

Vibration Analysis Via Neural Network Inverse Models To Determine Aircraft Engine Unbalance Condition

Xiao Hu¹, John Vian², Joseph R. Slepksi², Donald C. Wunsch II¹

¹Applied Computational Intelligence Laboratory, Dept. of ECE, U. of Missouri-Rolla, Rolla, MO, 65409

²Boeing Phantom Works, P.O.Box 3707, MC 45-85, Seattle, WA, 98124

Abstract-This paper describes the use of artificial neural networks (ANN's) with the vibration data from real flight tests for detecting engine health condition – mass imbalance herein. Order-tracking data, calculated from time series is used as the input to the neural networks to determine the amount and location of mass imbalance on aircraft engines. Several neural networks methods, including multilayer perceptron (MLP), extended Kalman filter (EKF) and support vector machines (SVMs) are used in the neural networks inverse model for the performance comparison. The promising performances are presented at the end.

I. INTRODUCTION

The use of machine condition monitoring can provide considerable cost savings in many industrial applications especially where large rotating machines are involved, for example generators in power stations [16]. The monitoring of vibrations of these machines has been reported as being a useful technique for analysis for their conditions [15][17]. However, advanced techniques that have effective approximation properties, such as neural networks have been only recently introduced into this field, especially on the health monitoring of aircraft engines.

The focus of this paper relates to the research efforts to employ machine vibration analysis on the sensor data from airplane engine in flight to determine the feasibility of artificial neural networks to diagnose machine health conditions [3][15]. Given engine vibration data, the neural network inverse model that attempts to predict aircraft propulsion systems rotor imbalance is studied as part of an overall Integrated Vehicle Health Management (IVHM) system. Initial goals of the research are to establish propulsion system inverse models, understand vibration data characteristics, and develop algorithms that can analyze sensor data for monitoring and diagnosis of IVHM subsystem. Fig. 1 illustrates an approach for the inverse problem. In [3], we used data generated from engine structural finite-element model and achieved exceptional performance from neural networks on diagnosis of mass imbalance on engine. This paper presents results using actual flight test data. For the given speed of operation an intentionally induced mass imbalance produces vibration in the engine case and mounting structure that contains sensors

at various locations. These sensor outputs are then transformed to generate a steady state frequency response, which is called order-tracking data [4]. Using the frequency information, the neural network inverse model will be trained to predict the mass imbalance on the engine.

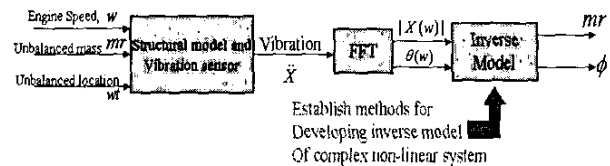


Figure 1. The inverse model of engine health condition diagnosis.

II. DATA ACQUISITION AND PREPROCESSING

To design an effective machine condition monitoring system, data must be acquired for all the conditions that need to be classified [16]. It is necessary to have data for use in the design stage of the condition monitoring system (this is the training data for the artificial neural networks) as well as independent test data that has not been used in the design stage to validate the system. The data used in this paper is acquired from multiple test flights. The tests included ground and flight tests with various combinations of imbalances to the engine fan and low-pressure turbine. The results herein are based on using only the in-flight data. The test matrix is shown as follows in Table 1. Besides the baseline cases, there are at total of 5 different artificially induced mass imbalances on the engine, three on the low pressure turbine (LPT) and two on the fan. Engine 1 and Engine 2 were assumed to produce measurements having the same features, which represent the state of engine unbalance. In each test, the engines run at 12 different fixed speeds. There are three accelerometers (synonyms for sensors, channels) located on the engine case and mounting structure that are used to collect the vibration signals from the engines. The sensor data is in the form of time series. General locations of induced mass imbalance and sensor locations are illustrated on the schematic of a representative jet engine in figure 2.

