}$$

$$z_{k+1} = h(x_{k+1}) + v_{k+1} \tag{1.6}$$

where $f$ and $h$ refer respectively to the nonlinear system and measurement models. Introduced in the 1960s, the most popular and utilized estimation method remains the Kalman filter (KF) [2, 3]. A statistically optimal solution is provided by the KF for linear estimation problems. NASA utilized the KF strategy for lunar observations and the Apollo missions, and quickly became known as the 'workhorse' of estimation problems [4, 5].

The KF is formulated in a predictor-corrector manner, and is implemented recursively at each time step. Prediction stage: using the system model, the states are estimated. These states are referred to as a priori estimates, which means 'prior to' knowledge of the measurements. Update stage: based on the innovation (or measurement error), the states are updated. These states are referred to as updated or a posteriori state estimates, which means 'subsequent to' the observations. A large number of problems have used the KF algorithm: signal processing, state and parameter estimation, fault detection and diagnosis, target tracking, and financial markets [6, 7].

The KF popularity and success is based on the optimality of the gain derivation. The KF gain minimizes the trace of the updated state error covariance. The trace is used since it represents the error in the state estimates [8]. The most popular and nonlinear form of the KF is the extended Kalman filter (EKF) [4, 9]. The EKF process is very similar to the KF; however, it requires linearization of the nonlinearities by first-order Taylor series expansion. The KF optimality directly affects filter robustness and stability. The three main KF assumptions are as follows: system model is linear and l, the states have initial known values and conditions, and the system and measurement noises are zero mean Gaussian (or 'white') [4, 1]. In most estimation and control problems, these assumptions are typically not met, which leads to suboptimal results and KF instability [10]. Another form of the KF was introduced by Kalman and Bucy, and is a continuous-time version [11].

The core of the KF algorithm is based on equations (1.7) through (1.11). Equation (1.7) is the a priori state (or predicted) estimate, and equation (1.8) is the a priori state error covariance matrix.

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k \tag{1.7}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \tag{1.8}$$

The KF gain $K_{k+1}$ is calculated as per (1.9), and is used to update the state estimate $\hat{x}_{k+1|k+1}$ as per (1.10). The Kalman utilizes an innovation covariance matrix $S_{k+1}$. The innovation covariance matrix is defined as the inverse term in (1.9).

$$K_{k+1} = P_{k+1|k}C^T\left(CP_{k+1|k}C^T + R_{k+1}\right)^{-1} \tag{1.9}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left(z_{k+1} - C\hat{x}_{k+1|k}\right) \tag{1.10}$$

The a posteriori (or updated) state error covariance matrix represented by $P_{k+1|k+1}$ is calculated as per (1.11), and is iteratively used as per (1.8).

$$P_{k+1|k+1} = (I - K_{k+1}C)P_{k+1|k}(I - K_{k+1}C)^T + K_{k+1}R_{k+1}K_{k+1}^T \tag{1.11}$$

Different forms and derivations of the KF have been studied throughout the literature, with summaries available in [2, 3, 8]. As mentioned, if the KF assumptions are not maintained, estimation results provided by the KF become suboptimal and can be numerically unstable [10]. In addition, the strategy may be sensitive to numerical precision and the computations issues when handling matrix inversions [12]. A number of methodologies have been presented in the literature to improve the KF stability, robustness, and efficiency, as well as the KF's nonlinear variants [13, 14, 15].

This paper provides a comparison of two popular 'intelligent' KF methods: fuzzy-based and machine learning-based. These strategies are applied on a flight surface system (actuator) and the corresponding results are compared and discussed. These algorithms can be applied to robotic systems and sensors. Future trends in intelligent estimation theory are also considered. Section 2 summarizes the two intelligent KF methods. Before concluding the paper, Section 3 describes the simulation setup and results.

## II. Intelligent Kalman Filters

### A. Fuzzy-Based KFs

Fuzzy logic, in contrast to Boolean logic, allows for varying degrees of truthfulness between 0 and 1, rather than absolute truth and falsity. In order to design a fuzzy controller, membership functions must be developed for the system input and output, coupled with a set of rules to handle the inputs and determine what output is appropriate for the current state of the system [16].

