Latin American Applied Research 45:261-270 (2015)

SENSOR FUSION BASED UPON SELF ORGANIZING MAPS AND LOCAL KALMAN FILTERS FOR LOCAL FAULT DIAGNOSIS

H. BENÍTEZ-PÉREZ[†], M. VÁZQUEZ-HERNÁNDEZ[†] AND A. DURÁN-CHAVESTI[†]

[†]*I.I.M.A.S., D.I.S.C.A., U.N.A.M. Apdo. Postal 20-706, Del. A. Obregón, C.P. 01000, México,*
hector.benitez@iimas.unam.mx

Abstract— **A challenging strategy for fault diagnosis is related to data fusion and feature extraction such as having local models of the plant integrated through patterns. To pursuit this goal a group of Kalman Filters is proposed, one per sensor in order to estimate a state of the plant per sensor. Thereafter, a self-organized map processes the information in order to determine the response areas within the map. Moreover, Kalman filter auto tuning procedure, becomes an interesting issue due to correlated noise amongst them as well as local state dependency at this point this may call a pre-processing stage. Furthermore, once global state vector is produced, the classification stage takes place as long as this vector exists as post processing stage. The resulting maps determine areas of fault free scenarios or non-linear and no healthy responses classified as faults. This last point is obtained through a high accuracy of the system.**

Keywords— **Self Organizing Maps, Fault Isolation, Feature extraction, Local Fault Diagnosis.**

I. INTRODUCTION

The objective of the paper is to determine a local model based strategy capable to detect early condition at fault presence as well as an accurate classification during neglected stages. Furthermore, classification should be stable in terms of the evolving fault conditions. Classic approaches tend to be restrictive to fault detection model and/or numeric conditions of system response. Therefore, it is necessary to establish certain conditions where model based techniques are useful in a local scale with certain relationships but not suitable for general and global relationship, then, the use of a map is required in order to mitigate multivariable datum difference.

There are many processes for fault detection and their respective solution based upon data. The use of ARMAX algorithms allows to constructing a polynomial representation of a faulty situation which is very useful up to certain extended version considering fault evolution through time, however, it is very restricted to be responsive for unknown fault scenarios as presented by Benitez-Perez *et al*. (2009). In the case of using wavelets, this is quite useful technique for different kind of faults considering frequency and power variations, however, early fault detection becomes difficult and quite uncertain response where a partial model is constructed. Therefore it may be useful for early fault detection when fault is bounded up to certain extend to use an alternative approximation for like polynomic strategy to determine certain signature of the fault. In this paper, the use of Kalman filter allows to cope with system variations presented at ARMAX approach and performs a prompt response during fault evolution whereas wavelet based approach is not entirely guaranteed.

Moreover, the use of SOM presents de advantage of strong classification technique by creating a new and consistent map, whereas in the case of other NN it is not guaranteed this multiple classification. The following technique confirms the basic idea of classifying multiple unknown scenarios based upon two techniques integrated for classification purposes, the advantage of presented strategy is the fast response for detection and classified patterns although is not ultimate strategy available to be implemented. It constitutes another possibility on to this basic idea.

It is important to mention that fault diagnosis it is feasible during the evolution of the fault due to its deviation from formal response. This deviation may occur due to several reasons, and it will evolve at different stages since several conditions. One of them it is the inherent nonlinearity that may accomplish during a particular fault. What is important is to isolate this inherent nonlinearity in order to study its behavior this is the main goal of this paper.

II. GENERAL BACKGROUND

The Kalman filter, also known as linear quadratic estimation (LQE) and is used as part of sensor fusion, is an algorithm that uses a series of measurements observed over time, containing noise namely random variations or inaccuracies, it estimates certain unknown variables that tend to be more precise than those based on a single measurement alone. Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state.

A common application of the Kalman filter is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics.

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces an estimation current state variable, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimations are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state; no additional past information is required.

Extensions and generalizations to the method have also been developed, such as the extended Kalman filter and the unscented Kalman filter which work on nonlinear systems. The underlying model is a Bayesian model similar to a hidden Markov model but where the state space of the latent variables is continuous and where all latent and observed variables have Gaussian distributions (Tong *et al.*, 2008; Mohinder and Angus, 2008).

Second algorithm used in this paper, the self organizing map that the purpose of data (Kohonen, 1989) self-organizing feature map is to capture the topology and probability distribution of input data (Kohonen, 1989 and Hassoum, 1995; Figure 1). Firstly, a topology of self-organizing map is defined as a rectangular grid (Nelles, 2001; Figure 2). Different types of grid may be used, although Fig. 2 presents a homogenous response suitable for noise cancellation. The neighborhood function with respect to a rectangular grid is based upon bi-dimensional Gaussian functions shown in Eq. 1 which is classic with respect to other such representtations (Benítez-Pérez *et al*., 2007b; Benítez-Pérez and Benítez-Pérez, 2009 and Benítez-Pérez *et al*., 2012). This sort of grid allows regular representation where a clear isolation from fault free and fault responses is achievable. In here, the novelty of this work is to present a feature extraction approach for fault isolation based upon enhance information using Kalman filter and a particular grid.

Equation 1 is the basis of the neural network, in this equation the weight matrix is updated based upon bi-dimensional indexing named $h(i_1,i_2)$. This equation is used during training (off-line) stage.

$$h(i_1,i_2)=\exp\left(-0.5*\frac{\left(i_1^{win}-i_1\right)^2+\left(i_2^{win}-i_2\right)^2}{\sigma^2}\right) \quad (1)$$

where i_1 and i_2 are the index of each neuron. σ is the standard deviation from each Gaussian distribution. This distribution determines how the neurons next to winner neuron are modified. Each neuron has a weight vector (w_i^j) that represents how this is modified by an input updating. $h(i_1,i_2)$ is the Gaussian representation that permits the modification of neighbor neurons. This bi-dimensional function allows the weight matrix to be updated in a global way rather than just to update the weight vector associated to the winner neuron.

