

NASA-CR-197516

NCC2-374

IN-05-CR

33996

OVERRIDE

P. 138

UNIVERSITY OF CALIFORNIA

Los Angeles

The Accuracy of Parameter Estimation in System Identification of Noisy Aircraft Load Measurement

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Electrical Engineering

by

Jeffrey Kong

1994

(NASA-CR-197516) THE ACCURACY OF PARAMETER ESTIMATION IN SYSTEM IDENTIFICATION OF NOISY AIRCRAFT LOAD MEASUREMENT Ph.D. Thesis (California Univ.) 138 p

N95-19130

Unclas

G3/05 0033996

© Copyright by

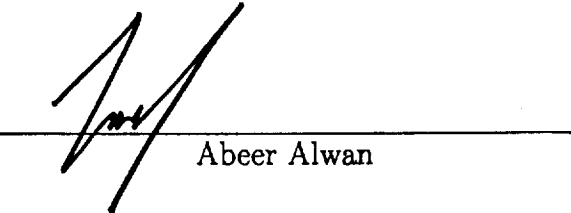
Jeffrey Kong

1994

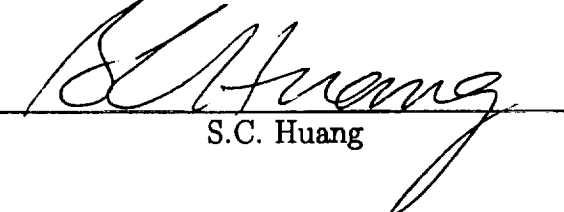
The dissertation of Jeffrey Kong is approved.




Nhan Levan



Abeer Alwan



S.C. Huang



Kung Yao, Committee Chair

University of California, Los Angeles

1994

DEDICATION

I would like to dedicate this dissertation to my late father and my mother.

Contents

DEDICATION	iii
List of Figures	ix
ACKNOWLEDGEMENTS	x
VITA	xi
ABSTRACT	xii
1 Introduction	1
2 Least-squares Estimation	6
2.1 Normal Equation Method	7
2.2 QR Factorization Method	7
2.3 Singular Value Decomposition Method	8
3 Two Equivalent Approaches to Load Measurements	9
3.1 Approach 1 - Linear Dependency of Load Values on Gage Values .	11
3.2 Approach 2 - Linear Dependency of Gage Values on Load Values .	17
3.3 Limits on the Number of Strain Gages	21

4	A Bound for Noise Perturbed Systems	24
4.1	Distance Between Subspaces	24
4.2	A Bound for the Noise Perturbed Residual	25
5	Noise Sensitivity Analysis for the Flight Stage Load Estimation	32
5.1	Effects of Errors on Load Prediction in Calibration and Flight Stage	32
5.2	Properties of Flight Stage Noise and Load Estimate Error	35
5.3	The Sensitivity Measure for Noise Perturbed Systems	37
5.4	The Trade-off between Model Accuracy and Noise Sensitivity	37
5.5	Single Coefficient Noise Sensitivity Reduction(SCNR)	38
5.5.1	Sub-optimal Single Coefficient Noise Reduction for practical applications	39
5.5.2	Performance Analysis of the SCNR procedure	41
5.6	Generalized Method for Reduction of Noise Sensitivity(GNR)	46
5.6.1	Sub-optimal Generalized method for Noise Reduction	47
5.6.2	Performance Analysis of the GNR procedure	48
5.7	Conclusion and Remarks	49
6	Total Least Squares and Correspondence Analysis	50
6.0.1	Total Least Squares(TLS)	51
6.0.2	Correspondence Analysis(CA)	51
6.1	The Equivalence of TLS and CA	56
6.1.1	Necessary and Sufficient Conditions for the Equivalence of TLS and CA in 2 dimensional space	56
6.1.2	TLS and CA Criterion for general dimensions of data matrix	59