Test #	Engine 1 Balance	Engine 2 Balance
T1	Baseline	Baseline
T2	Unbalanced LPT 14.5 oz-in @ 9.1 deg	Unbalanced Fan 14.9 oz-in @ 240 deg
T3	Unbalanced Fan 20 oz-in @ 170 deg	Unbalanced LPT 23.1 oz-in @ 90 deg
T4	Unbalanced LPT 28.7 oz-in @ 229.4 deg	Unbalanced LPT 28.7 oz-in @ 170.4 deg
T5	Baseline	Baseline
T6	Balanced w/ Generic IC's 8.0 @ 90 - 14.5 @ 159	Fan Balanced w/ GenIC's Lgt installed 50 deg error 6.1 @ 146 - 20.3 @ 169**
T7	Same as T6	Balanced w/ Generic IC's 6.1 @ 146 - 20.3 @ 218

Table 1. Test Matrix: T6-Engine1 and T7-Engine2 are for balanced engines, which are taken as ideal non-imbalance case. T1 is the baseline with mass imbalance. T2-T4 are tests with the intended known mass imbalance on LPT or Fan. The numbers shown in table denote the amount of the imbalance and its angular location. T5 is the repeated test as T1. Engine1 and Engine2 are assumed to be symmetric

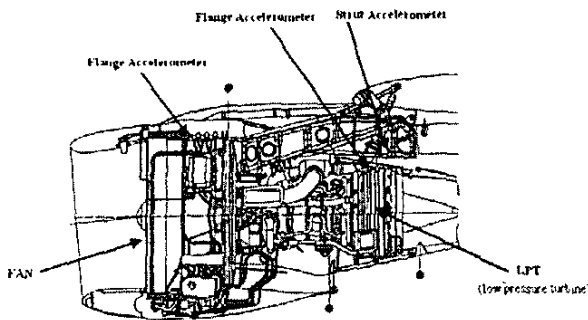


Figure 2. General locations of mass imbalance and sensor locations illustrated on typical fan-jet engine. Mass imbalances are added on Fan and LPT. Sensors are located on flanges and strut.

The available dataset is order-tracking tables, which are generated from the processing on the time domain data collected from sensors. The objective of order tracking is to measure the level of vibration due to a rotating shaft or other harmonic signals. The term means to track an order of a harmonic signal. Order-tracking is performed as two separate measurements: 1) a frequency counter measurements the frequency of the tachometer signal, 2) the magnitude, phase and variance of each response channel is measured by re-sampling the signal, synchronous averaging, and Fourier transform. It can handle speed (frequency) variations, and can distinguish between rotor-related signals and noise from other sources (not rotor-related). It is similar to the dataset we used in [3], which was generated from an engine structural model in the frequency domain, including the magnitude and phase information of the vibration data from sensors. Figure 3 illustrates typical characteristics of magnitude and the phase of measurements from the accelerometer located on a forward flange for the mass imbalance added on the LPT in all the flight tests. For each

sensor, we end up with 12 stable frequency pins (points) or each test, corresponding to the 12 speeds running in the each test.

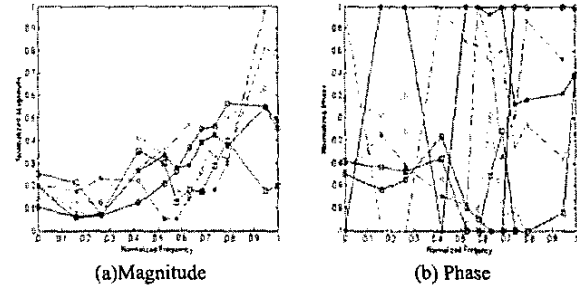


Figure 3. (a) Magnitude of the measurement from forward flange for the mass imbalance on LPT in the form of order track data for all the available flight tests. (b) Phase of the same measurement.

From flight tests data, we have 4 different mass imbalances (0 oz-in, 14.5 oz-in, 23.1 oz-in and 28.7 oz-in) on the LPT of the engine and 3 different mass imbalances (0 oz-in, 14.9 oz-in and 20 oz-in) on the fan of the engine, including the balanced engine from T6 and T7. Since we have the assumption that the state of engine in T6 and T7 as the ideal non-imbalance case. Then we can recalculate the actual mass imbalance at original order-tracking data based on the ideal non-imbalance situation T6 and T7 using vector addition, as shown in Table 2. Enough data points corresponding to different region of the target space are needed for the neural network inverse model to learn nonlinear mapping between order-tracking data and target mass imbalance.