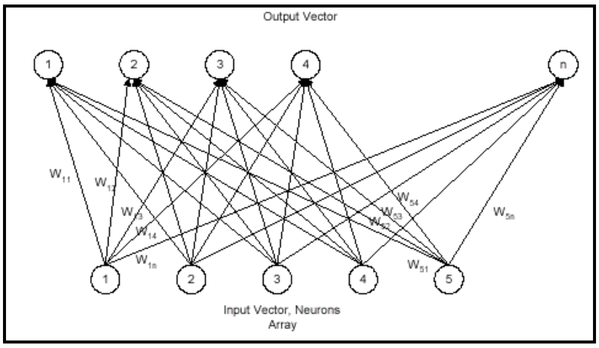

Figure 1. Topology Network.

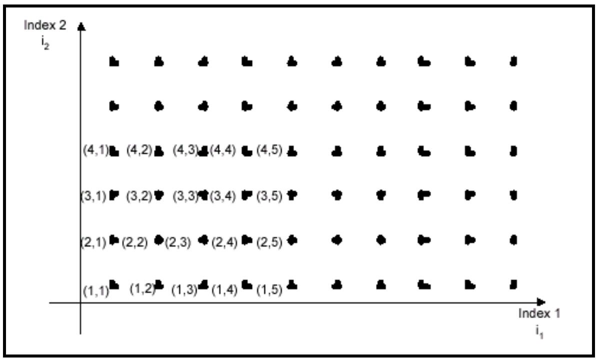

Figure 2. Index Grid.

An inner product is performed between weight matrix *W* and input vector (*I*) in order to define the winner neuron. Having calculated this product, the maxi-mum value is determined by the comparison between each scalar from resultant vector. This value is declared as winner as in the technique named the winner take all. The related bi-dimensional index (Fig. 2) is calculated in order to determine how the weight matrix is modified (Benítez-Pérez and Benítez-Pérez, 2009 and Benítez-Pérez *et al*., 2012).

The process updating the weight matrix is achieved as in Eq. 2.

$$w_j^{new}=w_j^{old}+\eta*h\left(i_1,i_2\right)*\left(I-w_j^{old}\right) \quad (2)$$

where η represents a constant value equals to 0.7. This parameter can be tuned as learning parameters. Finally, I represent the current input vector.

There are two thresholds related to the number of neurons to be generated (β) and the learning capacity (η). First threshold is used to generate more neurons in order to increase the number of rows and columns. In this case, a group of neurons is created where just one has the updated value and the others are filled with the nearest neighbor value. Different experiences have been followed in order to fulfill this requirement, where last is an approach.

III. INTEGRATION STRATEGY

The integration strategy for the implementation of sensor fusion is based upon a group of local Kalman filters which are estimating different local signals who have in common certain uncertainty due to common observed noise. In order to integrate common strategy and behavior of the system, the self organizing map is used as shown in Fig. 1.

This integration is pursued every sampling period since the self organizing map is already trained as shown in Fig. 2. Moreover, Figure 3 shows how this strategy pursues fault classification as presented in (Benítez-Pérez and Benítez-Pérez, 2009) based upon cluster classification where common noise is degraded as common characteristic not suitable for classification. In this paper, a common tradeoff between fault detection and sampling window is present. Several strategies have been pursued like OAN (Benítez-Pérez *et al*., 2007a) where fault diagnosis and classification is pur-

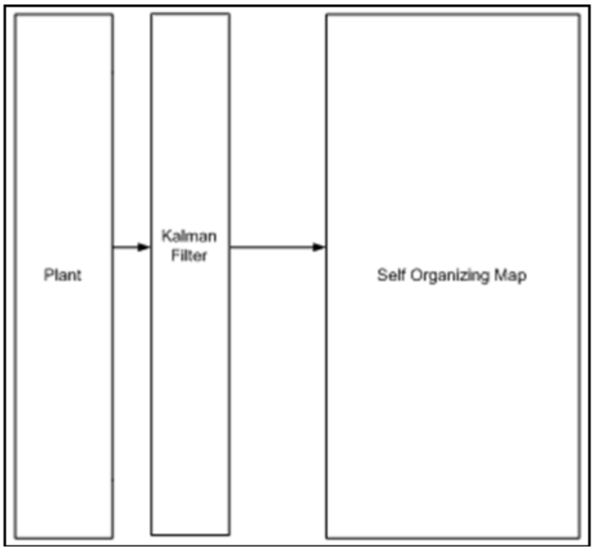

Figure 3. Sensor based upon SOM and Kalman filter.

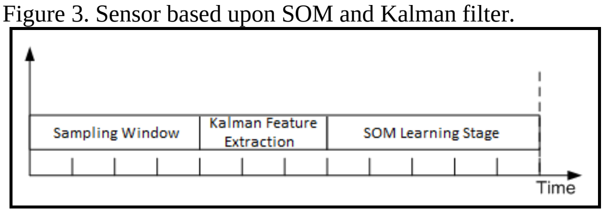

Figure 4. Off-Line Stage.

sued in terms of overlapping neural networks. However, periodic fault classification needs to hire fault in terms of exact initialization meaning early detection in terms of current fault localization. Since this extraction needs an accurate definition of fault characteristics it relays on a precise classification strategy for isolation purposes. Alternatively, fault detection based on local modeling is possible since most of fault conditions are presented locally at early stages.

Now from this perspective, an integration of these algorithms in order to isolate typical response from fault response through Kalman Filters and SOM network is pursued. Moreover, the use of an overlapping time window is pursued in order to determine bounded time varying response from a faulty scenario. This is shown in Fig. 3.

Three steps are defined during offline stage as depicted in Fig. 4 in terms of data processing. Firstly, a sampling window of input and output data is taken and normalized, thereafter, this information is processed by Kalman Feature Extraction to perform feature extraction, and the local matrix is classified by the SOM in learning mode. In this case, it is necessary to provide with enough information and a diversity of scenarios (fault and fault-free) the neural network in order to assure a suitable system identification.

