Partial Copyright License

Title of Dissertation:

Approaches for Early Fault Detection in Large Scale Engineering Plants

Author: ______________________________

Stephen W. Neville, B.Eng., M.A.Sc.
January 31, 1998.

Kourounakis, N.P., S.W. Neville and N.J. Dimopoulos, “Early Fault Detection in Cable Television Networks (the Case of the Reverse Pilot)”, *1997 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, Victoria, Canada, August 20-22, 1997, pp.511-554.

Dimopoulos, N.J., A. Watkins, S. Neville, and K.F. Li, "An Expert Network Analyzer: Knowledge Acquisition, Fault Diagnosis and Prediction", presented at the *1993 DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics*, Ottawa, Ontario, Nov. 14-17, 1993.

Dimopoulos, N.J., S. Neville, A. Watkins, K.F. Li, and E.G. Manning, "Neural Networks in Fault Identification in Large Communication Networks", 2*nd IEEE Mediterranean Symposium on New Directions in Control and Automation*, Malene-Chania, Crete, Greece, June 19-22, 1994, pp. 25-31

Neville, S. and N.J. Dimopoulos, "Reliable Detection of Current Faults in Cable Amplifier Networks", *Technical Report ECE-95-03*, Dept. of Electrical and Computer Engineering, University of Victoria, June, 1995.

Neville, S. and N.J. Dimopoulos, "Techniques for Confident and Reliable Fault Detection in Large Scale Engineering Plants", *Proceedings of the 1995 IEEE Conference on Systems, Man and Cybernetics*, Vancouver, B.C., Oct. 23-25, 1995, pp.1807-1812.

Dimopoulos, N.J., S. Neville, J.T. Dorocicz, and C. Jubien, "Training Asymptotically Stable Recurrent Neural Networks", *Proceedings of the 1995 IEEE Conference on Systems, Man and Cybernetics*, Vancouver, B.C., Oct. 23-25, 1995, pp. 4392-4397.

Dorocicz, J., E. Kosmatopoulos, S. Neville, and N.J. Dimopoulos, "Recurrent Neural Networks in System Modeling and Error Detection", *Second World Automation Congress*, Montpellier, France, May 27-30, 1996.

Dorocicz, J., E. Kosmatopoulos, S. Neville, and N.J. Dimopoulos, "Recurrent Neural Networks for Fault Detection", *Proceedings of the 4th IEEE Mediterranean Symposium on New Directions in Control and Automation*, Chania, Greece, June 10-14, 1996, pp 17-22.

Kosmatopoulos, E., J. Dorocicz, N.J. Dimopoulos, S. Neville, A. Watkins, and K.F. Li. "Recurrent Neural Networks in Modelling and Fault Detection of the Forward Pilot in Main Trunk Amplifiers", *Proceedings of the Canadian Cable Television Association*, Edmonton, Canada, June 2-5, 1996, pp 21-28.

Dimopoulos, N.J., J. Dorocicz, C. Jubien, and S. Neville, "Training Asymptotically Stable Recurrent Neural Networks", *Intelligent Automation and Soft Computing*, Vol. 2, No. 4, 1996, pp.375-388.

Neville, S. and N.J. Dimopoulos, "Noise Estimates and Smoothing Techniques for the Signals of Limit Checking Fault Detection Systems", *Technical Report*, Dept. of Electrical and Computer Engineering, University of Victoria, Jan. 1997.

Vita

Surname: Neville Given Names: Stephen William

Place of Birth: Victoria, British Columbia, Canada

Educational Institutions Attended:

University of Victoria 1985 to 1998

Degrees Awarded:

B.Eng. (Computer Engineering) 1990

M.A.Sc. (Electrical Engineering) 1992

Honours and Awards:

NSERC Post Graduate Scholarships 1991 - 1994

University of Victoria President's Research Grant 1991 - 1994

B.C. Science Council G.R.E.A.T. Scholarship 1991 - 1993

B.C. Advanced Systems Institute G.R.A.P. Scholarship 1992

Association of Professional Engineers and Geoscientists B.C. Award for Most Promising Computer Engineering Student 1990

Publications:

Dimopoulos, N.J., K. Li, A. Watkins, S. Neville and A. Rondogainnis, "An Expert Network Analyzer", *Technical Papers of the 35th Canadian Cable Television Annual Convention*, Vancouver B.C., June 1-3, 1992, pp. 123-127.

Watkins, A., N.J. Dimopoulos, S. Neville, and K.F. Li, "Flowtool: A Procedural-Knowledge Acquisition Tool", *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, B.C., May 19-21, 1993, pp. 31-34.

Dimopoulos, N.J., K. Li, A. Watkins, and S. Neville, "An Expert Network Analyzer: Fault Diagnosis and Prediction", *Proceedings of the 1993 Canadian Conference on Electrical and Computer Engineering*, Vancouver B.C., Sept. 14-17, 1993, pp. 1213-1216.

cepts for the lower bounds were generated from the interval given by $[2I_{min} - I_{max}, I_{min}]$. All possible pairing of these upper and lower thresholding functions were then compared to determine which resulting set had the lowest enclosed area.

This set provided an initial estimate of the regions in which the optimal thresholding functions were likely to be found. Specifically, the quantization step intervals in which these initial upper and lower thresholding functions solutions were found, given as $\left[q^u_{min}, q^u_{max}\right]$ and $\left[q^l_{min}, q^l_{max}\right]$, were then searched more exhaustively to determine if a more optimal solution than the initial solution could be found. In particular, the same search strategy as above was utilized with the exception that 100,000 upper and lower thresholding functions were generated from the respective quantization step intervals. All possible pairing of these thresholding functions where then explored to determine the pair which had the lowest enclosed area, as specified by Eq. 4.38.

The behavioural map was then offset by $-T_{min}$ along the x-axis to place it back in its original position. This had the effect of shifting the near-optimal thresholding functions y-intercepts by $-\alpha^u T_{min}$ and $-\alpha^l T_{min}$, respectively. This final pair of lines was then deemed as the near-optimal linear thresholding functions for the given setting of the probability of false alarm.

Appendix D:

Brute Force Search Strategy

D.1 Introduction

In Section 4.2.7.5 of Chapter 4, a brute force search strategy was utilized to determine the near-optimal upper and lower linear thresholding functions for a given current versus temperature behavioural model under a given probability of false alarm setting. This section will provide a detailed description of the brute force search strategy which was employed.

D.2 Search Strategy

The basic idea behind the brute force search strategy was to develop dense sets of upper and lower thresholding functions which met the requirements of having a probability of false alarm of $\frac{\gamma}{2}$, under the policy of equally splitting the probability of false alarm between the upper and lower thresholding functions. All the possible combinations of these thresholding functions were then analyzed in terms of the resulting enclosed area, given by Eq. 4.38, to determine the pair of upper and lower thresholding functions which had the minimum enclosed area. This pair of thresholds was then chosen as the near-optimal solution for the given probability of false alarm setting.

The actual generation of the dense set of lines was done in a series of steps. First, the behavioural model was offset to the left along the x-axis by the minimum observed temperature level, T_{min}, such that initial point on the behavioral model, (T_{min}, I_{min}), was located on the y-axis. A set of 10,000 upper bound line intercepts were then generated by uniformly partitioning of the y-axis interval $[I_{min}, I_{max}]$. A simple search procedure was then utilized to find the each of the slopes corresponding to the given y-intercepts which produced an upper bound line with the required probability of false alarm $\frac{\gamma}{2}$. 10,000 lower bound lines were then generated in a similar manner with the exception that the y-inter-

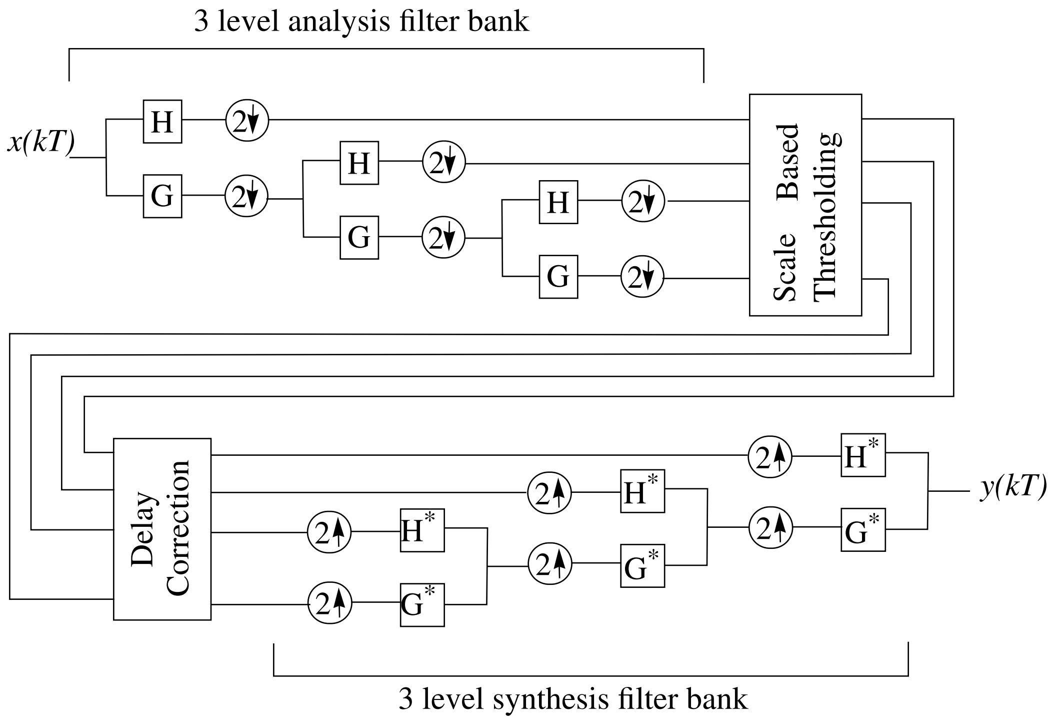


FIGURE C-3.
Block diagram of example implementation of wavelet de-noising for a 3 level wavelet decomposition.

optimally setting the fault detection thresholds for each of the scales is also, obviously, not able to be implemented directly in a real-time system. A solution to this problem is to utilize historical data sets to identify the optimal detection thresholds, and then update these thresholds as needed when new data is obtained.

This inverse discrete wavelet transform is then given by

$$(Gy, GHy, GH^2y, GH^3y, \ldots, GH^{j-1}y, H^jy) \rightarrow y = \sum_{j=0}^{n-1} (H^*)^j G^* r^{(j)} + (H^*)^n s^{(0)}$$

and the pyramid processing scheme of Figure C-2. In this case, the signal is up-sampled by 2 at each stage by inserting a zeros between each sample of the signal.

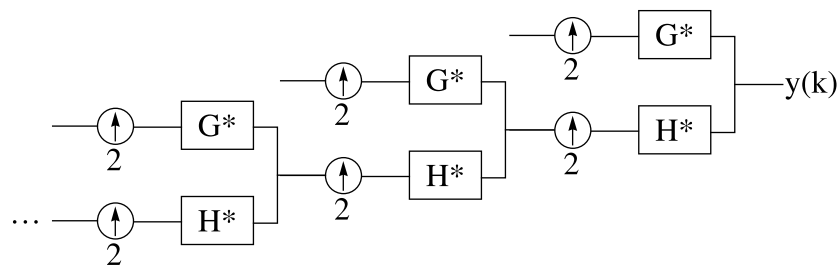


FIGURE C-2.
Inverse discrete wavelet pyramidal processing scheme (Synthesis filter bank).

The process of wavelet de-noising therefore becomes one of utilizing the analysis filter bank to perform the conversion from time domain to wavelet domain, performing the necessary thresholding operation at each scale, and then utilizing the synthesis filter bank to convert the signal back to time domain. Figure C-3 shows the block diagram describing this process for an example 3 level wavelet decomposition. One complicating factor in utilizing this type of de-noising approach is that the signal delay between the various signal paths is not identical, and hence particular care must be taken to ensure that the signal is reconstructed appropriately, particularly when signal processing steps, such as de-noising, are introduced between the analysis and synthesis filter banks.

An additional factor of concern with the wavelet de-noising technique is that it will introduce a delay into the fault detection system. No direct attempt has been made to quantify the nature of this delay. But it is anticipated, since the delay is mainly comprised of the filters' group delay plus the additional processing time required for the de-nosing operation, that the delay within a real-time implementation of the de-noising procedure would be of a relatively short duration and would not significantly impact on the early fault detection abilities of the resulting system. The methodology presented previously for

information is captured through $r^{(j-1)}$ via the mirror filter G. The wavelet transform therefore retains all the information about the original signal and is an invertible operation. The discrete wavelet transform of a signal y can then be written as:

$$y \rightarrow (Gy, GHy, GH^2y, GH^3y, \ldots, GH^{n-1}y, H^ny) \tag{C.3}$$

This formulation results in a pyramidal processing scheme, shown in Figure C-1, which similar to those used in image processing [10][72]. At the first level of processing the highest resolution signal features are extracted. At each further processing level the next highest resolution features are removed. At the final level, only the very coarse feature remain. Both outputs from each subsequent stage are effectively oversampled since when viewed as baseband signals they now have approximately half their former bandwidth. The sub-sampling by 2 removes this redundant information and leaves the outputs critically sampled. In general, for a signal record of length 2^L the decomposition proceeds for L-1 stages, with 2^l wavelet coefficients being generated at each scale $l = 0,\ldots,L-1$.

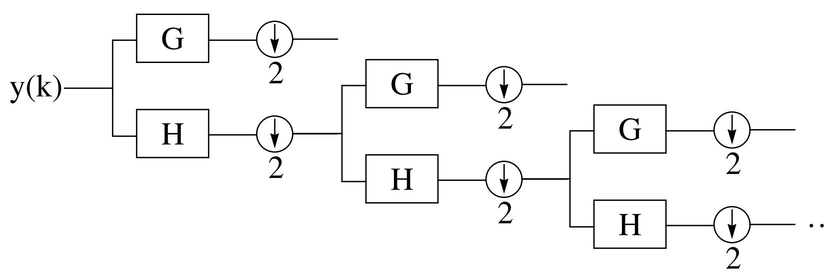


FIGURE C-1.
Discrete wavelet transform pyramidal processing scheme [79] (Analysis filter bank).

To generate the inverse wavelet transform the adjoint operators H^* and G^* are defined as

$$(H^*a)_n = \sum_k h(n-2k)\, a_n \tag{C.4}$$

$$(G^*a)_n = \sum_k g(n-2k)\, a_n \tag{C.5}$$

Appendix C:

Real-Time Implementation of the Wavelet De-noising Process

C.1 Introduction

In Chapter 5, a methodology for performing the wavelet de-noising of the forward pilot status data signals was introduced. In that chapter, the de-nosing methodology was developed in terms of a batch processing context. Obviously to by useful in an operational fault detection system it is necessary to be able to perform the wavelet de-nosing in real-time. To this end, this appendix will present an overview of how wavelet filters can be implemented to process real-time data through the used of filter bank theory.

C.2 Real-Time Implementation

Through filter bank theory, real-time implementations of wavelet transforms are possible [78][82]. In particular, the sequences $\{h(k), k \in Z\}$ and $\{g(k), k \in Z\}$, given in Eq. 5.22 and Eq. 5.23, can be viewed, as the coefficients of quadrature mirror filters. In particular, the $h(k)$ terms represent low pass filtration while the $g(k)$ terms represent high pass filtration. Function operators *H* and *G* can be defined for a sequence $a = \{a_n\}$ such that,

$$(Ha)_k = \sum_n h(n-2k)\,a_n \tag{C.1}$$

$$(Ga)_k = \sum_n g(n-2k)\,a_n \tag{C.2}$$

Using this operator notation, if the original signal is given by $s^{(n)}$ then each successive scale of the wavelet transform is generated by $s^{(j-1)} = Hs^{(j)}$ and $r^{(j-1)} = Gs^{(j)}$. At each scale, the signal is smoothed by the low pass filter H and the lost high frequency

the 1 and 2 minute sampling intervals. Hence, Eq. B.3 can be used to estimate the constant sampling rate of the example plant's status monitoring system. In particular, this method gives an estimate of $77.26\,\mathrm{sec.} \leq T_2 < 77.77\,\mathrm{sec.}$ for the example plant's status monitoring system's polling period. This bound is narrow enough to permit the assumption that the sampling period is constant.

if the probability of 1 minute sampling intervals, as seen from C_1, is to be P_1.

The true value of T_2 therefore lies within the interval given by the lower bound of Eq. B.1 and the upper bound of Eq. B.2:

$$T_1 (P_2 + 1) \leq T_2 < T_1 (2 - P_1) \tag{B.3}$$

Eq. B.3 can therefore be used to generate an estimate of T_2 provided that estimates of P_1 and P_2 can be obtained and that $T_2 \in (T_1, 2T_1)$. In the case where there are no other sampling intervals besides the 1 and 2 minutes intervals, Eq. B.3 collapses, as expected, to

$$T_2 = T_1 (1 + P_2) \tag{B.4}$$

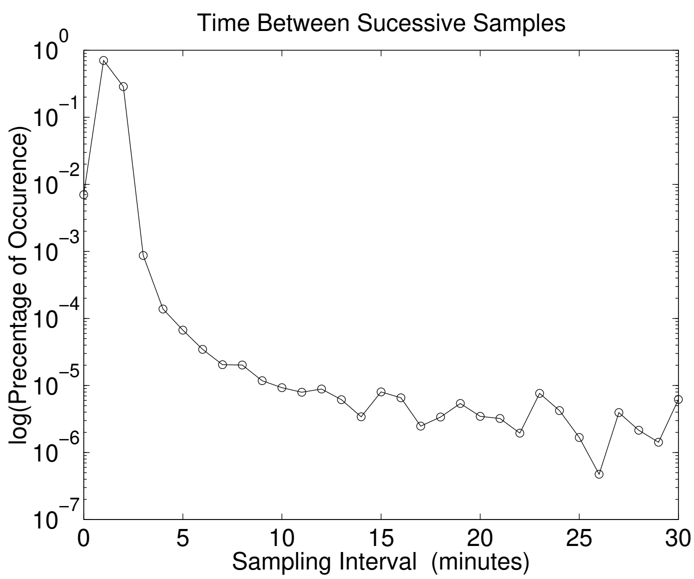


FIGURE B-2.
Histogram of time between successive samples for the 170 randomly selected amplifier from the example engineering plant (October 1995 to October 1996)

Figure B-2 shows the distribution of the sampling intervals for the complete 13 month data set obtained from the example plant under consideration within the body of this work. It is quite clear from this figure that the majority of the sampling intervals (99.15%) are

ples. For example, if one of C_2's ticks is assumed to fall within the *[0,1]* of C_1, then it would be time stamped in reference to C_1 as occurring at time 0. If the subsequent tick fell within in *[1,2]* of C_1 then that tick would be time stamped time 1. The sampling interval between the two ticks would then be 1, as measured via C_1.

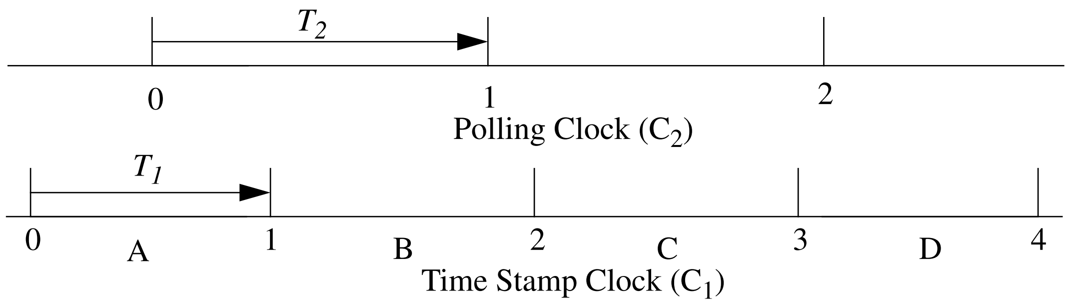


FIGURE B-1.
Illustration of relationship between the time stamp clock at the polling clock.

Let A be some interval of C_1, picked at random, which contains one C_2 tick. Then, the location of C_2's tick within A will be uniformly distributed. Then, let P_1 be the probability of C_2's second tick falling in the second interval of C_1, labelled B in Figure B-1, (i.e. time stamps 1 minute apart) and let P_2 be the probability of C_2's second tick falling in the third interval of C_1, labelled C in Figure B-1, (i.e time stamps 2 minutes apart). Then, if C_2's first tick occurred within the upper P_2 percent of A then C_2's second tick must occur within interval C, given that $T_1 < T_2 \le 2T_1$. This means that the sampling interval between these two ticks will be measure as 2 minutes from the C_1 time stamps. Hence,

$$T_1 (P_2 + 1) \le T_2 < T_1 (P_2 + 2) \qquad \text{(B.1)}$$

if the probability of 2 minute sampling intervals, as seen from C_1, is to be P_2.

Alternatively, if C_2's tick occurred within the lower P_1 percent of interval A then C_2's next tick should occur within interval B. Hence,

$$T_1 < T_2 < T_1 (2 - P_1) \qquad \text{(B.2)}$$

Appendix B:

Validation of the Constant Sampling Rate Assumption

B.1 Introduction

In Chapter 3, an assumption was made that the data that was collected form the example engineering plant could be assumed to have been sampled at a constant sampling rate. This appendix will provide the justification for this assumption.

B.2 Estimation of the Sampling Period

The data received by the example plant's status monitoring system is time stamped with its arrival time by the status monitoring system. In the data set of interest within this work, these time stamps are only given to 1 minute resolutions. This creates a problem since the example plant's polling period is of the same order of magnitude. In particular, the monitoring system appears to employ a non-constant sampling interval since the subtraction of the successive time stamps produces a non-constant result. This effect will be subsequently shown to be an artifact of the low resolution of the time stamps.

The actual constant sampling period of the monitoring system can be determine by analyzing the statistics of the sampling intervals (the time periods, measured utilizing the time stamps, between the successive data samples). In essence the problem of determine the monitoring system sampling period can be viewed as the problem of determining an unknown clock signal period from knowledge of a known clock signal and the statistics of their interaction.

In particular, it can be assumed that there are two clock signals C_1 and C_2 with constant periods. C_1 has a period of T_1 which is defined to be 1 minute in duration and C_2 has a period of T_2, which is contained in the interval $(T_1, 2T_1)$ but its exact value is unknown. Figure B-1 shows an illustration of these two clock signals and their relationship. If C_1 is assumed to be the time stamp clock then the period of C_2 can be determined by analyzing the statistics of sampling intervals obtained via the C_1 time stamps, where these sampling intervals are obtained by subtracting the time stamps of successive sam-

requires that the status of the reverse amplifier module be measured from the head-end. The amplifiers, when installed, therefore must be adjusted in series starting from the one closest to the head-end and proceeding one amplifier at a time down the various branches.

response centred in the lower region of the network's bandwidth and is termed the low pilot amplifier. The second type of amplifier has a frequency response centred in the upper portion of the network's bandwidth and is termed the high pilot. The intermediate frequencies see the effects of the overlapping high and low frequency pilots. The gain for these frequencies can therefore be controlled by controlling the shape of the respective upper and lower skirts of the low and high pilot amplifiers. The overall goal is to obtain a flat frequency independent gain across the network's entire bandwidth by carefully adjusting the shape of the high and low pilot gain functions, as illustrated in Figure A-2. A secondary effect of the differential attenuation across the frequency range is that the low pilot amplifiers can be spaced farther apart than the high pilot amplifiers.

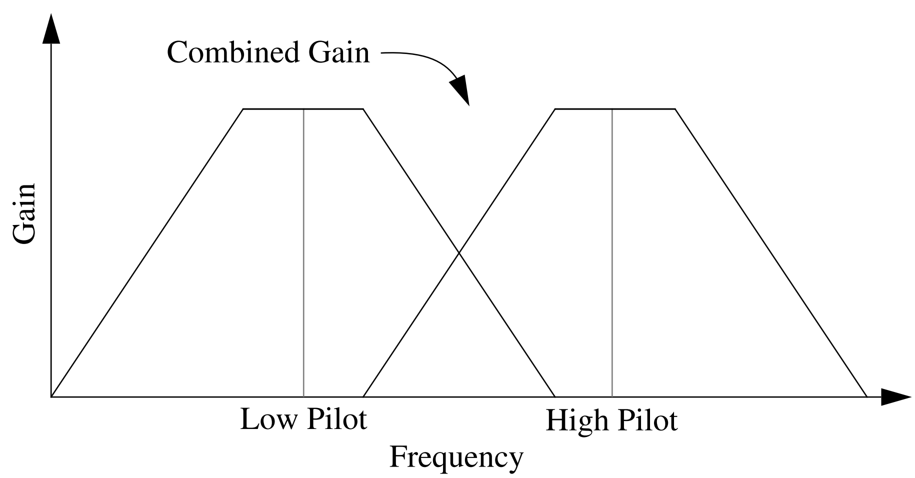


FIGURE A-2.
Flat cross-frequency response achieved with combined low and high pilot amplifier responses.

Since the upstream (reverse pilot) path is a low bandwidth path, only one type of reverse amplifier is required. As in the forward pilot case, the gain of each amplifier is used to compensate for the loss incurred over a single cable span. The difference in the reverse case is that instead of compensating for the loss that was incurred on the previous span the reverse amplifiers are set to compensate for the loss that is about to be incurred in the subsequent span. For example, in Figure A-1 the reverse gain of amplifier Amp_241 would be adjusted to recover for the loss incurred in sending the reverse signal from amplifier Amp_241 to amplifier Amp_240. The nature of the reverse path therefore

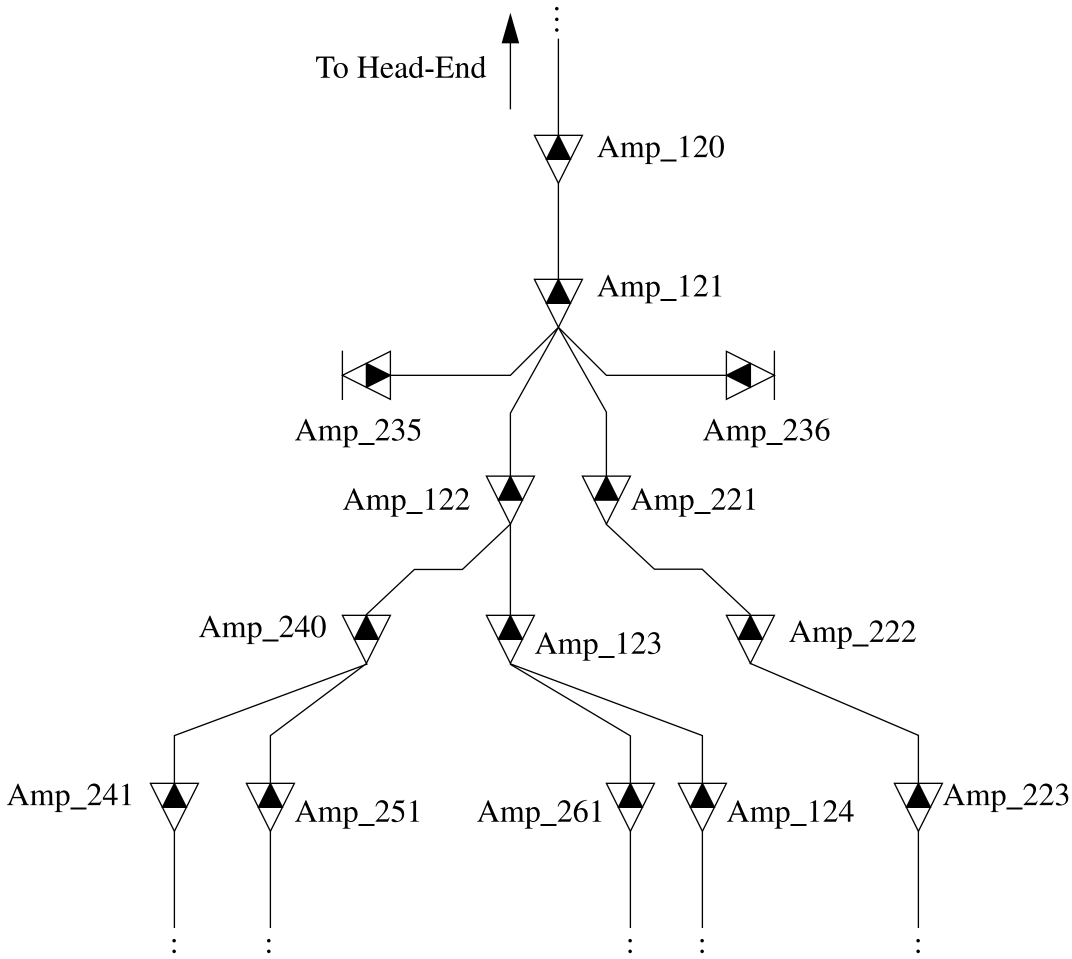


FIGURE A-1.
Typical structure of a cable trunk amplifier network.

traditionally been used to send status data from the trunk amplifiers back to the head-end. More recently this upstream path has also been used to provide a data path from the subscribers to the head-end for use in interactive services.

The basic theory of operation of the downstream (forward pilot) path of the trunk amplifier portion of the network is that each amplifier amplifies the cable signals by the amount of loss incurred over the preceding cable span. This principle is somewhat complicated by the fact that the cable network is designed to operated over a very broad frequency range. Obviously, different frequency signals are attenuated to different degrees during their transmission. In general, this problem is dealt with by utilizing two different forms of trunk amplifiers within the network. The first type of amplifier has a frequency

Appendix A:

Overview of Cable Trunk Amplifier Networks

A.1 Introduction

At the heart of cable television distribution systems are the cable trunk amplifier networks. These networks provide the framework for the controlled distribution of the cable signals from a centrally located injection site, termed the head-end, to the subscribers' homes, located throughout a surrounding metropolitan region. As such, cable trunk amplifier networks are critical in providing consistent and high quality cable services. This appendix will provided an overview of cable amplifier networks in terms of their structure and basic theory of operation.

A.2 Structural Overview

Typically cable plants are structured as a tree which spans across a given metropolitan area. At the trunk of the tree is the head-end where the cable signals are injected. Forming the branches of the tree are groups of trunk amplifiers, serially interconnected by coaxial cable spans, which form the main distribution network. Off the trunk amplifiers are the distribution amplifiers which in turn form tree structured distribution networks at the neighborhood level. Taps from the distribution networks then feed the cable signals into the subscribes homes. A typical metropolitan cable plant can contain on average several hundred to several thousand trunk amplifiers, each of which is capable of driving ten to twelve distribution amplifiers which in turn can provide cable service to up to two or three hundred subscribers. Figure A-1 illustrates the typical structure of the cable trunk amplifier network portion of a typical cable plant.

A.3 Theory of Operation

Most cable amplifier plants, and specifically the example plant discussed in this work, are two way asymmetrical communication networks. The downstream path, from the head-end to the subscribers homes, is a high bandwidth path which is used mainly for delivering cable television services. The upstream path is a low bandwidth path which has

[78] Strang, G. and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Massachusetts, 1996.

[79] Vetterli, M. and C. Herley, "Wavelets and Filter Banks: Theory and Design", *IEEE Transactions on Signal Processing*, Vol. 40, No. 6, Sept. 1992, pp. 2207-2232

[80] Viadakovic, B., and P. Müller, "Wavelets for Kids", *Duke University*, 1991.

[81] Viadakovic, B., "Wavelets and Bayesian Statistics", Invited talk at: *Interface '94- 26th Symposium on the Interface Computing Science and Statistics*, June 15-18, 1994, RTP, NC.

[82] Wickerhauser, M.V., *Adapted Wavelet Analysis from Theory to Software*, IEEE Press, Piscataway, New Jersey, 1994.

[83] Widrow, B. and M. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", *IEEE Proceedings, Special Issue on Neural Networks*, Vol. 78, No. 9, Sept. 1990.

[84] Wilbers, D.N. and J.L. Speyer, "Detection Filters for Aircraft Sensor and Actuator Faults", Proceedings of the ICCON '89 International Conference on Control and Applications, Jerusalem, April 1989.

[85] Willsky, A.S., "A Survey of Design Methods for Failure Detection in Dynamic Systems", *Automatica*, Vol. 12, pp. 601-611, 1976.

[86] Windham, "Geometrical Fuzzy Clustering Algorithms", in *Fuzzy Models for Pattern Recognition*, J.C Bezdek and S.K. Pal, Editors, IEEE Press, Piscataway, New Jersey, pp. 123-129, 1992.

[87] Zadeh, L.A., "Fuzzy Sets", *Information and Control*, Vol 8, pp. 338-353, 1965.

[88] Zurada, J.M, *Introduction to Artificial Neural Systems*, West Publishing Co., St. Paul, Minnesota, 1992.

[67] Neville. S.W., and N.J. Dimopoulos, "Techniques for Confident and Reliable Fault Detection in Large Scale Engineering Plants", *Proceeding 1995 IEEE International Conference on Systems, Man, and Cybernetics*, Vancouver, Canada, Oct.22 - 25, 1995, pp. 1807-1812.

[68] N.J. Dimopoulos, S.W. Neville, J.T. Dorocicz and C. Jubien, "Training Asymptotically Stable Recurrent Neural Networks", *Proceeding 1995 IEEE International Conference on Systems, Man, and Cybernetics*, Vancouver, Canada, Oct.22 - 25, 1995, pp. 4392-4397.

[69] Patton, R.J., P.M. Frank and R.N. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Applications*, Prentice-Hall, 1989.

[70] Peebles, P.Z., *Probability, Random Variables, and Random Signal Principles*, McGraw-Hill, Inc., Princeton, New Jersey, 1993.

[71] Rumelhart, D.E. and J.L. McClelland, *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1 and 2, M.I.T. Press, 1986.

[72] Rosenfield, E.A., *Multiresolution Techniques in Computer Vision*, Springer-Verlag Ltd., New York, 1984.

[73] Ross, T.J., *Fuzzy Logic with Engineering Applications*, McGraw-Hill Inc., Princeton, New Jersey, 1995.

[74] Takagi, T. and M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp. 116-132, 1987.

[75] Saito, N., "Feature Extraction using Local Discriminant Basis", *Ph.D. Dissertation*, Yale University, December 1994.

[76] Sripad, A.S. and D.L. Snyder, "A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-25, Oct. 1977 pp. 442-448.

[77] Stien, C., "Estimation of the Mean of a Multivariate Normal Distribution", *Annals of Statistics*, Vol. 9, No. 6, 1981, pp. 1135-1151.

[57] Kourounakis, N.P., S.W. Neville and N.J. Dimopoulos, "Early Fault Detection in Cable Television Networks (the Case of the Reverse Pilot)", *1997 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, Victoria, Canada, August 20-22, 1997, pp.511-554.

[58] Laukonen, E.G., K.M. Passino, V. Krishnaswami, G.-C. Luh, and Rizzoni, G., "Fault Detection and Isolation for an Experimental Internal Combustion Engine via Fuzzy Identification", *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 3, 1995, pp. 347-355.

[59] Lou, X.C., A.S. Willsky and G.L. Verghese, "Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems", *Automatica*, Vol. 22, 1986, pp. 333-344.

[60] Mallat, S.G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, July 1989, pp. 674-693.

[61] Massey, B., "Introduction to Status Monitoring II", Rogers Cablesystems Ltd., internal document, 1991.

[62] Nahi, N.E., *Estimation Theory and Applications*, John Wiley and Sons, Inc., New York, New York, 1969.

[63] Nanon, G.P., "Wavelet Regression by Cross-Validation", *Technical Report 447*, Dept. of Statistics, Stanford University, 1994.

[64] Narendra, K.S. and K. Parthasarathy, "Identification and Control of Dynamical Systems using Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, March 1990, pp. 4-27.

[65] Neville, S.W., "A Prototype Expert System Based Diagnostic Tool for Cable Trunk Amplifier Networks", *M.A.Sc. Thesis*, University of Victoria, 1992.

[66] Neville, S. and N.J. Dimopoulos, "Reliable Detection of Current Faults in Cable Amplifier Networks", *Technical Report ECE-95-03*, Dept. Electrical and Computer Engineering, University of Victoria, June, 1995.

[45] Gray, Robert M., “Quantization Noise Spectra”, *IEEE Transactions on Information Theory*, Vol. 26, No. 6, Nov. 1990, pp. 1220-1244.

[46] Haykin, S., *Digital Communications*, John Wiley and Sons, New York, New York, 1988.

[47] Haykin, S., *Adaptive Filter Theory*, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1996.

[48] Haykin, S., *An Introduction to Analog and Digital Communications*, John Wiley & Sons Inc., New York, 1989.

[49] Hirsh, M.W. and S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra*, Academic Press Ltd., 1974.

[50] Hoehfeld, M. and S.E. Fahlman, “Learning with Limited Numerical Precision Using the Cascade-Correlation Learning Algorithm”, *IEEE Transactions on Neural Networks*, Vol. 3, No. 4, pp. 602-611,July 1992.

[51] Hornik, K., M. Stinchcombe and H. White, “Multilayer Feedforward Networks are Universal Approximators”, *discussion paper*, Dept. of Economics, University of California, San Diego, June, 1988.

[52] IIsermann, R., “Process Fault Detection Based on Modeling and Estimation Methods - A Survey”, *Automatica*, Vol. 20, pp. 387-404, 1984.

[53] Johnson, B.W., *Design and Analysis of Fault Tolerant Digital Systems*, Addison-Wesley Publishing Co., 1989.

[54] Jubien, C.S., *Asymptotically Stable Neural Networks for Identification of Nonlinear Dynamic Systems*, M.A.Sc. Thesis, University of Victoria, Victoria, B.C., 1992.

[55] Kitamura, M., “Detection of Sensor Failures in Nuclear Plant Using Analytical Redundancy”, *Transactions of the American Nuclear Society*, Vol. 34, pp. 581-583, 1980.

[56] Kosko, B., “Fuzzy Systems as Universal Approximators”, *IEEE Transactions on Computers*, Vol. 43, No. 11, pp. 1329-1333, Nov. 1994.

[34] Doraiswami, R., “Performance Monitoring and Fault Prediction Using a Linear Predictive Coding Algorithm”, *Automatica*, Vol. 29, No. 4, pp. 1115-1120, 1993.

[35] Dorocicz, J., “Asymptotically Stable Recurrent Neural Networks: Theory and Application”, *M.A.Sc. Thesis*, Dept. Electrical and Computer Engineering, University of Victoria, 1997.

[36] Fahlman, S.E., “The Recurrent Cascade-Correlation Architecture”, *Advances in Neural Information Processing Systems 3*, R. P. Lippmann, J. E. Moody, and D. S. Touretzky (eds.), Morgan, Kaufmann Publishers, Los Altos, California, pp. 190-196, 1991.

[37] Fahlman, S.E. and C. Lebiere, “The Cascade-Correlation Learning Architecture”, *Advances in Neural Information Processing Systems 2*, D. S. Touretzky (ed.), Morgan Kaufmann Publishers, Los Altos, California, pp. 524-532, 1990.

[38] Frank, P.M., “Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy - A Survey and Some New Results”, *Automatica*, Vol. 26, No. 3, pp. 459-474, 1990.

[39] Frank, P.M., “Fault Diagnosis in Dynamic Systems via State Estimation”, *System Fault Diagnosis, Reliability and Related Knowledge Based Approaches*, Vol. 1, pp. 25-98, 1987.

[40] Frank, P.M., “Advanced Fault Detection and Isolation Schemes using Nonlinear and Robust Observers”, *Proceedings of the 10th IFAC World Congress*, Munich 27-31, July 1987.

[41] Friedland, B., *Control Systems Design: An Introduction to State-Space Methods*, McGraw-Hill Inc., Princeton, New Jersey, 1986.

[42] Gersho, A. and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, Massachusetts, 1991.

[43] Gopinath, R.A. and C.S. Burus, “A Tutorial Overview of Filter Banks, Wavelets and Interrelations”, *Proceedings of ISCAS-93*, 1993.

[44] Gopinath, R.A., “An Introduction to Time-Varying Filter Banks and Wavelets”, *internal paper*, Dept. of Electrical and Computer Engineering, Rice University, Houston, Texas, 1993.

[22] Clavier, A.G., P.F. Panter, and D.D. Grieg, “Distortion in a pulse count modulation system”, *AIEE Transactions*, Vol. 66, 1947, pp. 989-1005.

[23] Clavier, A.G., P.F. Panter, and D.D. Grieg, “PCM distortion analysis”, *Electrical Engineer*, Nov. 1947, pp. 1110-1122.

[24] Coifmann, R.R. and D.L. Donoho, “Translation-Invariant De-Noising”, Stanford University, *Wavelets and Statistics*, A. Antoniadis ed., Springer-Verlag Lecture Notes, 1995.

[25] Cooke, W., “Canadian Trunk/Bridger Amplifier”, *Technical Report SP-001-84.O1L*, Rogers Cablesystems Engineering, London, Ontario, 1984.

[26] Cybenko, G., “Approximation by Superposition of a Sigmoidal Function”, *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, pp. 303-314, 1989.

[27] Dexter, A.L., “Fuzzy Model Based Fault Diagnosis”, *IEE Proceedings on Control Theory Applications*, Vol. 142, No. 6, pp. 545-550, 1995.

[28] Dimopoulos, N.J., “A Study of the Asymptotic Behaviour of Neural Networks”, *IEEE Transactions on Circuits and Systems*, Vol. 36, No.5, pp.687-694, May 1989.

[29] Dimopoulos, N.J., K. Li, A. Watkins, S. Neville and A. Rondogainnis, “An Expert Network Analyzer”, *Technical Papers of the 35th Canadian Cable Television Annual Convention*, Vancouver B.C., June 1-3, 1992, pp. 123-127.

[30] Donoho, D.L. and I.M. Johnstone, “Adapting to Unknown Smoothness via Wavelet Shrinkage”, *Technical Report 425,* Dept. of Statistics, Stanford University, June 1993.

[31] Donoho, D. L. “De-Noising via Soft Thresholding”, *Technical Report 409*, Dept. of Statistics, Stanford University, Nov. 1992

[32] Donoho, D.L. and I.M. Johnstone, “Ideal Spatial Adaptation via Wavelet Shrinkage”, *Technical Report 400*, Dept. of Statistics, Stanford University, July 1992.

[33] Donoho, D. L. and I.M. Johnstone, “Minimax Estimation via Wavelet Shrinkage”, *Technical Report 402*, Dept. of Statistics, Stanford University, July 1992.

[11] C-COR Electronics Inc., *FT-295X Series Feedforward Trunk and Bridgers Stations Instruction Manual*, Rev. 1, Oct. 1988.

[12] C-COR Electronics Inc., *Instruction Manual for Status Monitoring System*, Rev. 0, May 1984.

[13] C-COR Electronics Inc., *Preliminary Instruction Manual for the B-507 Remote Bridger*, Rev. 0, July 1983.

[14] Castro, J.L. and M. Delgado, "Fuzzy Systems with Defuzzification are Universal Approximators", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 26, No. 1, pp. 149-152, February, 1996.

[15] Chorafas, D.N. and H. Stienmann, *Intelligent Networks: Telecommunications Solutions for the 1990's*, CRC Press Inc., Boca Raton, Florida, 1990.

[16] Chow, E.Y. and A.S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems", *IEEE Transactions on Automated Control*, Vol. 29, pp. 603-614, 1984.

[17] Chow, M., R.N. Sharpe, and J.C. Hung, "On the Application and Design of Artificial Neural Networks for Motor Fault Detection - Part I", *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 2, pp. 181-188, April 1993.

[18] Chow, M., R.N. Sharpe, and J.C. Hung, "On the Application and Design of Artificial Neural Networks for Motor Fault Detection - Part II", *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 2, pp. 189-196, April 1993.

[19] Chow, M. and S.O. Yee, "Methodology for On-line Incipient Fault Detection in Single-Phase Squirrel-Cage Induction Motors using Artificial Neural Networks", *IEEE Transactions on Energy Conversion*, Vol. 6, No. 3, pp. 536-545, Sept. 1991.

[20] Chui, C.K., *An Introduction to Wavelets*, Academic Press, Inc. San Diego, California, 1992.

[21] Claasen, T.A.C.M. and A. Jongepier, "Model for the Power Spectra Density of Quantization Noise", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-29, Aug. 1981, pp. 914-917.

Bibliography

[1] Antoniou, A., *Digital Filters: Analysis, Design and Applications*, McGraw-Hill Inc., Princeton, New Jersey, 1993.

[2] Aubrun, C., S. Suater, H. Noura, and M. Robert, "Fault Diagnosis and Reconfiguration of Systems using Fuzzy Logic: Application to a Thermal Plant", *International Journal of System Sciences*, Vol. 24, No. 10, pp. 1945-1954, 1993.

[3] Bendat, Julius D. and Allan G. Piersol, *Random Data: Analysis and Measurement Procedures*, John Wiley & Sons Inc., New York, 1986.

[4] Bennett, W.R., "Spectra of Quantized Signals", *Bell System Technical Journal*, Vol. 27, July, 1948, pp. 446-472.

[5] Berger, J., R.R. Coifmann, and M. Goldberg, "Removing Noise from Music using Local Trigonometric Bases and Wavelet Packets", *Journal of the Audio Engineering Society*, Vol. 42, 1994.

[6] Bernieri, A., G. Betta, A. Pietrosanto, and C. Sansone, "A Neural Network Approach to Instrument Fault Detection and Isolation", *IEEE Transactions on Instrumentation and Measurement*, Vol. 44, No. 3, pp.747-750, June 1995.

[7] Bernieri, A., M. D'Apuzzo, L. Sansone, and M. Savastano, "A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems", *IEEE Transactions on Instrumentation and Measurement*, Vol. 43, No. 6, pp- 867-873, December,1994.

[8] Bezdek, J.C., "A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms", in *Fuzzy Models for Pattern Recognition*, J.C Bezdek and S.K. Pal, Editors, IEEE Press, Piscataway, New Jersey, pp. 130-137, 1992.

[9] Bezdek, J.C., R.J. Hathaway, M.J. Sabin, and W.T. Tucker, "Convergence Theory for Fuzzy c-Means: Counter Examples and Repairs", in *Fuzzy Models for Pattern Recognition*, J.C Bezdek and S.K. Pal, Editors, IEEE Press, Piscataway, New Jersey, pp. 138-142, 1992.

[10] Burt, P.J., and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on Communications*, Vol. COM-31, April 1983, pp. 532-540.

- Novel fault modalities occur.

Since the solutions to the early fault detection problem presented in this work did not violate any of this assumptions, these techniques therefore represent general fault detection approaches which are suitable to a wide range of large scale engineering plants.

8.2 Future Work

The theoretical areas which were touched on in this work which should be areas of future research are as follows:

- A theoretical analysis of the effects of low order quantizers on noise contaminated signals should be undertaken to determine the theoretically optimal underlying signal estimate and an associated optimal de-noising methodology.

- A theoretical methodology of setting the recurrent neural network training and fault detection thresholds should be developed.

- A theoretical analysis of the transition area statistics should be undertaken so that the optimal analytical fuzzy membership function can be identified.

The practical areas which were touched on in this work and should be areas of future work are as follows:

- The fuzzy membership function based fault detection approach should be implemented as a fully operational real-time fault detection system.

- The wavelet de-noising methodology that was presented should also be re-cast as a real-time system.

- Larger scale trials of both the fuzzy membership function and recurrent neural network based fault detection approaches should be undertaken in which both systems operate in real-time. Ideally, these trails would be undertaken in an environment in which the true faults and/or repair information regarding the engineering plant were available.

The major contribution of this work to the fields of fault detection, wavelet applications, and neural networks were therefore:

- The development of a novel fuzzy membership function based fault detection system for coarsely quantized status data signals exhibiting unknown linear dependencies.
- The development of a wavelet based methodology of obtaining an estimate of the underlying sensor signal for noise contaminated signals which have undergone coarse quantization.
- The development of a heuristic methodology of setting the training and fault detection thresholds for a recurrent neural network based fault detection system.
- The statistical analysis of the performance of the fuzzy membership function based fault detection system on data from an operational real-world large scale engineering plant.
- The statistical analysis of the performance of the recurrent neural network based fault detection system on data from an operational real-world large scale engineering plant.

The major significance of these contributions is that prior to this work no generally applicable approach to fault detection in large scale engineering plants was available. In addition, fault detection methodologies tend not to be evaluated on operational data sets. This is particularly true within the setting of large scale engineering plants; hence, it is also significant that the proposed detection methodologies have been tested and shown to be viable within a real-world setting.

It is important to note that the novel fault detection techniques which were developed in this work are general approaches. These approaches were designed under the set of assumptions which were presented in Chapter 1. Most notably these assumptions included that,

- The number and type of fault modalities experienced by the plant are unknown.
- The plant components act with unknown non-linearities and dynamics.
- Analytical models of the plant and/or its components are unknown or unavailable.
- Little or no symbolic information or experiential knowledge regarding the operation/behaviour of the plant is available.

Chapter 8:

Conclusions and Future Work

8.0 Introduction

This chapter will present the conclusions of this work and outline the contributions of the work to the fields of fault detection, wavelet applications, and neural networks research. The chapter will also indicate areas which were touched on this work which should be the subject of further research.

8.1 Conclusions

This work began with the basic problem of how to improve on the fault detection capabilities of traditional limit checking in such a way that the resulting systems would be amenable to the task of early fault detection and would be able to be retrofitted into existing industrial status monitoring systems, particularly those designed to employ limit checking. The first step in this process was to identify three typical classes of signal dependencies to be found within the status data signals, namely independent signals, linearly dependent signals, and non-linearly dynamically dependent signals.

For the linearly dependent systems, a novel fuzzy membership function based fault detection system was developed which was able to better track the primary signal dependency and which allowed the threshold bounds to be set in accordance to a prescribed probability of false alarm. For the non-linearly, dynamically dependent signals, a recurrent neural network based fault detection system was developed which was able to accurately track the primary signal dependency. For this recurrent neural network approach a heuristic methodology of setting the threshold bounds was developed centred on a wavelet based approach of estimating the underlying sensor signal from the coarsely quantized raw status data signals. Both of these proposed fault detection methodologies were evaluated on real-world data obtained from and operational large scale engineering plant. In both cases, these proposed fault detection techniques were shown to have fault detection properties superior to those of the traditional limit checking approach.

7.4 Conclusions

In this chapter the results obtained from utilizing the fuzzy membership function and recurrent neural network based fault detection systems were compared to the results obtained from an operation limit checking fault detection system. In both cases, the proposed fault detection systems were better able to model the known dependencies present in the given status data signal classes. This lead to the proposed systems being able to utilize considerably tighter detection thresholds, while maintaining lower probabilities of false alarm and higher fault detection sensitivities than the *in situ* limit checking system. Both fault detection systems were also shown to be amenable to real-time fault detection tasks.

It is important to note that these result were produced for real data sets obtained from an operational system. No *a priori* assumptions about the engineering plant in question were incorporated into the modelling techniques, nor were any assumptions made about the possible fault modalities which may have been present in the system. The results presented in this chapter, therefore, validate the conclusion that the proposed fault detection techniques provide a generic methodology, under the applicable signal classes, for performing fault detection tasks in large scale engineering plants under the assumptions that were presented within Chapter 1.

The relaxation or tightening of the parameters which govern what is considered an acceptable behavioural model such that more or less correlation fluctuations are permitted requires knowledge of what levels of cross correlation are important in a fault detection sense for a given plant. Hence, these adjustments must be tailored to specific plants and require knowledge of what consisted important events for the given plant.

7.3.2.3 Average Bound Widths

Unlike the fuzzy membership function based technique and limit checking, the recurrent neural network detection system utilized a constant threshold bound width for all of the amplifiers which were modelled. In particular, the detection threshold was set, in accordance with the work of Section 6.3, to 0.2 dBmV. In comparing this width to that employed by the *in situ* limit checking system it is important to remember that this threshold is utilized as a threshold on the absolute value of the difference signal between the recurrent neural network's output signal and the de-noised wavelet forward pilot signal estimate. Hence, the threshold value of 0.2 dBmV must be doubled to account for the absolute value operation. In actuality, therefore, the recurrent neural network detection technique utilizes a 0.4 dBmV threshold which is on the order of 10 times tighter than the average detection threshold employed by the *in situ* limit checking system.

It should be noted that the recurrent neural network bound widths that were utilized in this comparison are very tight. Likely in an operational system these bounds would be relaxed, particularly if the subtle behavioural changes, which were seen to be detectable by the system in Section 7.3.2.2.2, were shown not to be related to important or critical changes in the given component's operation. Tight bounds were utilized in the comparison to demonstrate the fault detection sensitivity present in the recurrent neural network detection system and to confirm that even with this tight threshold setting that the detection technique did not produce any false alarms, in the sense that all the flagged events occurred for explainable reasons. Hence, even with this tight threshold setting the proposed detection methodology generated accurate event flags.

in the forward pilot signal were negatively correlated with changes in the temperature were shown to been associated with abnormal amplifier behaviour. Figure 7-16 shows a comparison, for one of the example amplifiers, of the locations of the areas in which the neural network was unable to learn with the areas where the cross-correlation fluctuated from positive to negative. It can be clearly seen from this figure that these two sets of areas coincided and this result was true in general for all the amplifiers which had areas which were not able to be learned by the neural network.

It should be noted that although the large middle area shown in Figure 7-16, appears that the area contains exclusively negative cross correlations, small areas of positive correlation are present within this area. As was previously stated, negative cross correlations are known to be associated with abnormal amplifier behaviours. The normal behaviour of the amplifiers is to have their forward pilot signals positively correlated with the associated temperature signal. Hence, areas of purely negative correlation tend to be interspersed with small areas of positive correlation which leads to the neural network's failure to learn. This is what occurs within the middle area shown in Figure 7-16. The scale of the plot though does not permit these small areas of positive cross correlation to be easily identified within the figure.

It should be noted that the recurrent neural network is able to learn areas which are either purely positively correlated or purely negatively correlated. What it is incapable of learning are areas which contain both positive and negative correlations. The neural network detection system's sensitivity to the areas of fluctuating correlation can be controlled to some extent by varying the parameters which govern what is considered an acceptable behavioural model. If the 0.1 dBmV threshold is increased or the 90% requirement is relaxed then the detection system will become more tolerant to fluctuations in the cross correlation. The exact nature of how these adjustments should be made to apply the recurrent neural network detection system to a given plant have not been explored and have been left as an area of future work. Nor has any work been done on what levels of fluctuation in the cross correlation are significant in a fault detection sense to the example plant under consideration in this chapter.

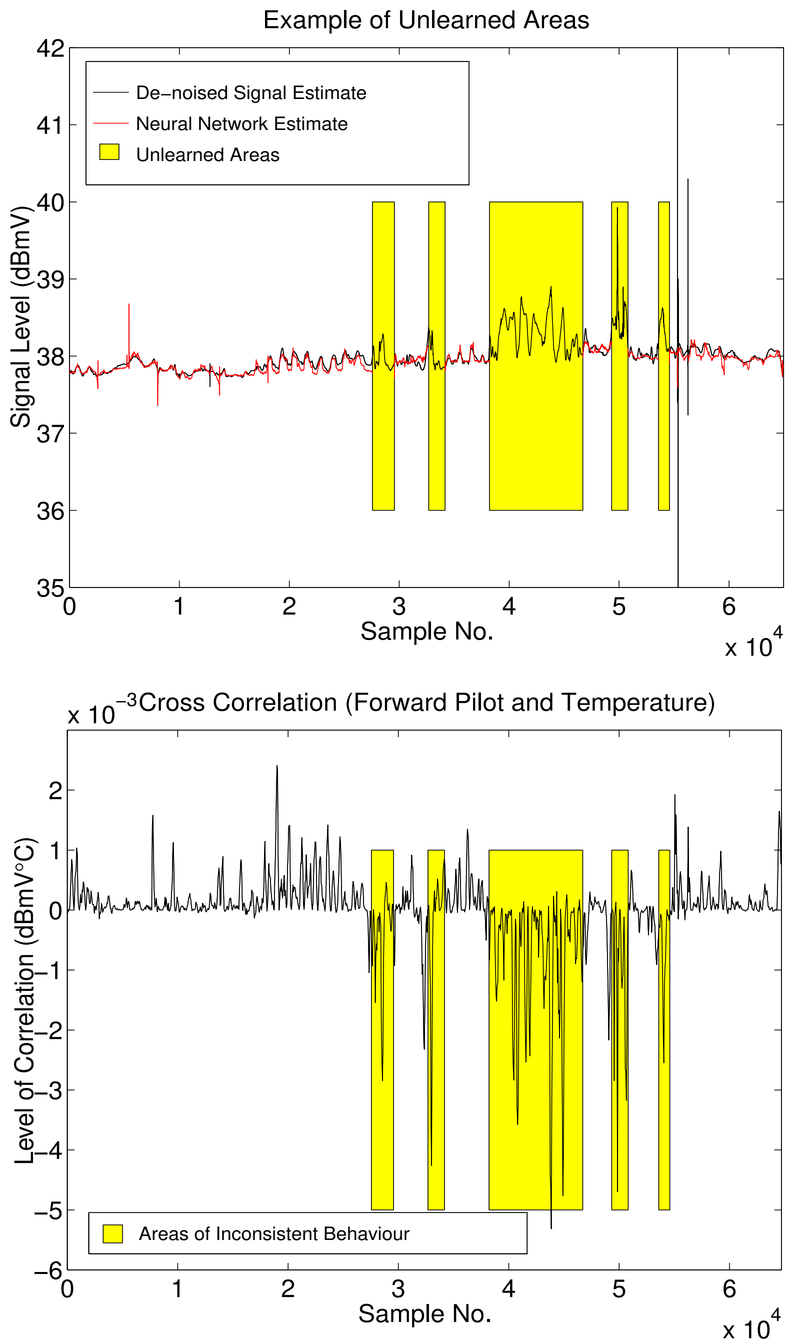


FIGURE 7-16.
Comparison of areas in which the neural network was unable to learn and areas of fluctuating cross correlation between the de-noised forward pilot and enclosure temperature status data signals for one of the example amplifiers.

It should be noted that, in general, tightening of the *in situ* limit checking system's bounds would not have improved the resulting comparison. The recurrent neural network detection system is able to model the signal dependency, hence its threshold bounds are not located at fixed levels in their time domain representations. The bounds move up and down as they track the depend on the temperature. The limit checking bounds, on the other hand, are fixed within the time domain. Hence, they cannot track the drifts in the forward pilot signal level. Tightening the bound, therefore, would have just resulting in an increase in the system's false alarm rate. It would not change the ability of the limit checking system to accurately identify and located subtle behavioural changes within the forward pilot status data signals.

7.3.2.2.3 Areas of Inconsistent Behaviour

The remaining 152 flagged events were caused by the neural network being unable to learn the behaviour pattern present in the given 3000 sample data record during the given retraining event. The neural network requires that a consistent behaviour be present in the training data record if it is to be able to learn the behaviour to the prescribed degree of accuracy. If the forward pilot signal's behaviour is not consistent in the training data record then the neural network will fail to learn. This in itself is an important event, in that it can be used to identify areas containing inconsistent behavioural patterns and these areas have a higher probability of being associated with fault conditions.

Obviously, if the areas where the neural network is unable to retrain are to be used as indicators of areas of interest, in a fault detection sense, then means of confirming that these unlearned areas are always associated with inconsistent behavioural patterns are required. To this end, the cross correlation between the de-noised forward pilot and enclosure temperature signals were measured using the correlation detector which was developed in [57]. Of particular interest, is the identification of areas in where the cross correlation between the de-noised forward pilot and enclosure temperature status data signals fluctuated from positive to negative. As was discussed in Chapter 3, changes in the forward pilot signal level are typically positively correlated with changes in the enclosure temperature signal, When the temperature goes up the forward pilot level rises. When the temperature goes down the forward pilot signal level drops. In [67], areas where changes

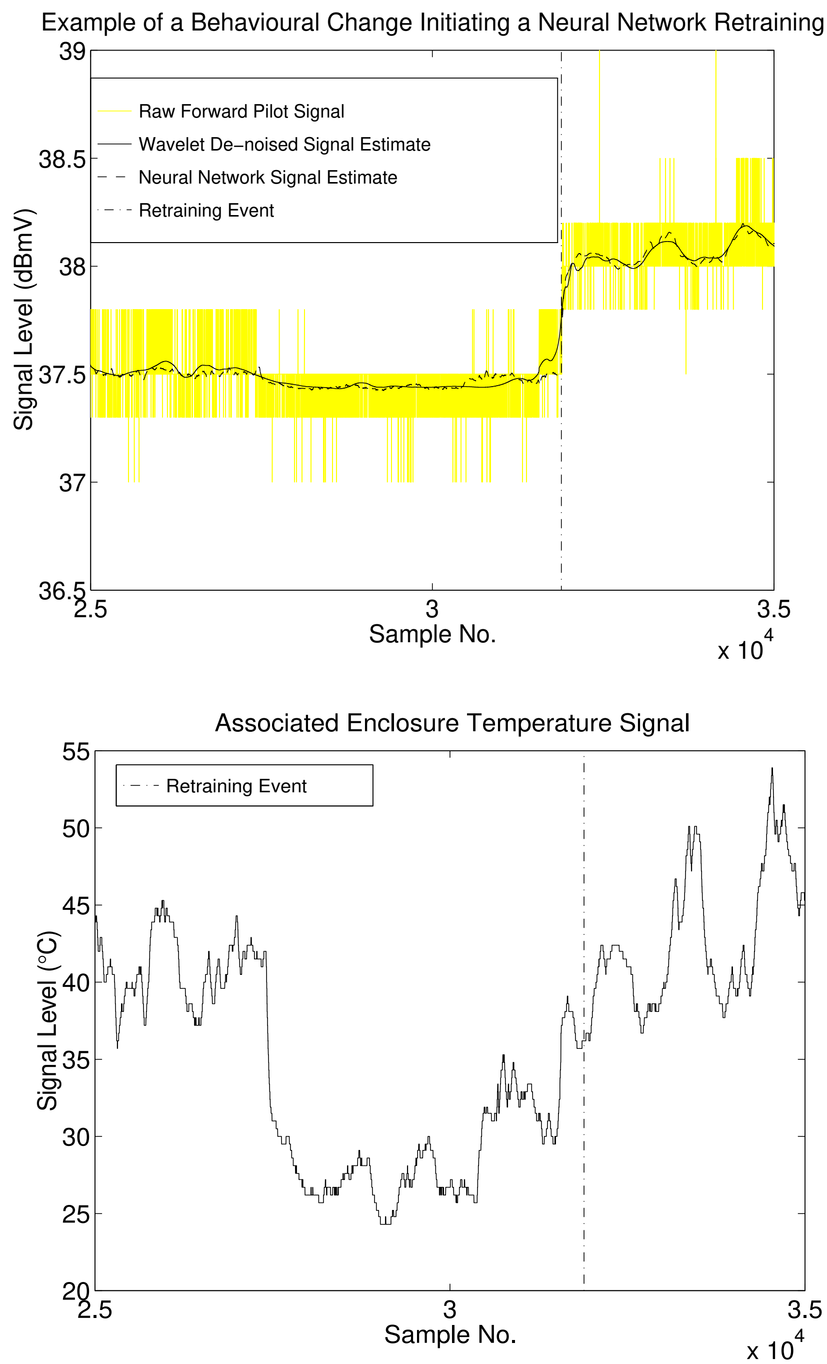


FIGURE 7-15.
Example behavioural change event which caused a neural network retraining event to be initiated.

7.3.2.2.2 Behavioural Changes

Of the remaining 298 flagged events, 146 were caused by behavioural changes with the de-noised forward pilot status data signal. Figure 7-15 shows an example of one of these behavioural changes and the resulting recurrent neural network retraining events which was initiated. In this case, the enclosure temperature signal started off in the range of 35 to 40 ˚C, it then dropped into the range of 20 to 30 ˚C and this drop was associated with a slight drop in the forward pilot signal level, as to expected. When the enclosure temperature signal subsequently returned to the 35 to 40 ˚C range though, the forward pilot signal level did not return to its previous level. In fact the average forward pilot level within this region rose by approximately 2 quantization levels above its previously held levels for temperatures within this range. Obviously, this indicated that the behaviour of the amplifier has changed. For the same temperature range, the amplifier no longer produces the same forward pilot signal levels. The recurrent neural network fault detection system was able to accurately identify this behavioural change at the point it occurred and initiate a neural network retraining event. The retrained neural network was then able to continue the tracking of the forward pilot signal under the new behavioural model.

It is important to note the sensitivity of the recurrent neural network fault detection system against that of the *in situ* limit checking system. The recurrent neural network was able to identify that a behavioural change had occurred when there was only a 1 quantization step difference between the de-noised wavelet forward pilot signal estimate and the recurrent neural network signal estimate. As was seen in Section 7.3.1.3, the *in situ* limit checking system would have, on average, required the behavioural change to have caused a 10 quantization step offset before it would have been able to generate and event flag, provided of course that the upper and lower thresholds were centred appropriately about the forward pilot status data signal. Hence, the behavioural change shown in Figure 7-15 which was identified by the recurrent neural network detection system went undetected by the *in situ* limit checking system. This result is typical of the 146 behavioural change events identified by the recurrent neural network detection system. In each case the recurrent neural network detection system was able to identify and accurately temporally locate subtle behavioural changes which went unnoticed by the *in situ* limit checking system.

not to be related to fault conditions within the example plant then steps could be taken to adjust the recurrent neural network detection system such that these events were not flagged and, hence, did not produced re-training events.

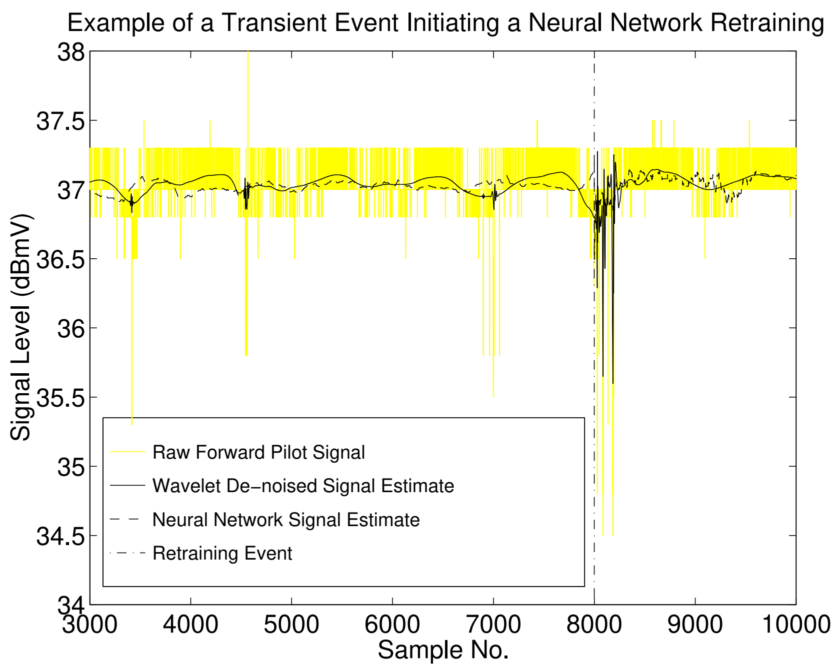


FIGURE 7-14.
A typical example of a transient event which caused a neural network retraining event to be initiated.

In particular, one method of addressing this problem would be to continue modeling the de-noised forward pilot signal with the original neural network model once a re-training event has occurred. If after a short time, the de-noised forward pilot resumes its original behaviour then it can be accurately assumed that a transient event has occurred. A decision can then be made as to whether or not the transient event should be reported. Since, in the example plant's case, 3000 data samples must be collected before a re-training event can be initiated, continuing to model the system with the original neural network would not place an extra burden on the system, and in fact it may allow some of retraining events to be eliminate which would otherwise be required.

7.3.2.2 Alarm Events

Over the set of 45 data records the recurrent neural network fault detection system produced a total of 324 event flags[9], which is considerably more than the 0 flags generated over the same data set by the *in situ* limit checking system[10]. These flags were attributable to three distinct causes within the de-noised forward pilot status data signals, namely transient events, behavioural changes, and areas of inconsistent behaviour over which the recurrent neural network was unable to re-train. Because the *in situ* limit checking fault detection system utilized much wider fault detection bounds (to be seen in Section 7.3.2.3), it did not detect any of these event within the given forward pilot status data records. The next three sections will look at each of these types of flagged events in turn in order to provide confirmation that there are no events which were flagged by the recurrent neural network detection systems which did not have a known cause.

7.3.2.2.1 Transient Events

Of the 324 event flagged by the recurrent neural network detection system, 26 were due to transient events within the de-noised forward pilot status data signals. Figure 7-14 shows an example of one of these transient events and the resulting retraining of the recurrent neural network. These transient events represent "spikes" on the raw forward pilot status data signals which are large enough to pass through the wavelet de-noising process. In accordance with the work of [35], these "spikes" must also be long enough in duration to cause the neural network model to exceed the detection threshold for more than 5 samples. A single, isolated "spike" event will not cause the neural network detection system to initiated a neural network retraining process. At this time, the nature of these transient events and their relationship to fault conditions within the example engineering plant are unknown. Hence, it is reasonable at this point for the recurrent neural network detection system to flag these events as behavioural anomalies. Should these transient events prove

---

9. Under the assumption that one event flag would be generated for each 3000 sample data set which over which the recurrent neural network was unable to train.

10. It should be noted, that the 0 generated event flags represent just the event flags which were generated over the 65,356 length sample records. The *in situ* system may have generated event flags before or after these samples. It just did not generate any event flags over the data records randomly selected for this trial.

7.3.2.1 Evaluation Process

Due to the computational cost of the recurrent neural network modelling approach, it was necessary to test the validity of the detection approach on a much smaller test set than was utilized for the fuzzy membership function based fault detection system. Specifically, the recurred neural network fault detection system was tested on a set of 45 randomly chosen data records each of which were 65,356 samples in length (corresponding to approximately 2 months of data). For each of these data sets, the recurrent neural network detection system was utilized in the same manner as proposed in Chapter 6.