7	Neural Networks and its applications	64
7.1	Introduction	64
7.2	An Overview of Neural Networks	65
7.2.1	Basic Neural Units or Neurons	66
7.2.2	Basic Neural Networks	67
7.2.3	Multiple Layer Perceptron (MLP)	68
7.2.4	Training Techniques for Neural Networks	69
7.3	Training by Back Propagation(BP) Technique	70
7.3.1	Notations and Definitions	70
7.3.2	Derivation	71
8	Load Estimation Problem - Linear and Non-linear Least Squares Approach	75
8.1	Linear Approach	81
8.2	Non Linear Least Squares Approach	82
8.2.1	Neural Network Approach	82
8.3	Simulation Results and Observations	84
8.4	Distributed Load Estimation Problem	90
8.5	Pattern Recognition by Neural Network	92
8.6	Results and Observation	95
9	Performance Analysis and Design Criteria of Neural Networks	98
9.0.1	Generalization and Training Accuracy	101
9.0.2	SVD, CA and Collinearity Method for Reducing Neurons in a Layer	101

10 Conclusions and Future Work	105
A Computer Codes for Neural Network Load Estimation	108
B Calibration Data from NASA F-111 Load Measurement	121
References	123

List of Figures

1.1	Parameter Estimation Process-Calibration Stage	4
1.2	Parameter Estimation Process-Calibration Stage	4
3.1	Residuals vs. number of gages used	22
5.1	Combined Noise Effects in Calibration and In-flight Stage	34
5.2	ΔS_{SCNR} vs. ΔA for NASA Data	40
5.3	Probabilty Distribution for load estimate \tilde{L}_o	42
5.4	<i>Probabilty</i> vs ΔA for NASA data	45
6.1	Correspondence Analysis of NASA HWTSS load data	54
6.2	Two data vectors in 2 dimensional space	56
6.3	Conditions for Equivalence of TLS and CA Criterion	57
6.4	Mean translated data vectors	58
7.1	A basic neuron	66
7.2	A sigmoid function	67
7.3	A Multiple Layer Perceptron Neural Network	68
8.1	Wing Structure in Calibration Stage of Load Condition $\mathcal{LC}^{(g)}$. . .	77
8.2	Gage vs. Load measurement from F-111 Data in Calibration Stage	79
8.3	The Load Estimation Problem	80
8.4	The Back Propagation Training Process	83

8.5	The Load Prediction by a Trained Neural Network	84
8.6	Relative Error vs. Test Load Samples for NN and LS approach(Load Condition 1)	86
8.7	Relative Error vs. Test Load Samples for NN and LS approach(Load Condition 2)	87
8.8	Relative Error vs. Test Load Samples for N and LS approach(Load Condition 3)	88
8.9	Single Equivalent Load and Distributed Load on a Wing Surface .	91
8.10	Classifier in Nonlinear Separable Patterns	93
8.11	Neural Network for Load Pattern Recognition	96
8.12	Neural Network for Load Pattern Recognition	97
9.1	Average Relative Error vs. number of neurons in the first hidden layer	100
9.2	Correspondence Analysis of the weight matrix W	103
9.3	Collinearly Index of the weight matrix W	104

ACKNOWLEDGMENTS

I would like to thank Professor Kung Yao for his kind support and encouragement. I would like to thank Professor Abeer Alwan, Professor Nhan Levan, and Professor S.C. Huang for serving on the committee. I would also like to express my appreciation to Mr. Karl Anderson and Mr. Steve Thornton of NASA/Dryden Flight Research Center for devoting their time on numerous discussions.

This work was partially supported by the NASA Grant under NCC 2-374.

VITA

Jeffrey Kong

██████████ Born, ██████████

June 1982 Bachelor of Science in Electrical Engineering, magna
cum laude
University of Santa Clara
Santa Clara, California

June 1983 Master of Science in Electrical Engineering
California Institute of Technology
Pasadena, California

PUBLICATIONS

K. Yao "Comparison Between Correspondence Analysis and
F. Lorenzelli Total Least Squares", *Proc. ICASSP '92, San Fran-
J. Kong cisco, USA.*