During offline stage, the neural network does not perform any learning procedure. The comparison between already classified patterns against current scales vectors produced by Kalman filters for sensor fusion is performed obtaining the minimal value amongst all patterns is defined as winner and compared with β. If this value is smaller than β this is a correct winner otherwise neural network is not capable to perform this comparison. This value is called error for system evaluation. Similar to offline stage, it takes four major steps in order to produce a result as shown in Figure 5. In this case, SOM does evaluate the resultant local model generated by the group of Kalman filters feature extraction. The decision making module inherent to SOM determines if this classification is valid or not where its time consumption is neglected.

The online results are produced every *w* sampling time windows which is the time taken by the application in order to produce a result (Fig. 6). This time window *w* becomes crucial in terms of the frequency of the plant and this sampling window is a multiple of inherent period of case study. In fact, the plant response during fault and fault free scenarios need to bounded to this parameter in order to guarantee a reliable answer.

As presented in different studies like Chiu *et al.* (2012), Yang *et al.* (2011), Minatohara and Furukawa (2011) and Chen and Young (2005), in here, fault classification is pursued in terms of cluster classification where local differences are pursuit for single clusters in terms of the map as presented in Benítez-Pérez and Benítez-Pérez (2009).

IV. CASE STUDY

The linear variable differential transformer (LVDT) (also called just a differential transformer) is a type of electrical transformer used for measuring linear displacement (position). A counterpart to this device that is used for measuring rotary displacement is called a rotary variable differential transformer (RVDT).

LVDTs are robust, absolute linear position/ displacement transducers; inherently frictionless, they have a virtually infinite cycle life when properly used. As AC operated LVDTs do not contain any electronics, they can be designed to operate at cryogenic temperatures, in harsh environments, under high vibration and shock levels. LVDTs have been widely used in applications such as power turbines, hydraulics, automation, aircraft, satellites, nuclear reactors, and many others.

The LVDT converts a position or linear displacement from a mechanical reference (zero, or null position) into a proportional electrical signal containing phase (for direction) and amplitude (for distance) information. The LVDT operation does not require an

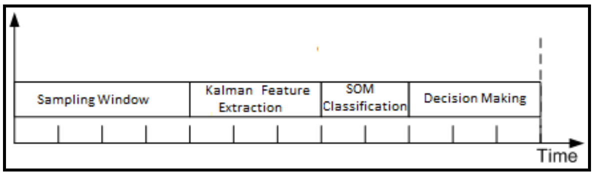

Figure 5. Online Stage.

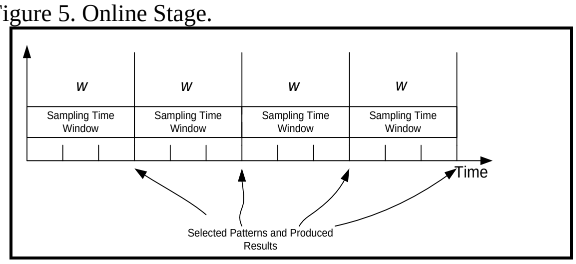

Figure 6. Online Produced Results.

electrical contact between the moving part (probe or core assembly) and the coil assembly, but instead relies on electromagnetic coupling.

The analysis and design of LVDT's is usually based on a semi-empirical approach. In particular, the electromagnetics of these devices tends to be over-simplified. New models and geometries are usually interpolated from existing designs. The impedance of the primary winding is then calculated using standard formulae and the voltage ratio is estimated.

Another important characteristic of a LVDT is the null voltage. Ideally, the output of an LVDT at null core position should be zero. However, quadrature voltages and harmonic components of the excitation source do not cancel out. Quadrature voltages result from the difference in winding capacitance between the primary and each secondary. The secondary outputs do not have exactly the same phase angle; therefore, they will not completely cancel out at any core position. LVDT characteristics are usually established experimentally. Simplified mathematical analysis of an LVDT assumes an idealized distribution of the leakage flux, which ignores any non-uniformity of the field strength along the axis of the coil and neglects end effects. For a sinusoidal primary excitation current (rms) of I at a frequency f, the differential output voltage may be expressed as:

$$\Delta v = k_1 x\left(I - \frac{x^2}{k_2} \right) \tag{3}$$

where Δv is the differential output voltage, k_1 is the sensitivity of the LVDT, x is the Core displacement, I is the excitation current and $k_2 =(I-x^2/k_2)$ is the coefficient of non-linearity.

The coefficient k_2 in Eq. (3) can be also evaluated from:

$$k_2 = \frac{1}{4}\left(l^2 - b^2\right) \tag{4}$$

where l is the core length and b is the axial separa-tion between secondary windings. The sensitivity of the LVDT is given by:

$$k_1 = \frac{4\pi\mu_0 f l N_p N_s k_2}{dc \ln(r_2 / r_1)} \tag{5}$$

where N_p and N_s are the number of turns on the primary and secondary, respectively, c is the length of the core, d is the length of the secondary, r_1 is the core radius and r_2 is the inner radius of the outer case.

V. CURRENT CONFIGURATION

The system to implement sensor fusion is basically composed of the following elements:

A. Acquisition Card Humusoft MF624.

To control the direction of stepper motor are used digital outputs. It obtained LVDT sensor signal through an analog to digital converter.

B. Stepper Motor Haydon Switch and Instrument.

Converts a series of electrical impulses in discrete angular displacements of 7.5 °, the first rod to move in and then out of the LVDT, the motor has four wires that control their movement shown in Figure 7 motor wiring diagram.

The rotary motion of a stepper motor can be converted into linear motion by several mechanical means. These include rack and pinion, belt and pulleys and other mechanical linkages. All of these options require various external mechanical components. The most effective way to accomplish this conversion is within the motor itself. Conversion of rotary to linear motion inside a linear actuator is accomplished through a threaded nut and leadscrew. In order to generate linear motion the lead screw must be prevented from rotating. As the rotor turns the internal threads engage the lead screw resulting in linear motion. Changing the direction of rotation reverses the direction of linear motion.