Specifically, for each data set the given recurrent neural network was trained on the first 3000 samples of the given temperature and wavelet de-noised forward pilot status data samples. The neural network was then utilized as a free-running model of the given forward pilot signal. If the temperature signal exceeded the training temperature range then a temperature re-training event was initiated which caused the behavioural model to be augmented to include the observed forward pilot behaviour within the new temperature range, as per the work of [35]. If on the other hand, the recurrent neural network forward pilot signal estimate exceeded the wavelet de-noised forward pilot signal by more than the 0.2 dBmV threshold (discussed in Section 6.3) then an event flag was generated and the next 3000 data samples were used to re-train the recurrent neural network to the new behavioural model. If, for either re-training case, the recurrent neural network was unable to successfully model the forward pilot signals behaviour then the recurrent neural network skipped ahead 3000 samples to the next block of data and the re-training process was re-tried. Successful re-training is defined in this context, in accordance with Section 6.2.2.1, as more than 90% of the modeled training samples being with 0.1 dBmV of their corresponding sample from the wavelet de-noised signal. The recurrent neural network system was, therefore, tested in the exactly same manner as it would be utilized within an operational system (i.e. with the required re-training methodologies in place).

limit checking system over the 13 month period. Obviously, this represents a significant limit on the early fault detection abilities of the limit checking system, particularly for fault conditions which exhibit a relatively slow onset. For fault conditions that result in a sudden, large scale change in the signal level the limit checking system would have still been able to detect the fault condition relatively quickly. What it would have had trouble detecting, and what is difficult to detect reliably with limit checking systems in general, are those fault conditions whose onset was more subtle. In particular, fault conditions which resulted in a gradual change in the signal level or signal behaviour would have likely gone undetected until their severity had increased significantly enough such that the limit checking bounds were exceeded.

7.3.2 Recurrent Neural Network Modeling Results

Within this section the results of applying the recurrent neural network fault detection systems, presented in Chapters 5 and 6, to the example plant's forward pilot status data signals will be presented. This section will begin with a discussion of the evaluation methodology which was employed in comparing the results of the recurrent neural network based detection system with those of the *in situ* limit checking system. Unlike the fuzzy membership function based system, though, a statistical comparison of the events flagged by the two systems will not be presented. As will be subsequently seen, the recurrent neural network detection system is much more sensitive to behavioural changes in the forward pilot status data signal than the *in situ* limit checking system. Hence, the recurrent neural network based system tends to produce more event flags than the limit checking system. For this reason, the majority of the analysis of the recurrent neural network detection system results will focus on validating that the event flags produced by the system occur for known reasons and are not attributable to false alarms. The section will conclude by comparing the average bound widths employed by the recurrent neural network based technique and the *in situ* limit checking system for the forward pilot status data signals. In this way, as discussed in the introduction, it will be shown that the recurrent neural network detection system is the superior detection system for the modeling of the forward pilot status data signal since it utilizes tighter threshold bounds and it has a higher fault detection sensitivity.

It should be noted once again that, in general, there is no correlation between the class membership of a given amplifier obtained through analyzing a particular status data signal and the membership classes assigned through the analysis of the other status data signals. Additionally, as in the current draw signal case, the classification of an amplifier as being “well” behaved or “poorly” behaved is only within the context of the analyzing the limit checking results and status data statistics over the historical data sets. It does not guarantee that an amplifier classed as “well” behaved did not undergo a behavioural change nor is the classification methodology which was utilized applicable to real-time.

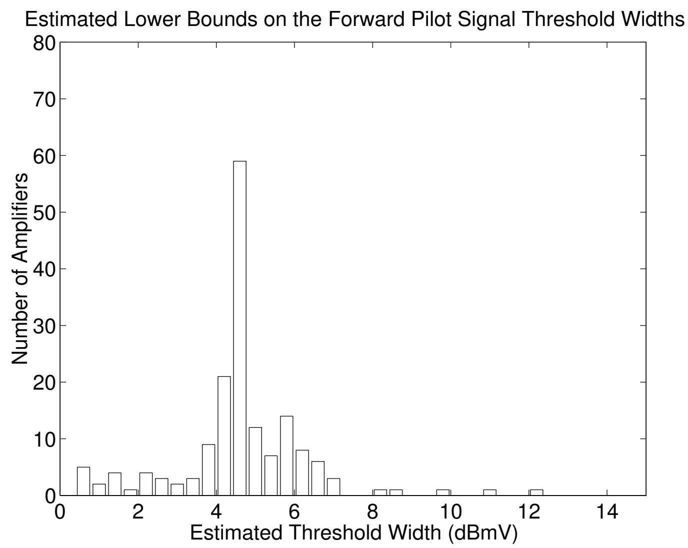


FIGURE 7-13.
Lower bound estimates of the utilized forward pilot signals’ limit checking threshold widths.

7.3.1.3 Estimated Bound Widths

As in the current draw signal case, the lower bounds on the employed threshold widths for the forward pilot signals can be estimated from the 13 month data set. The histogram of these estimated lower bounds is shown in Figure 7-13. Within the forward pilot signal, the average quantization step size was in the range of 0.20 dBmV. The bounds estimates in Figure 7-13, therefore, relate to average bounds widths typically greater than 20 quantization steps (corresponding to bound widths of greater than 4 dBmV) were employed by the

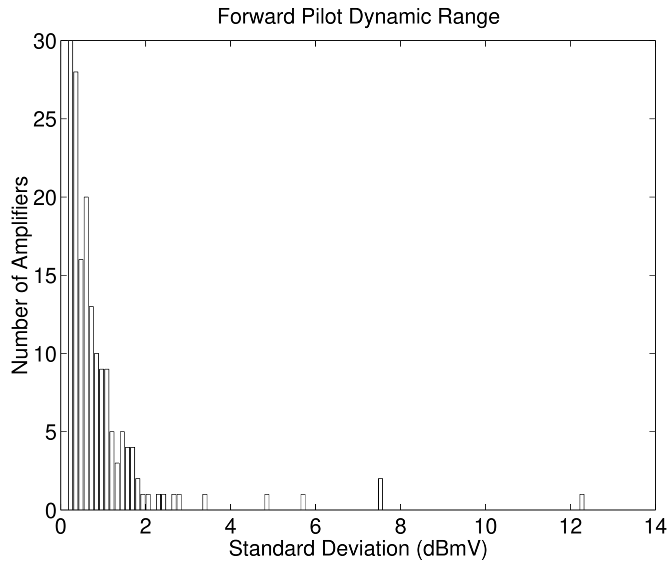


FIGURE 7-12.
Dynamic ranges of the forward pilot signals from the 170 selected amplifiers.

7.3.1.2 Dynamic Range

Figure 7-12 shows a plot of the dynamic ranges of the forward pilot signals for the 170 selected amplifiers as measured in terms of the standard deviation of their forward pilot signals across the full 13 month period. Once again, as in the current draw signal case, there is no clear separation of the "well" behaved and "poorly" behaved amplifiers within this feature space. Hence, the same procedure used in the current draw signal case was also utilized to separate the forward pilot signal behaviours of the amplifiers. In this case, 16 of the 170 amplifiers were classed as being "well" behaved throughout the 13 month period and these amplifier accounted for 582 of the 2,534,327 forward pilot limit checking flags produced during this period. It was also determined that 9 of the 170 amplifiers were within the "poorly" behaving class for this 13 month period and that these 9 amplifiers accounted for 864,868 of the 2,534,327 fault flags which were generated by the limit checking fault detection system for the forward pilot status data signal. The remaining 145 amplifier fell somewhere between the "well" behaved and "poorly" behaved amplifier classes over the given 13 month period and accounted for the remaining 1,668,877 of 2,534,327 generated fault flags.

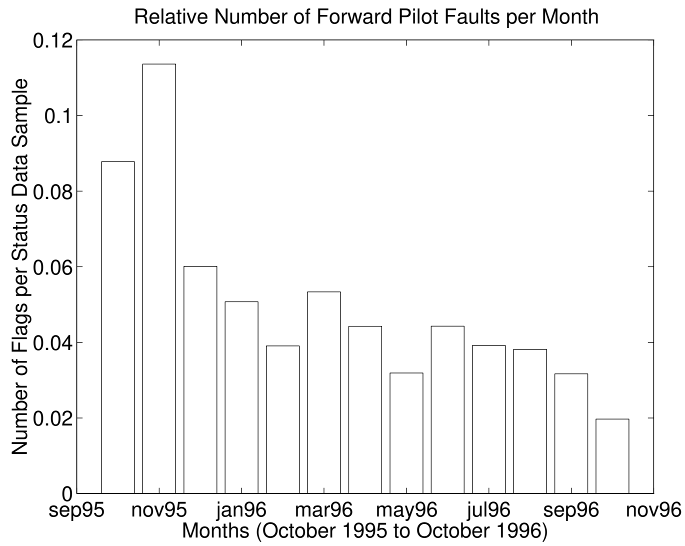


FIGURE 7-11.
Number of limit checking fault flags per data sample which were produced for the forward pilot signal across 170 amplifiers from the 13 month data set.

Table 7.11: Distribution of limit checking forward pilot flags across the 170 amplifiers.

Amplifier Classes Based on the Number of Fault Flags	**Number of Amplifiers in the Class**	**Total Number of Fault Flags Generated by Limit Checking for Amplifiers of the Given Class**
# of Fault Flags ≤ 1	1 of 170	1 of 2,534,327
1 < # of Fault Flags ≤ 50	25 of 170	474 of 2,534,327
50 < # of Fault Flags ≤ 1,000	64 of 170	18,309 of 2,534,327
1,000 < # of Fault Flags ≤ 10,000	35 of 170	132,800 of 2,534,327
# of Fault Flags > 10,000	45 of 170	2,382,743 of 2,534,327

selected amplifiers. Unlike the current draw signal though, a large set of “fault-free” amplifiers does not exist. All the amplifiers generate some fault flags and there is a fairly uniform distribution of amplifiers within the various fault flag classes.

7.3 Forward Pilot Status Data Signals

Within this section, the fault detection results obtained from the *in situ* limit checking system will be compared to the results obtained from the recurrent neural network based detection system for the forward pilot status data signals. As in the previous section the full example data set will be used in the evaluation of the limit checking fault detection system; but, due to computational demands the performance of the recurrent neural network modeling approach will be evaluated over a smaller, randomly selected data set.

7.3.1 Limit Checking Results

Within the example plant, the forward pilot status data signal's primary dependency can be modelled as a non-linear, dynamic dependency on the ambient temperature within the given amplifier, as measured via the enclosure temperature status data signal. This section will quantify the number and nature of the forward pilot field flags that were produced by the *in situ* limit checking system and provide estimates of the threshold bound widths that were employed for these signals during the 13 month period. Based on this analysis, the amplifiers under consideration will be separated into the classes of being "well" behaved, "poorly" behaved, or in between (as discussed in Section 7.1), based on the correlation between their dynamic ranges and the number of fault flags that they generate.

7.3.1.1 Alarm Events

During the 13 months, the limit checking fault detection system generated a total of 2,534,327 forward pilot status data fault flags for the 170 selected amplifiers. This volume of fault flags comprises approximately 4.62% of the total number of forward pilot data samples collected over this period. Figure 7-11 shows the normalized number of faults occurring within each of the 13 months. Unlike, the current draw signals, the forward pilot signals do not show a strong seasonal variation in the number of fault events occurring per month.

In Table 7.11, the amplifiers are divided into relative groupings based on the number of forward pilot fault flags that were generated for the particular amplifier across the data set. Once again it is quite clear from this table that the majority of the limit checking flags for the forward pilot signal are related to the behaviours of a relatively small set of the 170

progresses the amplifier may be slowly drifting into a new behavioural pattern. This drift will cause the transition areas to become "smeared" over the course of the month. The failure of the partial models in these cases, therefore may possibly be an indication of the onset of a fault condition within the affected amplifiers. This drifting effect would be identifiable since if it was the cause of the wide transition areas, then over any small subset of the one month data set the transition areas would always appear quite narrow. Over the full month though these narrow transition areas would coalesce to form the wide transition areas which cause the broad boundaries to be generated the full month long behavioural models.

Alternatively, the given amplifiers may just have abnormally broad transition areas. This would be the case if the broad transition areas also appeared throughout the smaller data sets. In general, though this condition would likely lead to smaller differences in the observed probability of false alarm than in the previous case. It is possible, within the example plant, that the presence of these abnormally broad transition areas may be useful as an indication of the onset of fault conditions. This possibility though would require further research to confirm.

Despite these problems, the partial models were able, in a large number of the cases, to adequately model the full month behavioural of the amplifiers, within the temperature range over which the partial models were constructed. This provides confirmation that the fuzzy membership function based fault detection system is amenable to real-time fault detection tasks. Once again, though, it should be noted that the above validation process does not represent a complete solution to the issues concerning the real-time implementation of the fuzzy membership function based fault detection system. The choice of selecting 10 days data from which to build the partial models was an arbitrary one and it is quite clear from the results that this choice may not be optimal for all of the amplifiers. Some may require more data from which to build the partial behavioural models and some may require less. The exploration of how such choices should be made and the resulting full implementation of a real-time fuzzy membership function based fault detection system has been left as an area of future work. The work presented in Chapter 4 though does represent the core theoretical work required for such a real-time implementation.

Table 7.10: Observed probabilities of false alarms for the partial behavioural models which were constructed from the first 10 days of data from the full month data sets.

Observed Probability of False Alarm, *P*	**Linear Regression**	**Fixed Point Linear Regression**	**Weighted Linear Regression**
$P \leq 0.001\%$	263 of 443	250 of 443	263 of 443
$0.001\% < P \leq 0.01\%$	32 of 443	36 of 443	32 of 443
$0.01\% < P \leq 0.1\%$	57 of 443	56 of 443	57 of 443
$0.1\% < P \leq 1.0\%$	49 of 443	51 of 443	47 of 443
$1.0\% < P \leq 10.0\%$	39 of 443	46 of 443	41 of 443
$P > 10.0\%$	3 of 443	4 of 443	3 of 443

the observed probability of false alarm. For these amplifiers the smaller data sets do not represent the amplifiers' behaviours as completely as the full data sets, but the partial models are still able to provided reasonable estimations of the amplifiers' behaviour over the duration of the month.

For the remaining amplifiers, those with $P > 0.1\%$, there is a significant increase in the observed probability of false alarm when the partial models are utilized. For these amplifiers this increase is due to the fact that 10 day data sets fail to provide an adequate representation of the behavioural maps transition areas. Typically, the transition areas for these amplifiers are abnormally wide. When these transition areas are viewed over the smaller data sets, they appear to be narrower than they actually are when viewed over the full one month data sets. This leads to the fuzzy membership function based approach underestimating the boundaries of the membership functions. It generates boundaries in the partial models that are significantly steeper than those generated in the full models. This has the effect of causing tighter thresholds to be generated, which in turn cause more event flags to be generated over the full month long data sets, leading to the higher observed probabilities of false alarm.

Once possible reason for the occurrence of these wide transition areas may be because the behaviour of the amplifier is drifting during the course of the month. Early in the month the amplifier may be exhibiting a particular behavioural pattern, as the month

A total of 2,210 one month data sets existed from the 170 example plant amplifiers over the October 1995 to October 1996 time period. Of these data records, 794 produced 0 event flags when modelled over the full month by the fuzzy membership based approach. Of these 794 one month data sets, 443 had enough transitions within the first 10 days of data to allow the partial behavioural models to be constructed. It was for these 443 data sets that the comparison between the fault detection capabilities of the full and partial behavioural models was performed.

For each of these data sets, an initial behavioural model was constructed from the full one month data set under the 0.001% probability of false alarm criterion. Partial behavioural models were then constructed utilizing just the first 10 days data from each of these month long data sets under the same 0.001% probability of false alarm criterion. These partial behavioural models were then utilized to model the given amplifier's behaviour over full month long data set, though only in the temperature range from which the partial model was developed. The event detection results obtained from these partial models were then compared to the results obtained from the full behavioural models.

In general, the utilization of the partial models should cause a measurable increase in the false alarm rate, given by Eq. 4.12, and the degree of this increase can be used to gauge the applicability of the modelling approach to real-time fault detection. If, for the majority of the cases, this increase in the false alarm probability is small then it can be claimed that the 10 day data sets can be used to generated behavioural models which can accurately model the given amplifier's full month behaviour, over the subset of temperatures from which the 10 day models were generated. The results of this comparison, for the three threshold bound generation methodologies, are show in Table 7.10.

From this table, it is quite clear that for more than half of the data sets under consideration the 10 day behavioural models proved to be sufficiently capable of modelling the full month long behaviour within the temperature range over which the partial models were generated. For these behavioural models the reduction in the data set from which they are built results in no change in the observed probability of false alarm. The observed probability of false alarm remains at its initial 0.001% setting. For approximately one quarter of the amplifiers ($0.001\% < P \leq 0.1\%$), the reduction in the data set causes a slight increase in

the *in situ* limit checking system and simultaneously produces significantly fewer event flags. As with the number of generated fault flags, there is no significant difference in terms of the utilized threshold bound widths between the three bound generation methodologies which would indicate that one should be employed preferentially.

7.2.2.4 Real-Time Applicability

What was not confirmed in the preceding section was whether or not the hypothesis, developed in Chapter 4, that the fuzzy membership function based fault detection approach is applicable to real-time fault detection is in fact true. The presentation to this point has been framed within a batch processing context with full month data sets being used in the development of the behavioural models. Within a real-time context, it is desirable to be able to utilize small data sets within the model development phase and then to utilize these models to perform fault detection on larger data sets. Hence, to confirm that hypothesis that the fuzzy technique can be applied to a real-time setting it is necessary to validate the assumption that behavioural models constructed over a short time period are able to provide adequate modeling of data obtained over longer time spans.

To test the applicability of the fuzzy based approach, behavioural models were generated from the first 10 days data from selected one month data sets. The behavioural modelling capabilities of these partial models were then compared to those of the full behavioural models constructed over the full one month data sets. Obviously, if the modeling results obtained from 10 day data sets are to be compared to the full month data set results then it is desirable to choose month long data sets which are "well" behaved under the fuzzy membership function based approach (i.e. the data sets which produced 0 event flags in the work of Section 7.2.2.2). Additionally, it is important to choose data sets which are likely to have enough transitions (greater than 2) within the 10 day data sets to allow the current implementation of the fuzzy based approach to be able to build the required behavioural models[8].

---

8. As was mentioned previously in Chapter 4, the current implementation of the fuzzy membership function approach automatically defaults to a limit checking approach if less than 3 quantization steps are present in the given data set.

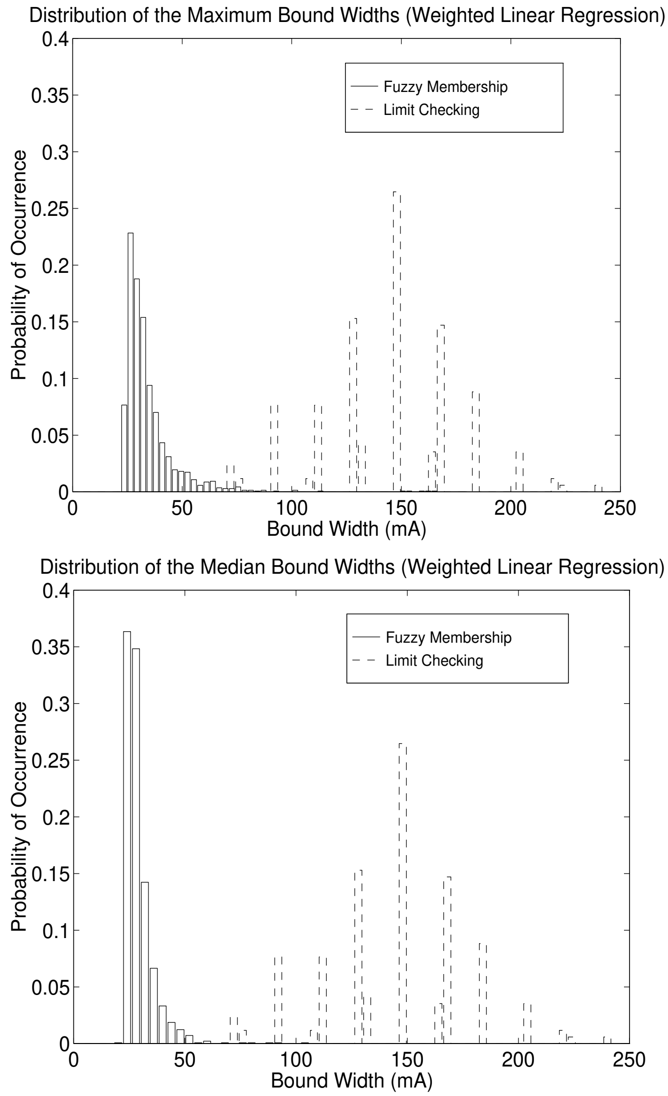


FIGURE 7-10.
Distributions of the maximum and median bounds widths generated by the fuzzy membership function based modeling technique (Fixed Point Linear Regression).

with different slopes). What is particularly noticeable from these figures is that, for each of the maximum bound width histograms, the fuzzy based modeling technique, utilizes bounds that are on the order of twice as tight as those used in the *in situ* limit checking system, which were shown in Figure 7-4. These plots confirm that the fuzzy membership function based modeling techniques are better at tracking the known temperature dependency of the example plant's current draw signal since it both utilizes tighter bounds than

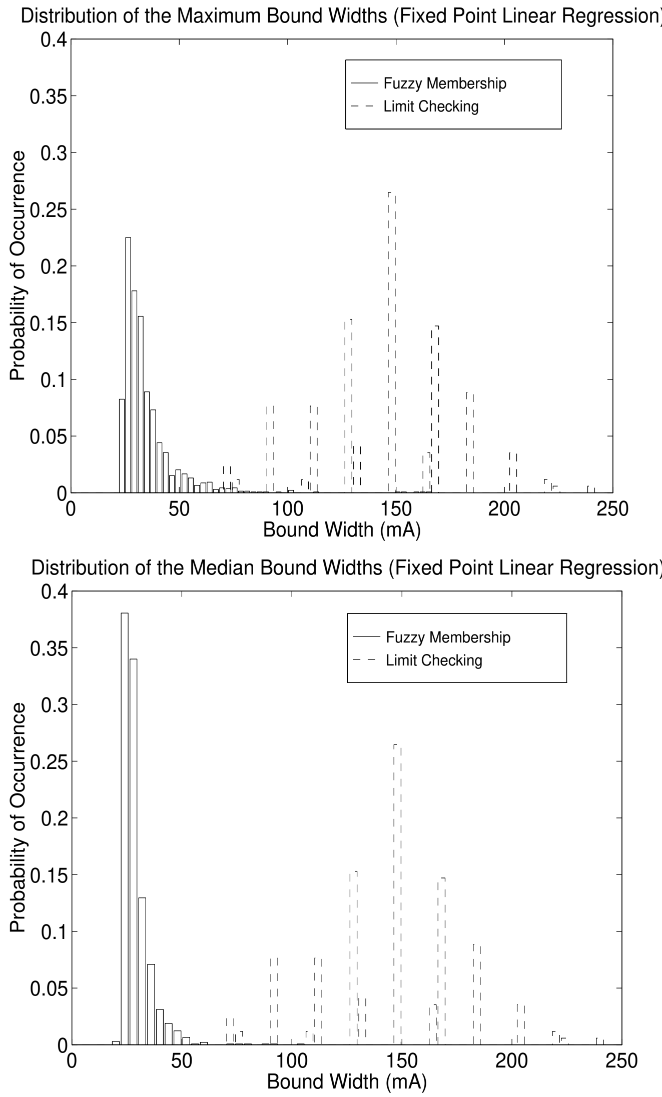


FIGURE 7-9.
Distributions of the maximum and median bounds widths generated by the fuzzy membership function based modeling technique (Fixed Point Linear Regression).

each of the three bound generation approaches for the set of amplifiers which were able to be modelled, where the histograms were constructed by combining the monthly data obtained across the full 13 months. Both the maximum and median measures were used to characterize the bound widths since, unlike in the limit checking case, the bounds produced by the fuzzy membership function based technique do not maintain a constant relative distance to one another (due to the possibility of having linear bounding functions

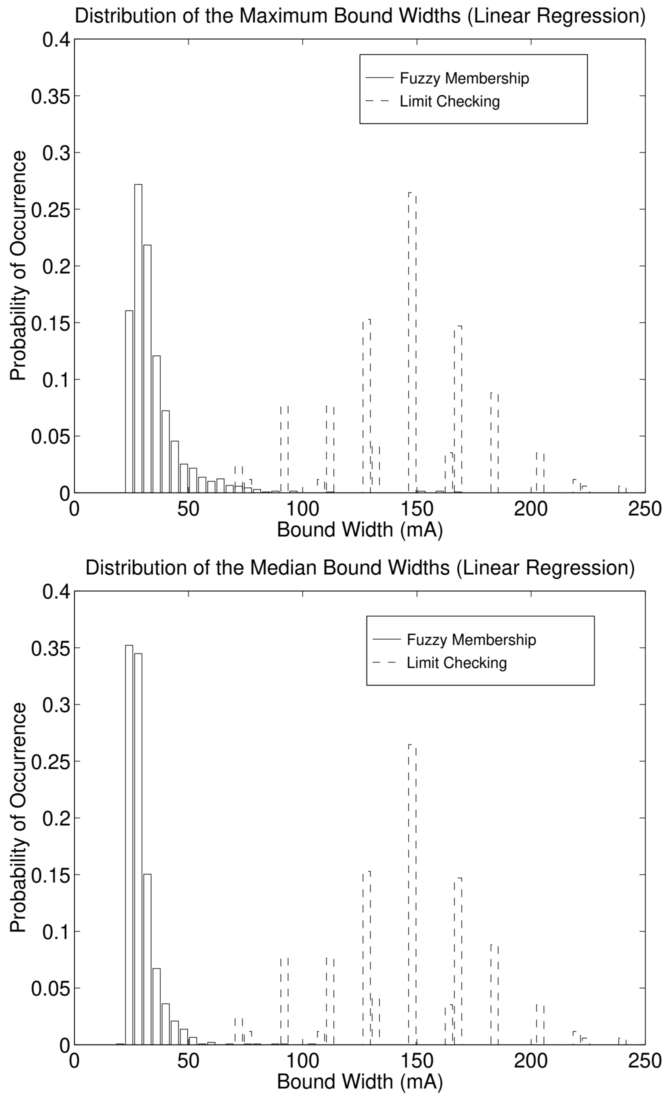


FIGURE 7-8.
Distributions of the maximum and median bounds widths generated by the fuzzy membership function based modeling technique (Linear Regression).

7.2.2.3 Average Bound Widths

Clearly, just generating fewer fault flags is not sufficient for a fault detection system to be comparably better than an alternate system. It is also necessary that the fault detection system with the fewer fault flags also utilizes tighter bounds in its monitoring of the allowed status data signal behaviour. In Figure 7-8, Figure 7-9, and Figure 7-10 histograms are shown detailing the monthly maximum and median bound widths utilized by

ferent than that of a real-time version of the fuzzy detection system detecting a behavioural change. In this case, the anomaly is within the training data and, hence, it gets inherently incorporated into the fuzzy membership functions themselves. Once again, it should be noted that the raw behavioural maps utilized in the behavioural model generation have been filtered to remove samples which were not consistent with assumed "staircase" shape. This was done to allow behavioural models to be constructed on data sets which were not purely "well" behaved. It is for these data sets that the fuzzy membership function based detection system produces the large number of event flags.

It is important to note that these problems are caused by limitations within the implementation of the fuzzy membership function based fault detection system, particularly with regards to how "well" behaved training areas were obtained, and not by theoretical limitations of the fuzzy modeling approach itself. Most of these problems can be mitigated by treating each of the component fuzzy membership functions independently and not by assuming that they possess a prescribed degree of similarity. It should also be noted that, despite these known limitations of the fuzzy membership function based detection system as it has been implemented, still produces fewer fault flags than the *in situ* limit checking system. Improving the implementation to account for these slight problems would only improve the ability of the fuzzy membership function based modeling technique to model the example plant's known current draw signal dependency.

A comparison of the results obtained under each of the three threshold generation methodologies also shows relatively close correspondence between the techniques both in terms of the number of flags generated and the classification of the amplifiers. In particular, the number of amplifiers within each class (Tables 7.7 to 7.7) is very similar across all three bound generation techniques. This provides further support to the conclusion reached in Chapter 4 regarding the relative insensitivity of the fuzzy detection approach to the actual bound generation methodology employed.

For the amplifiers in the classes with greater than 50 fault flags per month, the fault flags likely tend to be caused by the given amplifier's behavioural map deviating from the ideal "staircase" shape. In particular, three main assumptions with regards to the "staircase" have been made within the implementation of the fuzzy membership function based fault detection system. First, as in the work of Chapter 4, it has been assumed that each of the "steps" on the staircase would be close to the same width. This property will be termed the symmetry of the staircase. If the given amplifier's current versus temperature behavioural map is not symmetric, then the linear bounds that will be produced by the implementation will be excessively tight, in that the complete behaviour of the amplifier over the training data set will not be encompassed within the bounds, typically, resulting in the generation of a significant number of fault flags.

Secondly, it was assumed that each of the behavioural map's transition areas would occur over approximately the same temperature period. For example, for a particular amplifier all the transitions areas may each span a period of approximately 5 degrees. If there is a significant variation in the width of the transition areas, then the implementation of the fuzzy membership function based detection will tend to generate excessively tight bounds which will cause fault flags to be generated for the part of the current signal which generated the widest transition area in the behavioural map.

Thirdly, and most obviously, the "staircase" behavioural map of Chapter 4 assumes that once a given current level has occurred on the behavioural map it will not re-occur latter in the map. In particular, a current level will occur in the behavioural map first within the transition area of the preceding current level. The current level will then ideally have a constant area which is followed by a second transition area where the current level increases to the next quantization step. Hence, at no point after a current level has passed through its second transition level or prior to its passing through its first transition level should it appear in the behavioural map. The implemented fuzzy membership function modeling technique will produce bounds which generate significant numbers of fault flags for amplifier which do not follow this assumption. This problem accounts for all of the cases where the implemented fuzzy detection systems generates 1,000 or more fault flags for a given amplifier over the training data set. It should be noted, that this problem is dif-

Table 7.9: Distribution of fault flags generated by utilizing the fuzzy membership function based modeling technique to create monthly behavioural models (Weighted Linear Regression).

Month	# of Fault Flags ≤ 1	1 < # Fault Flags ≤ 50	50 < # Fault Flags ≤ 1,000	1,000 < # Fault Flags ≤ 10,000	1,000 < # Fault Flags ≤ 10,000
June 1996	76	45	30	1	0
July 1996	82	45	29	1	0
August 1996	72	48	36	1	0
September 1996	86	40	30	1	0
October 1996	97	44	20	1	0

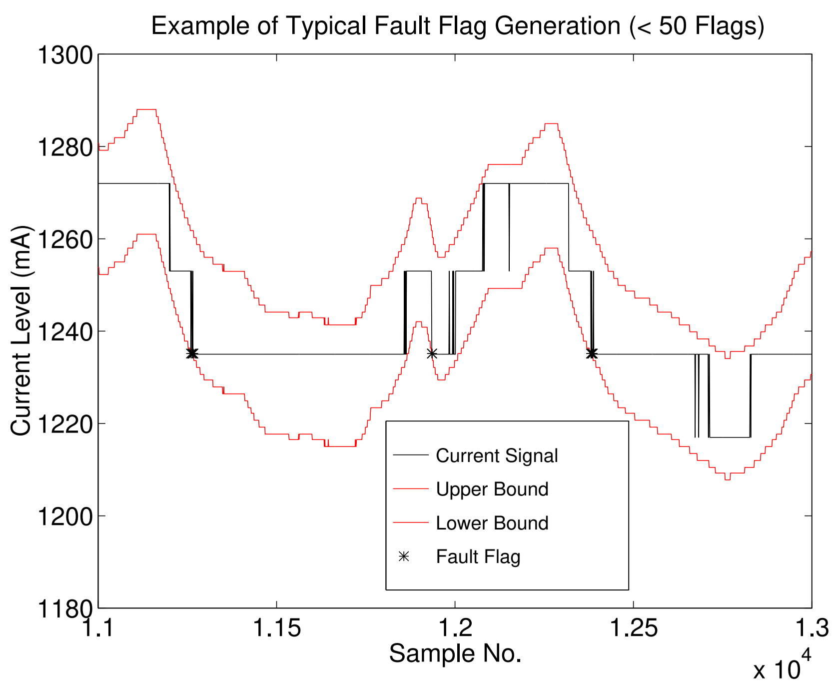


FIGURE 7-7.
Example showing the typical cause of fault flags for the amplifiers which generated less than 50 fault flags per month via the fuzzy membership function based fault detection technique.

Table 7.8: Distribution of fault flags generated by utilizing the fuzzy membership function based modeling technique to create monthly behavioural models (Fixed Point Linear Regression).

Month	# of Fault Flags ≤ 1	1 < # Fault Flags ≤ 50	50 < # Fault Flags ≤ 1,000	1,000 < # Fault Flags ≤ 10,000	1,000 < # Fault Flags ≤ 10,000
February 1996	82	54	24	0	0
March 1996	70	49	24	1	0
April 1996	89	38	29	1	1
May 1996	72	52	29	0	0
June 1996	77	39	35	1	0
July 1996	74	59	32	1	0
August 1996	70	50	36	1	0
September 1996	81	38	36	2	0
October 1996	92	46	23	1	0

Table 7.9: Distribution of fault flags generated by utilizing the fuzzy membership function based modeling technique to create monthly behavioural models (Weighted Linear Regression).

Month	# of Fault Flags ≤ 1	1 < # Fault Flags ≤ 50	50 < # Fault Flags ≤ 1,000	1,000 < # Fault Flags ≤ 10,000	1,000 < # Fault Flags ≤ 10,000
October 1995	0	3	110	1	0
November 1995	88	38	27	1	0
December 1995	70	51	27	1	0
January 1996	102	43	16	1	0
February 1996	94	45	26	0	0
March 1996	82	44	26	0	0
April 1996	88	41	27	1	1
May 1996	77	53	23	0	0

Table 7.7: Distribution of fault flags generated by utilizing the fuzzy membership function based modeling technique to create monthly behavioural models (Linear Regression).

Month	**# of Fault Flags ≤ 1**	**1 < # Fault Flags ≤ 50**	**50 < # Fault Flags ≤ 1,000**	**1,000 < # Fault Flags ≤ 10,000**	**1,000 < # Fault Flags ≤ 10,000**
October 1995	0	3	110	1	0
November 1995	88	40	26	0	0
December 1995	72	48	28	1	0
January 1996	102	43	16	1	0
February 1996	97	45	18	0	0
March 1996	83	45	24	0	0
April 1996	90	40	26	1	0
May 1996	79	52	22	0	0
June 1996	78	43	30	1	0
July 1996	82	46	28	1	0
August 1996	73	47	36	1	0
September 1996	87	39	30	1	0
October 1996	99	42	20	1	0

Table 7.8: Distribution of fault flags generated by utilizing the fuzzy membership function based modeling technique to create monthly behavioural models (Fixed Point Linear Regression).

Month	**# of Fault Flags ≤ 1**	**1 < # Fault Flags ≤ 50**	**50 < # Fault Flags ≤ 1,000**	**1,000 < # Fault Flags ≤ 10,000**	**1,000 < # Fault Flags ≤ 10,000**
October 1995	0	3	109	2	0
November 1995	89	38	27	0	0
December 1995	67	53	27	2	0
January 1996	94	49	18	0	0

In the October 1995 data set, there is an observable fault event which occurs throughout the plant (shown in Figure 7-6). This event is the main cause of the large discrepancy between the limit checking and fuzzy based techniques for the October 1995 data set. The event occurs plant wide, and it was not detected for most of the amplifiers by the *in situ* limit checking system, but it has been detected for most of the amplifiers by the fuzzy modeling techniques. Table 7.6 shows the total percentage reduction in the number of fault flags generated for the three fuzzy membership function based techniques both before and after the October 1995 data set is removed from the comparison.

The next question to be answered is the determination of where the fuzzy membership function based detection system's fault flags occurred. In particular, it is desirable to determine how the fault flags generated within any given month are distributed across the modeled amplifiers. Table 7.7, Table 7.7 and Table 7.7 show the monthly fault flags generated by the fuzzy membership function based modeling technique classified in accordance with the same fault classes utilized in Table 7.1. From these tables, it can clearly be seen that, for the majority of amplifiers within in any given month, discounting October 1995, less than 50 fault flags are generated for all three threshold generation approaches.

In general, for amplifiers within this less than 50 fault flag class during a given month, the fault flags tend to be generated primarily due to edge effects. As the temperature increases/decreases, a corresponding increase/decrease occurs in the current data signal. Due to the coarseness of the quantization of the current signal, though, the current transition will occur immediately. At times, the suddenness of the current transition is such that it is not tracked by the temperature based upper/lower bounds causing fault flags to be generated (an example of which is shown in Figure 7-7). Hence, these fault flags tend not to be related to actual behavioural changes, but instead are caused by the combined effect of the comparatively tight bounds[7] produced by setting the probability of false alarm to 0.001% and the coarseness of the current draw signals' quantization.

---

7. As will be seen in the next section, the threshold bounds produced by the fuzzy membership based approach are significantly tighter than those employed by the *in situ* limit checking fault detection system.

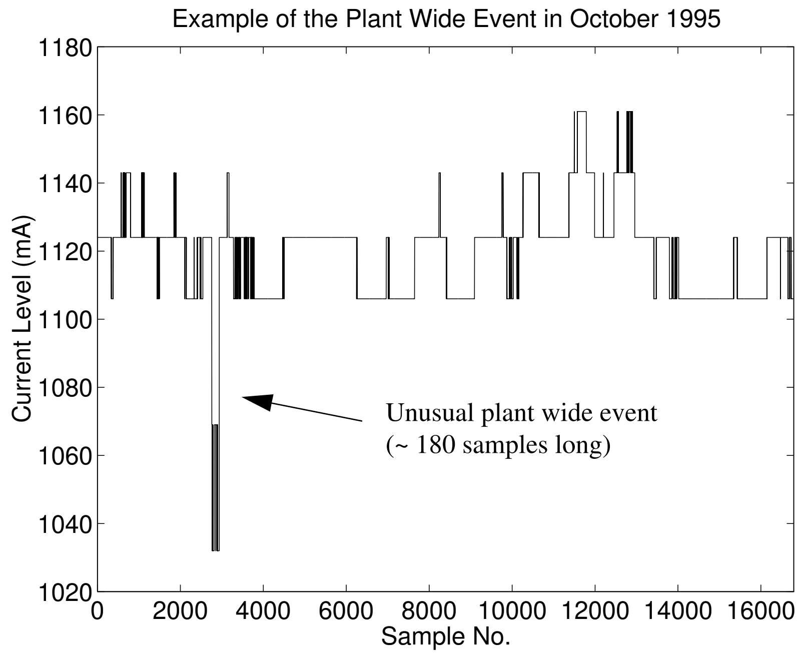


FIGURE 7-6.
Example of the plant wide event which occurred during October 1995.

Table 7.3, Table 7.4, and Table 7.5 show the comparison of the number of fault flags generated by the *in situ* limit checking against those produced by the fuzzy membership function based technique in turn for each of the three bound generation methodologies which were presented in Chapter 4. This comparison is done on a month by month basis for the set of amplifiers within each month which were able to be modeled by the fuzzy techniques. In each technique, for all of the months except October 1995, the utilization of the fuzzy membership based modeling greatly reduced the number of fault flags which were generated. Additionally, it should also be noted that the seasonal event flag trend present in the *in situ* limit checking results has been greatly reduced by the application of the fuzzy membership based detection system. In the *in situ* system's results there is a significantly higher number of fault events within months with temperature extremes. Within the fuzzy membership results, though, there is a relative balance between the number of events flags produced month to month. Hence, the fuzzy membership function based modeling approach has been able to capture these seasonal behavioural changes within the month by month behavioural models.

Table 7.5: Comparison of the number of fault flags generated on a month by month basis for the *is situ* limit checking fault detection system and the fuzzy membership function based fault detection system (Weighted Linear Regression).

Month	# of Fault Flags Generated by Limit Checking	# of Fault Flags Generated by Fuzzy Membership Function Based Modeling	% Reduction in # of Fault Flags Generated
October 1995	3,102	22,618	-629.14%
November 1995	21,763	5,490	74.78%
December 1995	122,503	7,295	94.05%
January 1996	73,069	5,019	93.13%
February 1996	141,131	5,179	96.33%
March 1996	91,408	7,058	92.28%
April 1996	24,609	8,008	67.46%
May 1996	26,669	5,888	77.92%
June 1996	39,444	8,163	79.30%
July 1996	93,371	9,006	90.35%
August 1996	101,055	10,357	89.75%
September 1996	39,450	10,760	72.72%
October 1996	8,219	5,871	28.57%
Total	**785,795**	**110,712**	**85.91%**

Table 7.6: Comparison of the reduction in the number of generated fault flags for the three threshold bound generation methodologies.

Threshold Generation Methodology	Reduction in No. of Fault Flags (with October 1995 data set)	Reduction in No. of Fault Flags (without October 1995 data set)
Linear Regression	86.25%	89.15%
Fixed Point Linear Regression	83.57%	86.71%
Weighted Linear Regression	85.91%	88.79%

Table 7.4: Comparison of the number of fault flags generated on a month by month basis for the *is situ* limit checking fault detection system and the fuzzy membership function based fault detection system (Fixed Point Linear Regression).

Month	# of Fault Flags Generated by Limit Checking	# of Fault Flags Generated by Fuzzy Membership Function Based Modeling	% Reduction in # of Fault Flags Generated
October 1995	3,102	24,652	-694.71%
November 1995	21,763	4,933	77.34%
December 1995	122,503	10,948	91.06%
January 1996	73,069	5,191	92.90%
February 1996	141,131	6,312	95.53%
March 1996	91,408	12,337	86.50%
April 1996	24,609	8,081	67.16%
May 1996	26,669	6,903	74.12%
June 1996	39,444	9,697	75.42%
July 1996	93,371	9,302	90.04%
August 1996	101,055	10,6041	90.04%
September 1996	39,450	13,666	65.36%
October 1996	8,219	6,457	21.44%
Total	**785,795**	**129,120**	**83.57%**

Table 7.3: Comparison of the number of fault flags generated on a month by month basis for the *is situ* limit checking fault detection system and the fuzzy membership function based fault detection system (Linear Regression).

Month	**# of Fault Flags Generated by Limit Checking**	**# of Fault Flags Generated by Fuzzy Membership Function Based Modeling**	**% Reduction in # of Fault Flags Generated**
October 1995	3,102	22,739	-633.04%
November 1995	21,763	4,442	79.59%
December 1995	122,503	8,039	93.44%
January 1996	73,069	5,026	93.12%
February 1996	141,131	4,790	96.61%
March 1996	91,408	6,389	93.01%
April 1996	24,609	7,705	68.69%
May 1996	26,669	4,975	81.35%
June 1996	39,444	8,124	79.40%
July 1996	93,371	8,954	90.41%
August 1996	101,055	10,511	89.60%
September 1996	39,450	11,035	72.03%
October 1996	8,219	5,286	35.69%
Total	**785,795**	**108,015**	**86.25%**

Table 7.2: Classification of the amplifiers' current draw signals based on their correspondence to the ideal "staircase" behavioural map.

Month	**# of Amplifiers with too few Transitions to be Modeled**	**# of Amplifiers with "Staircase" Shaped Behavioural Maps**	**# of Amplifiers with "non-Staircase" Shaped Behavioural Maps**
October 1995	6	114	50
November 1995	1	154	15
December 1995	2	149	19
January 1996	2	162	6
February 1996	2	160	8
March 1996	3	152	15
April 1996	1	157	12
May 1996	4	153	13
June 1996	1	152	17
July 1996	2	157	11
August 1996	2	157	11
September 1996	1	157	12
October 1996	3	162	5

draw data samples which were located two or more quantization steps away from the preceding current draw signal level. This filtering allowed fuzzy membership function based models to be constructed for data sets which were less than ideally behaved. This filtering also is the reason why the fuzzy membership function based modeling technique was able to generate event flags events over the training sets. The basic shape of the "staircase" is not modified by the filtering process; hence, this filtering does not affect the comparison of the fuzzy membership function based modeling technique with the *in situ* limit checking system.

The filtered raw behavioural maps were utilized in the generation of the behavioural models and the subsequent fault detection results which are presented in the tables in this section and the figures to be presented in Section 7.2.2.3.

approach to the *in situ* limit checking system can only be performed for the amplifiers which possess the expected “staircase” shaped behavioural map. It is, therefore, also important to verify that the expected behavioural map shape does occur for the majority of the 170 amplifiers during the 13 month period of interest. Table 7.2 shows the number of amplifiers which were, on a month by month basis, classed as belonging to each of these three behavioural map classes. From this table it is quite clear that the majority of the amplifiers do posses a “staircase” shaped behavioural maps in any given month, and hence are able to be modeled by the fuzzy membership function based technique.

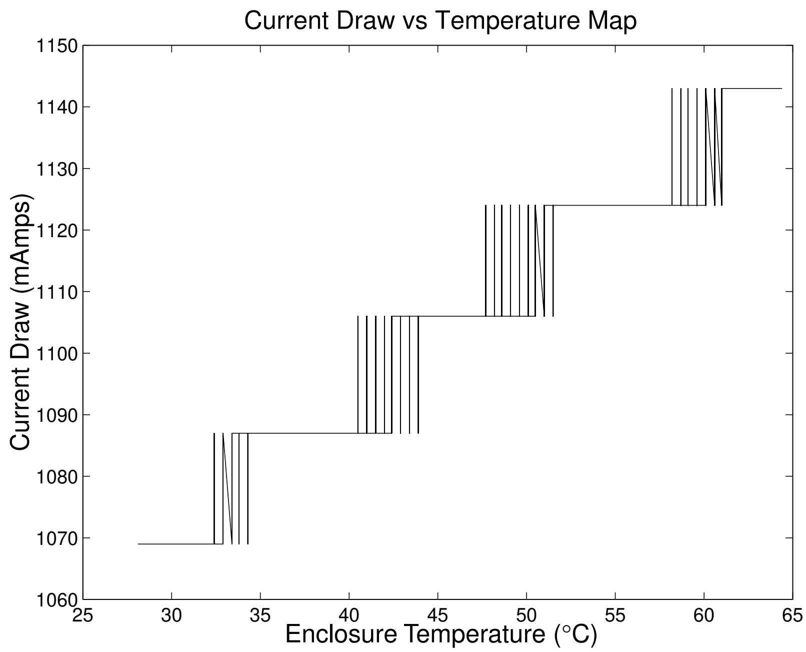


FIGURE 7-5.
Expected “staircase” shaped of the current draw signal versus enclosure temperature signal behavioural map.

It should be noted, that within the implementation of the fuzzy modeling technique, some filtering of the raw current versus temperature maps was undertaken to remove small deviations from behavioural maps. This was required due to the necessity of providing the fuzzy modeling techniques with “well” behaved data areas from which the behavioural model could be constructed. The filtering that was undertaken was to remove any current

that the resulting behavioural models are better able to track the known signal dependency. Conformation that the fuzzy membership based modeling approach is applicable to a real-time setting will be addressed in Section 7.2.2.4.

7.2.2.2 Alarm Events

The application of the fuzzy membership function based fault detection system to the modeling of the monthly current signals effectively results in partitioning the amplifiers within the monthly data sets into three distinct groups:

1. Those that have too few variations within a given month to permit the fuzzy membership function based behavioural map to be constructed.
2. Those that have atypical behaviour within the given month in that their behavioural maps do not follow the expected "staircase" shape, and hence are not able to be modeled by the fuzzy membership function based approach.[6]
3. And, those amplifiers whose behavioural maps follow the expected "staircase" shape and have enough monthly variation to be model permit a fuzzy membership based behavioural map to be constructed.

The membership based fault detection technique presented in Chapter 4 inherently assumes that the fuzzy membership functions are generated over areas of relatively "well" behaved data and, hence, that the current versus temperature map has the characteristic "staircase" form, discussed in Chapter 3 and reproduced for convenience in Figure 7-5. If, within any given monthly data set, there are less than 2 transition areas in the respective amplifier's current versus temperature map, then the fuzzy membership based modeling technique will be unable to construct an appropriate behavioural model for the given amplifier, and will revert to utilizing limit checking bounds.

Alternatively, the modeling technique may fail if the given amplifier exhibits a behavioural map other than the typical "staircase" map. In this case, the particular implementation of the fuzzy modeling approach will fail and generate erroneous linear bounding functions. Obviously, an evaluation of the fuzzy membership function based modeling

---

6. It should be noted, though, that despite the fact that these amplifier behaviours are not able to be modelled by the fuzzy membership function based approach, they will still be "flagged" by the detection system as abnormal behaviours. In essence, the failure to model can itself be used as an initial test of the acceptability of the amplifier's behaviour.

plant and the second that the such a fault detection system based on this modeling approach would have better fault detection characteristics than the traditional limit checking approach.

In the second part of the evaluation procedure, the fault detection capabilities of behavioural models constructed from the full month data sets were compared to those constructed data sets comprising just the initial 10 days data from the given month. In this way, the real-time applicability of the fuzzy membership function based fault detection system was analyzed. The purpose of this part of the evaluation was not to develop a full solution to the real-time implementation issues but instead to validate that a real-time implementation is feasible under the proposed approach. As this dissertation is primarily concerned with the development of suitable fault detection approaches, and not necessarily the direct requirements of their final implementation, the full implementation of a real-time fuzzy membership function based fault detection system was left as an area of future work.

Throughout the evaluation process for the fuzzy membership function based fault detection technique the desired probability of false alarm was set to 0.001%. This false alarm rate setting corresponds to an expected number of false alarms of only 548 over the entire 13 month data set. This corresponds to a rate of slightly more than one false current draw signal alarm per day across the full 170 amplifiers. The chosen false alarm probability of 0.001% is relatively arbitrary, but increasing the false alarm rate would result in an high number of false alarms which would likely be unacceptable within a real-world implementation. It will be shown that, even with this low probability of false alarm, the resulting fuzzy membership based fault detection system still utilizes bounds which are, on average, much tighter than the *in situ* limit checking system and simultaneously it generates much fewer event flags over the given example data set.

Once again, it should be noted this evaluation of the fuzzy membership based approach against the in situ limit checking system will only be used to confirm that the proposed modelling approach is generally applicable across a large statistical data set and

dure lies in the fact that the fuzzy membership based modeling technique has currently been implemented to model data records which contain a “reasonable” number of transition events (i.e greater than 2). If not enough transition events are present in a given data record then the system will not be able to generate a suitable model of the given amplifier’s current versus temperature behaviour. In particular, if too few transition areas are present in the given data record then the system will default to limit checking fault detection with the bounds set, in the case of the temperature versus current behaviour, to one and a half quantizations steps (36 mA) above and below the maximum and minimum current values occurring in the given data record.

Due to the relatively slow varying nature of the example plant’s current signals, data sets comprising less that one month of data tended not to have enough variations to allow the current implementation of the fuzzy modeling approach to build adequate current versus temperature maps. Hence, data sets of less than one month in duration were not considered within this large scale test of the fuzzy membership based fault detection approach. Time periods of larger than one month were also not considered in this part of the evaluation procedure due to the increase in the probability of behavioural changes as the size of the data records is expanded. The fuzzy membership based modeling approach inherently must have a “well” behaved section of data on which to “learn” the behavioural model. The presence of behavioural changes, therefore, could cause significant perturbations in the “learned” membership functions which would result in an inaccurate comparison of the approaches. Hence, the fuzzy models which were developed were based on one month data sets, with the complete data set, excluding those samples which were filtered out[5], being employed in the given model’s development.

This first step of the evaluation procedure was therefore employed to validate two specific assumptions. The first being that the fuzzy membership function based modeling approach presented in Chapter 4 was generally applicable and could be employed to model the majority of the current versus temperature behaviours present in the example

5. Some filtering of the data was required to remove those samples which caused the behavioural map to deviated from its “staircase” shape. The nature of this filtering process will be explained more fully in Section 7.2.2.2.

followed by a discussion of the threshold bound widths that were employed by the fuzzy membership based detection system, under each of the three bound generation methodologies presented in Chapter 4, for the amplifier data sets under consideration. These thresholds will then be compared to those employed by the limit checking system for the same amplifier set. It should be noted that each of these comparisons will be done only for the current draw status data signal as this is the only status data signal within the example plant whose behaviour belongs to the class of signals for which the fuzzy membership function based detection technique is directed, namely those exhibiting linear dependencies.

The comparison with of the fuzzy membership function based fault detection system modeling approach against that of the *in situ* limit checking approach will only be used to confirm that the fuzzy modeling approach is generally applicable and that it is better able to model the known signal dependency. This comparison though is not able to provide any indication of the real-time applicability of the fuzzy membership function based approach. Hence, within the final portion of this section the hypothesis that the fuzzy membership function based detection system presented in Chapter 4 is applicable to real-time fault detection tasks will be explored in more detail.

7.2.2.1 Evaluation Process

The evaluation process that was utilized in the evaluation of the fuzzy membership function based approach was a two stage process. First, the fuzzy membership based detection technique was compared to the *in situ* limit checking detection system by attempting to generate monthly fuzzy membership based models of the individual amplifiers current behaviours across the full set of 170 amplifiers over the full 13 month time period. For those amplifiers which were able to be modelled[4], the resulting models where then utilized to generate fault reports for the data sets upon which they had been generated. These fault reports were then compared with the fault reports generated by the *in situ* limit checking system over the same data sets. The rationale behind this evaluation proce-

---

[4] The nature of which amplifiers were able to be modelled under the fuzzy membership function based fault detection approach will be more fully discussed in Section 7.2.2.2.

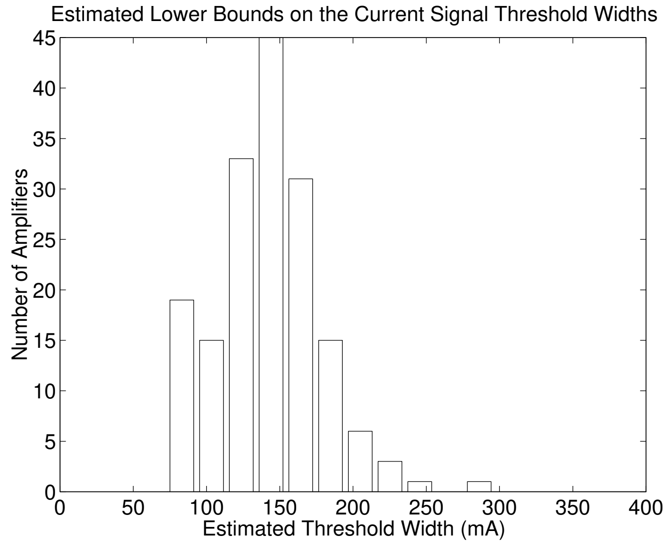


FIGURE 7-4.
Lower bound estimates of the utilized current draw signals' limit checking threshold widths.

mated threshold widths utilized for the current draw signal over the 13 month data set is presented. It is important to note that the data within this histogram represents a lower bound on the individual amplifier threshold widths. In practice, the utilized bound widths were likely larger than those estimated.

7.2.2 Fuzzy Logic Membership Function Based Modeling Results

Within this section the results of applying the fuzzy membership function based fault detection system, presented in Chapter 4, to the example set of current draw signals will be presented. Specifically, the pseudo-Gaussian analytical membership function will be used as the basis for the statistical representations and all three of the bound generation methods (linear regression, fixed point regression, and weighted regression) will be used to generate the linear thresholding functions. This section will begin by overviewing the evaluation process that was undertaken and discussing why the chosen evaluation procedure, and its resulting conclusions, are relevant and applicable to an operational, real-time fuzzy membership function based fault detection system. A statistical comparison of the fault flags generated by the two detection systems will then be undertaken. This will be

1. The upper and lower thresholds must each be exceeded at least once over the given data set.

2. The nature of the data must be such that for at least one flag generated by the upper/lower threshold the associated data value must be within one quantization step above/below the given threshold level.

3. The nature of the data must be such that non-flagged data values exist which are within one quantization step of the upper and lower thresholds.

4. The thresholds must have remained constant throughout the time period over which the given data set was generated.

Within the example data set, the first condition is known not to have been met by at least 130 of the 170 example amplifiers, namely those that have 1 or fewer fault flags throughout the 13 months. In addition, data records for the remaining 40 amplifiers may or may not have met conditions 2 and/or 3. It is quite possible that no data values occurred within one quantization step of the threshold for any given amplifier. If this was the case then the above threshold estimation methodology would have produced erroneous results. Finally, there is no guarantee, for any given amplifier, that the thresholds remained constant throughout the 13 month period. In fact, it can be reasonably assumed that it is fairly likely that the thresholds were widened for amplifiers exhibiting high apparent false alarm rates, as measured by the relative number of fault flags which they generated. With the available data, therefore, a precise, accurate estimation a given amplifier's threshold levels was difficult to obtain.

What was feasible though was to generate a lower bound estimate of the threshold widths that were employed for each of the amplifiers. In particular, the non-flagged portion of the current draw signal from a given amplifier were analyzed and the maximum and minimum data values were taken as the initial estimates of the threshold locations. The flagged data values were then analyzed and if there existed flagged values within the initial threshold window the thresholds were adjusted such that all the flagged values occurred externally to the window. Finally, to account for the fact that ideally the thresholds would be placed at the mid-point between quantization steps, the estimated thresholds were shifted outward by half of the median quantization step size occurring in the given amplifier's current draw status data signal. In Figure 7-4 a histogram of the esti-

fault detection system. Of the remaining amplifiers, 22 were classed as “poorly” behaved and these amplifiers accounted for 574,225 of the 817,281 limit checking fault flags generated during this period. The remaining 77 amplifiers have been assumed to have behaviours located somewhere between those of the “well” behaved and “poorly” behaved classes since they were not identified as clearly belonging to either the “well” behaved or the “poorly” behaved class.

It is important to note that these classifications are only with respect to the amplifiers' current draw signal behaviour. It is entirely possible that an amplifier may be within the “well” behaved class with respect to the current signal and be within the “poorly” behaved class with respect one of its other status data signals. Additionally, it should be stressed that the classification of an amplifier as being “well” behaved within the above context does not mean that the amplifier is guaranteed to be fault free. The amplifier may have undergone a behavioural change or behavioural changes which were not detected by the limit checking fault detection approach and which did not cause the amplifier to appear abnormal under the observed statistics. The above methodology only allows the amplifiers to be classed in accordance with their performance under limit checking over historical data sets. It is not a real-time classification methodology, nor is it particularly sensitive in detecting subtle behavioural changes.

7.2.1.3 Estimated Threshold Widths

Within the 13 month data set, there exists no direct information detailing the actual limit checking thresholds that were in use. Hence, no direct method was available to calculate the employed threshold widths. It is these widths, though, that give some information regarding the fault detection sensitivity of the given limit checking system. In theory, these widths can be estimated by analyzing the fault flags generated by the limit checking system. Specifically, the lower threshold could be determined as the mid-point between the lowest non-flagged data value and the next lower value and the upper threshold could be determined as the mid-point between the highest non-flagged data value and the next higher value. The accuracy of this estimation methodology though depends on four conditions being meet. Namely,

This is consistent with the known experiential knowledge, since due to the relative rarity of fault events within the example plant, the majority of the amplifiers should be “well” behaved within any given month. The “well” behaved amplifier can then be identified by using the standard Euclidean distance metric. For each of the amplifiers, the mean Euclidean distance between itself and the other 169 amplifier is measured in the feature space. Those amplifiers that have a mean distance which is less than the median distance for the set are then classed as forming part of the “well” behaved set for the given month. The median Euclidean distance being chosen as the thresholding level since, through experiential knowledge, it is known that fault events are relatively rare and that the majority of the amplifiers within a given plant will be “well” behaved through the majority of any given month. Hence, the amplifiers which are “poorly” behaved within a given month would have a high probability of having Euclidean distances which are above the median level obtained from the full set. The remainder of the “well” behaved set is then made up of those amplifiers with a standard deviation which is less than the minium standard deviation of any of the already selected amplifiers. These amplifiers represent the ones whose standard deviations and number of fault flags are so low as to remove them from the tightly grouped cluster of “well” behaved amplifiers since they are behaving on average even better than this group.

The above methodology allowed amplifiers within a given month to be classed as being “well” behaved or not. For the comparison of the detection systems, though, what was desired was to classify the amplifiers as being “well” or “poorly” behaved across the full 13 month data set. Ostensibly, classification across the full data set allows tight bounds to be placed on the amplifiers’ membership to either of these two classes. The data obtained from the above monthly classification process was used to provide classification across the full 13 month data set by assigning to the “well” behaved class all amplifiers which were “well” behaved in each of the 13 months, and assigning to the “poorly’ behaved class all amplifiers which were “well” behaved in no more than 1 of the 13 months. Under this classification scheme, 71 of the 170 amplifiers were classed in terms of their current draw signals as being “well” behaved over the 13 months. These amplifiers accounted for 6 of the 817,281 current draw fault flags generated by the limit checking

within the test set. Amplifiers were then classes as "well" behaved within a given month, with respect to a given status data signal, if their standard deviation for that month was less than or equal to the median standard deviation for the given month calculated across the full 170 amplifier test set.

The standard deviation measure by itself does not indicate whether or not the amplifier was within its allowable range. An amplifier may have had a current draw signal which had a small standard deviation during a particular time period, but the mean level of this signal may have been far from the amplifier's normal behavioural range. This problem can be overcome by incorporating the limit checking fault detection flags into the behavioural analysis. In particular, a feature space can be constructed from the tuples formed by the normalized number of limit checking fault flags and the normalized estimated standard deviations for each of the amplifiers on a month-by-month basis, as shown in Figure 7-3. The resulting feature space will have the "well" behaved amplifiers clustered within a fairly dense region with the "poorly" behaved amplifiers located at the periphery.

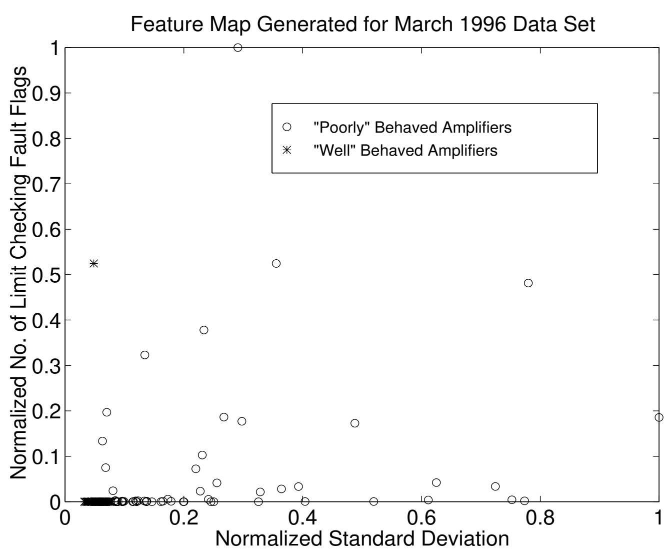


FIGURE 7-3.
Example feature space generated by normalized limit checking fault flags and monthly standard deviation estimate tuples.

7.2.1.2 Dynamic Range

Figure 7-2 shows a plot of the dynamic ranges of the current signals for the 170 selected amplifiers measured in terms of the standard deviations of their current signals across the full 13 month period. It is quite clear that the amplifiers do not separate cleanly within this feature space. In particular, it is known experientially that amplifiers with small standard deviations are more likely to be “well” behaved, while amplifiers with large standard deviations over a given time period are likely to be “poorly” behaved. Across the full 13 month data set though these differences become obscured because of the normal temperature induced variation of the signals due to seasonal temperature changes.

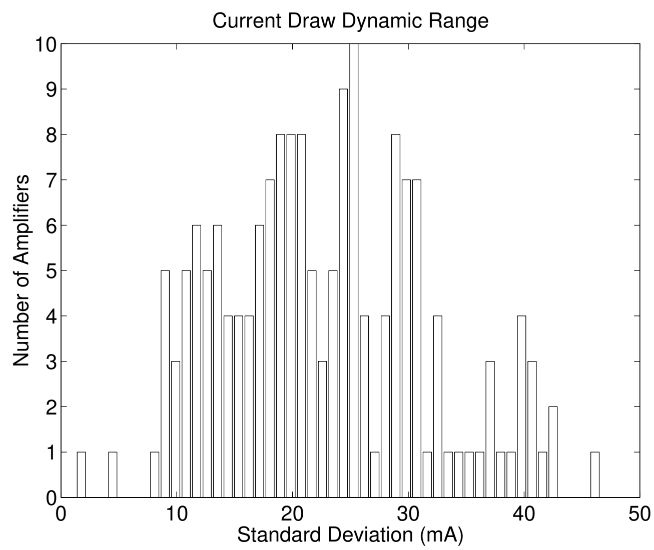


FIGURE 7-2.
Dynamic ranges of the current draw signals from the 170 selected amplifiers.

Month long data records, on the other hand, do not suffer significantly from these long term seasonal variations and still provide long enough data records to generate relatively good estimates of the signals’ standard deviations. Whereas, very short data records may be subject to localized variations in the amplifier’s behaviour which would give misleading estimates of the overall signal’s standard deviation. The approach taken within this work to classify the amplifiers according to their behaviours was therefore to generate the standard deviation estimates over each of the 13 months for each of the 170 amplifiers

In Table 7.1, the amplifiers are divided into relative groupings based on the number of fault flags generated for each amplifier across the data set. It is quite clear from this table that the majority of the limit checking current flags were related to the operation of a very small number of amplifiers.

Table 7.1: Distribution of limit checking current flags across the 170 amplifiers.

Amplifier Classes Based on the Number of Fault Flags	Number of Amplifiers in the Class	Total Number of Fault Flags Generated by Limit Checking for Amplifiers of the Given Class
# of Fault Flags ≤ 1	130 of 170	8 of 817,281
1 < # of Fault Flags ≤ 50	5 of 170	125 of 817,281
50 < # of Fault Flags ≤ 1,000	7 of 170	2,821 of 817,281
1,000 < # of Fault Flags ≤ 10,000	9 of 170	39,027 of 817,281
# of Fault Flags > 10,000	19 of 170	775,300 of 817,281

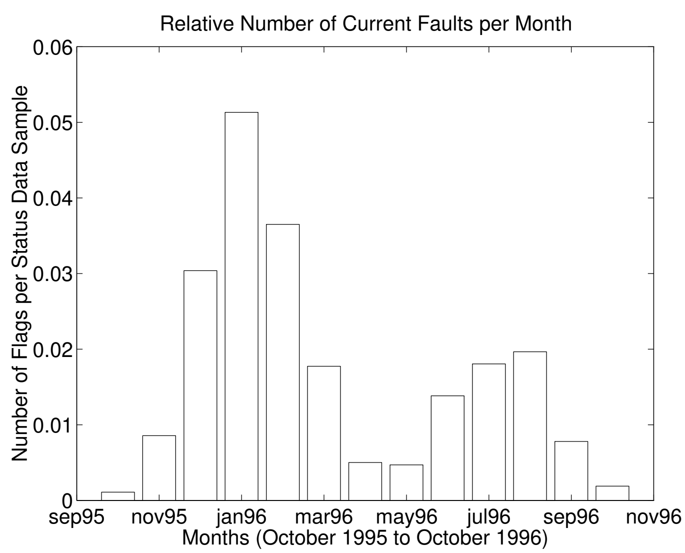


FIGURE 7-1.
Number of limit checking fault flags per data sample which were produced for the current draw signal across 170 amplifiers from the 13 month data set.

7.2 Current Draw Status Data Signals

The example plant had in place a limit checking fault detection system during the 13 months over which the status data set was collected. Within this section, the fault detection results obtained from this limit checking system will be compare to the results obtained from the fuzzy membership function based detection system over the same set of current draw status data signals.

7.2.1 Limit Checking Results

Within the example plant, the current draw status data signal's primary dependency can be modelled as a linear dependency on the ambient temperature of the given amplifier, measured via the enclosure temperature status data signal. This section will quantify the number and nature of the current field flags that were produced by the *in situ* limit checking status monitoring system and provide estimates of the threshold bound widths that were employed for the current signals during the 13 month period. Within the context of this analysis, the amplifiers under consideration will be separated into the three classes, as discussed in Section 7.1, based on the correlation between their dynamic ranges and the number of fault flags they generated.

7.2.1.1 Alarm Events

Within the 13 month period, the 170 amplifiers generated a total of 817,281 limit checking fault flags comprising approximately 1.49% of the total collected current draw status data samples. Figure 7-1 shows the number of faults occurring within each of the 13 months (October 1995 through to October 1996) normalized against the number of status data samples produced during each of the months. This figure clearly shows a seasonal trend, with the number of faults increasing fairly substantially during seasonal periods characterized by temperature extremes.

In terms of the amplifiers' dynamic ranges, it can be reasonably assumed that amplifiers which exhibited small dynamic ranges are less likely to have undergone significant behavioural changes and, hence, more likely to have been relatively fault free. Alternatively, amplifiers which exhibited wide dynamic ranges are more likely to have undergone significant behavioural changes and, hence, more likely to have been affected by one or more fault conditions. It is plausible, though, for amplifiers with small dynamic ranges to also have experienced fault conditions. In particular, this would be the case if the observed dynamic range was far from what is deemed to be the given amplifier's "normal" range. Hence, solely utilizing information about the amplifiers' dynamic ranges was insufficient to group them into behavioural classes.

Similarly, if a large number of fault flags were generated by the *in situ* limit checking detection system, the particular amplifier may have been subject to significant behavioural changes, provided the limit checking systems bounds were centred about the "normal" signal range. If the bounds were miss adjusted then a "well" behaved amplifier may have generated a significant number of fault flags even though its behaviour may not have changed significantly. Alternatively, if the bounds were set particularly wide, then an amplifier which underwent significant behavioural changes, resulting in a large dynamic range, may have produced relatively few limit checking fault flags. Hence, solely utilizing information about the amplifiers limit checking fault flags was also insufficient to classify their gross behaviours.