Test #	Engine 1 Balance	Engine 2 Balance
T1	8.0 oz-in@270deg on Fan 14.5 oz-in@339 deg on LPT	6.1 oz-in@326deg on Fan 20.3 oz-in@38 deg on LPT
T2	8.0 oz-in@270deg on Fan 28 oz-in@354 deg on LPT	16.5 oz-in@261.7 deg on Fan 20.3 oz-in@38 deg on LPT
T3	20.2 oz-in@193 deg on Fan 14.5 oz-in@339 deg on LPT	6.1 oz-in@326 deg on Fan 39 oz-in@65.8 deg on LPT
T4	8.0 oz-in@270deg on Fan 27.5 oz-in@259 deg on LPT	6.1 oz-in@326 deg on Fan 21.2 oz-in@125.4 deg on LPT
T5	8.0 oz-in@270deg on Fan 14.5 oz-in@339 deg on LPT	6.1 oz-in@326 deg on Fan 20.3 oz-in@38 deg on LPT
T6	Balanced on Fan and LPT	0 oz-in@0 deg on Fan 16.8 oz-in@103.5 deg on LPT
T7	Same as T6	Balanced on Fan and LPT

Table 2. Converted Test Matrix: T6-Engine 1 and T7-Engine 2 are for balanced engine, which are taken as ideal non-imbalance case. The numbers shown in table denote the amount of the imbalance and its angular location. Engine1 and Engine 2 are assumed to be symmetric.

Since Engine 1 and Engine 2 are assumed symmetric in location and the same kind of engine, we treat the measurements from Engine 1 and Engine 2 equally by assuming they have the same representative features of the vibration engine. For each order-tracking table, there are two

target mass imbalances associated with it, on LPT and FAN respectively, then we end up with applying surface fitting on the original data points in 3-dimension. Theoretically, we know from the engine vibration model, the mapping between the vibration data and the mass imbalance target should be a smooth nonlinear function. So we use a cubic interpolation method herein to approximate such a smooth nonlinear function. The accuracy of the surface to the real function is limited by number of flight test points. Better confidence in the accuracy of this approximation is easily achievable given more mass imbalance test conditions. Figure 4 gives the cubic interpolation between these points at a fixed frequency in the order-tracking tables. We end up with more data points for the need of the training and testing of the neural networks inverse model.

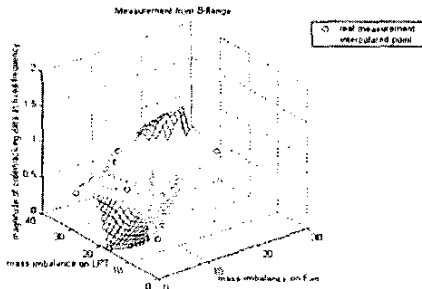


Figure 4. Cubic interpolation between the magnitudes of the original order-tracking points measured from forward flange at a certain frequency bins. X-axis is the mass imbalance on fan, Y-axis is the mass imbalance on LPT, Z-axis is the magnitudes of the original data and interpolated data points.

III. NEURAL NETWORKS INVERSE MODEL METHODS AND EXPERIMENT RESULTS

Artificial neural networks have been well known for their good function approximation property and extraordinary ability on pattern recognition. In this application, for the purpose of performance comparison, we employ several representative neural networks: multilayer perceptron (MLP) using backpropagation, extended Kalman filter (EKF), and support vector machines (SVMs). In this section we will discuss the details of individual network architecture as well as experiment results.

(1) Multilayer perceptron using backpropagation:

Let's start with the well-known back propagation (BP) algorithm to train multilayer feed-forward neural networks (also known as multilayer perceptron – MLP) with differentiable transfer functions to perform function approximation [1][19]. It is the most common method used in the training of neural networks. The results in this report were generated by using *MATLAB 6.1 Neural Networks Toolbox* and running on a 733 MHz PIII processor.

Experiment Results:

We have enriched the dataset by applying the cubic surface interpolation between the magnitudes of order-tracking data at each frequency bin for the converted test matrix (as Table 2). Performance is measured using a three-way cross-validation experiment. The data set is divided into three parts, 50% for training, 25% for validation, and 25% for testing. In Table 3 below, we present network training/testing results using standard backpropagation and its variants [5][19]. There are 8 neurons in the hidden layer and 2 neurons in the output layer because the corresponding target of input data is the mass imbalance on LPT and Fan. In this way, the neural network diagnoses the amount and the location of the mass imbalance at the same time. Levenberg-Marquardt training algorithm (TRAINLM) seems outperforms others in the context of mean square error (MSE) performance on test set and real flight data set. The results in the Table 3 are the average performance of the neural networks inverse model based on 5 trials. Magnitude 1 and magnitude 2 denoted in Fig 5 refer to the magnitude of the interpolated or real measurement at two fixed frequency bins of the order-tracking data respectively.

