Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques

Ryan Ahmed, *Student Member, IEEE*, Mohammed El Sayed, S. Andrew Gadsden, *Member, IEEE*, Jimi Tjong, and Saeid Habibi, *Member, IEEE*

***Abstract*—In this paper, an engine fault detection and classification technique using vibration data in the crank angle domain is presented. These data are used in conjunction with artificial neural networks (ANNs), which are applied to detect faults in a four-stroke gasoline engine built for experimentation. A comparative study is provided between the popular backpropagation (BP) method, the Levenberg–Marquardt (LM) method, the quasi-Newton (QN) method, the extended Kalman filter (EKF), and the smooth variable structure filter (SVSF). The SVSF is a relatively new estimation strategy, based on the sliding mode concept. It has been formulated to efficiently train ANNs and is consequently referred to as the SVSF-ANN. The accuracy of the proposed method is compared with the standard accuracy of the Kalman-based filters and the popular BP algorithms in an effort to validate the SVSF-ANN performance and application to engine fault detection and classification. The customizable fault diagnostic system is able to detect known engine faults with various degrees of severity, such as defective lash adjuster, piston chirp (PC), and chain tensioner (CT) problems. The technique can be used at any dealership or assembly plant to considerably reduce warranty costs for the company and manufacturer.**

***Index Terms*—Engines, estimation, extended Kalman filter (EKF), fault detection and classification, neural networks, smooth variable structure filter (SVSF).**

I. Introduction

In the last three decades, a range of artificial intelligence (AI) algorithms have been developed and applied to pattern classification problems. Among these, artificial neural networks (ANNs) have been prevalent as they are adaptive and show exceptional nonlinear input–output mapping ability [1]. ANNs are information processing models inspired by the human brain. The human brain has over 100 billion neurons that communicate with each other using chemical and electrical signals. ANNs are a mathematical rendition of neurons that communicate with one another and learn from experience. Training of an ANN is achieved using data sets that represents a specific input–output mapping. It is typically implemented in applications such as automatic vehicle control [2], pattern recognition [3], [4], function approximation, and robotic applications [5].

Fault detection and isolation (FDI) techniques are used for detecting fault conditions and isolating them. Therefore, FDI plays an important role in modern engineering systems due to increasing demand for safety and reliability, particularly in the automotive and the aerospace sectors. While different classical FDI techniques have been implemented, AI-based methods such as neural networks and fuzzy logic have been prevalent. These methods have proven to increase reliability and decrease the probability of producing false alarms. A fault is an unpredictable change in system behavior that deteriorates the system's performance. Two types of faults are considered: intermittent and permanent faults. Intermittent faults occur at intervals, which are usually irregular, in a system that functions normally at other times, whereas permanent faults exist from their inception until the faulty component or system is replaced or repaired.

FDI techniques have been divided into three main categories: signal-based fault detection, model-based fault detection, and AI techniques, as elaborated in the following signal-based fault detection that involves extracting the fault signature by comparing system measurements against their nominal operational trends. Analysis is performed in the time, frequency, or time–frequency domain for extracting features or trends that can be attributed to fault conditions [6]. Various signal-based FDI techniques, particularly for internal combustion engine (ICE) fault detection, have been discussed [7], [8]. Most of these FDI techniques use either noise levels, as well as pressure or vibration signals, to detect faults. In 1979, Chung *et al.* implemented an engine fault detection methodology using sound measurements by acquiring sound intensities using microphones [9], [10]. This technique is one of the oldest practices that have been implemented at the General Motors research laboratories [9]. The method can effectively generate a thorough mapping of engine noise using cross-spectral analysis [9]. This method is able to identify the noise source by using a noise-source ranking methodology [9]. In 2002, Leitzinger provided a comparison between laser Doppler vibrometers, microphones, and accelerometers to detect engine faults [11], [10]. The research concludes that microphones provide an easy noncontact measuring system; however, they

Manuscript received February 1, 2012; revised December 11, 2012, April 4, 2013, and June 12, 2013; accepted September 17, 2013. Date of publication April 21, 2014; date of current version January 13, 2015.

R. Ahmed and S. Habibi are with the Department of Mechanical Engineering, McMaster University, Hamilton, ON L8S4L7, Canada (e-mail: ryan.ahmed@mcmaster.ca).

M. El Sayed and J. Tjong are with the Ford Canada, Windsor, ON N8Y 1W2, Canada.

S. A. Gadsden is with the Department of Mechanical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250 USA (e-mail: gadsden@umbc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2014.2317736

may generate inconsistent results and produce false alarms [11]. In addition, the research shows that accelerometers and laser Doppler vibrometers provide more reliable measurements [11]. Acoustic tests on ICEs in a production environment using two overhead microphones to measure sound pressure are described in [12]. In [13], a real-time model-based methodology is presented for the diagnosis of sensor failures in automotive engine control systems. Faults considered include the manifold absolute pressure and throttle position sensors. Experimental results demonstrate the effectiveness of the proposed technique.

The qualitative trend analysis (QTA) is one of the most widely used feature extraction techniques. QTA is a data-driven FDI methodology that works by extracting features (trends) from the measured signals and accordingly takes decisions. QTA has been extensively applied for process fault detection and diagnosis [14]. Alternatively, feature extraction can be performed using discrete-wavelet-based techniques [15]. Fault detection using discrete wavelet transforms (DWTs) is broadly deliberated in [16]. DWT techniques involve two main steps: measured signal decomposition and signal edge detection that may occur due to faults.

Model-based FDI is mainly based on residual generation. Residuals represent inconsistencies between the actual physical system measurements and the mathematical model of the system. In general, model-based FDI techniques can generate residuals by using observers, parameter estimation, and parity space comparison. A model-based engine fault detection using cylinder pressure estimates, combustion heat release, and torque estimates from nonlinear observers was implemented by Kao and John [17] and Minghui and Moskwa [18]. Results from the proposed methodology showed relatively good performance, with fast convergence and stability. A neural-network-based adaptive observer for aircraft-engine parameter estimation is provided in [19]. This adaptive observer combines the Kalman filter (KF) with neural networks and is able to compensate for nonlinearities that cannot be handled simply by the filter. In [20], Chen *et al.* introduced a novel approach to the design of optimal observer-based residual generators for detecting incipient faults in flight control. This approach reduced the probability of generating false alarms [20]. Furthermore, an observer-based fault detection system in robots using nonlinear and fuzzy-logic residual evaluation is discussed in [21]. A fault diagnostic scheme for aircraft-engine sensor fault is presented in [22]. The proposed methodology can distinguish between modeling uncertainties and occurrence of faults to reduce false alarms. Observer-based FDI for a drivetrain of a Jaguar vehicle involving an automatic transmission is presented in [23]. A model representing the drivetrain of the vehicle is derived using nonlinear polynomials that relate manifold pressure, engine speed, and the wheel speed.

ANNs provide a powerful tool for fault detection and prognosis [24]. This is due to the fact that ANNs have powerful self-learning and self-adapting characteristics, effective online adaptation algorithms (in addition to their parallel and pipeline processing characteristics), good noise rejection capabilities, and excellent nonlinear approximation properties [25]. Furthermore, ANNs provide the aptitude to include models with partly known physical structure, resulting in semi-physical models. (Wang *et al.* described this in [26].)

The ANN-based fault detection represents a competitive advantage over other FDI techniques such as wavelet analysis, which has been extensively researched in the literature [10] since the ANN-based fault detection technique represents a blackbox generic approach applied to any fault condition without the need to know the specific crank angle where the fault occurs or a specific frequency. Accordingly, any fault condition can be further added to the algorithm. In addition, as more data sets become available, the ANN learning capability increases significantly and potentially generates higher classification accuracy.

A blend of physical modeling of main features and secondary effects by ANNs resulted in an enhanced overall performance. Numerous FDI applications that involve ANNs have been reported. Space-shuttle main engine modeling using feedforward neural networks with a sigmoidal activation function is presented in [27]. Leonard and Kramer discussed the application of the radial basis function networks for fault diagnosis and classification [28]. Naidu *et al.* applied backpropagation (BP) neural networks for sensor failure detection in process control systems [29].

Terra and Tinós applied ANN-based FDI to a three-joint PUMA manipulator [30]. They used a multilayer perceptron (MLP) trained with BP to reproduce the dynamic behavior of the robot. An ANN-based residual evaluation technique to detect and classify faults online in an industrial actuator benchmark problem is presented in [31]. The actuator in this benchmark problem was a brushless synchronous dc motor. Two faults were considered: actuator current fault due to an end-stop switch and a position sensor fault. Results show that the proposed ANN algorithm can predict and classify faults with relatively high accuracy. An FDI methodology to detect faults in robotic manipulators for nontrained trajectories is presented in [32]. Two-level neural networks are used for residual generation and residual evaluations. False alarms that occur due to modeling errors are avoided since the FDI methodology does not use a model.