The linear travel per step of the motor is determined by the motor's rotary step angle and the thread pitch of the rotor nut and leadscrew combination. Coarse thread pitches give larger travel per step than fine pitch screws. However, for a given step rate, fine pitch screws deliver greater thrust. Fine pitch screws usually can not be manually "backdriven" or translated when the motor is unenergized, whereas many coarse screws can. A small amount of clearance must exist between the rotor and screw threads to provide freedom of movement for efficient operation. If extreme positioning accuracy is required, axial play (also called backlash) can be compensated for by always approaching the final position from the same direction. Accomplishing the conversion of rotary to linear motion inside the rotor greatly simplifies the process of delivering linear motion for many applications. Because the linear actuator is self contained, the requirements for external components such as belts and

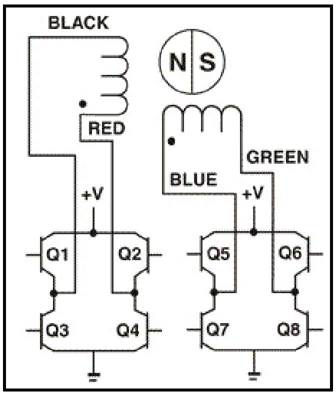

Figure 7. Stepper motor wiring diagram.

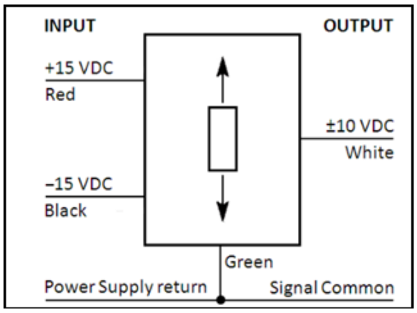

Figure 8. Block LVDT.

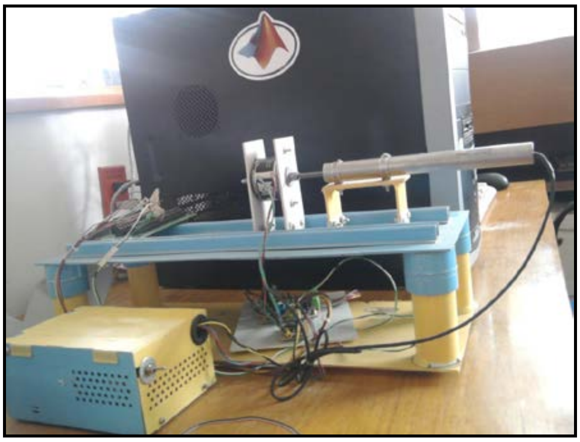

Figure 9. Current configuration of the system. This image shows the LVDT DC-EC 1000 sensor, the motor (moves the rod through the LVDT), power supply, the Humusoft MF624 data acquisition interface and the Pentium D PC with Matlab.

pulleys are greatly reduced or eliminated. Fewer components make the design process easier, re-duce overall system cost and size and improve product reliability.

C. LVDT Hampton DC-EC-1000.

This device (Measurement Specialities, 2013a, b) allows to measure the displacement of the rod, handles ± 10 VDC output and a sensitivity of 0.39 VDC / mm. Shown in Fig. 8 LVDT block.

D. System Configuration

PC with Matlab and Simulink. All control is programmed using Matlab Simulink and the interface between the program and the real model is the Humusoft card. Figure 9 shows current system configuration as in real state.

Consider the state-space model of a linear discrete time dynamical system according to:

$$x(k+1) = Ax(k) + Bu(k) + r \quad (6)$$

$$y(k) = Cx(k) + Du(k) + v \quad (7)$$

where x is the state vector - displacement measurement, u is the known input, y is the output, r are the random variables-supply noise and v are the random variables - spring - measurement error.

The rotary torque of the stepper-motor is transferred through a worm gear, which generates back-and-forth straight-line motion. The stepper-motor runs with constant stepping rate. In order to determine stepper motor movements, the linear variable differential transformer (LVDT) is used (Mohinder and Angus, 2008). The position of the worm gear is the resulting signal. These measures fed into computer through a 14-bit data acquisition (DAQ) MF 624 card. In the data acquisition stage the raw data (in volts) from the sensor are acquired via the DAQ.

LVDT is very sensitive, any insignificant vibrations from the test platform or environment cause disturbances for measurement. Hence, resulting noisy experimental data is not satisfactory. Therefore, a Kalman filter was proposed to solve this measurement noise problem. In order to compensate the output nonlinearity a self- organizing map (SOM) neural network was used.

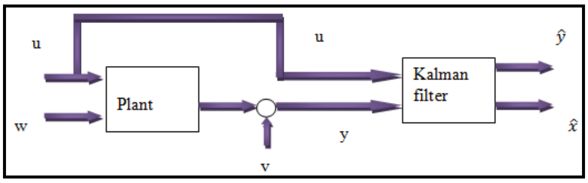

Figure 10. Kalman filter chart.

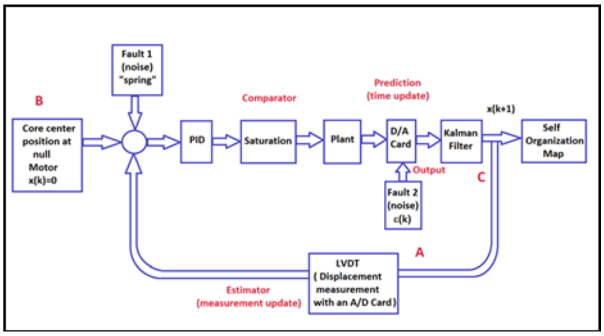

Figure 11. System configuration.

Moreover the use of these networks allows to isolate certain fault scenarios through a properly tuned Kalman Filter.

The use of Kalman Filters for the implementation of sensor fusion presents a powerful approximation for early fault detection using model based technique. Moreover, the use of Self Organizing Maps allows fault isolation at early stages with a very accurate performance. Since local model based strategies are well tuned and stated (Simon, 2001).

As we can see in the Fig. 10, the estimator uses the known inputs u and the measurements y to generate the output and state estimates $\hat{x}$ and $\hat{y}$. Note that $\hat{y}$ estimates the true plant output. To see the filter response, we use the data of LVDT getting in and out.