Therefore, without ground truthed data, a reasonable means of separating the amplifiers into behavioural classes was to correlate the amplifier's fault flag generation with the amplifier's dynamic range. Amplifiers which exhibited both a low number of fault flags and a relatively small dynamic range were classed as being more likely to have been relatively fault free. Amplifiers with a high number of fault flags and a large dynamic range were classed as being more likely to have experienced fault conditions. The remaining amplifiers' behaviours were classed as likely to have been located somewhere between these two extremes.

modeling approaches, have been presented within a batch processing context. The nature of the modifications to these techniques such that they would be suitable for real-time implementation has been left as an area of future work. Hence, there is a requirement to discuss the issues regarding the validity of the proposed evaluation approach in terms of the how the resulting conclusions pertain to the real-time operation of the novel detection techniques. These discussions will be presented within each of the techniques' results section as warranted.

This chapter will begin by presenting an overview of the classification procedure which will be used for the results obtained from the *in situ* limit checking system. This classification procedure is common to the analysis of *in situ* limit checking results for both the current draw and forward pilot status data signals and, hence, it will be discussed prior to the actual detection system comparisons. Once the classification procedure has been presented, the comparison of the proposed detection techniques' results with those of the limit checking system will then be presented. These comparisons will be presented in two parts with the first part comparing the *in situ* limit checking results with those of the fuzzy membership function based modelling system for the current draw status data signals and the second part comparing the *in situ* limit checking results with those of the recurrent neural network modeling system for the forward pilot status data signals.

7.1 General Classification Procedure for the Limit Checking Results

The example plant had in place a limit checking fault detection system during the 13 months over which the status data set was collected. The main purpose of the detailed analysis of the *in situ* limit checking system's event flags was to partition the 170 amplifiers into rough classes based on their associated limit checking fault flags and their dynamic ranges across the 13 month period. In particular, the amplifiers were divided into those which were essentially "well" behaved, those which alternated between "well" behaved and "poorly" behaved, and those which were essentially "poorly" behaved throughout the 13 month period. As will be recalled, these classifications were required in the previous chapters (most notably Chapters 4 and 6) in the development of the proposed fault detection approaches; hence, their generation is a required first step in the evaluation process.

It should be noted that the results for the limit checking system were for a real, operational limit checking system. In the ideal case, for a given data record it may be possible to set the limit checking bounds such that they produce a detection system with more optimal detection abilities and a lower false alarm rate than what was observed. In practice, for an operational system though, the bounds are set in accordance with the methodology given in Chapter 2 and that are modified relatively rarely. This tends to result in the utilization of bounds that may be relatively distant from their optimal settings[2] for the given data record. It should be recalled that one of the goals of this work is to develop detection techniques that require the same or less human intervention than traditional limit checking systems and still provide improved fault detection capabilities. Hence, the comparison of the *in situ* limit checking system results against those of the fuzzy membership and recurrent neural network based techniques is fair in the sense that these techniques have also not been "hand-optimized" for the particular test data sequences. In essence, the algorithms and techniques which determine the key system parameters for the new detection techniques are as they would be within an operational system.

Because of the large size of the data set (approximately 54 million status data records), it was not feasible to fully test[3] the fuzzy membership function and recurrent neural network based detection techniques across the complete data set. Instead, the techniques will be compared against the *in situ* limit checking system results in a statistical sense, over randomly selected sub-sets of the complete 13-month data set. The exact nature of the set selection procedure for the fuzzy membership and recurrent neural network detection approaches will be discussed within their respective results sections.

Additionally, it should be noted that the detection techniques developed in the preceding chapters have not been tailored for direct application to a real-time setting. The techniques as they stand, particularly the wavelet de-noising and the fuzzy membership

---

2. Here optimal is used in the sense of what the optimal limit checking bounds should have been for a historical data set given hindsight and a knowledge of what fault events occurred.

3. Within his context the term "full test" is used to denote the notion of testing the proposed detection approaches in a manner identical to how they would be deployed in an operation setting. Therefore, this notion includes completely testing the approaches across the full data set in a real-time approach (i.e. with appropriate re-training policies in place).

true classification of the amplifiers would be required if any resulting conclusions were to be valid. As no such knowledge is available for the example engineering plant, no comparison of the amplifiers' behavioural classifications will be undertaken.

The general goal of this chapter will also not be to directly compare the techniques against known fault occurrences, since the details of these are unavailable. Instead, the techniques will be compared in terms of their average threshold bound widths and their flagged events. The general assumption being that a fault detection system which utilizes, on average, tighter threshold bounds is a better fault detection system provided that it is also better able to track the known signal dependencies[1]. In particular, there are two possible cases which may occur. First, the detection system which utilizes the tighter bounds may simultaneously produce fewer event flags than the competing detection system. In this case, it is quite clear that the system with the tighter bounds must be better able to track the known dependencies; hence, it will have superior performance in terms of the fault detection sensitivity and the false alarm rate.

Second, the tighter bounded system may produce more fault flags than the competing system. In this case, the tighter bounded system may still be the superior system, in regards to the fault detection sensitivity and false alarm rate, provided that the event flags produced by it can be accounted for in terms of changes in the signal's behaviour. The tightly bounded system may produce more event flags because it is more sensitive to behavioural changes than the more loosely bounded system. In this case, though, all these additional event flags must be accounted for to ensure that the increase in the event flags is not due to the tightly bounded system possessing a higher false alarm rate. As will be seen in their respective results sections, the fuzzy membership function based detection technique falls into the first case while the recurrent neural network detection technique falls into the second case. Hence, the analyses that will be presented for each of these proposed fault detection methodologies will be tailored to suite the nature of the comparison.

---

1. Within this context, the term "known dependencies" refers to persistent status data signal dependencies which are known not to be related to the occurrence of fault events.These effects are related to normal variations known to be present in the status data signal due to known causes.

Chapter 7:

Results

7.0 Introduction

In this Chapter, a comparison is made between the fault detection methods developed in this work and traditional limit checking fault detection for signals obtained from a real-world, operational, large scale engineering plant. The focus of this comparison will be on quantifying the relative improvements obtained through the fuzzy membership function and recurrent neural network based techniques, particularly in regards to improvements in their false alarm rate and the fault detection sensitivity. To this end, the techniques will be compared over a status data set comprising 13 consecutive months of operation (October 1st, 1995 to October 31st, 1996) of 170 randomly selected cable trunk amplifiers from the total of 354 monitored amplifiers within the example large scale engineering plant.

Within this example plant, there exists no "ground truthed" data for the period of interest, in the sense that no records exist from this time period denoting what faults occurred, nor do repair histories exist indicating what corrective actions may have been taken. The only fault related data which exists for this period are the fault flags which were generated by the limit checking detection system. Hence, in order to develop a partial understanding of the fault state of the plant during this period a statistical analysis of the limit checking detection system's fault flags will be undertaken. The purpose of this analysis will be to classify the amplifiers in terms of being "well" behaved, "poorly" behaved, or somewhere in between. It will be recalled, that these classifications were utilized in the previous chapters as a means of validating the proposed detection techniques. Hence, it is necessary to describe the means by which these classifications were arrived at.

It should be noted though, that due to the dissimilarities in the flagged "events" between the various approaches, these classes will not be utilized directly in comparing the techniques. These classes will only be used in reference to the *in situ* limit checking fault detection system. No attempt will be made to similarly classify the amplifiers under the other techniques. If such a comparison were to be made then some knowledge of the

lower bound on the width utilized by the *in situ* limit checking system over the analyzed data set is 4.63 dBmV or approximately 23 quantization step widths. A discussion of how this estimate was obtained from the particular status data set under consideration will be presented in Chapter 7.

6.4 Conclusions

In this chapter, heuristic methodologies for setting both the training and free-running neural network fault detection thresholds were presented based on the statistical analysis on the difference signals between the neural network system models and the wavelet de-noised forward pilot signal estimates. In both cases these analysis were performed on data sets obtained from the set of 16 amplifiers with “well” behaved forward pilot signals. Also, in both cases the thresholds were selected to generate a worst-case comparison between the neural network modeling approach and the example plant’s *in situ* limit checking fault detection system. An analytical approach to generating these thresholds was precluded in both cases due to the coarse quantization functions utilized in the example plant’s status monitoring system and due to the inherent problems in quantifying the modeling abilities of an given neural network modelling approach for an arbitrary data sequence.

In the following chapter, these heuristic methodologies will be used in the comparison of the neural network based fault detection system against the example engineering plant’s *in situ* limit checking fault detection system.

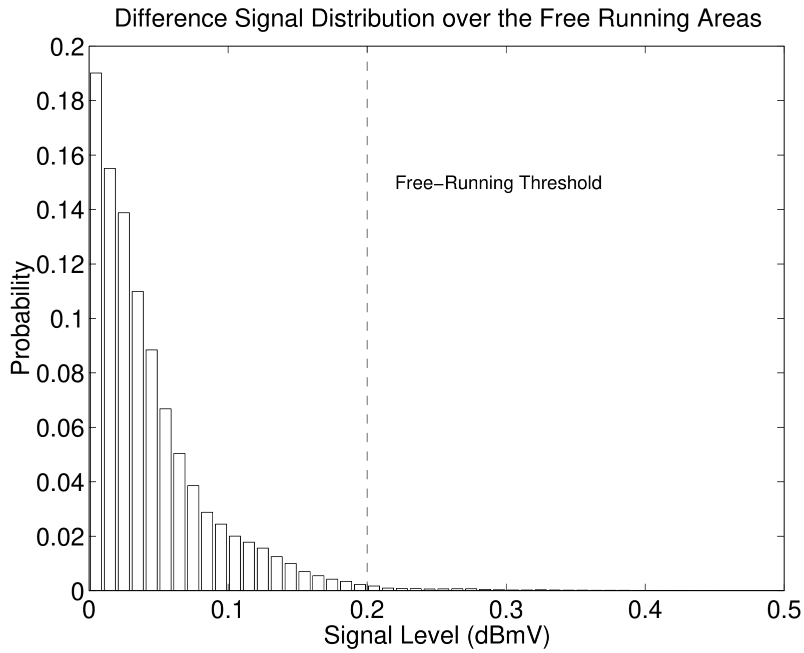


FIGURE 6-11.
Difference signal distribution over the neural networks' free-running areas.

Hence, the free-running threshold heuristic that was selected was to place the threshold at this one quantization step width or 0.2 dBmV level. Since the portion of the distribution which occurs to the right of this threshold level may be due either to real fault events or due to false alarms, and the distribution was generated over 991,285 sample points which represents a statistically significant number of samples, its area represents can be used to provide and upper bound estimate on the resulting neural network fault detection system's false alarm rate. In particular, with the free-running threshold set to 0.2 dBmV for the example plant, the resulting neural network fault detection system would have a false alarm rate on the order of 1.1%. It should be noted, due to the utilization of the absolute value function in the free-running and training thresholds, both these thresholds must be doubled to provide comparable threshold widths to the *in situ* limit checking system which does not utilize a absolute value function is its detection procedure. Hence, the 0.2dBmV level employed above is effectively a 0.4 dBmV, or 2 quantization step width level, under direct comparisons to the *in situ* limit checking system. For reference the estimated mean

6.3 Free-Running Threshold Heuristic

Once the neural network model has been adequately trained, it is then operated as a free-running black-box system model. Behavioural changes in the components are then detected as deviations between the neural network model and, in the case of the example plant, the wavelet de-noised forward pilot signal. The question to be addressed within this section is how the threshold to detect these behavioural changes should be set such that noise effects are not seen as indications of behavioural changes. As in the previous section, this question reduces to the problem of quantifying the expected difference between the neural network estimate and the wavelet de-noised estimate, except in this case the neural network is acting as a free-running, black-box system model. Hence, additional modeling errors, introduced by the incomplete characterization of the given behaviour by the training data set and/or by the underlying noise effects, must be accounted for.

As in the previous case, a theoretical evaluation of this threshold is difficult to obtain. In particular, it is a significantly more complex problem to address than determining a bound on the training error over the training set due to the presence of additional modeling error sources, such as those introduced if the training set failed to fully describe the system's behaviour exhibited within the free-running data set. As was stated earlier, the generation of the theoretical bound for the training case itself is infeasible; hence, no attempt was made to pursue a theoretical approach to estimating the expected difference between the neural network and the wavelet de-noised forward pilot estimates in the free-running case. Instead, the approach taken to once again develop the appropriate heuristic through an experimental trail.

In particular, in the previous trail the neural network was allowed to be free-running in the regions between their temperature re-training events. As in the case of the re-training areas, point by point difference signals can be constructed in these free-running areas and combined to produce a free-running difference signal distribution. A plot of the difference signal distribution obtained from the free-running areas of the 16 example amplifiers with "well" behaved forward pilot signals is shown in Figure 6-11. From this figure it is quite clear that even in the free-running section the vast majority, 98.90%, of the difference signal sample points are under the typical forward pilot quantization step width of 0.2 dBmV.

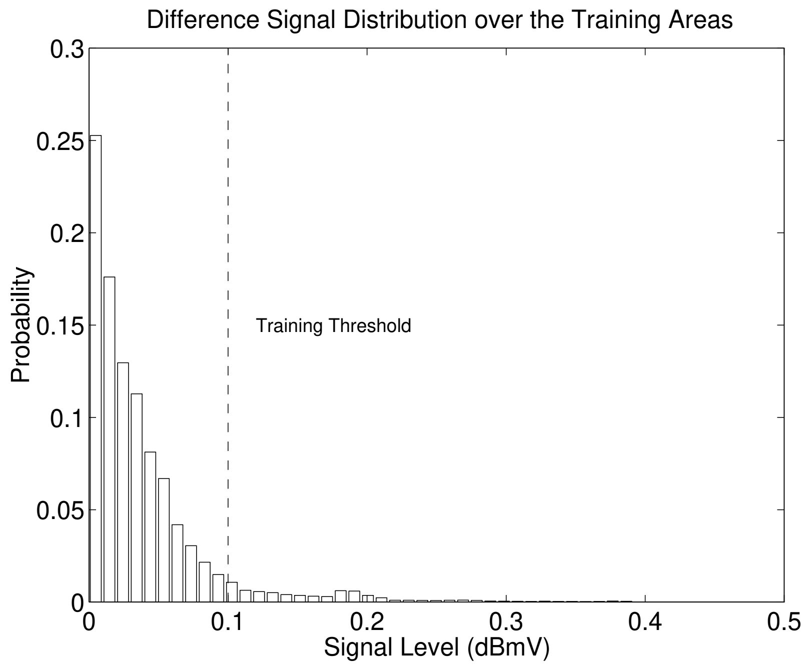


FIGURE 6-10.
Placement of the training threshold with respect to the difference signal distribution over the neural networks' training areas.

As will be seen in Chapter 7, this heuristic generates significantly tighter bounds on the modeling than that utilized in practice for the example plant's *in situ* limit checking fault detection system, and still produces significantly fewer fault flags, despite its worst case nature. This heuristic can be generalized by utilizing the percentage of the differences under a specific threshold as a general classification measure for the neural network training. Thereby, allowing the neural network models to be classified as "poorly" trained, "well" trained, or somewhere in between based on the chosen threshold setting. This type of classification scheme would be useful in enabling the neural network modeling technique to be easily applied to alternate large scale engineering plants. The training threshold heuristic, as given above, is specific to the given example engineering plant and would need to be modified to suite the signal characteristics of other plants to which the neural network modeling approach may be applied.

6.2.2.1 Training Threshold Heuristic

In the development of the training heuristic for the example plant, it should be noted, that for comparison purposes within this dissertation it is desirable to set the training and free-running neural network fault detection thresholds as tightly as is feasible. In essence, this represents a worst case scenario for the neural network fault detection system in that the resulting system will be quite sensitive to modeling errors. If the neural network approach performs better than the *in situ* limit checking system when it is utilizing very tight thresholds, then it will clearly also perform better when the thresholds are relaxed, since the threshold settings do not directly effect the neural network based detection system's modeling capabilities. Tight thresholds therefore allow a worst-case comparison to be made between the two very different fault detection methodologies.

Under this condition, the resulting training heuristic that was determined from the experimental trail, for the given training parameter settings, was to deem the neural network modeling as adequate if 90% or more of the resulting neural network's behavioural model was within 0.1 dBmV of the wavelet-de-noised forward pilot signal. This threshold level was selected by analyzing the distribution of the absolute differences between the neural network models and the de-noised forward pilot signals at each of the sample points, shown with the superimposed training threshold level in Figure 6-10. It is quite clear from this figure that the majority of these difference (93.05%) are below the 0.1 dBmV level. The 0.1 dBmV level translates into a behavioural model which is accurate to within one-half of one quantization step of the de-noised forward pilot signal, over the training set. The additional criteria requiring 90% of the difference signal to be located below the threshold was needed to account for the instances where there exist small areas behavioural changes within the training region which are not significant enough to disrupt the neural network's training. This 90% level was determined through the analysis of a number of these small behavioural change events and is intended to be wide enough to allow a large number of training sets to be classed as acceptable, once again in a direct attempt to develop a worst-case scenario for the neural network fault detection system.

abnormal behaviour. A training threshold heuristic should, therefore, be designed such that it accepts this type of neural network training as adequate since the failure of the neural network to track this small region of abnormal behaviour is not indicative of the quality of the neural network model of the majority of the training set.

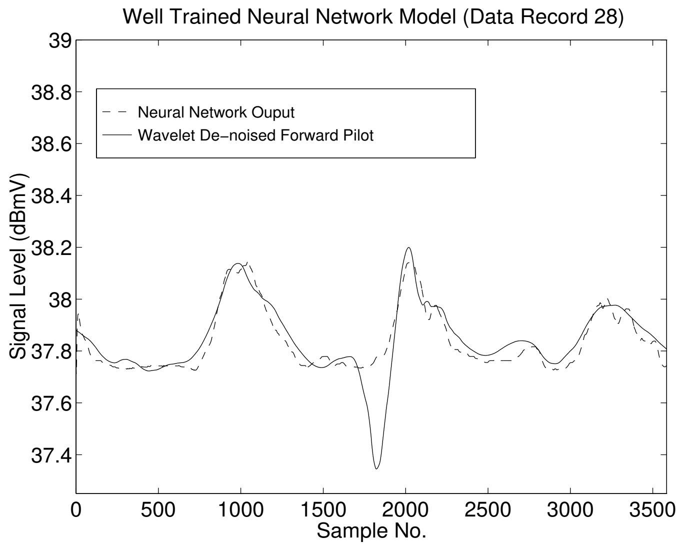


FIGURE 6-9.
Example "well" learned data record showing short duration of abnormal forward pilot behaviour.

It should also be noted that, although an individual neural network's modeling accuracy could likely be improved by fine tuning its training parameters, it is important to recall that one of the goals of this work is to develop techniques which require a minimum of intervention. Hence, it is more desirable to utilize fixed training parameters that provide generally adequate modeling over a wide range of signal behaviours, than to obtain the optimum modeling performance for each forward pilot signal. The resulting models, although not as accurate as may be possible utilizing recurrent neural network technology, are significantly more accurate then the modeling typically available through traditional limit checking approaches, as will be illustrated in Chapter 7. The overall goal is therefore to balance the computational cost of the neural network models against their modeling accuracy in order to achieve a fault detection system which reasonably accurately models a wide range of signal behaviours at a reasonable computational cost.

when it is considered that the normal raw forward pilot quantization step width is 0.2 dBmV's. In the next section, this training region difference signal distribution will be used to develop a specific training threshold heuristic for the example plant.

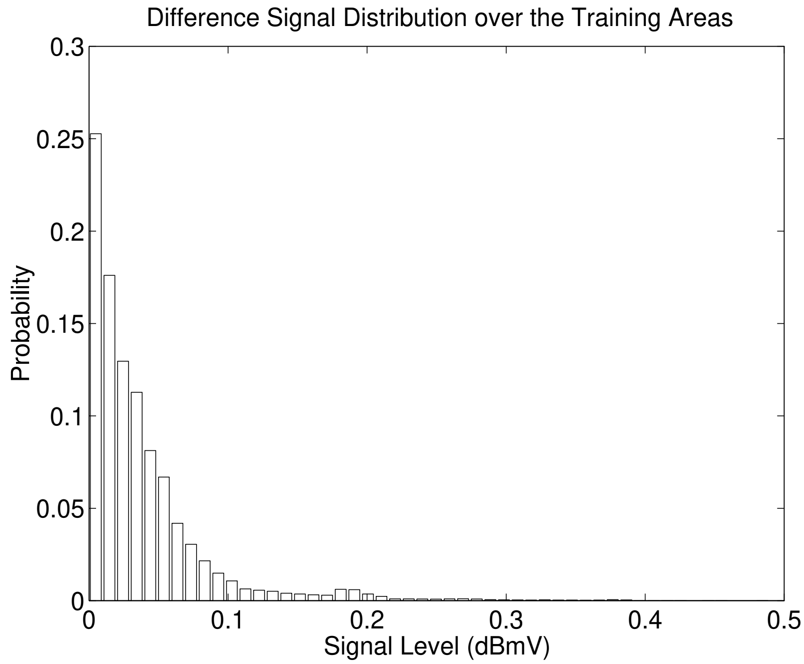


FIGURE 6-8.
Difference signal distribution over the neural networks' training areas.

One factor complicating the development of this heuristic, though, is that there can be regions within a given training block for which the neural network model is unable to track the forward pilot signal's behaviour. Typically, this problem is due to a localized behavioural change within the forward pilot signal itself. An example of this type of occurrence is shown in Figure 6-9. The heuristic which is used to detect whether or not adequate training occurred therefore must be able to account for these problem areas appropriately. Due to the typically small duration of these abnormally behaved areas, it would be prohibitively expensive, in a computational sense, to discard the neural network's training in these cases. In the example given, the neural network adequately models the majority of the forward pilot signal's behaviour, both before and after the occurrence of the abnormal behavioural pattern. It just fails to track the small region of

dynamics of the given data set's enclosure temperature signal. The size of each of the training areas, though, is fixed and related to the training parameters provided during the neural network initialization.

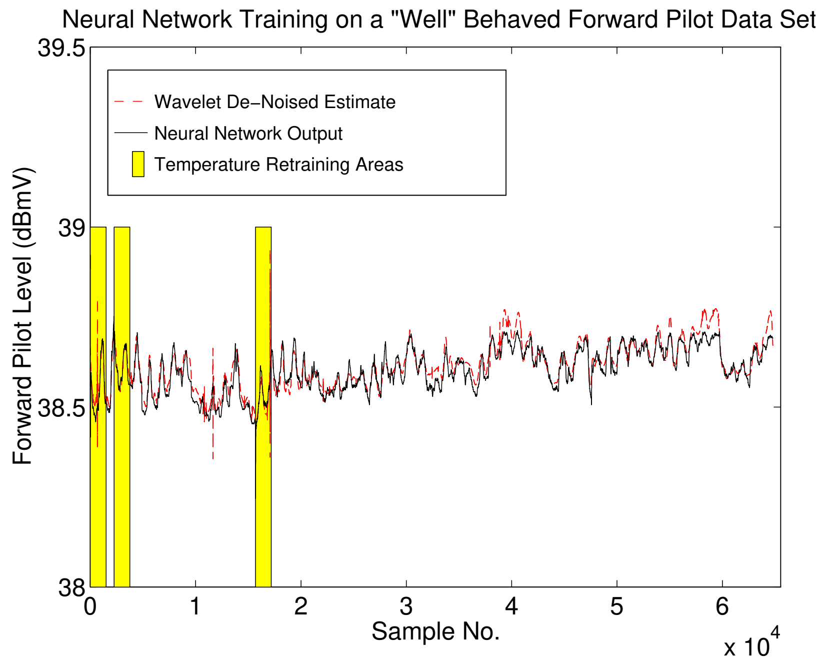


FIGURE 6-7.
Example of neural network training over one of the 16 "well" behaved amplifier data sets.

Once all the neural networks were run on the respective data sets, statistics were then collected with regards to the training region difference signals obtained by subtracting the given neural network's output signal and the associated wavelet de-noised forward pilot signal estimate on a point by point basis over each of the training regions which occurred in the given data set. In particular, each of these training region difference signals, representing a total of 45,759 data samples, were combined to generated the overall training region difference signal distribution shown in Figure 6-8. From this figure it is quite clear that the neural network modeling approach provides quite accurate modeling of the wavelet de-noised estimates of the forward pilot signals within the training regions, particularly

the purpose of these trials is to determine the expected difference between the neural network and wavelet de-noised estimates of the underlying sensor signals such that the validity of the neural network model can be quantified. Utilizing data records in which the given amplifier is suspected of abnormal behaviour does not aid in this goal since the given neural network structure, described above, is designed to learn consistent behavioural patterns. Hence, any inconsistent behaviours are inherently unlearnable by the given neural network architecture. In particular, as mentioned in Chapter 2, the normal behaviour of the amplifiers is that their forward pilot signals are positively correlated with the associated enclosure temperature signal. One type of abnormal or inconsistent behaviour exhibited by the amplifiers is that their forward pilot signals become negatively correlated to the associated enclosure temperature status data signal. In some cases, the neural network's training set may be composed of sections of both positive and negative correlation behaviours. In this case, the neural network structure given in the previous section will be unable to adequately learn the component's behaviour over the training region due to its inconsistency. Hence, to avoid these type of problems, it is generally desirable to utilize "well" behaving sample data records as the basis for determining the expected difference between the neural network and wavelet de-noised underlying signal estimates.

6.2.2 Experimental Trial Results

In the experimental trial, 16 neural network models were trained, one for each of the 16, 65,536 sample long data records which were extracted from the set of amplifiers with "well" behaved forward pilot signals. As will be recalled from Chapter 3, the neural network's modeling error increases when the input signal exceeds the range which it assumed in the training set. This necessitates that the neural network be re-trained with a training set which has been augmented to include the new temperature range. These temperature based re-trainings were the only re-training events which were allowed within the experimental trial. The output of the neural networks, within this trial, therefore contain a number of data areas over which neural network training occurred, as shown in Figure 6-7. The exact number of training areas occurring for any given data set is dependent on the

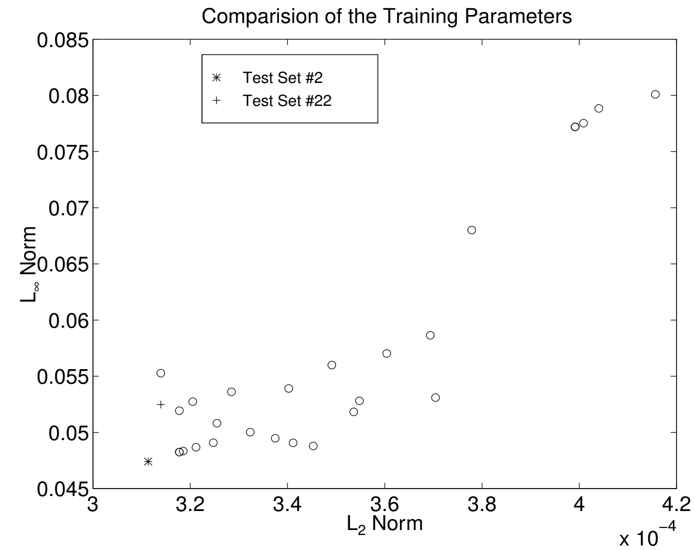


FIGURE 6-6.
Comparison of various neural network training parameters on the example data record.

between these test sets being only a reduction in the number of training epochs utilized). there are other test sets that are closer to test set #2, but each of these test sets also utilize 5,000 training epochs and hence they also have approximately the same computational complexity as test set #2. The resulting system will have a marginally higher modeling error due to the reduction of the number of training epochs but this source of modeling error will be relatively small compared to the other sources of modeling error, described previously, which are inherent in the system. The resulting models will be generated significantly quicker due to the reduction in the training epochs from 5,000 to 1,000. The training parameters of test set #22 therefore are used throughout the remainder of this work.

6.2.1.3 Data Selection

The data for the experimental trial were taken from portions of the sample data set for which the given amplifiers were believed to be “well” behaved. It is important, within this trial, to utilize data in which the amplifiers are behaving “well” since the dynamics of the data from sections of “poor” behaviour tend to be considerably more complex and hence, more difficult for the neural network to learn in a given number of epochs. Additionally,

Table 6.1: Evaluation of the Neural Network Training Parameters

No.	Epochs	Learning Rate	Sampling Rate	L_2 Norm ($x\ 10^{-3}$)	L_∞ Norm
28.	4,000	0.900	0.095	0.3453	0.0488
29.	7,500	0.900	0.095	0.4040	0.0788
30.	10,000	0.900	0.095	0.4009	0.0775

The approach taken within the experimental trials, therefore, was to manually explore a portion of the parameter space over for a given sample data record. The particular sample data record that was chosen had wide temperature variations such that the neural network's learning capabilities would be more fully exercised. Obviously, an easily learned data record would not be very useful in determining the relative merits of the training parameter settings since very accurate learning of such a record may be possible for a large number of the set of training parameter settings. The sampling interval, learning rate, and training epochs were each varied through the set of test runs. The resulting system models were then classified according to their L_2 and L_∞ norms. These norms were selected since the neural network utilizes the L_2 norm in it back propagation learning algorithm and the L_∞ norm is related to the absolute difference of the signals which is used within the fault detection thresholding process. Table 6.1 contains the results of this comparative process for the example data record and graphical plot of this information is shows in Figure 6-6. For each of these trials, the structure of the neural network was kept constant at one input layer neuron, five hidden layer neurons, fifteen scheduler layer neurons, and one output neuron.

From this figure it is quite clear that most of the training parameters given in Table 6.1 result in very little variation in the resulting neural network model, as measure by the L_2 and L_∞ norms. Test set number #2 performs best in terms of the norms, but it does so at a significant computational cost due to its utilization of 5,000 training epochs. A detection system which is computationally less expensive can be constructed by utilizing the training parameters given in test set #22 instead of those given in test set #2 (the difference

Table 6.1: Evaluation of the Neural Network Training Parameters

No.	Epochs	Learning Rate	Sampling Rate	L_2 Norm (x 10^{-3})	L_∞ Norm
4.	5,000	1.000	0.095	0.3248	0.0491
5.	5,000	0.975	0.100	0.3186	0.0483
6.	5,000	1.100	0.095	0.3403	0.0539
7.	5,000	1.000	0.090	0.3178	0.0482
8.	5,000	0.990	0.095	0.3491	0.0560
9.	5,000	1.095	0.095	0.3779	0.0680
10.	5,000	0.985	0.095	0.3536	0.0518
11.	5,000	0.990	0.095	0.3548	0.0528
12.	5,000	1.010	0.095	0.3604	0.0570
13.	10,000	0.995	0.095	0.3992	0.0772
14.	5,000	1.010	0.085	0.3205	0.0527
15.	10,000	0.990	0.095	0.3991	0.0772
16.	7,500	0.990	0.095	0.2988	0.0445
17.	5,000	1.100	0.099	0.3694	0.0586
18.	2,500	0.990	0.095	0.3285	0.0536
19.	1,000	0.990	0.095	0.3140	0.0553
20.	10,000	1.010	0.099	0.4156	0.0801
21.	500	0.900	0.095	0.3704	0.0531
22.	1,000	0.900	0.095	0.3139	0.0525
23.	1,500	0.900	0.095	0.3178	0.0519
24.	2,000	0.900	0.095	0.3255	0.0508
25.	2,500	0.900	0.095	0.3324	0.0500
26.	3,000	0.900	0.095	0.3375	0.0495
27.	3,500	0.900	0.095	0.3412	0.0491

resulting detection system's false alarm rate is to be compared accurately. The approach taken to address this problem was to remove the first and last 256 data samples from the de-noised underlying signal estimate before the signal was passed on to the neural network. The threshold of 256 data samples was chosen since an analysis of the resulting de-noised forward pilot signals in the wavelet domain showed that, typically, the boundary effects did not occur above the 5th scale level. At this scale level the wavelet basis functions have a width of 256 sample points for a 4096 point data record. Hence, if it is assumed that the 5th scale level will be the coarsest scale level affected by the boundary effects, then removing the first and last 256 data samples from the de-noised signal will remove any boundary effects which may have been present at or below the 5th scale level.

It should be noted that these effects are due to the batch processing context under which the wavelet de-noising was performed within this dissertation. These effects could be eliminated within an appropriately designed real-time detection system.

6.2.1.2 Training Parameter Selection

Obviously, the ability of a given neural network to model a given system behaviour is dependent on the neural network's structure and its training parameters. Without, an independent method of determining a theoretical limit on the neural networks training ability, the adjustment of the network's structure and training parameters such that the "best" training is achieved is difficult. The nature of the neural network's training process, in particular its computational cost, is such that the utilization of optimization procedures to identify optimal or near optimal setting for these parameters is computationally infeasible.

Table 6.1: Evaluation of the Neural Network Training Parameters

No.	Epochs	Learning Rate	Sampling Rate	L_2 Norm (x 10^{-3})	L_∞ Norm
1.	5,000	0.950	0.095	0.3178	0.0482
2.	5,000	0.900	0.095	0.3114	0.0474
3.	5,000	0.975	0.095	0.3212	0.0487

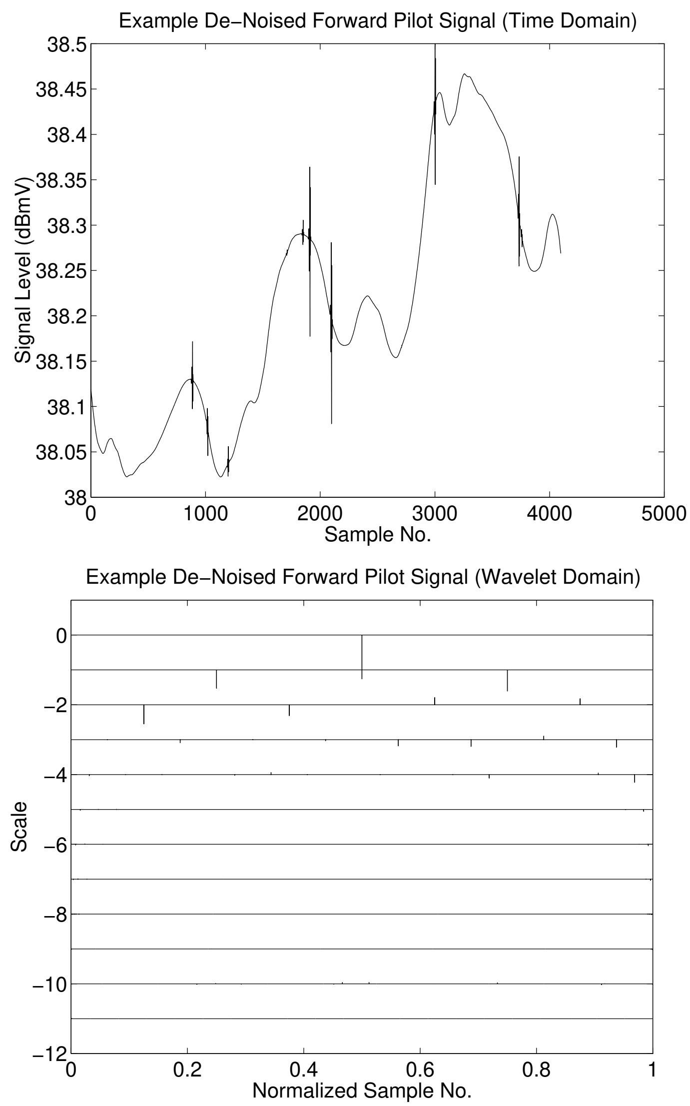


FIGURE 6-5.
Example demonstrating the occurrence of wavelet boundary effects in the de-noised underlying signal estimates.

The presence of these boundary effects can cause problems during the neural network training process. In particular, these effects may appear as behavioural inconsistencies to the neural network which may result introducing significant modeling errors into the behavioural mapping performed by the neural network. The effects are not a true indication of the underlying sensor signal's behaviour and, hence, should not be included as part of the neural network's training set. Obviously, these artifacts need to be removed if the

wavelet de-noising process, this assumption can lead to artifacts being present in the underlying signal estimates [24][78]. In particular, if the far left and far right portions of the signal are located at significantly different levels then, when the signal is periodized, sharp transitions will be introduced into the augmented signal (Figure 6-4). These sharp transitions appear in the wavelet transform as significant wavelet coefficients at the transform's boundaries which may occur across several of the wavelet scales. These resulting coefficients, if they are large enough to pass through the de-noising process, may in turn result in signal artifacts being produced at the leading and trailing edges of the underlying signal estimate. The magnitude of these artifacts will be dependent on the magnitude of the mismatch between the given signal's leading and trailing edges. Figure 6-5 shows an example of this phenomenon for one of the de-noised forward pilot signals obtained from the example plant. In particular, it can be seen quite clearly that, for scales 5 and below, non-zeros coefficients exist close to the edge boundaries. These coefficients are most likely associated with the boundary wavelets at the given scale with the miss-match between the beginning and end of the signal causing significant coefficients to be generated for these boundary wavelets which were large enough to pass through the de-noising process.

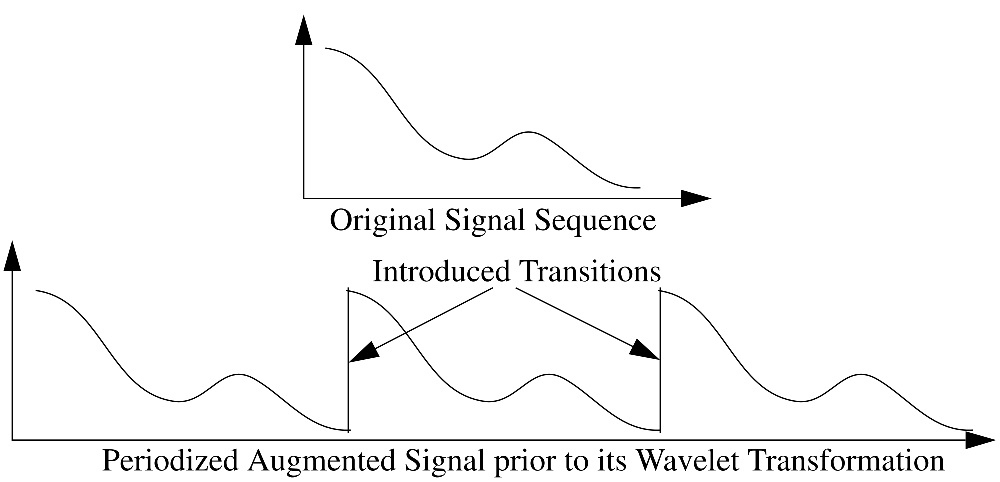


FIGURE 6-4.
Effect of introducing significant transitions, resulting in signal artifacts, due to the application of the proposed wavelet de-noising process.

bilities of a given neural network to model the behaviour of a given system is an open research area. Therefore, the approach that was taken was to obtain a numerical estimate of the expected error through an experimental analysis of the neural network's modeling capabilities.

In particular, a data record of length 65,536 was selected at random from each of the 16 amplifiers which were classed as having "well" behaved[1] forward pilot signals. The de-noised underlying signal estimates for these 16 data sequences were then obtained and used, in accordance with Figure 6-3, as the desired output signal for the neural network training. Appropriate settings for the two neural network training parameters were then selected and fixed for each of the 16 neural networks to be trained. This section will begin by overviewing the training issues relevant to this testing procedure. This overview will be followed by the analysis of the testing results and the development of the training threshold heuristic.

6.2.1 Training Issues

There are several issues regarding how the data was generated for this experimental trial and how the neural networks were set up that reflect on the applicability of this trial, and hence the selected training threshold heuristic, to the real-world application of the neural network based fault detection technique. Of particular interest are the issues of how the boundary effects of the wavelet de-noising process were addressed, how the neural networks training parameter were selected, and the rational behind selecting the given data sets as the ones to include in the experimental trial. Each of these issues will be addressed in turn in the following sub-sections.

6.2.1.1 Boundary Effects of Wavelet Transform

The wavelet de-noising approach presented in Chapter 5 is based on the utilization of a periodic wavelet transformation function. This approach to performing wavelet transform, is quite similar to standard fast Fourier transform in that it implicitly assumes that the signal sequence to be transformed represents one period of a periodic signal. Within the

1. The nature of how the "well" behaved amplifiers, with respect to their forward pilot status data signals, were selected will be more formally discussed in Section 7.3.1 of Chapter 7.

much more likely to result in the neural network learning the "true" behaviour of the system, as represented by the training set. Hence, the fault detection performance of the system should improve through a reduction in the false alarm rate.

6.2 Training Threshold

The first step in the utilization of the recurrent neural network based fault detection system is to train the neural network so that it provides a black-box model of the component's dynamic behaviour over the given training set. The question to be addressed within this section is, therefore, how to quantitatively determine when the neural network has trained adequately over a given data sequence. The answer to this question lies in determining what is the expected difference between the neural network's signal estimate and the estimate obtained via the wavelet de-noising over areas of training data. Within Chapter 1, it was discussed that recurrent neural networks are capable of learning to model any non-linear, dynamic systems, once trained over an appropriate data set. The crux of this possibility rests on three general assumptions:

1. The neural network is composed of enough neurons to fully capture the system dynamics.
2. The neural network has been trained over a suitably long number of epochs enabling it to learn the system dynamics.
3. The neural network's training parameters (i.e the learning rate and sampling interval in the case of the recurrent neural network model) have been set appropriately.

Currently, within neural network theory, a quantitative solution, for a given data set, to these conditions is an open research area. Therefore, within a real-world neural network implementation, the neural network may not be able to model a given system to a desired degree of accuracy due to limitations in the number of neurons employed, the training time allowed, or the training parameter settings. Hence, in the case of the recurrent neural network based detection system some modeling error between the neural network system model and the wavelet de-noised signal is to be expected even within the training set. An analytical analysis of the nature of this modeling error is not feasible due to the complexity inherent in the neural network modeling approach. How to quantify the modeling capa-

ing signal. Instead, this rise is more probably due to the neural network's output being a scaled version of the input temperature signal. The rise exists within the temperature signal, but it did not produce a significant effect in the underlying sensor signal, as indicated by its failure to significantly influence the raw forward pilot signal. Some of the other fine detailed signal variations which occurred for the neural network output in Figure 6-1 can also be ascribed mainly to the direct result of the neural network's output signal being a scaled version of the temperature status data signal. These small variations do not relate in general to variations that are likely to have existed in the underlying sensor signals under the assumptions made about this class of signals likely properties in the preceding chapters.

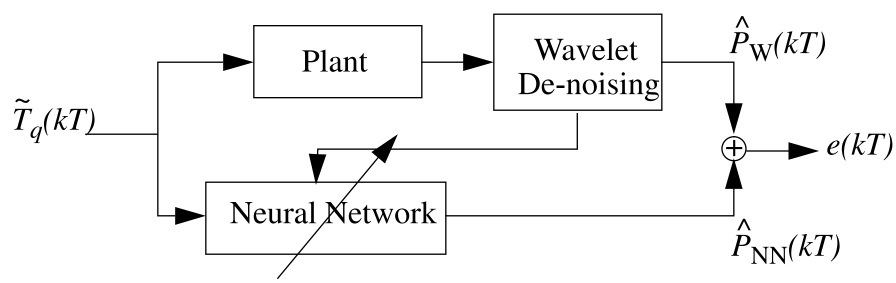


FIGURE 6-3.
Re-structured Recurrent neural network based fault detection system.

Artifacts such as these, if not accounted for appropriately could cause a significant increase in the detection system's false alarm rate. For the particular example given, the difference in the two signals caused by this effect is on the order of one-quarter of a quantization step, clearly large enough to adversely effect the positioning of the fault detection threshold. Since this problem is due mainly to the high noise levels of the forward pilot signals, it can be reasonably addressed by modifying the detection system to utilize the wavelet de-noised underlying signal estimate as the desired neural network output signal during the training phase, as shown in Figure 6-3. In this manner, the neural network is able to learn the system's behaviour with quicker and more accurately since the noise effects have been removed. The least mean square optimization process, therefore, is

over small sample records, and if the rise is assumed to be a true indication of the underlying sensor signal, then the raw pilot signal should have also exhibited a rise in its level over the same sample range.

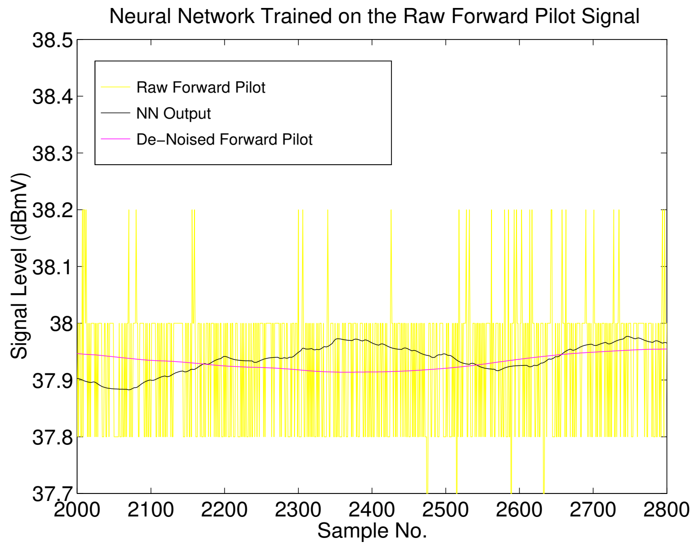


FIGURE 6-2.
Detailed view of the data in the range 2,000 to 2,500 of Figure 6-1 demonstrating the neural network's poor estimation of the underlying sensor signal as indicated by it rise to very near the 38.0 quantization level.

Overall, the raw forward pilot signal, shown in Figure 6-1, is consistent with an underlying noise amplitude in the range of one half of a quantization step, therefore as the underlying sensor signal crosses the half-way point between two adjacent quantization levels, quantized values from both levels should be produced within the given section of the raw signal. This is inconsistent with the results shown in Figure 6-2. The de-noised forward pilot estimate rises significantly over the samples in the range of 2,300 to 2,500. in particular, the neural network output rises to within one-quarter of a quantization step to the next quantization level. If this rise was a true indication of the underlying sensor signal, then the raw data signal should have contained significant numbers samples from the quantization step above 38.0 dBmV with the 2,300 to 2,500 sample range. Because no such rise in the raw data signal occurs within this range it is unlikely that the rise shown in the recurrent neural network's output signal is an accurate estimation of the true underly-

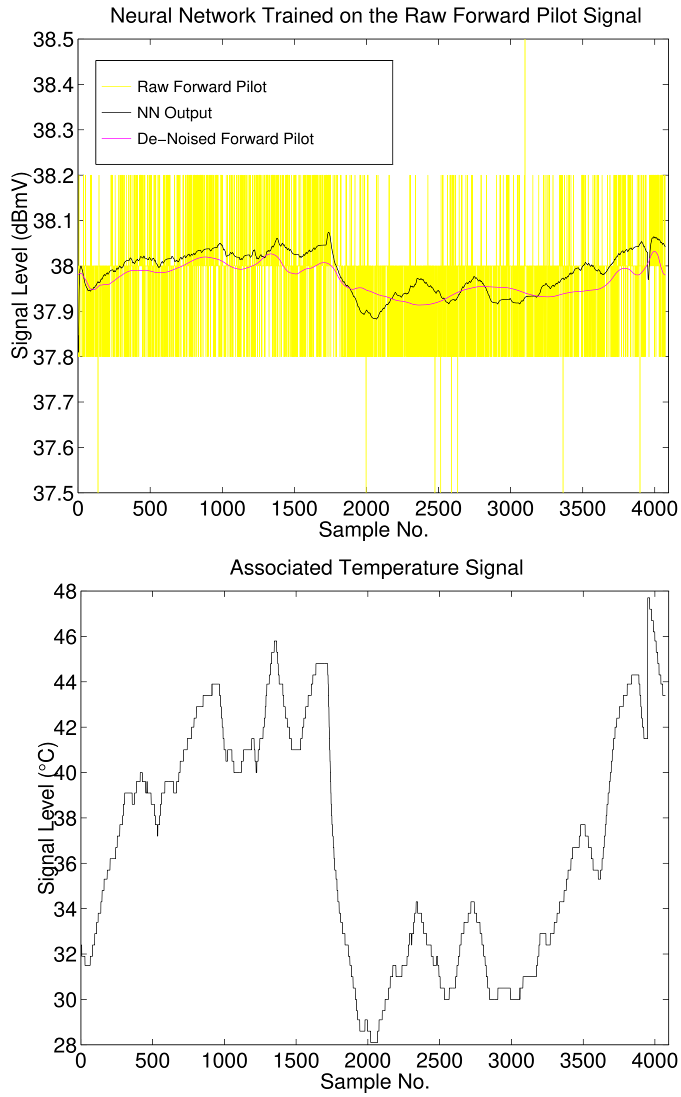


FIGURE 6-1.
Comparison of neural network trained on raw pilot signal and wavelet de-noised estimate of the underlying sensor signal.

enlarged view given in Figure 6-2. Within the sample range of 2,200 to 2,500 the neural network's underlying signal estimate rises noticeably above the underlying signal estimate provided by the wavelet de-noising technique. The presence of this rise, though, is inconsistent with the assumptions about the nature of the underlying sensor signal made in the previous chapter. In particular, if the underlying sensor signal is assumed to be a low pass signal, the underlying noise processes is assumed to be Gaussian and relatively stationary

relatively small behavioural sample. Within the much larger data set over which the neural network is free-running, small variations in the system's behaviour may be present which were not contained in the training set and hence not learned by the neural network. The presence of these small behavioural changes are not significant in a fault detection sense since they do not indicate that there has been significant changes in the given component's operational state. Therefore, what is required is a methodology of setting the free-running fault detection thresholds in a prescribed manner such that the presence of small modeling errors can be differentiated from statistically significant fault events. Ideally, the resulting thresholds should also allow the detection system to maintain a low false alarm rate combined with a relatively high fault detection sensitivity.

6.1 Detection System Structure

In Chapter 5, the proposed recurrent neural network based detection system utilized the raw forward pilot signal as the desired output signal for the recurrent neural network during the training phase. Because of the relatively high underlying noise levels present on the forward pilot signals and the coarse quantization processes employed, these signals consist mainly of rapid transitions between adjacent quantization levels overlaid on a relatively slowly varying, temperature dependent, underlying sensor signal. This creates significant problems in terms of the neural network's ability to estimate the underlying sensor signal's behaviour from the raw data signal. In particular, the neural network "learns" by performing a least means squared optimization to identify the weight space which provides the best mapping between, in this case, the input raw temperature signal and the raw forward pilot signal. Unfortunately, in some instances this training procedure can lead to the neural network arriving at a least means square solution which essentially performs the mapping by scaling the temperature signal such that it is within the raw forward pilot signal's dynamic range. This effect can be see in the example neural network training shown below in Figure 6-1.

This process deleteriously affects the manner by which the fault detection thresholds can be set. In particular, the resulting neural network's system model may tend to have variations within it that are inconsistent with the assumptions made about the nature of the underlying sensor signals. This effect can be quite clearly seen in Figure 6-1 and the

Chapter 6:

Recurrent Neural Network Based Fault Detection: Threshold Determination

6.0 Introduction

In Section 5.1 of the previous chapter, a basic methodology was introduced for utilizing a recurrent neural network as the basis for a fault detection system. In this chapter, this proposed detection system will be elaborated on, particularly in terms of how the fault detection thresholds may be determined for a given data set. First, however, some limitations of the previously presented system will be discussed and an approach to address these limitations will be presented. Once this modified detection system has been presented, two issues will be discussed which are key to enabling the detection threshold to be set in a prescribed manner under the two neural network operating states: training and free-running. The first issue involves the development of a methodology that permits a quantitative measurement of whether or not a given neural network model has trained sufficiently such that it adequately models the behaviour exhibited within a given data set. Through the associated discussion it will be seen that, in essence, this issue reduces to one of determining the expected difference between the trained neural network system model and the system over the training set. Due to practical limitations in the number of neurons employed, the number of training epochs, and the optimality of the training parameters, the neural network system model will be unlikely to exactly model the desired system. Modeling errors will exist and these errors need to be accounted for if the fault detection thresholds are to be set in a prescribed manner.

Once the neural network has been trained, it is used as a free-running, black-box system model. Hence, the second issue to be addressed involves how to quantitatively set the fault detection thresholds over these free-running sections. Obviously, some modeling errors will be introduced at this stage since the data set used in the neural network's training may not have completely captured the system behaviour exhibited over the free-running section. In essence, the neural network learns the system behaviour by training over a

distinct time domain status data sequences due to their random nature. Each underlying noise signal will cause different transitions to occur at different times within their associated status data signals. These effects will be particularly noticeable at the extremes of the status data signals' dynamic ranges. The best that can be hope for, therefore, is that the overall, or gross, behaviour of the two status data signals is similar, as measure by the Chi squared goodness-of-fit test performed on the probability distributions of their quantization levels, as is the case with the signals of Figure 5-6 and Figure 5-19.

5.3 Conclusions

In this chapter, the nature of how the recurrent neural network modeling procedure, presented in Chapter 1, could be utilized as a basis for a fault detection systems for plants exhibiting unknown, non-linear, dynamic dependencies within their collected status data signals was explored. From this discussion it was determined that in order to be able to set prescribed fault detection thresholds for the neural network modeling approach, particularly under the coarse quantization effects present in the status monitored signals of limit checking fault detection systems, a methodology of estimating the underlying sensor signal and the statistics of the underlying noise signal was required.

To this end, a methodology for estimating the underlying sensor signal was proposed based on the utilization of wavelet de-noising techniques. In particular, the best de-noising techniques for the forward pilot and enclosure temperature signal classes of the example plant were identified and a methodology to correct for the known non-Gaussian noise contamination on these signals was presented. The chapter then concluded by presenting a numerical method by which the statistics of the underling noise signal could be estimated based on the transition statistics present in the original status data signals and the wavelet de-noising estimate of the underlying sensor signal.

The next chapter will detail how the underlying sensor signal and the underling noise estimates can be utilized to enable the fault detection threshold of the recurrent neural network based fault detection system to be set in a prescribed manner.

underlying sensor signal estimate provides a relatively strong indication of the “reasonableness” of the underlying sensor signal estimate, and, hence, validation of the de-noising methodology.

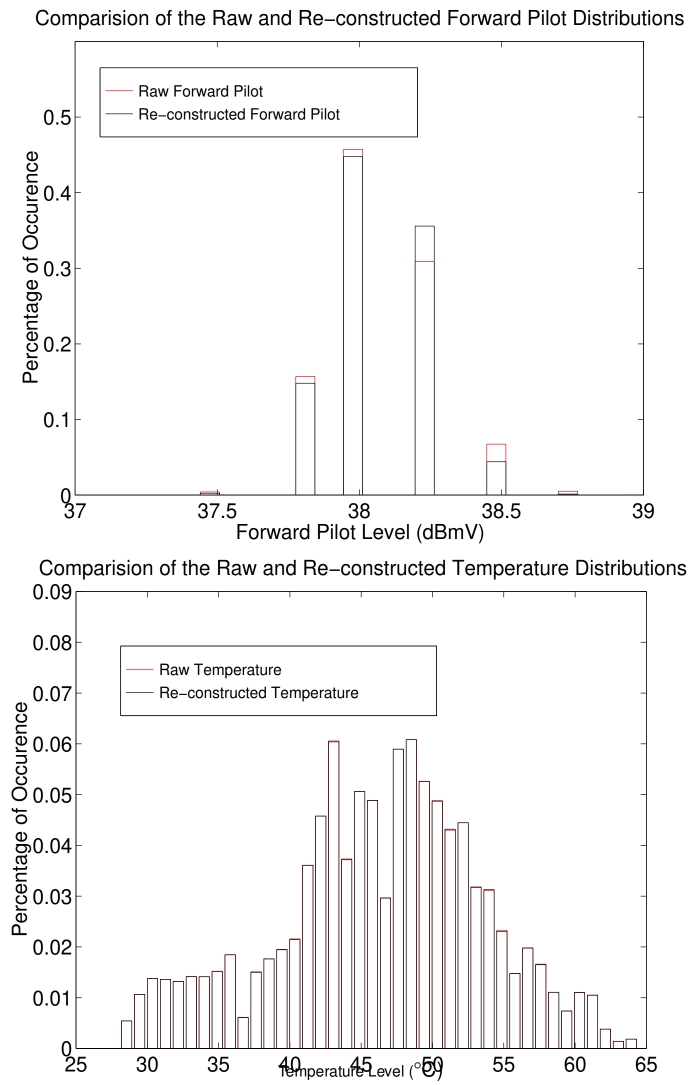


FIGURE 5-20.
Quantization level probability distributions for the raw and re-constructed forward pilot and enclosure temperature signals.

It is important to note that an exact re-construction is not feasible due to the random nature of the underlying noise. Two underlying noise signals can be statistically equivalent, in terms of the effects that they produced on the status data signal, and still produce

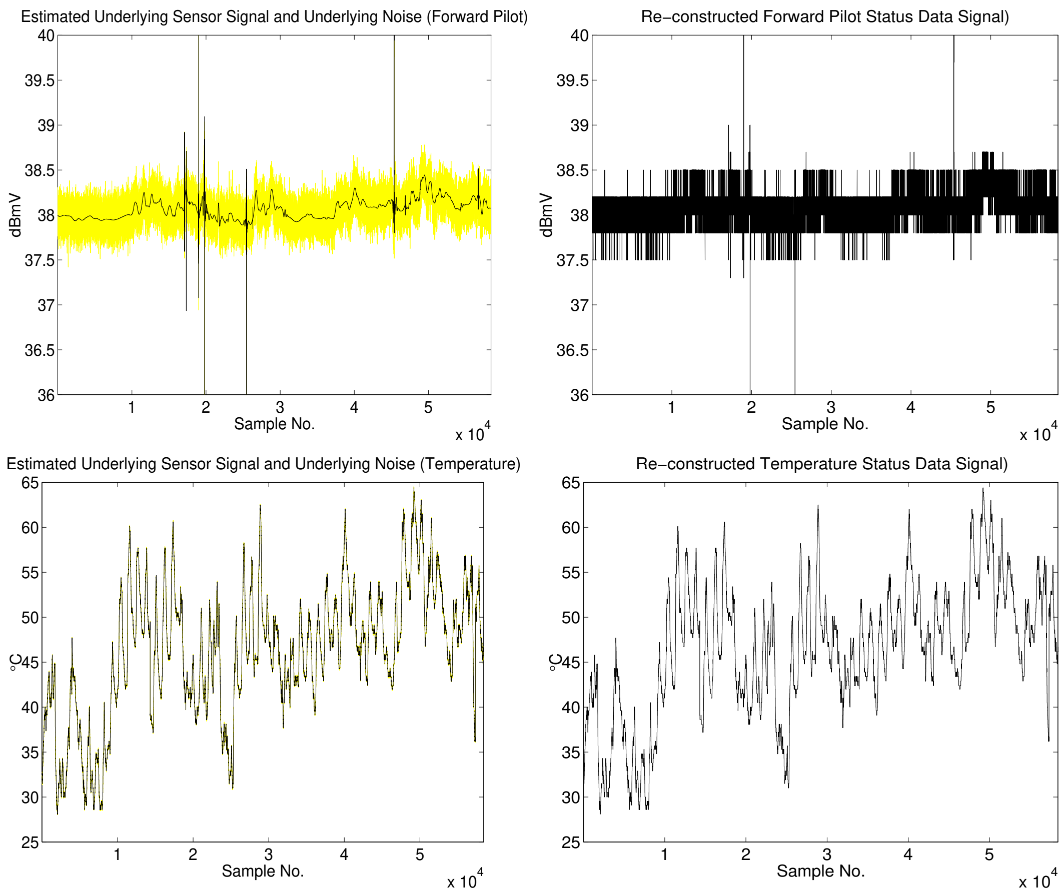


FIGURE 5-19.
Re-constructed status data signals obtained utilizing the estimated underlying sensor signal and noise estimates (forward pilot estimate - N(0,0.1197) Gaussian underlying noise, enclosure temperature estimate - N(0,0.0236) Gaussian underlying noise)

More particularly, a Chi-squared goodness-of-fit test [3] was utilized to provide confirmation that the quantization level probability distributions, shown in Figure 5-20, for the reconstructed forward pilot and enclosure temperature signals were statistically similar to the distributions obtained from their respective raw data signals. Both of the reconstructed signals' quantization level probability distributions easily satisfied the Chi-squared test with a confidence level set to 95%. The closeness of these distributions provides some indication that the re-constructed signals are "reasonable" in the sense that the Gaussian noise assumption, upon which the reconstruction is based, is a valid assumption. The fact that a statistically similar estimate of the raw data signal is obtainable from the generate

of the associated underlying noise signal. As the variance of the underlying noise increases there is an associated increase in the average number of transitions occurring within the status data signal. Therefore, the underlying noise signal's variance can be estimated by the following procedure,

1. Generate an estimate of the average number of transitions per data sample occurring in the given status data sequence.
2. Generate an estimate of the underlying sensor signal via the wavelet de-noising technique outlined previously.
3. Generate a $N(0,\sigma)$ Gaussian noise sequence and add it to the underlying sensor signal estimate.
4. Quantize the resulting signal utilizing the known quantization function $\tilde{Q}(.)$ obtained from the status data sequence.
5. Calculate the average number of transitions per data sample occurring in this re-constructed signal.
6. If it is less than the estimate obtained in Step 1, go back to Step 3 increasing the variance of the generated noise sequence.
7. If it is more than the estimate obtained in Step 1, go back to Step 3 decreasing the variance of the generated noise sequence.

Obviously, some care must be taken in how the calculations determining the average number of transitions per data sample for a given data sequence are performed to ensure that the resulting estimate's variance is suitably small. This can be accomplished fairly easily by dividing the signal over which the estimate is to be obtained into N equally sized chunks. The estimate of the average number of transitions per data sample can then be obtained for each data chunk and this set of estimates averaged to generate the estimated number of transitions per data sample occurring over the complete data sequence. Figure 5-19 shows the results of utilizing the above procedure to estimate the underlying noise present in the example status data signals of Figure 5-6.

As can be seen by this figure, the above process results in reasonably good estimates of the statistics of the underlying noise processes, in the sense that these estimates produced re-constructed status data signals that are quite similar to the actual status data signals upon which the underlying noise and underlying sensor signal estimates were based.

estimate in the wavelet selection and non-Gaussian noise correction processes does not adversely effect the ability of the technique to reconstruct the class of transients which are of interest in this work.

5.2.3.4 Underlying Noise Estimation and Process Validation

Once a reasonable estimate for the underlying signal has been determined then this signal can be used as the basis for determining a statistical estimate of the parameters of the underlying noise process. In particular, in Section 2.3, it was assumed that the status data signals were given by

$$\tilde{y}_q(kT) = \tilde{Q}(\,[\tilde{x}(t) + \tilde{n}(t)]\,|_{t = kT}) \tag{5.37}$$

where $\tilde{y}_q(kT)$ is the status data signal generated by the status monitoring system, $\tilde{x}(t)$ is the underlying sensor signal, $\tilde{n}(t)$ is the underlying noise, which has been assumed to be Gaussian and wide-sense stationary over the given data record from which the noise estimate is to be obtained, and $\tilde{Q}(.)$ is the known quantization function extractable from the collected status data sequence. Hence the problem of estimating the underlying noise process becomes one of substituting the wavelet underlying signal estimate $\hat{x}(kT)$ into Eq. 5.37 and solving for the only remaining unknown $\tilde{n}(kT)$, or more particularly solving for its statistical parameters, namely its standard deviation and mean (under the assumption that it is Gaussian noise). Of course, once the statistics of the noise have been estimated, then an additional step is required to confirm that the original hypothesis that the noise is Gaussian is valid. A process by which this confirmation can be obtained will be discussed at the end of this section.

Once again the presence of the coarse quantization function $\tilde{Q}(.)$ impedes the application of direct analytical solutions to the problem of estimating the underlying noise statistics. A solution to the above problem though can be arrived at numerically. In particular, it can be recalled from Section 3.1.2 that the presence of the underlying noise is the main contributing factor to the presence of the transition noise on the coarsely quantized status data signals and that the statistics of the transition noise effects are related to the variance

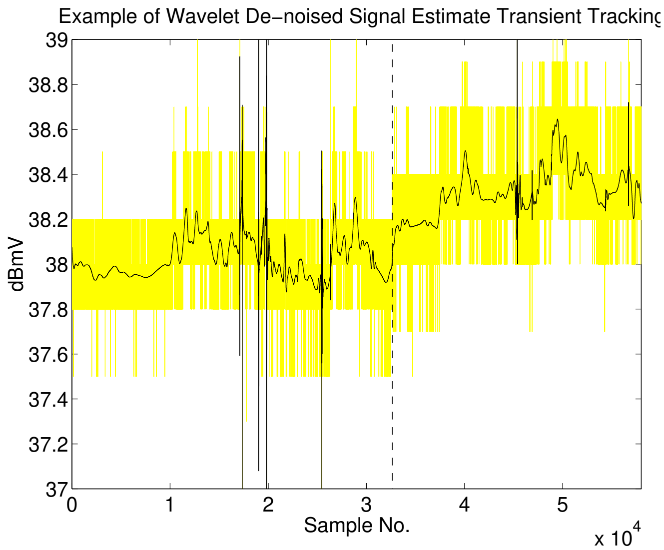


FIGURE 5-18.
Example demonstrating the offset transient tracking ability of the complete wavelet de-noising methodology (the location of the transient event is indicated by the dashed line).

mated underlying sensor signal track them accurately. To validate the selected de-noising technique in regards to this condition, an artificial offset of 0.2 dBmV (or one quantization step) was introduced to the raw forward pilot sensor signal of Figure 5-6 at one of the moving average sample points, since this represents a worst-case offset in terms of both magnitude and location. The resulting signal was then de-noised in accordance with the procedure outlined above, as shown in Figure 5-18. It is clear from this figure that, although the wavelet is selected based on the moving average estimate and this estimate is also used in the non-Gaussian noise correction procedure, the utilization of the moving average estimate in these ways does not impeded on the abilities of the complete wavelet de-noising approach to track transient events of this kind.

It should be noted that other transients are possible in the raw status data, such as very short duration spike events. These other types of transient events though are not of concern within this work since they are not significant to the early fault detection process since they pertain mainly to transient noise effects. A full analysis of the de-noising methodology's ability to re-construct transient effects has been left as an area of future. What has been presented here is an example indicating that the utilization of the moving average

$\{x_{avg}(nT)\}$ could be obtained through a constraint first order optimization procedure. This procedure was straight forward since each of the thresholds could be optimized independently of the others due to the orthogonal nature of the dyadically scaled and shifted wavelet functions. Once, the optimal threshold setting for the non-Gaussian noise case has been determine the sensor signals can then be re-processed to result in de-noised underlying signal estimates which are not conditioned on the Gaussian noise assumption. Once, again this two-stage de-noising procedure is necessary to account for the coarse quantization effects present in the status data signals, which generate signal dependent noise effects. Figure 5-17 shows the results of applying this non-Gaussian noise correction procedure on the sample signals of Figure 5-6. It can be clearly seen that this procedure results in an estimate of the underlying sensors signal, which has considerably less high frequency noise content than the non-corrected estimate shown in the upper left quadrant of Figure 5-16. It should be noted that this correction procedure was only performed on the 10^{th} order Daubechies and 4^{th} order Coiflet wavelet de-noising techniques identified in the previous section.

It is important to re-iterate at this point that the above corrected estimate is still based on the underlying signal estimate provided by the cross-validation thresholding methodology. This threshold selection methodology is based on a mathematical analysis of how smooth estimates of Gaussian noise contaminated signals can be extracted. Therefore, the estimate given in Figure 5-17 above does have a grounding in mathematical theory; the only change has been that a correction step has been included to account for the known non-Gaussian nature of the underlying noise. Since only increases in the thresholding levels are allowed in the correction step, the basic underlying signal, identified by the cross-validation technique, remains unchanged by the optimization procedure. Only thresholds associated with the high frequency noise effects, seen in Figure 5-16, are changed.

It is important to also verify, at this stage, that the utilization of the moving average signal estimate does not adversely effect the ability of the complete wavelet de-nosing process to track transient signal events. In particular, the transient events of primary interest are those that result is a sudden shift in the mean signal level. Since these types of shifts are highly likely to be indicative of behavioural changes, it is desirable that the esti-

sian noise contamination is not valid for the signals generated for the type of status monitoring systems of interest within this work, and particularly for those of the example plant. Hence, the de-noising techniques identified in Section 5.2.3.2 will produced sub-optimal results for the status data signals of interest within this work, which is apparent in particular for the forward pilot estimate shown previously in Figure 5-16.