Training algorithm	Epochs	Time consumed (s)	MSE (on training set)	MSE (on testing set)	MSE (on real flight test data)
TRAINRP	633	10.1	0.3293	0.3647	1.4750
TRAINCGF	126	4.65	5.8337	7.2348	28.1301
TRAINCGP	273	8.9	2.9381	0.6845	3.8564
TRAINCGB	335	13.3	0.0321	0.0313	0.3580
TRAINSOG	227	6.8	1.0410	0.6254	3.6575
TRAINBFG	266	11.1	7.51e-04	6.82e-04	0.0051
TRAINOSS	75	3.65	11.0275	2.2150	10.8500
TRAINLM	218	21.38	9.93e-06	1.42e-05	3.25e-04

Table 3. Performance for the interpolated measurements and real flight test measurements from Strut for the mass imbalance on LPT. All the training parameters except the training algorithm are the same for each evaluation. Two layers, 8-2. The activation functions of hidden layer and output layer are 'tansig' and 'purelin', respectively. Target of the mean square error for training set is 1e-05. Shaded line means the training didn't reach the MSE goal and stopped early due to validation.

(TRAINRP: Resilient Backpropagation; TRAINCGF: Conjugate Gradient-Fletcher-Reeves Updates; TRAINCGP: Polak-Ribiere Updates; TRAINCGB: Powell-Beale Updates; TRAINSOG: Scaled Conjugate Gradient; TRAINBFG: Quasi-Newton-BFGS Algorithm; TRAINOSS: Quasi-Newton-One Step Secant Algorithm; TRAINLM: Levenberg-Marquardt.)

(2) Extended Kalman Algorithm:

The extended Kalman filter (EKF) algorithm has become a standard technique used in a number of nonlinear estimation and machine learning applications. Parameter estimation for nonlinear system identification (e.g. learning the weights of a neural network) is one of them. Puskorius and Feldkamp [6] and others have posed the weight estimation in a state-space framework to allow

for efficient Kalman training of a neural network. We implemented Node Decoupled Extended

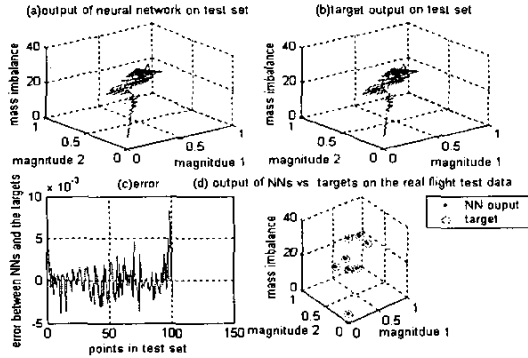


Figure 5. Performance of the neural network using Levenberg-Marquardt training rule on the interpolated measurements and real flight test data from Strut for the mass imbalance on LPT. Magnitude 1 and magnitude 2 refer to the magnitude of the interpolated or real measurement at the two fixed frequency bins of the order-tracking data respectively. (a) – Output of the neural network on the test set; (b) – Target mass imbalance on the test set; (c) – Error between (a) and (b); (d) – Output of the NNs and the target on the real flight test data

Kalman Filter [7][8] in the previous work as a natural simplification of the global extended Kalman algorithm (GEKF) by ignoring the interdependence of mutually exclusive groups of weights, thereby allowing the computational complexity of EKF to be adjusted to the low requirements of the computational resources. Since GEKF is supposed to produce a better approximation and we are not getting into a problem of limitation of computational resources in this application, we use GEKF in this application instead of NDEKF. Generally, to produce the comparable results to standard backpropagation (SBP), EKF often requires significantly fewer presentations of training data and less overall training epochs than SBP. This is one of the main reasons we investigate the EKF algorithm in this application.