Pattern recognition is the process of mapping patterns to various groups or categories [33]. It aims at classifying patterns to groups that share the same set of properties [34]. It has been implemented in many applications, such as machine vision [35], speech recognition [36], image processing and analysis [37], medical diagnosis [38], and fault detection and diagnostics [39]. In [40], a robust multisensor FDI technique applied to a marine diesel engine is discussed. In this study, four engine faults are induced. Since all of these malfunctions affect pressure, temperature, and vibration measurements, an ensemble of three neural networks are trained, i.e., one for each of the three mentioned signals. Consequently, a system that is resistant to sensor failure is obtained. The output of the ensemble is created by taking a majority vote over the three networks.

An ANN-based fault detection algorithm to detect lubrication faults of locomotive diesel engines using frequency-domain analysis of vibrations is presented in [41]. An experimental result on more than 50 lubrication pumps shows

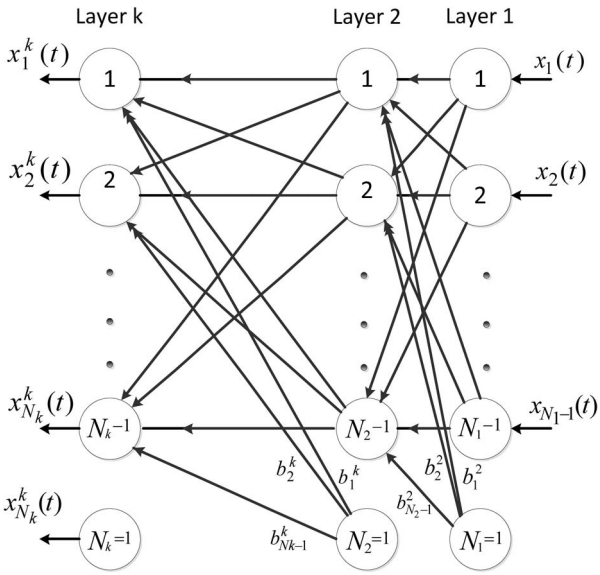

Fig. 1. Schematic of feedforward MLP network [13].

the effectiveness of the proposed methodology. Neural networks with N-version programming have been applied in [42]. The idea of N-version programming is to train multiple networks with different measurements signals (e.g., pressure, vibration, and temperature) for detecting fault conditions and selecting using a voting system for fault classification. This technique can be applied for safety-critical systems [43]. Detection of motor bearing faults using a signal-based ANN FDI technique is discussed in [44]. An ANN-based fault detection technique has been applied to engine fault detection in [45] and [46]. In these studies, two forms of new estimation strategy known as the smooth variable structure filter (SVSF) has been applied to train ANNs and applied to engine fault detection. This paper provides a detailed description of the presented technique and expands on the fault detection map. Moreover, an automotive ICE fault detection and classification technique using vibration data in the crank angle domain is presented. These data are used in conjunction with ANNs as applied to detect faults in a four-stroke eight-cylinder engine built for experimentation. A comparative study is provided between the commonly used BP, Levenberg–Marquardt (LM), quasi-Newton (QN), extended KF (EKF), and the recently proposed SVSF.

II. Feedforward Multilayered Neural Network

A multilayer feedforward network consists mainly of sensory units that constitute the input layer, one or more hidden layers, and an output layer.

As shown in Fig. 1, each node is connected to all nodes in the adjacent layer by links (weights), and each node computes a weighted sum of the inputs. An offset (bias) is added to the resultant sum followed by a nonlinear activation function application. The input signal propagates through the network in a forward direction on a layer-by-layer basis. Consequently, the network represents a static mapping between inputs and outputs.

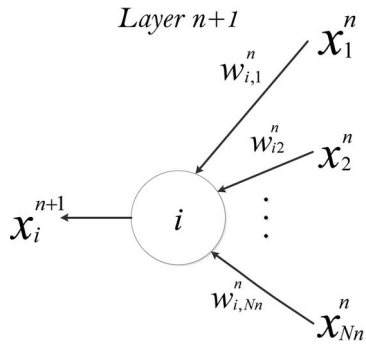

Fig. 2. Node $(n+1,\ i)$ representation.

Let k denote the total number of layers, including the input and output layers. node$(n,\ i)$ denotes the ith node in the nth layer, and $N_n - 1$ is the total number of nodes in the nth layer. As shown in Fig. 2, the operation of node$(n+1,\ i)$ is described by the following:

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

where $x_i^n(t)$ denotes the output of node$(n,\ j)$ for the t training pattern, and $w_{i,\,j}^n$ denotes the link weight from node$(n,\ j)$ to the node$(n+1,\ i)$. b_i^n is the node offset (bias) for node$(n,\ i)$.

The function $\varphi(\cdot)$ is a nonlinear sigmoid activation function defined by

$$\varphi(w) = \frac{1}{1+e^{-aw}}, \qquad a > 0; \quad -\infty < w < \infty. \tag{2}$$

For simplicity, the node bias is considered to be a link weight by setting the last input N_n to node$(n+1,\ i)$ to the value of one as follows:

$$x_{N_n}^n(t) = 1, \qquad 1 \le n \le k$$
$$w_{i,N_n}^n = b_i^{n+1}, \qquad 1 \le n \le k-1.$$

Consequently, (1) can be rewritten in the following form:

$$x_i^{n+1}(t) = \varphi\left(\sum_{j=1}^{N_n} w_{i,\,j}^n x_j^n(t)\right). \tag{3}$$

III. Global and Decoupled Extended Kalman Filter-Based Neutral Network Training

The EKF has been tailored to train feedforward neural networks by formulating the network as a filtering problem [45]. Accordingly, feedforward MLP network behavior can be described by a nonlinear discrete-time state-space representation [46] such that

$$w_{k+1} = w_k + \omega_k \tag{4}$$
$$y_k = C_k(w_k,\ u_k) + v_k. \tag{5}$$

Equation (4) represents the system equation. It demonstrates the neural network as a stationary system with an additional zero-mean white Gaussian noise ω_k with a covariance described

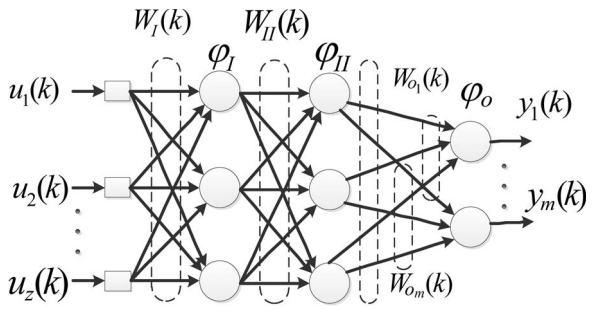

Fig. 3. Feedforward MLP with z inputs, two hidden layers, and m outputs.

by $[\omega_k\ \omega_l^T] = \delta_{k,l} Q_k$. Neural network weights and biases w_k are regarded as the system's state. Equation (5) is the measurement (observation) equation. It is a nonlinear equation relating network desired (target) response y_k to the network input u_k and weights w_k. The nonlinear function C_k represents the measurement function. A zero-mean white Gaussian noise v_k is added with a covariance defined as $[v_k\ v_l^T] = \delta_{k,l} R_k$.

Consider a feedforward MLP network with two hidden layers, as shown in Fig. 3. All activation functions of the first, second, and output layers are nonlinear sigmoidal functions denoted by φ_I, φ_{II}, and φ_o respectively. The network transfer function in terms of network weights, inputs, and activation functions can be mathematically defined as

$$y_i(k) = \varphi_o\left(w_{o_i}(k)\varphi_{II}\left(w_{II}(k)\varphi_I\left(w_I(k)u(k)\right)\right)\right) \tag{6}$$

where m denotes the number of output neurons, and w_I, w_{II}, and w_o are group weight matrices for the first hidden layer, the second hidden layer, and the output layer, respectively.

Linearization is performed by differentiating the network transfer function with respect to network synaptic weights (i.e., deriving the Jacobian). The Jacobian matrix $C_{k|\text{linearized}}$ can be mathematically expressed as follows:

$$C_{k|\text{linearized}} = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_1}{\partial w_2} & \cdots & \frac{\partial y_1}{\partial w_{N_T}} \\ \frac{\partial y_2}{\partial w_1} & \frac{\partial y_2}{\partial w_2} & & \frac{\partial y_2}{\partial w_{N_T}} \\ \vdots & & \ddots & \vdots \\ \frac{\partial y_m}{\partial w_1} & \frac{\partial y_m}{\partial w_2} & \vdots & \frac{\partial y_m}{\partial w_{N_T}} \end{bmatrix} \tag{7}$$

where N_T denotes total number of synaptic weights (including bias), and z specifies the number of input neurons. By differentiating (6) with respect to different weight groups w_I, w_{II}, and w_o, and for $i, l = 1, 2, \ldots, m$, the following is obtained:

$$\frac{\partial y_i}{\partial w_{o_l}} = \begin{cases} \acute{\varphi}_o\left(w_{o_i}\varphi_{II}\left(w_{II}\varphi_I(w_I u)\right)\right) \\ \quad \times \varphi_{II}\left(w_{II}\varphi_I(w_I u)\right), & \text{if } i = l \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\frac{\partial y_i}{\partial w_{II}} = \acute{\varphi}_o\left(w_{o_i}\varphi_{II}\left(w_{II}\varphi_I(w_I u)\right)\right) \times w_{o_i}\acute{\varphi}_{II}\left(w_{II}\varphi_I(w_I u)\right)\varphi_I(w_I u) \tag{9}$$

$$\frac{\partial y_i}{\partial w_I} = \acute{\varphi}_o\left(w_{o_i}\varphi_{II}\left(w_{II}\varphi_I(w_I u)\right)\right) \times w_{o_i}\acute{\varphi}_{II}\left(w_{II}\varphi_I(w_I u)\right) w_{II}\acute{\varphi}_I(w_I u). \tag{10}$$

By placing (8)–(10) in one matrix

$$C_{k|\text{linearized}} = \begin{bmatrix} \frac{\partial y}{\partial w_o} & \frac{\partial y}{\partial w_I} & \frac{\partial y}{\partial w_{II}} \end{bmatrix}. \tag{11}$$

$C_{k|\text{linearized}}$ is the m-by-N_T measurement matrix of the linearized model around the current weight estimate.