System:

$$x(k+1) = Ax(k) + Bu(k) + r(k) \quad (8)$$

$$y(k) = Cx(k) + v(k) \quad (9)$$

$$\hat{x}(k+1) = A\hat{x}(k) + Hu(k) + K(y(k) - C\hat{x}(k)) \quad (10)$$

where $\hat{x}(k)$ is the estimated state, A is the state transition matrix, H is the measurement matrix, $\hat{P}$ is the estimated error covariance, Q is the process noise covariance and R is the measurement noise covariance.

Previous state: The priori estimate error covariance is then formulated as:

$$P_k^- = Q + AP_{k-1}A^* \quad (11)$$

Figure 11 shows our case study. The direction of step motor is defined by a sawtooth signal modulated by a square waveform. The excitation signal of LVDT is a sine waveform. The fault 1 was implemented with a spring. The distance of the test is monitored and controlled using a computer by a DAQ card. It provides position error correction for the Kalman Filter. In Simulink this job is done with an Analog/Digital input block. The saturation limits to the actuator are defined in a high value (25 V), because the actuator never reach the saturation limit. In addition to the actuator actual position, the actuator condition data set involves: (a) Estimates of the actuator input–output relationship and (b)

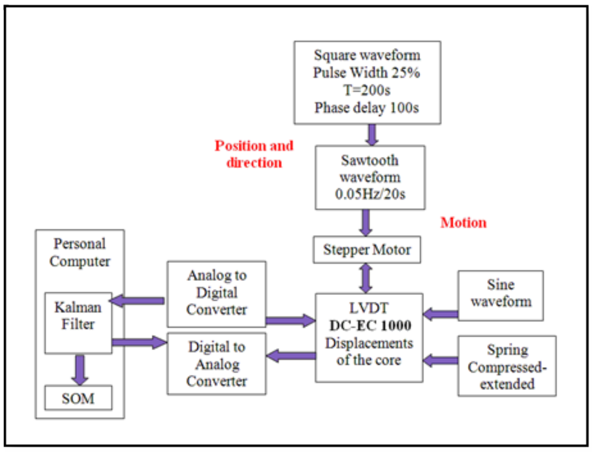

Figure 12. Schematic diagram of dynamic system.

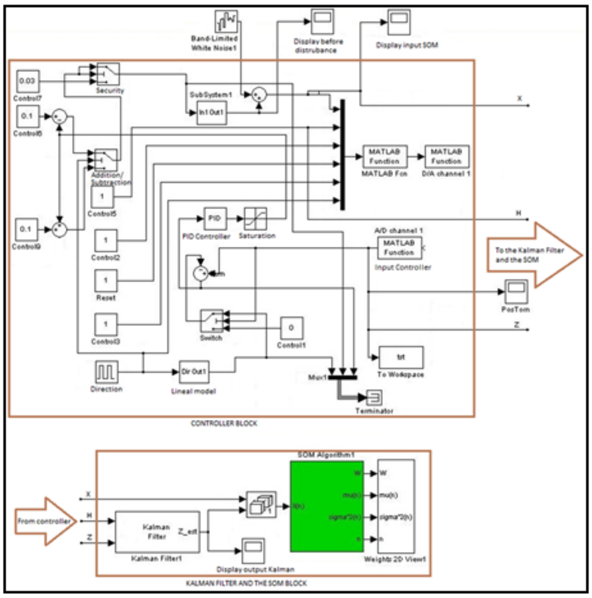

Figure 13. Simulink blocks for displacement measure-ments Kalman filter algorithm.

Estimates of noninvertible nonlinearities. To avoid deg-radation sensor data Proportional-Integral-Derivati-ve (PID) controller is required due to changes in the plant and/or the actuator characteristics. The controller can accommodate undesirable actuator characteristics to ob-tain desired values. In order to compensate the pro-cess input or output nonlinearity a Kalman Filter and self-organizing map (SOM) neural network were used.

The Fig. 12 shows the general system control chart. The schematic diagram shows the pulse train modulated which is used to excite the motor and the output is the LVDT's core displacement. A channel of acquisition data card is used to receive the analog signal from LVDT and converting it in a digital signal. Afterwards the signals are computer processed under Matlab and lineal displacement is registered.

The experimental setup for distance measurement with LVDT DC-EC 1000 sensor is illustrated in Figure 9. The maximum range of LVDT is 10 mm. The ac-quired, conditioning, processing data and graphical in-terface are all programmed in Matlab/Simulink envi-ronment.

The parameters of the Kalman filter are as follows:

$$A=\begin{bmatrix} -2.083e-6 & 0.0001601 & 3.827e-5 & -5.005e-5 & 5.556e-5 & 0.0003541 \\ 5.3e-5 & -0.006038 & -0.02909 & 0.02879 & 0.01404 & -0.02367 \\ -4.284e-6 & 0.03194 & 0.005456 & -0.06846 & 0.03746 & -0.1602 \\ -2.949e-6 & -0.01551 & 0.1006 & 0.0136 & 0.128 & -0.1121 \\ -3.244e-6 & 0.005334 & -0.04153 & -0.1372 & 0.00679 & 0.08591 \\ 4.28e-5 & 0.001896 & 0.1546 & 0.1338 & -0.02563 & -0.04105 \end{bmatrix}$$

$$B=\begin{bmatrix} 1.01e-9 \\ -7.25e-8 \\ -3.664e-7 \\ -8.964e-7 \\ 6.218e-7 \\ -7.825e-7 \end{bmatrix} \qquad C^{*}=\begin{bmatrix} 1692 \\ -13.7 \\ -15.88 \\ -3.481 \\ 1.336 \\ -10.23 \end{bmatrix}$$

$$\hat{P}=12 \qquad Q=0.005 \qquad R=1$$

$$K=\begin{bmatrix} 8.283e-6 \\ 0.0001636 \\ -3.483e-5 \\ 7.036e-5 \\ 5.863e-5 \\ 9.141e-5 \end{bmatrix}$$

The Kalman filter for sensor fusion estimates a pro-cess by using a form of feedback control: The filter es-timates the process state at some time and then obtains feedback in the form of noisy measurements. As such, the equations for the Kalman filter fall into two groups: time update equations (prediction) and measurement update equations (estimator). The time update equations are responsible for projecting forward (in time) the cur-rent state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measure-ment update equations are responsible for the feedback, for incorporating a new measurement into the *a priori* estimate to obtain an improved the *a posteriori* estimate (Tong *et al.*, 2008).