Given a network with M weights and N_L output nodes, the weights update for a training instance at time step n of GEKF is given by:

$$A(n) = [R(n) + H'(n)P(n)H(n)]^{-1}, \quad (4)$$

$$K(n) = P(n)H(n)A(n), \quad (5)$$

$$\hat{W}(n+1) = \hat{W}(n) + K(n)\xi(n), \quad (6)$$

$$P(n+1) = P(n) - K(n)H'(n)P(n) + Q(k), \quad (7)$$

$$P(0) = \frac{1}{\eta_p}I, \quad R(n) = \eta_r I, \quad Q(k) = \eta_q I. \quad (8)$$

In above equations, R(n) is a diagonal N_L -by- N_L matrix whose diagonal components are equal to or slightly less than 1. H(n) is a M-by- N_L matrix containing the partial derivatives of the output node signals with respect to the

weights. P(n) is a M-by-M matrix defined as the approximate conditional error covariance matrix. A(n) is a N_L -by- N_L matrix that we refer to as the global scaling matrix. K(n) is a M-by- N_L matrix containing the Kalman gains for the weights. $\hat{W}(n)$ is a vector of length M containing the all the weights values. $\xi(n)$ is the error vector of the network's output layer. While the motivation for the use of artificial process noise in equation (7) was to avoid numerical difficulties, we have found in addition that it significantly enhances the performance of the GEKF algorithms in terms of rate of convergence, avoidance of local minimum and quality of solution.

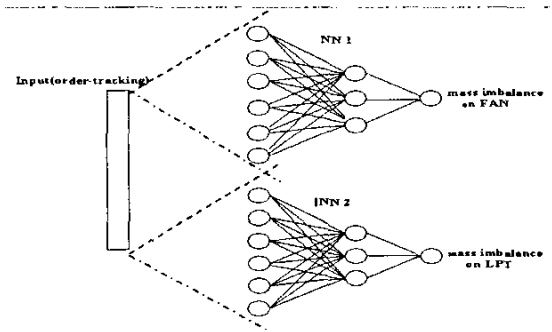


Figure 6. Network architecture

Experiment Results on order-tracking data:

Since we have mass imbalance on both Fan and LPT and they are supposed to be independent, we apply two EKF networks to diagnose the mass imbalances on Fan and LPT separately. In this way, we can speed up the convergence of the EKF network and diagnose the location of the mass imbalance at the same time. We construct the network as 6-3-1, using sigmoid function in the two hidden layers and pure linear function in the output layer, as in Figure 6. Figure 7 shows the best performance of network diagnose the mass imbalance only on LPT using the same data set as in BP neural networks based on 5 trials.

At the end of 100 epochs training, the performance of the EKF network is superior to the most of the fast variants of backpropagation algorithms in Table 3. Since the extended Kalman Filter is very sensitive to the initial values of the network, we usually present the best performance from a few trials as the result.

(3) Support Vector Machines:

In the last few years, there has been a surge of interest in Support Vector Machines (SVMs). SVMs have empirically been shown to give good generalization performance on a wide variety of problems such as handwritten character recognition, face detection and function approximation

[10][11][12]. Basically, Support Vector Machines are algorithms in machine learning. The main idea of support vector machines (SVMs) is to construct a hyper-plane decision surface in such a way that the margin of separation

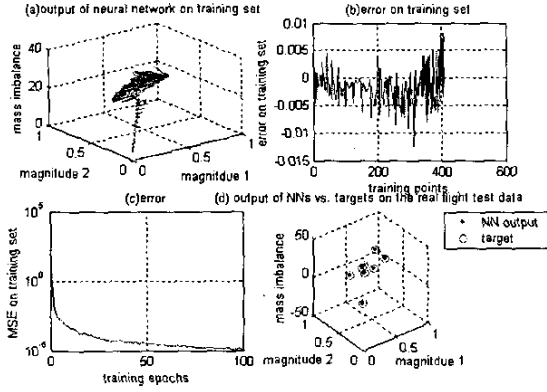


Figure 7. Performance of the network using GEKF algorithm on the interpolated measurements and real flight test data from Strut for the mass imbalance on LPT. Magnitude 1 and magnitude 2 refer to the magnitude of the interpolated or real measurement at the two fixed frequency bins of the order-tracking data respectively. At the end of the 100 epochs training, MSE on the training set and on the real flight test data is 5.8784e-05 and 2.3125e-04, respectively. (MSE=mean squared error), (a) – Output of the neural network on the training set; (b) – Error on the training set; (c) – MSE through training; (d) – Output of the NNs and the target on the real flight test data.

between examples is maximized. The machine achieves this desirable property by following a principled method of structural risk minimization [20].