The decoupled EKF (DEKF)-based neural network training introduced by Singhal and Wu is known as the global EKF (GEKF) [47]. In the GEKF algorithm, all network weights and biases are simultaneously processed, and a second-order information matrix correlating each pair of network weights is obtained and updated [48]. Consequently, the GEKF computational complexity is $O(mN_T^2)$. A storage capacity of $O(N_T^2)$ is required, which is relatively high compared with the standard BP algorithm. The DEKF-based neural network training algorithm illustrated in the following represents the most general EKF-based neural network training method. The GEKF is a special form of the DEKF where the weight group number g is set to one. Neural network training using the DEKF algorithm can be expressed as follows [49]:

$$\Gamma_k = \left[\sum_{i=1}^{g} \left(C_k^i\right)^T P_k^i C_k^i + R_k\right]^{-1} \tag{12}$$

$$K_k^i = P_k^i C_k^i \mathbf{\Gamma_k} \tag{13}$$

$$\alpha_k = d_k - \hat{d}_k \tag{14}$$

$$\widehat{w}_{k+1}^i = \widehat{w}_k^i + K_k^i \alpha_k \tag{15}$$

$$P_{k+1}^i = P_k^i - K_k^i \left(C_k^i\right)^T P_k^i + Q_k^i \tag{16}$$

where the following nomenclature applies:

- $\mathbf{\Gamma}$ m-by-m matrix known as global scaling matrix (or global conversion factor);
- C n_i-by-m gradient matrix, which involves weights gradient with respect to every output node;
- α m-by-1 matrix representing innovation, which is the difference between desired (target) and actual network output;
- P n_i-by-n_i error covariance matrix;
- Q n_i-by-n_i process covariance matrix;
- K n_i-by-m Kalman gain matrix;
- R m-by-m measurement noise covariance matrix;
- $\hat{d}_k$ m-by-1 matrix representing actual network output;
- d_k m-by-1 matrix representing target (desired) output.

The above DEKF training algorithm operates in a serial mode fashion. In serial mode, one training sample is involved in error calculation, gradient computation, and synaptic weight update. A problem known as the "recency phenomenon" arises with serial mode when training tends to be influenced by the most recent samples [48]. Consequently, a trained network fails to remember former input–output mappings; thus, serial mode training reduces training performance. The recency phenomenon can be circumvented using the multistreaming training technique [50]–[52]. Multistreaming EKF training allows multiple training samples to be batched and processed. It involves training M multiple identical neural networks using

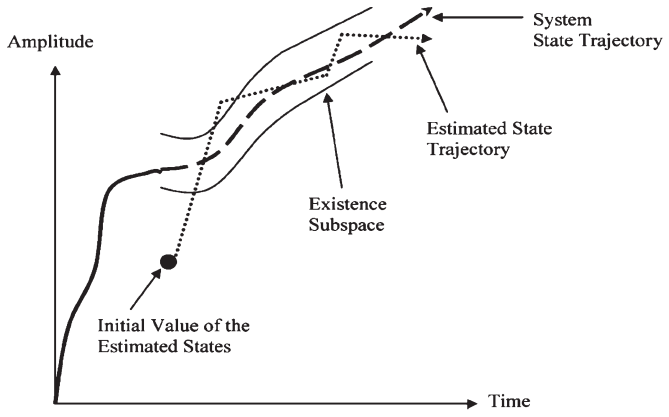

Fig. 4. SVSF estimation concept [54].

several training samples followed by weight update using overall networks' errors. The given algorithm can be adjusted to multistreaming mode by replacing matrix dimension m in Γ, C, α, K, and R with $M \times m$ [53].

IV. Smooth Variable Structure Filter-Based Artificial Neural Network Training

In 2007, the SVSF was introduced based on variable structure theory and sliding mode concepts [54]. It implements a switching gain to converge the estimates to within a boundary of the true states (i.e., existence subspace). In its present form, the SVSF has been shown to be stable and robust to modeling uncertainties and noise [55], [56]. The basic estimation concept of the SVSF is shown in Fig. 4. The SVSF method is model based and may be applied to differentiable linear or nonlinear dynamic equations. The original form of the SVSF as presented in [54] did not include covariance derivations. An augmented form of the SVSF was presented in [57], which includes a full derivation for the filter.

The SVSF can be applied for training nonlinear feedforward neural networks by estimating network weights. In the same fashion as the KF, the SVSF has been adapted to train feedforward neural networks by visualizing the network as a filtering problem where F, G, and C_k are the system, input, and output matrices, respectively, as follows:

$$\widehat{w}_{k+1|k} = F\widehat{w}_{k|k} + Gu_k \tag{17}$$

$$y_k = C_k(w_{k|k},\, u_k). \tag{18}$$

The global SVSF training algorithm is iterative and is summarized by the following steps, assuming a training data set defined by $\{x_k,\, z_k\}$:

Step 1: Network weight initialization

A priori state estimates (network weights) $\widehat{w}_{k|k}$ are randomly initialized ranging from -1 to $+1$.

Step 2: Calculation of the predicted (a posteriori) weight estimates $\widehat{w}_{k+1|k}$ from (17)

For neural network training, the system matrix F is an identity matrix, and the system input u_k is set to zero. Consequently, when the algorithm is initialized, the *a posteriori* weight matrix is the same as the *a priori*; thus, (17) is rewritten as follows:

$$\widehat{w}_{k+1|k} = \widehat{w}_{k|k}. \tag{19}$$

Step 3: Jacobian matrix calculation (linearization) of the measurement matrix C_k

The algorithm for Jacobian matrix calculation is the same as that stated earlier in (7). After applying the algorithm, $C_{k|\text{linearized}}$ is obtained, as in (11).

Step 4: Calculation of the estimated network output (measurements) $\hat{z}_{k+1|k}$

The linearized Jacobian measurement matrix $C_{k|\text{linearized}}$ and the *a priori* network weights $\widehat{w}_{k+1|k}$ yield the estimated network output as follows:

$$\hat{z}_{k+1|k} = C_{k|\text{linearized}}\widehat{w}_{k+1|k}. \tag{20}$$

Step 5: Measurement error $e_{z_{k+1|k}}$ calculation

Using the output $\hat{z}_{k+1|k}$ and the corresponding target (from the neural network training data set) z_k, the measurement errors $e_{z_{k+1|k}}$ may be calculated as follows:

$$e_{z_{k+1|k}} = z_k - \hat{z}_{k+1|k}. \tag{21}$$

Step 6: SVSF gain calculation

The SVSF gain is a function of the *a priori* and the *a posteriori* measurement errors $e_{z_{k+1|k}}$ and $e_{z_{k|k}}$, the smoothing boundary layer widths ψ, the "SVSF" memory or convergence rate γ, and the linear measurement matrix $C_{k|\text{linearized}}$. (For the derivation of the SVSF gain K_{k+1}, see [54] and [57].) The SVSF gain may be defined diagonally as follows:

$$K_{k+1} = C_{k|\text{linearized}}{}^{+}\text{diag}\left[\left(|e_{z_{k+1|k}} + \gamma|e_{z_{k|k}}|\right) \circ \text{sat}\left(\frac{e_{z_{k+1|k}}}{\psi}\right)\right]\text{diag}(e_{z_{k+1|k}})^{-1}. \tag{22}$$

Step 7: Calculation of the updated state estimates $\hat{w}_{k+1|k+1}$

The updated weights are calculated as follows:

$$\widehat{w}_{k+1|k+1} = \widehat{w}_{k+1|k} + K_{k+1}e_{z_{k+1|k}}. \tag{23}$$

Step 8: Calculation of a posteriori output estimate $\hat{z}_{k+1|k+1}$ and measurement errors $e_{k+1|k+1}$