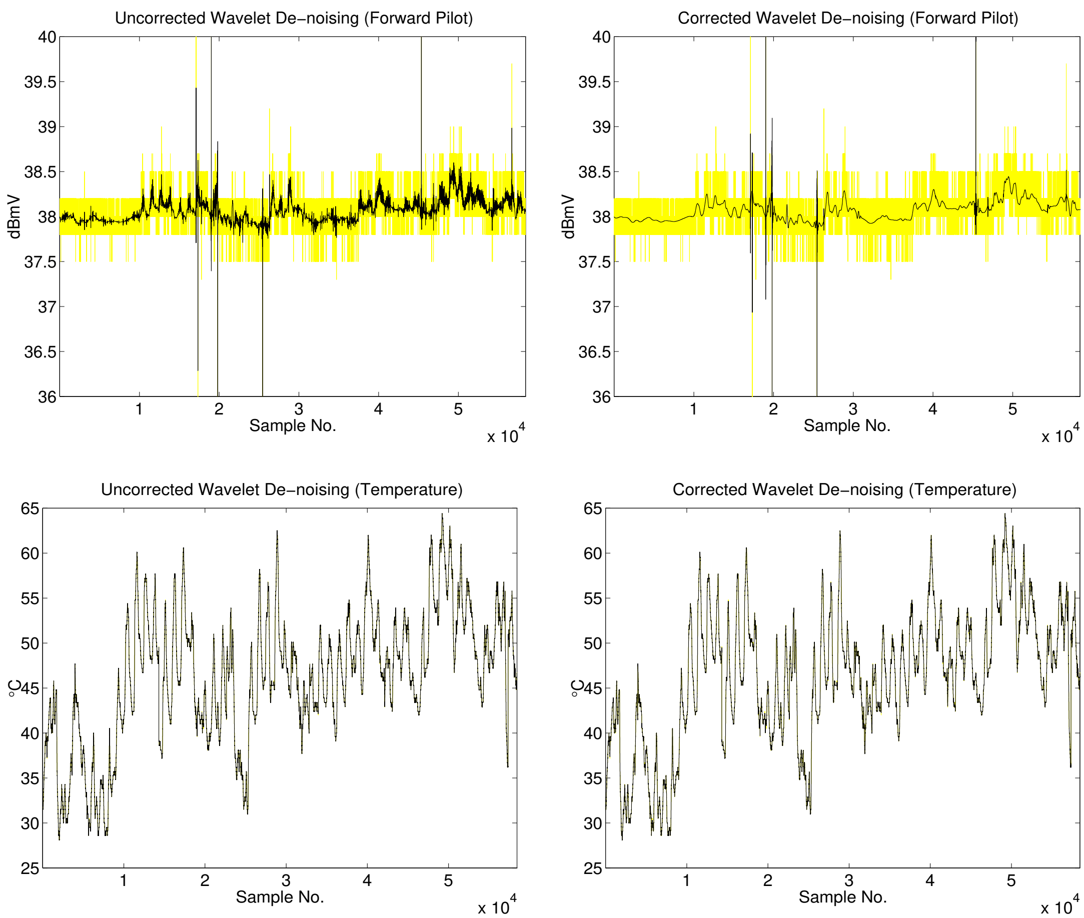


FIGURE 5-17.
Underlying signal estimates after application of the correction procedure required due to the known presence of non-Gaussian noise.

The approach taken to address this problem was to utilize the thresholds, obtained through the above de-noising process, as an initial starting point for an optimization procedure utilizing Eq. 5.35 as the objective function. In essence, each of the thresholds, at the various scale levels, were adjusted upward to determine if a better approximation to

mation is gained through the de-noising operations. These operations, though, require a significant computational expense. For this reason, within Chapters 6 and 7, only the raw enclosure temperature signal will be utilized.

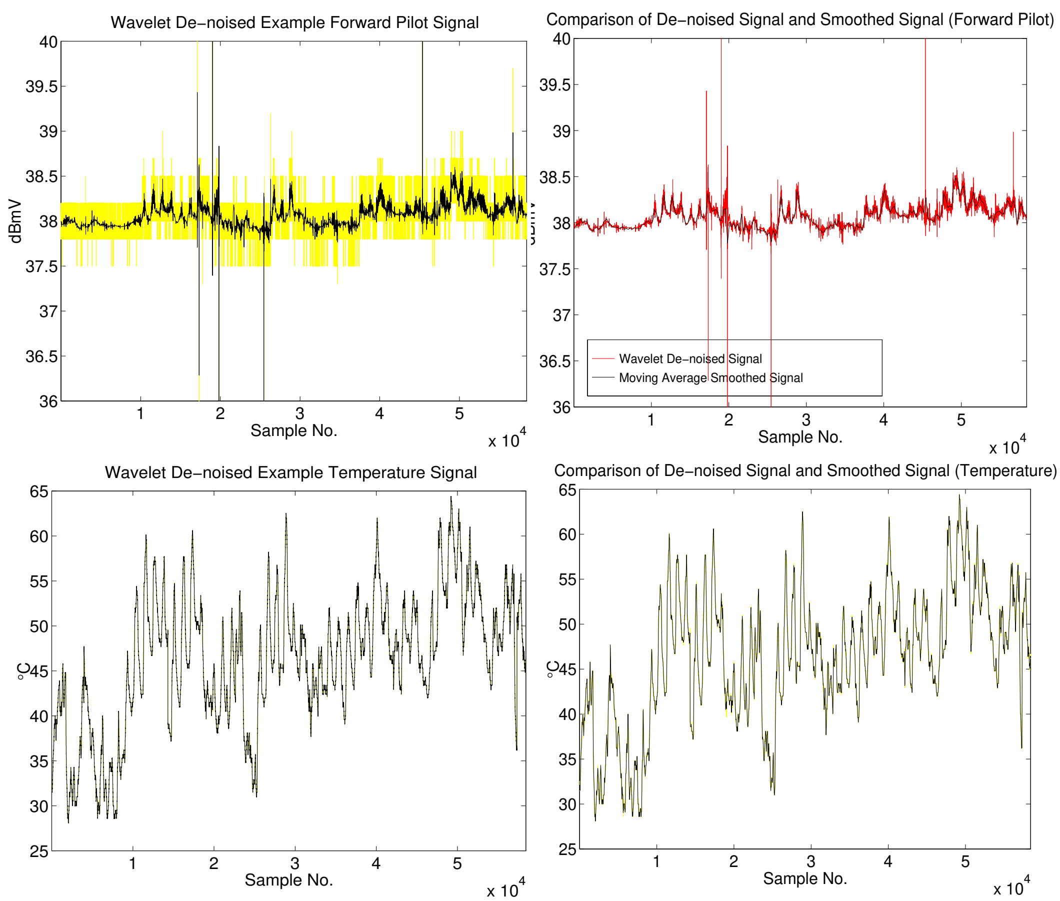


FIGURE 5-16.
Best wavelet de-noising methodologies for the forward pilot and temperature signal classes (forward pilot - 10^{th} order Daubechies with cross-validation and soft thresholding, temperature - 4^{th} order Coiflet with cross-validation and soft thresholding)

5.2.3.3 Correction for Known Non-Gaussian Noise Effects

In the previous section, the methodologies of de-noising the class of signals present in the example plant were identified subject to the minimization of Eq. 5.35 and the assumption of additive $N(0,\sigma)$ Gaussian noise contamination. Obviously, as was discussed in Chapter 3, due to the utilization of coarse quantization functions, the assumption of Gaus-

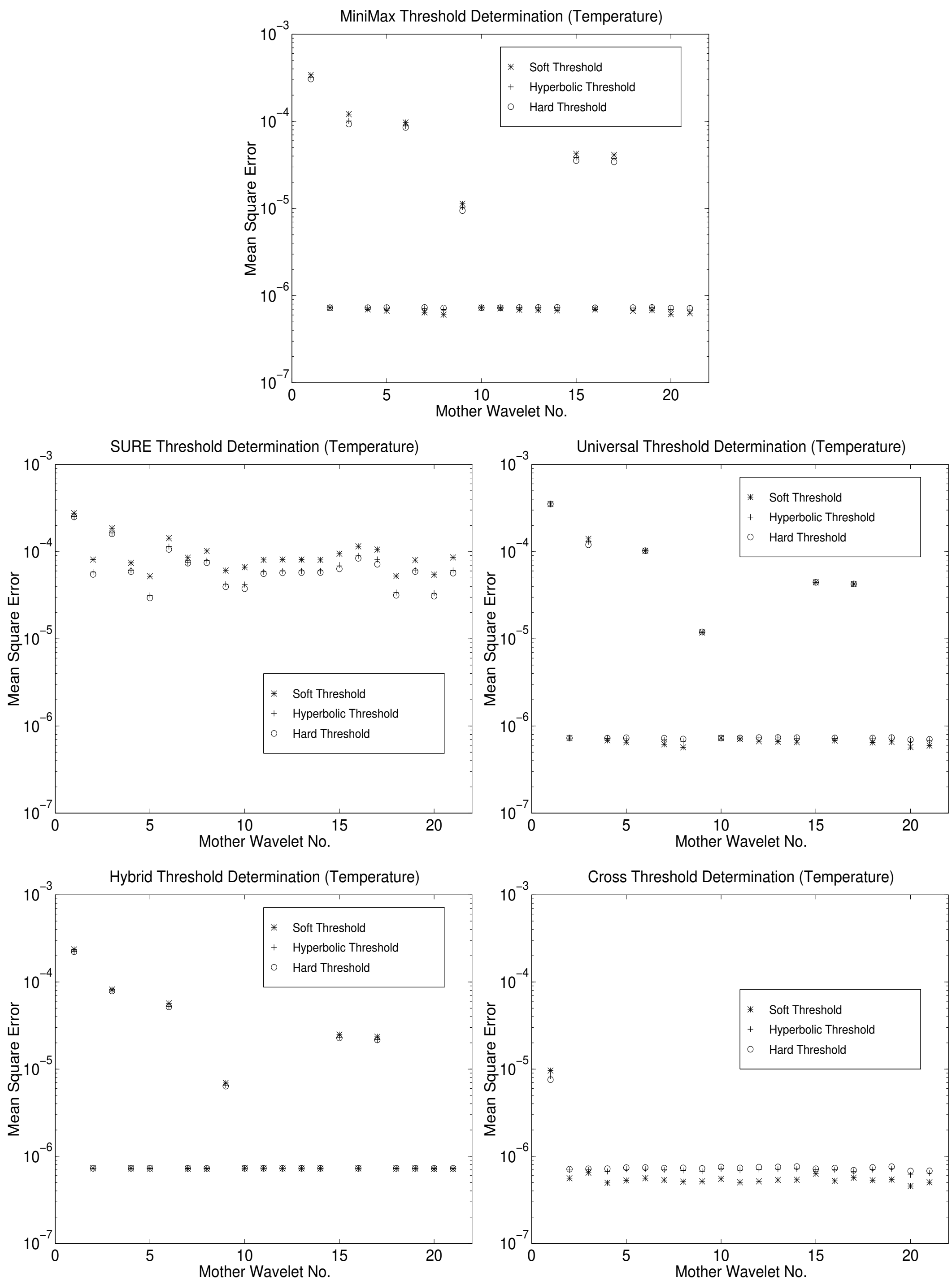


FIGURE 5-15.
Evaluation of the wavelet de-noising methodologies for the set of 75 randomly selected, one month duration, enclosure temperature data sequences (the x-axis within these plots corresponds to the mother wavelet function in the order in which they were presented in Table 5.1).

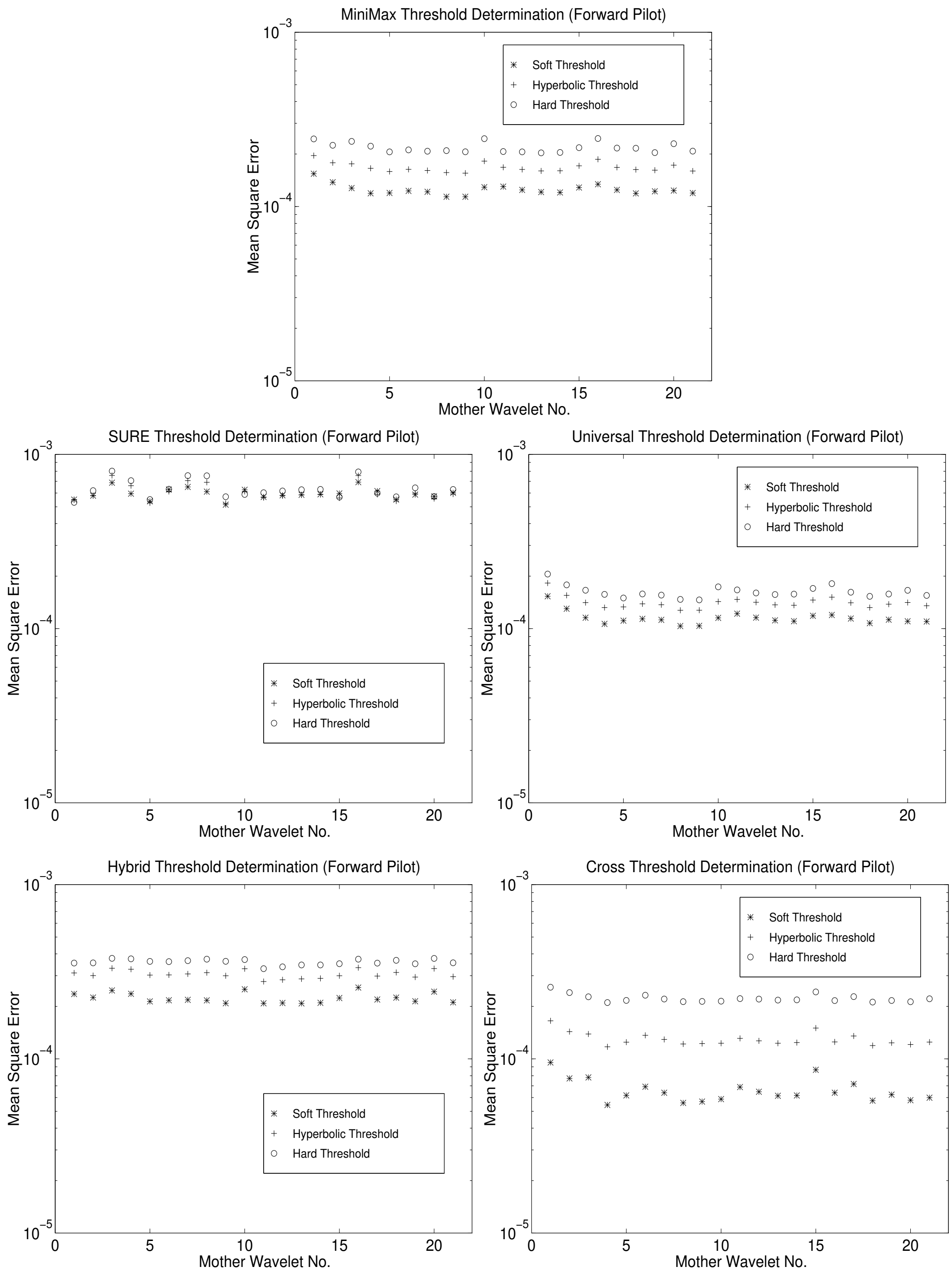


FIGURE 5-14.
Evaluation of the wavelet de-noising methodologies for the set of 75 randomly selected, one month duration, forward pilot data sequences (the x-axis within these plots corresponds to the mother wavelet function in the order in which they were presented in Table 5.1).

monly used orthogonal mother wavelet functions. What is desired within this work is, therefore, to determine which of these mother wavelet functions provides the “best” wavelet de-noising for the class of signals which occur in the example plant. Identifying or developing a mother wavelet function which was optimal in a de-nosing sense for these classes of signals was deemed outside the scope of this work.

From these figures it is clear that, in general, for the forward pilot signal soft thresholding out performs the other thresholding methodologies, and that, in general, cross thresholding provides a threshold determination methodology that results in de-noised signals which are closest to those generated by the moving average smoothing technique, at the data points where the moving average is computed. For the temperature signal, there is significantly less variation in the de-noising approaches, which is to be expected since the noise effects are considerable less severe within this signal class. Once again though, cross-validation with soft thresholding produces the underlying signal estimates closest to those produced through the moving average technique. Hence, in both cases cross-validation with soft thresholding was chosen as the de-noising methodology.

In terms of the mother wavelet functions, it can be clearly seen, for the forward pilot signal class, under cross-validation with soft thresholding, that utilizing a 10^{th} order Daubechies wavelet function results in the minimum average means square error. For the temperature signal class, under cross-validation with soft thresholding, there is very little variation, excluding the first three Daubechies wavelets, in the average mean square error level caused by varying the mother wavelet function that was utilized. The 4^{th} order Coiflet wavelet, though, does provide a slightly reduced average mean square error value. Figure 5-16 shows the de-noising results obtained utilizing cross-validation with soft thresholding for the de-noising of the example status data signals of Figure 5-6, utilizing 10^{th} order Daubechies and 4^{th} order Coiflet mother wavelets for the forward pilot and temperature signals respectively. It should be noted, that the de-noised temperature signal is almost identical to both the raw temperature signal and the moving averaged temperature signal at the scale shown in the Figure 5-16. Due to the low quantization and underlying noise effects which are present on the enclosure temperature signal little additional infor-

selection of the de-noising methodology, with the exception of the threshold determination, will be done in a one time only manner for each of the signal classes. Hence, since the specific nature of a given data sequence is not be known *a priori*, a methodology must be selected which performs adequately across a range of the given signal class's behaviour without necessarily being the optimal approach for any given data sequence from the class.

Table 5.1: Mother wavelet functions utilized in de-noising evaluations.

Mother Wavelet	Order
Daubechies	4,6,8,10,12,14,16,18,20
Symmlet	4,5,6,7,8,9,10
Coiflet	1,2,3,4,5

To this end, a set of 75 one-month data sequences were selected at random from the example plant's available historical data set. These example data sequences were then normalized, in accordance with the normalization procedure utilized in the recurrent neural network modeling, to remove any biased effects introduced by variations in the sequences' dynamic ranges. Each of the permutations of the previously described de-noising techniques were then evaluated in accordance with Eq. 5.35. Figure 5-14 and Figure 5-15 show the results of this evaluation procedure, where each of the subplots corresponds to a particular threshold determination methodology. In each of these figures, the values plotted represent the average mean square value of the objective function obtained across the complete set of mother wavelet functions (given in Table 5.1) for the given de-noising methodology. The x-axis of the plots corresponds to the sequence of mother wavelet functions in the order in which they are presented in Table 5.1. Basically, the x-axis corresponds to the wavelet functions in ascending function order, starting with the Daubechies family and ending with the Coiflet family. These wavelets represent the complete set of orthogonal wavelets available within the WaveLab^{tm} Matlab^{tm} wavelet toolbox, which was the toolbox employed for the wavelet material presented within this work. As such this set of wavelet functions is not complete, though it does represent most of the com-

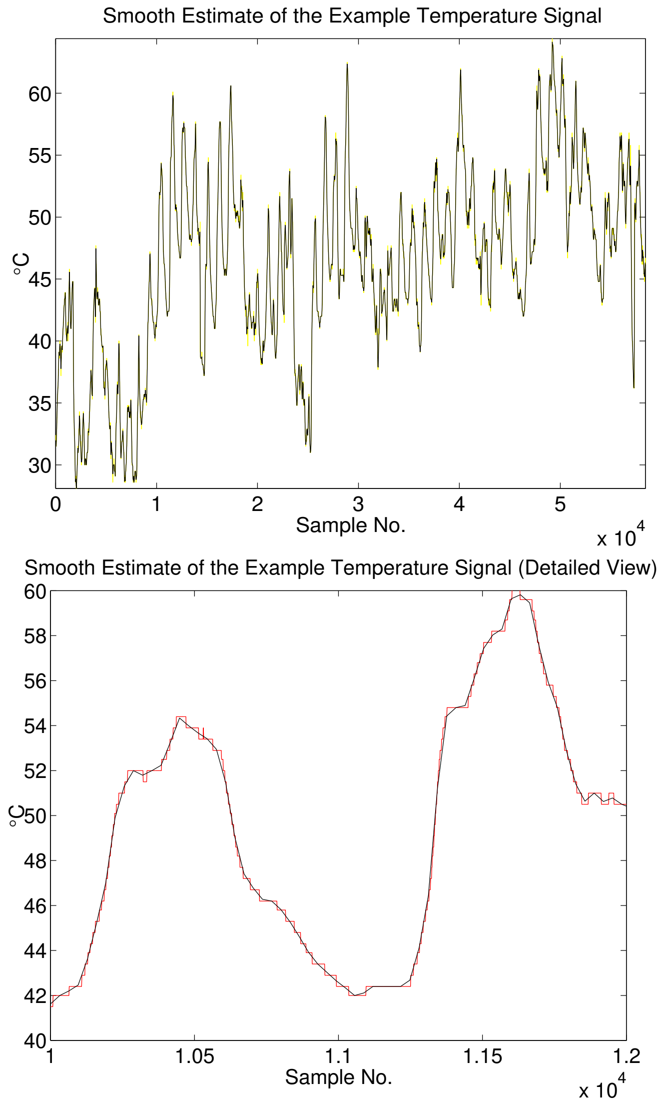


FIGURE 5-13.
Smoothed version of the example enclosure temperature signal of Figure 5-6.

5.2.3.2 Comparison of Wavelets and Thresholding Techniques

Obviously, to be useful in a real-time setting, the de-noising technique selected should be the one which performs optimally or near optimally across a range of example sequences from the given class of status data signals. A given de-noising technique may perform exceptionally well for a given data sequence but perform poorly for other sequences selected from the same signal class. It is assumed within this work, that the

The window width which should therefore be selected is the one which “best” balances these conflicting criteria of noise reduction and signal detail preservation. More particularly, there exists a relatively flat area (between the window widths of 2^8 to 2^{10}) in Figure 5-11 which relates to window widths which provide a fairly good balance between these competing criteria, in that most of the noise effects are removed but not too much of the signal detail has been removed. In particular, the window width of 2^8 was chosen because it occurred at the knee of the plot and hence it preserved the most signal detail while still removing a large portion of the noise. An example of the application of the moving average smoothing process, width the window width set to 2^8, to the example forward pilot signal of Figure 5-6 is shown in Figure 5-12. It should be noted that this smooth underlying signal estimation methodology make no assumptions about the nature of the noise contamination of the status data signals.

For the enclosure temperature signal though, the window width of 256 sample produced a signal that was too “smooth” in the sense that the signal no longer appropriately followed the enclosure temperature status data signal. In particular, due to the very low underlying noise levels on this signal a bound can be placed on where the actual underlying temperature signal must have been, in the sense that the true underlying temperature must have been within one-half a of a quantization step of the recorded status data signal. Hence, by offsetting the measured signal by plus and minus one-half of one quantization step width, upper and lower bounds can be placed on the location of the true underlying temperature signal. The moving average window width was then reduced until an estimate of the signal’s behaviour was produced that was within these known bounds, which occurred for the width of 32 samples. An example of the application of this smoothing process to the example enclosure temperature signal of Figure 5-6 is shown in Figure 5-13. In both cases the window width was selected as a power of two to avoid any boundary effects associated with the power of two signal record length required by the wavelet transformation process.

characteristics. If the window width was made too small then too much of the noise would have been preserved. If the window width was made too large then the fine detail of the signal would have been lost. Figure 5-11 shows a plot of the effect that varying the window width has on the mean square error between the raw forward pilot signal of Figure 5-6 and the resulting moving average signal estimate. It is quite clear from Figure 5-11 that, for window width of less than 2^7, a significant amount of the signal noise is preserved in the moving average estimate which results in the declining mean square error. In the extreme case, a window width of 1 will result in a mean square error of zero since the raw data signal, including its noise, will be perfectly estimated. It is equally clear from the figure that, for window widths of greater than 2^{10}, the mean square error begins to increase quickly due to the loss of signal detail in the moving average estimate. In the extreme case, the window width would be equal the data record length and the signal will be estimated by a single point, resulting in the loss of all trend information contained in the signal.

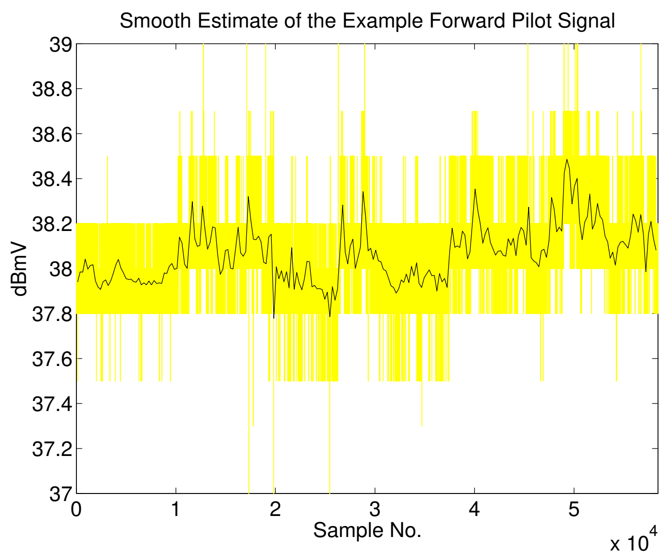


FIGURE 5-12.
Smoothed version of the example forward pilot signal of Figure 5-6.

daily temperature variations. Because the daily variations contain higher frequencies than the seasonal variations, a window width which is appropriate to track the daily variations will also track the seasonal temperature variations.

Each day approximately 1000 to 1200 data samples are collected from each of the status monitored amplifiers. Hence, if a moving average window width of 256 samples is employed then the daily temperature variations will be tracked, since this window width is small enough to exceed the Nyquist sampling criteria for the diurnal variations. The use of a larger window width would result in too few moving average samples being generated per day to allow the reconstruction of the daily variations; a smaller window width would cause the moving average to become overly restrictive in terms of its influence on the wavelet de-noised underlying signal estimates.

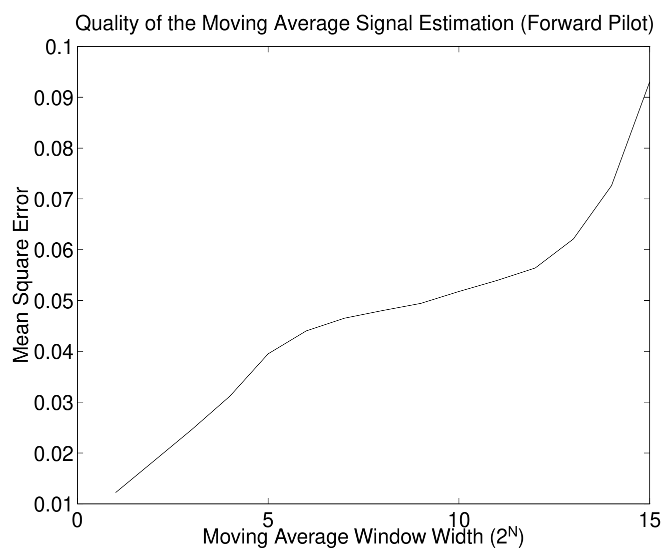


FIGURE 5-11.
The effect of the window width on the estimation ability of the moving average signal estimation technique for the example forward pilot status data signal of Figure 5-6.

The second reason for the particular forward pilot moving average window width selection had to do with the desire to generate a signal estimate that was as noise-free as possible while still being as detailed an estimate as possible. This required that a balance be achieved between removing noise from the signal estimate and preserving its detailed

$\{x_{avg}(nT) \mid n = 1, \ldots, N\}$ which can be compared in a mean square error sense to the equivalent samples, $\{\hat{x}_W(nT) \mid n = 1, \ldots, N\}$ produced by the various wavelet de-noising approaches.

$$J = \frac{1}{2} \sum_{n=1}^{N} (x_{avg}(nT) - \hat{x}_W(nT))^2 \tag{5.35}$$

In both cases, the moving average windows were chosen to be non-overlapping to improved the computational efficiency of the optimization procedure. This resulted in the moving average signal estimates effectively becoming sub-sampled. Hence, the actual cost functions that were utilized were the sub-sampled cost functions given by

$$J = \frac{1}{2} \sum_{n=1}^{\frac{N}{\alpha}} (x_{avg}(\alpha nT) - \hat{x}_W(\alpha nT))^2 \tag{5.36}$$

where $\alpha = 256$ in the case of the forward pilot signals and $\alpha = 32$ in the case of the enclosure temperature signals.

There were two primary reasons behind the forward pilot window width selection. The first reason had to do with the fact that within the example plant the principle status data signal variations are caused by variations in the amplifier's enclosure temperatures. As have been seen in the previous chapters, the temperature signals themselves have significant variations due to both daily and seasonal temperature variations. Obviously, since the proposed fault detection systems utilize the temperature signal as their input, it is desirable that the moving average signal estimate be able to track these seasonal and daily temperature variations. At the same time though the moving average signal estimate should not restrict the wavelet de-noised estimates in how they generate smooth estimates of the status data signals' variations which occur at higher frequencies than those caused by the

3. The de-noising approach which minimizes the objective function over the given signal class will be selected as the given class' de-noising methodology. (Section 5.2.3.2)

4. A procedure to correct for the Gaussian noise assumption inherent in the selected de-noising technique will then be identified. (Section 5.2.3.3)

5. The resulting underlying signal estimate will then be used to re-construct an estimate of the original raw data signal, under the assumption of Gaussian underlying noise. (Section 5.2.3.4)

6. This re-constructed raw data signal will then be compared statistically to the original raw data signal to validate that the de-noising methodology produces a "reasonable" underlying signal estimate under the assumptions from which it was developed. (Section 5.2.3.4)

5.2.3.1 Objective Function Determination

In order to be able to quantify the "goodness" of a given wavelet de-noising approach, some process by which the "closeness" of the resulting estimate of the underlying signal to the true underlying signal is needed. Unfortunately, in the case of data obtained by the type of status monitoring systems of interest within this work, the true underlying signal is typically unknown. This is particularly true when limit checking thresholding techniques are in place due to their typical utilization of coarse quantization functions in the status data sampling process. One method available in this situation, is to analyze the statistics of the raw status data signal transitions in order to generate an unrelated estimate of the underlying signal. In particular, a moving average sequence can be constructed with a suitable window width such that the average variations of the underlying sensor signal, indicated by the changing probability distribution of adjacent quantization levels, can be tracked. Within this work, a window width of 256 samples was selected for the forward pilot signal and a width of 32 samples was selected for the temperature signal. The moving average approach therefore generates a sequence of moving average samples

signal $y^{even} = \{y_i^{even} = \frac{y_{2i} + y_{2i+2}}{2}, \quad i = 1, \ldots, \frac{N}{2}\}$, $T_{soft}(d, \lambda)$ is the soft thresholding function given Eq. 5.26 with $\delta = \lambda$, and $\{y_i, i = 1, \ldots, N\}$ is the time domain signal to be de-noised.

Once the value of λ which minimizes $M(\lambda)$, denoted λ^*, is determined through an optimization procedure then the threshold for the de-noising operation can be calculated as

$$\delta = \lambda^* \left(1 - \frac{\log(2)}{\log(N)}\right)^{1/2} \tag{5.34}$$

The above optimization is not particularly onerous since it has been shown in [63] that $M(\lambda)$ is a convex function of λ. A complete description of the cross-validation methodology for threshold selection can be found in [63].

5.2.3 Underlying Sensor Signal Estimation

As mentioned previously, the determination of which of the above wavelet functions, thresholding techniques, or threshold determination methodologies should be employed to provide the "best" estimation of the underlying signal for a particular class of noise contaminated signals is currently an open research area. Hence, the determination of what constitutes a "good" technique for a particular class of noisy signals requires that the permutations of the above techniques and methodologies be explored for example signals of the given class. Additionally, there is no theoretical basis available to prove that the resulting underlying signal estimate is in fact a "good" estimate.

Therefore, the approach that will be taken in this section to first identify an appropriate de-noising methodology and then to validate that the methodology produces "reasonable" underlying signal estimates is as follows:

1. An independent methodology of arriving at an underlying signal estimate will be identified. (Section 5.2.3.1)

2. A objective function will be specified, based on this independent underlying signal estimate, which will be used to compare the above de-noising approaches. (Section 5.2.3.1)

SURE threshold determination methodology when enough non-zero wavelet coefficients are present, within the given wavelet scale level, and operates according to the universal thresholding methodology when there is a dearth of non-zero coefficients. The exact condition under which the switch between universal and SURE thresholding occurs can be found in [30]. This technique attempts to balance the results obtained by minimizing the risk associated with the setting of a given threshold level (SURE thresholding) and optimizing the "visual smoothness" of the resulting signal estimate (Universal thresholding) [30].

5.2.2.2.4 MinMax Threshold

Like SURE thresholding, MinMax thresholding is a methodology of setting the threshold such that the risk associated with the given setting is minimized. The difference between the two being that in this case the risk is minimized through a minimax optimization procedure. The exact formulation of the MinMax thresholding methodology can be found in [32].

5.2.2.2.5 Cross-Validation Threshold

Cross-validation is a slightly more involved technique of threshold determination, though it is an optimal technique in the sense that the threshold determination is based on the minimization of the cost function given by

$$M(\lambda) = \sum_{j,k}\left[T_{soft}\left(d_{j,k}^{even}, \lambda\right) - d_{j,k}^{odd}\right]^2 + \sum_{j,k}\left[T_{soft}\left(d_{j,k}^{odd}, \lambda\right) - d_{j,k}^{even}\right]^2 \tag{5.33}$$

where $d_{j,k}^{even}$ are the wavelet coefficients associated with the j^{th} scale level of the wavelet decomposition of the signal $y^{odd} = \{y_i^{odd} = \frac{y_{2i-1} + y_{2i+1}}{2},\ \ i = 1, \ldots, \frac{N}{2}\}$, $d_{j,k}^{odd}$ are the wavelet coefficients associated with the j^{th} scale level of the wavelet decomposition of the

5.2.2.2.2 SURE Threshold

The SURE thresholding technique is based on Stein's unbiased estimate of risk [77] which allows the risk given by

$$R(\hat{y}, y) = E\left[\frac{1}{N}\sum_{k=0}^{N}(\hat{y}(kT) - y(kT))^2\right] \tag{5.30}$$

associated with selecting a given estimator $\hat{y}$ of a given quantity y to be minimized. Within the wavelet de-noising context this can be viewed as finding the de-noising thresholds which minimize the risk associated with resulting de-noised signal estimate. Under this threshold determination method, a separate threshold is determined for each of the wavelet scales and it is applied to filter the coefficients independently of the thresholds utilized at the other scales. The exact formulation of the threshold under this methodology can be found in [30] and is given by

$$\delta_j = \min \arg_{\delta \geq 0} SURE(\delta; \boldsymbol{d}_j) \tag{5.31}$$

where

$$SURE(\delta, \boldsymbol{d}_j) = 2^j - 2 \cdot \#\{k : |d_{j,k}| \leq \delta\} + \sum_k \min(d_{j,k}, \delta) \quad , \quad k = 1, \ldots, 2^{j-1} \tag{5.32}$$

and $\boldsymbol{d}_j$ is the set of wavelet coefficients associated with scale level j, *#{.}* is the number of elements within the set which satisfy the given boolean condition, and *min(x,y)* is a functional operator which equals x if $x \leq y$ and y otherwise.

5.2.2.2.3 Hybrid Threshold

One problem with the SURE threshold determination technique is that it is based on an analysis of the statistics of the wavelet coefficients within the given scale level. If only a relatively few of the coefficients at a given scale are non-zero then the threshold value obtained by the SURE estimate tends to be biased. A solution to this problem, presented in [30], is to utilize a hybrid thresholding technique which operates in accordance to the

where N is the length of the time domain data sequence and $\hat{\sigma}$ is an estimate of the variance of the noise present on the signal. Obviously, with this technique, some means of estimating the noise variance of the signal is required. Fortunately, a reasonable estimate of the noise variance can be obtained, under the assumption that the underlying signal is bandlimited and affected by $N(0,\sigma)$ Gaussian noise [30], by

$$\hat{\sigma} = \frac{1}{\left(\frac{N}{2} - 1\right)} \sum_{k=1}^{\frac{N}{2}} \left(d_{N,k} - \bar{d}\right)^2 \tag{5.29}$$

where $\{d_{N,k} \mid k = \left(1, \ldots \frac{N}{2}\right)\}$ is the set of wavelet coefficient associated with the highest scale (detail level) of the multi-resolution analysis and $\bar{d} = \frac{2}{N} \sum_{k=1}^{\frac{N}{2}} d_{N,k}$. In essence, because the signal is assumed to be bandlimited then it can also be assumed that only noise effects are present at the highest scale of the multi-resolution analysis. The other lower scale levels are also affected by noise but they also are likely to contain effects due to the underlying signal. Only at the highest scale level is it likely that the wavelet coefficients are only related to the noise effects. Obviously, the data records to be analyzed must be chosen such that they are long enough that the information in their highest scales is only due to the noise effects[9]. Hence, the variance at this level can be used to give an estimate of the variance of the noise within the time domain signal, provided that the noise present in the system is Gaussian, since it has been shown in [30] that the noise variance is preserved by the wavelet transform. This thresholding technique was designed to attempt to optimize the "visual smoothness" of the resulting de-noised signal estimate [30].

---

9. It can be recalled from Section 5.2.1 that a data record of length 2^N can be decomposed by the wavelet transform into N scales (labeled $0,\ldots,N\text{-}1$) each of 2^k, $k = 0,\ldots, N\text{-}1$ elements. The highest scale for a 2^N length data record will therefore be at scale $N\text{-}1$.

5.2.2.1.3 Hyperbolic Thresholding

Hyperbolic thresholding, described in [81], attempts to address the perceived limitation of soft thresholding by decrementing the coefficients associated with the underlying signals in a manner that considers the relative distance between the threshold and the coefficients which exceed it, as measure by the Euclidean norm. Mathematically, this thresholding function is given by

$$y_{hyperbolic}(t) = \begin{cases} \operatorname{sgn}(x(t))\left(\sqrt{x(t)^2 - \delta^2}\right) & if\ |x(t)| > \delta \\ 0 & if\ |x(t)| \le \delta \end{cases} \tag{5.27}$$

For values of *x(t)* whose magnitudes are much larger than δ, this thresholding technique produces results, for the given coefficients, which are very similar to those produced by hard thresholding.

5.2.2.2 Methodologies of Threshold Determination

Obviously, each of the given thresholding methods presented above require some means of determining values for δ prior to their application. Not surprisingly, there are also a number of different methodologies available within the literature to perform this task. The following four sub-sections will overview some of the more generally applied methods. The first four methods presented are based on the work described in [30], [31], [32], and [33]. The final method is based on the work described in [63].

5.2.2.2.1 Universal Threshold

In universal thresholding, the threshold, δ, is set such that it is proportional to the variance of the signal's noise. In particular,

$$\delta = \sqrt{2\log(N)}\,\hat{\sigma} \tag{5.28}$$

5.2.2.1.1 Hard Thresholding

The simplest thresholding technique, and the one presented in Figure 5-9, is hard thresholding. In this technique, all coefficients whose absolute values are below the threshold value δ are assigned zeros. Mathematically, this thresholding function is given by

$$y_{hard}(t) = \begin{cases} x(t) & if\ |x(t)| > \delta \\ 0 & if\ |x(t)| \le \delta \end{cases} \tag{5.25}$$

Obviously, as discussed above, this thresholding technique does not attempt to address the noise associated effects which occur for the coefficients associated with the underlying signal.

5.2.2.1.2 Soft Thresholding

Soft thresholding, described in [31], is slightly more involved than hard thresholding and attempts to address the noise associated effects on the underlying signal coefficients by decrementing the absolute values of all the coefficients by δ and then assigning zeros to those coefficient whose absolute values then fall below zero. Mathematically, this thresholding function is given by

$$y_{soft}(t) = \begin{cases} \mathrm{sgn}(x(t))\,(|x(t)| - \delta) & if\ |x(t)| > \delta \\ 0 & if\ |x(t)| \le \delta \end{cases} \tag{5.26}$$

One perceived limitation to this thresholding technique is that it decrements the coefficients equally without regards to the relative distance between the coefficient and the threshold.

sections, some of the general techniques available in the literature to perform these tasks will be presented. Currently, no general theory is available which identifies which wavelet function, thresholding technique, or threshold determination methodology should be employed to perform wavelet de-noising on a given type or class of signals. Hence, for any given signal class, a comparison of the various available techniques and their permutations must be undertaken if the optimal or near optimal wavelet de-noising approach is to be identified. Therefore, the subsequent sections are intended to provide an overview of the available techniques prior to their evaluation for de-noising of the coarsely quantized signals present in the example plant (to be presented in Section 5.2.3).

5.2.2.1 Thresholding Techniques

There are three main thresholding techniques available in the literature to perform the thresholding operation: hard thresholding, soft thresholding, and hyperbolic thresholding. They differ principally in how the coefficients associated with the underlying signal are processed. The thresholding example presented in Figure 5-9 was a simplification, in the sense that it was assumed that the coefficients associated with the underlying signal were unaffected by the presence of the noise. In practice, this is not typically the case. The underlying signal coefficients will be perturbed to some extent by the underlying noise effects; hence, within the literature [30][31][32][62], thresholding techniques to compensate for this effect have been developed. The following three sections will present the mathematical descriptions of each of the three commonly used thresholding techniques in turn, the latter two of which provide methods of addressing the noise associated effects on the retained coefficients. Each of these thresholding technique can be applied either to a single scale level or to the wavelet transform as a whole. Within this work, the thresholding techniques are applied to each scale individually with separate thresholds being generated at each scale level. For simplicity though, the thresholding functions will be introduced without reference to the thresholds' scale subscript.

domain, identifying the appropriate thresholding level for each of the scales, performing thresholding operations, and then converting the signal back to time domain, utilizing the inverse wavelet transform (Figure 5-10).

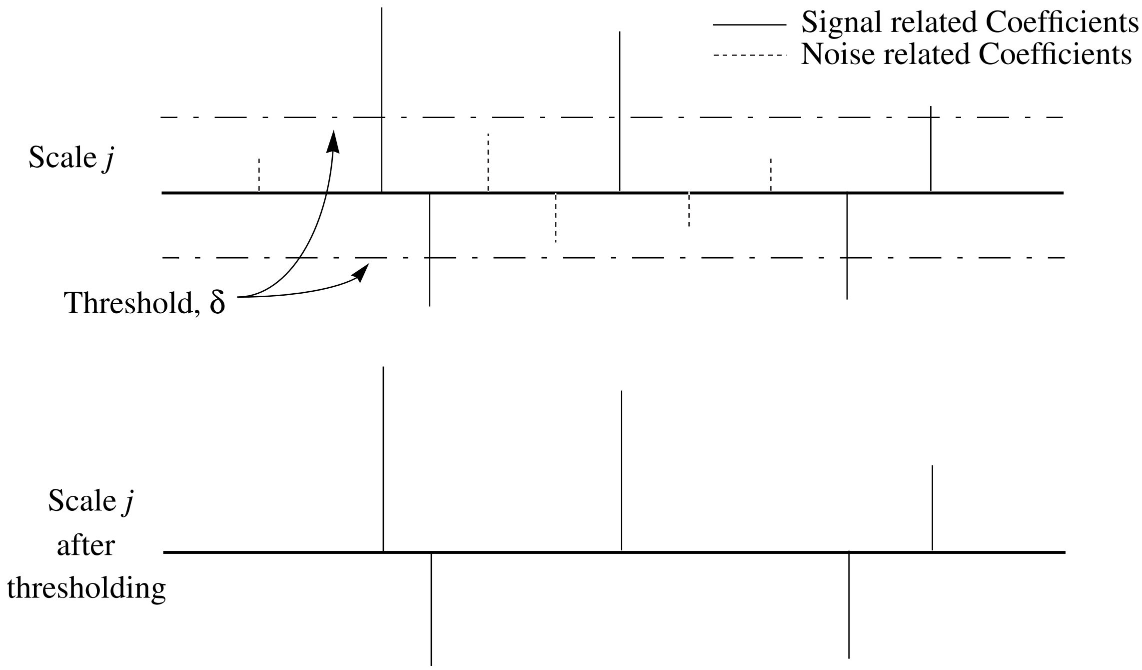


FIGURE 5-9.
Wavelet de-noising process for the wavelet coefficients at scale j.

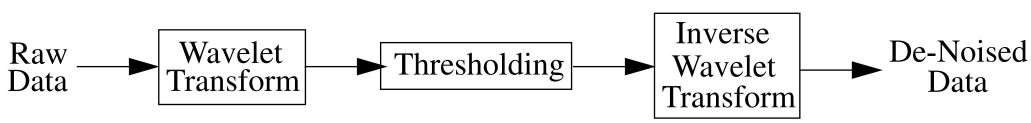


FIGURE 5-10.
Block diagram of the wavelet de-noising process.

The process of wavelet de-noising, once an appropriate mother wavelet has been selected, is therefore concerned primarily with methods of obtaining the threshold and applying the threshold such that the noise and signal effects can be separated from each other at each of the wavelet scales. In the following two sections, and their associated sub-

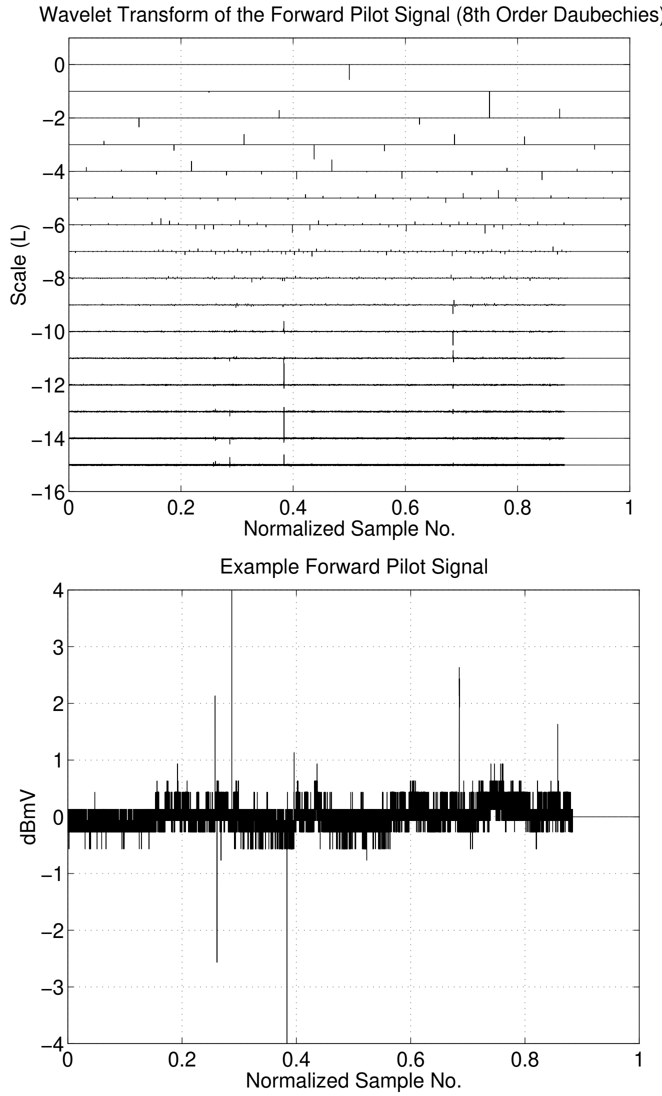


FIGURE 5-8.
Time domain and wavelet transform of the example forward pilot signal showing the transient events at the normalize time locations of approximately 0.3, 0.4, and 0.7.

free estimate of the original signal, while simultaneous preserving any short duration, high frequency signal events. Figure 5-9 illustrates the general process of wavelet de-noising for a set of the wavelet coefficients generated for a particular scale. The complete wavelet de-noising process therefore becomes one of transforming the input signal to the wavelet

over shorter time periods. In each case the significant coefficients relate to areas in the signal where there were significant transient events (approximately at the 0.3, 0.4 and 0.7 time periods on the normalize time scale (Figure 5-8)). The temperature signal transform shows no such events for the higher scale levels, indicating that no such transients exist within the associated the time domain signal. In addition, the forward pilot transform of Figure 5-5 can be compared with the wavelet transform of the previously obtained neural network model, shown in Figure 5-7. From this comparison, it is clear that, in general, the higher scale levels are associated with the majority of the quantization and underlying noise effects present in the forward pilot signal. Hence, the wavelet de-noising process will involve techniques that will allow these noise associated coefficients to be detected and isolated from the underlying signal dependent coefficient prior to the inverse transformation of the data sequence.

5.2.2 Overview of Wavelet De-Noising

The general approaches developed in the literature[8] for wavelet de-noising assume that the signal to be de-noised consists of two parts:

1. A relatively slowly varying signal which is assumed to be the signal to be recovered, and
2. Noise, which is assumed to be Gaussian in nature.

If these assumptions are combined with the knowledge that the given signal's signal-to-noise ratio is sufficient to allow an estimate of the underlying signal to be produced, then the absolute value of the coefficients, at each scale, related to the signal will be larger in magnitude than those related to the noise. This is the general assumption utilized within the wavelet de-noising process. Under this assumption, de-noising can be accomplished by finding an appropriate thresholding value such that the noise related coefficient's absolute values fall below the threshold and the signal coefficient's absolute values fall above the threshold. Once the threshold is determined, all coefficients whose absolute value is below it are set to zero and the inverse transform is performed, creating an ideally noise-

8. The works being referred to within this section are those of [24], [30], [31], [32], [33], [63], [78], and [82].

in terms of the negative of the scale number. Hence, for these plots time resolutions increase as the plotted scale number decreases. In addition, the wavelet coefficients within each scale have been normalized with respect to the largest coefficient occurring within the given scale. Hence, the relative magnitude of the coefficients between scales is not shown within these plots. In general, for most real-world signal, there is a trend for the absolute magnitude of the coefficients to decrease considerably as one moves from the coarse scales to the finer scales. Finally, due to the dyadic shifting which occurs within the wavelet transform, the original signals of Figure 5-6 have been padded with zeros such that their sequence lengths are integer powers of 2.

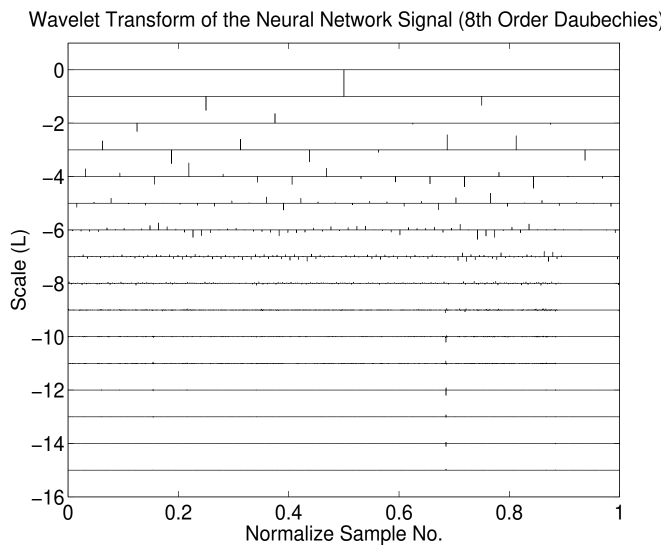


FIGURE 5-7.
Wavelet transform of the recurrent neural network signal model of Figure 5-2 utilizing and 8^{th} order Daubechies mother wavelet function.

From these plots, some facets of the wavelet transform's signal analysis capabilities can be seen. In particular, the forward pilot signal's transform contains three areas, associated with fine resolution scales ($L < -9$), for which there are significant wavelet coefficients, in the sense that at the given scale there exist several coefficients that are considerably large than the other coefficient at that scale. The presence of these coefficients demonstrates the wavelet ability to isolate higher frequency events which occur

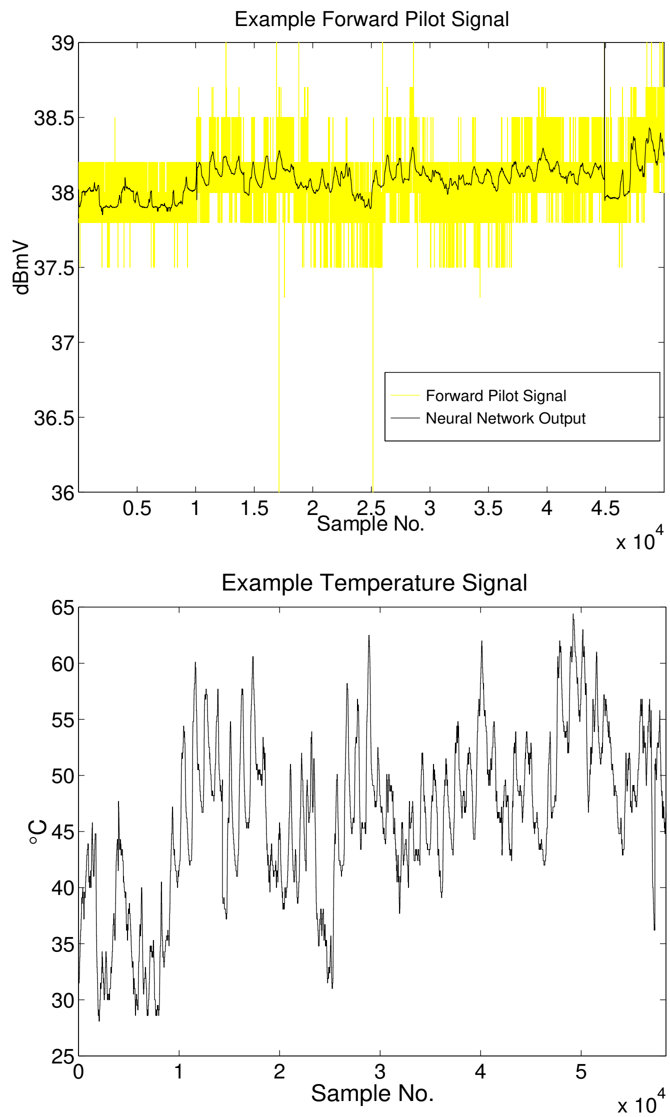


FIGURE 5-6.
Forward pilot and enclosure temperature status data signals from Figure 3-1.

hence, for example, the coefficient at $L = 0$ occurs at the centre of the time axis. These introduced zeros are an artifacts of the plotting process and should not be viewed in any way as coefficients resulting from the wavelet transform. Also, within these plots the time resolution of the scale increases as the plotted scale number decreases. Hence, $L = 0$ represents the coarsest scale, and $L = -16$ represents the finest scale. This convention of plotting the finer scales below the coarser scales results in the necessity of labeling the y-axes

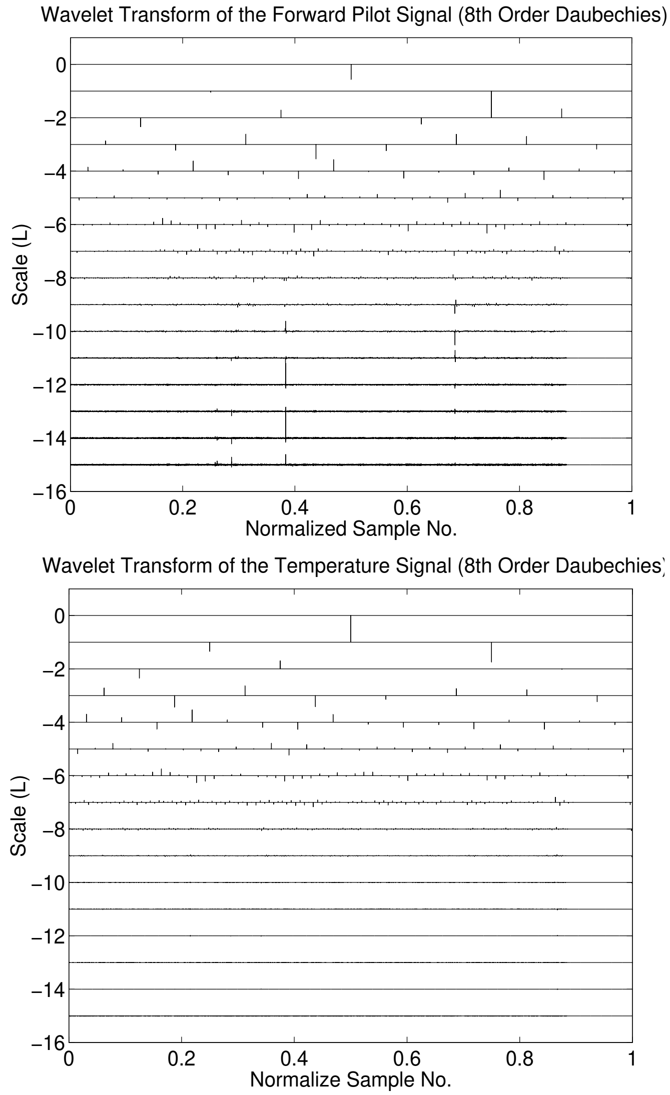


FIGURE 5-5.
Example wavelet transforms of the example forward pilot and temperature signals of Figure 3-1 utilizing an 8^{th} order Daubechies mother wavelet function.

ficient, at the second level there are two coefficients, and at the last level there are 2^{15} coefficients. The number of levels (scales) that the signal can be decomposed into is therefore a function of the length of the given data sequence. In order to preserve the relative location of the coefficients, with respect to the time domain instance of the signal, each scale level within the plots have been arbitrarily padded with zeros at the locations, within that particular scale, which do not correspond to actual wavelet coefficients at that scale;

A graphical comparison between the wavelet based multi-resolution analysis and the short-time Fourier Transform can therefore be made, and is shown in Figure 5-4. In the short-time Fourier transform, the given signal is partitioned in time into a set of non-overlapping blocks through the use of a windowing function which by definition must be a windowing function in both time and frequency domains [20][7]. Each of these blocks is then Fourier transformed, generating the frequency content of the blocks. Equivalently, this process can be viewed as generating the inner product of the signal with modulated versions of the windowing function, generating the characteristic constant time-frequency tiling shown in right hand side of Figure 5-4 [79]. This modulation effect can be seen more clearly by analyzing the short-time Fourier transform expression

$$STFT(\omega, \tau) = \int_{-\infty}^{\infty} e^{-j\omega t} w(t-\tau)\, g(t)\, dt = \langle w_{\omega, \tau}(t), g(t) \rangle \tag{5.24}$$

The short-time Fourier transform therefore has the effect, due to its use of a fixed windowing function [79], of fixing both the time and frequency resolution throughout the analysis. Once again, as in the wavelet case, the area of the resulting time-frequency tiles are bounded from below by Heisenberg's uncertainty principle [20]. With the wavelet transform, though, the time and frequency resolution of the tiles are not fixed and in fact vary allowing resolution in one domain to be traded off for better resolution in the other. The utilization of dyadically scaled wavelet basis function produces the characteristic tiling shown in Figure 5-4, with the time resolution doubling and the frequency resolution halving each time at each subsequent scale.

Figure 5-5 shows resulting wavelet transforms for the example forward pilot and temperature signals of Figure 3-1 (which are reproduced in Figure 5-6 for convenience), generated utilizing an 8th order Daubechies mother wavelet function. Within these plots, the number of wavelet coefficients present at each of the scales is given by 2^L, where L is the indicator of the scale level. Hence, at the first scale level, $L = 0$, there is one wavelet coef-

---

7. A non-trivial function $w(x) \in L^2(\Re)$ is called a window function if $xw(x)$ also belongs to $L^2(\Re)$ [20].

basis functions and the given data signal will therefore provide a very high resolution look at the signal's low frequency content but it will poorly localize the occurrence of these low frequencies in time. At high scales the opposite occurs, the mother wavelet is compressed in time to form the high scale basis functions. This increases the resulting wavelet's frequency content, reducing its frequency resolution, content while simultaneously increasing its time resolution. The inner product of these high scale wavelet basis functions and the signal will therefore provide good temporal localization of the high frequency content of the signal at the cost of poorly resolving which high frequencies occurred. Wavelet theory therefore offers a method of trading off frequency resolution for time resolution and *vice versa*. It should be noted that the areas of the resulting wavelet basis functions' time-frequency bands (or tiles) are fixed and equal. The actual area is a property of which mother wavelet function or scale function was selected to perform the multi-resolution analysis and is bounded from below by Heisenberg's uncertainty principle [20].

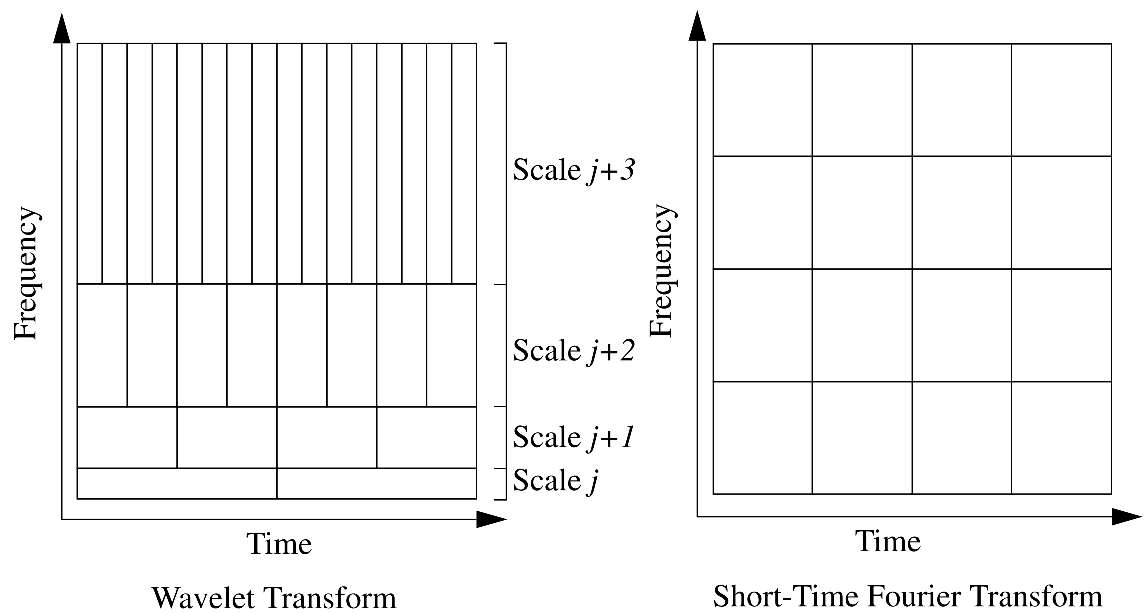


FIGURE 5-4.
Comparison of time-frequency tiling generated by the wavelet transform and the short-time Fourier transform.

each have distinct temporal bands and frequency bands. In particular, all the wavelet functions at a given scale will extract information from the signal in the same frequency band but due to their dyadic shifts they will do so within different temporal bands. Wavelet functions at the next lower scale will extract information from the signal within a lower frequency band and over a broader temporal band. If orthogonal wavelet functions are used, then each of the temporal-frequency bands related to the multi-resolution analysis will be independent and contain no duplicate signal information.

Another feature of wavelet analysis is that the scale function and mother wavelet function are related to one another. More specifically, if the scaling function is given by $\phi(t) \in V_0$, then $\phi(t)$ can be expressed as a linear combination of basis functions of V_1 since $V_0 \subset V_1$, by Eq. 5.12. In particular,

$$\phi(t) = \sum_{k \in Z} h(k)\sqrt{2}\phi(2t-k) \tag{5.21}$$

where

$$h(k) = \langle \phi(t), \sqrt{2}\phi(2t-k) \rangle \tag{5.22}$$

The wavelet basis function, or mother wavelet, $\psi(t)$ can then be defined in terms of $\phi(t) \in V_0$ as

$$\begin{aligned} \psi(t) &= \sqrt{2}\sum_{k \in Z} (-1)^k h(-k+1)\phi(2t-k) \\ &= \sum_{k \in Z} g(k)\sqrt{2}\phi(2t-k) \end{aligned} \tag{5.23}$$

One way to view the effect of the wavelet transform, and particularly the dyadic scaling, is as follows. Each of the wavelet functions contains a set of frequencies which exist over a bounded time period. At low scales the mother wavelet is stretched out in the time domain. This reduces its frequency content, giving it a better frequency resolution, while simultaneously reducing its time resolution. The inner product of these low scale wavelet

resented to a desired degree of accuracy by its component elements from each of the subspaces $V_0, W_0, W_1, \ldots, W_j$. Specifically, $\phi(t)$ forms a basis of V_0, $\psi_{0,0}(t)$ forms a basis of W_0, $\psi_{0,1}(t)$ and $\psi_{1,1}(t)$ form a basis of W_1, etc.; therefore, a time domain signal can be represented in terms of its component elements with respect to each of the bases. More particularly, this relationship can be described by the set of coefficients derived by performing the inner product operation, in turn, between the signal and each of the respective basis functions. The generation of these coefficients constitutes the discrete wavelet transform. The dyadic scaling and dilation/contraction, given by $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$, act to separate the functional space or signal space into orthogonal components. Like the Fourier transform, the wavelet transform is a linear, invertible process. Mathematically, the coefficients associated with the wavelet transform can be given by

$$d_{0,0} = \langle g(t), \phi(t) \rangle \tag{5.17}$$

$$d_{j,k} = \langle g(t), \psi_{j,k}(t) \rangle, \quad j = 1, \ldots, N, \quad k = 1, \ldots, 2^{j-1} \tag{5.18}$$

where $\langle . \rangle$ represents the inner product operation given by

$$\langle f_1(t), f_2(t) \rangle = \left\| f_1(t), f_2(t) \right\| = \int_{-\infty}^{\infty} f_1(t) \overline{f_2(t)}\, dt \tag{5.19}$$

These coefficients can be utilized in the inverse transform to reconstruct $g(t)$ as follows

$$g(t) = d_{0,0}\phi(t) + \sum_{j,k} d_{j,k}\psi_{j,k}(t) \tag{5.20}$$

This decomposition process is therefore very similar to how the Fourier transform utilizes sine and cosine basis functions to separate time domain signals into their constituent frequencies. In this case though, because the wavelet functions have compact support in both the time and frequency domains, the signal is separated into constituent parts which

functional subspaces. The orthogonality of the subspaces allows the decomposition to be performed without redundancy with each subspace "peeling-off" different layers of detail of the signal.

Obviously, if the above decomposition procedure is to be useful in analyzing real signals, then basis functions which span each of the subspaces need to be found. In this way, an arbitrary signal could be described in terms of a predetermined set of basis functions, with each basis function or group of basis functions describing the behaviour of the signal at a particular level of detail or, as more commonly termed, at a particular scale level. In particular the basis function which spans V_0 is termed the scale function and is denoted by $\phi(t)$. The actual nature of this function is determined by various mathematical properties and relations which it must satisfy such that it in fact forms a basis for the V_0 functional subspace [20].

Within wavelet theory the basis functions which span the subspaces W_j are selected in a particular manner. Specifically, if a function $\psi(t)$ is found which meets certain mathematical relations and has certain mathematical properties (most notably compact support in both time and frequency domain, inclusion in $L^2(\Re)$ [6], and $\int_{-\infty}^{\infty}\psi(t) = 0$), then, together with the set of functions formed by its dyadic shifts (shifts by powers of 2), it will form a basis of W_1 [20]. If such a function is found then by Eq. 5.13 and Eq. 5.15, the set of functions formed by the first order time dilation of the W_1 basis functions will form a basis for W_2. Hence the complete set of basis functions for each of the W_j subspaces can be constructed from the initial mother wavelet function through an iterative process.

In particular the complete set of basis function can be described by the family of functions $\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \quad j \in Z, k \in \{1, \ldots, 2^{j-1}\}\}$. Since $\bigcup_{j \in Z} V_j = \bigcup_{j \in Z} W_j$ then by Eq. 5.14 $\bigcup_{j \in Z} W_j$ is dense in $L^2(\Re)$ and, hence, any signal in $L^2(\Re)$ can be rep-

---

6. $L^p(\Re)$ consists of the set of functions *f(t)* for which $\int_{-\infty}^{\infty}|f(t)|^p dt < \infty$.

given subspace. In particular, a functional subspace $W_{j'}$ can be found such that it contains the portion of the subspace V_{j+1} which is not part of V_j. Specifically, W_j can be defined as the orthogonal complement of V_{j+1}, such that

$$V_{j+1} = V_j \oplus W_j \tag{5.15}$$

where $\oplus$ denotes an orthogonal summation (the inner product (or union) of the subspaces V_j and W_j will be zero. This is because W_j is the portion of V_{j+1} which is not contained within V_j). Hence, a given subspace V_N can be decomposed into an initial subspace, V_0, orthogonally summed with a set of subspaces $\{W_1, \ldots, W_N\}$ and, due to the orthogonality of the subspaces, the resulting signal decomposition will also result in $g(t)$ being partitioned into orthogonal components.

$$V_N = V_0 \oplus W_1 \oplus \ldots \oplus W_N \tag{5.16}$$

The preceding process produces a set of orthogonal functional subspaces which, if they are selected appropriately [20], can be used to perform multi-resolution analysis of arbitrary signals. The V_0 subspace will contain information about the signal at the coarsest resolution, and each subsequent W_j subspace provides additional levels of detail; the highest level of detail being provided by the projection of the signal into subspace W_N. Hence, the resolutions within the decomposition are controlled by the selection of an appropriate, V_0, as the decomposition's starting point, since, for the case of discrete data, the level of decomposition is controlled by the length of the given data record. In particular, a discrete data record of length 2^N will be decomposed through multi-resolution analysis into the wavelet subspaces $\{W_0, \ldots W_N\}$, under the assumption that dyadic scaling and shifting, the meaning of which will be explained later in this section, are employed. The complete signal can be reconstructed by recombining the information contained within each of the

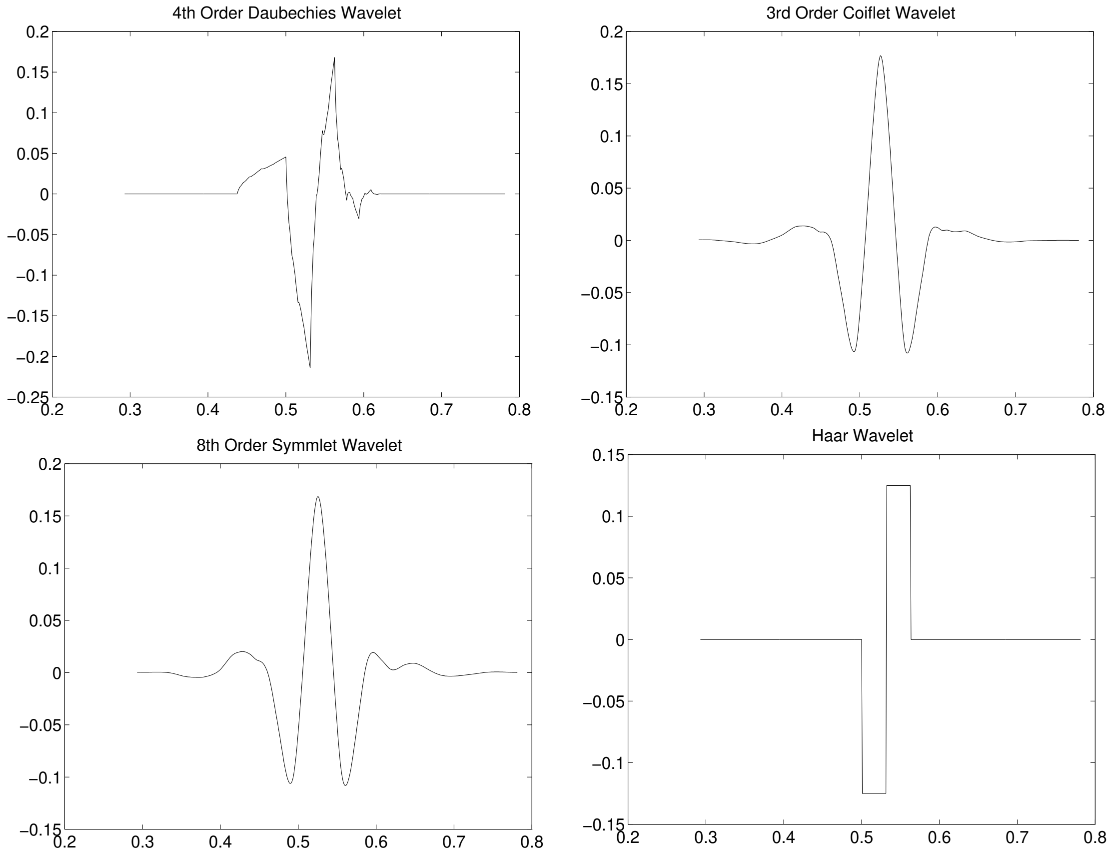


FIGURE 5-3.
Examples wavelet functions from the of the Daubechies, Coiflet, symmlet, and Haar wavelet families.

$$\bigcup_{j \in Z} V_j \quad \text{dense in } L^2(\Re) \tag{5.14}$$

And finally, that any function $g(t) \in L^2(\Re)$ can be approximated, as closely as is desired, by combining the various projections of function *g(t)* in the functional spaces, V_j, since the union of these functional spaces is dense in $L^2(\Re)$. Obviously, since $V_{j-1} \subset V_j$ approximating *g(t)* by its projections into the various V_j subspaces is not particularly useful since the subspaces overlap each other resulting in significant redundancy. What is required is a means of determining the portion of the function *g(t)* which is unique to the

made. At the low scales, the signal is viewed with a coarse time resolution and a high frequency resolution, while at the high scales the situation is reversed. This ability to look at a signal concurrently at different resolutions makes wavelets amenable to the analysis of signals with non-stationary and/or transient behaviours. At the heart of wavelet analysis is the basic or mother wavelet function, $\psi(t)$, and its associated scale function, $\phi(t)$. There are many different types or families of wavelets, depending on the particular application for which they were designed, but all possess the general properties of compact support[3] in both the time and the frequency domain, inclusion in $L^2(\Re)$ [4], and $\int_{-\infty}^{\infty} \psi(t) = 0$. Figure 5-3 shows some typical examples of commonly used wavelets.