ABSTRACT OF THE DISSERTATION

**The Accuracy of Parameter Estimation in
System Identification of Noisy Aircraft Load
Measurement**

by

Jeffrey Kong

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 1994

Professor Kung Yao, Chair

This thesis focuses on the subject of the accuracy of parameter estimation and system identification techniques. Motivated by a complicated load measurement from NASA Dryden Flight Research Center, advanced system identification techniques are needed. The objective of the problem is to accurately predict the load experienced by the aircraft wing structure during flight determined from a set of calibrated load and gage response relationship. We can then model the problem as a black box input-output system identification from which the system parameter has to be estimated. Traditional LS techniques and the issues of noisy data and model accuracy are addressed. A statistical bound reflecting the change in residual is derived in order to understand the effects of the perturbations on the data. Due to the intrinsic nature of the LS problem, LS solution faces the dilemma of the trade off between model accuracy and noise sensitivity. A method of relating the two conflicting performance indices is presented,

thus allowing us to improve the noise sensitivity while at the same time confining the degradation of the model accuracy. SVD techniques for data reduction are studied and the equivalence of the Correspondence Analysis(CA) and Total Least Squares Criteria are proved. We also looked at nonlinear LS problems with NASA F-111 data set as an example. Conventional methods are neither easily applicable nor suitable for the specific load problem since the exact model of the system is unknown. Neural Network(NN) does not require prior information on the model of the system. This robustness motivated us to apply the NN techniques on our load problem. Simulation results for the NN methods used in both the single load and the "warning signal" problems are both useful and encouraging. The performance of the NN(for single load estimate) is better than the LS approach, whereas no conventional approach was tried for the "warning signals" problem. The NN design methodology is also presented. The use of SVD, CA and Collinearity Index methods are used to reduce the number of neurons in a layer.

Chapter 1

Introduction

Parameter estimation and system identification are general analytical techniques applicable to many engineering problems. The problem can generally be viewed as a black box system identification process from which input and output relationship is identified and determined. Once the system parameters are determined, it can be used to estimate the output when a new set of input data are presented to the system. As in most cases such as spectral estimation in signal processing, statistical analysis and data reduction techniques, our problem reduces to finding the solution of the linear system of equations represented by $Ax \approx B$, where A is the input matrix and B is the output matrix. The parameter x relating A and B is then determined using linear algebraic techniques. In the past, normal equations approach is mostly used until the development of newer techniques such as Singular Value Decomposition, QR factorization, etc. However, there are still drawbacks in these methods which we need to understand and analyze. There are factors affecting the accuracy of the system identification process which we need to address and analyze. Problems that appeared in this process include the

redundancies in the data matrices, the need for subset data selection, the noise in the data matrices, the selection of system parameters and the accuracy of the linear model assumption. Some of these concerns are studied in [30] [9] such as using SVD to solve the problem of data redundancies. In [3] and [9], Correspondence Analysis is used in data reduction. In [7] [10] [11], the presence of noise in both the data matrix A and output matrix B are considered using Total Least Squares. Also, we need to address the intrinsic problem of model accuracy and its robustness to noise.

In practical situations the input-output relationship of the black box system cannot be accurately modeled as a linear systems. As a result linear methods may not be adequate for these situations. Techniques such as spline approximation and nonlinear optimization are widely used in solving nonlinear least square problems. Spline approximation is relatively simple in dealing with only one or two dimension problem but becomes increasingly complex in multivariate problems. On the other hand, the use of nonlinear optimization techniques is not always straightforward and simple. Often, the assumption of the type of nonlinear function is required for the model. Recently, much attention is given to the use of Neural Network approach in applications arising from different areas of interest. From pattern recognition, function approximation, linear and nonlinear programming to singular value decomposition, Neural Network are used instead of more traditional methods. In some cases when model assumption is not easily obtainable, Neural Network will be most suitable since no model assumption is required. This advantage of NN becomes more apparent when nonlinearities is present in the data matrix A . Linear methods are no longer adequate in these situations, as a results nonlinear least square techniques must be used. Since

NN is based on search algorithms such as gradient descent algorithm, the computation effort required for the training of the NN is generally very intensive. In addition there is no definite criteria for selecting which type of NN to be used for the problem, the use of NN sometimes becomes heuristic and ad-hoc in nature. However, the ease of use, the lack of necessary model assumptions and the ability to handle wide variety of problems makes it a relatively popular method among various area of interest.