Similar to the EKF strategy, the output estimates and *a posteriori* measurement errors are calculated, respectively, as follows:

$$\hat{z}_{k+1|k+1} = C_{k|\text{linearized}}\widehat{w}_{k+1|k+1} \tag{24}$$

$$e_{z_{k+1|k+1}} = z_{k+1} - \hat{z}_{k+1|k+1}. \tag{25}$$

Steps 3–8 are iteratively repeated while shuffling (randomly shifting) the training data set at each time step. Training proceeds until one of the stopping conditions (stated later) occurs. As per [54], the estimation process is stable and convergent if the following lemma is satisfied:

$$|e_{k|k}| > |e_{k+1|k+1}|. \tag{26}$$

The proof, as defined in [54], yields the derivation of the SVSF gain from (26). Expanding (16) using (15) and the standard SVSF gain yields the following:

$$e_{z,k+1|k+1} = e_{z,\,k+1|k} - HK_{k+1}. \tag{27}$$

Substituting (27) into (26) yields

$$|e_{z,\,k|k}| > |e_{z,\,k+1|k} - HK_{k+1}|. \tag{28}$$

By simplifying and rearranging (28), we obtain

$$|HK_{k+1}| > |e_{z,\,k+1|k}| + \gamma|e_{z,\,k|k}|. \tag{29}$$

Based on the fact that $|HK_{k+1}| = HK_{k+1} \circ \text{sign}(HK_{k+1})$, the standard SVSF gain can be derived from (29), i.e.,

$$K_{k+1} = H^{-1}\left(|e_{z,\,k+1|k}| + \gamma|e_{z,\,k|k}|\right) \circ \text{sign}(HK_{k+1}). \tag{30}$$

Equation (30) may be further expanded based on the fact that $\text{sign}(HK_{k+1}) = \text{sign}(e_{z,\,k+1|k})$, as per [54], such that

$$K_{k+1} = H^{-1}\left(|e_{z,\,k+1|k}| + \gamma|e_{z,\,k|k}|\right) \circ \text{sign}(e_{z,\,k+1|k}). \tag{31}$$

Note further that the SVSF switching may be smoothed out by the use of a saturation function, such that (31) becomes

$$K_{k+1} = H^{-1}\left(|e_{z,\,k+1|k}| + \gamma|e_{z,\,k|k}|\right) \circ \text{sat}(e_{z,\,k+1|k}) \tag{32}$$

where the saturation function is defined by

$$\text{sat}(e_{z,\,k+1|k}) = \begin{cases} 1, & e_{z,\,k+1|k} \geq 1 \\ e_{z,\,k+1|k}, & -1 < e_{z,\,k+1|k} < 1 \\ -1, & e_{z,\,k+1|k} \leq -1. \end{cases} \tag{33}$$

Finally, a smoothing boundary layer ψ may be added to further reduce the magnitude of chattering, leading to

$$K_{k+1} = H^{-1}\left(|e_{z,k+1|k}| + \gamma|e_{z,k|k}|\right) \circ \text{sat}(e_{z,\,k+1|k}/\psi). \tag{34}$$

Note that the gain described in (34) is slightly different than that presented earlier as (22). A diagonalized form was created, as described in [57] and [58], to formulate an SVSF derivation that included a covariance function. The form shown as (34) was presented as the original or "standard" SVSF in [54].

V. Experimental Setup

The experimental setup, as shown in Fig. 5, involves a four-stroke eight-cylinder gasoline engine. The test is performed at Ford's Powertrain Engineering Research and Development Centre (PERDC). The test is conducted in a semi-anechoic chamber to isolate the engine from any external noise that might affect the vibration response of the system.

The system block diagram is shown in Fig. 6. The dynamometer is controlled using ADACS software to control engine speed and load conditions [59]. A coolant tower is used to set the engine at desired temperature conditions. Real-time monitoring of the engine parameters is performed using ETAS/INCA software packages [60]. Engine vibration characteristics depend on the operating conditions, such as load,

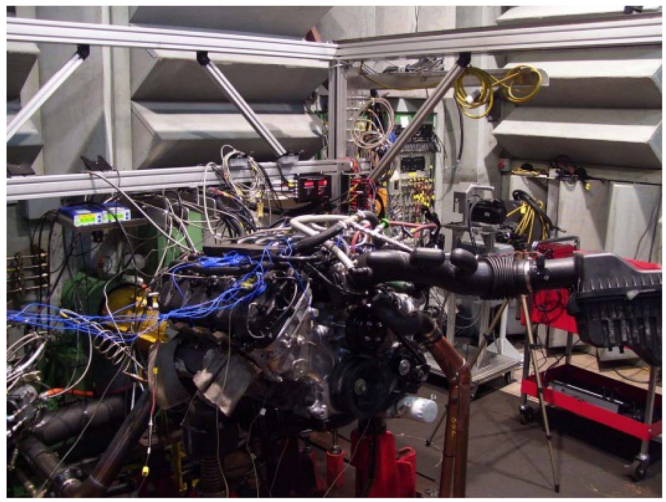

Fig. 5. Experimental setup in a semi-anechoic chamber at Ford's PERDC.

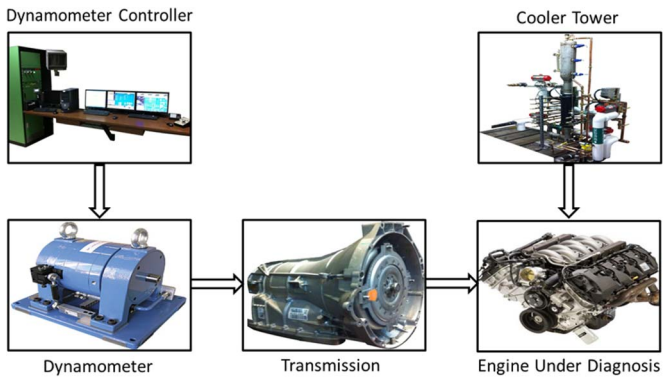

Fig. 6. Block diagram of the experimental setup.

speed, engine oil, and coolant temperatures. Engine temperatures may affect the vibration signatures, as engine components might expand or contract, and thus creates more or less clearance compared with normal operating conditions. Accordingly, all of these parameters have to be held constant for comparison purpose with other engines. Engines studied were run with a speed of approximately 2000 r/min. Engine cooling water and oil temperatures were held around 180 °F–190 °F. Charge-type triaxial piezoelectric accelerometers were used for vibration measurements. A triaxial piezoelectric sensor has been used to acquire data for this paper, but only one axis is used for network training. Piezoelectric accelerometers were used as they are relatively small in size, fairly linear, and can provide high durability, low cost, wide frequency range, and good measurement stability. In addition, they can withstand high temperatures such that they can be located on the engine where it gets hot due to radiation from the exhaust manifold. Accelerometers were calibrated using a calibration exciter to ensure data consistency across various tests. The accelerometer has been attached to the engine lug in a premeditated position to detect faults of interest. Vibration data were recorded over 4 s using a PROSIG 5600 data acquisition system with a built-in 16-bit analog-to-digital converter card set at a sampling frequency of 32 768 Hz [61].

After data acquisition, the time-domain data were converted offline to the crank angle domain using the cam identification

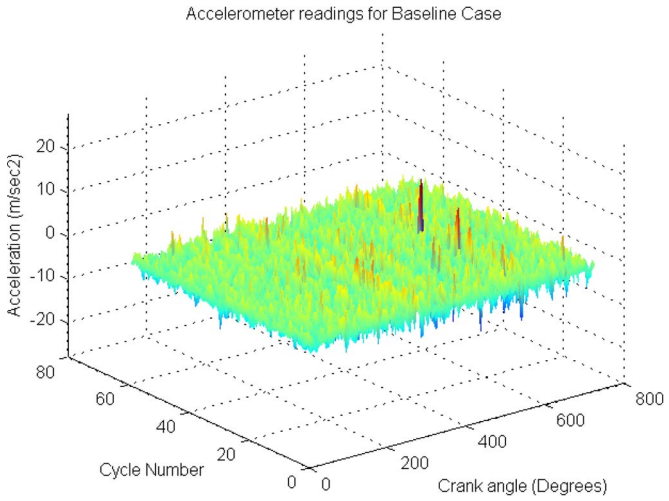

Fig. 7. Vibration data for the BL case in the crank angle domain.

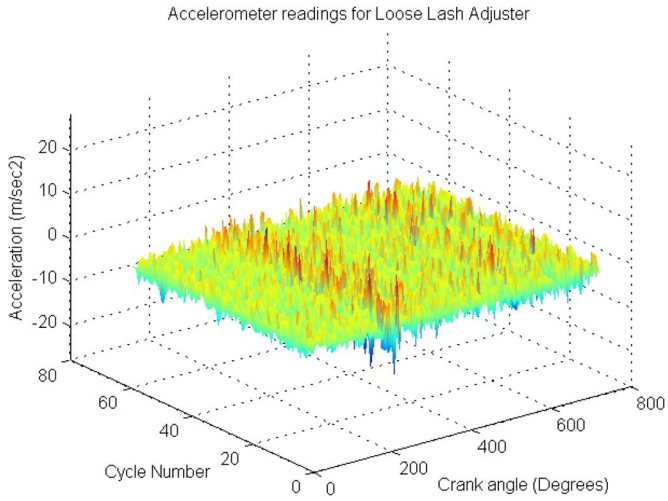

Fig. 8. Vibration data for the LA fault in the crank angle domain.