Multi-resolution analysis[5] is the process by which the time-frequency space is partitioned. It, therefore, forms the basis for wavelet filtering. In multi-resolution analysis the functional space $L^2(\Re)$ is assumed to be approximated by a sequence of functional spaces $\{V_{j \in Z}\}$ which have the following properties [60][79][80]:

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \tag{5.12}$$

The functional space is partitioned into subspaces, with each subsequent subspace being a superset of the subspaces which preceded it.

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1} \quad , j \in Z \tag{5.13}$$

If a function $f(t)$ belongs to the functional space immediately preceding the given functional space then this implies that the dilated function $f(2t)$ belongs to the given subspace, and *vice versa*.

3. A function $f(x)$ which has compact support is exactly zero for all values of x outside of some finite interval $[a,b]$.

4. $L^P(\Re)$ consists of the set of functions $f(t)$ for which $\int_{-\infty}^{\infty} |f(t)|^P dt < \infty$.

5. The structure of this discussion of multi-resolution analysis as it applies to wavelet theory is based on the works presented in [60], [79] and [80].

If, though, a relatively smooth estimate of the underlying sensor signal were available, which was determined independently of the neural network modeling approach, then it could be used to produce a relatively smooth difference signal[2]. Hence, the fault detection threshold would not be limited by the coarseness of the quantization functions in use, but by the underlying noise effects present in the signal. The remainder of this chapter will look at how to produce such an estimate of the underlying sensor signal and how it can then be used to provide a statistical estimate of the signal's associated underlying noise process.

5.2 Wavelet De-noising

Obviously, a technique suited to the analysis of non-stationary signals is required to provide an estimate of the underlying sensor signal, particularly if the estimate is to be accurate within regions of behavioural change. Therefore, traditional frequency based filtering techniques are unsuitable. Wavelet theory and the associated field of wavelet de-noising, however, offer estimation techniques that are suitable to the analysis of non-stationary signals. The remainder of this section will therefore briefly overview wavelet theory and wavelet de-noising, and then discuss in detail how this technique can be utilized to provide a reasonable estimate of the underlying sensor signal for a given data sequence. Of particular interest, within this section, is how these tasks can be accomplished in such a manner as to adequately compensate for the coarse quantization effects present in the example plant.

5.2.1 Overview of Wavelet Theory

Wavelet theory provides a signal processing framework in which signals can be decomposed into a set of orthogonal time-frequency elements. The power of wavelets comes from the fact that, through multi-resolution analysis, this decomposition is not done using a single time-frequency resolution, as is the case in windowed Fourier transform based signal analysis techniques. Instead, a variety of resolutions or scales are used. At each of these scales, a different trade-off between the time and frequency resolution is

2. The term smooth is used in this context to indicate a signal which has been filtered such that the quantization and underlying noise effects are minimized. It is not used within this context to denote the notion of mathematical smoothness.

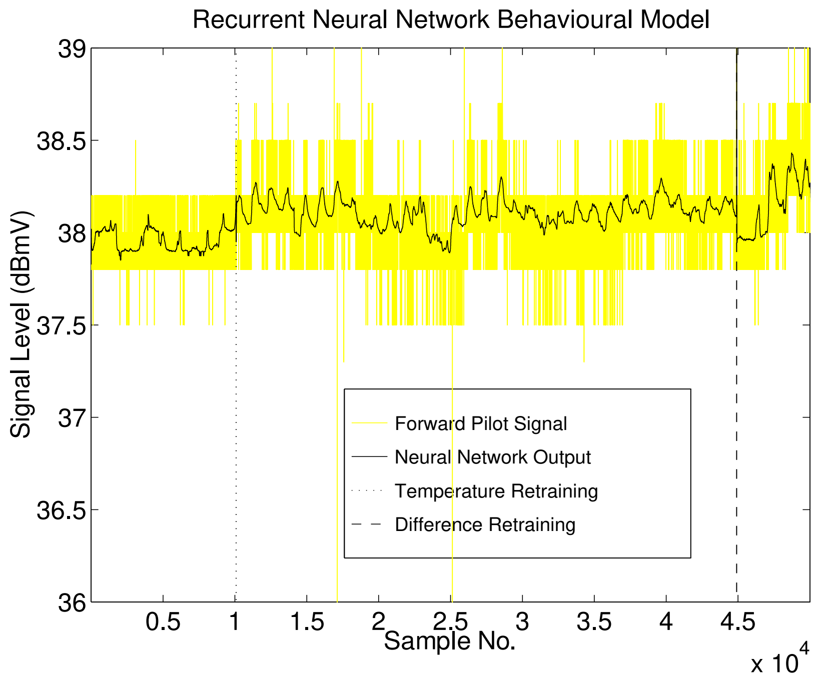


FIGURE 5-2.
Neural network behavioural model of the example forward pilot signal of Figure 3-1.

A second issue which is apparent from the plot of Figure 5-2, is that for the neural network fault detection methodology to be useful in a real system, the effects of the quantization and underlying noise processes must be taken into account. In particular, the preceding discussion, outlining how the neural network could be used as the basis for a fault detection system, implicitly assumed that the resulting difference signal would be relatively noise-free. Obviously, in a real system this assumption would not be valid, and in particular for the example system it is clearly not the case. Without having a relatively noise-free difference signal, it would be very difficult to set a fault detection threshold for the difference signal such there would be a low probability of false alarm, while maintaining a relatively high fault detection sensitivity. This requirement is particularly important if early fault detection tasks are to be performed. The fault detection system, as it was presented above, would have a fault detection threshold, and hence a fault detection sensitivity, which was limited by the coarse quantizations present in the output signal.

extrapolation tasks for data sequences outside of its "experience". It should be noted that this problem is distinct from the case where the model fails due to a behavioural change in the component; in the former case, the behaviour remains constant but the dynamic range over which the behaviour is experienced is expanded. A more complete discussion of solutions to these problems and the general training process which has been utilized to apply the neural network architecture of Section 1.4.2.2 to the modeling of the forward pilot-temperature dependency occurring in the example plant can be found in [35]. The specific training issues associated with utilizing the recurrent neural network behavioural modeling a basis for a fault detection system will be presented in Chapter 6.

Figure 5-2 shows the results of applying the neural network model, under the methodology presented in [35], to the example forward pilot and temperature signals of Figure 3-1. For this particular example, the neural network was initially trained over the first 5000 data samples, and re-training, due to the temperature signal exceeding the dynamic range present in the training set, was initiated at the location indicated by the dotted vertical line, in accordance with the methodology described in [35]. It is quite clear from this figure that the neural network does a fairly good job of estimating the forward pilot signal from the temperature signal. In particular, the estimation is quite accurate in areas where training occurred, and is slightly less accurate in areas where the neural network is free-running. This difference occurs since, the neural network is exposed to the complete behaviour of the system in the training regions; whereas in the free-running regions the neural network must interpolate the behaviour of the system based on the "knowledge" that it has gained during the training phase. The complete behaviour of the system may not be captured within the subset formed by the training regions. Hence, there may be slight variations in the free-running regions which the neural network is unable to track as accurately. One of the key problems to be solved in utilizing a neural network approach, as the basis of a fault detection system, is therefore how to differentiate between significant behavioural changes, occurring in the free-running regions, and the effects of a consistent behaviour which may not have been fully characterized by the training sets. The development of a solution to this problem will be discussed in Chapter 6.

$$\dot{\hat{P}}(kT) = f\left[\hat{P}((k-1)T), \ldots, \hat{P}((k-M)T), \tilde{T}_q(kT)\right] \tag{5.10}$$

such that the error given by

$$J = \frac{1}{2}\sum_{k=1}^{N} [\tilde{P}_q(kT) - \hat{P}(kT)]^2 \tag{5.11}$$

is minimized over the training sequence. It is important to note that the terms $\hat{P}((k-1)T), \ldots, \hat{P}((k-M)T)$ of Eq. 5.10 result from the recurrent nature of the neural network architecture and that the exact value of M is unknown and is not extractable, in general, from the trained network. In addition, the function f of Eq. 5.10 is determined through the training process and is contained within the structure and weights of the neural network and, due to the black-box nature of the neural network approach, it also is, in general, not extractable from the trained network. Once trained, the neural network can be utilized as a free-running, black-box model of the system, which implements the behavioural dynamics present during the training sequence. If, for a given component, there is a significant deviation[1] between the actual $\tilde{P}_q(kT)$ and the estimate $\hat{P}(kT)$, then the component can be flagged as having undergone a behavioural change.

Some care, obviously, must be taken during the training process to ensure that the neural network trains adequately over the training sequence. In particular, for this class of neural networks the training data must be normalized within the interval of *[0,1]* in a consistent manner, in the sense that the normalization function should be independent of the given data sequence. In this way, it is possible to compare the results obtained from the neural network models obtained over different time spans of data. Additionally, when the neural network is free-running there is the problem that it will not perform adequately for input data which was not “close” to the input data range which occurred within the training set. This problem results from the neural network having difficulties performing

---

1. The nature of what is meant by a significant deviation and the process by which it can be detected will be addressed in Chapter 6.

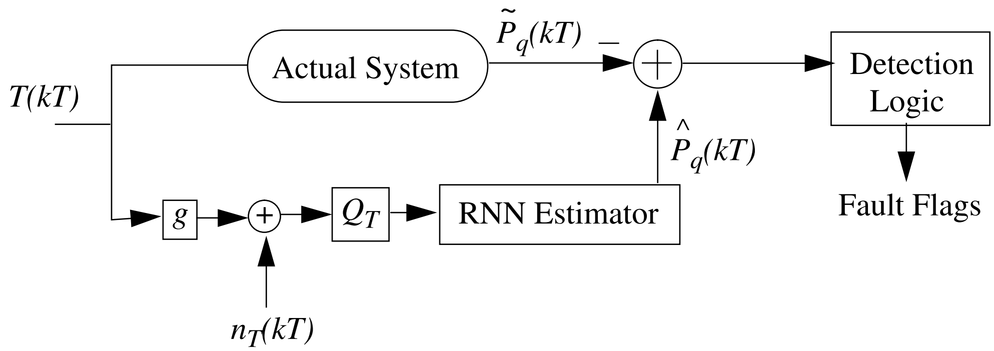


FIGURE 5-1.
Fault detection through the use of a black-box, recurrent neural network estimator.

cult to model a system which exhibits arbitrary non-linear, dynamic dependencies using classical approaches. Fortunately, the neural network presented in Section 1.4.2.2 and described by the differential equation

$$\dot{\boldsymbol{O}} = -\boldsymbol{T}\boldsymbol{O} + \boldsymbol{W}f(\boldsymbol{O}) + \boldsymbol{b}(t) \tag{5.8}$$

can perform this estimation task provided that it is adequately trained. In particular, the neural network can be trained to model a given behavioural period, through a modified back propagation algorithm, by presenting the network with a sequence of input/output pairs which describe the amplifier's behaviour during the given period.

$$\{(\tilde{T}_q(kT), \tilde{P}_q(kT)) \mid k = 1, \ldots, N\} \tag{5.9}$$

During the training phase, the training algorithm adjusts the neural network's weights such that the neural network "learns" the non-linear dynamics of the system, in the sense that the neural network "learns" to perform the appropriate mapping task to generate the output data sequence, $\{\tilde{P}_q(kT) \mid k = 1, \ldots, N\}$, from the input data sequence, $\{\tilde{T}_q(kT) \mid k = 1, \ldots, N\}$. In particular, the neural network becomes a black-box implementation of the estimator given by

where

$$\tilde{P}_q(kT) = \tilde{Q}_P\left[\left(\tilde{P}(t) + \tilde{n}_p(t)\right)\Big|_{t = kT}\right] \tag{5.4}$$

$$\tilde{T}_q(kT) = \tilde{Q}_T\left[\left(\tilde{T}(t) + \tilde{n}_T(t)\right)\Big|_{t = kT}\right] \tag{5.5}$$

and the functions $\tilde{Q}_P(.)$ and $\tilde{Q}_T(.)$ represent the non-linear quantization functions employed by the status monitoring system.

It is important to note that the value $\tilde{T}_q(kT)$ shown above may not directly relate to the sampled enclosure temperature $T(kT)$. The value measured by the temperature sensor, and reported as the enclosure temperature by the monitoring system, may itself have some non-linear, dynamic relation with the "true" enclosure temperature, such that

$$\dot{T}_{measured}(t) = g(T(t)) \tag{5.6}$$

and

$$\tilde{T}_q(kT) = Q_T\left[\left(\tilde{T}_{measured}(kT) + \tilde{n}_T(kT)\right)\right] \tag{5.7}$$

For the purposes of neural network system modeling, this actuality does not pose a significant problem since the dynamics present in the measurement of the enclosure temperature will be captured as part of the forward pilot-temperature modeling process.

Fault detection for the system of Eq. 5.3 can be accomplished by providing an estimate of $\tilde{P}_q(kT)$ such that the difference between the estimated $\tilde{P}_q(kT)$ and the actual $\tilde{P}_q(kT)$ can be used to provide an indication of the occurrence of behavioural changes in the system, as shown in Figure 5-1. In general, as discussed in Chapter 1, it is quite diffi-

signal's underlying noise process. The underlying sensor signal and underlying noise estimations will then be used in Chapter 6 to develop a methodology for setting the neural network detection system's fault detection threshold such that the probability of detecting statistically "real" fault events is relatively high, while the probability of generating false alarms due to noise effects is kept relatively low. Hence, within Chapter 6 some issues related to the training of the neural network models will be discussed.

Throughout Chapter 5 and Chapter 6 it will be assumed that the system's output signal is non-linearly, dynamically dependent on only one input signal. The more general case where the output signal is non-linearly, dynamically dependent on a set of input variables is not considered within this work and has been left as an area of future work. Hence, as in Chapter 4, the goal is to develop a methodology suitable for modeling the primary dependency present in the system.

5.1 Recurrent Neural Network Modeling

It can be recalled from Chapter 3, that a system exhibiting unknown non-linear, dynamic dependencies, conditioned on a single input, can be expressed mathematically by

$$\dot{x}_o(t) = f(x_i(t)) \tag{5.1}$$

where $x_o(t)$ is the system output signal, $x_i(t)$ is the system input signal and $f(.)$ is an unknown, non-linear function.

In particular, for a given amplifier in the example plant, the primary forward pilot-temperature dependency is assumed to be given by

$$\dot{P}(t) = f(T(t)) \tag{5.2}$$

where $P(t)$ is the continuous time forward pilot signal, and $T(t)$ is the associated continuous enclosure temperature signal. In terms of the status data, Eq. 5.2 can be re-written, as per the notation of Section 2.3, as

$$\Delta\tilde{P}_q(kT) = f(\tilde{T}_q(kT)) \tag{5.3}$$

Chapter 5:

Recurrent Neural Network Based Fault Detection: Wavelet De-Noising and Underlying Noise Estimations

5.0 Introduction

In this chapter, a basic fault detection methodology for systems exhibiting unknown non-linear, dynamic dependencies is introduced. At the core of this methodology, is the use of an asymptotically stable recurrent neural network to model the system's unknown non-linear, dynamic behaviour. Surrounding this core, are the techniques which will be developed, within this chapter and Chapter 6, which allow the fault detection thresholds for the recurrent neural network modeling technique to be set in a prescribed manner. The chapter will begin by reviewing the recurrent neural network modeling technique, which was presented in Chapter 1, with particular emphasis being placed on how this network can be utilized as the basis for a fault detection system. This review will take the form of overviewing how the neural network can be used to model the forward pilot versus temperature behaviour exhibited within the example plant. As in Chapter 4, the approach of presenting the chapter's concepts through the use of a particular example signal has been done for the sake of clarity. The modeling approach to be presented, though, is not restricted to signals of only this one class; it is a general approach applicable to signals exhibiting unknown, non-linear, dynamic dependencies.

The issues involved in the details of the training of the neural networks will not be discussed directly within this chapter as they have been presented in [35]. Instead, the focus of this chapter and Chapter 6 will be on the development of an appropriate framework for setting the fault detection thresholds, such that the neural network system model can be utilized as the basis for an early fault detection system. To this end, the majority of the chapter will focus on the development of a wavelet de-noising technique for estimating the underlying sensor signals solely based on the information contained within the status data signals. Once the estimate of the underlying sensor signal has been determined, this estimate will then be utilized in the generation of a Gaussian noise approximation of the

events will produced fault flags. If it turns out that these transient events are not significant in a fault detection sense then appropriate pre-processing filters can be implemented such that the temperature signals utilized by the fuzzy membership function base fault detection system are transient free.

It should be note that the detection method which was presented is not restricted to solely using either the pseudo-Gaussian or sigmoidal membership functions. In addition, the method is not limited to modeling just the current-temperature behaviour of the example plant. The method is generally applicable to system's exhibiting linear dependencies which are obscured by coarse quantization processes and underlying noise effects. In the next two chapters, the problem of modeling non-linear, dynamic signal dependencies under the same constraints of coarse quantization processes and underlying noise effects will be addressed.

can be adjusted to more accurately reflect their uniqueness. In particular, the test for appropriateness described in Section 4.2.5 can be used to determine when the membership functions, and the associated upper and lower thresholding functions, should be re-generated.

Obviously, if a more complex relation between the membership functions existed then a more elaborate approach than direct copying would have to be employed. The key idea though is that for incremental model development to be possible *a priori* information about the intra-relationship between the membership functions must be available.

4.6 Conclusions

In this chapter, a technique based on fuzzy membership functions was presented which was capable of fault detection for systems exhibiting linear dependencies. In addition, the formulation of the technique permitted both the generation of optimal upper and lower linear thresholding functions, with respect to a probability of false alarm, and the direct testing of the method appropriateness. Two specific analytical membership functions were then introduced and compared in terms of their ability to model the current versus temperature dependency present in the example plant's status data signals. Due to the pseudo-Gaussian membership function approaches better modeling accuracy and better correspondence between the theoretical and experimental results, it was chosen as the approach to use in the more complete fault detection system comparison to be carried out in Chapter 7. The effects of the inherent uniform temperature distribution assumption were then discussed and a possible resolution was proposed. The chapter concluded by discussing the conditions and approaches that would allow for incremental model development.

The one facet that this chapter did not address was the issue of the reliance of the technique on the raw independent signal. In particular, transient events on this signal will cause alarm flags to be generated by the fuzzy membership functions based modeling technique. This issue was not addressed at within this Chapter since it is not known at this time whether or not the transient events which exist on the example plant's temperature status data signals are significant in a a fault detection sense. Hence, the current detection system, as described in this Chapter, errors on the side of caution in that these transient

Because the experimental and theoretical results obtained for the example plant's linear dependent signal class were in close agreement, this additional correction was not implemented within this work, and the actual formation of the correction has been left as an area of future work. It is important to note, though, that this implicit assumption does exist within the false alarm probability formulations which have been given in this chapter.

4.5 Incremental Model Development

All of the preceding discussions have assumed that the complete behavioural map describing the current-temperature relationship is available for analysis *a priori*. Obviously, in a real-world implementations it would be desirable to be able to apply this detection technique without necessarily having to wait for enough data to be collected for the complete map to be described. Ideally, one would like to require as little information as possible about the behavioural map during the initial model development and incrementally improve the model as more information, related to the same behavioural state, becomes available. In the case of the fuzzy base membership modeling technique, the ability to proceed in this manner hinges on there existing some relationship between the k membership functions. In the simplest case, each of the membership functions would be nearly identical to each other. Hence, the complete behavioural model could be constructed utilizing only the information required to describe one of the membership functions.

For example, it has been shown in [66] that, the step widths for the current-temperature staircase function, generated for a given amplifier in the example plant, tend to be similar ($< 20\%$ variation). It was also shown that this result was true across the majority of amplifiers within the plant. Hence a reasonable approach to develop incremental models in this case is to collect data until one complete step on the staircase function has been observed. From this step the associated membership function can be generated, and since the steps are known to be similar, the behavioural model can be generated by shifting an scaling copies of this one membership function. Hence, the initial estimates of the upper and lower thresholding functions can also be generated. As more information about the current-temperature behaviour becomes available, the individual membership functions

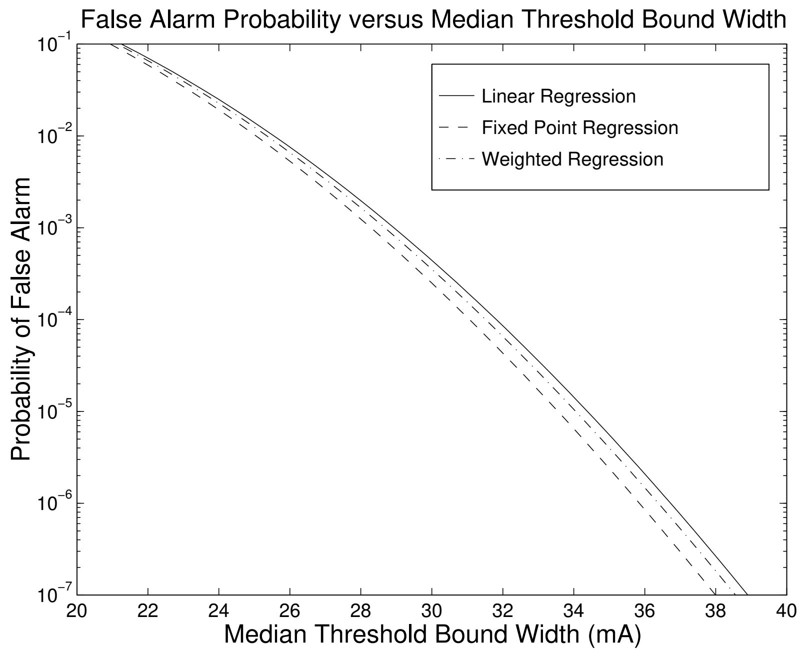


FIGURE 4-24.
Relationship between the probability of false alarm and resulting median threshold bound width.

against a variety of false alarm probability settings. The results of such a plotting, for the example current signal of Figure 4-23, are shown in Figure 4-24 for each of the three threshold generation methodologies which have been discussed.

4.4 Temperature Distribution Correction

Although the theoretical and experimental results for both the sigmoidal and pseudo-Gaussian membership functions modeling approaches are quite close, they could be improved by performing an additional correction step. In particular, throughout this chapter it has been implicitly assumed, within the probability of false alarm formulations, that the temperature distribution is uniform. Obviously, in practice this is not the case. Hence, some error will be introduced by this assumption. This error can be reduced by weighting the membership functions within the false alarm probability formulations by their probability of occurrence. Each of the membership functions essentially describe a particular regions of the temperature space; hence, the temperatures probability distribution can be utilized to estimate the relative contribution of the various membership functions.

signal has on its associated temperature signals. In Chapter 7, it will be seen that this result holds true in general for the fuzzy membership function based fault detection technique for the example plant's linearly dependent signal class.

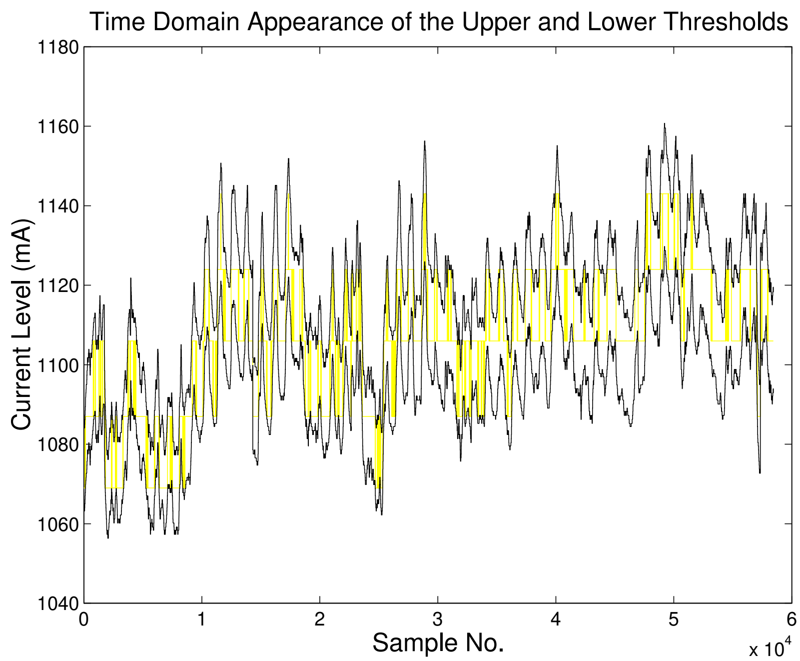


FIGURE 4-23.
Time domain appearance of the upper and lower thresholding functions (based on the pseudo-Gaussian modeling approach) for the raw behavioural map given in Figure 4-1 and under a false alarm probability of 10^{-5}.

From a practical point of view, it is also desirable to have some feeling for how the setting of the false alarm rate effects the resulting median threshold bound width. In particular, traditional limit checking detection thresholds are typically specified in terms of the bounds widths, hence, from the point of view of the technical personnel who set up and maintain a given plant's fault detection system it would be nice, in practice, to have a methodology of relating the probability of false alarm to the resulting threshold bound width. This can be done in a straight forward manner within the fuzzy membership function based detection methodology by plotting the achieved median threshold bound width

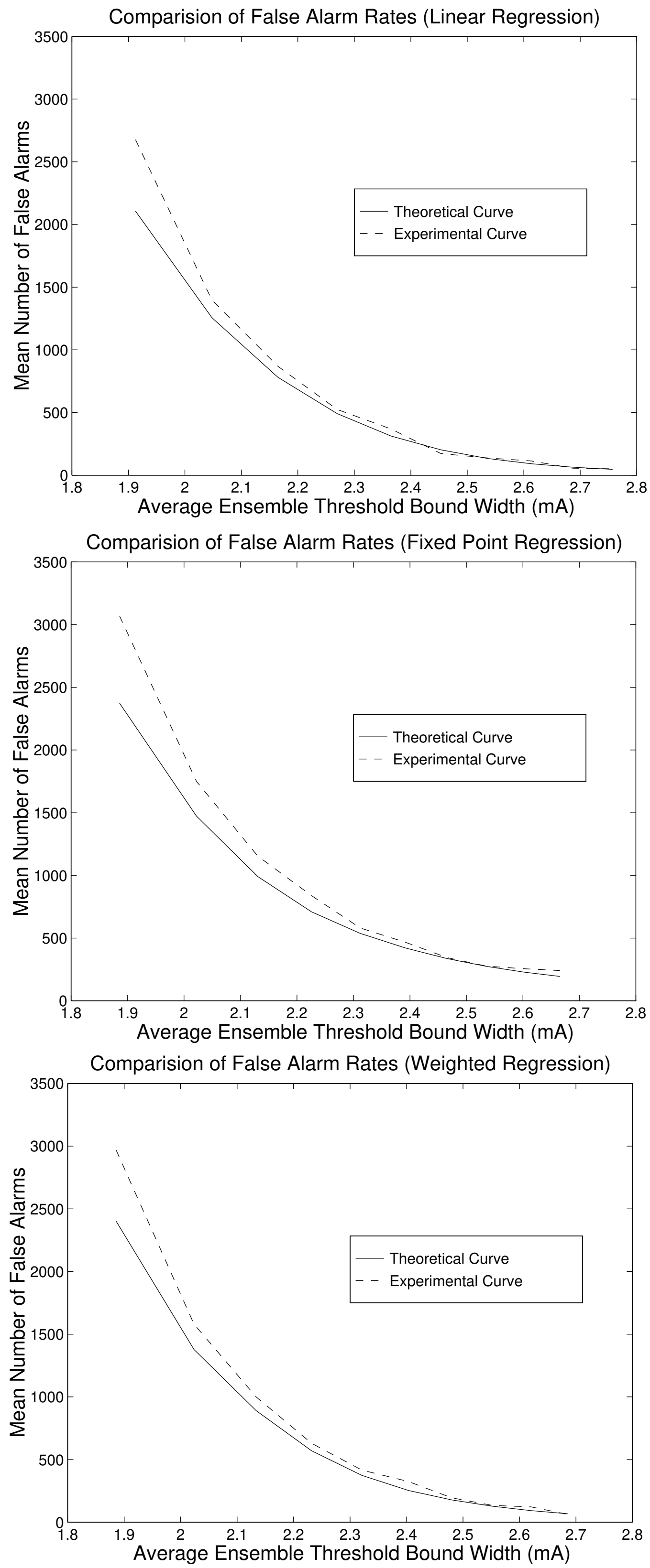


FIGURE 4-22.
Comparison of the experimental and theoretical mean number of false alarms for the three threshold generation techniques across the set of 22 example data records (pseudo-Gaussian membership function case).

dence, across all three techniques, between the experimental and theoretical results. In particular the correspondence between the theoretical and experimental results are slightly better for the pseudo-Gaussian modeling approach than they were for the sigmoidal modeling approach. As in the sigmoidal membership function case, the pseudo-Gaussian membership functions can also be utilized as the basis for a consistent fault detection system for the example plant's current versus temperature behaviour. What is noticeable from the plots of Figure 4-19 and Figure 4-22 is that the pseudo-Gaussian membership functions provide a closer correspondence between the theoretical and experimental results when low probabilities of false alarm are specified (which corresponds to wider bound widths). Typically, within an operational plant it would be these low probabilities of false alarm which would be employed. Hence, since the pseudo-Gaussian analytical membership function provides a closer correspondence between the experimental and theoretical thresholding in this region, it will be the analytical membership function utilized throughout the remainder of this work to model the example plant's linearly dependent signal class.

4.3 Time Domain Appearance

Up to this point the discussion has focused on the appearance and location of the linear thresholding functions within the domain of the raw behavioural maps and their associated behavioural models. In practice, though, within an operational fault detections system, the bounds are employed directly on time domain signals. Hence, it is important to have a feeling for how the bounds appear in the time domain and understanding of how the probability of false alarms affects the resulting threshold bound width. To this end, Figure 4-23 shows the time domains threshold bounds produced by utilizing the pseudo-Gaussian membership functions as the basis for the linear threshold function generation, under the linear regression approach, for the example current signal of Figure 4-1 with the false alarm probability set to 10^{-5}. It can be seen quite clearly from this figure that the thresholding technique results in very tight bounds being placed on the current signal. Despite the tightness of these bounds, no alarm flags are produced in this case; the fuzzy membership based modeling approach accurately models the linear dependency that the current

As in the sigmoidal membership function case, the consistency of the pseudo--Gaussian membership function modeling approach was evaluated on the set of 22 data records in the same manner as described in Section 4.2.8.1. Figure 4-21 shows the same example plot of one of the 22 experiments which was shown in Figure 4-18, with the exception that pseudo-Gaussian, instead of sigmoidal, membership functions have been employed to construct the behavioural model. It can be seen that both Figure 4-18 and Figure 4-21 are very similar, indicating that the consistency of the fault detection approach is not heavily reliant of the actual nature of the membership functions employed, providing of course that the model the raw membership functions reasonably accurately.

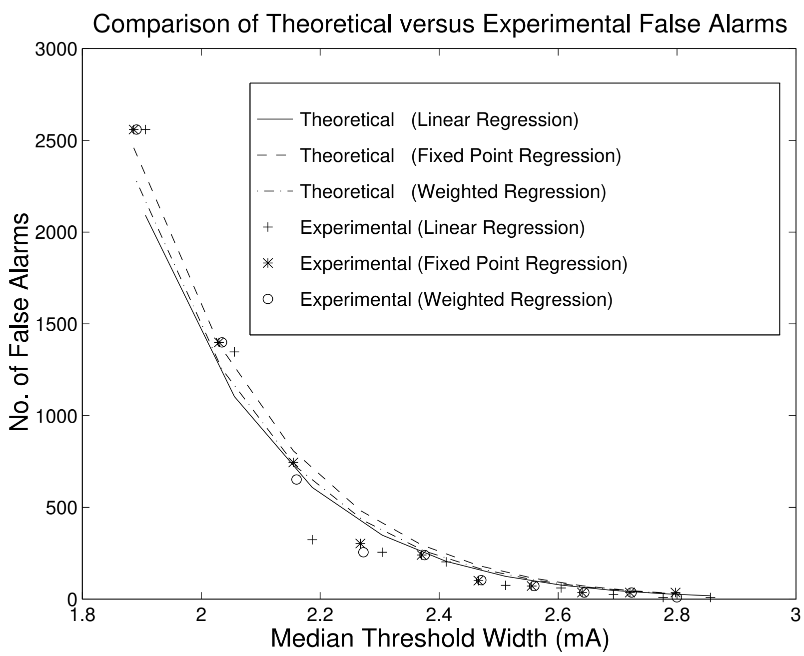


FIGURE 4-21.
Comparison of the theoretical and experimental false alarms for one of the 22 example data records whose behavioural map was modeled with the sigmoidal membership function (pseudo-Gaussian membership function case).

This conclusion is re-enforce by Figure 4-22 which shows the plots comparing the average number of false alarms generated in both the experimental and theoretical cases for the three threshold generation methodologies which were used. As in the sigmoidal membership function case, it is quite clear from these plots that there is a good correspon-

again, no such theoretical framework is available within the literature for the linearly dependent signal class of interest within the example plant and the development of such a framework has been left as an area of future work.

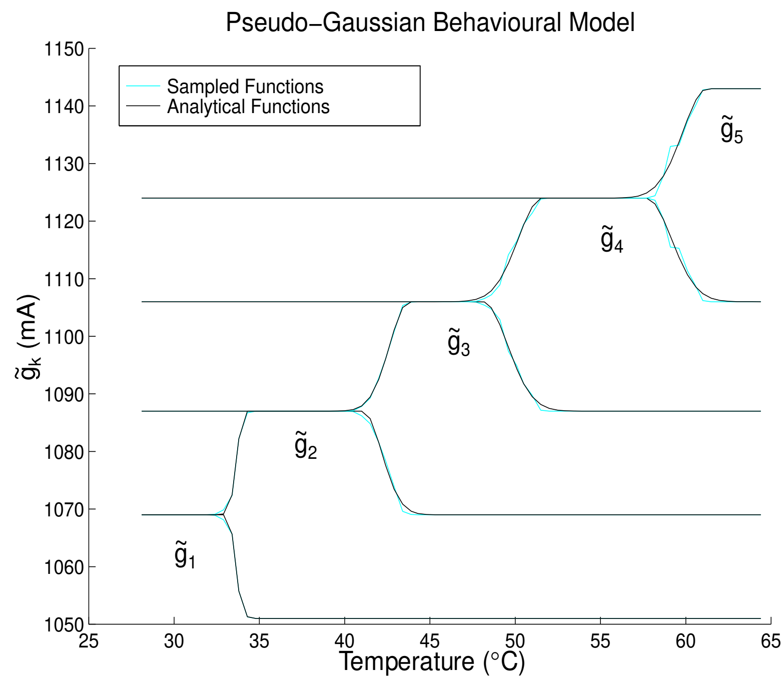


FIGURE 4-20.
Current versus temperature behavioural model generated from the optimized pseudo-Gaussian analytical membership functions.

Table 4.3: Pseudo-Gaussian membership function parameters obtained through gradient descent optimizations for the example behavioural model of Figure 4-4.

Membership Function	μ_k^u	μ_k^l	σ_k^u	σ_k^l	**% error**	**No. Iterations**	**Flops**
I_1	27.33	33.14	0.404	0.404	0.453	144	447,381
I_2	34.11	41.08	0.389	1.091	1.927	429	1,260,056
I_3	43.77	48.12	1.122	1.431	2.175	95	297,933
I_4	51.60	57.70	1.414	1.461	3.510	238	711,116
I_5	61.26	67.07	1.435	1.435	3.761	31	98,498

results agree quite closely, over all three threshold function generation methods. This provides a strong indication that the utilization of sigmoidal membership functions in the modeling process results in a consistent fault detection approach.

4.2.8.2 Pseudo-Gaussian Membership Function

Figure 4-20 and Table 4.3 show the results of applying the pseudo-Gaussian membership function in the generation in of the behavioural model which was given in Figure 4-4. It is quite clear from both Figure 4-20 and the table that the pseudo-Gaussian membership function also performs quite well in modeling the raw behavioural model. In fact, the pseudo-Gaussian membership function tend to model the sampled membership functions about 1% better than what was achieved with the sigmoidal membership functions. Since the optimization parameters were set identically for the both the sigmoidal and pseudo-Gaussian approaches, there are two likely factors which account for this improved modeling ability. The first factor involves that fact that for each of the pseudo-Gaussian membership functions involved, both the right and left boundaries are optimized independently. Unlike in the sigmoidal membership function case, neither boundary is dependent upon its neighbouring membership function. All boundaries are optimized independently. This should tend to improve the modeling accuracy of the approach, though at the cost of significantly increasing the computational cost of determining the modeling parameters. It should be recalled that the independency of the pseudo-Gaussian boundaries comes about due to the desire to utilize symmetric membership functions, in the sense that both the right and left boundaries have the same functional form. Since it is not possible, when utilizing Gaussian boundaries, to meet this condition and the point-by-point summation condition simultaneously, each of the boundaries must be optimized independently. The second factor which may account for the improved modeling is that the pseudo-Gaussian membership function may innately model the transition behaviour present in the example plant's current versus temperature raw behavioural maps more accurately than the sigmoidal membership functions. Obviously, this claim cannot be verified without a having a complete understanding of the theoretical nature of the transition areas statistics. Once

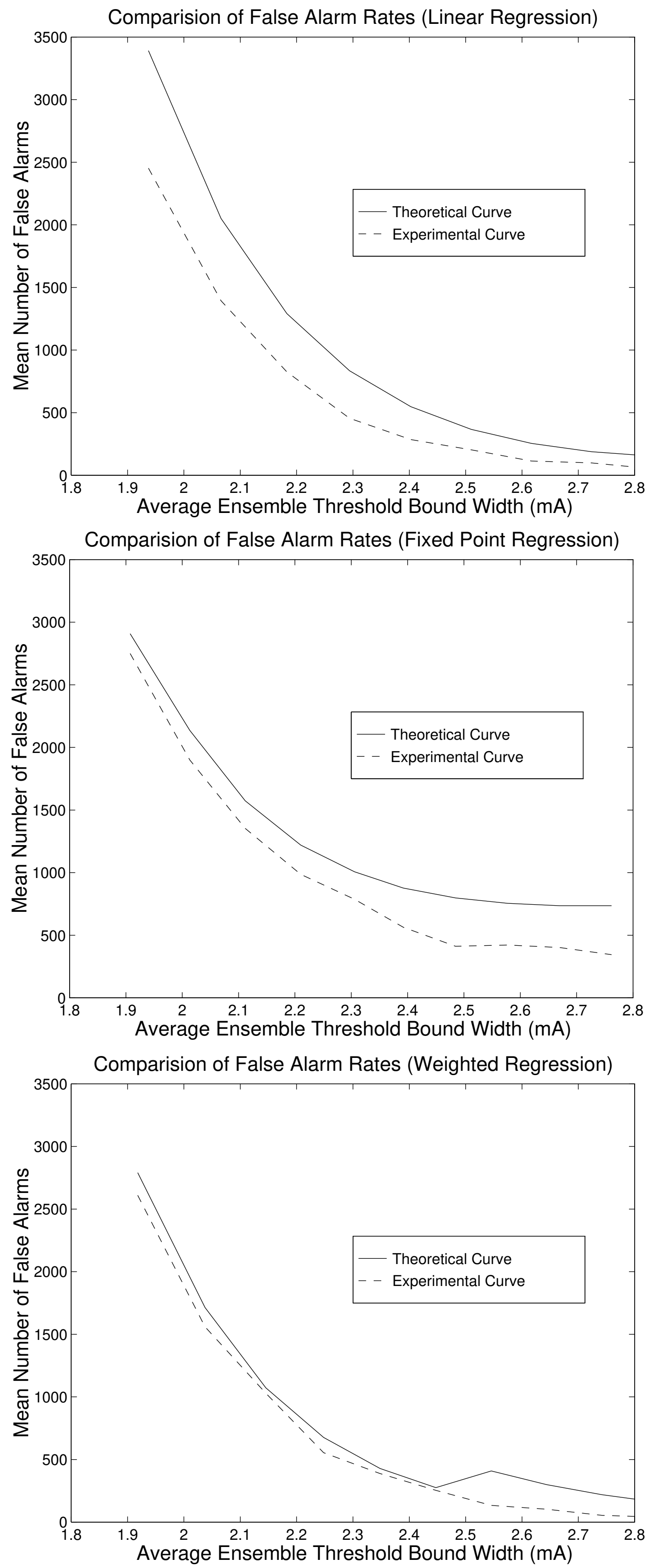


FIGURE 4-19.
Comparison of the experimental and theoretical mean number of false alarms for the three threshold generation techniques across the set of 22 example data records (sigmoidal membership function case).

and experimental false alarms produced by each of the threshold generation approaches plotted relative to the resulting median threshold widths (measured as the median distance between the given upper and lower thresholding functions).

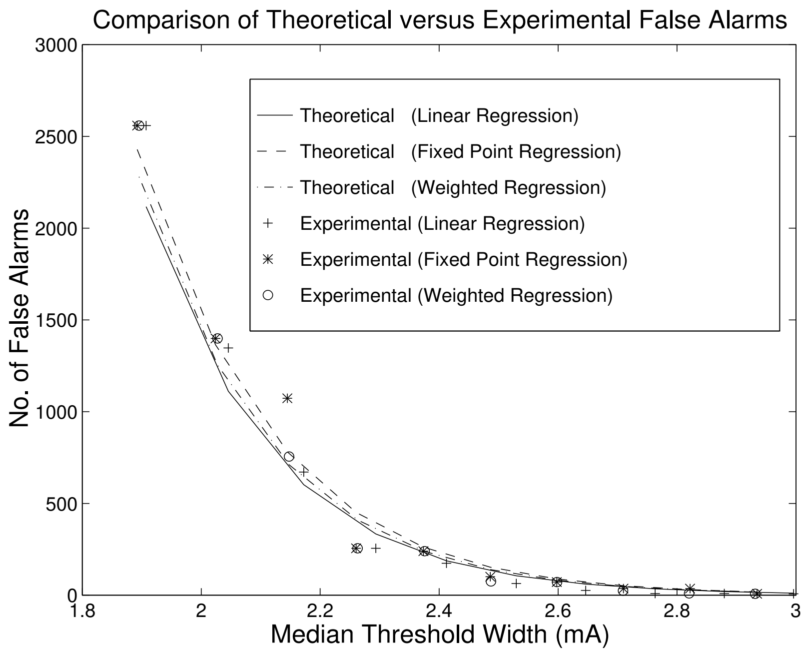


FIGURE 4-18.
Comparison of the theoretical and experimental false alarms for one of the 22 example data records whose behavioural map was modeled with the sigmoidal membership function (sigmoidal membership function case).

This figure shows that the experimental and theoretical results are in close agreement. Additionally, it demonstrates that the utilization of the various threshold generation techniques produces little variation in the end results. Each of the generation methods produces essentially the same, or very similar, bounding functions. These results are consistent with those obtained across the full 22 record data set. In particular, Figure 4-19 shows the comparison of the theoretical and experimental mean number of false alarms, obtained across the full set of 22 data records, for each of the prescribed false alarm probabilities. In essence the plots in this figure show on average how many false alarms are expected in both the theoretical and experimental case for each of the probability of false alarm rates. Hence, these plots show that on average the theoretical and experimental

The other criterion, which must be met by any analytical membership function which is used to model the dependency map, is that the resulting fault detection process, which is based on the given analytical function, is consistent. In particular, for given upper and lower thresholding functions, the theoretical probability of false alarm should concur with the experimental false alarm rate obtained utilizing the same thresholding functions. To test this hypothesis behavioural models were obtained for 22 amplifiers utilizing data from the November 1995 to December 1995 period. These 22 amplifiers were hand picked to have "well" behaved current versus temperature behaviours. Obviously, if the consistency of the methodology is to be determined then relatively fault free data is required. As will be explained more thoroughly in Chapter 7, the data set available for the evaluation of the detection procedures developed in this work is not a "ground truthed" data set, in that the spatial and temporal locations of fault events within the data set are unknown. Hence, in order to test the above procedure it was necessary to hand search the data set to obtain example signal records which closely followed the presumed "staircase" functionality. It is important to recall at this point, that by definition the fuzzy membership function based detection procedure is limited to those data records which follow or closely follow a "staircase" shaped dependency map. Hence, selecting these type of data records to determine the consistency of the system does not weaken the resulting claims. The detection system is designed to model "staircase" dependencies and hence the consistency of the system need only be determined for records possessing a "staircase" dependency.

Once the 22 data records were selected and the behavioural models produced, a series of tests were then undertaken in which the probability of false alarm was varied along a logarithmic scale from 20% (1 in 5) to 0.000391% (1 in 2650). For each of these false alarm probabilities, the three proposed linear thresholding function generation approaches were utilized to generated both the upper and lower thresholding functions. These thresholding functions where then employed to bound the raw data signal and the experimental false alarm rate associated with the given threshold was determined, under the assumption that each of the 22 data records were in fact fault-free. In Figure 4-18 an example plot of one of these experiments is shown. In particular, this plot shows the number of theoretical

boundaries. The resulting computational savings are only possible due to the fact that the sigmoidal membership function meets the point-by-point summation condition given in Section 4.2.4.

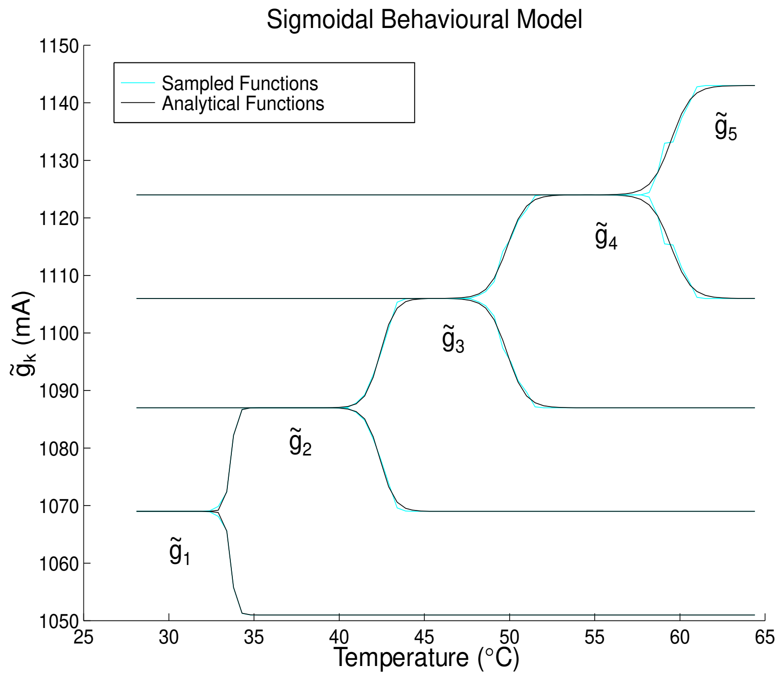


FIGURE 4-17.
Current versus temperature behavioural model generated from the optimized sigmoidal analytical membership functions.

Table 4.2: Sigmoidal membership function parameters obtained through gradient descent optimization for the example behavioural model of Figure 4-4.

Membership Function	α_k	μ_k	β_k	% error	No. Iterations	Flops
I_1	6.12	33.63	28.10	0.434	69	110,782
I_2	2.33	42.40	36.05	1.413	61	119,031
I_3	1.85	49.86	45.70	3.260	59	128,097
I_4	1.73	59.49	54.52	4.106	57	109,073
I_5	-	-	64.40	4.858	-	-

the sense of the two criteria outlined above. The development of the "optimal" analytical membership function for a given transition behaviour would require that a formal mathematical description of the nature of the transition statistics be available. Currently, no such description is available within the literature for the general problem of noisy, linear dependent signals which are affected by lower order quantization processes. Hence, the approach which as been taken within this work is to present two possible analytical membership functions and then to demonstrate that they provide a reasonable approximation to the transition behaviour which is exhibited in the example plant's current versus temperature maps.

4.2.8.1 Sigmoidal Membership Function

Figure 4-17 and Table 4.2[6] show the results of utilizing the sigmoidal analytical membership function in the generation of the behavioural model which was given in Figure 4-4. It is quite clear from both Figure 4-17 and the table that the sigmoidal membership function does a reasonably good job of approximating the statistical behaviour of the transition areas which are present in the raw behavioural map. In particular, the worst-case percentage error for the five modeled membership function is only in the range of approximately 5%. This is a relatively small error in the statistical modeling of the raw membership functions given that a number of criteria, such as the linearity of the dependency and underlying noise statistics, affect the nature of the transition statistics, and no overt assumptions have been made about these criteria within the statistical modeling process itself. What is also evident from the data presented in the table is that the sigmoidal modeling approach is computationally inexpensive. Very few optimization steps were required in order to determine the required analytical function parameters. In part, the low computational cost of this technique is due to the ability of the technique to determine the complete behavioural model by only performing optimizations on the four right-hand side

6. Due to how the sigmoidal membership function was formulated in Section 4.2.6.1, only 4 membership functions' parameters need to be determined for the behavioural model shown in Figure 4-17; hence, the parameters for current level I_5 are not shown in the table since they are derived directly from those of current level I_4.

It should be noted that in Figure 4-16 the underlying behavioural model was constructed utilizing the pseudo-Gaussian membership functions. A similar comparison utilizing the sigmoidal membership functions as the basis for the behavioural model was not undertaken. The fact that only one of the proposed analytical membership functions was used in the comparison does not weaken the conclusion that the heuristic methodologies provide near optimal thresholding functions. Both the brute force search strategy and the heuristic methodologies have at their core the same analytical membership function. Changes to the analytical membership function, therefore, affect the absolute value of the probabilities of false alarm which are computed for a given threshold function, but the changes will not affect the comparison between the near optimal solution found by the brute force search and the solutions found through the heuristic methodologies.

Within the next section the two proposed analytical membership functions will be compared to determine their relative merits.

4.2.8 Membership Function Comparison

The analytical membership functions described above represent two possible membership functions which can be utilized to model the raw membership functions obtained from the raw current versus temperature behavioural map. In general, the applicability of a given analytical membership function to adequately model the transition effects depends on two criteria. First, the analytical function should provide a reasonably accurate estimate of the transition statistics in the sense that the percentage error between the raw membership functions and their analytical counterparts should be suitably small. Secondly, it is important to show that the complete modeling technique which results is sound, in the sense that the theoretical false alarm probabilities predicted by the technique should agree closely with the false alarms rates which are actually achieved. The relative ability of the two proposed analytical membership functions to satisfy these criteria will be discussed in the following two sections.

It should be noted, that there will be no attempt to identify the “best” analytical membership function for the current versus temperature behaviour within this work. Instead the approach will be to validate that the two proposed membership functions are reasonable in

in all four cases the generated thresholding functions are quite similar. Hence, Figure 4-16 provides validation that the heuristic methodologies can in fact be used to provide near optimal thresholding functions at a fraction of the computational cost that would be involved in performing the full constrained optimization. The comparison of the resulting bounded area of the four sets of thresholds are shown in Table 4.1.

Table 4.1: Comparison of the enclosed areas obtained by the three heuristic threshold bound generation methodologies and the near-optimal area obtained by the brute force search strategy.

Threshold Function Determination Methodology	Enclosed Area (given by Eq. 4.38)
Brute Force Search	98.141 mA˚C
Linear Regression	104. 338 mA˚C
Fixed Point Linear Regression	101.732 mA˚C
Weighted Linear Regression	106.316 mA˚C

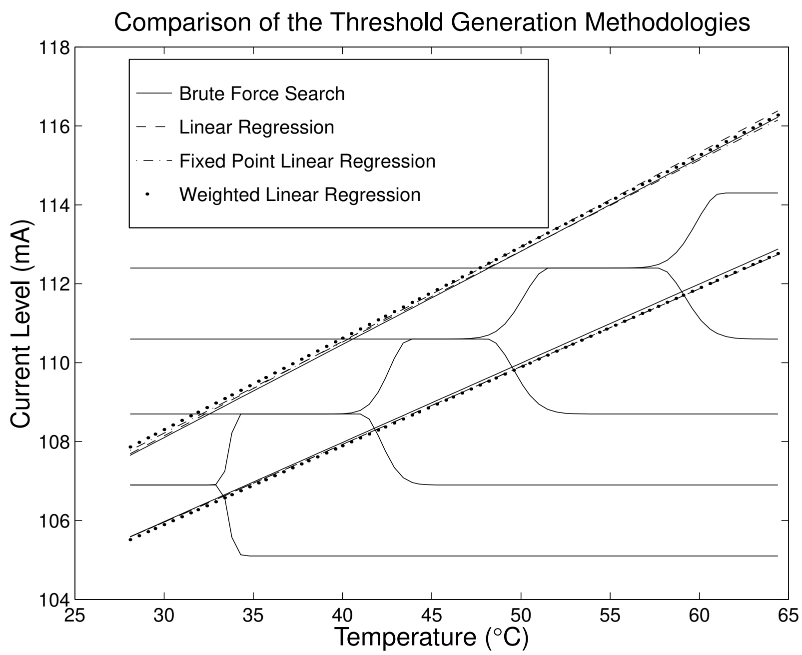


FIGURE 4-16.
Comparison between the thresholding functions produced by the three heuristic threshold generation methodologies and those produced by the brute force search strategy.

Other measures of the sensitivity of the membership functions to changes in their intersection point can be found, but the nature of these measures and a comparison of the respective measures has been left as an area of future work. It should be noted that the sensitivity measure as given above is, by definition, bounded between 0 and 1.

4.2.7.4 Weighted Linear Regression

Since sensitivity measures must be obtained for all the membership functions in order to determine which function is the most sensitive, it would be reasonable to weight each of the intersection points contribution to the linear regression by its sensitivity instead of just fixing one of the points. If this is done then a weighted linear regression results and the upper threshold linear regression formulae become

$$\alpha^u = \frac{\left(\sum_{k=1}^{N} S_k^u\right)\left(\sum_{k=1}^{N} S_k^u I_k T_k^u\right) - \left(\sum_{k=1}^{N} S_k^u I_k\right)\left(\sum_{k=1}^{N} S_k^u T_k^u\right)}{\left(\sum_{k=1}^{N} S_k^u\right)\left(\sum_{k=1}^{N} S_k^u \left(T_k^u\right)^2\right) - \left(\sum_{k=1}^{N} S_k^u T_k^u\right)^2} \tag{4.45}$$

and

$$\beta^u = \frac{\sum_{k=1}^{N} S_k^u I_k - \alpha^u \sum_{k=1}^{N} S_k^u T_k^u}{\sum_{k=1}^{N} S_k^u} \tag{4.46}$$

4.2.7.5 Comparison to Full Optimization

Figure 4-16 shows the comparison between the linear thresholding functions generated with the above heuristic methodologies and the near optimal linear thresholding functions determined through a brute force search procedure[5]. It can be seen quite clearly, that

---

[5] In this procedure 100,000 upper and lower thresholding functions were independently produced which met the required probability of false alarm criterion. All combinations of upper and lower thresholding functions were then analyzed to determine the pair which produced the least enclosed area in accordance with Eq. 4.38. A more complete description of this brute force search strategy can be found in Appendix D.

point. Under this approach, if the j^{th} membership function's upper threshold intersection point is the most sensitive to location shifts and it is therefore fixed, then the linear regression formulae for the upper thresholding function are given by

$$\alpha^u = \frac{\sum_{k=1,\,k\neq j}^{N} (I_j - I_k)\left(T_j^u - T_k^u\right)}{\sum_{k=1,\,k\neq j}^{N} \left(T_j^u - T_k^u\right)^2} \tag{4.42}$$

and

$$\beta^u = I_j - \alpha^u T_j^u \tag{4.43}$$

One approach to measuring the relative sensitivity of the intersection point to location shifts is to determine the relative area under the membership functions contained within a prescribed region of the intersection point. The membership function with the largest area will be the function whose individual probability of false alarm will be most sensitive to small shifts in the location of its intersection point, in the sense that its individual false alarm probability will vary the most. Specifically, if the area of interest is chosen to be $\pm\delta$ around each of the ideal intersection points, then for the upper bounding function the relative sensitivities, S_k^u, can be given by

$$S_k^u = \frac{\int_{\left(T_k^u - \delta\right)}^{\left(T_k^u + \delta\right)} f_k^u(T)\ (dT)}{\sum_{k=1}^{N}\left[\int_{\left(T_k^u - \delta\right)}^{\left(T_k^u + \delta\right)} f_k^u(T)\,dT\right]} \tag{4.44}$$

$$\beta^u = \frac{\sum_{k=1}^{N} I_k - \alpha^u \sum_{k=1}^{N} I_k}{N} \tag{4.41}$$

These equations can be converted to the ones for the lower thresholding function $f_l(T)$ by substituting T_k^l for T_k^u.

4.2.7.3 Linear Regression with One Fixed Point

A problem with the linear regression approach is that the generation of the bounding function via linear regression causes the intersection points between the current levels and the thresholding functions to be shifted from their ideal locations. The ideal locations being defined as the locations which result in an exactly equal distribution of the false alarm probability across the membership functions. If the membership functions have similar transition statistics and staircase step widths then these shift are of minor concern and relate just to the balancing of the competing criteria of equal distributions and a specified false alarm rate. If the assumptions of similar transition statistics and step width do not hold, then the utilization of the linear regression approach may cause a significant shift in the balance of the false alarm probability between the membership functions due to the shifts in the intercept locations. Since the membership functions will generally not be identical in terms of their transition statistics or step width, their sensitivity to these location shifts will also not be equal. The major concern with the linear regression approach, therefore, is that these location shifts occur irrespective of the sensitivity of the given membership function's resulting false alarm probability. Ideally, it would be better to minimize the shifts for those membership functions which were more sensitive and allowing larger shifts for those functions which were less sensitive. One approach, therefore, is to fix the most sensitive membership function's intersection point to the value which produces the $\frac{\gamma}{2N}$ false alarm probability and then perform the linear regression about this

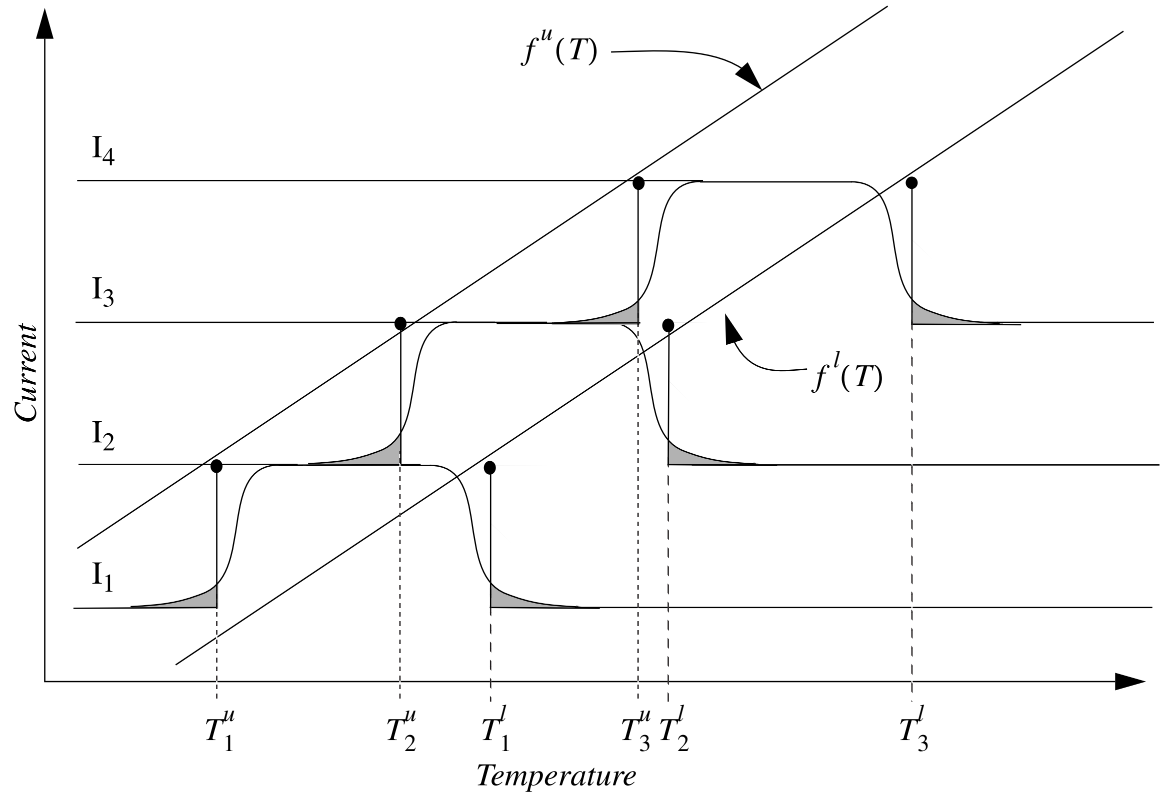


FIGURE 4-15.
Illustration of how setting a prescribed false alarm probability results in the generation of a set of current versus temperature tuples for the upper and lower thresholds under the constraint that the false alarm probability should be equally distributed.

of false alarm and the equal distribution criteria. There is no guarantee that this line though is the optimal solution, but this approach should produce a reasonable solution at a low computational cost. In particular the linear regression solution for the upper thresholding function $f^u(T) = \alpha^u T + \beta^u$ is given by

$$\alpha^u = \frac{N\sum_{k=1}^{N} T_k^u I_k - \sum_{k=1}^{N} T_k^u \sum_{k=1}^{N} I_k}{N\sum_{k=1}^{N}\left(T_k^u\right)^2 - \left(\sum_{k=1}^{N} T_k^u\right)^2} \tag{4.40}$$

and

of false alarm criterion is met. Hence, the original criterion is still met and the additional criterion is only utilized as a methodology to simplify the threshold function generation process.

The next sections will describe the three heuristic methodologies in more detail. Once these methodologies have been presented, they will then be utilized to generate the linear thresholding functions associate with the example current versus temperature behaviour shown in Figure 4-1. The resulting thresholding functions will then be compared to the near optimal linear thresholds obtained through a brute force search strategy. In this way the near optimality of these simpler heuristic approaches will be validated.

4.2.7.2 Linear Regression

The first of the heuristic methodologies is to utilize standard linear regression to determine the respective thresholding functions. Under the equal distribution policy for the probability of false alarm, each of the boundary areas is set to have a false alarm probability of $\frac{\gamma}{2N}$ if the total probability of false alarm is to be γ. For each of the membership functions, this desired individual false alarm probability relates to a specific temperature value. Therefore, selection of the overall probability of false alarm defines a set of current versus temperature tuples within the behavioural model which the linear thresholding function would have to pass through if the condition of equal false alarm probabilities is to be met. Specifically the sets of tuples $\{\left(T_k^u, I_k\right) | k = 1, \ldots, N\}$ and $\{\left(T_k^l, I_k\right) | k = 1, \ldots, N\}$ are generated for the upper and lower thresholding functions, respectively. Figure 4-15 illustrates this principle.

The problem of finding the "optimal" linear thresholding functions therefore becomes one of finding the line which best fits the set of tuples which have been defined by the equal distribution policy. The best line in a mean square error sense which passes through a set of points is given by standard linear regression. Hence, linear regression on this set of tuples can be used to find a line which should be close to meeting both the total probability

Within this optimization, the values of T_k^u and T_k^l are dependent upon the parameters α^u, α^l, β^u, and β^l. Because the values T_k^u and T_k^l occur in the limits of the integral of Eq. 4.39, the constrained optimization is very difficult to implement in practice. Additionally, the optimization, once implemented, would be computationally quite expensive since the integrals must be evaluated at each step in the optimization to ensure that the constraint is not being violated.

For these reasons the full optimization approach was not utilized within this work in the determination of the thresholding functions. Instead, three different heuristic approaches of arriving at the thresholding functions were implemented. These heuristic approaches are all computationally much simpler than the full optimization and much easier to implement. The basic idea behind each of these three heuristic methodologies is that the computational complexity of the optimization can be significantly reduced by adding an additional criterion to the optimization process. In particular, if it is assumed that, in addition to Eq. 4.39, the probability of false alarm must also be equally distributed between the membership functions, then, as will be seen, the constraint optimization problem reduced to a much simpler linear regression problem.

Obviously, the resulting linear thresholding functions may be unable to simultaneously meet both the equally distributed false alarm probability and the specific total false alarm probability criteria. In general, a simultaneous solution to these criteria is only possible if and only if the transition statistics are identical throughout the behavioural map and the staircase step widths are exactly equal. If these assumptions do not hold exactly then some deviation from these criteria is inevitable. In particular, if the total false alarm probability criterion is to be met exactly, then, under the assumption that linear thresholding functions will be employed, it will not be possible to meet the criterion that the false alarm probability be equally distributed between the membership functions. The approach utilized with the heuristic methodologies is therefore to initially begin by meeting just the equally distribution criterion. This allows initial estimates of the thresholding functions to be found. Once these functions are found, this equally distribution criterion is relaxed and the lines are adjusted, by rotating the lines about their centres such that the original total probability

$$\frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{x_k^u}^{T_k^u} f_k(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k(T)\,dT}\right] + \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{T_k^l}^{x_k^l} f_k(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k(T)\,dT}\right] = \gamma \qquad (4.39)$$

where the universe of discourse of the membership functions is given by $T \in [T_{min}, T_{max}]$.

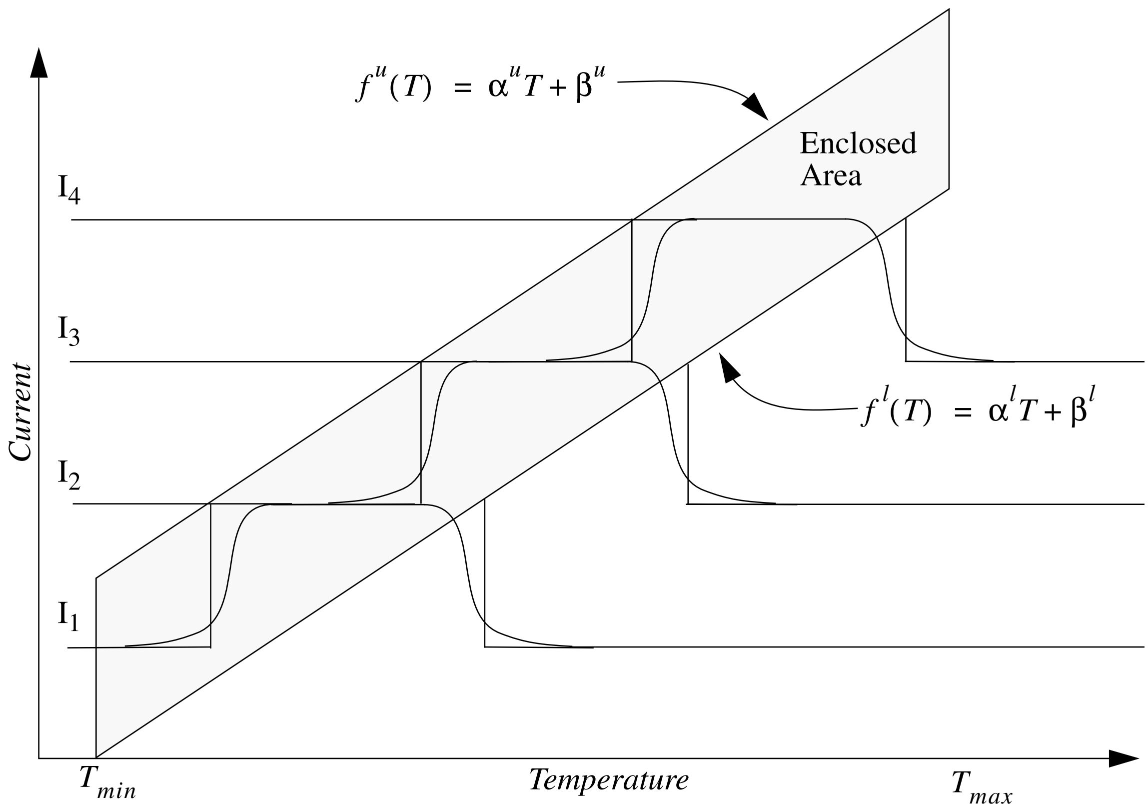


FIGURE 4-14.
Illustration of the area enclosed between the given upper and lower bounding functions which is minimized under the total probability of false alarm constraint under the constrained optimization approach.

be advantageous though has not been explored and throughout the remainder of this work it will be assumed that the probability of false alarm is balanced equally between the upper and lower thresholding functions.