VI. RESULTS

In deriving the algorithm for the Kalman filter, one should begin with the goal of finding an equation that computes an a posteriori state estimate as a linear com-bination of an a priori estimate and a feedback from er-ror between an actual measurement and a measurement prediction with a weighted gain K. In order to find a suitable gain K that minimizes the a posteriori error co-variance is given by

$$K=P_k^{-}C^{*}\left(CP_k^{-}C^{*}+R\right)^{-1} \tag{12}$$

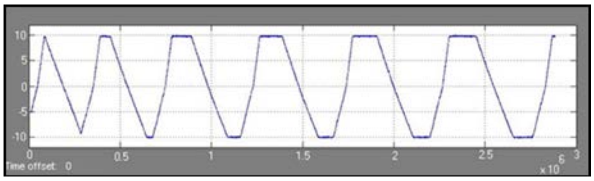

Figure 14. Output signal of the Kalman filter without fault on the plant.

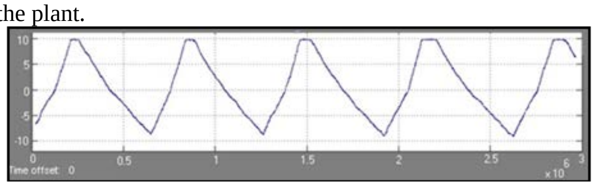

Figure 15. Output signal of the Kalman filter with a fault scenario caused by the spring disturbance.

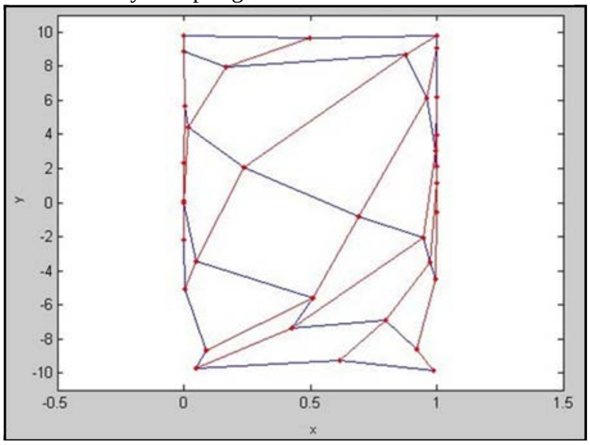

Figure 16. Graph of the SOM without fault on the plant.

Figure 14 shows the signal of Kalman filter without fault on the plant. In this case, there is no external excitation to prevent the free movement of the rod.

Figure 15 shows the signal of Kalman filter with a fault scenario. The fault was caused by a spring positioned at the end of the rod, which produced a movement effort.

Figure 16 shows the graph of the SOM without fault on the plant. The plant is not affected by any external agent. This scenario is related to system response from Fig. 14.

Figure 17 shows the graph of the SOM with a fault scenario. There is a shift of the upper points of the neural network to the results obtained in normal state Fig. 16, the movement of these selected grid elements are to the left and to the right side. In this case, it is ob-served that the SOM detects a change in the input vectors. This is associated to the influence of the spring on the rod movement, therefore, SOM network generates new patterns to a different input signal.

The Figure 18 shows the output of the plant as the input of the network and as result of the Kalman Filter considering both scenarios.

From these two considered scenarios; Figure 19 presents the results related to fault and fault-free scenarios. The period of the signal is 70 s, when the fault caused by the spring disturbance appears the sensor time delay response is 3 s, in the figure 19 shows the 280 seconds (4 cycles), which is time system recovering process

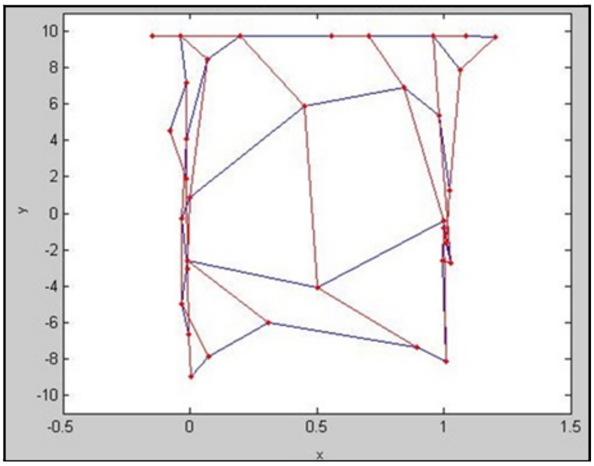

Figure 17. Graph of the SOM with a fault scenario ca-used by the spring disturbance.

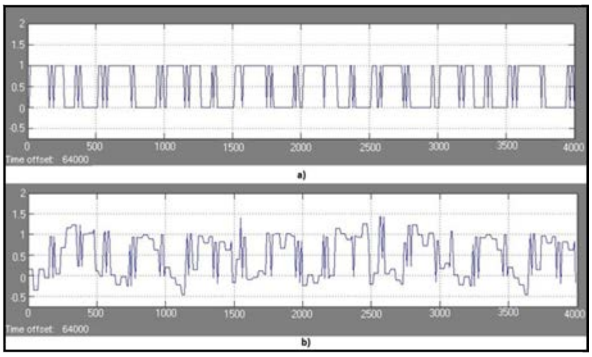

Figure 18. SOM input without fault a) SOM input with fault caused by the spring disturbance b). This signal is the output of the plant and this is one of the input signals of the SOM.