(CID) sensor signal. The CID sensor connector was located on the top portion of the front cover so that it was easily accessible in the vehicle. It is a noncontact electromagnetic sensor that detects the position of the camshaft angle and generates pulses at specific angles, i.e., crank angles of $90°-120°-60°-120°-60°-180°-90°$. The sinusoidal pulse zero-crossing indicates that the first cylinder is $10°$ away from the top-dead-center (TDC). After transformation to the crank angle domain, data resampling is performed so that each engine cycle has the same number of points. Seven faults were induced in the engine: missing bearing (MB) fault, piston chirp (PC) fault, chain tensioner (CT) fault, collapsed lash adjuster (severe) (CLA) fault, loose lash adjuster (LA) fault, chain sprocket (CS), and CC (CC) fault.

PC faults occur due to dislocation of the engine's piston ring, which leads to excessive wear and high engine noise. MB faults occur due to an assembly problem throughout the manufacturing process and cause severe vibration spikes. CT faults occur due to low oil pressure applied to the tensioner, which may cause a "rattling" noise and severe vibrations. A lash adjuster helps in preserving zero-valve clearance and leads to quiet operation, as well as avoids the necessity of intermittently fine-tuning the valve clearance. Under LA fault condition, oil leakage occurs; thus, the lash adjuster response is relatively slow compared with a healthy one, which leads to valve rattling noise. A faulty CS may produce a very fast "clicking" noise that occurs due to a manufacturing problem or excessive wear. CC noise occurs due to insufficient cap tightening torque.

Each of these faults has a specific vibration signature (see Fig. 7 for the baseline (BL) case) across various crank angle domain cycles, as shown in Figs. 8–14. Vibration signals recorded from these seven fault cases are used as a training data set for the ANN training used to generate the experimental results.

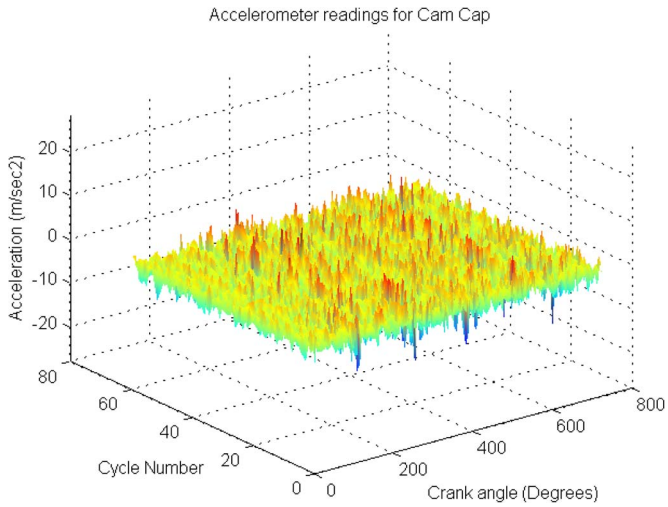

Fig. 9. Vibration data for the CC fault in the crank angle domain.

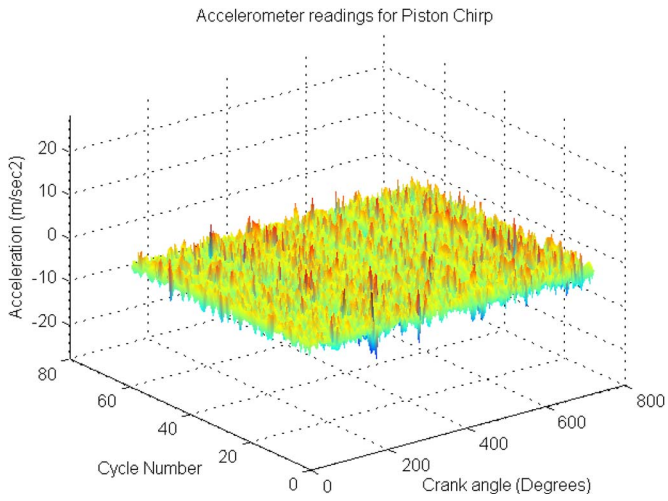

Fig. 10. Vibration data for the PC fault in the crank angle domain.

VI. Experimental Results

In this paper, a fully connected feedforward MLP is used with a number of input neurons representing sampled vibration data in the crank angle domain, two hidden layers with eight neurons each, and eight output units representing the network classification results. Trained networks should be able to

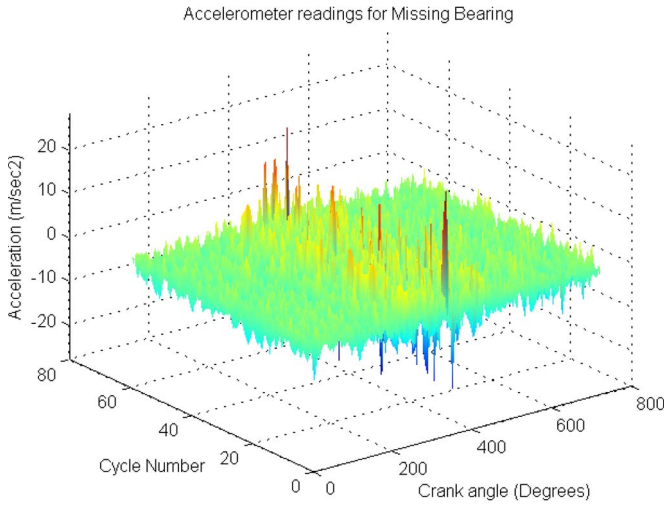

Fig. 11. Vibration data for the MB fault in the crank angle domain.

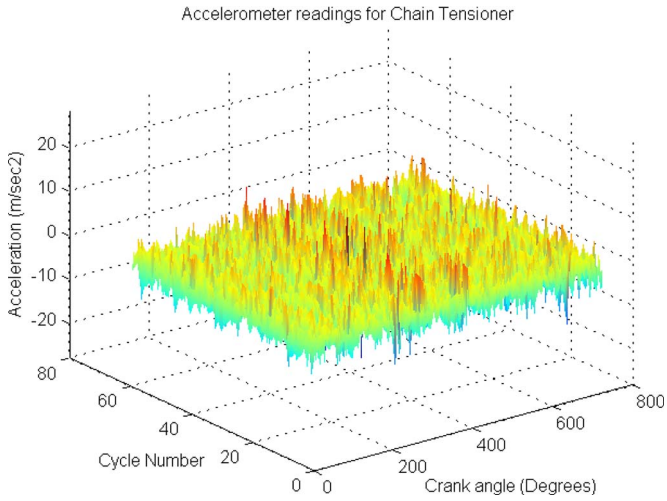

Fig. 12. Vibration data for the CT fault in the crank angle domain.

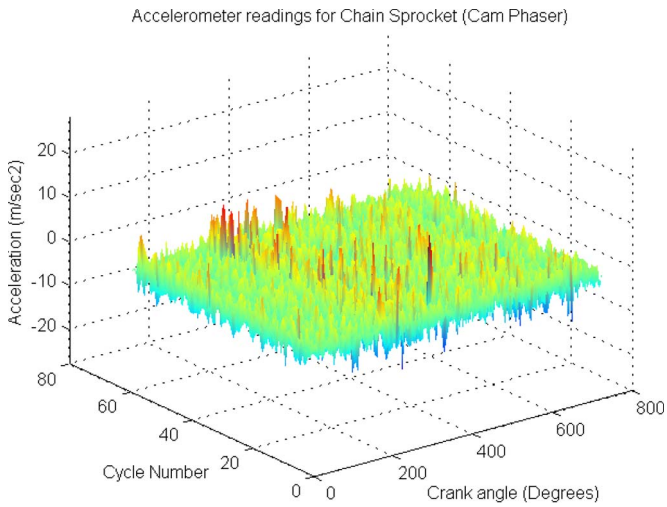

Fig. 13. Vibration data for the CS fault in the crank angle domain.

classify engine vibration samples to either one of the seven induced faults or BL case, as follows:

- (1, 0, 0, 0, 0, 0, 0, 0): BL case;

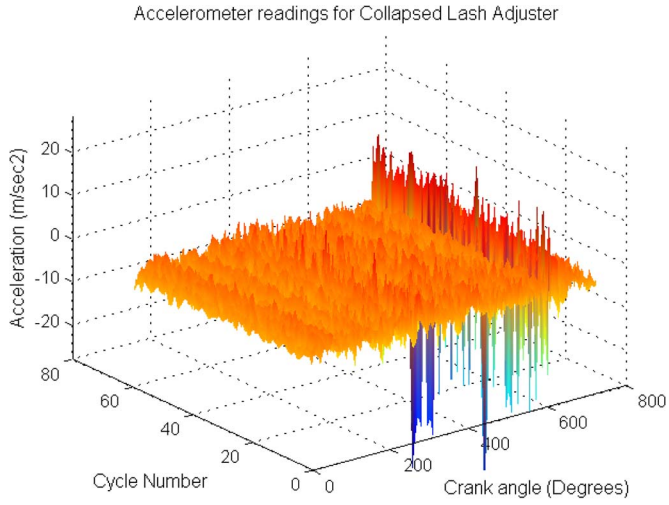

Fig. 14. Vibration data for the CLA fault in the crank angle domain.