It should be noted that the overall methodology itself is not limited to the use of linear thresholding functions; though, the use of higher order thresholding functions would necessitate using higher order curve fitting techniques and care would need to be taken to ensure that the higher order bounding functions did not encroach on the analytical membership functions. In any event, any non-linear thresholding functions would need to exhibit the characteristic of being monotonically increasing functions. Alternatives to linear thresholding functions were not considered within this work.

The remainder of this section will look at techniques for the generation of thresholding functions that are both accurate, in the sense of determining optimal or near optimal thresholding functions, and computationally efficient.

4.2.7.1 Full Optimization

The first approach to be considered is the most obvious one. That is to perform a full constrained optimization across the set of membership functions to determine the optimal linear thresholding functions which meet the prescribed probability of false alarm and are as tight as possible, in the sense of being as close together as possible. Conceptually this is a simple approach. Unfortunately, the constrained optimization is difficult to implement due to the nature of the constraint. There is also a high computational cost associated with performing the constrained optimization. In particular, the optimization process can be given mathematically in terms of finding the slopes and intercepts which minimize the enclosed area between the two thresholding functions. This enclosed area is illustrated in Figure 4-14 and is given mathematically by

$$J = \frac{\alpha^u - \alpha^l}{2}\left(T_{max}^2 - T_{min}^2\right) + \left(\beta^u - \beta^l\right)(T_{max} - T_{min}) \quad (4.38)$$

subject to the constraint that

4.2.6.2.4 Inverse Calculation

For the pseudo-Gaussian membership functions, the functional inverses of Eq. 4.34 and Eq. 4.35 are given by

$$T_k^u = \mu_k^u - \sqrt{2}\sigma_k^u erf^{-1}\left(C_k^u - \sigma_k^u A_k p_k^u \right) \tag{4.36}$$

and

$$T_k^l = \mu_k^l + \sqrt{2}\sigma_k^l erf^{-1}\left(C_k^l - \sigma_k^l A_k p_k^l \right) \tag{4.37}$$

where $erf^{-1}(.)$ is the inverse error function defined such that if $y = erf(x)$ then $x = erf^{-1}(y)$.

4.2.7 Linear Thresholding Function Generation

Once suitable analytical membership functions have been found, the next issue to be addressed is how to utilize them to generate the upper and lower thresholding functions in some optimal or near optimal way such that the prescribed probability of false alarm criterion for the resulting detection system is met. In part, this issue becomes one of determining a policy with regards to how the prescribed probability of false alarm should be "shared" between the upper an lower bounding functions. One approach is to share the prescribed probability of false alarm equally between the upper and lower thresholds. Under this policy, a prescribed total probability of false alarm γ for the given behavioural model would require that the linear thresholding functions each achieved a false alarm probabilities of $\frac{\gamma}{2}$.

This policy is a reasonable policy if, as in the case of the example plant, no *a priori* information is available regarding how the false alarm probability should be shared between the thresholding functions. Unequal balancing of the false alarm probability though may be desirable in some cases. Hence, this asymmetrical balancing is allowed within the proposed methodology. The nature of when an asymmetrical balancing would

4.2.6.2.2 Upper Bound Equation

The probability of this particular membership function generating false alarms due to exceeding the given upper threshold can therefore be given by

$$p_k^u(false\ alarm) = \frac{1}{A_k \sigma_k^u}\left[C_k^u - erf\left(\frac{|T_k^u - \mu_k^u|}{\sqrt{2}\sigma_k^u}\right)\right] \tag{4.34}$$

where $C_k^u = erf\left(\frac{|\mu_{k-1}^l - \mu_k^u|}{\sqrt{2}\sigma_k^u}\right), \frac{1}{\sigma_k^u}\left[C_k^u - erf\left(\frac{|T_k^u - \mu_k^u|}{\sqrt{2}\sigma_k^u}\right)\right]$ is the area under the given membership function's right boundary (the numerator in Eq. 4.15) and A_k is the total area under the given membership function which was given by Eq. 4.29 (the denominator in Eq. 4.15).

4.2.6.2.3 Lower Bound Equation

The probability of this particular membership function generating false alarms due to exceeding the given upper threshold can therefore be given by

$$p_k^l(false\ alarm) = \frac{1}{A_k \sigma_k^l}\left[C_k^l - erf\left(\frac{|T_k^l - \mu_k^l|}{\sqrt{2}\sigma_k^l}\right)\right] \tag{4.35}$$

where $C_k^l = erf\left(\frac{|\mu_{k+1}^u - \mu_k^l|}{\sqrt{2}\sigma_k^l}\right), \frac{1}{\sigma_k^l}\left[C_k^l - erf\left(\frac{|T_k^l - \mu_k^l|}{\sqrt{2}\sigma_k^l}\right)\right]$ is the area under the given membership function's right boundary (the numerator in Eq. 4.15) and A_k is the total area under the given membership function which was given by Eq. 4.29 (the denominator in Eq. 4.15).

4.2.6.2.1 Parameter Selection

The membership function of Eq. 4.28 has four parameters that must be determined prior to its use, namely the means, μ_k^u and μ_k^l, and standard deviations, σ_k^u and σ_k^l, of the two pseudo-Gaussian boundaries. A simple way to generate these parameters is to utilize a gradient descent optimization. In particular, for each current level, I_k, the cost function J_k can be minimized via a gradient descent optimization procedure where

$$J_k = \frac{1}{2}\sum_{j=1}^{N_k} [f_k(T_j) - g_k(T_j)]^2 \tag{4.31}$$

$$\nabla J_k = \sum_{j=1}^{N_k} [f_k(T_j) - g_k(T_j)]\, \nabla f_k(T_j) \tag{4.32}$$

$$\nabla f_k(T_j) = \begin{cases} \nabla_u\left[e^{-\frac{\left(x-\mu_k^u\right)^2}{2\left(\sigma_k^u\right)^2}} \right] & \text{if} \quad T_j < \mu_k^u \\ 0 & \text{if } \mu_k^u \le T_j \le \mu_k^l \\ \nabla_l\left[e^{-\frac{\left(x-\mu_k^l\right)^2}{2\left(\sigma_k^l\right)^2}} \right] & \text{if} \quad T_j > \mu_k^l \end{cases} \tag{4.33}$$

and $\nabla_u(.) = \left[\frac{\partial}{\partial\mu_k^u}\ \frac{\partial}{\partial\sigma_k^u}\right]^T$ and $\nabla_l(.) = \left[\frac{\partial}{\partial\mu_k^l}\ \frac{\partial}{\partial\sigma_k^l}\right]^T$.

where *erf(x)* is the standard error function given by $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-x^2}dx$ (4.30)

Basically, this membership function consists of asymmetrical Gaussian boundaries and a constant core. The asymmetric Gaussian boundaries model the transition areas to and from the given dependent signal levels and the constant core models the flat areas within a given dependent signal level. This particular analytical function was chosen both because the Gaussian shape of its boundaries closely models the actual statistical behaviour of the current-temperature maps transition areas and because, since this function is based on a Gaussian function, it is also simple to compute the required area calculations through the used of the standard error function, *erf(.)*. In addition, the boundary between the Gaussian sections and the central constant area are smooth since the Gaussian function derivatives are zero at the mean. It should be noted though that this membership function does not meet condition 2 given by Eq. 4.14. In particular, since if *f(x)* is a Gaussian function then *1-f(x)* by definition cannot be Gaussian; hence, it is not possible to have a membership function with Gaussian transition areas which satisfies both condition 1 (symmetry) and condition 2 (point-by-point summation equalling one).

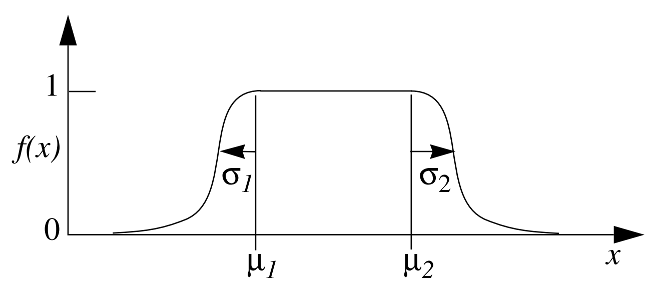


FIGURE 4-13.
Pseudo-Gaussian fuzzy membership function.

In Section 4.2.8 the relative merits of both the sigmoidal and pseudo-Gaussian membership functions in modeling the transition effects present in the current versus temperature behaviours will be explored in more detail.

$$T_k^l = \frac{1}{\alpha_{k+1}} \ln\left(\frac{C_k^l e^{p_k^l A_k \alpha_{k+1}}}{1 - C_k^l e^{p_k^l A_k \alpha_{k+1}}} \right) + \mu_{k+1} \tag{4.27}$$

These formulae are then used to generate the sets of upper and lower bound intersection points, used in the threshold function determination process, given by $\{\left(I_k, T_k^u \right) \middle| k = 1, \ldots, N\}$ and $\{\left(I_k, T_k^l \right) \middle| k = 1, \ldots, N\}$ which are associated with a particular individual false alarm probability given by $p_k^u = \frac{\gamma}{2N}$ and $p_k^l = \frac{\gamma}{2N}$.

4.2.6.2 Pseudo-Gaussian Membership Function

The other analytical membership function which was used in the modeling of the current versus temperature behaviour was a pseudo-gaussian membership function, shown in Figure 4-13, and described analytically by

$$f_k(x) = \begin{cases} 0 & \text{if } x < \mu_{k-1}^u \\ e^{-\frac{\left(x - \mu_k^u\right)^2}{2\left(\sigma_k^u\right)^2}} & \text{if } \mu_{k-1}^u \le x < \mu_k^u \\ 1 & \text{if } \mu_k^u \le x \le \mu_k^l \\ e^{-\frac{\left(x - \mu_k^l\right)^2}{2\left(\sigma_k^l\right)^2}} & \text{if } \mu_k^l < x < \mu_{k+1}^u \\ 0 & \text{if } x \ge \mu_{k+1}^u \end{cases} \tag{4.28}$$

and has an area of support given by $\left[\mu_{k-1}^l, \mu_{k+1}^u \right)$, and a total area given by

$$A_k = \sqrt{\frac{\pi}{2}} \left(\sigma_k^u + \sigma_k^l \right) + \mu_k^l - \mu_k^u + \frac{1}{\sigma_k^u} \left[erf\left(\frac{\left|\mu_{k-1}^l - \mu_k^u\right|}{\sqrt{2}\sigma_k^u} \right) - 1 \right] + \frac{1}{\sigma_k^l} \left[erf\left(\frac{\left|\mu_{k+1}^u - \mu_k^l\right|}{\sqrt{2}\sigma_k^l} \right) - 1 \right] \tag{4.29}$$

4.2.6.1.3 Lower Threshold Probability of False Alarm

Similarly, once the analytical membership functions have been determined and a particular lower thresholding function, $f^{l}(T)$, has been identified, the set of intersections points $\{T_k^l \big| k = 1, \ldots, N\}$ can be found and these points can then be utilized to find the individual probabilities of false alarm, p_k^l, given by

$$p_k^l\,(false\ alarm) = \frac{1}{\alpha_{k+1}A_k}\ln\left(\frac{\left[e^{\alpha_{k+1}\left(T_k^l - \mu_{k+1}\right)}\right]}{C_k^l\left[1 + e^{\alpha_{k+1}\left(T_k^l - \mu_{k+1}\right)}\right]}\right) \tag{4.25}$$

where $C_k^l = \frac{\left[e^{\alpha_{k+1}(\beta_{k+1} - \mu_{k+1})}\right]}{\left[1 + e^{\alpha_{k+1}(\beta_{k+1} - \mu_{k+1})}\right]}$, $\frac{1}{\alpha_{k+1}}\ln\left(\frac{\left[e^{\alpha_{k+1}\left(T_k^l - \mu_{k+1}\right)}\right]}{C_k^l\left[1 + e^{\alpha_{k+1}\left(T_k^l - \mu_{k+1}\right)}\right]}\right)$ represent the given membership function's right boundary area (the numerator of Eq. 4.15), and A_k, given by Eq. 4.19, represents the total area under the given membership function (the denominator of Eq. 4.15).

4.2.6.1.4 Inverse Calculation

Obviously, within the process of setting the linear thresholding functions such that a prescribed probability of false alarms is obtained, it is desirable to be able to easily determine, for a given analytical membership function, what value of T will produce a given false alarm rate. For Eq. 4.24 and Eq. 4.25, the formulae for the respective inverses are given by

$$T_k^u = \frac{1}{\alpha_k}\ln\left(C_k^u e^{p_k^u A_k \alpha_k} - 1\right) + \mu_k \tag{4.26}$$

and

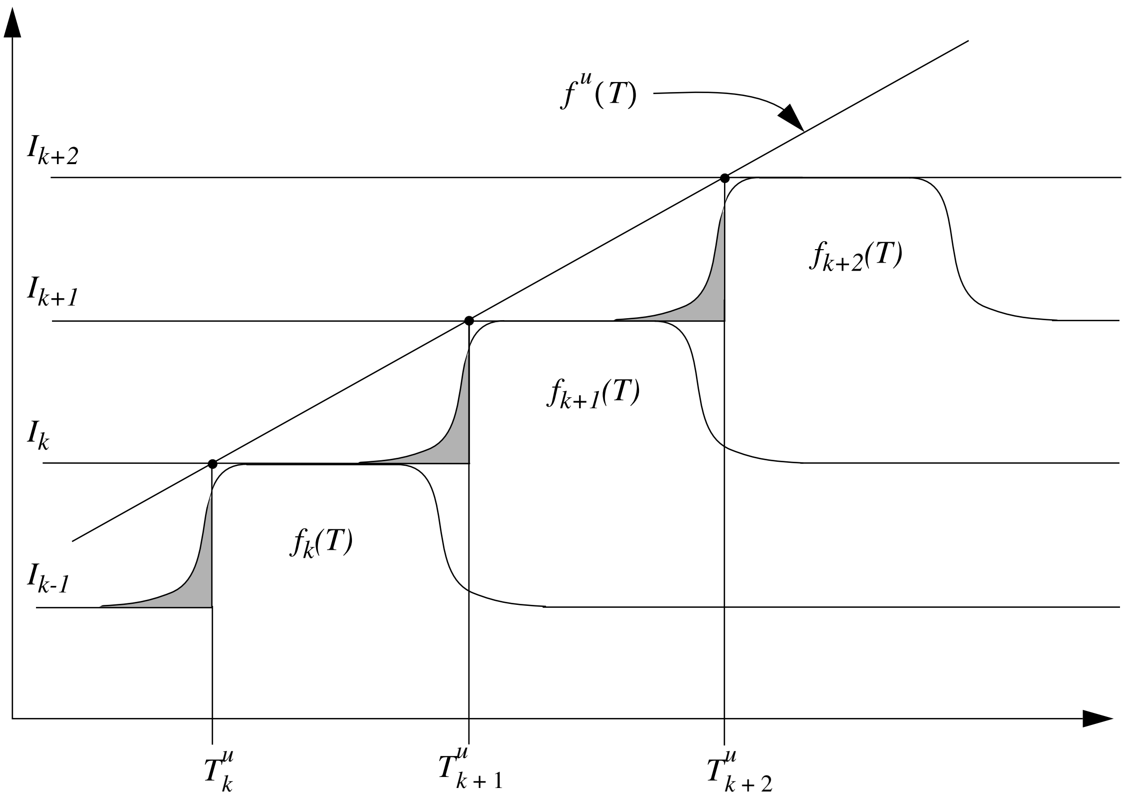


FIGURE 4-12.

Selection of the set $\{T_k^u\}$ to generate prescribed false alarm rates for each of the given membership functions.

where $C_k^u = \dfrac{\left[1 + e^{-\alpha_k(\beta_{k-1} - \mu_k)}\right]}{\left[e^{-\alpha_k(\beta_{k-1} - \mu_k)}\right]}$, $\dfrac{1}{\alpha_k}\ln\left(\dfrac{\left[1 + e^{-\alpha_k\left(T_k^u - \mu_k\right)}\right]}{C_k^u\left[e^{-\alpha_k\left(T_k^u - \mu_k\right)}\right]}\right)$ represent the given membership function's left boundary area (the numerator of Eq. 4.15), and A_k, given by Eq. 4.19, represents the total area under the given membership function (the denominator of Eq. 4.15).

of these regions, $\{T_k^*\}$, give an accurate estimate of the $\{\beta_k\}$ in the sense that the error given by $|f_k(\beta_k) - g_k(T_k^*)| \approx 0$ since $g_k(T_k^*) = 1$ by definition and $f_k(\beta_k) \approx 1$ provided that $f_k'(\beta_k) \approx 0$. The interval used in the optimization procedure can therefore be specified in practice as $T \in (T_k^*, T_{k+1}^*)$.

This approach works well for the membership which are completely contained within the behavioural model. For the membership functions $f_I(T)$ and $f_N(T)$, though, there is a slight problem in that only information regarding, respectively, their right and left boundaries is contained within the raw behavioural map. Their indescribable boundary area exists outside of the temperature range covered by the raw behavioural map. For these two membership functions, the mid-points, β_1 and β_N, can be estimated, respectively, by the minimum temperature at which $g_1(T) = 1$ and the maximum temperature at which $g_N(T) = 1$.

4.2.6.1.2 Upper Threshold Probability of False Alarm

Once the analytical membership functions have been identified and a particular upper thresholding function, $f^u(T)$, has been determined, the resulting probability of false alarm can then be found. More particularly, if the set of intersection points between the upper thresholding function and the N current levels of the raw behavioural map are given by $\{T_k^u | k = 1, \ldots, N\}$ (as shown in Figure 4-12) then the individual probabilities of false alarm, p_k^u, due to the upper threshold, can be given by

$$p_k^u(false\ alarm) = \frac{1}{\alpha_k A_k} \ln\left(\frac{\left[1 + e^{-\alpha_k\left(T_k^u - \mu_k\right)}\right]}{C_k^u\left[e^{-\alpha_k\left(T_k^u - \mu_k\right)}\right]} \right) \tag{4.24}$$

$$f_k(x) = 1 - f_{k-1}(x), \quad \beta_k \le x < \beta_{k+1} \tag{4.20}$$

The process of determining the appropriate parameters required to construct the behavioural model from the raw behavioural map is basically one of modeling the left boundaries of the raw membership functions. This is done quite easily by utilizing a gradient descent optimization procedure to minimize

$$J_k = \frac{1}{2}\sum_T [f_k(T) - g_k(T)]^2 \qquad T \in (\beta_k, \beta_{k+1}) \tag{4.21}$$

where

$$\nabla J_k = \sum_T [f_k(T) - g_k(T)]\, \nabla f_k(T) \qquad T \in (\beta_k, \beta_{k+1}) \tag{4.22}$$

$$\nabla f_k(T) = \begin{cases} \nabla_k\left(\dfrac{1}{1 + e^{-\alpha_k(T - \mu_k)}}\right) & \text{if } x < \beta_k \\ \nabla_{k+1}\left(\dfrac{1}{1 + e^{\alpha_{k+1}(T - \mu_{k+1})}}\right) & \text{if } x \ge \beta_k \end{cases} \tag{4.23}$$

and $\nabla_k(.) = \left[\dfrac{\partial}{\partial\alpha_k}\ \dfrac{\partial}{\partial\mu_k}\right]^T$.

If this optimization is procedure is performed for each of the $k = 1, \ldots, N-1$ right boundaries of the membership functions then the two parameter sets given by $\{\alpha_1, \ldots, \alpha_{N-1}, \alpha_{N-1}\}$ and $\{\mu_1, \ldots, \mu_{N-1}, \mu_{N-1}\}$ can be produced. Obviously, in practice the set of mid-points $\{\beta_1, \ldots, \beta_N\}$ are not known until the membership functions' parameters have actually been found. Therefore, in practice the interval upon which T is defined in Eq. 4.21 must be estimated. Fortunately, a reasonable estimate of this interval can be obtained by identifying the regions over which the $g_k(T) = 1$. The midpoint

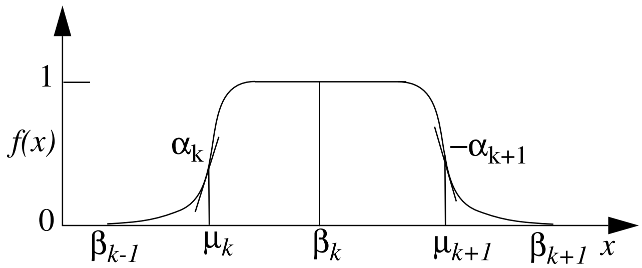


FIGURE 4-11.
Sigmoidal fuzzy membership function.

desirable from a theoretical point of view since these analytical membership function are intended to provide a behavioural model of the current versus temperature behaviour. It should also be noted though that this membership function is not continuous. A discontinuity in the derivative of $f_k(x)$ will exist at point β_k. This discontinuity though will tend to be quite small if the slopes of the respective sigmoid functions are approximately zero in the neighbourhood of β_k. The total area under this membership function, which is required in the generation of the false alarm probabilities, is given by

$$A_k = \frac{1}{\alpha_{k+1}} \ln\left(\frac{\left[e^{\alpha_{k+1}(\beta_k - \mu_{k+1})}\right]\left[1 + e^{\alpha_{k+1}(\beta_{k+1} - \mu_{k+1})}\right]}{\left[1 + e^{\alpha_{k+1}(\beta_k - \mu_{k+1})}\right]\left[\left(e^{\alpha_{k+1}(\beta_{k+1} - \mu_{k+1})}\right)\right]} \right) + \frac{1}{\alpha_k} \ln\left(\frac{\left[1 + e^{-\alpha_k(\beta_k - \mu_k)}\right]\left[e^{-\alpha_k(\beta_{k-1} - \mu_k)}\right]}{\left[e^{-\alpha_k(\beta_k - \mu_k)}\right]\left[1 + e^{-\alpha_k(\beta_{k-1} - \mu_k)}\right]} \right) \quad (4.19)$$

4.2.6.1.1 Parameter Selection

If the membership function of Eq. 4.18 is to be used to model a given raw membership function then the four parameters (α_k, α_{k+1}, μ_k, and μ_{k+1}) need to be identified. Since these membership functions do not exist in isolation, due to the requirements imposed by Eq. 4.14, only two of these parameters (α_{k+1} and μ_{k+1}) are actually unknown. In particular, once all the right boundaries have been found for the behavioural model then the left boundaries are already determined since

4.2.6.1 Sigmoidal Membership Function

The sigmoidal analytical membership function is based on the combination of two sigmoidal functions. In particular, for the k^{th} membership function the right boundary is defined in terms of independent sigmoid function, specified by its centre μ_k and slope α_k. In order to preserve the condition imposed by Eq. 4.14, the k^{th} sigmoidal function's left boundary must be given by $1 - f_{k-1}(x)$ (i.e one minus the right boundary of the subsequent membership function)[4]. The k^{th} membership function can therefore be expressed analytically by the iterative formulation

$$f_k(x) = \begin{cases} 0 & \text{if } x > \beta_{k-1} \\ \dfrac{1}{1 + e^{-\alpha_k (x - \mu_k)}} & \text{if } \beta_{k-1} \le x < \beta_K \\ \dfrac{1}{1 + e^{\alpha_{k+1} (x - \mu_{k+1})}} & \text{if } \beta_k \le x < \beta_{k+1} \\ 0 & \text{if } x \ge \beta_{k+1} \end{cases} \tag{4.18}$$

where $\beta_k = \dfrac{\alpha_k \mu_k - \alpha_{k+1} \mu_{k+1}}{\alpha_k + \alpha_{k+1}}$ represents the intersection point between the sigmoid functions which respectively describe the k^{th} membership function's left and right boundaries, and interval of support is given by $[\beta_{k-1}, \beta_{k+1})$. A graphical representation of this sigmoidal membership function is shown in Figure 4-11.

This membership function meets most of the desired mathematical properties given in Section 4.2.4. The only exception being that it does not necessarily provide the "best" modeling of the current versus temperature transition areas, as will be seen in Section 4.2.8. The sigmoidal membership function does though meet condition 2 which is quite

---

4. Equivalently, the choice could have been made to have the right boundary defined by $1\text{-}f_{k+1}(x)$ to preserve the condition imposed by Eq. 4.14. This would have changed the specifics of the resulting formulation but not the general intent. To meet Eq. 4.14 the complete membership function must be composed of one independent boundary and one boundary which depends on either the previous or subsequent membership function, depending upon which boundary has been chosen as the independent one.

This represents one specific test which can be used to determine the applicability of the fuzzy membership function based modeling techniques. Within the framework of the modeling technique though, other specific formulations for applicability tests can be used depending on the exact nature of the system to be modeled. For example, it may be desirable in some instances, not to weigh each of the membership functions' modeling accuracy equally in the calculation of the percentage error. For example, it may be more desirable to scale the modeling accuracy by the probability that the component's behaviour will actually occur within the given portion of the behavioural map. Within this work the use of test other than that of Eq. 4.17 has not been explored as this has been left as an area of future work.

The importance of having a direct method of determining the applicability of the modeling technique is that failure to model, in itself, can be used as an indicator of a fault condition within the engineering plant. For example, in [66] it was shown, for the example plant, that the amplifiers' whose current-temperature behavioural maps failed to follow the general staircase function, and hence would have failed the test for appropriateness of the fuzzy membership modeling technique, were in fact not operating "normally". The test for appropriateness therefore provides an additional means of performing fault detection which is not available within traditional limit checking approaches.

4.2.6 Example Analytical Membership Functions

The material above outlined the process by which generic analytical membership functions can be utilized to generate a behavioural model. The next two sections will introduce the two specific analytical membership functions, and their accompanying probability of false alarm formulae, which were evaluated within this work to model the statistics of the current versus temperature transition areas. These functions are the sigmoidal membership function and the pseudo-Gaussian membership function. Both of these functions meet some, but not all, of the desired mathematical properties outlined in Section 4.2.4.

$$p_k^l(false\ alarm) = \frac{\int_{T_k^l}^{x_k^l} f_k(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k(T)\,dT} \tag{4.16}$$

4.2.5 Test for Appropriateness

It is important in applying any modeling techniques to have a means of directly determining if the given technique applies to the particular data set to which it is being employed. This is particularly important within the context of automated fault detection systems since without this ability the detection system looses much of its automated nature. For the fuzzy membership function based modeling technique outlined above a straight forward method is available to determine the applicability of the approach. Each of the analytical membership functions models the statistical behaviour of the current signal over a prescribed temperature range to a given degree of accuracy, which can be numerically assessed by computing the percentage error between the analytical membership function and its sampled counterpart. Placing a threshold on the total allowable percentage error across the complete set of membership functions, generated over a given current-temperature data sequence, directly allows the applicability of the fuzzy membership function based modeling technique to be determined. Under this approach, the test for applicability can be given mathematically as

$$J = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{M}\left(\frac{|f_k(T_i) - g_k(T_i)|}{g_k(T_i)}\right) < \lambda \tag{4.17}$$

where λ is a threshold on the allowable percentage difference between the analytical membership functions and their sampled counterparts. If $J > \lambda$ then the fuzzy-membership based modeling technique is deemed not to accurately model the given current-temperature sequence.

4.2.4.2 Lower Bound False Alarm Probability

For the lower thresholding function, the probability of false alarm associated with a general membership function relates to an area contained within the membership function's right boundary and which is shaded gray in Figure 4-10. It is slightly more difficult to see, in this case, how this area relates to the occurrence of current levels which will exceed the lower thresholding functions. This property can be seen more clearly if it is

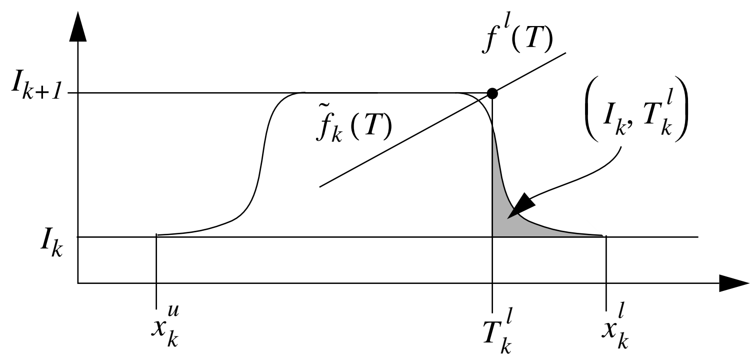


FIGURE 4-10.
Effect of the lower thresholding function $f^l(T)$ on a given membership function $\tilde{f}_k(T)$.

recalled that when $f_k(T) = 1$, or equivalently $\tilde{f}_k(T) = I_{k+1}$, the probability of current level I_{k+1} occurring for the given temperature T is 1. Hence, the area to the right of the intersection point between $y = I_{k+1}$ and $f^l(T)$ relates to the occurrence of the current level I_k, which would cause false alarms to be generated. The lower threshold false alarm rate for a given membership function at current level I_{k+1} is therefore defined by this intersection point T_k^l. Mathematically this probability of false alarm can be given by

4.2.4.1 Upper Bound False Alarm Probability

For the upper thresholding function, the probability of false alarm associated with a general membership function $f_k(T)$ relates to the area contained within the membership function's left boundary and which is shaded gray in Figure 4-9. It is easy to see, in Figure 4-7, that this area relates to the occurrence of current levels which will exceed the upper thresholding function since transitions in this area are related to the occurrence of current levels at the next higher quantization step. The upper threshold false alarm rate for a given membership function at current level I_k is therefore defined by the intersection of the upper thresholding function and the line given by $y = I_{k+1}$ which occurs at point T_k^u. Mathematically this probability of false alarm determined from the behavioural model's functional components, $\tilde{f}_k(T)$, is identical to the probability of false alarm derived previously in Eq. 4.13. Specifically,

$$p_k^u(false\ alarm) = \frac{\int_{x_k^u}^{T_k^u} f_k(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k(T)\,dT} \tag{4.15}$$

$$= \frac{\int_{x_k^u}^{T_k^u} [\,(I_{k+1} - I_k) f_k(T) + I_k - I_k]\,dT}{\int_{x_k^u}^{x_k^l} [\,(I_{k+1} - I_k) f_k(T) + I_k - I_k]\,dT}$$

$$= \frac{\int_{x_k^u}^{T_k^u} \left(\tilde{f}_k(T) - I_k\right) dT}{\int_{x_k^u}^{x_k^l} \left(\tilde{f}_k(T) - I_k\right) dT}$$

There are also certain mathematical properties that are desirable, for computational and theoretical reasons, for the analytical membership functions to posses. In particular, these are that

1. The membership functions should be symmetric in that the same functional form should be utilized to generate both the right and left boundaries.
2. The summation across the set of membership functions, $\{f_k(T),\ k=1,\ldots,N\}$, should equal to 1 for all $T \in \Re$.

$$\sum_{k=1}^{N} f_k(T) = 1 \qquad T \in \Re \tag{4.14}$$

3. The membership functions boundaries should closely approximate the statistics of the raw behavioural maps transition areas (Eq. 4.7).
4. Ideally, the membership function's integral from x to ∞ should be easily calculable.

The following two sections will look at how the probability of false alarm can be calculated given a particular setting of the upper and lower thresholding functions in more detail.

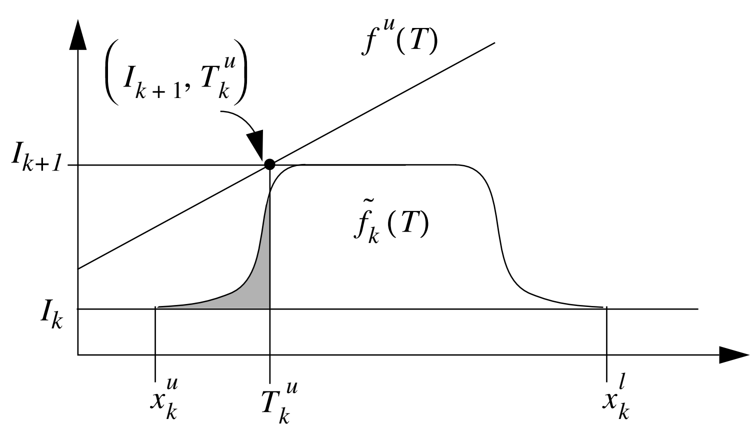


FIGURE 4-9.

Effect of the upper thresholding function $f^u(T)$ on a given membership function $\tilde{f}_k(T)$.

where N is the number of membership functions whose areas of support are within the temperature region which is assumed to be uniformly distributed[3].

Obviously the uniform distribution assumption significantly simplifies the false alarm probability formulation, but this simplification is at the cost of introducing additional errors into the modeling process. As will be seen in Section 4.2.8, these additional errors are small and do not significantly affect the resulting probability of false alarm estimate. The reason why this simplifying approximation does significantly alter the probability of false alarm estimate is two fold. First, within the central mass of the distribution the uniform distribution assumption results in the introduction of a small error, as seen in Figure 4-8. The uniform distribution closely models the true distribution over for the temperature contained within the central mass. For temperatures outside the central mass, a significant error is incurred under the uniform distribution assumption, but this error is encountered infrequently since these temperatures occur infrequently.

The total error introduced by the uniform distribution approximation is, therefore, obtained as the summation of the small errors associated with the central mass of the distribution, which occur most of the time, and the larger errors associated with the other temperatures, which occur infrequently. Hence, the total error will tend to be weighted towards the small errors associated with approximating the central mass of the temperature distribution with a uniform distribution and, hence, the total error will tend to be small. Obviously, other more complex estimates of the true temperature distribution can also be utilized within Eq. 4.11. The nature and formulation of these estimates, though has been left as an area of future work. The advantage of the first order approximation provided by assuming that $p_{Temp}(T)$ is uniformly distributed, though, is that the probability of false alarm formulation becomes independent of the temperature distribution. This independence with respect to the temperature distribution would not occur if other more complex estimates of the temperature distribution were employed.

---

3. This definition of N assumes that the area modeled as being uniformly distributed is such that contains only complete (whole) membership functions. Therefore, Eq. 4.13 represents a simplification of the more general case. This simplified formulation was utilized to aid the understanding of the presented concepts which would have been obscured by the complexity of the general formulation.

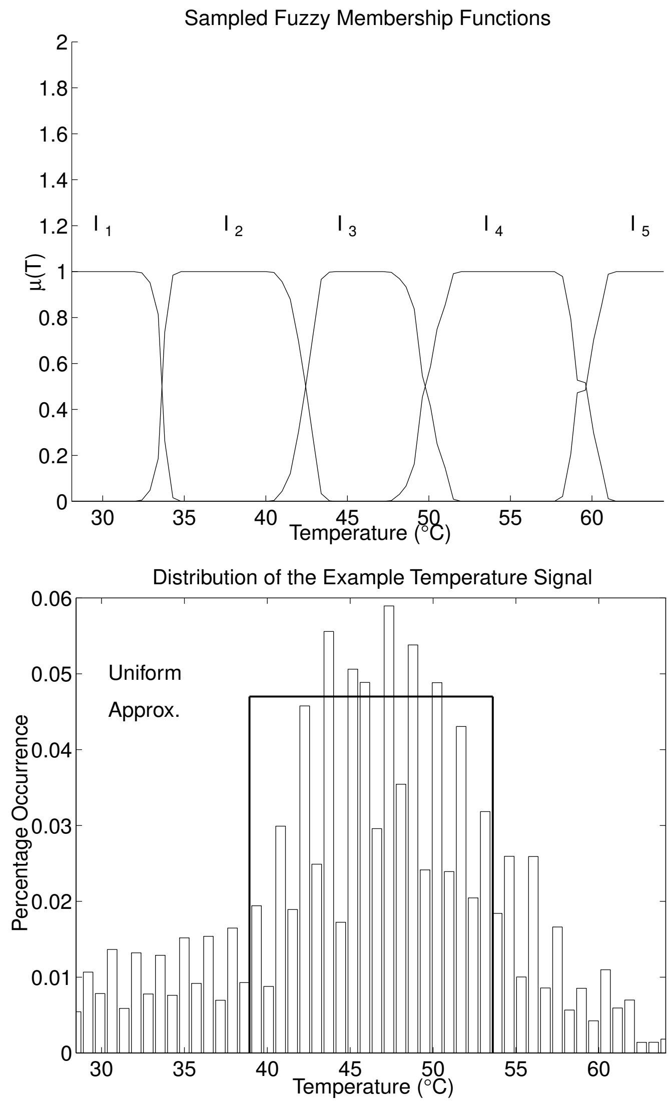


FIGURE 4-8.
Approximation of the actual example temperature signal distribution's central mass by a uniform distribution, where the central mass has been defined as one standard deviation either side of the mean.

$$p\,(false\ alarm) \;=\; \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{x_k^u}^{T_k^u} f_k\,(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k\,(T)\,dT}\right] + \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{T_k^l}^{x_k^l} f_k\,(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k\,(T)\,dT}\right] \tag{4.13}$$

the general case, by the summation of the probability of false alarm associated with each of the individual membership functions weighted by the probability distribution of the temperature.

$$
\begin{aligned}
p\,(false\ alarm) &= \frac{1}{N}\sum_{k=1}^{N}\left[\int_{x_k^u}^{T_k^u} p_k^u\,(false\ alarm)\,p_{\text{Temp}}\,(T)\,dT\right] + \\
&\quad \frac{1}{N}\sum_{k=1}^{N}\left[\int_{T_k^l}^{x_k^l} p_k^l\,(false\ alarm)\,p_{\text{Temp}}\,(T)\,dT\right] \\
&= \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{x_k^u}^{T_k^u} f_k\,(T)\,p_{\text{Temp}}\,(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k\,(T)\,p_{\text{Temp}}\,(T)\,dT}\right] + \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\int_{T_k^l}^{x_k^l} f_k\,(T)\,p_{\text{Temp}}\,(T)\,dT}{\int_{x_k^u}^{x_k^l} f_k\,(T)\,p_{\text{Temp}}\,(T)\,dT}\right]
\end{aligned}
\tag{4.12}
$$

where N is the number of membership functions occurring in the given behavioural model, and $p_{Temp}(T)$ is the probability distribution of the temperature signal.

Obviously, this formulation is quite difficult to work with in practice if the temperature distribution, $p_{Temp}(T)$, is assumed to be an arbitrary probability distribution. Figure 4-8 shows the temperature distribution associated with the example raw membership functions which were shown in Figure 4-3. By definition, most of the time the portion of the behavioural model associated with the central mass of the temperature distribution will be utilized in the modeling of the current versus temperature behaviour. If the central mass of the temperature distribution is estimated using a uniform distribution, as shown in Figure 4-8, then Eq. 4.12 can be significantly simplified. In particular, if $p_{\text{Temp}}\,(T) = \xi$ then the probability of false alarm given in Eq. 4.12 will be independent of the temperature distribution with the region over which the temperature distribution is assumed to be uniform.

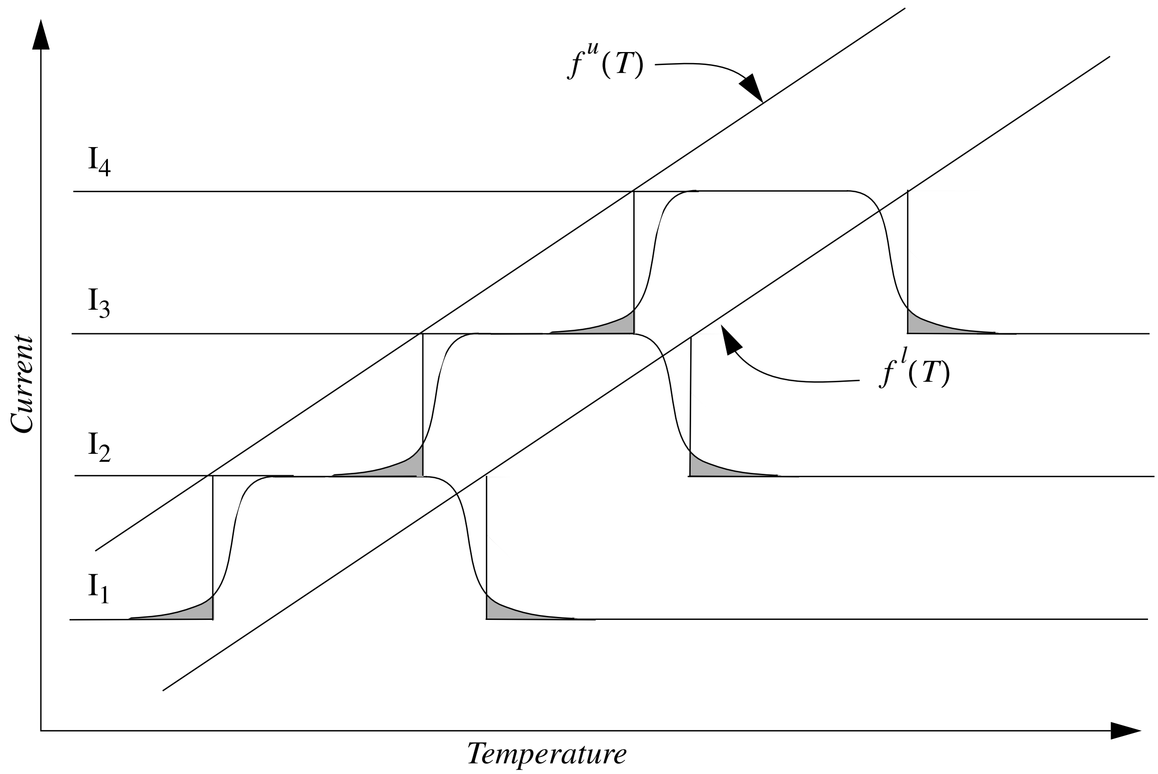


FIGURE 4-7.
Constructed, behavioural model with associated upper and lower linear thresholding functions.

membership functions are replaced with their analytical counterparts then the same computational process can be performed with respect to the behavioural model, formed by scaled, offset versions of the analytical membership functions.

$$\tilde{f}_k(T) = (I_{k+1} - I_k) f_k(T) + I_k \tag{4.11}$$

Figure 4-7 shows the effect of placing linear thresholding functions, $f^u(T)$ and $f^l(T)$, above and below the region of "normal" operation on the constructed behavioural model. The regions shaded in gray represent the portions of the behavioural model that will cause false alarms to be generated given the particular choice of thresholding functions. Hence, the complete detection system's probability of false alarm can be given, in

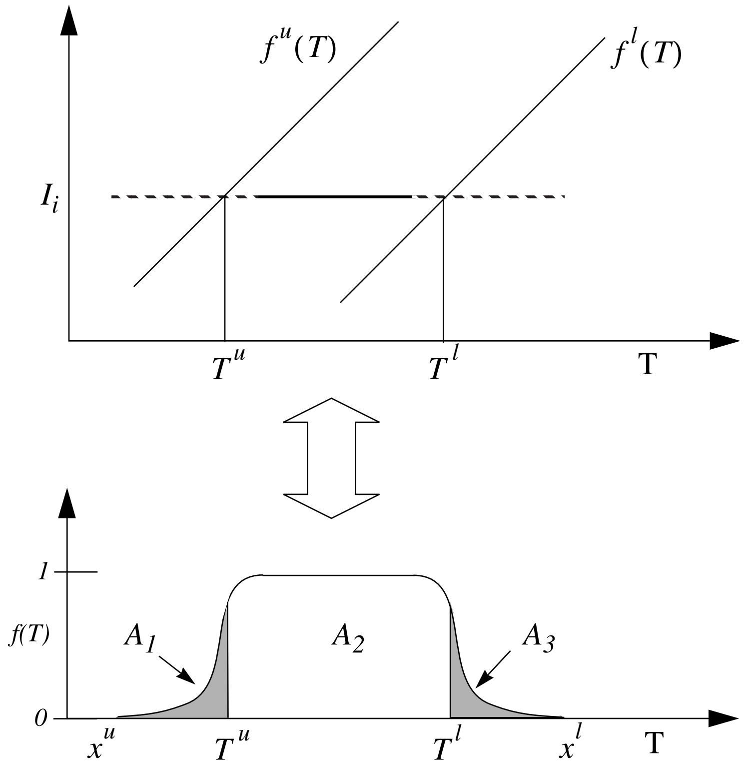


FIGURE 4-6.
Illustration of how the fuzzy membership functions can be used in the computation of the probability of false alarm associated with a given current level and given upper and lower bounding functions.

$$p_k(false\ alarm) = \frac{A_1 + A_3}{A_1 + A_2 + A_3} = \frac{\int_{x^u}^{T^u} f_k(T)\,dT + \int_{T^l}^{x^l} f_k(T)\,dT}{\int_{x^u}^{x^l} f_k(T)\,dT} \tag{4.10}$$

Obviously, the total probability of false alarm, associated with the given upper and lower bounding functions, can then be obtained by applying this approach to each of the current levels occurring within the behavioural map and combining the results. If the raw

4.2.4 Determination of the Probability of False Alarm

The question to be answered at this point is how the probability of false alarm, associated with given upper and lower linear bounding functions, can be calculated from the behavioural model. Taking a step back to the raw behavioural map, it is obvious that the total probability of false alarm can be computed as the summation of the individual false alarm probabilities computed in turn for each of the individual current levels. In particular, for a given current level, I_k, the effect of the upper and lower bounding functions is to partition the given level into three distinct sections: a central section in which the given current level is expected to occur, and right and left hand sections (bounded by $T<T^u$ and $T>T^l$) which generate false alarms (if there exist current samples at current level I_k for either $T<T^u$ or $T>T^l$).The nature of this partitioning is shown in Figure 4-6. If the total number of current samples occurring at quantization level I_k is assumed to be N_k then the false alarm probability associated with the given setting of the upper and lower thresholding functions, $f^l(T)$ and $f^u(T)$, can be given by

$$p\,(\text{false alarm}) = p\left(I_k \middle| T < T^u\right) + p\left(I_k \middle| T > T^l\right) \tag{4.9}$$

These probabilities can be computed directly by analyzing the raw fuzzy membership function which describes the statistical behaviour of the data at current level I_k, which it can be recalled is described by the k^{th} membership function. In particular, the setting of the upper and lower bounding functions also causes the fuzzy membership function to be partitioned into three sections, as shown in Figure 4-6. Specifically these sections are a middle area (A_2) related to the normal or expected occurrence of current level I_k and areas to the left and right (A_1 and A_3) related to the generation of current level I_k at lower or higher temperatures than expected. If the raw fuzzy membership function is converted into a probability density function (by dividing through by its total area) then the probability of false alarm associate with the given bound settings for current level I_k can be computed directly from the membership function $f_k(T)$ as

is minimized through the use of the gradient given by

$$\nabla J_k = \sum_{j \in Z} (f_k(T_j) - g_k(T_j)) \nabla f_k(T_j) \tag{4.8}$$

It is important to note, within this context, that the mathematical simplicity of the required fuzzy membership function is implicitly related to the assumption that transitions between multiple quantization steps occur rarely and are not generally associated with the "normal" component behaviour. The accommodation of multi-step transitions within the above model would likely require the use of subnormal and non-convex membership functions. Since, for the example plant, multi-step transition were rare ($< 1\%$ of the total transitions occurring in the 13 month period) the modeling performance of only simple membership functions will be analyzed. The extension of the modeling technique to the multi-step transition case was not explored.

In doing the optimization between the sampled membership functions and their analytical approximations, some care must be taken in determining the nature of membership functions whose region of support extends outside of the temperature sequence's range. Specifically this will occur for the extreme left and right membership functions of the behavioural map. For these membership functions, no information is available regarding their left and right boundaries, respectively. This being said, it may be possible to generate rough estimates of the unknown boundary regions through the analysis of the other analytical membership functions generated from the raw behavioural map. It is also important to note that, in general, the membership functions resulting from the optimization will tend to be asymmetrical. The nature of the boundaries relates to the statistical nature of their respective transition areas and in general there are no grounds to assume that two neighbouring transition areas should have similar statistics; hence, asymmetrical membership functions should be permitted and generally will occur.

The following section will describe how the probability of false alarm can be computed from the analytical membership functions and the resulting analytical representation of the raw behavioural map.

4.2.3 Analytical Membership Functions

It is important to note, that the membership functions discussed so far are sampled functions, as opposed to analytical functions, and as such they are not particularly amenable to the optimization of the upper and lower fault detection thresholds. These sampled membership functions may also be affected by abnormal behavioural patterns in the data. This would tend to limit the generation of the membership functions only to areas of known "good" behaviour, if the resulting functions are to be used in the bound setting process. Both of these limitations can be mitigated through the use of analytical membership functions which closely model the transition areas. Additionally, the above behavioural model utilizes what are in effect arbitrary membership functions. In general, these membership functions will not be simple mathematical functions. Instead, they will tend to be quite complicated functions which could complicate the bound generation process by increasing its computational complexity. Because of the raw membership functions' arbitrary nature, significant amounts of status data would need to be analyzed if accurate models of the membership functions were to be generated. If, on the other hand, the raw membership functions can be modeled using analytical functions then the amount of data needed to be analyzed to generate the behavioural model can be reduced. The analytical function used to model the raw membership functions could also be chosen such that its mathematical properties simplify the bound generation process with respect to the number of computations involved in the determination the linear bounding functions.

In particular, if a parameterized function $f_k(x)$, with support in the interval $\left[x_k^u, x_k^l\right]$ [2], is chosen as the analytical membership function then it can be "fitted" to the k^{th} raw membership function through the use of a gradient descent optimization procedure such that

$$J_k = \frac{1}{2}\sum_{j \in Z} (f_k(T_j) - g_k(T_j))^2 \tag{4.7}$$

---

2. As will be seen latter in this Chapter, the actual area of support for a given analytical membership function depends on the function's specific characteristics. Hence, the notation $\left[x_k^u, x_k^l\right]$ is used here to donate the generalized area of support for a general analytical membership function.

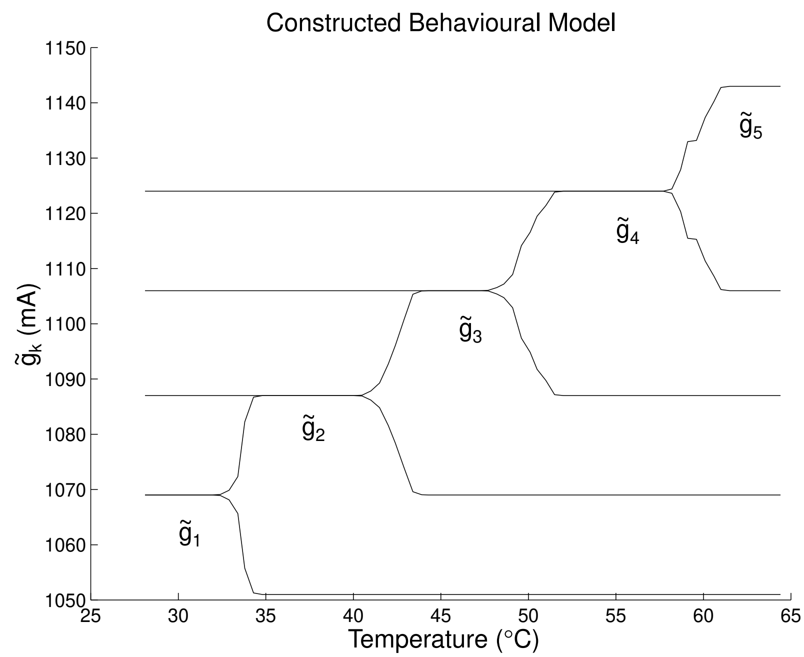


FIGURE 4-4.
Constructed behavioural model of the current-temperature behavioural map of Figure 4-1.

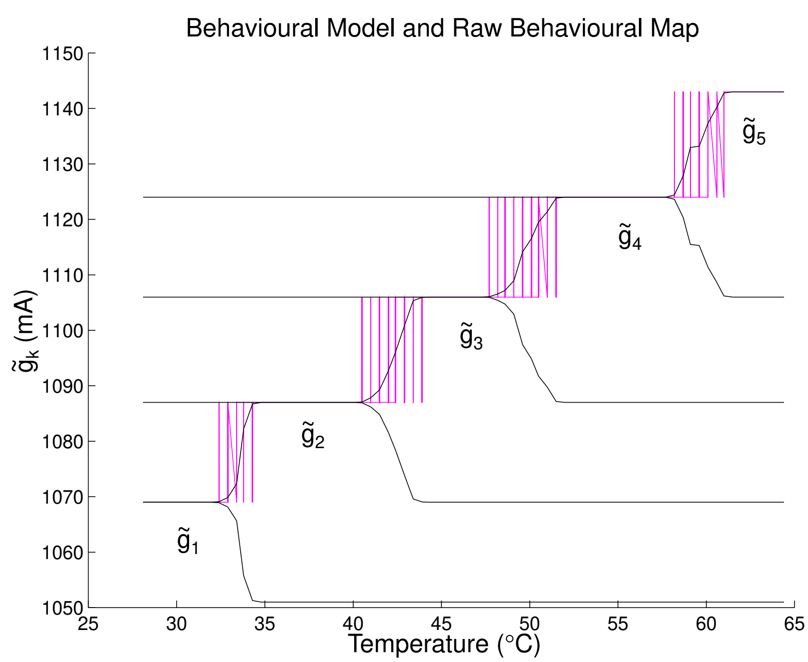


FIGURE 4-5.
Relationship between the behavioural model and the raw behavioural map.

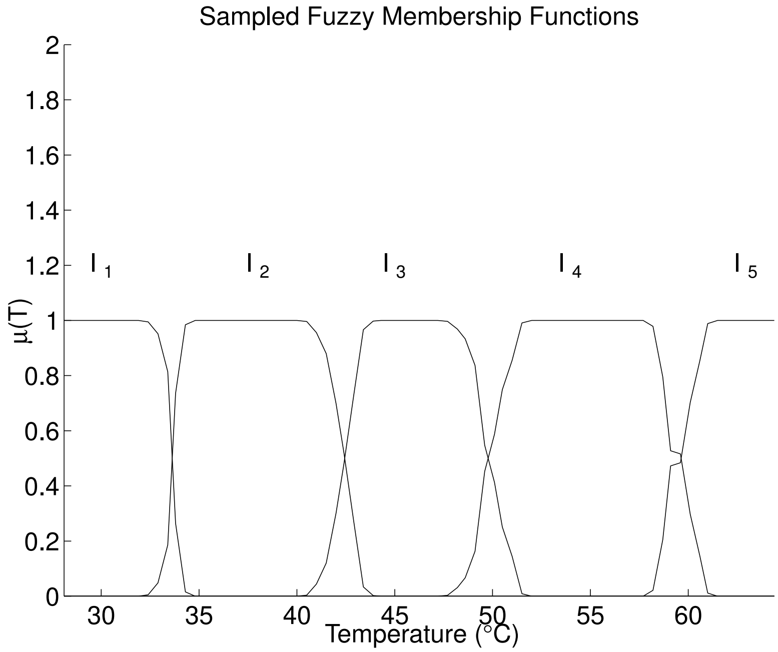


FIGURE 4-3.
Raw membership functions produced from the example linear dependency map of Figure 4-1.

ing the overlay of the two graphs. Quite clearly, this new representation provides a better indication of the statistical effects of the underlying noise process. In particular, the function describing the expected current across the given temperature range can be easily discerned from the behavioural model as the line, starting at current level I_1, which progresses to the left and up to each subsequent current level.

The work which follows will be framed within the context of the behavioural model as this provides a good approach of bridging the gap between the basic linear fault detection approach presented in Section 4.1 and the improved approach to be developed subsequently. It should be noted, though, that all the subsequent mathematical formulations are based directly on the membership functions themselves. Hence, the behavioural model is required only to aid understanding and is not itself a corner stone or requirement of the proposed approach.

mates of the probability functions describing the probability of occurrence of each current level at each of the individual temperature levels which occur within the particular status data sequence of interest. In essence, these estimates form conditional probability density functions $p(I_k|T_j)$ at each specific, quantized temperature step, T_j, which occurs within the given status data sequence. A function $g_k(T_j)$ that describes the probability of the given current level, I_k, occurring over the full range of temperatures can then be constructed such that

$$g_k(T_j) = p(I_k|T_j) \ , \quad j \in Z \tag{4.5}$$

One of the features of this function is that it describes the transition effects caused by the underlying noise. In addition, this function takes the form of a sampled fuzzy membership function with an initial boundary area describing the first transition to the particular current quantization level, followed by a flat area of value 1, indicative of the constant current level over a range of temperatures, which in turn is followed by the second boundary area describing the transition from the given current level up to the next current level. Figure 4-3 shows the membership functions generated from the raw enclosure temperature and current draw signals and produced utilizing the above technique for the example linear dependency map of Figure 4-1. Throughout the remainder of this work, membership functions generated from the raw status data signals will be referred to as “raw membership functions”.

For ease of understanding, a conceptual relationship, termed the “behavioural model”, between the membership functions of Figure 4-3 and the raw behavioural map of Figure 4-1 can be constructed. In particular, if the raw membership functions are offset by

$$\tilde{g}_k(T_j) = (I_{k+1} - I_k)\, g_k(T_j) + I_k \tag{4.6}$$

then a behavioural model, quite similar to that of Figure 4-1, can be produced, which represents the transition areas in terms of their statistics (Figure 4-4). Figure 4-5 highlights the relationship between the raw behavioural map and this behavioural model by present-

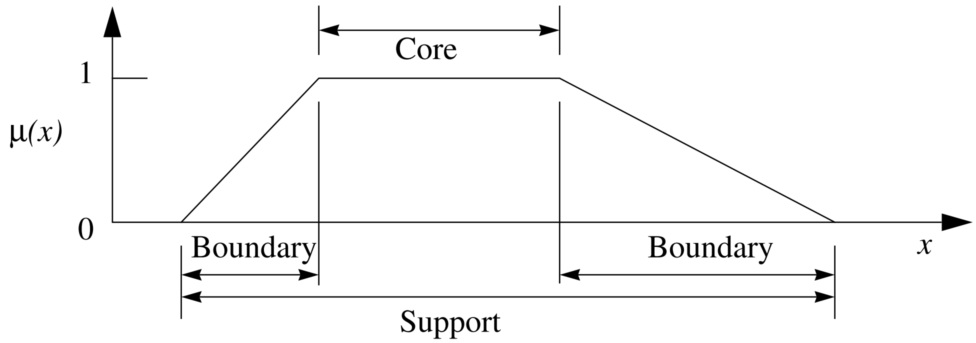


FIGURE 4-2.
Example fuzzy membership function.

Figure 4-2 shows an example fuzzy membership function and some of the related terminology utilized in fuzzy logic theory[1]. The core of the fuzzy membership function is defined as the region with the function over which $\mu_{\tilde{A}}(x) = 1$. The boundaries of the membership function are those areas for which x has some partial membership in $\tilde{A}$ (i.e. $0 < \mu_{\tilde{A}}(x) < 1$). The support of the membership function is the region over all $x \in \tilde{A}$ for which $\mu_{\tilde{A}}(x) > 0$. Fuzzy membership functions can also be described as being either normal or subnormal, indicating whether or not there exists at least one x for which $\mu_{\tilde{A}}(x) = 1$. In addition, the terms convex and non-convex are used within fuzzy theory to differentiate between membership functions that contain a single peak and those that contain multiple peaks within their given area of support.

4.2.2 Re-formulation of Dependent-Independent Behavioural Maps

The first step towards utilizing the concept of fuzzy membership functions in the modeling of the linear dependency maps is to covert these maps to a form which reflects the statistical effects of the underlying noise. In the case of the example engineering plant's linear dependency, this task can be done by utilizing the raw data map to generate esti-

1. The notation and terminology used for fuzzy membership functions within this work is taken from [73].

by this technique. In the example plant, temperature glitches generally relate to temporally transient events, something which this approach is not well suited to modeling. Additionally, the technique is based on modeling the “normal” behaviour of the linear dependency. Abnormal behaviour, which causes the assumptions regarding the linear dependency of the system to fail, obviously cannot be modeled; nor can the behaviour be automatically detected unless a test for the appropriateness of the technique to given data sets exists.

4.2 Statistical Modeling through Scaled Fuzzy Membership Functions

The following section will describe how the dependency map can be re-expressed, through the use of scaled fuzzy membership functions, to take into account the statistics of the underlying noise effects. By doing so, a majority of the limitations of the basic modeling approach, presented above, can be overcome. In particular, methods for computing the optimal or near optimal upper and lower thresholds, for a prescribed false alarm rate, will be determined, as well as a method for determining the applicability of the approach to specific prescribed data sets.

4.2.1 Fuzzy Membership Function Definition

Within the field of fuzzy logic, fuzzy membership functions, denoted $\mu_{\tilde{A}}(x)$, are used to quantify the degree of membership that a given element, *x,* has within a given fuzzy set, $\tilde{A}$, defined over a particular universe of discourse. In general, these membership functions are constructed such that the degree of membership is quantified over the interval $[0, 1]$. If $\mu_{\tilde{A}}(x) = 0$ then the element x is not a member of the set $\tilde{A}$. If $\mu_{\tilde{A}}(x) = 1$ then the element x is a full member of the set $\tilde{A}$. While, if $0 < \mu_{\tilde{A}}(x) < 1$ then x has some degree or partial membership in the set $\tilde{A}$.

When α^u and α^l are set to zero, this technique reduces to limit checking fault detection since both the upper and lower bounds become constant functions. In particular it becomes limit checking fault detection with the upper and lower bounds set to β^u and β^l, respectively.

4.1.2 Limitations

One limitation of the above approach is that it provides no means of determining what the optimal setting of the value of κ should be for a particular data set under a prescribed false alarm rate; it is left for the user to determine what an appropriate value is through a trial and error process. Ideally, the fault detection system should be able to model the system under study with a minimum of user intervention. The optimal or near optimal value of κ may vary widely between plant components; hence, it may be a significant burden to leave the selection of κ to the user. Additionally, it is far easier to achieve consistency in fault detection if the value of κ is set automatically according to a prescribed algorithm.

A second limitation is that the points a, b, c, and d are selected without regards the statistical nature of the transition regions. For each of the two transition regions, the extreme upper left and lower right points are selected regardless of how often the given point occurs within the current-temperature map. A single instance of a particular current-temperature pair could cause a significant variation in which upper or lower bounding function was generated, and, hence cause significantly different fault detection results. Additionally, there is no prescribed method within the approach to determine which two transition areas should be selected as the two from which the upper and lower bounding functions should be generated. Once, again the choice in which transition areas to utilize could greatly affect the fault detection performance of the system. Nor, is there a clear method of determining the applicability of the above approach. If the given dependent-independent signal map does not follow the assumed staircase function, then there is no clear way of determining this in an automated sense.

Finally, because this technique utilizes bounds based on the raw independent signal, there is an inherent sensitivity to noise on this signal. Glitches in the independent signal, if they occur frequently, could account for a significant number of the false alarms produced

where

$$\alpha^u = \frac{I_b - I_a}{T_b - T_a} \text{ and } \beta^u = \frac{I_a T_b - I_b T_a}{T_b - T_a} \tag{4.2}$$

and referring to Figure 4-1, $(I_k, T_k) ;k = a, b, c, d$ denote the current-temperature coordinates of the points indicated in the figure. The lower bound can be constructed in a similar manner from points *c* and *d*. These two functions define an upper line, passing through point *a* and *b*, and a lower line, passing through points *c* and *d,* shown as the dashed lines in Figure 4-1. As long as the temperature-current behaviour of the amplifier remains in the area between these lines, the amplifier is deemed to be operating normally. Outside of this area the amplifier is deemed to be in a fault condition.

This form of the bounding functions implicitly assumes that each of the transition widths are identical, and that the current and temperature signals are noise free. In practice, neither of these assumptions hold strictly true; therefore, a simple approach to deal with these problems is to modify the equations for the y-intercepts to include a scale factor κ as follows:

$$\beta^u = (1 + \kappa)\,\beta^u \tag{4.3}$$

and

$$\beta^l = (1 - \kappa)\,\beta^l \tag{4.4}$$

This scale factor has the effect of moving the two bounds outwards, increasing the area within which the amplifier is deemed to be operating normally. In general, the value of κ must be found through an iterative, trial and error process whereby κ is gradually increased until the false alarm rate is suitably small. Care must be taken though not to make κ too large since increasing κ also causes a decrease in the fault detection sensitivity of the approach. In general, therefore, the desire for a low false alarm probability must be balanced against the desire for fault detection sensitivity.

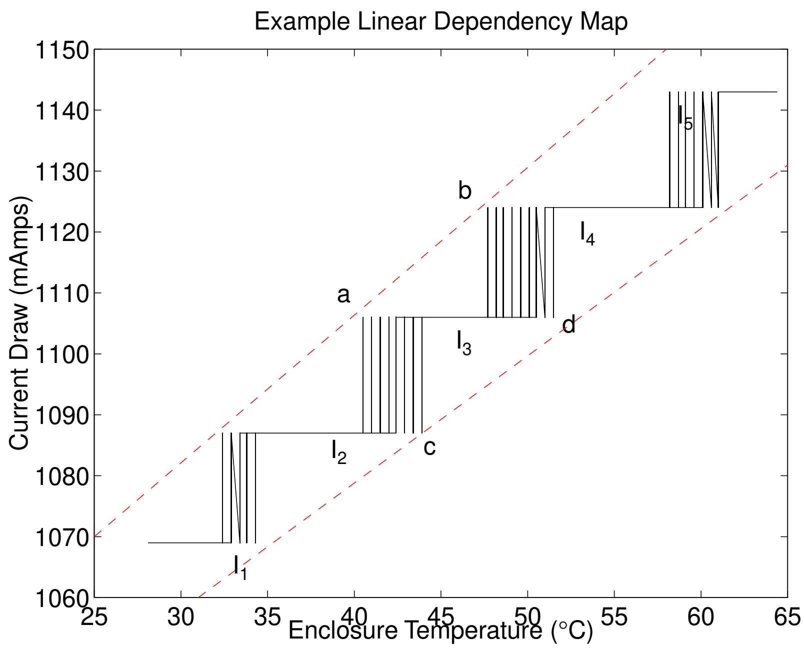


FIGURE 4-1.
General current draw signal versus temperature map for an amplifier within the example plant.

purposes of fault detection these bounding functions must also satisfy the constraints that they are easily computable from a limited number of observations, and that they produce tight bounds on $\mathcal{M}(I, T) = 0$.

A straight forward way of computing these bounding functions can be given by selecting four points on the map which correspond to two transition areas, as indicated in Figure 4-1. Specifically these points represent the upper left and lower right extremes of the two selected, transition areas. Using these points, the upper bounding function can therefore be given by:

$$f^u(T) = \alpha^u T + \beta^u \tag{4.1}$$

4.1 Basic Modeling Approach

In the following two sections a basic modeling approach for signals with unknown, linear dependencies will be introduced along with its limitations. This technique is based on the work presented in [66] and [67]. Within the following sections, the discussions will be framed within the context of the particular linear dependency relationship which is believed to exist within the example plant, namely the current draw signal's linear dependency on the temperature signal. This dependency is believed to be linear, or primarily linear, due to the correspondence between the current-temperature maps produced from the raw status data signals, and the theoretical maps produced by two signals which are assumed to be linearly dependent and affected by coarse quantization and underlying noise processes. Hence, the principal dependency between the current and temperature signals can be explained adequately by assuming that there is a linear relationship between these two status data signals.

The approach of presenting this detection technique in the context of a particular status data dependency has been taken for the sake of clarity. It is important to note though, that the technique presented is generally applicable to signals possessing linear dependencies and is not strictly limited in applicability to the cited example.

4.1.1 Overview

The sole information upon which fault detection can be performed for signals with unknown, linear dependencies is contained within the behavioral map generated from the given signals' relationship. This map quantifies the relationship between the dependent and independent status data signals. The general form of this map is that which was shown in Figure 3-4 and has been reproduced, for a specific sequence of current draw and temperature status data signals from the example plant, in Figure 4-1.

This map can be denoted as $\mathcal{M}(I, T) = 0$, where I is the measured current and T is the enclosure temperature. The modeling process can then be defined as the process of generating two bounding functions $f^l(T)$ and $f^u(T)$ such that, for any temperature T and for all values of the current $\tilde{I}$ which satisfy $\mathcal{M}(\tilde{I}, T) = 0$ then $f^l(T) < \tilde{I} < f^u(T)$. For the

Chapter 4:

Fuzzy Membership Function Based Fault Detection

4.0 Introduction

In this chapter, an extension to the traditional limit checking fault detection technique will be presented which is capable of modeling systems exhibiting linear dependencies despite the presence of coarse quantization and underlying noise effects. The chapter will begin by first introducing a basic modeling approach that can be utilized to model linear dependent systems. The limitations of this basic approach will then be discussed. An extension, through the use of membership functions similar to those used within the fuzzy logic field, to the basic approach will then be proposed which allows for the generation of statistical models of the linear dependency map. A technique will be developed which utilize these statistical maps in the generation of optimal or near optimal upper and lower fault detection bounds, in the sense of meeting a prescribed probability of false alarm over the region of “normal” operation. A technique for directly determining the appropriateness of this modeling approach to specific status data sets will then be presented. The approach that will be taken in describing the modeling technique and its properties will be to first introduce the concepts in terms of a general membership function and then to develop two specific analytical membership functions and compare their capabilities. The chapter will conclude by presenting a method by which the fuzzy-membership based modeling technique can be developed incrementally such that small amounts of data can be utilized to develop the initial model and subsequent data can then be used to refine the model.