A fuzzy system has three parts: fuzzification, rule evaluation, and defuzzification. A set of crisp inputs, for example sensor input data, is transformed into a set of fuzzy inputs through fuzzification. A set of input membership functions, which encompass the relationship between all possibly input values, is used to convert these sensor input values to a fuzzy input value ranging between 0 and 1. Developing appropriate membership functions for the input set is important; using too few can lead to slow system response and using too many can cause instability in the system. After the crisp inputs are converted to fuzzy inputs, these values are fed through a set of rules developed for the system. These rules are used to calculate the system output based on measurement data or the sensor input data in the form of an IF-THEN statement, which relates the output (dependent) variables to the input (independent) variables. Based on the fuzzy input values, the rules are evaluated and the rule that is most true is used to determine the fuzzy outputs. Finally, the fuzzy outputs are converted into crisp outputs through defuzzification, which requires a second set of membership functions, converting the fuzzy outputs between 0 and 1 to meaningful output values.

*B. Machine Learning-Based KFs*

The most popular intelligent KF is based on machine learning or artificial neural networks (ANNs). As the name suggests, ANNs are self-learning such that internal system parameters adapt and change a time [17]. With the advent of faster computer processors, ANNs have become extremely popular, and are used in a variety of applications: internet search engines, data mining, face recognition, and fault detection and identification, to name a few [18, 19]. There are a number of different types of ANNs, with the most popular being based on back propagation (BP), multilayers, and gradient algorithms.

In multilayer perceptron training literature, back propagation (BP) is the most commonly used strategy [20]. It is based on the first-order stochastic gradient descent. BP adjusts weights in an effort to minimize the output error in a supervised manner, and this is done iteratively. The BP strategy uses a constant learning rate, such that the rate of convergence (to a final solution) is slow. A number of strategies have been studied and introduced to improve training performance, speed of convergence, and the overall mapping accuracy [7]. The most common techniques are the quasi-Newton and Levenburg-Marquardt methods. These strategies demonstrate better performance in terms of accuracy because they make use of second-order information as opposed to only first-order.

One of the most popular ANN strategy is a multilayer feed forward network. It consists of a set of inputs that represent the input layer, and a set of one or more hidden layers (i.e., the output layer). Figure 1 illustrates the basic feed forward structure. All of the nodes are interconnected by weighted links, and are used to compute a weighted sum. A bias (i.e., offset) is added to the sum followed by an activation function. The corresponding input signal propagates in a forward direction through the network (layer by layer). Essentially, this network structure represents a mapping of the system inputs to the desired system outputs.

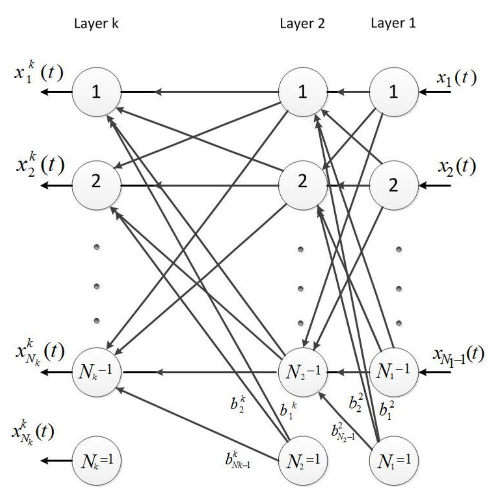


Figure 1. A feed-forward multilayer perceptron network [18].

Let $k$ represent the number of total layers (which includes both the output and input layers). $\text{Node}(n, i)$ represents the $i^{th}$ node in the $n^{th}$ layer. $N_n - 1$ is the number of total nodes in the $n^{th}$ layer. The operation of $\text{node}(n + 1, i)$ is shown as per Fig. 2 and is described as follows:

$$x_i^{n+1}(t) = \varphi\left(\sum_{j=1}^{N_n-1} w_{i,j}^n \, x_j^n(t) + b_i^{n+1}\right) \tag{2.1}$$

where $x_i^n(t)$: output of $\text{node}(n, j)$ for the $t$ training pattern, $w_{i,j}^n$: link weight from $\text{node}(n, j)$ to the $\text{node}(n + 1, i)$, $b_i^n$: node offset (bias) for $\text{node}(n, i)$. Also, note that the function $\varphi(.)$ is a nonlinear sigmoid activation function defined by the following:

$$\varphi(w) = \frac{1}{1 + e^{-aw}} \quad a > 0, -\infty < w < \infty \tag{2.2}$$

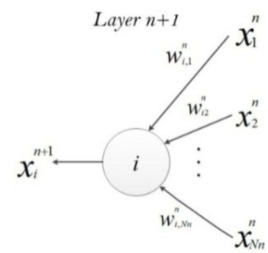


Figure 2. Node $(n + 1, i)$ representation [18].