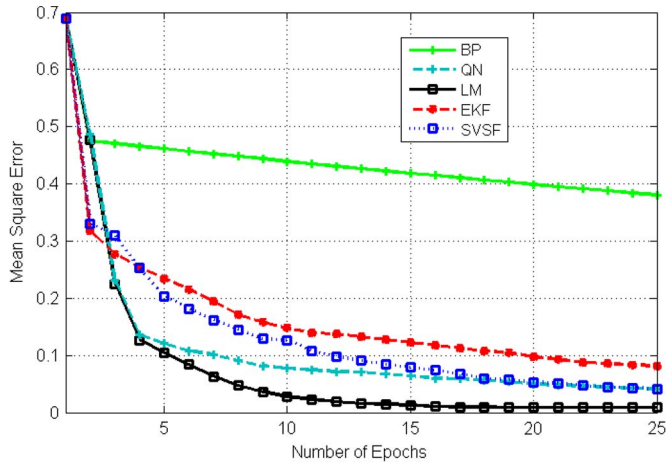

Fig. 15. Convergence rate of training techniques.

- (0, 1, 0, 0, 0, 0, 0, 0): MB fault detected;
- (0, 0, 1, 0, 0, 0, 0, 0): PC fault detected;
- (0, 0, 0, 1, 0, 0, 0, 0): CS fault detected;
- (0, 0, 0, 0, 1, 0, 0, 0): CC fault detected;
- (0, 0, 0, 0, 0, 1, 0, 0): CLA fault detected;
- (0, 0, 0, 0, 0, 0, 1, 0): CT fault detected;
- (0, 0, 0, 0, 0, 0, 0, 1): LA fault detected.

The test has been conducted through several runs: 50 engine cycles from each case, resulting in 400 training sets. Trained ANNs were tested using 25 engine cycles from each case, resulting in 200 testing sets. The networks were trained using the BP, EKF, LM, QN, and SVSF methods. Fig. 15 shows the MSE variation for the first 20 time steps. The reason why the particle-filter-based NN was not implemented and presented in this paper is mainly due to the fact that the method is computationally inefficient and complex. ANNs are computationally demanding without further adding complexity, making the overall strategy prohibitively computationally expensive. The SVSF converges faster than the BP and EKF and slower than the LM and QN. Note that the SVSF and QN reach the same MSE compared with the QN method after about 18 time steps. The aforementioned training techniques, except for the

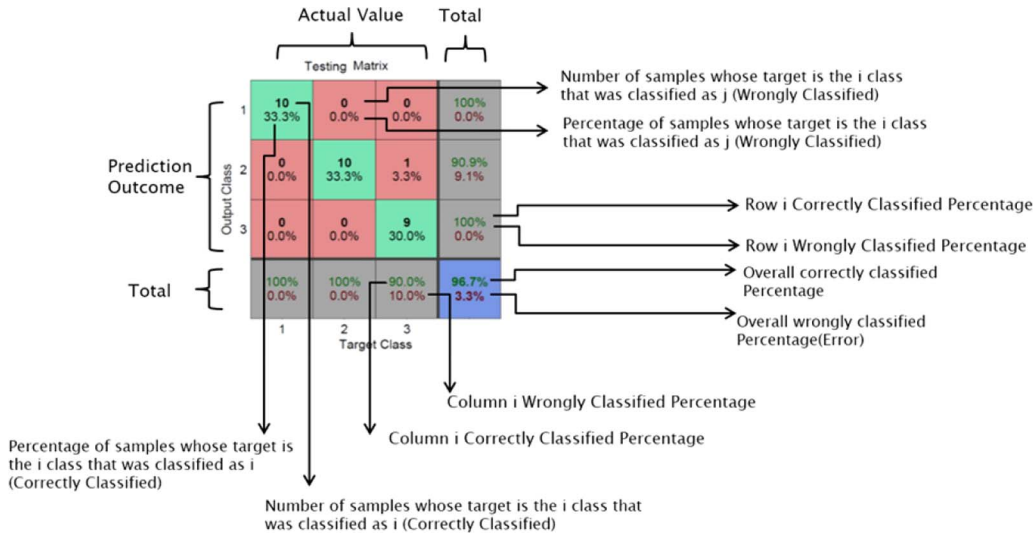

Fig. 16. Description of confusion matrix.

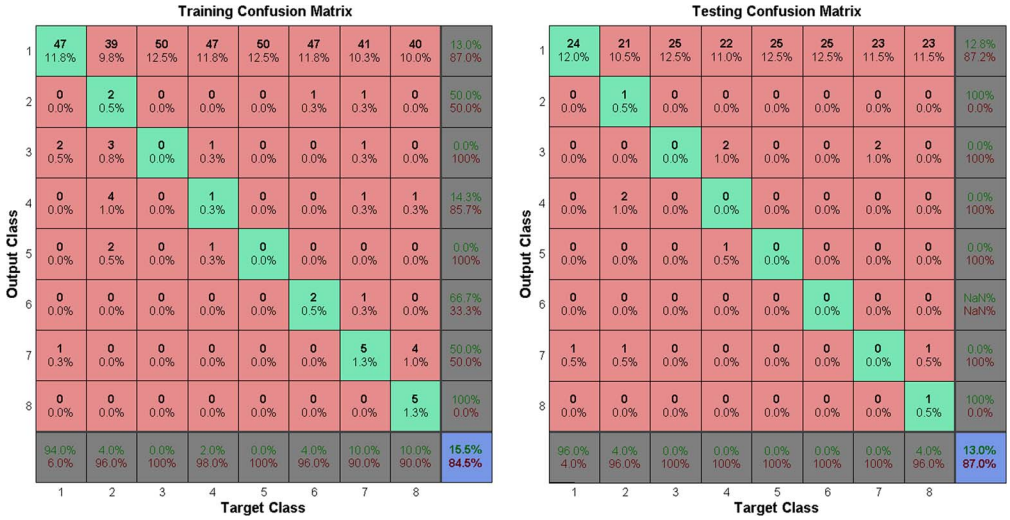

Fig. 17. Training and testing confusion matrices for the BP method.

BP and EKF, converge to within a similar degree of accuracy (less than 0.1 MSE) after about 12 or 13 time steps.

Training and testing results are summarized into confusion matrices, as described in Fig. 16.

Figs. 17–21 demonstrate training and testing classification results for trained networks using first-order BP, second-order LM, QN, EKF, and SVSF, respectively.

The EKF and SVSF are used in a global form and in a multistreaming fashion. Rows and columns are numbered from 1 to 8 as follows: 1) BL case; 2) MB case; 3) PC fault; 4) CS fault; 5) CC fault; 6) CLA; 7) CT fault; and 8) LA.

For the BP technique, the trained network failed to detect most of the fault conditions and misclassified all faults as the BL case, resulting in overall classification accuracy of 13%.

For the LM algorithm, the trained network shows poor generalization capability for CS and CT fault conditions (rows 4 and 7), resulting in classification accuracy (during testing) of 81%.

For the QN technique, the trained network misclassified almost half the number of cycles of the CS case (48%), thus resulting in 84.5% overall classification accuracy during testing. This fault condition is mostly seen as a BL case, which means that the network missed this fault condition (failed to detect existence of a fault condition).

For the EKF-based method, the network achieved 99.8% training accuracy and 96% testing accuracy. During testing, the trained network was able to classify all data sets (25 each) of the BL case, as well as the MB, CLA, and LA faults with 100% accuracy. However, the network misclassified one of the PC faults as the BL case, three cycles of the CS fault as the PC fault, three cycles of the CC fault as the BL case and the MB fault, and one cycle of the CT fault as the BL case.

For the SVSF-based method, the network achieved 100% training accuracy and 97% testing accuracy. During testing, the trained network generates a false alarm for one data set of the BL case as a CT fault. The network misclassified one data set of the PC fault as a BL case, one data set of the CS fault as the CT case, two of the CC faults as CS fault, and one CT fault as a CC fault.

Fig. 18. Training and testing confusion matrices for the LM method.

Fig. 19. Training and testing confusion matrices for the QN method.

Fig. 20. Training and testing confusion matrices for the EKF method.