A support vector machine problem can be stated as follows: Given the training samples $\{(x_i, d_i)\}_{i=1}^N$, where x_i is the input vector, d_i is the corresponding target output, find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximize the objective function:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(x_i, x_j) \quad (9)$$

subject to the constraints:

$$\sum_{i=1}^N \alpha_i d_i = 0 \text{ and } 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \quad (10)$$

where C is a user-specified parameter. The $K(x_i, x_j)$ is the kernel function which maps the multidimensional input space (nonlinearly separable patterns) into a new feature space with higher dimension where patterns are linearly separable. We can solve this optimization problem with quadratic programming and get the optimized Lagrange multiplier. Then we can get the weight vector:

$$W_o = \sum_{i=1}^N \alpha_i d_i \varphi(x_i) \quad (11)$$

Where $\varphi(x_i)$ is the image induced in the feature space due to x_i . The support vector machine output y becomes

$$y = \sum_{i=1}^N \alpha_i d_i \varphi^T(x) \varphi(x_i) \quad (12)$$

Here we apply SVMs technique to solve a regression problem, which we call support vector regression [1][10]. Given the measurement from the sensors, SVM attempts to diagnose the mass imbalance on the engine.

Experiment Results on order-tracking data:

Data generated from the converted test matrix is applied for SVMs. It is also the same data set as used in MLP and EKF. After we try a few different kernel functions and parameter C , we find SVMs are able to give fairly good performance with Radial Basis Function (RBF) kernel and $C = 500$ in such a diagnosis. Similar to EKF, we still can apply two SVMs to diagnose the mass imbalance on FAN and LPT, respectively. In this way, the amount and the source of the mass imbalance are diagnosed at the same time. But remember, we need enough original flight test data with different mass imbalance cases to validate this method. Figure 8 shows the results on the training set and test set measured from 'strut' for the mass imbalance on LPT. MSE on the training set and on the real flight test data is 2.4271e-04 and 0.003642, respectively.

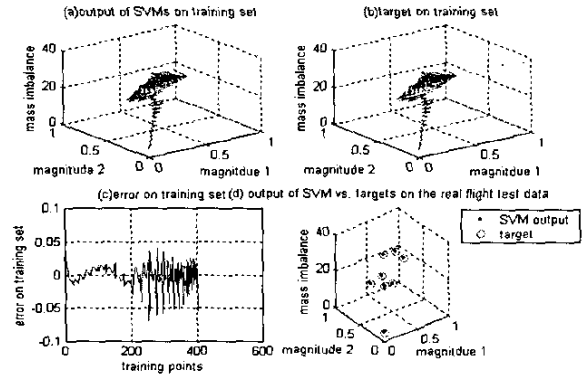


Figure 8. Performance of SVM on the interpolated measurements and real flight test data from Strut for the mass imbalance on LPT. Magnitude 1 and magnitude 2 refer to the magnitude of the interpolated or real measurement at the two frequency bins of the order-tracking data respectively. At the end of the training, MSE on the training set and on the real flight test data is 2.42e-04 and 3.64e-03, respectively. (MSE=mean squared error) (a) – Output of SVM on training set; (b) – Target on the training set; (c) – Error on the training set; (d) – Output of SVM and the target on the real flight test data.

III. CONCLUSIONS

Several neural network methods are applied on the real flight test data to diagnose aircraft engine condition, herein mass imbalance on the low-pressure turbine (LPT) and fan. To enrich the data set, we use the interpolation technique on the order-tracking tables to generate more data points to be presented to neural network for training purpose. Especially for the measurement from the converted test matrix, we try to use a smooth interpolation surface to fit the original data points to approximate the nonlinear mapping function between order-tracking data and mass imbalance target. MLP, EKF and SVMs are applied on the surface-interpolated data from the converted test matrix. Since the mass imbalances on LPT and Fan are independent, this makes it possible to use two separated networks to identify the mass imbalance on LPT and Fan for the same data set, in which the amount and the location of the mass imbalances are diagnosed simultaneously. The performance comparison among the best configurations in these three methods is shown in the Table 4.