The results obtained by applying this approach to the 13 months of status data collected from the example plant will be left until Chapter 7. In this way, the results can be compared more easily to the fault detection results obtained through the limit checking detection method which was in place within the example plant during this period, the fault detection results of which will also be presented in Chapter 7.

fault detection can be performed for the latter two classes, starting in Chapter 4, with the class of signals which have linear dependencies with one or more of the other status data fields.

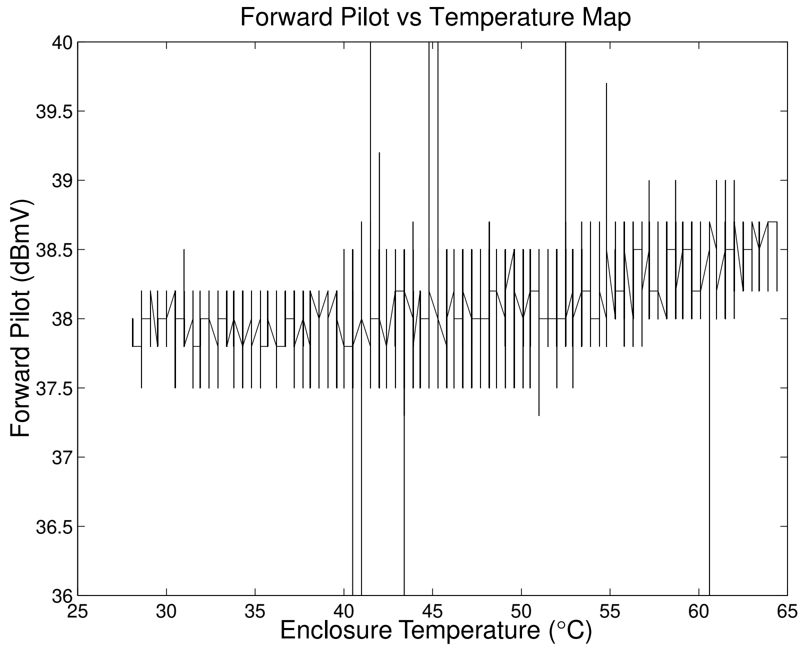


FIGURE 3-7.
Typical forward pilot versus enclosure temperature map.

It is important to note that, for this class of dependencies, the focus of this work is not on the training of the neural network, as this has been done in [35], but on the development of a theoretical frame work which allows for the fault detection threshold to be set appropriately.

3.3 Conclusions

In this chapter, the worst-case nature of status data signals generated from the limit checking fault detection systems traditionally utilized for large scale engineering plants was discussed. Arising from this discussion, was a classification of the status data signals into three distinct groups: those that were independent or “nearly” independent, those that were linearly dependent, and those that were non-linearly, dynamically dependent. For each of these classes, an example signal from the example large scale engineering plant was used to illustrate the class’s properties and fault detection techniques suitable for each class were introduced. The next three chapters will provide a more detailed look at how

for identifying actual fault conditions (as will be seen in Chapter 6). For this reason, this work will only be concerned with modeling the primary dependency for this class of signals.

Obviously, this class of signals is going to be poorly modeled by both limit checking and the linear dependency method. In both cases a large number of false alarms will be generated due to the failure of the modeling techniques to take into account the true nature of the dependency. Recurrent neural networks though offer an attractive modeling approach. As was mentioned in Chapter 1 these networks are capable of modeling arbitrary non-linear, dynamic relationships to an arbitrary degree of accuracy, provided that a training set can be constructed with enough input/output samples and enough neurons are employed. One of the significant challenges of using a neural network to perform the modeling is how to go about setting the fault detection threshold. The neural network, in this scenario, essentially acts as a step ahead predictor. A fault, therefore, is detected by comparing the output of the neural network with the signal it is modeling. If the difference is large enough then the component is deemed to have undergone a behavioural change. The difficulty arises in providing a theoretical framework to quantify, given the input and output signals, what magnitude of a difference signal signifies a significant behavioural change in the component, as opposed to a noise effect. Chapter 5 will lay the groundwork for this theoretical framework by developing a general noise estimation technique for the status data signals. Chapter 6 will build on the work of Chapter 5 to provide a methodology for setting the fault detection thresholds for the neural network based detection system in a prescribed manner.

In the example plant, the forward pilot signal's variations can be seen to be correlated with those of the temperature signal (Figure 3-1). If a map similar to the one in Section 3.2.2 is produced, then it can clearly be seen that the behaviour of the forward pilot signal does not have a simple linear relationship to the temperature (Figure 3-7). In fact, it is known that the example plant's amplifiers contain both temperature compensation and automatic gain control circuitry [11][12][13]. Hence, it can be reasonably assumed that the forward pilot signal is sensitive to temperature variations and the measured forward pilot status data signal will have a fairly complex temperature dependency.

cases tend to vary considerably. This necessitates the need to model each of the amplifiers current behaviour independently, and does not permit the grouping of amplifiers into supersets, based on behaviour, and “training” one model per set.

In some cases, a signal may have a linear dependency on more than one of the other status signals. This does not pose a problem, even though the preceding discussion implicitly assumed a linear dependency on only one other status signal. Due to the linear nature of the relationship any linear dependency on multiple status data signals can be expressed as a single linear dependency on the summation signal, formed by summing the individual independent signals. Hence, a system which models single linear dependency is also capable of modeling multiple concurrent linear dependencies.

3.2.3 Signals with Unknown Non-linear, Dynamic Dependencies

The most general class of signal dependencies, are those signals which have non-linear, dynamic dependencies on one or more of the other status data fields. Once again due to the effects of the underlying noise, both the exact nature of the non-linearities and the dynamics will be unknown. Mathematically, a sensor signal with this type of dependency can be represented by

$$\dot{x}_1(t) = f(x_2, \ldots, x_M) \tag{3.5}$$

where f is an unknown non-linear function. This class of signals represents the full spectrum of dependencies which may exist past those that are linear. Obviously, there exist some potential dependencies that straddle the boundary between these two classes. As was mentioned previously, the nature of this boundary and modeling of signal dependencies within its neighborhood is considered outside the scope of this work. What is of concern within in this work is the modeling of signal dependencies which clearly fall within either of these two classes. More particularly, for this class of signals, this work is only concerned with the modeling of the primary dependency. Because of the non-linear nature of the relationship of Eq. 3.5, the individual effects of each of the independent signals cannot be easily accounted for. This greatly complicates the development of a formal framework

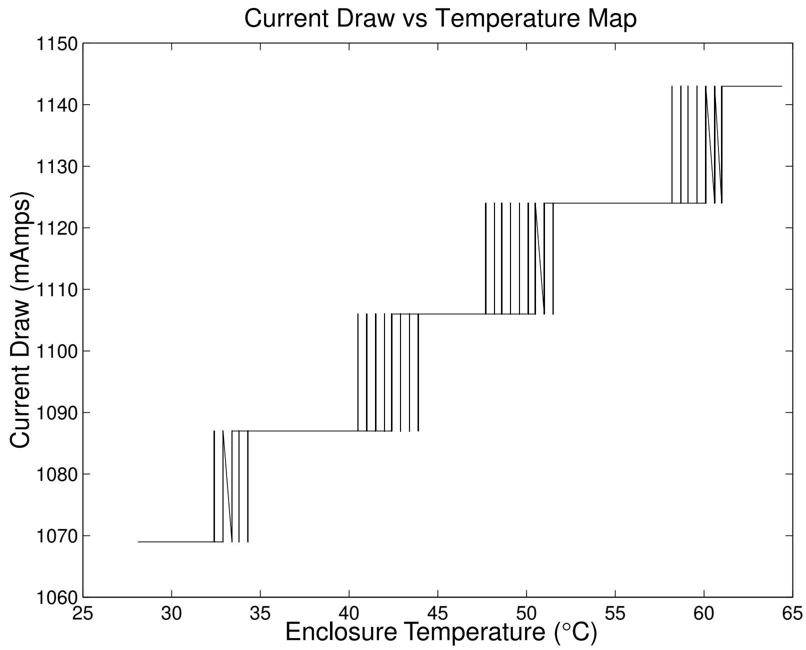


FIGURE 3-6.
Typical current draw signal versus temperature signal map.

Within the example cable plant, the current draw signal is most noticeable in exhibiting a linear dependency relationship. In Figure 3-1, it can be quite clearly seen that the current signal variations occur in step with the variations in the temperature signal. In fact, if a plot of the current level is made against each of the temperature levels that are seen in Figure 3-1, then the map shown in Figure 3-6 is produced. This map is consistent with the type of map to be expected if the current draw signal was linearly dependent on the temperature. Detailed analysis of the current signal within the cable amplifier plants has shown that this linear relationship occurs consistently for almost all of the amplifiers within these plants [66]. Those amplifiers that do not follow this relationship typically behave abnormally, with respect to their current signal, and hence a modeling approach which permits the direct evaluation of its own applicability to model a given component is useful, in this context, for identifying amplifiers with outlying behaviours. The analysis provided in [66] also showed that although most of the amplifiers have similar general staircase relationships with their associated temperature signal the "slopes" of the stair-

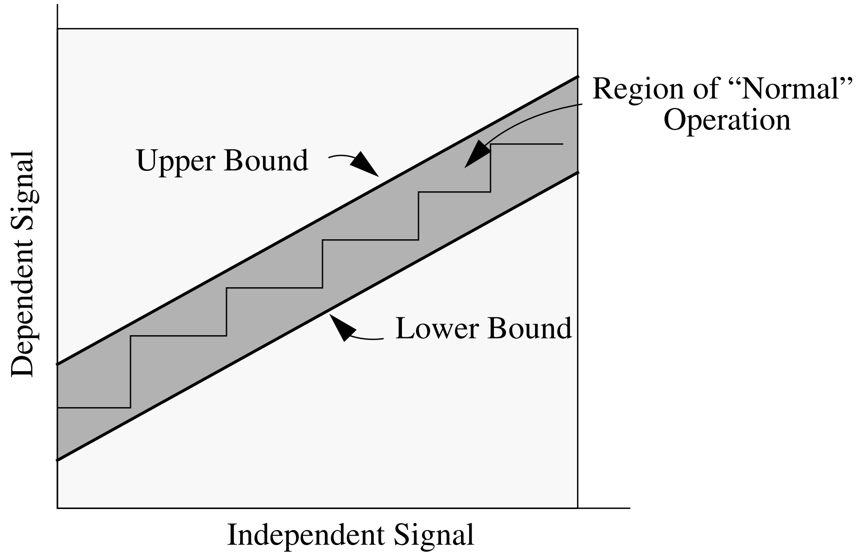


FIGURE 3-5.
Ideal bound location for signals with unknown, linear dependencies.

should not be unduly influenced by statistically rare events. Additionally, some analytical means of detecting the validity of this modelling approach should be apparent. If the staircase assumption is invalid, for a particular component which was assumed to have a linear dependency, then this modeling problem should be easily identifiable. Failures to model the component's behaviour may in itself be indicative of the presences of a fault condition within the component, provided of course that the assumption of linearity was in fact valid. A fuzzy membership function based method which meets these goals will be presented in Chapter 4. Additionally, Chapter 4 will address methods of incrementally building up the complete models in such a way that "reasonable" models can be generated using only relatively small amounts of data. In this way, the fuzzy membership technique will be able to "learn" the signal's appropriate behavioural model over time. Neural network techniques are not suitable for modelling this type of dependency due to its relatively slow varying nature. Neural network techniques are generally better suited to modeling signal with a more continuous variation. Large areas of constant or "near" constant behaviour tend to cause problems within the neural network's training phase.

Basically, the effect of the quantization is to transform the linear function into a "non-linear staircase". It is important to note that the staircase function relates to the effect of quantizing both the dependent and independent variables, typically with a different quantization function used for each. In reality, both signals exist as status data signals and hence are quantized within their respective status data transponders. The step function seen in their relationship, therefore, is related to the two quantizers used, but it is not a direct image of either one. With just the quantization effects accounted for, the linear relationship can still be directly obtained from the staircase function. Once the effects of the underlying noise are accounted for, then this is no longer the case. The underlying noise obscures where the exact transitions between adjacent quantization levels occur. Once underlying noise is introduced, many different linear functions can result in the same noisy, staircase. The true linear relationship between the dependent and independent status data signals is therefore unknown and must be estimated.

Obviously, if limit checking fault detection was to be used for this class of signals, then quite likely a significant number of false alarms would be produced (assuming that the upper and lower bounds are set reasonably tightly). The upper and/or lower bounds would be crossed due to the known linear dependency of the signal. The "fault" flags produced by these crossings would not be related to changes in the behaviour of the component and, therefore, would not indicate the presence of actual faults. In fact, they would tend to complicate the fault detection process by obscuring the occurrence of actual fault related flags in a sea of false alarms. Ideally, a detection system for this class of signals would, therefore, consist of upper and lower bounds which are slightly offset variations of the linear function relating the dependent and independent sensor signals. These bounds would therefore form a diagonal band of "normal" operation within the independent/dependent signal map (Figure 3-5).

Equally as obvious, is that the existence of the underlying noise effects makes the estimation of these upper and lower bound functions non-trivial. This is particularly true if the bounds are to be generated in such a way as to take into account the statistics of the dependency map. Each feature of the map should only influence the location of the upper and lower bounds in accordance with the statistical likelihood of its occurrence. The bounds

use of limit checking fault detection will not result in any increased fault detection latency over other fault detection techniques, provided of course that the upper and lower bounds are placed close to the “normal” signal level.

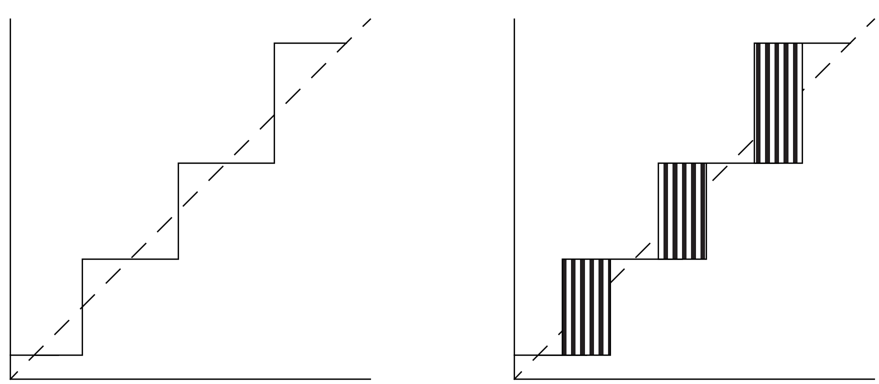
3.2.2 Signals with Unknown Linear Dependencies

The next level of complexity occurs when the status data signal of interest is linearly dependent on one or more of the other status data signals. Mathematically, this type of dependency, for a particular sensor signal, can be expressed by

$$x(t) = a_1 x_1(t) + \ldots + a_M x_M(t) + a_{m+1} \tag{3.4}$$

where $a_1, \ldots, a_{M+1}$ are constants.

In general, due to the effects of using low order quantizers within the data transponders, the exact nature of this linear relationship is unknown. This relationship is further obscured by the presence of underlying noise on the sensor signals. Eq. 3.4 provides the basic relationship for a linearly dependent sensor signal. For fault detection systems though, the only available signals are the status data signals obtained after sampling and quantization processes have been performed. Figure 3-4 illustrate the effect that the quantization process and underlying noise has on this linear relationship.



a) Quantized Linear Dependency b) Quantized Noisy Linear Dependency

FIGURE 3-4.
Staircase mapping of a quantized linearly dependent signal.

Since the rationale behind monitoring large scale engineering plants is to obtain an accurate indication of the plant's behaviour, it is highly unlikely that major faults within a plant will result in only one sensor's readings being affected. The plant's sensors generally tend to overlap to some extend in their observations of the plant's behaviour. Hence, it is unlikely that a driving force, which causes significant variations in one sensor signal, will not be seen at least to some extent, on some of the other sensor signals. A corollary to this argument is that a sensor signal which is independent of all the other sensor signals is not likely to contain significant levels of variation. Instead the signal is much more likely to be nearly constant.

An example of such a signal can be found in the B+ voltage signals measured in the example cable amplifier plant. Figure 3-1 clearly shows that the B+ voltage signal is nearly constant, with only slight variations due to effects caused by changes in temperature. These effects though are very slight, causing only a one quantization step change in the data across a full two month time period. Within the given plant, this signal relates to the output voltage of the regulated power supply housed within the cable amplifier. In general, regulated power supplies are designed to provide a nearly constant output power level despite variations in their input voltage and output loads. Hence, it is to be expected that the related status signal's variations will be relatively independent of the other status data signal over periods of "normal" operation and that variations within these periods will be relatively small. Both of these expectations are evident in the example B+ signal.

Because the independent status data signals are typically nearly constant, they are ideal candidates for limit checking fault detection. The required upper and lower bounds can be easily set and do not, in general, need to be varied. In particular, the bounds can be easily centred one or two quantization steps above/below the "normal" signal level, depending on the desired fault detection sensitivity. The use of limit checking fault detection systems for this class of signals results in both reliable and confident fault detection since any variations in the behaviour of the signal are necessarily due to variations in the component's behaviour and not due to unrelated factors influencing the status data signal. Additionally, since there is no actual "trend" information with these types of signals, the

be paid to the worst case status data signal scenarios as described previously. Additionally, as each of the classes is presented, an example status data signal from the example large scale engineering plant will be given to illustrate the class' characteristics.

It should also be noted that, in general, if two signals are statistically correlated, then it may be equally valid to claim that either one is the independent signal. For the purposes of this work, though, the independent signal will be taken to be the one which, taking into account the nature of the plant, is most likely to be the driving signal. For example, in plants utilizing high frequency components, it is more likely that temperature variation will cause variation in the output signal levels than *vice versa*. Hence, the status data signal from a sensor measuring the output of the high frequency component would be viewed as having a dependency on the temperature status data. The temperature status data signal would not be viewed as being dependent on the output status data signal. Additionally, as it is a natural progression to first account for the main source of variations and then to account for the lesser sources, this work focuses on modeling the main signal dependencies, in the sense, of modeling the dependencies which account for the majority of the given signal's variations.

3.2.1 Independent or "Nearly" Independent Signals

The first and simplest class of dependencies, are those status data signals which are independent or "nearly" independent of all the other monitored status data signals. Some slight dependencies may exist, but these will result in only slight variations in the signal which will typically occur relatively seldom. The variations in this type of signals are typically due to effects which are not measured either directly or indirectly by any of the other collected signals. Typically, due to the nature of large scale engineering plants, this type of signals are generally nearly constant, with any variation being almost solely due to variations within the sensor itself, local variations within the component, which only affect the one particular sensor, or some minor influence of one or more of the other sensor signals. Mathematically, these sensor signals can be expressed by

$$x(t) \approx C \tag{3.3}$$

that early fault detection can be accomplished. It is important to note that a signal, although correlated with another, may not necessarily be the driving force behind the other signal's variations. Instead both signals may be influenced by an underlying process which is not directly measurable from the collected status data signals. In such a situation, the variations in the underlying process may cause the two sensor signals to be statistically correlated though not directly related.

Hence within this work, the term dependency is used to denote, in a broad sense, the existence of a relationship between two or more status data signals. It is important to note that within this context, the term dependency is not restricted to a direct relationship between the signals, but instead refers to the existence of a functional mapping from one status data signal space to another other. Mathematically, therefore, ignoring the digitization and underlying noise effects, a given monitored sensor signal $x_1(t)$ can be, in the most general form, expressed as having the following dependencies

$$\dot{x}_1(t) = f(x_2, \ldots, x_M, u_1, \ldots u_N) \tag{3.2}$$

where $\{x_2, \ldots, x_M\}$ is the set of the other monitored sensor signals upon which $x_1(t)$ depends, $\{u_1, \ldots, u_N\}$ comprise the set of unknown and un-monitored factors upon which $x_1(t)$ depends, and f is an unknown, non-linear function.

The next three sections will address the types of signal dependencies which typically occur within large scale engineering plants, namely independent or "nearly" independent signals, linearly dependent signals, and non-linearly, dynamically dependent signals. This group of 3 classes represents the full continuum of dependencies which may occur within a large scale engineering plant. Hence, a fault detection system built by combining fault detection techniques tailored to each class should be generally applicable to a wide range of large scale engineering plants. Within the following sections, particular attention will

Even if an exact estimate of the underlying noise is unattainable, some of its general properties can be deduced from the status data signals $y_q(kT)$. If a large enough status data sequence is observed, then it can be reasonably assumed that the overall statistics of the transition noise areas give a good impression of the underlying noise level. If very few transition areas are seen on the signal then it can be reasonably assumed that the underlying noise has a small amplitude in relation to the average quantization step size, as is the case with the temperature signal of Figure 3-1. If, on the other hand, there is a large number of wide, and possibly overlapping, transition areas, then it is reasonable to assume that the underlying noise amplitude is of the same magnitude as the quantization step size. This is the case with the forward pilot signal of Figure 3-1. Finally, if single transitions across multiple quantization levels are relatively rare, as in the case of the example plant, then a reasonable upper bound on the underlying noise level can be assumed to be the average quantization step size for the given status data signal.

3.2 Classes of Signal Dependencies

Through the central status monitoring system, a group of status data signals are collected representing the behaviour of the plant over a given time period. Each of these individual status data signals may have some statistical correlation with one or more of the other collected signals. In the ideal case, the status data signals would have little or no correlation to the other signals, and would be nearly constant under "normal" operational conditions and change radically when behaviour changes in the given component occur. Typically though, this is not the case for status data in real systems. Due to the overlap between the sensor readings and because of sensor dependencies, which may or may not be both known and accountable, the status data signals in real systems tend to contain significant levels of correlated variation, even under "normal" operating conditions. These dependencies tend to be responsible for a significant number of the false alarms which occur in traditional limit checking fault detection systems.

Each of the individual status data signals typically have some statistical correlation with one or more of the other collected signals. It is through the analysis of the nature of these correlations, and more precisely through the identification of changes in their nature,

Figure 3-3 shows the quantization of a noise-free sensor signal compared with the quantization of a noisy sensor signal. From this figure it can be seen that the effects of the underlying noise are only apparent when the sensor signal is “close” to the transition level between two adjacent quantization steps (assuming that the noise amplitude is less than the average quantization step size). More precisely, the underlying noise causes the quantized sensor signal to fluctuate between the adjacent quantization steps within this transition area. For a given sensor signal sequence, an increase in the amplitude of the underlying noise will result in an increase in the width of the transition area. These areas of fluctuation between adjacent quantization levels on the monitored status signals, $y_q(kT)$, which are caused by the underlying noise, will be referred to as transition noise areas.

In general, the width of any particular transition area will be dependent on a number of factors including the magnitude of the underlying noise, the proximity of the underlying sensor signal to the given transition level, and the rate at which the underlying sensor signal passes through the neighbourhood of the given transition level. If the sensor signal passes steeply through the transition level, then even high noise amplitudes will result in little or no transition noise. On the other hand, if the sensor signal remains in close proximity to the transition level over a long period, then even low noise levels will result in wide transition noise areas. If the noise amplitude is greater than one quantization step, then transitions across multiple quantization levels will occur.

In general, these properties make direct estimation of the underlying noise difficult to obtain, particularly since the actual underlying noise-free sensor signal is itself an unknown. If an estimation of the underlying sensor signal can be obtained which is independent of any knowledge of the underlying noise level, then this estimation can be used as a basis for estimating the underlying noise level, provided of course that the quantization function is known. This is the approach that will be taken in Chapter 5 to obtain estimates of the underlying noise levels.

coarseness of the quantization processes. This loss greatly complicates the development of early fault detection systems, particularly for the class of signals with unknown non-linear, dynamic dependencies.

3.1.3 Underlying Noise Effects

Within this work, the sensor signals available at the input to the status data transponder's quantizer(s) are assumed to be noisy. In particular, within Chapter 2, it was assumed that this noise was due to a number of effects including thermal and sensor noise, and that in general the cumulative effects of these noise sources could be modeled as a single additive, Gaussian noise source, which will be referred to as the underlying noise throughout the remainder of this work. The purpose of this section is to discuss the effects that the existence of this additive noise has on the status data signals, $y_q(kT)$. Since quantization is a non-linear process and since low order quantization is not amenable to the utilization the additive white noise approximation, an analytical analysis of the underlying noise effects is not feasible. Instead a descriptive overview of the underlying noise effects will be presented highlighting the areas of the status data signals $y_q(kT)$ which are most likely to suffer from its effects.

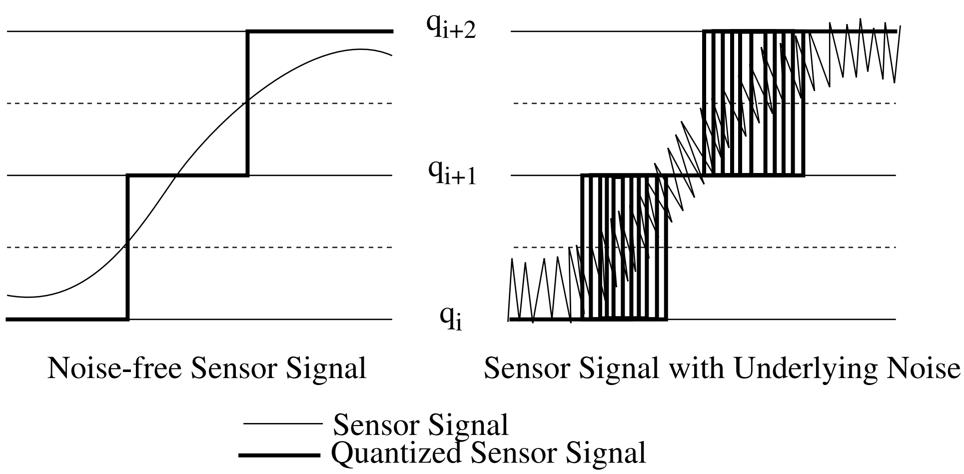


FIGURE 3-3.
Effects of the underlying sensor noise on the quantization process.

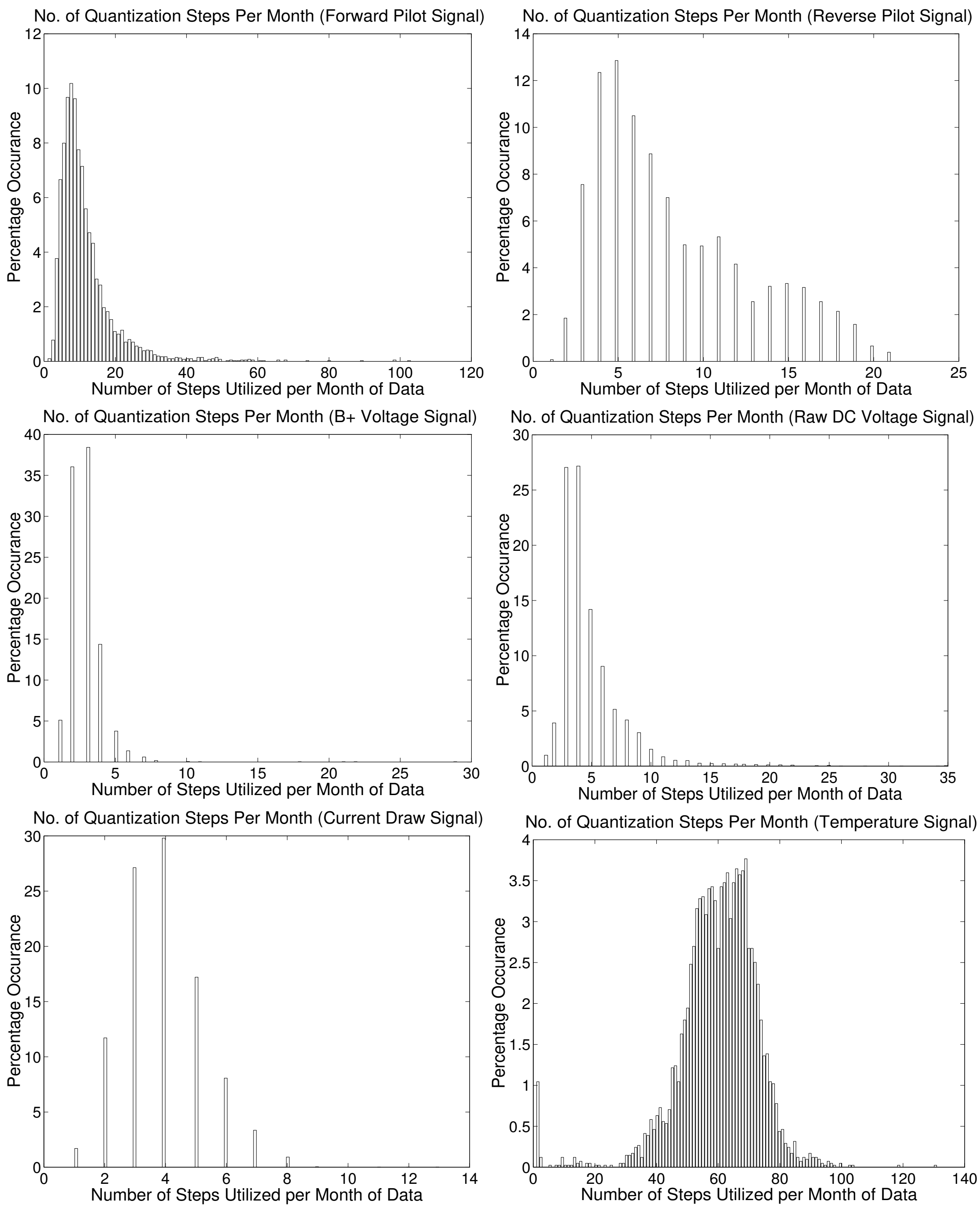


FIGURE 3-2.
Distribution of the number of quantization levels utilized per month by each of the six status data fields across each of the 354 monitored amplifiers (October 1995 to October 1996).

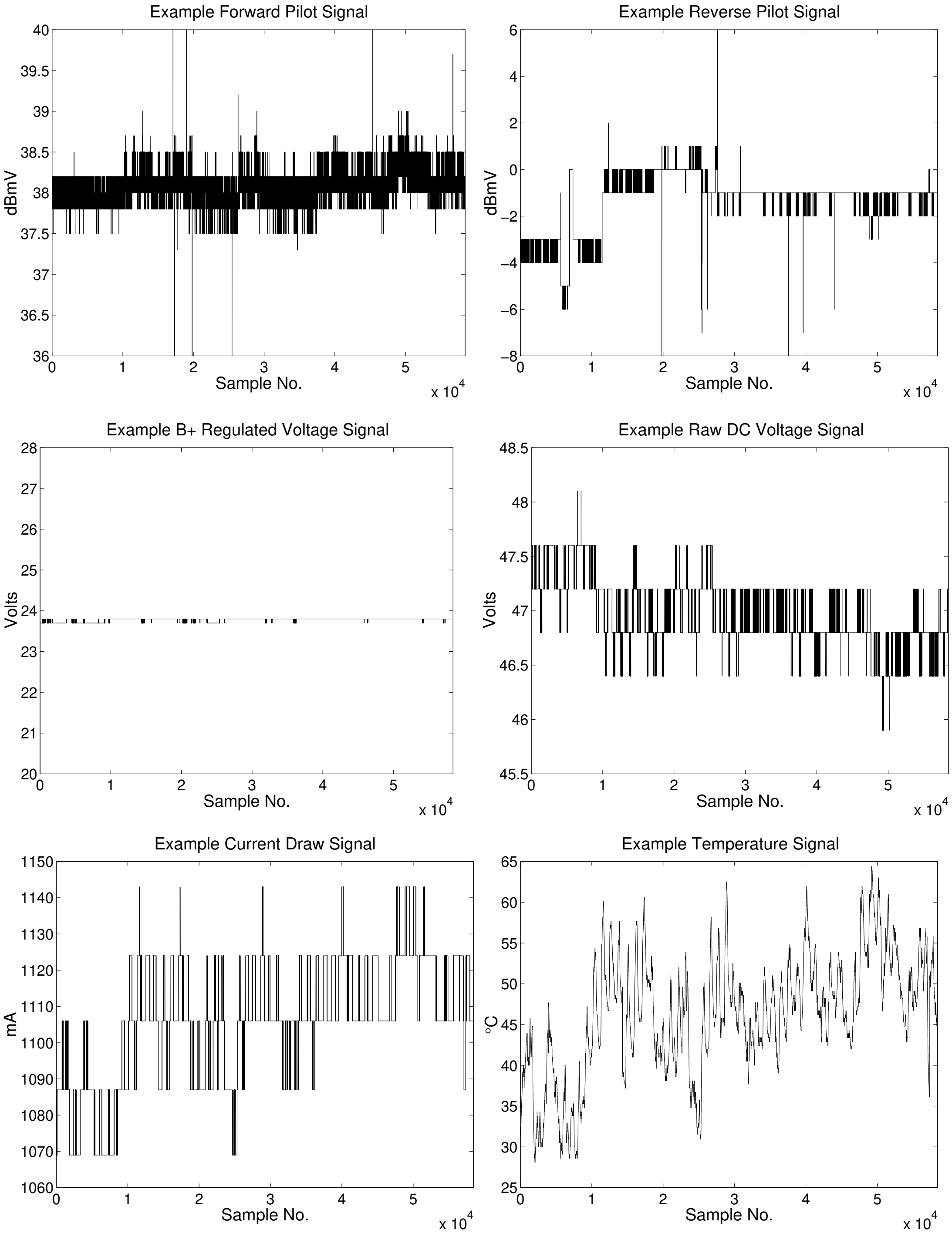


FIGURE 3-1.
Typical status data signals generated by a given cable trunk amplifier within the example large scale engineering plant during the time period of March 1, 1996 to April 30, 1996.

quantization process must be considered as a signal dependent noise source which cannot be easily isolated from the underlying sensor signal. To date, analytical solutions describing the noise effects of low order quantizers have only been developed for the cases involving up to two additive sinusoidal inputs [45]. An analytical solution for isolating the effects of low order quantizers on arbitrary input signals is an open area of research [45]. Because the effects of the quantization are not directly separated from the underlying sensor signal, techniques for the blind estimation of the underlying sensor signal had to be developed. These techniques, along with their utility, will be presented in Chapter 5.

Figure 3-1 shows the typically quantized status signals generated by the example large scale engineering plant. It is quite clear from this figure that, with the exception of the temperature signal, very coarse quantization steps are used within the plant. This conclusion is confirmed by Figure 3-2 which shows the distributions of the number of quantization levels utilized per month for each status data field over a 13 month period[2]. In nearly all cases, less than 64 quantization steps were employed for each status data fields for each of the amplifiers. In some cases, for the forward pilot signal, the number of utilized quantization steps exceeded the 64 step limit, but these cases relate to amplifiers experiencing abnormal behaviours. Even with the temperature signal, which is the least coarsely quantized signal, the 64 step limit is only exceeded in less than half of the cases.

It is important to note that the signal model of Eq. 3.1 results in quantization step sizes that are typically non-uniform, due to the absorption of the non-linear functions c_1 and c_2 into the sensor signal $\tilde{x}(t)$ (see Section 2.3). Though the use of non-uniform step sizes in a linear quantizer is unusual, it does not present any additional problems since the standard uniform white noise approximation does not hold. It is important to note that a significant portion of the trend information contained within the status data is lost due to the

---

2. Monthly data sets were chosen as the purpose of the figure is to show the “normal” number of quantization steps utilized by each field. Utilizing time periods of longer than one month would greatly increase the probability that the statistics would be artificially increased due to changes in the amplifiers’ behaviours.

consideration can be modeled as having a "nearly" constant sampling rate. This assumption is necessary in order to be able to apply standard signal processing techniques. As long as the sampling rate, over a large enough data sequence, varies only by a small amount then the error introduced by the non-constant sampling can be viewed as relatively insignificant. The example engineering plant presented in Chapter 2 can be assumed to have a "nearly" constant sampling rate (see Appendix B).

The discussion and subsequent conclusions presented within the following sub-sections assume that the status data is continuously monitored within the plant. In some plants, status data is only collected for prescribed periods following the occurrence of "significant" events. In these systems, the status data is used primarily to provide an audit trail for the event and, hence, early fault detection is generally not applicable. For this reason, plants of this type are considered outside the scope of this work.

3.1.2 Quantization Effects

In general, since it has been assumed that the plant's original status monitoring system was designed to utilize limit checking, it can also be assumed that relatively few quantization steps are used in the status data digitization process. In limit checking systems the behaviour of the data within the "normal" window is unimportant; the only important behaviour is whether or not the upper or lower bounds were crossed. Hence, coarse quantization levels are perfectly acceptable for these types of fault detection systems. In addition, the ability to utilize coarser quantization levels also allows lower resolution sensors and/or A/D converters to be used. This typically would result in lower overall implementation costs for the monitoring system.

Coarse quantization levels, though, cause significant problems when it comes to retrofitting advanced early fault detection systems to monitoring systems of existing large scale engineering plants. Typically the number of quantization steps utilized over the status data signals' dynamic range is far less than the approximately 64 steps required for the standard uniform white noise approximation to hold[1]. Therefore, the noise effects due to the

---

1. Quantization effects can be generally modeled as an additive, uniform white noise source to the signal of interest provided that certain necessary conditions are satisfied [44][47].

3.1 General Characteristics

In general, the status data signals generated by a status monitoring system within a large scale engineering plant can be considered to be a set of random variable sequences, the digitization of which is governed by the sampling rate of the system and the quantization process employed. Mathematically this process, for a given sensor, was expressed in Chapter 2 as

$$y_q(kT) = \tilde{Q}(\,[\tilde{x}(t) + \tilde{n}(t)]\,|_{t = kT}) \tag{3.1}$$

In the following sections, the worst-case scenarios for the sampling process $t = kT$, the quantization process $\tilde{Q}(t)$ will be discussed. Additionally, the effects of the underlying noise $\tilde{n}(t)$ on the status data signal $y_q(kT)$ will be outlined.

3.1.1 Sampling Rate

It is expected that the sampling rate for a given large scale engineering plant will be relatively low for two main reasons. First, it is assumed that the monitoring system has originally been designed to utilize limit checking; therefore, it can be assumed that signal reconstruction was not one of the principal design criteria for the data collection system. Limit checking operates on a point by point basis, hence there is no need to be able to reconstruct the sampled signals. The only requirement is that the sampling rate be sufficiently high to be able to detect “significant” changes in the status data within “reasonable” time periods. Hence, in general, a high sampling rate is not required. Secondly, since technical personnel are generally used to filter the “fault” flags in limit checking systems, sampling rates on the order of a few tens of seconds to a few minutes are typically used. The use of higher sampling rates would tend to overwhelm the technical personnel.

This being said, it is expected within this work that the status data is not undersampled, or equivalently that any necessary anti-aliasing filters are located within the sensor modules. If the data are undersampled then the trend information which is contained within the data is lost and early fault detection schemes, which generally detect faults by tracking these trends, cannot be applied. Additionally, it is assumed that the plant under

tems since changes in the nature of these dependencies are indicative of where and when behavioural changes within a given component have occurred. Obviously, the appropriateness and correctness of any modeling technique depend both on the characteristics of the digitized status data and on the type of dependencies that are present in the data sequences. Therefore, in the second part of this chapter, a classification of the type status data dependencies occurring within large scale engineering plants will be presented. This classification scheme will separate the status data sets into sequences that are independent or "nearly" independent of the other data sets, sequences that have an unknown linear dependency on one or more of the other data sets, and sequences that have an unknown non-linear, dynamic dependency on one of the other data sets. The class of status data signals which have unknown non-linear, dynamic dependencies with multiple data sequences will not be directly discussed as it is outside the scope of this work. The development of early fault detection techniques for this particular sub-class of signals is viewed as outside the scope of this work and an area for future investigation.

This work will focus on the fault detection of signals which clearly fall within the defined classes. Early fault detection techniques for signals which occur near the boundaries of the classes, particularly the boundary between linearly dependent and non-linearly, dynamically dependent, has been left as an area for future work. The main focus of this work is also on modeling the primary signal dependency as this dependency contains the largest portion of information regarding the signal's behaviour. Combining the modeling of the primary dependency with the modeling of less significant dependencies has been left as an area for future work. As each of the dependency classes is introduced, the appropriate fault detection technique, within the context of the worst-case assumptions for the status data, for the given class will also be discussed. Throughout the chapter status data signals from the example cable plant will be used to illustrate the signal characteristics and dependency classes being discussed.

Chapter 3:

Status Data Characteristics

3.0 Introduction

The previous chapter addressed the issues surrounding the general structure and properties of large scale engineering plants as well as the limit checking fault detection schemes generally utilized in their operation. This chapter focuses on the characteristic of the status data that are typically generated from these limit checking fault detection systems. In general, this data can be viewed as a set of random variable sequences, one sequence from each of the monitored sensors, which have certain properties due to the nature of their digitization. The chapter will begin by discussing these digitization properties and the effects they introduce into the status data signals. In particular, the worst case scenarios for the digitized status data sequences will be explored. The decision to focus on the worst-case status data scenarios has been made for two reasons. First, it can be generally assumed that the monitoring systems that generate the worst-case status signals are also the cheapest and easiest to employ. Despite their worst-case nature, these systems are still well suited to limit checking fault detection; hence, they likely occur in large numbers within industry. Secondly, a detection system which is designed to function adequately within these worst-case status data assumptions should inherently operate well under less onerous constraints. The goal of this work is to develop advanced fault detection techniques which can be retro-fitted to existing monitoring systems, hence the assumption that the status data is worst case allows a more general fault detection approach to be developed. Within this discussion, the effects of the underlying sensor noise on the status data's digitization process will also be explored.

The collected status data is the only information generally available about the operation of large scale engineering plants which is useful in the performance of real-time fault detection tasks. Typically, a given status data sequence generally will have some dependency on one or more of the other collected status data sequences. The accurate modeling of these dependencies is the keystone to the development of an early fault detection sys-

ter concluded by briefly introducing the example engineering plant which will be used to validate the proposed early fault detection techniques. This overview of the example plant was followed by a discussion on how the plants status data would be used to validate the proposed detection methods and the validity of this validation approach.

The next step in the development of the early fault detection techniques is to develop an understanding what are the typical characteristics of the status data generated from the status monitoring systems of such large scale engineering plants. This is the subject of the Chapter 3.

2.4.1 Applicability

In general, information about industrial fault detection systems for large scale engineering plants is viewed as proprietary information by the respective companies. Hence, it is quite difficult to obtain specific information regarding the fault detection processes utilized within a particular industry or within a particular type of plant. This being said, the basic design considerations and constraints for large scale engineering plant fault detection systems are similar. In each system, there are a large number of components from which various status signals are collected. Once collected the data then needs to be analyzed for fault conditions. Traditionally, due to the relatively low computational power available, limit checking has been used as the method of choice for the fault detection process.

If it is assumed that the engineering design choices made for cable televisions plants, with respect to fault detection, are similar to the design choices made in other industries, then it can also be assumed that the conclusions reached regarding the characteristics of the status data signals and the methodology of fault detection within this example large scale engineering plant are typical of industry's approach to fault detection in general. For this reason, throughout the remainder of this work the validity of the proposed early fault detection techniques will be confirmed by evaluating the fault detection performance of these techniques on real-world operational status data obtained from the example cable trunk amplifier plant. Because no assumptions will be made regarding the structure of the plant in question or behaviour of the status data, and because the engineering decisions which went into the design of the status monitoring system for the cable amplifier plant can be assumed to be typical of design decisions made for other large scale engineering plants, the early fault detection techniques presented should be applicable to large scale engineering plants in general.

2.5 Conclusions

This chapter presented an overview of what features and characteristics constitute large scale engineering plants as defined within this work. The basic structure of a limit checking fault detection system was presented along with a signal model for the status data as it is seen by the limit checking portion of the status monitoring system. The chap-

mainly used for the transmission of the status data, though recently it has also been used for interactive services. These cable trunk amplifier networks provide the main functionality for maintaining signal integrity within metropolitan cable television distribution networks.

Fault detection is performed within these plants by limit checking fault detection with the central status monitoring system employing a fixed order polling scheme to collect the data from the amplifiers status transponders. In total six status data fields useful for early fault detection are monitored for each amplifier. Specifically these fields are:

1. The forward pilot level (a measure of the downstream signal strength)
2. The reverse pilot level (a measure of the upstream signal strength).
3. The raw DC voltage supplied to the amplifier.
4. The regulated or B+ voltage produced by the regulated power supply module located within the amplifier.
5. The current draw of the forward and reverse amplifier modules.
6. The ambient temperature as measured from within the amplifier‘s housing.

Four binary condition flags are also monitored but due to their binary nature, they do not provide any trend information and, hence, cannot be utilized in early fault detection tasks. The status data transponders within these plant do not utilize buffering and, hence, the sampling rate for the collected status signals is directly related to the monitoring system's polling cycle. A more detailed description of these plants and their structure is provided in Appendix A.

The specific data set that will be used in the validation of the proposed early fault detection techniques covers a 13 month period from October 1, 1995 to October 31, 1996 for Rogers Cablesystems Inc. Newmarket, Ontario cable trunk amplifier plant. In total this data set comprises approximately 2.3 Gigabytes of status data collected from 170 of the 354 monitored amplifiers within the plant.

underlying noise to be modeled despite the fact that c_1 and c_2 are unknown. This noise modeling, and subsequent techniques for the estimation of $\tilde{x}(t)$ will be presented in Chapter 5.

It is important to note that the data set $y_q(kT)$ for $k = 1, \ldots, K$ represents status data from only one sensor within one of the given plant's components. For the i^{th} sensor in the j^{th} plant component a separate $y_q^{ij}(kT)$ status data signal is to be produced. The goal of this work is to develop a set of techniques to model the set of status data signals $\{y_q^{ij}(kT) \mid i = 1, \ldots, N_s^j, j = 1, \ldots, N_c\}$ for each component in a manner which facilitates the early detection of fault conditions. No information other than that obtainable from the $y_q^{ij}(kT)$ statistical signals will be used in the development of the early fault detection techniques, leading to a detection system that is generally applicable to large scale engineering plants.

2.4 Example Large Scale Engineering Plant

The example large scale engineering plant that will be used to validate the concepts presented in this work is a cable trunk amplifier plant. These plants are at the heart of metropolitan cable television distribution systems which control the distribution of cable signals from centralized injection sites to the subscribers homes. They can therefore be viewed as a particular type of a large scale engineering plant within the communications industry.

The basic structure of a trunk amplifier plant is that of a tree with the main distribution point at the trunk and the subscribers homes at the leaf nodes. The plants typically consist of three main types of components: cable trunk amplifiers, the interconnecting coaxial cable spans, and the cable amplifier power supplies. Of these only the cable trunk amplifiers are typically status monitored. The plants are asymmetrical in nature with the main high bandwidth communication path being downstream (from the root node to the leaf nodes) and a low bandwidth path upstream (from the leaf nodes to the root node). The high bandwidth path is used mainly for television channels and the low bandwidth path is

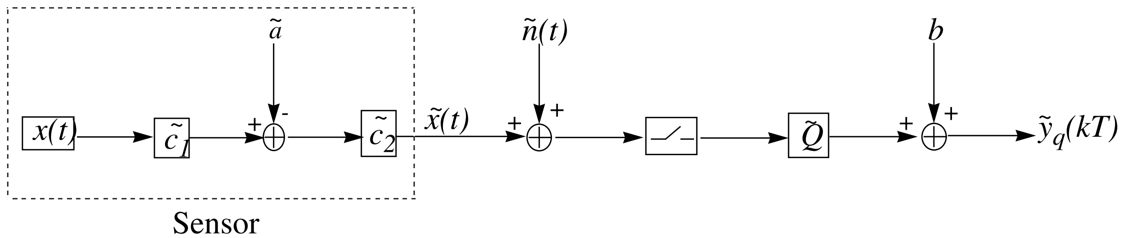


FIGURE 2-3.
Functionally equivalent data collection model.

model is given in Figure 2-3, where $\tilde{c}_1$, $\tilde{c}_2$, and $\tilde{Q}$ are chosen such that $\tilde{y}_q(kT) = y_q(kT)$. This new model can be viewed as looking at the system from the perspective of the output signal $y_q(kT)$ and it can be described by the simplified equation:

$$\tilde{y}_q(kT) = b + \tilde{Q}([\tilde{x}(t) + \tilde{n}(t)]|_{t = kT}) \tag{2.4}$$

This equation can be further simplified by assuming that $b = 0$. This has the effect of changing the absolute value of the quantization levels, q_i, while not changing the quantization step sizes, Δ_i.

$$\tilde{y}_q(kT) = \tilde{Q}([\tilde{x}(t) + \tilde{n}(t)]|_{t = kT}) \tag{2.5}$$

This reformulation of Eq. 2.1 has the advantage of absorbing the two unknown nonlinear functions into the sensor and of allowing the new quantization function $\tilde{Q}(x)$ to be fully described by the time series data given by $y_q(kT)$ for $k = 1, \ldots, K$. The fault detection techniques developed in this work utilize only the signals $y_q(kT)$ in the generation of the behavioural models, hence this approximation does not effect the early fault detection process. The main purpose of this approximation is to enable the quantization and

Mathematically, this sampling and quantization process can be described by the following equation:

$$y_q(kT) = b + c_2(c_1(Q([x(t) + n(t)]|_{t = kT})) - a) \tag{2.1}$$

where T is the sampling period and is assumed to be constant, $k = 0, 1, 2, \ldots$, and the quantization function $Q(.)$ is given by $Q(t) = \sum_{l=0}^{L} q_l u(t - T_l)$, where $u(t)$ is the unit step function and q_l is the l^{th} quantization level.

The functions c_1 and c_2 are nonlinear functions which are not known explicitly. Typically these functions would be hard coded into the low levels of the centralized status monitoring software suite. In addition the offset a would also be typically hard coded in the low levels of the software suite. Therefore, if the centralized status monitoring system is considered as a black box, then given only the values $y_q(kT)$ for $k = 0,...,K$, only b of Eq. 2.1 can be determined directly, since it can be assumed that

$$E[y_q(kT)] = b \tag{2.2}$$

over a suitably long period, $k = 0,...,N,$ of “normal” operation., which in turn implies that

$$E[c_1(c_2(Q([x(t) + n(t)]|_{t = kT})) - a)] = 0 \tag{2.3}$$

over regions of “normal” operation. It is important to note that $Q(x)$ will also be an unknown since the quantization levels will be modified by the unknown functions c_1 and c_2 and hence are not directly measurable from $y_q(kT)$.

A functionally equivalent system to the one above can be constructed by assuming that the operations performed within the central monitoring software on the status data, with the exception of the offset b, are performed instead within the sensor itself. This simplified

utilizes wide bounds and is relatively insensitive to real faults. One method to alleviate, to some extent, the effects of seasonal variations is to periodically manually reset the nominal levels. Obviously, this is a non-ideal solution to use within an automated fault detection system.

2.3 Status Data Signal Model

From the above discussion of the standard fault detection process utilized for large scale engineering plants, a signal model can be constructed for the collected status data. When viewed in terms of a single status monitored variable, the status monitoring system, up to the point where the data is limit checked, can be represented by the block diagram in Figure 2-2. The sensor, located within a component, produces a signal $x(t)$. This signal is then corrupted by various noise sources such as thermal noise, sensor noise, etc. It is assumed that these effects can be modeled cumulatively by the additive Gaussian white noise source, $n(t)$. This combined signal is then sampled and quantized within the component's status transponder. The data is then transmitted over a communications channel, assumed to be error free to the central status monitoring system. The monitoring system converts the data from the sensor domain to the measurement domain by the function c_1 and the offset a. Any corrections required to compensate for known sensor dependencies are performed by the function c_2 and the offset b. It is assumed that any anti-aliasing filters required prior to the sampling step are contained within the sensor itself and, hence, are not explicitly indicated in the figure.

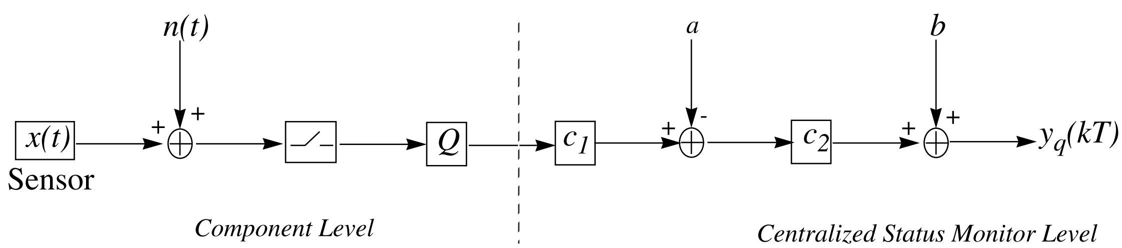


FIGURE 2-2.
Block diagram of the data collection process.

width of the bounds must be set in accordance with the signal's "normal" variations. If the bounds are set too tightly, then the "normal" variations in the data will cause false error flags to be generated. If the bounds are set too widely, then there is an increased probability that a given fault condition will not cause the data to exceed the bounds leaving the fault undetected. Obviously, in each of these cases some means of determining what constitutes "normal" variations is also required. It is important to note, that within limit checking there is an implicit assumption that the data to be monitored is either wide sense stationary or "nearly" wide sense stationary. In particular, the low order statistical moments of the data should change slowly enough to allow for the changes to be tracked through manual adjustment of the limit checking bounds, if the moments change at all.

Second, the technique is not able to detect faults that result in behavioural changes in the components but do not cause the bounds to be exceeded. As long as the data remains between the bounds, any type of behaviour is deemed acceptable. Clearly, this presents a limit on the technique's fault detection capabilities.

Third, with the exception of extreme faults, the technique tends to produce sporadic error flags. A typical fault will not result in the generation of error flags throughout the fault duration. Instead, flags will be produced only when the data happens to exceed the thresholds. This problem is clearly illustrated in Figure 2-1 and leads to significant difficulties when it comes to determining whether the flags were generated by a transient effect or by an ongoing fault condition. Additionally the lack of consistent flag generation complicates the fault diagnosis process.

Finally, even when the bounds have been set according to the principles outlined above, false alarms due to signal dependencies and system dynamics may and typically do still occur. For example, daily or seasonal temperature variations may cause significant perturbations in the status signals, which in turn can cause false alarms to be generated. These types of dependencies should not be reported as faults since they do not relate to repairable phenomena. In general, there is a trade-off between setting the fixed bounds tight enough that real faults are flagged and setting them wide enough that the "normal" status signal variations induced by these types of dependencies go unreported. Either the detection system utilizes tight bounds resulting in a large probability of false alarms or it

ity of the fault condition can be obtained. If only the first bounds are crossed then a minor "fault" has occurred. If both bounds are crossed then a major "fault" has occurred. It is important to note that these "faults" only indicate positions where the upper or lower bounds have been crossed. They do not necessarily relate to a bona fide fault condition within the given component/plant.

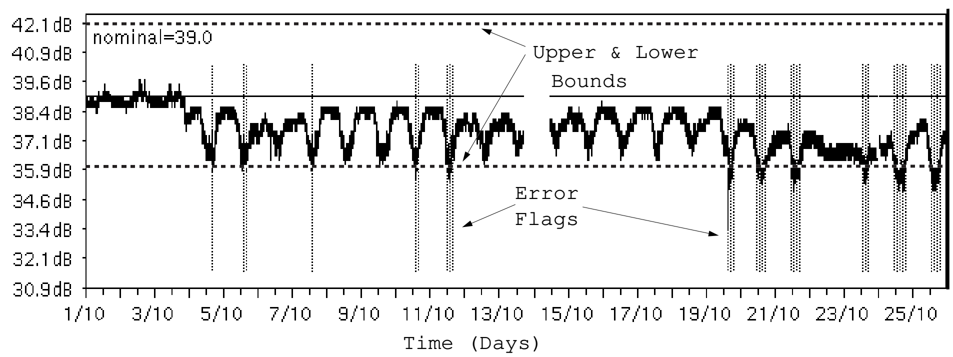


FIGURE 2-1.
Example of limit checking fault detection "fault" flag generation for a status data signal.

Figure 2-1 shows an example of the use of a single set of bounds for one of the status data signals from the example large scale engineering plant to be introduced in Section 2.4. The gray vertical lines indicate times when the limit checking fault section system produced a fault flag. The black horizontal lines indicated the locations of the upper and lower bounds. It is important to reiterate at this point, that the flags generated by this technique are produced on a sample by sample basis and do not rely on any trend information contained in the signal.

2.2.4.2 Limitations

Limit checking as a fault detection process has several serious limitations. First, it relies heavily on the accurate setting of the bounds for its fault detection sensitivity. Ideally, the bound window should be set such that it is centred around the "normal" variation of the fault-free data. If this is not the case (i.e. the signal's mean is close to one of the bounds), then false alarms may be generated due to the normal variations. Additionally the

system, since in general this may be detrimental to the plant and possibly to operational personnel. These facts hinder, in general, the applicability of pattern recognition type approaches to the task of early fault detection within large scale engineering plants.

2.2.4 Limit Checking Fault Detection

Limit checking fault detection is widely used for fault detection within large scale engineering plants mainly for its simplicity and low computational cost. Traditionally, PC's with a fairly low computational power have been used as the platform for the status monitoring systems. Therefore, if several hundred to several thousand components were to be monitored, only relatively simple computational tasks could be performed for fault detection. Hence, in many industrial settings the time domain trend information availability from the status data goes unused and the fault detection is performed, via limit checking, on a sample by sample basis. The development of more advanced detection techniques typically requires large historical status data sets to be available for analysis. In many industrial setting the historical status data sets tend to be discarded due to the economics of storage. The following two sections will outline the basic principles behind limit checking fault detection and its limitations.

2.2.4.1 Overview

The basic idea behind limit checking fault detection is to place bounds above and below the "normal" range of variation of a given status signal. When the signal is seen to pass above or below this "normal" window then a fault flag is generated and passed to the status monitoring system. Obviously, some care must be taken in setting these bounds. If the bounds are too wide then the system will be very insensitive to fault conditions. If, on the other hand, the bounds are too narrow then the detection system will produce a large number of false alarms related to "normal" variations of the signal. Typically, the initial bound setting for a given component is determined through the experiential knowledge of the technical personnel. Once set, if that component is seen to generate a large number of false alarms then its bounds are progressively widened until the false alarm rate drops to acceptable levels. In some systems, two bound windows are utilized with the second window encompassing the first. By utilizing two sets of bounds some indication of the sever-

system. The fault detection techniques implemented within the fault detection portion of the system govern which faults can be detected, the detection latency, whether or not the fault is flagged throughout its occurrence, and whether or not what is detected as a fault is a "true" plant/component fault. In general, it is desirable to have the detection system be capable of continuously flagging the fault condition during its entire duration. It is not desirable to sporadically flag the faults occurrence, since this may complicate the fault location task. Within this work, the term "reliable fault detection" will be used to identify a detection system which is capable of continuously flagging a fault during its complete duration. It is also desirable that the detection system has a low false alarm rate. As many of the flagged events should relate to real faults within the components/plant as possible. The term "confident fault detection" will be used within this work to indicate a detection system with a low false alarm rate. In Section 2.2.4 the principles of limit checking fault detection will be presented. This type of fault detection is the detection approach commonly employed for industrial large scale engineering plants.

Once the fault detection task has been performed, the final task of the monitoring system is to report the fault flags to the technical personnel responsible for the plant's maintenance. Typically, this is done through some type of interactive graphical display. A common feature of these user interfaces is to allow the user the ability to manually focus the data collection and fault detection processes to a specific portion of the plant or to specific components of interest. This feature is available to aid the technical personnel in their fault location tasks. It should be noted that while this feature is utilized, status monitoring for the plant's other components is, generally, suspended resulting in missing data samples for the time period of time in which the "focused" polling is employed.

It is important to note at this point that particular fault modalities may occur quite rarely within a given large scale engineering plant. A large number of possible fault modalities may also exist within the given plant, not all of which will be well understood or even known *a priori*. Typically, it is also not possible to artificially create a statistically representative set of fault behaviours for a given plant by injecting known faults into the

locations of missing samples. The records are time stamped by the central monitoring system, instead of by the transponder themselves, for both technical and economic reasons. It is easier and cheaper to maintain one time stamp clock than to have to maintain multiple clocks, one on each transponder. Additionally, it would be exceedingly difficult to synchronize a large number of independent clocks. The effect of the central time stamping is to introduce a more or less consistent delay into all of the transponder's status data record time stamps. Hence, this approach does not cause significant difficulties when it comes to processing the status data records, even though it does mean that the recorded time stamps do not provide a true measure of the sampling times.

Once the status data records have been time stamped, the central monitoring system must then convert the data from its sensor measurement values to the final signal level values. In general, the sensors do not directly measure the signals of interest. Instead they measure some alternate signal that has a known relation to the signal of interest. To simplify the transponders, it is generally easier, particularly in a homogeneous plant, to have the central monitoring system perform the necessary conversion from the measurement domain into the reported domain for the signal of interest. Typically, during this conversion some corrections to the measurement value may also be made to take into account known dependencies of the sensor. For example, the sensor may be particularly sensitive to temperature variations causing it to give slightly different reading for the same input signal across a certain temperature range. If this dependency is known, the temperature dependency of the sensor can be corrected resulting in a more reliable final sensor value. It is important to note that this type of correction can only be done when the influencing factor is itself subject to status monitoring. Additionally, a typical correction of this type will only reduce the factor's influence. The effected status signal will, therefore, not generally become independent of the influencing factor even after the correction has been performed. Dependencies such as these are major causes of false alarms within limit checking fault detection systems.

Once the status data have been converted to its reported domain and any known correction have been applied, the data can then be processed by the fault detection portion of the central status monitoring system. This section is the critical portion of the monitoring

In general, if there is a large number of transponders within the given plant, the central status monitoring system will use a polling scheme to collect the data from the transponders. Each transponder will be assigned a unique address and the monitoring software will poll each of the transponders in turn for its data. If the transponder does not respond to the poll in a prescribed period of time then that transponder will be labeled by the monitoring system as “not responding”. This information is useful in the fault location process but it is not particularly relevant to the early fault detection task since it occurs only after the fault has caused significant enough effects to make the component’s transponder unreachable. Typically, the order of the polling cycle will remain constant, with the exception of changes to the plant’s structure (i.e. a change in the number of the monitored components), from polling cycle to polling cycle.

If the transponders do not have a buffering capacity, then for each poll they will return one sample from each of their associated sensors. Hence, the sampling period for the status signals will be directly related to the polling cycle of the status monitoring system. In general, the sampling time for the status data will, therefore, vary with variations in the duration of the polling cycle. If standard signal processing techniques are to be used in the analysis of the data (i.e. for early fault detection) then it is important that the status data sampling time can be modeled as being very close to a constant. Obviously, variations in the polling cycle are likely to occur, particularly since the central monitoring system may have to wait to declare a transponder as “not responding”. If only slight variations in the polling cycle occur then it can be argued that the system has a nearly constant sampling time on the large scale. As long as a large enough data set is observed, the sampling time will appear to be constant over its first few significant digits. It is therefore important, in order not to violate this near constant sampling time assumption, that any gaps in the status data be identified and special processing steps be initiated at their beginning and end. Appendix B provides a verification of this constant sampling time assumption for the example large scale engineering plant, which will be introduced in Section 2.4.

The status data records are generally time stamped by the central monitoring system as they are received from the status transponders. These time stamps give the chronological order of the samples and are useful, when analyzing historical data sets, in determining

increases the cost of the transponders and also increases the transmission time for their data records, in general, plants with a large number of monitored components tend to utilize un-buffered transponders.

2.2.3 Status Monitoring System.

Status monitoring can be done either centrally or in a distributed manner. In either case, the issues involved in its development and the underlying processing limitations are similar. Therefore, for simplicity, this section will only directly discuss the issues as they pertain to centralized status monitoring. All of the conclusions though can be assumed to be equally valid for distributed monitoring systems. Typically, in the centralized approach there is one point at which all the status data is collected and analyzed within the large scale engineering plants. This point is generally termed the central monitoring system and typically exists as a software suite on a personal computer (PC). The central monitoring system is responsible for:

- the active collection of the data from the status transponder,
- the identification of un-responsive transponders,
- the conversion of the status data from its measurement domain to its reported domain,
- the analysis of the data for fault conditions (typically done by limit checking),
- the downloading of any required information to the transponders,
- and the reporting of any existing fault conditions to the technical personnel.

In order for the central monitoring system to collect the data from the status transponders there must be some form of communication channel between the transponders and the monitoring PC. This channel could be an application specific bus, a wireless channel, or, in the case of telecommunication plants, the communication channel(s) of the plant itself. For the purposes of this work, it is assume that whatever channel is in use, it acts as an error free channel. Faults are assumed to occur in the plant's components, within the transponders, and within the status monitoring software, but it is assumed that the data sent by the transponder is identical to the data received by the status monitoring software.

ical to the plant's operation will be monitored. Faults in non-monitored components, though, may cause noticeable changes in the observed sensors signal and, hence, may be indirectly observable. Of particular interest within this work are the sensors, typically analog, which provide trend information about the components' behaviours since it is these sensor signals which can be utilized to provide early fault detection. Sensors which provide binary information are not particularly useful to early fault detection tasks since they provide no trend information. They are however useful in determining probably locations of past faults in historical data sets.

2.2.2 Component Status Monitoring Transponders

At each monitored component, a status monitoring transponder is typically present which is responsible for the collection of the data from the component's sensors, the conversion of that data into digital form, and the transmission of that data back to the status monitoring system. Typically, this monitoring system is a digital computer and hence the transponder will be responsible for the sampling and quantization of all of its analog sensor inputs. In general, since a limit checking fault detection system will be in use, this sampling and quantization processes may be performed without the standard signal reconstruction precepts in mind. In particular, the sampling rate may be significantly lower than that warranted by the Nyquist criteria [46] and the number of quantization levels across the signal's dynamic range may be significantly fewer than required for the uniform white noise approximation for the quantization noise [4][46] to hold. Limit checking, to be described in detail in Section 2.2.4, looks at each status data sample in isolation and hence is not reliant on the ability to reconstruct the sampled signals for its fault detection capabilities.

In general, the transponders may or may not have a buffering capacity. If the transponders include status data buffers then the sampling rate for the status signals can be significantly higher than the rate at which the data is transferred to the monitoring system. Without a buffering capacity, the sampling rate of the signal becomes directly related to the rate at which data can be transferred to the monitoring system. Because buffering

tem employed for large scale engineering plants. Included in this discussion will be an analysis of the limitations of this approach, particularly with regards to early fault detection tasks.

2.2 Typical Fault Detection System

The fault detection systems typically employed for large scale engineering plants generally consist of four main components: status sensors located within the components which provide information regarding the "health" of the components, status data transponders located within the components which collect the information from the sensors, a status monitoring system which collects the status data from the transponders, and a limit checking fault detection system which analyzes the collected status data for fault conditions. The following sections will discuss each of these components in turn.

2.2.1 Component Sensors

To perform fault detection some information about the plant and its components must be obtained. This data is collected by placing sensors within the plant/components to measure specific signals of interest. Which signal are measured depends largely on which signals are deemed important by the manufacturer of the components or the operator of the plant. Obviously, the choice of signals to monitor greatly effects which type(s) of faults can be detected. Ideally, these status monitored signals contain sufficient information to detect all the significant fault conditions that may exist within the plant. Since this work is concerned with providing a value-added service to existing large scale systems, the choice of which signals to observe is outside the scope of this work. It is therefore assumed that, for a given plant, all the significant fault conditions cause observable changes to at least one of the signals that has been chosen to be measured. Faults which do not cause changes to the observed signals are deemed to be unobservable faults and, though it is recognized that they may occur, their detection is necessarily outside the scope of this work.

The sensors themselves can be either of an analog or digital nature. For the purposes of this work it is assumed that any necessary filtering, such as anti-aliasing filtering, is performed at the sensor level. It is important to note that not all the components within a plant may be monitored. Typically, for economic reasons only those components which are crit-

plants themselves. In some cases, the sensors measuring the “state” of the components may not provide a direct measurement of the components’ input/output signal(s) and hence white noise modeling techniques would not be applicable based on the available sensor measurements. The components’ behaviours may also exceed the linear system models typically assumed by the white noise modeling techniques. Generally, the only window to analyzing the *in situ* behaviours of the components is provided by the status data.

Accurate fault detection within the domain of large scale engineering plants is therefore a fairly difficult problem. This problem is complicated further by the deployment of fault tolerance techniques within the plants. As the plants under consideration are utilized by industry, some mean of attempting to maintain the integrity of the plant’s operation even through fault events is required. Therefore, as apparent faults are detected, the plants are generally designed to actively and automatically compensate to reduce the impact of the event. This fault tolerance can range from having components automatically switched in to replace ones which appear faulty to having modules within the components adaptively compensate for the component’s reduced performance level. In general, it is this low level fault tolerance which is not directly visible from the status data and hence causes difficulties during fault detection task. In particular, the activation of low level fault tolerance can lead to significant problems in detecting the occurrence and location of fault events since the fault tolerance acts to actively mask the fault’s effects. This problem is particularly severe in early fault detection tasks, since the fault tolerance tends to obscure the subtle behavioural changes in the components. The fault only becomes clearly apparent when the available low level of fault tolerance is consumed and the direct fault effects can be seen. Higher level fault tolerance (above the component level), if its activation is reported to the system through the status data, tends not to further complicate fault detection tasks.