This thesis is motivated by a real aircraft system identification problem encountered at NASA Ames-Dryden Flight Research Center. The physical problem deals with the wing surface of an aircraft which is constantly experiencing different loadings during the flight. The ability to estimate these in-flight loadings are essential to the understanding and design processes of the wing structure. Strain gages are mounted on different parts of the wing which are sensitive to the loads. In order to relate the gages' outputs to the loadings on the wing surface, a pre-flight calibration procedure is performed. The calibration stage is simply a procedure to obtain the gages' outputs when a set of known wing loads are applied to the wing structure. In Figure 1.1, we apply a known load L on the aircraft structure and obtain the strain gage measurements M . From these data, we obtain a set of parameters that characterizes the structure. In Figure 1.2, during the flight measurement stage, from the in-flight gage measurements \tilde{M} , we can then estimate the in-flight loads \tilde{L} .

Some major issues in the parameter estimation problem are noises present in the calibration and the in-flight stages; the accuracy of the model (linear or non-linear) used to establish the relationship between gages and loadings; and the number of the available gages. We shall consider some of these matter. In

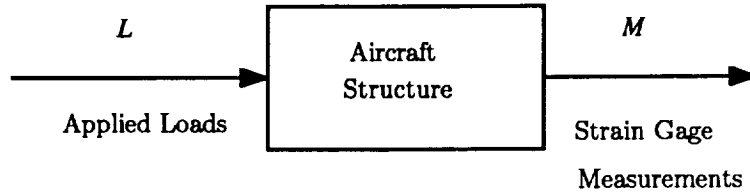


Figure 1.1: Parameter Estimation Process-Calibration Stage

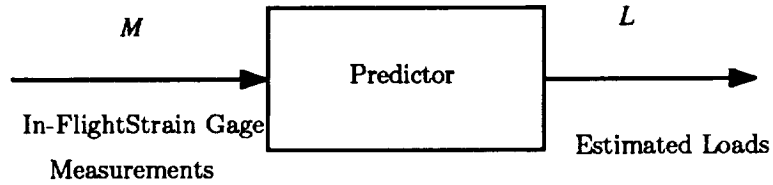


Figure 1.2: Parameter Estimation Process-Calibration Stage

chapter 2, some known analytical techniques useful for least-squares solution are summarized. In chapter 3, the equivalent approaches on the formulation of the load measurement problem are considered. In chapter 4, a bound for evaluating the effect of noise on the residual of the LS problem is derived. In chapter 5, the statistical properties of noise in both the calibration and in-flight stage are discussed. In this chapter the relationship between the noise sensitivity and the model accuracy is presented, which leads to the development of a new methodology for reducing the flight stage noise variance. It is also shown that the reduction of noise variance in the flight stage depends on the amount of deterioration of the model accuracy. In Chapter 6, the methodology of Total Least Square (TLS) [7], technical results and studies are presented with reference to the works by Golub in [7] [10] [11]. We will also discuss the method of Correspondence Analysis (CA) studied by Greenacre in [9]. In the same chapter, we will show the equivalence of TLS and CA as both methods require the minimization of the energy of perturbation error imposed on the data matrix.

Chapter 7 introduces the nonlinear least square problems together with a practical example by the NASA F-111 gage data. The inadequacy of the linear methods leads to the use of nonlinear least square or nonlinear regression techniques [2]. We also provide a background of the theory and applications of Neural Networks (NN) [20] [17] [12]. Different types of Neural Networks are also discussed and analyzed. One of the most useful type of NN is through the use of the Back Propagation training approach [27]. Since NN is shown to be useful in applications such as pattern recognition, function approximation, nonlinear programming, etc., examples and studies relating to these topics are also presented. In Chapter 8 we propose the use of NN in nonlinear least squares and pattern recognition approaches in solving the system identification problem represented by the NASA F-111 gage data. Other than estimating a single load on the wing surface, we propose the estimation of a distributed load. A distributed load estimate consists not only of the information of the single load estimate but also the individual loads experienced by different part of the wing. Pattern recognition is used in an effort of separating different loading conditions. Using NN, we showed that we could identify different loading conditions during flight stage with reasonably amount of accuracy. A simulated examples is also presented. In the same chapter, we also proposed a new method to estimate the single load estimate using NN. Since NN is capable of solving nonlinear least square problems, we could apply the techniques to the NASA F-111 data when conventional linear methods failed. Results using NN and linear methods are discussed and compared. In the last chapter, the performance of the NN is discussed and a design methodology is presented with simulation results.