Testing results are summarized in Table I. The SVSF achieved the highest testing (generalization) percentage in both training and testing, followed by the EKF, QN, LM, and BP methods. Results illustrate that conventional ANN training methods provide poor performance compared with estimation-based (SVSF and EKF-based) training techniques. This is

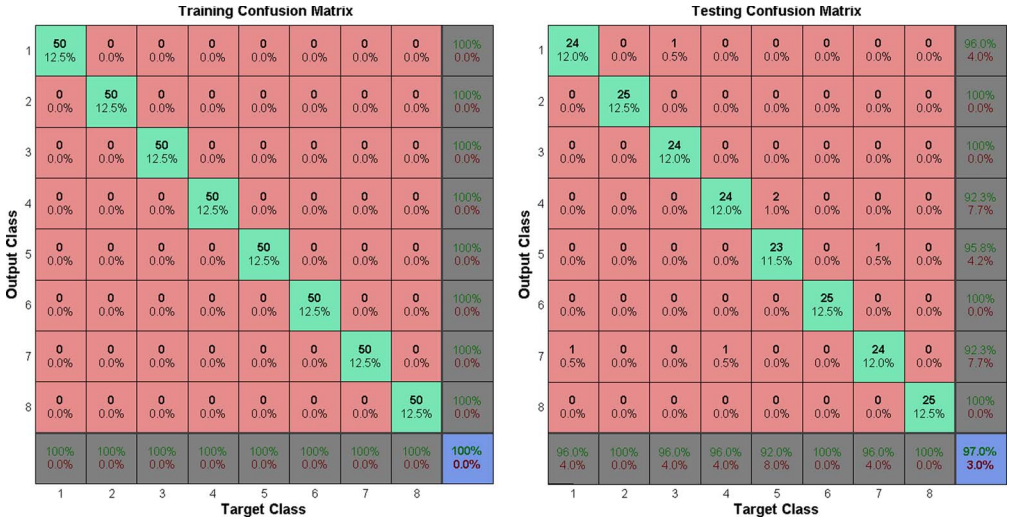

Fig. 21. Training and testing confusion matrices for the SVSF method.

TABLE I
FDI ACCURACY OF TRAINING METHODS

Training Method	Training (%)	Testing (%)
BP	15.5	13
LM	88.8	81
QN	88.3	84.5
EKF	99.8	96
SVSF	100	97

due to the fact that these methods suffer from local minima problems. However, SVSF and EKF-based techniques, since they involve second-order information and due to the chattering action of the SVSF gain, were able to escape the local minima and to reach the neighborhood of the global optimum.

The fault diagnostic system discussed in this paper could be used to effectively detect and isolate ICE MB, PC, CT, CLA, LA, CS, and CC defects at a dealership or assembly plant. The total time required for setting up the data acquisition hardware, collecting data, and performing FDI analysis is approximately 30 min. Implementation of the proposed FDI technique at every dealership and assembly plant significantly reduces warranty costs for the automotive original equipment manufacturer or the ICE supplier.

VII. CONCLUSION

In this paper, the SVSF has been presented in a global form for training of multilayered feedforward ANNs. The SVSF was successfully applied to detect and classify fault conditions in an ICE built for experimentation. The SVSF demonstrated stability, excellent generalization capability, and higher classification accuracy compared with other methods. The SVSF outperformed the popular first-order BP algorithm, and was comparable with the EKF algorithm.

A fault diagnostic system was developed and was successful in detecting a number of commonly occurring and known defects, such as defective lash adjuster, cam phaser, and CT. Engine defects can be identified using the proposed method with a relatively high success rate (97%). The engine fault diagnostic system is customizable and can be used to detect new faults by expanding the confusion matrix. Further work includes enhancing the classification accuracy by having more data sets and adding more accelerometers at different locations on the engine.

REFERENCES

[1] Z. Runxuan, "Efficient sequential and batch learning artificial neural network methods for classification problems," Ph.D. dissertation, School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, 2005.
[2] D. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*. Boston, MA, USA: Kluwer, 1993.
[3] E. Patuwo, M. Y. Hu, and M. S. Hung, "Two-group classification using neural netoworks," *Decis. Sci.*, vol. 24, no. 4, pp. 825–845, Jul. 1993.
[4] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *Amer. Statist.*, vol. 50, no. 4, pp. 284–293, 1996.
[5] R. A. Teixeira, A. De Baraga, and B. De Menezes, "Control of a robotic manipulator using artificial neural networks with on-line adaptation," *J. Neur. Process.*, vol. 12, no. 1, pp. 19–31, Aug. 2000.
[6] E. Sobhani-Tehrani, *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach*. New York, NY, USA: Springer-Verlag, 2009.
[7] R. Isermann, *Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. New York, NY, USA: Springer-Verlag, 2006.
[8] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[9] J. Chung, J. Pope, and D. Feldmaier, "Application of acoustic intensity measurement to engine noise evaluation," Soc. Automotive Eng., Warrendale, PA, USA, SAE Tech. Paper 790502, 1979.
[10] B. Ray, "Engine fault detection using wavelet analysis," M.S. thesis, Univ. Windsor, Windsor, ON, USA, 2007.
[11] E. Leitzinger, "Development of In-process engine defect detection methods using NVH indicators," M.S. thesis (M.A.Sc.), Univ. Windsor, Windsor, ON, USA, 2002.
[12] H. Jonuscheit, Acoustic Tests on Combustion Engines in Production, 2000.
[13] G. Rizzoni and P. S. Min, "Detection of sensor failure in automotive engines," *IEEE Trans. Veh. Technol.*, vol. 40, no. 2, pp. 487–500, May 1991.
[14] M. R. Maurya, P. K. Paritosh, R. Rengaswamy, and V. Venkatasubramanian, "A framework for on-line trend extraction and fault diagnosis," *Eng. Appl. Artif. Intell.*, vol. 23, no. 6, pp. 950–960, Sep. 2010.
[15] B. Bakshi and G. Stephanopoulos, "Representation of process trends—III. Multi-scale extraction of trends from process data," *Comput. Chem. Eng.*, vol. 18, no. 4, pp. 267–302, Apr. 1994.
[16] S. Postalcioglu, K. Erkan, and E. Bolat, "Discrete wavelet analysis based fault detection," *WSEAS Trans. Syst.*, vol. 5, no. 10, pp. 2391–2397, Oct. 2006.
[17] M. Kao and J. John, "Nonlinear diesel engine control and cylinder pressure observation," *Trans. ASME J. Dyn. Syst., Meas. Control*, vol. 117, no. 2, pp. 183–192, Jun. 1995.