As these plants are used for industrial purposes, some means of detecting the occurrence of faults and initiating the appropriate repair processes is required. The following section will describe the structure and operation of a typical industrial fault detection sys-

The structure of industrial large scale engineering plants tends to evolve over time in a piece-meal fashion through continual upgrade and maintenance processes. As new technologies become available, specific parts or components of the plant may be upgraded. As understanding is gained regarding the plant's operation changes to its components or structure may be made to make it more efficient or its operation, more cost effective. In telecommunications plants, the nature of the plant might change substantially as new services are introduced to existing customers or the existing services are extended to new customer groups. On the whole therefore, these plants tend to be continually changing and hence at any instant in time they are composed of many different type of components, each group of which varies considerably with regards to such things as the versions employed and the individual component ages. It is possible, particularly within the cable television industry to have plants whose major components are almost exclusively homogenous in their function. Even in this case, there tends to be large variations within the service time of the individual components. Some components will be quite old, having been in service for a long time relatively fault free, and some will be relatively new, having replaced components which have recently failed.

Because of these reasons, the interaction of a given plant's components and their *in situ* behaviours tend to be poorly understood. In general, it is not possible or feasible to describe the components' behaviours analytically[1]. In particular, the behaviours exhibited by a component on a lab bench may not concur with the behaviours it exhibits within the plant. The interactions of neighbouring components may result in the component exhibiting a different set of *in situ* behaviours. In general, it is also not possible to model the plant's behaviours by combining separate models for each of the components due to the complexity of the component interactions. Additionally, it is also generally not possible to determine the behaviour of the components *in situ* through pseudo-white noise system identification techniques [3][70]. This is largely due to the fact that the injection of pseudo white noise into the plants is not feasible due to safety concerns or due to the nature of the

---

1. Analytical models for some classes of large scale engineering plants do exist, and hence fault detection for these types of plants is relatively straight forward. Of interest in this work though is the more general set of large scale engineering plants which do not have accurate analytical models.

Chapter 2:

Large Scale Engineering Plants: General Structure and Status Monitoring

2.0 Introduction

The goal of this work is to provide the value-added service of early fault detection to existing large scale engineering plants. The features and characteristics of large scale engineering plants, particularly, as they pertain to fault detection, must therefore be understood. This chapter will focus on providing this overview, with particular attention being paid to the features of the limit checking fault detection systems typically used within such plants. Once the general structure of a typical limit checking fault detection system has been introduced, a signal model for the status data collection process will then be developed. The chapter will conclude by presenting the example large scale engineering plant which will be used throughout this work to provide concrete examples of the points under consideration.

2.1 Large Scale Engineering Plants

For the purposes of this work, large scale engineering plants will be defined as plants consisting of a large number of components (typically several hundred to several thousand) which have complex and typically poorly understood interactions. As such, this type of plant exists within many different industrial settings. For example, within the forestry industry, a pulp and paper mill could be considered as an example of such a plant. Alternatively, within the telecommunications industry, a telephone system or a cable television system could each be considered to be an example plant. Other industries, such as manufacturing or oil and gas, offer other examples of such plants. It is assumed within this work, that these large scale engineering plants include some form of status monitoring system whereby information about the operation of the plant's components, obtained from sensors located within the components, is collected and analyzed for the purposes of fault detection.

1.8 Trademarks

The following is a list of the trademarked words and terms used within this work:

Matlab is a trademark of MathWorks Inc.

C-COR and **C-COR Electronics** are trademarks of C-COR Electronics Inc.

Rogers, **INMS**, and **Demon** are trademarks of Rogers Cablesystems Ltd.

mination of this work will therefore be fault detection systems which are able to learn the behaviour of the plant's components from real, operational status data and be capable of the early detection of faults, including those that are novel.

1.7 Dissertation Outline

The chapter by chapter outline of the remainder of this work is as follows:

Chapter 2 will provide a description of the typical structure and composition of large scale engineering plants and describe how fault detection is usually performed within this domain. Additionally, the example real-world, large scale engineering plant will be introduced.

Chapter 3 will present a discussion of characteristics and features of status data typically generated by the type of large scale engineering plants discussed in Chapter 2. This discussing will lead to the classification of the status data signals into independent, linear dependent, and non-linear, dynamic dependent signals. This chapter will also outline the fault detection methodologies suitable for each class.

Chapter 4 will present a fault detection technique based on fuzzy logic membership functions which is suitable for fault detection of noisy, coarsely quantized signals which exhibit unknown linear dependencies.

Chapter 5 will present an overview of the basic theory of utilizing a recurrent neural network as the basis for detecting faults in signals exhibiting unknown nonlinear, dynamic dependencies. The majority of this chapter will be concerned with developing estimation methods for the underlying sensor signals and underlying sensor noise from the coarsely quantized status data signals.

Chapter 6 will build on the work of Chapter 5 and utilize the underlying sensor signal estimates and underlying noise estimates to developed a methodology for setting the fault detection threshold in a prescribed manner during periods when the neural network is free-running. Additionally, a methodology of quantify when the neural network has adequately learned a given training sequence will be developed.

Chapter 7 will present an evaluation of the two fault detection methodologies developed in this work for the real-world, large scale engineering plant introduced in Chapter 2. This evaluation will focus on quantifying the employed threshold bound widths and false alarm rate of fuzzy membership and recurrent neural network based detection methodologies in comparison with those of the example large scale engineering plant's *in situ* limit checking detection system. In addition, the real-time applicability of the two proposed fault detection methodologies will be validated.

Chapter 8 will present the conclusions obtained from this work and discuss directions for future work.

- The structure (topology) of the given plant is known.
- The sampling rate of the status monitoring system is sufficient to meet the Nyquist criteria.
- System modeling through the injection of pseudo-white random sequences is not possible since the components are assumed to only exhibit their true behaviours *in situ*. Additionally, it is assumed that the status data do not represent a direct measurement of the components' true input/output signal(s).

These assumptions represent a worst case scenario in which little is known or available about the given plant. As a result, a detection system which operates well under these condition should be well suited to modeling a wide variety of large scale engineering plants since little information about the plant or its operations is assumed to be directly available.

1.6 Goals

The goals of this work are, therefore, to develop fault detection approaches which are able to automatically generate behaviour models of the plant's components, and to utilize these models to detect faults within the components, as indicated by changes in their behaviour, as early as possible. To this end, the status data of the example large scale engineering plant will be analyzed to determine which input/output sets provide behavioural models which are most suitable to the fault detection task. The appropriate modeling technology for each particular input/output set will be determined and where possible this modeling will be done using the recurrent neural network architecture presented in Section 1.4.2.2. With the exception of some discussion of stopping points, the issues involved in the neural network training will not be discussed within this work as they are addressed in [35]. The major focus of this work will be on the development of a framework within which the detection technologies can be applied. In particular, techniques will be developed which allow the fault detection thresholds to be set in a prescribed manner. The development of these thresholds will then allow the resulting detection system's fault detection capabilities to be compared with those of the *in situ* limit checking system for a given set of status data obtained from the example large scale engineering plant. The cul-

values for the various learning parameters can also greatly affect the ability of the given network to converge and the number of epochs required. These problems are open questions within the field of neural networks [88].

Finally, the neural network approaches create a black box solution to the modeling problem. No direct information is gained regarding the operation of the given plant through the development or utilization of these models. The fuzzy logic approach presented does provide some information regarding the underlying operation of the given plant. Despite this cost the recurrent neural network approach, outline above, is the only approach presented which is able to directly model both arbitrary non-linearities and arbitrary dynamics. All the other techniques require some *a priori* information be available about the plant under consideration.

1.5 Underlying Assumptions

The assumptions regarding the large scale engineering plants which underlie the work of this dissertation are as follows:

- The number and type of fault modalities experienced by the plant are unknown.
- The plant components act with unknown non-linearities and dynamics.
- Analytical models of the plant and/or its components are unknown or unavailable.
- Large sets of numerical status data composed of both fault-free and faulty plant/component behaviour exist and are available.
- These status data sets contain noise; the exact analytical nature of which is unknown.
- Little or no symbolic information or experiential knowledge regarding the operation/behaviour of the plant is available.
- The plant's topology, components, and component modules change over time through maintenance and upgrade events.
- Novel fault modalities occur.
- For any given fault modality a statistically significant set of occurrences may not be available.
- Fault conditions do not result in clearly separable status data.
- All significant fault conditions will cause some variation in the behaviour of the plant/components as seen through the status data.

1.4.3 Summary of Artificial Intelligence Techniques

The above artificial intelligence modeling techniques offer methods for modeling systems containing arbitrary non-linearities. In all three cases, systems can be constructed which “learn” the system model from input/output “training” pairs to any desired level of modeling accuracy. Additionally, fault detection systems based on these approaches can be developed without *a priori* information regarding nature or number of the fault modalities which may occur in the modeled system. These types of systems therefore offer a promising approach to providing a generalized method of fault detection within the domain of large scale engineering plants.

The major disadvantage of using the artificial intelligence techniques presented to model dynamic non-linear systems is that a large amount of input/output data exhibiting the full range of variations for the given system must be available in order to train the given system to a reasonable level of accuracy. In general, these techniques only provide accurate modeling within the input range for which they were trained. They do not typically perform well on extrapolation tasks. As discussed previously, large scale engineering plants usually employ some form of status monitoring, so large data sets are usually easily obtainable and usually cover, at least in a statistical sense, the operational range of the plant in question.

Additionally, for the neural network approaches there is currently no theory describing how many hidden layer, or scheduler layer neurons will be required to perform a given modeling task. The design task is therefore usually done using rules of thumb gained through experience. The number of training iterations (epochs) required for the neural network weights to converge to an acceptable solution for a given modeling problem is also not known theoretically. Typically, the training process involves selecting a promising network architecture and then training the network until either the desired error level is reached or no further improvement in the error level is obtained. If the network was not able to model the given system, then more neurons are added to the hidden layer(s) and/or scheduler layer, or another hidden layer is added and the network is retrained. This process is then repeated until a suitable error level is reached. The initial weight values and the

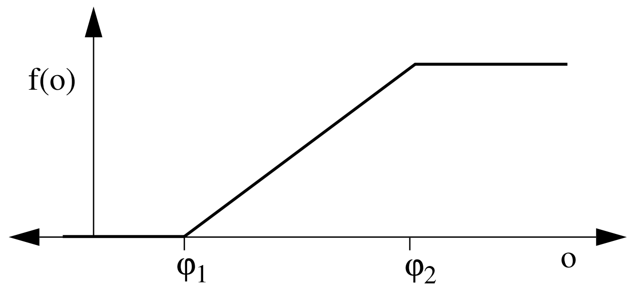


FIGURE 1-11.
Activation function of scheduler sub-class neurons.

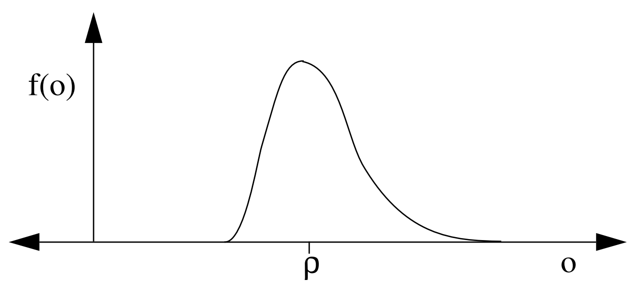


FIGURE 1-12.
Response function of the scheduler class neurons.

different input ranges. In particular, if the non-scheduler neurons utilize a linear activation function then the network can be viewed as approximating the given dynamic system by a series of linear system models, each operating during a particular input interval.

This technique of modeling nonlinear systems by a series of linear systems is used quite commonly to design nonlinear control systems using linear control theory [41][49]. When non-liner activation functions are used, then the neural network models the dynamic system as a series of non-linear systems, each becoming active over a particular input range. This neural network architecture has been shown to be capable of modeling complex nonlinear dynamic systems such as a PUMA robot arm [54].

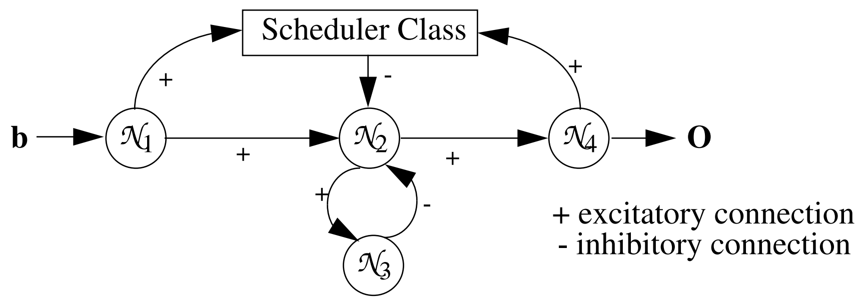


FIGURE 1-9.
Dynamic neural network architecture suitable for dynamic system modeling.

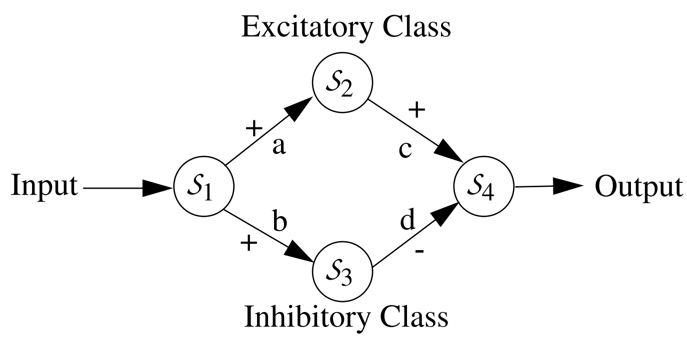


FIGURE 1-10.
Structure of the neurons in the scheduler class.

of scheduler neurons, termed the scheduler class. This scheduler class is actually composed of 4 sub-classes, as shown in Figure 1-10, with each of these sub-class neurons having an activation function of the form shown in Figure 1-11. When the appropriate weight matrices **a**, **b**, **c**, and **d** are chosen, this scheduler class acts as a peaked response network. In particular each of the hybrid neurons of the scheduler class have a response function of the form shown in Figure 1-12.

The scheduler neurons, therefore, each fire over a different input range, thereby, selecting which hidden layer neurons are active over each input range (Figure 1-9). This has the effect of enabling the neural network to employ different dynamic system models over the

$$\boldsymbol{T} = diag\{\tau_1, \tau_2, \ldots, \tau_N\}, \tau_i > 0 \tag{1.53}$$

The *Nx1* vector *b(t)* is the input to the neural network, and *f(**O**)* belongs to a class of non-linear functions termed neuromine functions, defined as $N = \{F|F:R^N \rightarrow R_+^M\}$, f is continuous, f is monotonically non-decreasing, satisfying a Lipschitz condition and $\exists\theta \in R^N$ such that $f(\theta) = 0\}$. Figure 1-8 shows an example of this type of dynamic recurrent neural network. It has been shown in [28] that a sufficient condition for this class of neural networks to be asymptotically stable is that all the positive entries of ***W*** must occur on one side of the main diagonal.

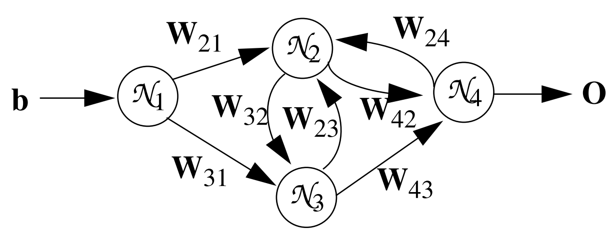


FIGURE 1-8.
Example dynamic recurrent neural network with 4 neuron classes.

This class of networks can be trained by using a slightly modified version of the traditional back-propagation algorithm [71][83][88], a traditional gradient descent optimization technique used for supervised training. The specific algorithm employed is described more fully in [35] and [54] and it differs most noticeably from traditional back-propagation in that the weight adjustment formulas are given in terms of differential equations. In this algorithm care must also be taken that the weight adjustment process does not result in the stability criterion being violated. This is done by clamping the weights such that inhibitory (negative) weight matrices do not become excitatory (positive) and vice versa.

Figure 1-9 shows the particular network architecture found in [54] to be particularly suited to the modeling of dynamic systems. This architecture consists of an input class, and excitatory hidden layer, an inhibitory hidden layer, an output layer, and a sub-network

works. Recurrent neural networks therefore do not act as pattern recognition systems. Instead, they learn the dynamics of the system under consideration. When presented with an input the trained recurrent neural network utilizes its internal dynamic system model to produce an estimate of the system's output given the same input. Recurrent neural networks therefore allow for black-box modeling of systems with both arbitrary non-linearities and arbitrary dynamics.

Recurrent neural networks have not been widely used in the construction of fault detection systems due mainly to stability concerns. For most recurrent neural networks there is no guarantee that, once trained, the resulting neural network will be stable. Of particular interest in this work is a class of recurrent neural networks which is provably asymptotically stable [28]. The operation of this class of networks is governed by the differential equation

$$\dot{\boldsymbol{O}} = -\boldsymbol{T}\boldsymbol{O} + \boldsymbol{W}f(\boldsymbol{O}) + \boldsymbol{b}(t) \tag{1.50}$$

where there are assumed to be N neurons divided into k classes, and

$$\boldsymbol{O} = \begin{bmatrix}\boldsymbol{O}_1 \ \boldsymbol{O}_2 \ \ldots \ \boldsymbol{O}_k\end{bmatrix}^T = \begin{bmatrix}o_1 \ o_2 \ \ldots \ o_N\end{bmatrix}^T \tag{1.51}$$

describes the state of the network. The matrix $\boldsymbol{W}$ describes the network topology and is composed of k^2 sub-matrices $\boldsymbol{W}_{ij}$, each sub matrix describes the connection weights from the i^{th} neuron class to the j^{th} class.

$$\boldsymbol{W} = \begin{bmatrix}\boldsymbol{W}_{11} & \boldsymbol{W}_{12} & \cdots & \boldsymbol{W}_{1k} \\ \boldsymbol{W}_{21} & & & \vdots \\ \vdots & & & \vdots \\ \boldsymbol{W}_{k1} & \cdots & \cdots & \boldsymbol{W}_{kk}\end{bmatrix} \tag{1.52}$$

The matrix $\boldsymbol{T}$ describes the neural relaxation constants given by

ing ability and can reduce the training time and/or the network's complexity. Once trained, the difference signal between the neural network and the operating plant can be used as a residual signal for fault detection.

The major limitation of this approach, like the fuzzy logic approach mentioned previously, comes when the system is used to model dynamic systems. Feed forward neural networks have no ability to directly incorporate time domain information. If a given input pattern generates a given output pattern at time A and the identical input pattern generates a different output pattern at time B then this temporal information will not be incorporated into the neural network system model. Similarly, the same output pattern could be generated by different input patterns at different times depending on the internal state of the plant; such information also would not be captured in the neural network model. In both these examples, depending on the plant under consideration, the patterns may just have to be statistically similar and not exact matches to cause problems for the neural network model. These problems can be mitigated to some extent by periodically re-training the neural network, once again provided that the system's dynamics cause the system to change suitably slowly. This re-training requires an external system to detect the changes in the operating point due to the plant's dynamics. Such a system may be quite difficult to construct and may also result in a large number of re-training phases to be initiated.

Unlike the classical techniques and fuzzy logic techniques, the application of the feed forward neural network models results in no increased understanding of the underlying plant operation. The neural network learns "rules" to perform the input/output mapping but these rules are contained within the interconnection weights, the choice of activation functions, and the network's structure. As such, they can not be extracted from the trained neural network. Neural networks are a black-box approach and, for most networks, there are currently no means by which to look into the box to determine its "rules" of operation.

1.4.2.2 Recurrent Neural Networks

Unlike feed forward neural networks, recurrent neural networks are capable of directly learning the temporal information present in the plant's input/output data pairs. This temporal processing is possible since the feedback path(s) impart a "memory" to the net-

In both cases, neural networks, once trained, basically act as input/output maps. The networks are therefore generally structured such that the number of input layer neurons is the same as the number of external inputs, and the number of output layer neurons is the same as the number of outputs. The number of hidden layers and the number of neurons in each hidden layer depends on the desired functionality of the network. In general, the more hidden neurons the more powerful the resulting network, although at the cost of longer training times and possibly poorer convergence. Currently, it is an open research problem to determine how many hidden layer neurons are required to be able to solve a given task. Some algorithms, though, do exist to iteratively determine near-optimal structures for the hidden layer(s) on an iterative basis for a given mapping task [36][37][50].

1.4.2.1 Feed Forward Neural Networks

Three layer feed forward neural networks have been proven to be universal approximators [26][51]. Provided enough hidden layer neurons are used, these networks can approximate any non-linear function to any desired degree of accuracy. These networks are therefore a natural choice for fault detection systems since they address the modeling error problems inherent in the classical techniques. Several fault detection systems based on feed forward neural networks have been developed [6][7][17][19].

Typically, these types of systems have been constructed for small scale plants with known fault modalities. In this context, the neural networks act as non-linear pattern recognition systems. During the training phase the neural networks learn to identify certain input pattern spaces with certain outputs. The feasibility of this approach relies on the fact that the input/output pattern space does not have an excessive number of pattern classes. If the number of pattern classes becomes too large (i.e. the input to output mapping approaches one to one) then the neural network will require an excessive number of neurons and/or an excessive training time to model the plant effectively. The advantage that these systems have over traditional pattern recognition techniques is that they inherently employ high order, discontinuous pattern boundaries. Additionally, no knowledge about what constitutes “good” features is required to build the classifier. The neural network inherently “learns” which features to utilize. Although not strictly required, pre-processing the raw input data into feature vectors can significantly improve the network’s model-

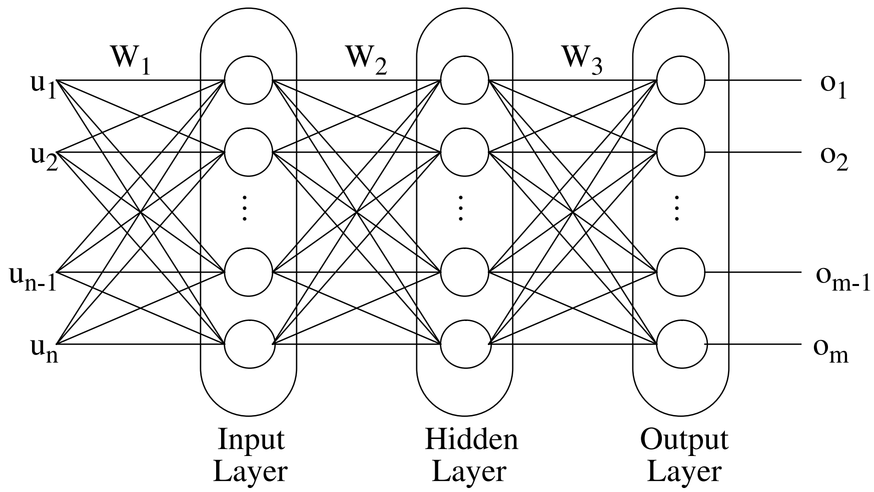


FIGURE 1-7.
Fully interconnected 3-layer feed forward neural network.

Under this model the functionality of the neural network is mainly determined by the values of its interconnecting weights. These weights are usually set through either a supervised or unsupervised learning process. In supervised learning a training set of input/output pairs are applied to the network and the interconnection weights are adjusted through an optimization procedure to minimize an objective error criteria, typically the mean square error between the neural network's actual output and its desired output. Typically this training procedure requires multiple passes (epochs) through the complete training set before a minimum is reached. Back propagation is one example of such an optimization procedure.

Unsupervised learning differs in that the desired output is not known *a priori*. Instead the weights are updated as the network processes data. An objective function, based on external criteria, is used to measure the quality of the network's performance for each input. If the network is performing well, then the weights that contributed to the given output are increased. If the network performs poorly then the contributing weights are decreased. Hebbian learning is an example of an unsupervised learning scheme.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{1.46}$$

and

$$f(x) = \frac{1}{1 + e^{-\alpha x}} - \frac{1}{2} \tag{1.47}$$

By themselves, single neurons can only perform fairly simple tasks such as the partitioning of a linear separable input space. The real power of the neural network paradigm comes when multiple neurons are interconnected to form neural networks. These interconnections are typically done by grouping the neurons into layers and then making the interconnections between the layers. The terminology that is used with this layer structure is that the initial layer is named the input layer, the middle layer or layers are named the hidden layer(s), and the last layer is named the output layer. Typically interconnections between neurons within the same layer are not allowed, although for some types of neural networks, most notably Hopfield-type networks, self-connections are allowed. If the interconnections provide no feedback path(s) then the network is termed a feed forward neural network, while if feedback path(s) exits then the network is termed a recurrent neural network. Figure 1-7 shows a fully interconnected 3-layer feed forward neural network. The output of each layer of this network is given by the matrix equation:

$$\boldsymbol{O}_k = f(\boldsymbol{W}_k \boldsymbol{U}_k) \tag{1.48}$$

where

$$\boldsymbol{W}_k = \begin{bmatrix} w_{11} & w_{21} & \dots & w_{n1} \\ w_{12} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{1m} & \dots & \dots & w_{nm} \end{bmatrix}, \boldsymbol{O}_k = \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_m \end{bmatrix}, \text{ and } \boldsymbol{U}_k = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \tag{1.49}$$

represent the weights, outputs, and inputs for layer k.

periodically re-training the fuzzy system. This re-training phase requires that some means of detecting a change in the plant's operational point caused by the inherent plant dynamics, as opposed to a change due to a fault condition, be available. This limits this approach to plant's possessing lower order dynamics since enough time must elapse for the training phase to be completed prior to the next change in the plant's operating point.

1.4.2 Neural Network Based Fault Detection

Neural networks are a computation paradigm originally based on attempts to simulate the processing techniques utilized by biological organisms. The basic computational element of a neural network is the neuron. This is a simple computational device, the output of which is typically generated by passing the sum of a set of weighted inputs through non-linear activation functions (Figure 1-6).

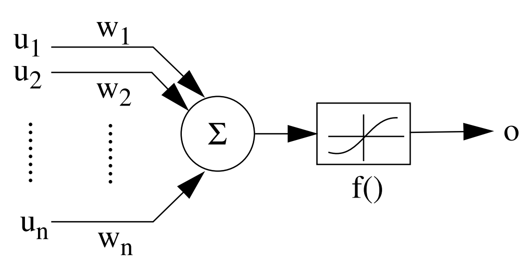


FIGURE 1-6.
Basic structure of an artificial neuron.

Mathematically this output is given by

$$o = f\left(\sum_{i=1}^{n} w_i u_i\right) \tag{1.45}$$

The activation function *f(.)* can be any of a variety of non-linear functions. Two of the most commonly used activation functions are the semi-linear and sigmoid functions given respectively by

$$\hat{\boldsymbol{y}} = g(\boldsymbol{x}, \theta) = \frac{\sum_{j=1}^{N} \mu_{H^j}(\boldsymbol{x})\,\hat{\boldsymbol{y}}^j}{\sum_{j=1}^{N} \mu_{H^j}(\boldsymbol{x})} \tag{1.44}$$

where $\theta = \left[\upsilon_1^T \ \dots\ \upsilon_N^T\ \boldsymbol{c}_1^T\ \dots\ \boldsymbol{c}_N^T\right]^T$.

This fuzzy system, given a particular input vector, generates an estimate of the plant's output by modeling the output as a weighted sum of a set of linear system models. The contribution of each model being controlled by the degree of membership the input data has in each of the *N* fuzzy classes, which is determined through the fuzzy membership functions. The difference between the estimated output and the actual plant output can then be used as a residual signal for fault detection, in the same way that residuals are used in the classical fault detection techniques outlined previously. This particular fuzzy system relies on three pieces of *a priori* information: the number of fuzzy rules *N*, the number of training data input/output pairs $m_{F,}$ and the stopping criteria ε_c for the learning phase. No assumptions are made regarding the number of fault modalities that may exist within the plant.

This particular system utilizes a piece-wise linear approximation to the given plant, but the if/then rules can be modified so that non-linearities can also be learned and modeled. Under certain conditions regarding the specifics of the rules and membership functions' structure, it has been proven that such systems operated as universal function approximators [14][56] and, therefore, can be used to model any arbitrary non-linear system. The number of rules, though, required to model complicated systems tend to become quite large.

The major limitation of this approach comes when it is employed to model dynamic plants. The system is not able to directly capture information about the plant's dynamics since it is acting basically as a pattern recognizer. For systems with low order dynamics (i.e. slow variations in the plant's nature of operation) this limitation can be mitigated by

$$\mu_{ij}^{l+1} = \left[\sum_{k=1}^{N} \left(\frac{\|\boldsymbol{x}_i - \upsilon_j^l\|_2^2}{\|\boldsymbol{x}_i - \upsilon_k^l\|_2^2} \right)^{m-1} \right]^{-1} \tag{1.40}$$

where $\|z\|_2 = (zz^T)^{1/2}$ and learning terminates when $\left\|\upsilon_j^{l+1} - \upsilon_j^l\right\| < \varepsilon_c$.

Once trained, the system then processes input data in accordance with a set of N if/then rules of the form:

$$\text{If } H^j \text{ Then } \hat{\boldsymbol{y}}^j = c_0^j + c_1^j x_1 + \ldots + c_n^j x_n$$

where $H^j = \{ (\boldsymbol{x}, \mu_{H^j}(\boldsymbol{x})) \mid x \in U_1 \times U_2 \times \ldots \times U_n \}$ is the j^{th} input fuzzy set, U_i is the i^{th} input universe of discourse, $\hat{\boldsymbol{y}}^j$ is the portion of the output generated by the j^{th} rule, and $\mu_{H^j}(\boldsymbol{x})$ is the membership function for H^j given by:

$$\mu_{H^j}(\boldsymbol{x}) = \left[\sum_{k=1}^{N} \left(\frac{\|\boldsymbol{x}_i - \upsilon_j\|_2^2}{\|\boldsymbol{x}_i - \upsilon_k\|_2^2} \right)^{m-1} \right]^{-1} \tag{1.41}$$

The values of $\boldsymbol{c}_j = \begin{bmatrix} c_0^j & c_1^j & \ldots & c_n^j \end{bmatrix}^T$ are obtained from the training set by minimizing

$$J_j = \sum_{i=1}^{m_F} \mu_{ij} (\boldsymbol{c}_j^T \boldsymbol{x}_i - \boldsymbol{y}_i)^2 \tag{1.42}$$

The values of the c_j which minimize J_j can be found analytically to be

$$\boldsymbol{c}_j = \left(\hat{X} D_j^2 \hat{X}^T \right)^{-1} \hat{X} D_j^2 Y \tag{1.43}$$

where $\hat{X} = \begin{bmatrix} 1 & \ldots & 1 \\ \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_{m_F} \end{bmatrix}$, $\hat{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \ldots & \boldsymbol{y}_{m_F} \end{bmatrix}$ and $D_j = diag\left(\begin{bmatrix} \mu_{1j} & \ldots & \mu_{m_F j} \end{bmatrix} \right)$.

The final system output, $\hat{\boldsymbol{y}}$, is then given as a weighted average of the N partial outputs by:

In the second approach the fuzzy rule base and membership functions are automatically "learned". Input/output data pairs are provided to the fuzzy systems during a training phase and the system learns to generate the appropriate crisp output for each given input. Once trained the difference signal between the fuzzy system and the operating plant or component can be used as the fault detection residual as in the classical methods outlined previously. There are a variety of methods by which the "learning" can take place. The following paragraphs detail one such system, described in [58]. This system is described in some detail to provide an overview of the components of such a detection system. Other fuzzy detection systems are similar in gross structure but vary in the details of the individual component functions.

The method given in [58] is based on the work of [74], and starts by specifying N fuzzy clusters, one for each fuzzy rule, with prototype centres υ_j which partition the input space. As the training data set is processed, these clusters are adjusted through fuzzy c-means clustering [8][9][86] to minimize:

$$J = \sum_{i=1}^{m_F} \sum_{j=1}^{N} (\mu_{ij})^m (\boldsymbol{x}_i - \upsilon_j)^T (\boldsymbol{x}_i - \upsilon_j) \tag{1.38}$$

where $m > 1$ is a design parameter which controls the level of "fuzziness" of the system, m_F is the number of training points, $\boldsymbol{x}_i = \begin{bmatrix} x_1^i & x_2^i & \dots & x_n^i \end{bmatrix}^T$ is the i^{th} n-dimensional input data vector, and μ_{ij} is the value of the j^{th} membership function for the i^{th} input vector.

The learning process begins with the random initialization of the υ_j's. These values are then iteratively adjusted through the training phase according to the following equations:

$$\upsilon_j^{l+1} = \frac{\sum_{i=1}^{m_F} (\mu_{ij}^l)^m \boldsymbol{x}_i}{\sum_{i=1}^{m_F} (\mu_{ij}^l)^m} \tag{1.39}$$

Two main approaches have been taken in applying fuzzy logic to the fault detection problem. These approaches mainly vary in how the fuzzy membership functions and fuzzy rules are generated. The first approach utilizes a linguistic fuzzy rule base of the form:

Example Linguistic Fuzzy Rule Base

If 'component A is faulty' then signal s_1 or s_4 will be medium or low
If 'component B is faulty' then signal s_1 will be medium or high
If 'component C is faulty' then signals s_2 will be low and s_1 will be high
If 'component D is faulty' then signals s_3 and s_4 will be low

$\vdots \qquad \vdots \qquad \vdots$

The rules, in this case, are developed manually with the aid of a domain expert. The membership functions $\mu_x(s_i)$, also must be manually derived and are typically of the form shown in Figure 1-5. Fault detection is performed by utilizing the inference engine to compare the state of the plant, given by the fuzzy inputs, with the fault states described in the fuzzy rule base. If a match is found then the given fault condition is reported. This approach has the advantage that it is easy to add diagnostic processing to the detection system. Additionally, the exact operation of the plant does not need to be known, though the nature of the fault modalities must be well understood. This type of fuzzy logic approach is unsuitable for plants with poorly understood characteristics or which have an incompletely known set of fault modalities. The work of [2] provides an example of such a fault detection system.

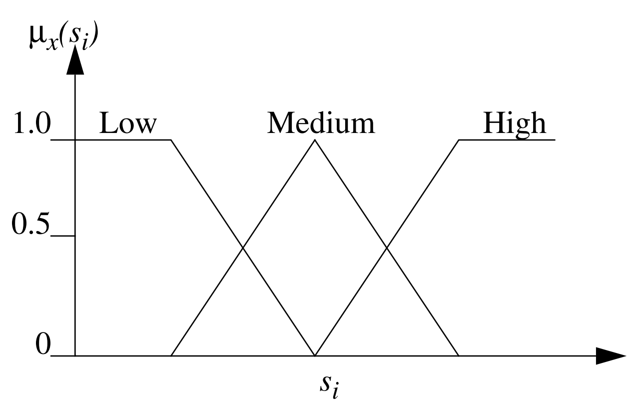


FIGURE 1-5.
Fuzzy membership functions.

detection. These techniques fall into two basic categories: those based on fuzzy logic theory, and those based on neural networks. The neural network category can be further separated into techniques based on feed forward networks and those based on recurrent networks. This section will provide a brief overview of the different technologies and how they have been applied to the fault detection task.

1.4.1 Fuzzy Logic Based Fault Detection

Fuzzy logic is a computational paradigm which allows for data processing to be performed despite uncertainty. The basic structure of a fuzzy logic system is shown in Figure 1-4. The constituents of a basic fuzzy system are the fuzzy sets, and related membership functions, the fuzzy rule base, the fuzzy inference engine, and the de-fuzzifier. The membership functions are used to convert crisp input data into weighted fuzzy sets, where the weights give a measure of how strongly each input belongs to each particular fuzzy set. Once fuzzified, the inputs are then operated on by the fuzzy inference engine in accordance to its fuzzy rule base, which is usually in the form of if/then statements. The resulting outputs of the inference engine must then be converted back to crisp values via the defuzzifier. This process can be performed in a number of ways depending on the particular implementation of the fuzzy system.

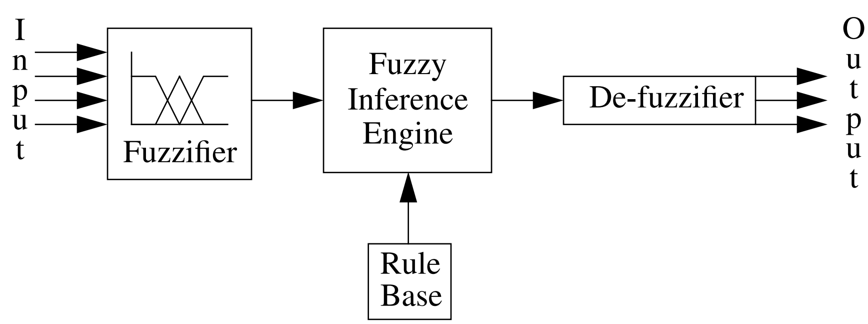


FIGURE 1-4.
Block diagram of a basic fuzzy processing system.

1.3.3 Summary of Classical Techniques

As indicated by the above discussions the classical dynamic system fault diagnosis and location techniques are not suitable for developing a general large scale fault detection system for the following reasons:

1. To be applied, an analytical model of the system must be available.
2. Nonlinear system models, if used, must be developed on a case by case basis.
3. Modeling errors are one of the most significant contributions to reduced system performance.
4. Most of the techniques require that the number of fault modalities be known *a priori*.

In large scale systems, analytical models are typically unavailable due to the complexity of the systems. A general detection system for large scale systems should also be able to model arbitrary non-linearities within the system, ideally to an arbitrary degree of accuracy. Due to the complexities of large scale systems caused by the large number of components undergoing complex interaction, it is not feasible to be able to know all the possible non-linear effects that the given system may display. Also, as components within the system are upgraded or repaired new non-linearities may become apparent, particularly during fault occurrences. Similarly, the complexity of large scale systems generally makes it infeasible to know all the possible fault modalities which may exist within a given system. Hence, a general detection system should not be reliant upon *a priori* knowledge about the number of fault modalities in order to perform accurately and efficiently. Robust detection methods based on the techniques outlined above have been developed which are less sensitive to the problem of modeling error [39][40][69]. The problem with these robust techniques, though, is that essentially they achieve robustness by treating the modeling error as a noise source and, hence, there is a loss of sensitivity in the resulting fault detection system.

1.4 Artificial Intelligence Fault Detection Techniques

To address some of the limitations of the classical techniques, particularly the need to model arbitrary non-linear dynamics, and the need to deal with an unknown set of fault modalities, artificial intelligence techniques have been applied to the domain of fault

1. Choose a parametric model of the system such as

$$a_n y^{(n)}(t) + \ldots + a_1 y(t) + y = b_0 u(t) + \ldots + b_m u^{(m)}(t) \tag{1.34}$$

2. Determine the relationship between the model parameters, θ_i, and the system's physical parameters, p_j.

$$\theta = f(\boldsymbol{p}) \tag{1.35}$$

where

$$\theta = \begin{bmatrix} a_1 \ \ldots \ a_n \ b_0 \ \ldots \ b_m \end{bmatrix}^T \tag{1.36}$$

3. Identify the model parameter vector θ using the input vector $\boldsymbol{u}$ and the measured output vector $\boldsymbol{y}$.

4. Determine the system's physical parameters by using the inverse function

$$\boldsymbol{p} = f^{-1}(\theta) \tag{1.37}$$

5. Calculate the deviation, $\Delta\boldsymbol{p}$, between the physical parameter vector and the nominal parameter vector obtained from the system when it was operating free of faults.

6. Perform fault detection by comparing the parameter deviation, $\Delta\boldsymbol{p}$, with a library of fault/parameter deviation relationships.

This technique, like the state estimation techniques, also requires an analytical model of the system to be constructed and, hence, will be subject to modeling errors. Like the dedicated observer schemes, non-linear models may be utilized to reduce these errors, though, these techniques are limited to an application by application approach. In addition, this technique performs fault detection by utilizing a history of fault occurrence and parameter deviation pairs. The process of recognizing these patterns is not a trivial task in large scale systems. Likewise, the process of identifying the measurement parameter to physical parameter relationships and their associated inverses may also be a non-trivial task within the context of large scale system.

$$\boldsymbol{r}_j = \boldsymbol{C}\varepsilon_j \tag{1.32}$$

by introducing the factor

$$\boldsymbol{k}_j^* = \boldsymbol{A}\boldsymbol{f}_j^* - \alpha\boldsymbol{f}_j^* \tag{1.33}$$

where α is an arbitrary scale factor and $\boldsymbol{f}_i^*$ is the fault direction associated with the j^{th} sensor such that $\boldsymbol{C}\boldsymbol{f}_j^* = \bar{\boldsymbol{k}}_j$.

The feedback gain matrix is then chosen in such a way as to maximize the separation between the various fault modalities by placing each of the r actuator and component fault residuals in the direction of $\boldsymbol{C}\boldsymbol{k}_i$, and the j sensor faults residuals in the plane described by $\boldsymbol{C}\boldsymbol{f}_j^*$ and $\boldsymbol{C}\boldsymbol{k}_j^*$. The main advantages of this approach over the dedicated observer approach is that the fault detection properties of the system are optimized by the appropriate choice of the feedback gain matrix, $\boldsymbol{H}$, and that the resulting residual are independent of the fault mode f_i. The residuals will be projected along the same direction regardless of the size or duration of the given fault.

Like the other systems mentioned, this approach also suffers from modeling error due to the use of the linear system model. In addition, since this approach does not account for effects due to disturbances, parameter variations, or system noise the system must be modeled very precisely for the resulting detection system to provide reasonable levels of performance. As can be seen by the above discussion, this approach is also limited in the sense that the number of failure modalities for the various parts of the system must be known *a priori*.

1.3.2 Parameter Estimation Techniques

Parameter estimation techniques differ from the state estimation technique in that the goal of the system is to detect faults based on changes in the system's physical parameters instead of by observing changes or inconsistencies in its state descriptions. The basic steps involved in a parameter estimation system are as follows [52][55]:

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \bar{\boldsymbol{k}}_j \bar{f}_j(t) \tag{1.24}$$

where $\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{y}$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are identical to the state space description used in the dedicated observer approach, and $\boldsymbol{k}_i$ represents the $nx1$ fault directions used to model actuator and component faults, $f_i(t)$ is an arbitrary time function which equals zero in the non-faulty case, and $i = 1, 2, \ldots, r$, where r represents the number of fault directions. Similarly $\bar{\boldsymbol{k}}_j$ and $\bar{f}_j$ represent the fault directions and fault modes for the j plant sensors.

The observer equations obtained from this state space description are given by

$$\dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{H}\boldsymbol{C})\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{H}\boldsymbol{y} \tag{1.25}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{x}} \tag{1.26}$$

In the case of an actuator component failure, the state estimation error and the output error are given by

$$\dot{\varepsilon} = (\boldsymbol{A} - \boldsymbol{H}\boldsymbol{C})\varepsilon + \boldsymbol{k}_i f_i \tag{1.27}$$

$$\boldsymbol{r} = \boldsymbol{C}\varepsilon \tag{1.28}$$

For the case of a fault within the j^{th} sensor the resulting errors are given by

$$\dot{\varepsilon}_j = (\boldsymbol{A} - \boldsymbol{H}\boldsymbol{C})\varepsilon_j + \boldsymbol{h}_j \bar{f}_j \tag{1.29}$$

$$\boldsymbol{r}_j = \boldsymbol{C}\varepsilon_j + \bar{\boldsymbol{k}}_j \bar{f}_j \tag{1.30}$$

where $\boldsymbol{h}_j$ is the j^{th} column of the feedback gain matrix. Eq. 1.29 and Eq. 1.30 can be rewritten as

$$\dot{\varepsilon}_j = (\boldsymbol{A} - \boldsymbol{H}\boldsymbol{C})\varepsilon_j - \boldsymbol{k}_j^* \bar{f}_j + \boldsymbol{f}_j^* \tag{1.31}$$

paring these non-faulty parameter levels with those obtained from the operating system. If the system's current innovation parameters exceed their nominal levels by the prescribed threshold(s) then the system is deemed to be experiencing a fault condition. Since, typically, several statistical parameters of the innovation need to be measured, this approach can also provide some degree of fault location through the use of multiple hypothesis testing techniques such as Bayesian decision theory.

In general, in order to perform fault location using dedicated observers it is necessary either to utilize multiple observers, one for each sensor, to generate the estimated output vector or to utilize a single observer for the most reliable output signal and to generate the entire estimated output vector from this observer's output. In both cases the estimated output vector is compared with the actual output vector in order to determine which sensor the fault is affecting. This technique, therefore, like the parity check scheme, requires some knowledge of the expected number of fault modalities that are likely to occur within the given plant.

The models described so far in this section are linear models and, therefore, they are also susceptible to modeling errors. Within the dedicated observer approach, it is possible to utilize non-linear system models. These non-linear models have to be tailored to the specific plant under consideration so that the model contains the same type of non-linearities exhibited by the plant. Although the use of non-linear models may improve the resulting system's overall accuracy, it may become difficult to maintain the resulting dedicated observer's stability. One of the advantages of staying with the linear system models is that, despite the increase in modeling error, the resulting observers are known to be stable.

1.3.1.3 Detection Filters

Detection filters are very similar to the dedicated observer approach except that a particular form of the feedback gain matrix, $\boldsymbol{H}$, is chosen and a slightly different form of the system's state space description is used [84].

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{k}_i \mathrm{f}_i(t) \tag{1.23}$$

The feedback gain matrix $\boldsymbol{H}$ is required by the system for several reasons. Namely, it helps to maintain the stability of the model when the plant enters unstable states, it compensates for differences in the initial conditions, and it provides a degree of freedom in the design to make the system more robust by designing a filter to de-couple the effects of faults from the effects of unknown inputs.

In this approach, the estimated state vector, $\hat{\boldsymbol{x}}$, and the estimated plant output, $\hat{\boldsymbol{y}}$, are given by the state space equations

$$\dot{\hat{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{HC})\hat{\boldsymbol{x}} + \boldsymbol{Bu} + \boldsymbol{Hy} \tag{1.19}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{x}} \tag{1.20}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are the normal state space matrices.

By combining Eq. 1.1, Eq. 1.2, Eq. 1.19 and Eq. 1.20, the state estimation error, $\varepsilon = \boldsymbol{x} - \hat{\boldsymbol{x}}$, and the output estimation error, $\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$, can then be given by

$$\dot{\varepsilon} = (\boldsymbol{A} - \boldsymbol{HC})\varepsilon + \boldsymbol{Ed} + \boldsymbol{Kf} - \boldsymbol{HFd} - \boldsymbol{HGf} \tag{1.21}$$

$$\boldsymbol{e} = \boldsymbol{C}\varepsilon + \boldsymbol{Fd} + \boldsymbol{Gf} \tag{1.22}$$

Ideally the output of the comparison of the actual output and the estimate output will be zero when the plant is operating correctly ($\boldsymbol{e} = \boldsymbol{0}$). Due to system noise this will not be the case and it is necessary to set an appropriate threshold level in order to determine the difference between the fault free and faulted cases. In practice, this threshold is quite difficult to determine since its optimal setting will vary with the changes in the input signals, the variations in the system dynamics, and the magnitude and nature of system disturbances. If the threshold is set too low then a large number of false alarms will be produced, and if it is set too high then some system faults will not be detected.

When a Kalman filter is used as the estimator, the resulting output of the comparisons is an innovation representing the inherent noise of the system. In this case, fault detection is performed by recording the nominal statistical parameters of this innovation and com-

The residual, therefore, for the temporal redundancy case is a simple input-output model for part of the plant dynamics and fault detection can be performed by comparing the residual to the threshold vector ***b*** as in the direct redundancy case.

As can be seen by the above discussion, this fault detection technique is fairly simple and easy to apply. This technique does suffer several problems though. First, there is the need to identify the appropriate values for the threshold vector. In practice this is very difficult to do and, in addition, as the dynamics of the plant vary the threshold vector must be continually updated to reflect these changes. Second, this technique only utilizes a linear model of the plant under consideration and, hence it will be subject to modeling errors which may cause faults to be missed or erroneous faults to be detected. Third, since the number of detectable faults within this system is dependent on the number of parity check equations available, the number of possible fault modes of the plant must be known *a priori*. Due to the complexity of large scale engineering plants, this information is not generally available.

1.3.1.2 Dedicated Observer Schemes

In the dedicated observer techniques, the fault detection task is performed by reconstructing the system outputs from the measurements with the aid of observers, for the deterministic case, or Kalman filters, for the stochastic case, and using the resulting estimation error or innovation as the residual [34][39][40]. Figure 1-3 illustrates the basic block diagram of these type of schemes.

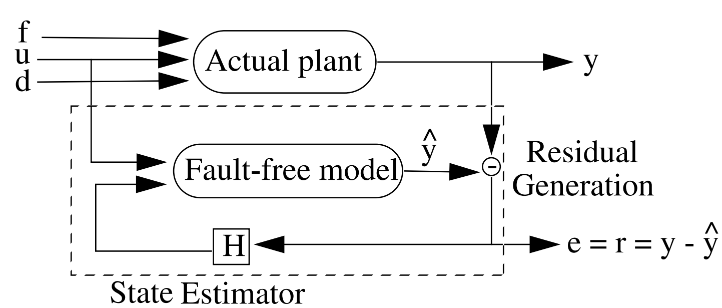


FIGURE 1-3.
Basic block diagram for dedicated observer fault detection schemes.

where $\boldsymbol{x}$ is the $nx1$ state vector, $\boldsymbol{u}$ is the $px1$ input vector representing the actuator inputs, $\boldsymbol{y}$ is the $qx1$ output vector representing the sensor measurements, and $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are the normal state space matrices.

A parity subspace of order s can then be defined as

$$\boldsymbol{P} = \left\{ \mathbf{v} | \mathbf{v}^T \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{\mathrm{S}} \end{bmatrix} = 0 \right\} \tag{1.15}$$

At a time instant k, any of the $(s+1)\ q$ dimensional vectors $\boldsymbol{v}$ may be used to perform a parity check by calculating the residual $\boldsymbol{r}(k)$ as

$$\mathbf{r}(\mathrm{k}) = \boldsymbol{v}^T \left[\begin{bmatrix} y(k-s) \\ \vdots \\ y(k) \end{bmatrix} - \boldsymbol{H} \begin{bmatrix} u(k-s) \\ \vdots \\ u(k) \end{bmatrix} \right] \tag{1.16}$$

where

$$\boldsymbol{H} = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ \boldsymbol{CB} & 0 & & & \vdots \\ \boldsymbol{CAB} & \boldsymbol{CB} & 0 & & \vdots \\ \vdots & & & 0 & \vdots \\ \boldsymbol{CA}^{s-1}\boldsymbol{B} & \dots & \dots & \boldsymbol{CB} & 0 \end{bmatrix} \tag{1.17}$$

Substituting Eq. 1.12 and Eq. 1.13 into Eq. 1.15

$$\boldsymbol{r}(k) = \boldsymbol{v}^T \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^s \end{bmatrix} \boldsymbol{x}(k-s) \tag{1.18}$$

The projection matrix $\boldsymbol{V}$ defines q distinct fault directions associated with each sensor measurement. If a fault occurs such that the i^{th} sensor measurement is affected then $\boldsymbol{p}$ will change in the direction determined by the i^{th} column of $\boldsymbol{V}$.

For the purposes of fault detection and location, a residual may be formed by comparing a model of the nominal plant with the actual plant

$$\mathbf{r} = \mathbf{y} - \mathbf{C}\hat{\mathbf{x}} \tag{1.10}$$

where

$$\hat{\boldsymbol{x}} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{y} \tag{1.11}$$

is a least-squares estimate of the true measurement vector $\boldsymbol{x}$. The residual vector $\boldsymbol{r}$ is related to the parity vector $\boldsymbol{p}$ in that

$$\boldsymbol{r} = \boldsymbol{V}^T\boldsymbol{p} \tag{1.12}$$

The task of fault detection and isolation under the parity check scheme for direct redundancy, therefore, becomes

- Find an estimate $\hat{\boldsymbol{x}}$ of the process variables.
- Calculate the residual vector $\mathbf{r}$ and determine if any of threshold bounds, b_i, are exceeded.

In the case of temporal redundancy the goal is to develop the set of parity equations in terms of the input/output relationship between the actuator inputs and the resulting sensor outputs. Assuming a discrete time system, the system model that is under consideration becomes

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \tag{1.13}$$

$$\mathbf{y}(\mathrm{k}) = \mathbf{C}\mathbf{x}(\mathrm{k}) \tag{1.14}$$

where $\boldsymbol{y}$ is the *qx1* measurement vector obtained from the sensors, $\boldsymbol{C}$ is the *qxn* measurement matrix, $\boldsymbol{x}$ is the *nx1* true (fault free) measurement vector, and $\Delta\boldsymbol{y}$ is the *qx1* error vector. A value of $\Delta y_i > b_i$ defines a fault within the plant which is indicated by the i^{th} measurement variable or i^{th} sensor where $i = 1, 2, \ldots, q$.

The goal of this type of fault detection scheme is to obtain a set of linearly independent parity check vectors that are only dependent on $\Delta\boldsymbol{y}$ and can therefore be compared directly to the fault threshold bounds, b_i, to detect the fault. The matrix of linearly independent parity check vectors is therefore given by

$$\boldsymbol{p} = \boldsymbol{V}\boldsymbol{y} \tag{1.4}$$

where

$$\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix} \tag{1.5}$$

and V is a *(q-n)xn* dimensional projection matrix defined such that

$$\boldsymbol{V}\boldsymbol{C} = \boldsymbol{0} \tag{1.6}$$

$$\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_q - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T \tag{1.7}$$

and

$$\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_{q-n} \tag{1.8}$$

Hence, the parity check vectors are dependent only on $\Delta\boldsymbol{y}$ and are given by

$$\boldsymbol{p} = \boldsymbol{V}\Delta\boldsymbol{y} \tag{1.9}$$

Classical signal processing and control theory provides two main approaches to track the changes in the dynamics of such a system for the purposes of fault detection. These two approaches are both model based approaches and differ in that one utilizes parameter estimation techniques to model the system and the other utilizes state estimation techniques [38].The remainder of this section will give a brief introduction to these two approaches to fault detection and illustrate the limitations of these techniques when they are applied to large scale engineering plants.

1.3.1 State Estimation Techniques

As the name implies, state estimation techniques perform the fault detection tasks by using analytical models to estimate the state that the plant is in. These state estimates are then compared with the plant's actual state, obtained from the sensors, and a residual is formed. The value of the residual is then used to determine whether or not a fault condition exists and if so, the cause. There are three distinct state estimation techniques: parity checks, dedicated observer schemes, and fault detection filters.

1.3.1.1 Parity Checks

In parity check schemes [16][59], the plant is monitored by performing consistency checks on the mathematical equations describing the plant using the actual plant measurements. A fault is deemed to have occurred once a preset bound, b_i, for the given parity check has been exceeded. These parity equations can be developed from either direct redundancy relationships, the analytical relationships between the values measured at the various sensors, or temporal redundancy relationships, the analytical relationships between the plant's inputs (actuator signals) and outputs (sensor signals).

For the case of direct redundancy the outputs of the plant under consideration can be modeled by

$$\boldsymbol{y} = \boldsymbol{Cx} + \Delta\boldsymbol{y} \tag{1.3}$$

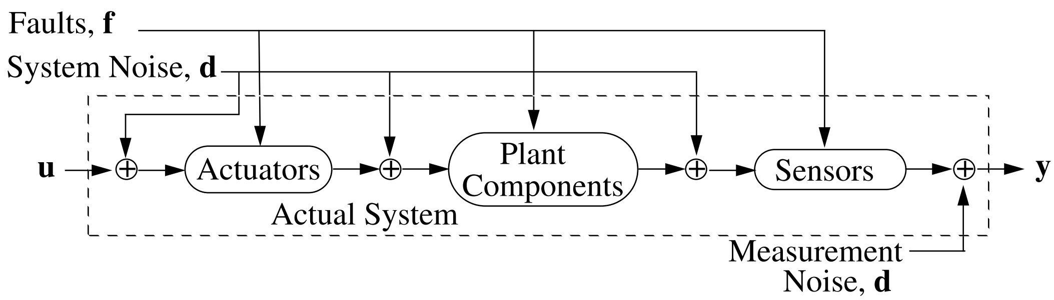


FIGURE 1-2.
System representation using classical model-based approach.

Within classical signal processing and control theory, the basic system model that is generally under consideration is the one shown in Figure 1-2. In this model, the system is operated on by the known inputs, $\boldsymbol{u}$, and its operation is observed through the available sensor outputs, $\boldsymbol{y}$. The actual system is assumed to be composed of actuators which receive the input signals $\boldsymbol{u}$, the plant components which are operated on by the actuators, and the plant sensors which provide the output signals, $\boldsymbol{y}$, which describe the plant's operational status. All the signals within the system are assumed to be affected by system noise. In addition, the actual characteristics of the actuators, components, and sensors are assumed to be unknown, and faults are assumed to occur throughout the system.

Analytically, this type of system can be described in continuous time by the state space equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{d}(t) + \boldsymbol{K}\boldsymbol{f}(t) \tag{1.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{F}\boldsymbol{d}(t) + \boldsymbol{G}\boldsymbol{f}(t) \tag{1.2}$$

where $\boldsymbol{x}(t)$ is the $nx1$ time varying state vector, $\boldsymbol{u}$ is the $px1$ known input vector, $\boldsymbol{y}$ is the $qx1$ measured output vector, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are the normal state space matrices describing the system's nominal operation, $\boldsymbol{E}\boldsymbol{d}(t)$ models the effects of unknown inputs on the actuators, $\boldsymbol{K}\boldsymbol{f}(t)$ models the occurrence of faults within the actuators and plant components, $\boldsymbol{F}\boldsymbol{d}(t)$ models the effects of unknown inputs on the sensors, and $\boldsymbol{G}\boldsymbol{f}(t)$ models the occurrence of faults within the plant's sensors.

The fault detection techniques which will be proposed in this dissertation are designed to perform well within this domain.

1.2.2 Fault, Failure and Critical Failure

For the purposes of this dissertation, the terms fault, failure, and critical failure are defined, in accordance with [53], as follows. A fault is a physical defect, imperfection, or flaw that occurs within a plant component. A failure is a plant component's or group of components' non-performance of some action that is due or expected or the performance of an action to a subnormal level of quality. A critical failure is a failure that results in a component or group of components becoming completely non-functional. Of concern, within this work, are only the set of faults which cause observable changes in the given plant's measured status data signals. A change, defect or flaw within a component that does not cause the status data to change is considered a non-observable fault. Since this work is focused on developing fault detection schemes based on the existing status data, no work is done to quantify or qualify the existence or rate of occurrence of these non-observable faults. The choice and selection of status signals is outside the purview of this work, though it is recognized that this choice does place boundaries on the types of faults that are detectable. The exact nature of these boundaries though is not explored.

1.3 Classical Model Based Fault Diagnosis Techniques

The major problem with limit checking fault detection schemes is that they do not track the changes in the system dynamics. Instead, the relationships between the patterns of event flags and the changes in the system dynamics are left for the operator to discern. An improvement on this type of fault detection system can be made by developing techniques which are capable of tracking the changes in the system dynamics and utilizing these changes to detect faulty plant components. In the literature, these techniques are referred to as analytical redundancy techniques since they utilize analytical models of the plant in question to track the change in the plant dynamics and, hence, to perform the fault detection tasks [38][69][85].

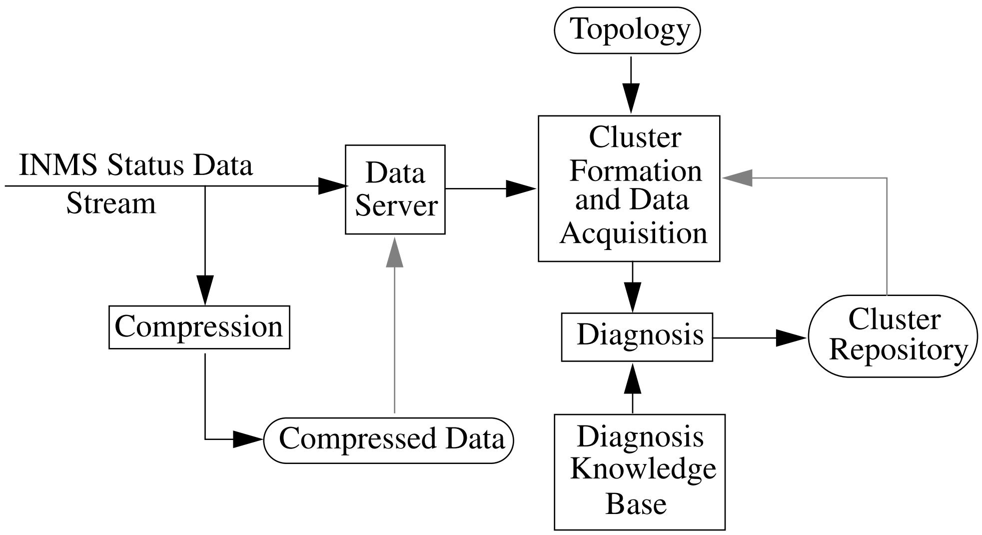


FIGURE 1-1.
Block diagram of the fault diagnosis system of [29].

1.2 Definitions

The following two sections will provided definitions of the terms large scale engineering plant, fault, failure, and critical failure. These terms are used extensively throughout this work.

1.2.1 Large Scale Engineering Plants

Within the context of this work the term large scale engineering plant is used to refer to the general class of systems which have a large number of components, whose interactions are poorly understood, and, therefore, are not amenable to analytical modelling techniques. As such, examples of these types of plants exist in a wide variety of industries. Some examples of which include:

- Petrochemical Refineries
- Pulp and Paper Mills
- Telecommunication Communication Systems
- Power Generation and Distribution systems
- Manufacturing Plants

techniques for complex, large scale engineering plants will then be discussed. Artificial intelligence techniques for dynamic systems modeling and their applications to fault detection will then be presented. Specifically, this will only include those artificial intelligence techniques suitable for building dynamic models based mainly on numerical data sets, namely fuzzy logic techniques, and feed forward and recurrent neural network techniques. Expert systems techniques will not be discussed as they are more suited to the processing of symbolic information. The assumptions and goals of this work will then be presented. The chapter will conclude by presenting a chapter by chapter outline of the remainder of the dissertation. A detailed discussion of limit checking fault detection systems and their limitations will be left until Chapter 2.

1.1 Motivation

In the work of [29], a rule-based fault diagnosis system was developed for the cable trunk amplifier domain; a block diagram of this system is shown in Figure 1-1. This system was designed to utilize the fault flags produced by Rogers' Integrated Network Management System (INMS), a threshold based fault detection and status monitoring system, as its input. From these flags, collected over a specified time window, fault clusters were then generated indicating groups of amplifiers suspected of being affected by a common fault. These clusters were then analyzed by the expert system shell, utilizing the associated knowledge base to determine a probable cause for the given fault cluster.

Analysis of this system after it was installed in the field, indicated that a large number of duplicate fault clusters were being generated and that in many cases the generated clusters did not correlate to actual fault conditions within the plant, as determined by Rogers' repair personnel. It was determined that the fault detection technique utilized by INMS was the major source of these difficulties. For this reason there was a desire to develop a more robust fault detection system than that employed by INMS. In particular the detection system was to have a low probability of generating false alarms while at the same time having a high probability of detecting real fault events. The development of such a robust detection system is the focus of this work.

of non-linearities or dynamics which may be present. The resulting systems are also designed to permit the detection of the onset of fault conditions as early as possible; ideally, prior to the critical failure of the given component(s).

Most large scale engineering plants have some form of existing fault detection system. Typically this system is in the form of a limit checking system in which the components' statuses are monitored in real-time and the resulting data are compared against preset fixed thresholds. Because the sensors and basic status monitoring are typically integrated into the plant at the component level they are difficult and usually prohibitively expensive to change. Hence, the detection techniques developed in this dissertation are designed to be retrofitted into these existing systems and therefore provide a value added service to existing status monitoring systems. Because the given plant's sensors will be used as is, a discussion of which parameters to monitor for a particular plant to optimize the fault detection capabilities of the system is considered outside the scope of this work.

Within this dissertation, the particular type of plant which will be utilized as an example large scale engineering plant is one of Rogers Cablesystems Ltd.'s cable trunk amplifier networks. This type of plant is at the core of the distribution system which transmits cable television signals from a centrally located head-end to the subscribers homes. Although the detection techniques were developed for status data specific to this type of engineering plant, the resulting detection system is generally applicable since it does not rely on any underlying assumptions about the type of dynamics, non-linearities, or fault modalities present within the cable plants. Instead, the system "learns" the behavioural model for the plant's components and utilizes these models to determine when behaviour changes occur. Therefore the work in this dissertation is generally applicable to other large scale engineering plants.

The remainder of this chapter will attempt to place the proposed systems within the context of existing fault detection technologies. This will be done by first providing the motivation for developing the proposed system. The definitions of the terms fault, failure, and critical failure will then be presented. This will be followed by a survey of the available advanced fault detection technologies starting with the classical fault detection techniques available through signal processing and control theory. The limitations of these

Chapter 1:

Introduction

1.0 Introduction

As engineering plants operate, faults occur necessitating repair and maintenance cycles to be initiated in order to keep the plant within its operational tolerances. A key to the effective long term operation of the plant, therefore, is the ability to accurately detect and locate fault conditions within the plant's components. To this end, many engineering plants, particularly large scale plants, employ some form of status monitoring to actively collect information regarding the plant's internal state. This real-time status data is then used as the basis for fault detection processes. As engineering plants grow in size and complexity, these fault detection processes become more onerous. In part, this is due to the large volumes of status data that are typically produced by such plants; but, this effect is also due to the complex interactions of the plant's components. In general, a fault within a given component will not only cause that component's status data to be affected, but it will also cause secondary effects in the status data of neighbouring components. For large scale plants, the volume of data and multiplicity of fault effects makes it infeasible to manually classify the status data. Hence, since the early 70's work has been done in the area of developing automated fault detection systems.

When accurate analytical models for engineering plants are available, the detection process is a relatively straight forward matter of analyzing the residual generated from the difference between the model's output and that of the actual plant. For large scale engineering plants, the size and complexity of the plant generally precludes the development of analytical models which are accurate throughout the plant's complete range of fault-free and faulty operation.

This work addresses the problem of how to perform fault detection within the domain of large scale engineering plants when little or no analytical information is available regarding their operation. In particular, techniques are developed which allow for the accurate black box modeling of the plant's components regardless of the particular types

Acknowledgments

I wish to thank Dr. Nikitas J. Dimopoulos for his guidance throughout the course of this work. I would also like to thank NSERC and the Canadian Cable Labs Fund for providing financial support to this research.

I also want to thank my parents, Bill and Doris, for their encouragement and support. In addition, I want to thank Nicolaos Kourounakis and Benedikt Huber for their friendship and encouragement while I was writing up. Finally, I wish to thank Wynne MacAlpine and Stephen Campbell for their friendship and support through trying times and for their tolerance of long conversations late at night. Thanks.

List of Tables

List of Figures

Table of Contents

formance in spite of the coarsely quantized status data signals which are typical of industrial status monitoring systems constructed to employ limit checking. The fault detection properties of the proposed fault detection systems were also compared to an *in situ* limit checking fault detection system for a set of real-world data obtained from an operational large scale engineering plant. This comparison showed that both of the proposed fault detection systems achieved marked improvements over traditional limit checking both in terms of their false alarm rates and their fault detection sensitivities.

Examiners:

Dr. N. J. Dimopoulos, Supervisor (Department of Electrical and Computer Engineering)

Dr. K. F. Li, Departmental Member (Department of Electrical and Computer Engineering)

Dr. F. El-Guibaly, Departmental Member (Department of Electrical and Computer Engineering)

Dr. Z. Dong, Outside Member (Department of Mechanical Engineering)

Dr. C. W. de Silva, External Examiner (Department of Mechanical Engineering, University of British Columbia)

Supervisor: Dr. Nikitas J. Dimopoulos

Abstract

In general, it is difficult to automatically detect faults within large scale engineering plants early during their onset. This is due to a number of factors including the large number of components typically present in such plants and the complex interactions of these components, which are typically poorly understood. Traditionally, fault detection within these plants has been performed through the use of status monitoring systems employing limit checking fault detection. In this approach, upper and lower bounds are placed on what is prescribed as "normal" behaviour for each of the plant's collected status data signals and fault flags are generated if and when the given status data signal exceeds either of its bounds. This approach tends to generate relatively large numbers of false alarms, due to the technique's inability to model known signal dependencies, and it also tends to produce inconsistent fault flags, in the sense that the flags do not tend to be produced throughout the "fault" event. The limit checking approach also is not particularly adept at early fault detection tasks since as long as the given status data signal remains between the upper and lower bounds any signal behaviour is deemed as acceptable. Hence, behavioural changes in the status data signals go undetected until their severity is such that either the upper or lower bounds are exceeded.

In this dissertation, two novel fault detection methodologies are proposed which are better suited to the early fault detection task than traditional limit checking. The first technique is directed at modeling of signals exhibiting unknown linear dependencies. This detection system utilizes fuzzy membership functions to model signal behaviour and through this modelling approach fault detection bounds are generated which meet a prescribed probability of false alarm rate. The second technique is directed at modelling signals exhibiting unknown non-linear, dynamic dependencies. This system utilizes recurrent neural network technology to model the signal behaviours and prescribed statistical methods are employed to determine appropriate fault detection thresholds. Both of these detection systems have been designed to be able to be retrofitted into existing industrial status monitoring system and, as such, they have been designed to achieve good modelling per-

Approaches for Early Fault Detection in Large Scale Engineering Plants

by

Stephen William Neville
B.Eng., University of Victoria, 1990
M.A.Sc., University of Victoria, 1992

A Dissertation Submitted in Partial Fulfillment of the
Requirements of the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

We accept this dissertation as conforming
to the required standard

Dr. N. J. Dimopoulos, Supervisor (Department of Electrical and Computer Engineering)

Dr. K. F. Li, Departmental Member (Department of Electrical and Computer Engineering)

Dr. F. El-Guibaly, Departmental Member (Department of Electrical and Computer Engineering)

Dr. Z. Dong, Outside Member (Department of Mechanical Engineering)

Dr. C. W. de Silva, External Examiner (Department of Mechanical Engineering, University of British Columbia)