Chapter 2

Least-squares Estimation

Least-squares is a general numerical method that can be used for solving a linear system of equations such that the sum of squares of the errors is minimized.

Suppose we have a system of equation

$$Ax \simeq b, \tag{2.1}$$

where A is a known $m \times n$ matrix, x is an unknown $n \times 1$ solution vector ,and b is a known $m \times 1$ vector. For overdetermined system of equations, we have $m \geq n$ and for underdetermined system, we have $m \leq n$. We also assumed all the data and solution are real-valued. The least-squares(LS) problem is equivalent to the minimization problem of

$$\min_x \|Ax - b\| = \|A\hat{x} - b\|, \tag{2.2}$$

where \hat{x} is the LS solution for equation (2.1). Three well known methods [7] for solving \hat{x} are summarized below.

2.1 Normal Equation Method

By expanding (2.2), we have

$$\begin{aligned} \|Ax - b\|^2 &= (Ax - b)'(Ax - b) = x'A'Ax - x'A'b - b'Ax + b'b \\ &= x'A'Ax - 2x'A'b + b'b. \end{aligned} \quad (2.3)$$

By taking the gradient of (2.3) with respect to x and set it to zero yields

$$2A'A\hat{x} - 2A'b = 0$$

and

$$\hat{x} = (A'A)^{-1}A'b. \quad (2.4)$$

Thus the normal equation approach to the least-squares solution is given by (2.4). This solution exists only if $m \geq n$ and A is full rank. However this method does not yield a numerically stable solution since it requires the inverse of the matrix $A'A$ whose condition number is the square of that of A . Thus this method is not useable if A is already nearly singular.

2.2 QR Factorization Method

Let A be a full rank $m \times n$ matrix that can be factored into the form of $A = QR$, where Q is a $m \times n$ matrix of orthonormal column vectors, and R is a $n \times n$ upper triangular matrix [7]. Then from (2.1), \hat{x} can be solved by the equations

$$QR\hat{x} = b. \quad (2.5)$$

$$R\hat{x} = Q'b. \quad (2.6)$$

Since R is a non-singular matrix, R^{-1} exists and (2.6) can be written as

$$\hat{x} = R^{-1}Q'b. \quad (2.7)$$

While the formal LS solution \hat{x} based on the QR method is given by (2.7), since R is upper triangular, \hat{x} in practice can be obtained by using the back substitution method once $Q'b$ is determined. In general, the QR method approach for the solution of \hat{x} is more stable than the normal equation approach.

2.3 Singular Value Decomposition Method

Let A be a $m \times n$ matrix of rank $r \leq \min(m, n)$. Then the SVD of A can be expressed as $A = U\Sigma V'$, where U is a $m \times m$ matrix with orthogonal columns (i.e., $U'U = I$); V is a $n \times n$ orthogonal matrix (i.e., $VV' = V'V = I$); and Σ is a $m \times n$ diagonal matrix = $\text{diag}[\lambda_1, \dots, \lambda_n]$, with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > \lambda_{r+1} = \dots = \lambda_n = 0$ [7]. Then from (2.1), \hat{x} can be solved by

$$\hat{x} = U'\Sigma^+Vb, \quad (2.8)$$

where $\Sigma^+ = \text{diag}[\lambda_1^{-1}, \dots, \lambda_r^{-1}, 0, \dots, 0]$. Thus the SVD solution of (2.8) is numerically stable even when A is rank deficient. As a result, the SVD approach toward the solution of \hat{x} in (2.8) allows effective reduced rank of the system of equations.