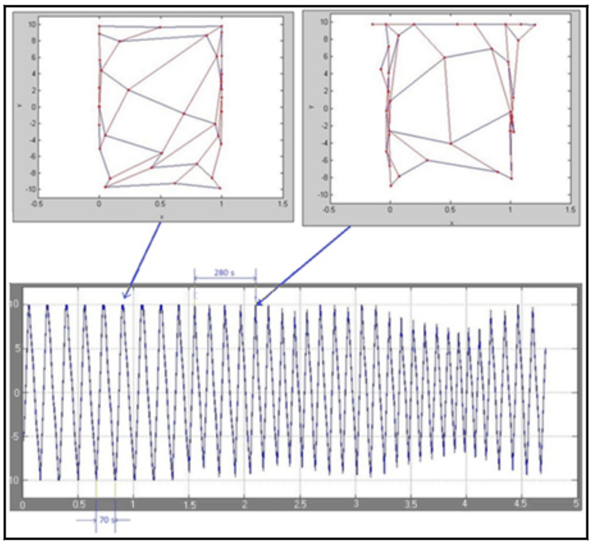

Figure 19. Fault-free and fault scenarios.

takes, this time delay gives an approximation of how long it takes to obtain a trustable response.

Each node is initialized with an activation weight, and each link between the nodes is initialized with a weight (Fig. 20). Weights are randomly initialized.

The two inputs observed as a color map are shown in Fig. 21 correspond to (a) original and (b) with noise. The algorithm works by grabbing one of these input vectors (left). It compares with (right) to every node inside the SOM. The distance in two dimensions space between two patters is the distance refereed as Mahalanobis distance (variance –covariance matrix). Once the best matching node is found, the node changes all of its weights to that of the input vector. The node then updates the weights of its surrounding nodes based on a formula involving the learning rate, the hexagonal layer topology function of dimensions 5x7, and the distance of the cell.

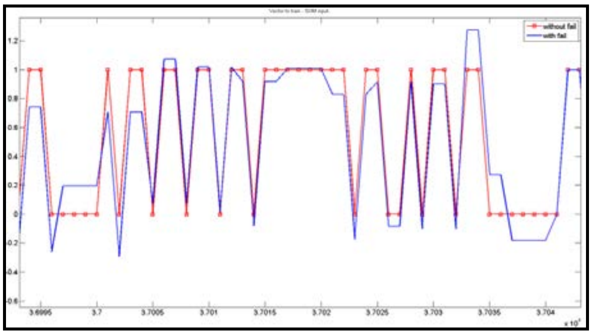

Figure 20. Vector to train, SOM input. Signal without fault (dot line) and signal with fault (solid line).

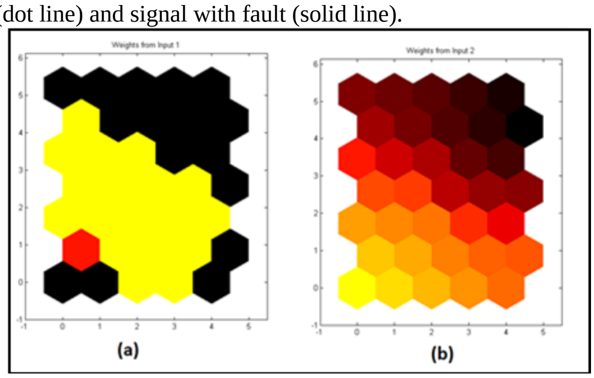

Figure 21. SOM observed as a different areas in the map. Fault-free a) and fault b) scenarios.

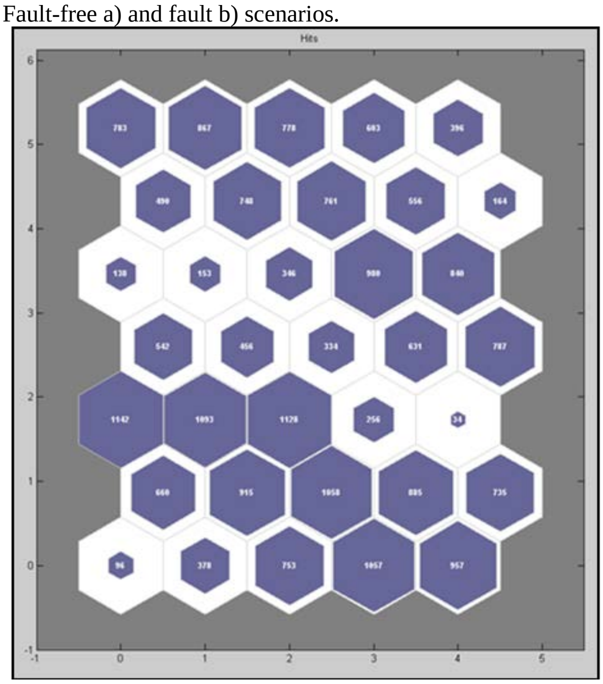

Figure 22. Number of occurrences of each weight in the iteration 200.

Since SOM networks can learn to detect regularities and correlations in their input and adapt their future responses to that input accordingly (Knudsen *et al.*, 1987), this valuable response has been used to isolate fault scenarios. Now, Self-organizing maps learn to re-cognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors.

In Fig. 22 it is presented the number of occurrences of each weight, as we can see majority of nodes are around (0,2) and around (2,2). It means, how many data points area associated with each neuron. It shows where they are concentrated. In this case are concentrated a little more to middle-down-left neurons. It can be also visualized in the Fig. 22, where lighter and darker colors represent larger and smaller weights, respectively.

The main goal of the SOM is to transform an incoming pattern of arbitrary dimension into a two- dimensional discrete map, and to perform this transformation adaptively in a topologically ordered to acquire signals in parallel.

In this study, a SOM toolbox (version 2) for Matlab 5.0 developed by the Laboratory of Information and Computer Science at the Helsinki University of Technology was used (Vesanto *et al*., 1999).

In order to setting initial parameter of the SOM. We assume that the input–output data have been scaled and transformed to reasonable ranges. We also assume that weight values are initialized to small random values. The approximate error covariance matrix is initialized to reflect the fact that no a priori knowledge was used to initialize the weights.

In our case the grid topology is hexagonal and the grid size is [5 x 7]. This one have 2 neurons as input. These neurons were placed in a two-dimensional lattice structure. This network represents a feedforward structure with a single computational layer consisting of neurons arranged in a 2D grid. The variables [x_{min}, x_{max}, y_{min} , y_{max}] = [-0.5,1.5,-11,11] are only used to render the network to the display area - each node is represented by a rectangular cell described by these values.