As per Fig. 2, the node bias is a weighted link based on the last input $N_n$ to $\text{node}(n + 1, i)$, as per the following:

$$\begin{aligned} x_{N_n}^n(t) &= 1, \;\; 1 \le n \le k \\ w_{i,N_n}^n &= b_i^{n+1}, \;\; 1 \le n \le k-1 \end{aligned} \tag{2.3}$$

Based on (2.2) and (2.3), (2.1) can be rewritten as follows:

$$x_i^{n+1}(t) = \varphi\left(\sum_{j=1}^{N_n} w_{i,j}^n \, x_j^n(t)\right) \tag{2.4}$$

In 1989, Singhal and Wu presented an EKF-based neural network training strategy [21]. Compared to other popular or conventional first-order gradient algorithms, such as BP, the EKF offers a powerful training capability [7]. As described in the literature, the EKF has been applied for training of both feed-forward [22] and recurrent networks [23, 24]. Note that this was in a global form (GEKF) as well as in a decoupled form (DEKF). The EKF performance is similar to a second-order derivative (batch-based method), however it avoids local minima by utilizing second-order information found in the state error covariance [7]. The EKF is an efficient alternative to other second-order methods for neural network training. Important applications for ANNs are in the area of fault detection and diagnosis [17]. This is due to a number of reasons: self-learning, adapting algorithm, effective online strategy, relatively good noise rejection, and good approximations to nonlinearities [19]. It is important to note that ANNs typically do not provide knowledge of a system model, but effectively map system inputs to outputs based on large sets of data [25].

## III. FLIGHT SURFACE ACTUATOR RESULTS

In this paper, a flight surface actuator based on an electrohydrostatic system (EHA) was studied. The EHA is a self-contained unit that utilizes a hydraulic pump, circuit, and actuating cylinder or device [26]. The main system components include the following: accumulator, check valves, cylinder or actuator, external gear pump, variable speed motor, and pressure relief mechanisms [27]. Modeling of the flight surface system has been described mathematically and systemically in [28, 29]. For this paper, the kinematic state space equations (3.1 through 3.3) will be considered. The system input is the rotational pump speed $\omega_p$, with units of $rad/s$. In this presented simulation, the sample rate was set to $T = 0.1\ ms$. The system equations (kinematic states) are defined as follows:

$$x_{1,k+1} = x_{1,k} + Tx_{2,k} + Tw_{1,k} \tag{3.1}$$

$$x_{2,k+1} = x_{2,k} + Tx_{3,k} + Tw_{2,k} \tag{3.2}$$

$$\begin{aligned} x_{3,k+1} &= \left[1 - T\left(\frac{BV_0 + M\beta_e L}{MV_0}\right)\right] x_{3,k} \\ &- T\frac{(A^2 + BL)\beta_e}{MV_0} x_{2,k} \\ &- T\left[\frac{2B_2V_0x_{2,k}x_{3,k}}{MV_0} \right. \\ &+ \left. \frac{\beta_e L\left(B_2x_{2,k}^2 + B_0\right)}{MV_0}\right] sign(x_2, k) \\ &+ T\frac{AD_p\beta_e}{MV_0}u_k + Tw_{3,k} \end{aligned} \tag{3.3}$$

Based on [28, 29], $A$: piston cross-sectional area, $B_{\#}$: friction load in the system, $\beta_e$: effective bulk modulus (i.e., hydraulic 'stiffness' in the circuit), $D_p$: pump displacement, $L$: leakage coefficient, $M$: load mass, and $V_0$: initial cylinder volume. In addition to normal operation, two models were created. One model was based on a severe friction fault (i.e., the friction coefficients or terms were increased 3 times). Another model was based on a severe leakage fault (i.e., the leakage coefficients or terms were increased 4 times). The following system matrices represent the normal system mode of operation, friction fault, and leakage fault, respectively:

$$F_1 = \begin{bmatrix} 1 & 0.0001 & 0 \\ 0 & 1 & 0.0001 \\ 0 & -41.0258 & 0.6099 \end{bmatrix} \tag{3.4}$$

$$F_2 = \begin{bmatrix} 1 & 0.0001 & 0 \\ 0 & 1 & 0.0001 \\ 0 & -51.8627 & 0.2226 \end{bmatrix} \tag{3.5}$$

$$F_3 = \begin{bmatrix} 1 & 0.0001 & 0 \\ 0 & 1 & 0.0001 \\ 0 & -73.5364 & 0.6015 \end{bmatrix} \tag{3.6}$$