	Epochs	MSE on training set	MSE on real flight test data
MLP (trainm)	218	9.93e-06	3.25e-04
EKF	100	5.87e-05	2.31e-04
SVMs	N/A	2.42e-04	3.64e-03

Table 4. Performance comparison between MLP, EKF and SVMs on the surface-interpolated data from the converted test matrix and the real flight order-tracking data.

As shown in the tables above, all three methods give the promising performance on the diagnosis of mass imbalance on engine. The superior performance by MLP using backpropagation could be caused by the overtraining on the interpolated data set since we don't have enough original flight test data with different mass imbalance available to do the validation. From the experiments implemented so far, we show that the neural network inverse model method has the capability to build up the valid model between the vibration flight test data and the mass imbalance located on the FAN and LPT of engine and will achieve fairly good performance and accuracy if given enough amount of original flight test data with different mass imbalance cases.

ACKNOWLEDGEMENTS

We greatly appreciate the assistance of Walter Viebrock and Joseph Kornowske of The Boeing Company in interpreting the flight test data. We also wish to acknowledge the National Science Foundation and the M.K. Finley Missouri endowment for partial support of this research.

REFERENCES

- [1] Simon Haykin, *Neural Networks: a comprehensive foundation*, 2nd Ed., Prentice Hall, New Jersey, 1999.
- [2] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, NJ, 1996.
- [3] Xiao Hu; John Vian; Jai Choi; David Carlson; Donald C. Wunsch II, "Propulsion vibration analysis using neural network inverse modeling", *Neural Networks, Proceedings of the IJCNN'02 International Joint Conference on*, vol.3, pp. 2866-2871, 2002.
- [4] Joseph R. Slepiski, "Data Summary", Boeing Technical Report, Jul. 2002.
- [5] Howard Demuth; Mark Beale, *Neural Networks Toolbox User's Guide (Version 4) For Use with MATLAB, Mathworks*.
- [6] Puskorius, G.V.; Feldkamp, L.A. "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks", *Neural Networks, IEEE Transactions on*, vol. 5, pp. 279-297, Mar. 1994.
- [7] Puskorius, G.V.; Feldkamp, L.A., "Decoupled extended Kalman filter training of feedforward layered networks", *Neural Networks, Proceedings of IJCNN-91-Seattle International Joint Conference on*, vol. 1, pp. 771-777, 1991
- [8] Murtuza, S.; Chorian, S.F., "Node decoupled extended Kalman filter based learning algorithm for neural networks", *Intelligent Control, Proceedings of the 1994 IEEE International Symposium on*, pp. 364-369, 1994.
- [9] John C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, Microsoft Research, 1998.
- [10] S.R. Gunn. "Support vector machines for classification and regression". Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton, 1998.
- [11] C. Burges. "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, 2(2): 121-167, 1998.
- [12] Chih-Chung Chang; Chih-Jen Lin., "Libsvm: a library for support vector machines (version 2.3)", 2001.
- [13] Singhal, S.; Wu, L., "Training feed-forward networks with the extended Kalman algorithm", *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP-89)*, vol.2, pp. 1187-1190, 1989.
- [14] Alexander Ypma, "Learning method for machine vibration analysis and health monitoring", Ph.D. Thesis, Delft University of Technology, 2001.
- [15] Brotherton, T.; Chadderton, G.; Grabill, P. "Automated rule extraction for engine vibration analysis", *Proceedings of IEEE Aerospace Conference*, vol. 3, pp. 29-38, 1999.
- [16] Andrew C. McCormick and Asoke K. Nandi, "Real-Time classification of Rotating Shaft Loading Conditions Using Artificial Neural Networks", *IEEE Trans. On Neural Networks*, Vol. 8, No.3, pp. 748-757, May, 1997.
- [17] J. T. Renwick, "Vibration Analysis - A proven technique as a predictive maintenance Tool," *IEEE Trans. Ind. Appl.*, vol. 21, pp. 324-332, Mar, 1985.
- [18] J. S. Mitchell. *An introduction to machinery analysis and monitoring - 2nd ed.* PennWell Publ. Comp. 1993.
- [19] Martin T. Hagan, Howard B. Demuth, Mark Beale, *Neural Network Design*, PWS Pub., Boston, 1996.
- [20] Vladimir N. Vapnik., *The Nature of Statistical Learning Theory*, Springer, NY, 1995.