Chapter 3

Two Equivalent Approaches to Load Measurements

There are two fundamental operations in the load measurement problem on an aircraft structure. In the calibration stage as shown in Fig. 1.1, we perform a system identification by measuring the responses of the strain gages mounted on different locations of the structure from a series of known applied load at various specified load points on the structure during the calibration stage on the ground. In the flight measurement stage, by using some characterization of the system obtained in the first stage, we can predict the actual equivalent load value and location from the gage measurements during the flight. Various known successful approaches and results have been reported in the past on the load measurement problem [14] - [26]. There are two fundamental and intuitively equally justifiable linear approaches arbitrarily denoted as Approach 1 and Approach 2, that are applicable to the load measurement problem. In Approach 1, we model the load value matrix L as dependent linearly on the influence coefficient value matrix M

measured by the gages. In Approach 2, we model M as dependent linearly on L . In general these matrices are rectangular, thus it is not immediately clear that these two approaches are equivalent. Historically, all the work in [14] [26] were based on that of Approach 1. Now, we shall show that these two approaches are indeed equivalent in most cases, and can be proved by the use of the SVD method of Section 2.3. On the other hand, if we only use the more conventional and previously used normal equation method ([14] [26] also called the linear regression technique) of Section 2.1, then the limitation of this analytical technique can only show the validity of Approach 1 when the number of gages n is less or equal to the number of loads m ($m \leq n$). Through the use of the concept of "minimum energy" solution, we can show that both approaches are equivalent. There are several theoretical, practical, and computational consequences to these observations.

At the most basic level of understanding, of course, it is theoretically important to know the equivalency of these two seemingly different approaches that yield the desired result. At the practical algorithmic operational level, the inadmissibility of having the number of gages n greater than the number of applied loads m in the calibration stage in Approach 1 is generally not serious. However, there are certain conditions in which we want to consider more gages than the number of loads in the calibration stage. Conventional normal equation approach (i.e., Approach 1) is not possible since $A'A$ needed in the processing is singular.

When the data from the gages are quite linearly independent, then there is no significant numerical difference between the use of the SVD technique or the normal equation technique. However, for highly dependent data, there can be significant advantages for the SVD technique. Detailed numerical computations based on practical observed gage measurements and load values are necessary to

verify their differences. The crucial point is that in all cases, the SVD approach is always computationally more costly as well as numerically more stable. For typical dimensions encountered in the load measurement problems, the additional computational cost of the SVD approach is not of significant concern, when we perform only few LS computations. However, when we perform the LS computations repeatedly (as we shall see in Section 3.3 under the exhaustive search method), then the higher SVD computational cost may be objectionable.

3.1 Approach 1 - Linear Dependency of Load Values on Gage Values

Load Matrix

Let $L \in \mathbb{R}^{m \times 3}$ be a load matrix

$$L = [L_1, L_2, L_3], \quad (3.1)$$

where

$$L_1 = L_S = [S_1, S_2, \dots, S_m]', \quad (3.2)$$

$$L_2 = L_B = [B_1, B_2, \dots, B_m]', \text{ and} \quad (3.3)$$

$$L_3 = L_T = [T_1, \dots, T_m]'. \quad (3.4)$$

Alternatively, the shear, moment and torque of the i -th element can be expressed as

$$B_i = S_i y_i, \quad i = 1, \dots, m, \text{ and} \quad (3.5)$$

$$T_i = S_i x_i, \quad i = 1, \dots, m, \quad (3.6)$$

where (x_i, y_i) represents the relative position of the load vector.

Gage Matrix

Let $M \in \mathbb{R}^{m \times n}$ be the gage matrix which is the response of the n gages to the m loads in the calibration process, specifically let

$$M = [m_{\bullet 1}, \dots, m_{\bullet n}] = \begin{bmatrix} M_{1\bullet} \\ \vdots \\ M_{m\bullet} \end{bmatrix}, \quad (3.7)$$

where each $m_{\bullet i}$, $i = 1, \dots, n$, represents the normalized response of the i -th gages to the m loads. Let the $n \times 3$ dependency coefficient matrix b consists of

$$b = [b_{\bullet 1}, b_{\bullet 2}, b_{\bullet 3}], \quad (3.8)$$

Then the linear relationship between the load and the gage matrix can be expressed as

$$L \simeq Mb \quad (3.9)$$

$$L \simeq M[b_{\bullet 1}, b_{\bullet 2}, b_{\bullet 3}] \quad \text{or} \quad (3.10)$$

$$L_i \simeq Mb_{\bullet i} \quad , i = 1, \dots, 3. \quad (3.11)$$

For $i = 1$, $b_{\bullet 1}$ yields the dependency of $L_1 = L_S$, the shear vector to the linear combinations of the influence coefficient vectors $m_{\bullet 1}, \dots, m_{\bullet n}$ of M in (3.7). Similarly, $b_{\bullet 2}$ yields the dependency of $L_{\bullet 2} = L_B$ and M and $b_{\bullet 3}$ yields the dependency of $L_{\bullet 3} = L_T$ and M .