For this simulation, the same input gain matrices were applied. The input gain matrix used is defined as follows:

$$G = \begin{bmatrix} 0 \\ 0 \\ 0.0135 \end{bmatrix} \tag{3.7}$$

In an effort to make the simulated problem more challenging, system and measurement noises were added. The added noise statistical characteristics are as follows: zero mean, Gaussian-distributed (white) covariance. The measurement and system noise covariance's $Q$ and $R$ were diagonal matrices with elements of $1 \times 10^{-6}$. In addition, to test and compare the intelligent KF methods, a modelling error or uncertainties of 20% was included. The following three figures show the desired kinematic trajectories: position, velocity, and acceleration. For the first 1.5 seconds of the simulation, the system model used was 'normal'. At 1.5 seconds, a friction fault was injected. At 3 seconds, the system experienced a leakage fault. After another 1.5 seconds, a friction fault was injected again (note: the leakage fault was removed). During the last two seconds, the system operated normally.

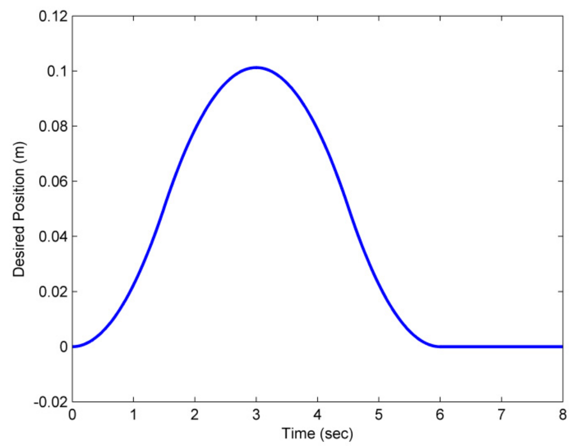


Figure 3. Desired flight surface position trajectory.

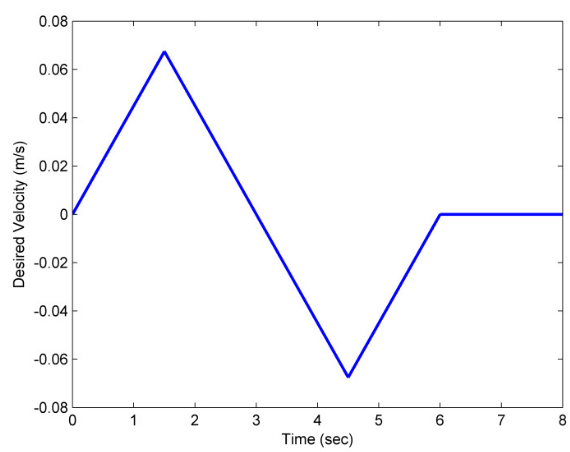


Figure 4. Desired flight surface velocity trajectory.

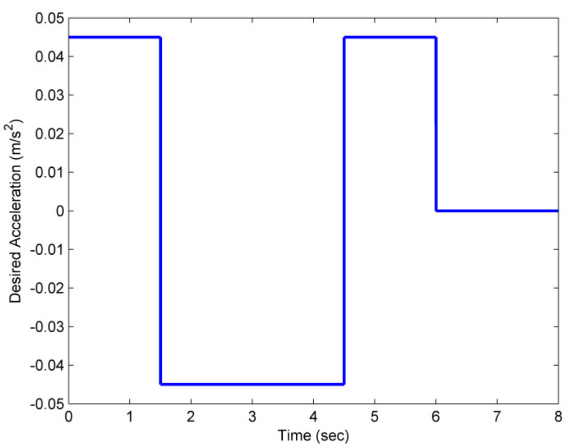


Figure 5. Desired flight surface acceleration trajectory.