Consequently, the neurons transform input signals into a place-coded probability distribution that represents the computed values of parameters by sites of maximum relative activity within the map (Knudsen *et al.*, 1987). In this work the initial value of standard deviation is σ_0=5, which define the topological neighborhood function. 200 iterations are chosen for the algorithm to learn as maximum, understanding that a computational cost is given to process information. Then training data are characterized by noisy measurements, therefore these required small values for the learning rate. Then initial value of the learning-rate parameter was setting 0.25.

The total weight associated with a postsynaptic (two dimensional array of neurons around winning) neuron is bounded. As a result some incoming connections are increased while others decrease. This is needed in order to achieve stability of the network due to ever-increasing values of synaptic weights. Topological neighborhood function decrease with a time constant of 1000 [ms]/log (10). The learning-rate parameter decrease with a time constant of several seconds. Each output neuron is fully connected to all the source nodes in the input layer.

From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones see Figure 16. Each zone is effectively a feature classifier; it means the graphical output as a type of feature map of the input space. As is showed in the trained network shown in Fig.17, the sides represent similar weight vectors.

VII. CONCLUSIONS

As we can see combining the SOM with Kalman filter can smooth and accelerate the learning and convergence of the SOM.

A good training speed, mapping accuracy and rea-son-nable performance is achieved using Kalman filter applied to the training and use a learning algorithm (SOM) for the implementation of sensor fusion and to determined a LVDT position and displacements.

Due to our inputs are homogeneous, it means that does not exist regions of rapid variation of inputs and outputs may be followed by regions of slow change, the SOM produce good results.

Experiments with real data show that a Kalman Filter plus SOM algorithm, provide an efficient way for spatial grouping of data, adaptively adjust the positions of neurons on the map according to the corresponding distances in the data space. This combination is also robust to parameter selection. Due to be based on competitive learning form of unsupervised learning in neural networks, where neurons compete for responding to inputs. When an input is presented, neurons compete with each other to see which neuron is most similar to it. Different neurons specialize in different regions of the input space. Which allow obtain a good visualization of the results.

The couple Kalman Filter and SOM is a standard tool in the control (analysis, estimation and prediction of processes).

In addition, the SOM model was successfully applied to isolate a fault.

ACKNOWLEDGMENTS

The authors acknowledge the support of UNAM-PAPIIT IN100813 and CONACYT num. 176556

REFERENCES

Benítez-Pérez, H., F. Cardenas-Flores, J. Ortega-Arjona and F. Garcia Nocetti, "Fault Localization Based Upon Non-Supervised Neural Networks and Unknown Input Observers for a Peripheral Element," *International Journal Control and Intelligent Systems*, **35**, 319-238 (2007a).

Benítez-Pérez, H., F. García-Nocetti and H. Thompson, "Fault Classification Based upon Self Organizing Feature Maps and Dynamic Principal Component Analysis for Inertial Sensor Drift," *International Journal of Innovative Computing, Information and Control*, **3**, 257-276 (2007b).

Benítez-Pérez, H. and A. Benítez-Pérez, "The use of ARMAX strategy and Self Organizing Maps for Feature Extraction and Classification for Fault Diagnosis," *International Journal of Innovative Computing, Information and Control, IJICIC*, **5**, 4787-4796 (2009).

Benítez-Pérez, H., J.L. Ortega-Arjona and A. Benítez-Perez, "Using Wavelets for Feature Extraction and Self Organizing Maps for Fault Diagnosis of Nonlinear Dynamic System," *Intech*, **3**, Maps. http://dx.doi.org/10.5772/50235 (2012).

Chen, Y.-Y. and K.-Y. Young, "Applying SOM as a Search Mechanism for Dynamic System," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain (2005).

Chiu, S.-H., C.-Y. Wen, J.-H. Lee, K.-H. Lin and H.-M. Chen, "A fast randomized generalized hough transform for arbitrary shape detection," *International Journal of Innovative Computing, Information and Control*, **8**, 1349-4198 (2012).

Hassoum, M.H., *Fundamentals of Artificial Neural Networks*, MIT Press, USA (1995).

Knudsen, E.I., S. du-Lac and S.D. Esterly. "Computational maps in the brain," *Annual Review of Neuroscience*, **10**, 41-65 (1987).

Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, Germany (1989).

Minatohara, T. and T. Furukawa, "The Self-Organizing Adaptive Controller," *International Journal of Innovative Computing, Information and Control*, **7** (2011).

Mohinder, S.G. and A.P. Angus, *Kalman Filtering: Theory and Practice using Matlab*, Wiley, California, U.S.A. (2008).

Measurement Specialities, *DC-EC Series – Gene-ral purpose DC LVDT*, In URL: http://www.meas-spec.com/downloads/DC_EC_Series.pdf (2013a).

Measurement Specialities, *The LVDT: Construc- tion and principle of operation. Technical Paper*, In URL: http://meas-spec.com/downloads/Principles_of_the_LVDT.pdf (2013b).

Nelles, O., *Non-Linear Systems Identification*, Springer-Verlag, Berlin, Germany (2001).

Simon, H., *Kalman Filtering and Neural Networks*, John Wiley & Sons (2001).

Tong, C.-C.. C.-J. Chao, T.-Y. Lin, W.-N. Lin and C.-T. Yang, "A Kalman Filter Approach to LVDT-based Heat Deformation Test," *IEEE International Conference on Systems, Man and Cybernetics*, 1861-1865 (2008).

Vesanto, J., J. Himberg, E. Alhoniemi and J. Parhankangas, "Self-organizing map in Matlab: the SOM Toolbox, *In Proceedings of the Matlab DSP Conference*, Espoo, Finland, 35–40 (1999).

Yang, J., Z. Pei, L. Zou and K. Qin, "Feature Extraction of the Radio Signals using Attribute Reduction of Formal Concepts," *International Journal of Innovative Computing, Information and Control*, **7**, 1349-4198 (2011).

Received: February 5, 2014.
Accepted: March 20, 2015.
Recommended by Subject Editor: Jorge Solsona.