[18] K. Minghui and J. J. Moskwa, "Model-based engine fault detection using cylinder pressure estimates from nonlinear observers," in *Proc. 33rd IEEE Conf. Decision Control*, 1994, pp. 2742–2747.
[19] R. K. Yedavalli, "Robust estimation and fault diagnostics for aircraft engines with uncertain model data," in *Proc. ACC*, 2007, pp. 2822–2827.
[20] J. Chen, R. Patton, and G. Liu, "Detecting incipient sensor faults in flight control systems," in *Proc. 3rd IEEE Conf. Control Appl.*, 1994, pp. 871–876.
[21] H. Schneider and P. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 3, pp. 274–282, May 1996.
[22] W. Li and R. K. Yedavalli, "Dynamic threshold method based aircraft engine sensor fault diagnosis," in *Proc. ASME Dyn. Syst. Control Conf.*, 2008, pp. 1179–1185.
[23] J. A. F. Vinsonneau, D. N. Sheilds, P. J. King, and K. J. Burnham, "Modelling and observer-based fault detection for an automotive drive train," in *Proc. ECC*, 2003.
[24] 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 *Proc. Int. Joint Conf. Neural Netw.*, 2005, pp. 2347–2352.
[25] G. V. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 3, pp. 303–314, 1989.
[26] X. Wang, N. McDowell, U. Kruger, and G. McCullough, "Semi-physical neural network model in detecting engine transient faults using the local approach," in *Proc. 17th World Congr. IFAC*, Jul. 2008, pp. 7086–7090.
[27] D. G. M. Sarvanan, "Modeling of the space shuttle main engine using feed-forward neural networks," in *Proc. Amer. Control Conf.*, 1993, pp. 2897–2899.
[28] J. A. Leonard and M. A. Kramer, "Diagnosing dynamic faults using modular neural-nets," *IEEE Expert Syst. Mag.*, vol. 8, no. 2, pp. 44–53, Apr. 1993.
[29] S. R. Naidu, E. Zafirou, and T. J. McAvoy, "Use of neural-networks for failure detection in a control system," *IEEE Control Syst. Mag.*, vol. 10, no. 3, pp. 49–55, Apr. 1990.
[30] M. H. Terra and R. Tinós, "Fault detection and isolation in robotic manipulators via neural networks—A comparison among three architectures for residual analysis," *J. Robot. Syst.*, vol. 18, no. 7, pp. 357–374, Jul. 2001.
[31] B. Koppen-Seliger, P. Frank, and A. Wolff, "Residual evaluation for fault detection and isolation with RCE neural networks," in *Proc. Amer. Control Conf.*, 1995, vol. 5, pp. 3264–3268.
[32] M. Terra and R. Tinos, "Fault detection and isolation in robotic systems via artificial neural networks," in *Proc. 37th IEEE Conf. Decision Control*, 1998, vol. 2, pp. 1605–1610.
[33] R. O. Duda, P. Hart, and D. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2000.
[34] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. San Diego, CA, USA: Elsevier, 2008.
[35] W. E. Snyder and H. Qi, *Machine Vision*. Cambridge, U.K.: Cambridge Univ., 2004.
[36] N. Morgan and H. Bourald, "Continuous speech recognition using multilayer perceptrons with hidden Markov models," in *Proc. Int. Conf. Accoustics, Speech, Signal Process.*, 1990, pp. 413–416.
[37] R. Kasturi and M. Trivedi, *Image Analysis Applications*. New York, NY, USA: Marcel Dekker, 1990.
[38] P. Degoulet, *Introduction to Clinical Informatics*. Berlin, Germany: Springer-Verlag, 1996.
[39] J. Zarei, P. Javad, and P. Majjd, "Bearing fault detection in induction motor using pattern recognition techniques," in *Proc. IEEE 2nd Int. Power Energy Conf.*, 2008, pp. 749–753.
[40] A. J. Sharkey, G. 0. Chandroth, and N. E. Sharkey, "Acoustic emission, cylinder pressure and vibration: A multisensor approach to robust fault diagnosis," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2000, vol. 6, pp. 223–228.
[41] K. Wang, "Neural network approach to vibration feature selection and multiple fault detection for mechanical systems," in *Proc. 1st Int. Conf. Innovative Comput., Inf. Control*, 2006, pp. 431–434.
[42] A. Sharkey and N. Sharkey, "Combining diverse neural nets," *Knowl. Eng. Rev.*, vol. 12, no. 3, pp. 231–247, Sep. 1997.
[43] A. Sharkey, N. Sharkey, and O. Gopinath, "Diversity, neural nets and safety critical applications," in *Current Trends in Connectionism*, L. F. Niklasson and M. B. Boden, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum, 1995, pp. 165–178.
[44] B. Li, G. Goddu, and M.-Y. Chow, "Detection of common motor bearing faults using frequency-domain vibration signals and a neural network based approach," in *Proc. Amer. Control Conf.*, 1998, vol. 4, pp. 2032–2036.
[45] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.
[46] P. Trebaticky and P. Jiri, "Neural network training with extended kalman filter using graphics processing unit," in *Proc. ICANN*, 2008, pp. 198–207.
[47] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended Kalman algorithm," in *Advances in Neural Information Processing Systems*. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 133–140.
[48] S. Haykin, *Kalman Filtering and Neural Networks*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.
[49] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 959–966, Apr. 1992.
[50] L. Feldkamp and G. Puskorius, "A signal processing framework based on dynamic networks with application to problems in adaptation, filtering, and classification," *Proc. IEEE*, vol. 86, no. 11, pp. 2259–2277, Nov. 1998.
[51] L. A. Feldkamp and G. V. Puskorius, "Training controllers for robustness: Multi-stream DEKF," in *Proc. IEEE Int. Conf. Neural Netw.*, 1994, pp. 2377–2382.
[52] L. A. Feldkamp and G. V. Puskorius, "Training of robust neurocontrollers," in *Proc. IEEE Int. Conf. Decision Control*, 1994, pp. 2754–2759.
[53] F. Heimes, "Extended Kalman filter neural network training: Experimental results and algorithm improvements," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 1998, pp. 1639–1644.
[54] S. R. Habibi, "The smooth variable structure filter," *Proc. IEEE*, vol. 95, no. 5, pp. 1026–1059, May 2007.
[55] S. R. Habibi and R. Burton, "The variable structure filter," *J. Dyn. Syst., Meas., Control*, vol. 125, no. 3, pp. 287–293, Sep. 2003.
[56] S. R. Habibi and R. Burton, "Parameter identification for a high performance hydrostatic actuation system using the variable structure filter concept," *ASME J. Dyn. Syst., Meas., Control*, vol. 129, no. 2, pp. 229–235, 2006.
[57] S. A. Gadsden and S. R. Habibi, "A new form of the smooth variable structure filter with a covariance derivation," in *Proc. IEEE Conf. Decision Control*, 2010, pp. 7389–7394.
[58] S. A. Gadsden, "Smooth variable structure filtering: theory and applications," Ph.D. dissertation, McMaster Univ., Hamilton, ON, Canada, 2011.
[59] [Online]. Available: http://www.horiba.com/automotive-test-systems/
[60] [Online]. Available: http://www.etas.com/en/products/inca_software_products.php
[61] [Online]. Available: http://prosig.com/solution/nvh/nvhPro.html

Ryan Ahmed (S'14) received the Bachelor of mechanical engineering degree with a mechatronics major from Ain Shams University, Cairo, Egypt, in 2007. He is currently working toward the Ph.D. degree with the Department of Mechanical Engineering, McMaster University, Hamilton, ON, Canada.

He was a Teaching Assistant with Ain Shams University and a Laboratory Mentor with The American University in Cairo, New Cairo, Egypt. He is currently a member of the Green Auto Power Train research team and the Centre for Mechatronics and Hybrid Technology, McMaster University. His research interests include artificial neural networks, control systems, state and parameter estimation, fault detection and diagnosis, and hybrid systems.

Mr. Ahmed is a Certified Programmable Logic Controller/Human–Machine Interface Programmer.

Mohammed El Sayed received the B.Sc. degree in mechanical engineering, the M.Sc. degree in mechanical engineering, and the Ph.D. degree in the areas of control theory and applied mechatronics from Helwan University, Helwan, Egypt, in 1999, 2003, and 2012.

He is currently with Ford Canada, Windsor, ON, Canada. His research interests include fluid power and hydraulics, state and parameter estimation, intelligent and multivariable control, actuation systems, and fault detection and diagnosis.

Dr. El Sayed is a Registered Member of Professional Engineers Ontario.

S. Andrew Gadsden (M'13) received the Ph.D. degree in the area of state and parameter estimation theory from the Department of Mechanical Engineering, McMaster University, Hamilton, ON Canada, in 2011. His work involved an optimal realization and further advancement of the smooth variable structure filter (SVSF).

He is currently an Assistant Professor with the Department of Mechanical Engineering, the University of Maryland, Baltimore County, Baltimore, MD, USA. His background includes a broad consideration of state and parameter estimation strategies, the variable structure theory, fault detection and diagnosis, mechatronics, target tracking, cognitive systems, and neural networks.

Dr. Gadsden is the recipient of a number of professional and scholarly awards and was a postdoctoral Fellow with the Centre for Mechatronics and Hybrid Technology at McMaster. He is an Associate Editor of the *Transactions of the Canadian Society for Mechanical Engineering* and is a member of the Professional Engineers of Ontario (PEO) and the Ontario Society of Professional Engineers (OSPE). He is also a member of the American Society of Mechanical Engineers (ASME) and the Project Management Institute (PMI).

Jimi Tjong is the Technical Leader and Manager of the Powertrain Engineering, Research and Development Centre (PERDC), Ford Canada, Windsor. It is a result-oriented organization capable of providing services ranging from the definition of the problem to the actual design, testing, verification, and, finally, the implementation of solutions or measures. The center is currently the hub for engineering, research, and development that involves Canadian universities, government laboratories, and Canadian automotive parts and equipment suppliers. The center includes 16 research and development test cells; prototype machine shop; and plug-in hybrid electric vehicle (PHEV), hybrid electric vehicle (HEV), and battery electric vehicle (BEV) development testing, which occupies an area of 200 000 ^{2}. The center is the hub for production/design validation of engines manufactured in North America and an overflow for the Ford worldwide facilities. It also establishes a close link worldwide within Ford Research and Innovation Centre, Product Development and Manufacturing Operations, that can help bridge the gap between laboratory research and the successful commercialization and integration of promising new technologies into the product development cycle. His principal field of research and development encompasses the following: optimizing automotive test systems for cost, performance, and full compatibility. It includes the development of test methodology and cognitive systems; calibration for internal combustion engines; alternate fuels, bio fuels, lubricants, and exhaust fluids; lightweight materials with the focus on aluminum, magnesium, and bio materials; battery, electric motors, supercapacitors, stop/start systems, HEV, PHEV, and BEV systems; nanosensors and actuators; high-performance and racing engines; nondestructive monitoring of manufacturing and assembly processes; and advanced gasoline and diesel engines focusing in fuel economy, performance, and cost opportunities. He has published and presented numerous technical papers in these fields internationally. He is also an Adjunct Professor with the University of Windsor, Windsor, ON; McMaster University; and the University of Toronto, Toronto, ON. He continuously mentors graduate students in completing the course requirements, as well as career development coaching.

Saeid Habibi (M'13) received the Ph.D. degree in control engineering from the University of Cambridge, Cambridge, U.K.

He is currently the Director of the Centre for Mechatronics and Hybrid Technology and a Professor with the Department of Mechanical Engineering, McMaster University, Hamilton, ON, Canada. His academic background includes research into intelligent control, state and parameter estimation, fault diagnosis and prediction, variable structure systems, and fluid power. The application areas for his research have included aerospace, automotive, water distribution, robotics, and actuation systems. He spent a number of years in the industry as a Project Manager and Senior Consultant for Cambridge Control Ltd., U.K., and as Senior Manager of Systems Engineering for Allied Signal Aerospace Canada.

Dr. Habibi is a member of the American Society of Mechanical Engineers (ASME) and the ASME Fluid Power Systems Division Executive Committee. He is with the Editorial Board of the *Transactions of the Canadian Society for Mechanical Engineering*. He received two corporate awards for his contributions to the Allied Signal Systems Engineering Process in 1996 and 1997. He received the Institution of Electrical Engineers F.C. Williams Best Paper Award in 1992 for his contribution to variable structure systems theory. He also received a Natural Sciences and Engineering Research Council of Canada International Postdoctoral Fellowship that he held at the University of Toronto, Toronto, ON, from 1993 to 1995; more recently, he held a Boeing Visiting Scholar sponsorship for 2005.