The error results for the first five seconds of the simulation are shown in the next three figures. It was found that the fuzzy-based and ANN-based KFs yielded better results than the standard KF, which was to be expected given the strict assumptions of the KF. The fuzzy and ANN methods yielded the same results for the first state, however slightly deviated for the second and third states. Interestingly, the fuzzy-based KF yielded the best acceleration estimate and was the least sensitive to modeling errors and uncertainties. The results of the ANN method could be improved further by varying or optimizing the number of hidden layers and operating nodes.

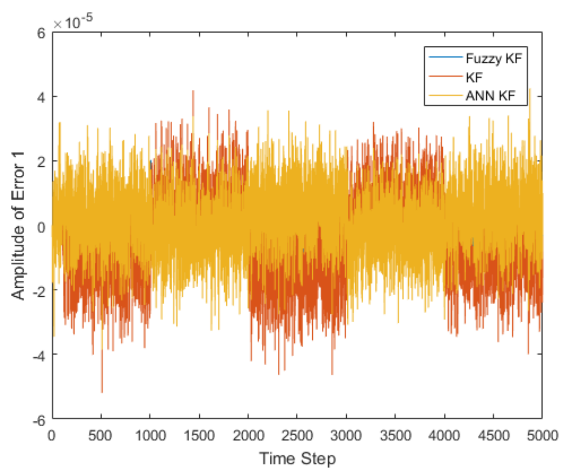


Figure 6. Position state error for KF-based methods.

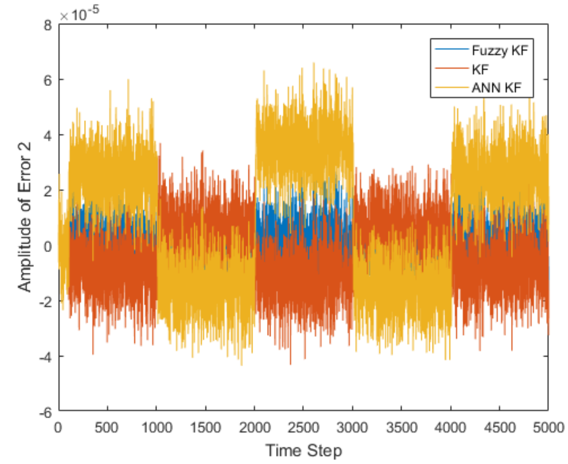


Figure 7. Velocity state error for KF-based methods.

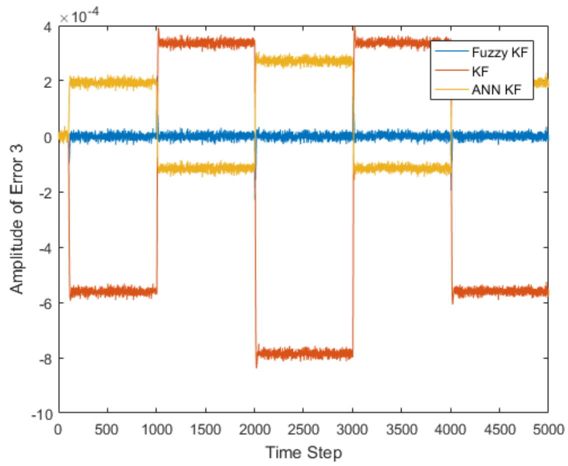


Figure 8. Acceleration state error for KF-based methods.

## IV. Conclusions and Future Work

This paper provided an overview and basic study of two intelligent Kalman filters (KFs). The standard KF, fuzzy-based KF, and artificial neural network (ANN) trained-KF were applied on a flight surface actuator. It was demonstrated that the intelligent-based KFs yielded accurate results in terms of estimation error, and were less sensitive to modeling uncertainties and changes. The intelligent KFs are often used in robotic systems and sensors. Future research will be more comprehensive and provide a more in-depth study. The authors will also look at comparing 'deep learning' strategies with the KF and other state and parameter estimation methods. The methodologies will be applied to a robotic system that is currently being built for experimentation.

## REFERENCES


[1] Y. Bar-Shalom, X. Rong Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation, New York: John Wiley and Sons, Inc., 2001.

[2] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering, Transactions of ASME,* vol. 82, pp. 35-45, 1960.

[3] B. D. O. Anderson and J. B. Moore, Optimal Filtering, Englewood Cliffs, NJ: Prentice-Hall, 1979.

[4] D. Simon, Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches, Wiley-Interscience, 2006.

[5] L. McGee and S. Schmidt, "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry," 1985.

[6] B. Ristic, S. Arulampalam and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Boston: Artech House, 2004.

[7] S. Haykin, Kalman Filtering and Neural Networks, New York: John Wiley and Sons, Inc., 2001.

[8] A. Gelb, Applied Optimal Estimation, Cambridge, MA: MIT Press, 1974.

[9] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," 2006.

[10] S. J. Julier, J. K. Ulhmann and H. F. Durrant-Whyte, "A New Method for Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control,* vol. 45, pp. 472-482, March 2000.

[11] R. Kalman and R. Bucy, "New Results in Linear Filtering and Prediction Theory," *ASME Journal of Basic Engineering,* vol. 83, pp. 95-108, March 1961.

[12] M. S. Grewal and A. P. Andrews, Kalman Filtering: Theory and Practice Using MATLAB, 3 ed., New York: John Wiley and Sons, Inc., 2008.

[13] S. A. Gadsden and A. S. Lee, "Advances of the Smooth Variable Structure Filter: Square-Root and Two-Pass Formulations," *Journal of Applied Remote Sensing,* vol. 11, no. 1, pp. 1-19, 2017.

[14] M. Al-Shabi, S. A. Gadsden and S. R. Habibi, "Kalman Filtering Strategies Utilizing the Chattering Effects of the Smooth Variable Structure Filter," *Signal Processing,* vol. 93, no. 2, pp. 420-431, 2013.

[15] S. A. Gadsden, S. R. Habibi and T. Kirubarajan, "Kalman and Smooth Variable Structure Filters for Robust Estimation," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 50, no. 2, pp. 1038-1050, 2014.

[16] G. Gerla, "Fuzzy Logic Programming and Fuzzy Control," *Studia Logica,* vol. 79, no. 2, pp. 231-254, 2005.

[17] E. Sobhani-Tehrani, K. Khorasani and S. Tafazoli, "Dynamic Neural Network-based Estimator for Fault Diagnosis in Reaction Wheel Actuator of Satellite Attitude Control System," *In Proceedings of the International Joint Conference on Neural Networks,* 2005.

[18] Y. Iiguni, H. Sakai and H. Tokumaru, "A Real-Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter," *IEEE Transactions on Signal Processing,* pp. 959-966, April 1992.

[19] G. V. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems,* vol. 2, 1989.

[20] P. Werbos, "Backpropagation Through Time: What it Does and How to Do It," *Proceedings of the IEEE,* vol. 78, p. 1550–1560, 1990.

[21] S. Singhal and . L. Wu, "Training Multilayer Perceptrons with the Extended Kalman Algorithm," *Advances in Neural Information Processing Systems 1,* p. 133–140, 1989.

[22] G. Puskorius and L. Feldkamp, "Decoupled Extended Kalman Filter Training of Feedforward Layered Networks," in *International Joint Conference on Neural Networks*, Seattle, 1991.

[23] R. J. Williams, "Training Recurrent Networks Using the Extended Kalman Filter," *IJCNN International Joint Conference on Neural Networks,* vol. 4, pp. 241-246, 1992.

[24] P. Sun and M. Kenneth, "Training Recurrent Neural Networks for Very High Performance with the Extended Kalman Algorithm," *Intelligent Engineering Systems Through Artificial Neural Networks,* vol. 8, pp. 121-126, 1998.

[25] X. Wang, N. McDowell, Kruger, Uwe and G. McCullough, "Semi-physical neural network model in detecting engine transient faults using the local approach," *17th World Congress International Federation of Automatic Control, IFAC,* July 2008.

[26] H. H. Afshari, S. A. Gadsden and S. R. Habibi, "Robust Fault Diagnosis of an Electro-Hydrostatic Actuator using the Novel Optimal Second-Order SVSF and IMM Strategy," *International Journal of Fluid Power,* vol. 15, no. 3, pp. 181-196, 2014.

[27] S. A. Gadsden, Y. Song and S. R. Habibi, "Novel Model-Based Estimators for the Purposes of Fault Detection and Diagnosis," *IEEE/ASME Transactions on Mechatronics,* vol. 18, no. 4, 2013.

[28] M. A. El Sayed, "Multiple Inner-Loop Control of an Electrohydrostatic Actuator," Hamilton, Ontario, 2012.

[29] S. A. Gadsden, "Smooth Variable Structure Filtering: Theory and Applications," Hamilton, Ontario, 2011.