For the cases of $m \geq n$, normal equation and Approach 1

In the calibration process, the matrix M as well as L_1, L_2 , and L_3 are available. If we define the pseudo-inverse of $M \in \mathbb{R}^{n \times m}$ as a matrix from the normal

equation point of view, we have

$$M^+ = (M'M)^{-1}M', \quad (3.12)$$

where $m \geq n$ and all columns of M are linearly independent. Then the LS solution \hat{b} of the linear system $L \simeq Mb$ using the normal equation method is given by

$$\hat{b}_{\bullet i} = M^+L_i, \quad i = 1, 2, 3, \quad \text{and} \quad (3.13)$$

$$\hat{b} = M^+L. \quad (3.14)$$

In the flight measurement stage, we observe one $1 \times n$ dimensional gage measurement vector \tilde{M} (corresponding to the first row vector of M in (3.7)). From (3.9), the predicted 1×3 load vector \tilde{L} is given by

$$\tilde{L} = [\tilde{S}, \tilde{B}, \tilde{T}] = \tilde{M}\hat{b} = \tilde{M}M^+L = \tilde{M}(M'M)^{-1}M'L \quad (3.15)$$

$$= \tilde{M}M^+[L_{\bullet 1}, L_{\bullet 2}, L_{\bullet 3}]. \quad (3.16)$$

Then the first element of \tilde{L} yields the predicted shear \tilde{S} , the second element yields the predicted moment \tilde{B} , and the third element yields the predicted torque \tilde{T} as given by

$$\tilde{S} = \tilde{M}M^+L_{\bullet 1}, \quad (3.17)$$

$$\tilde{B} = \tilde{S}\tilde{y} = \tilde{M}M^+L_{\bullet 2}, \quad \text{and} \quad (3.18)$$

$$\tilde{T} = \tilde{S}\tilde{x} = \tilde{M}M^+L_{\bullet 3}. \quad (3.19)$$

From (3.19), we can solve for \tilde{y} and \tilde{x} as

$$\begin{aligned} \tilde{y} &= \frac{\tilde{M}M^+L_{\bullet 2}}{\tilde{M}M^+L_{\bullet 1}} \quad \text{and} \\ \tilde{x} &= \frac{\tilde{M}M^+L_{\bullet 3}}{\tilde{M}M^+L_{\bullet 1}}. \end{aligned} \quad (3.20)$$

Thus,(3.19) and (3.20) represent the predicted equivalent net shear, bending moment location, and torque location of the applied load that yielded the measured gage vector \tilde{M} using the normal equation approach.

For the cases of $m \leq n$, normal equation and Approach 1

For this case, we have an underdetermined linear system. Since $m \leq n$, we can solve the linear system $Mb = L$ exactly. However, a unique LS solution doesn't exist for the normal equation method. Therefore a "minimum energy" solution can be chosen for this case.

Let $f(b)$ be an energy function to be minimized with a constraint that $M\hat{b} = L$, we can use Lagrangian multiplier method such that

$$f_k(b_{\bullet k}) = b'_{\bullet k} b_{\bullet k} + \sum_{j=1}^m \lambda_j (m_{j\bullet} b_{\bullet k} - L_{\bullet k}) \quad (3.21)$$

$$= b'_{\bullet k} b_{\bullet k} + \lambda' [M b_{\bullet k} - L_{\bullet k}] = b'_{\bullet k} b_{\bullet k} + [b'_{\bullet k} M' - L'_{\bullet k}] \lambda, \quad (3.22)$$

where $k = 1, 2, 3$, $\lambda = [\lambda_1, \dots, \lambda_m]'$ is the vector of the lagrangian multiplier. Now setting the gradient to 0 yields

$$\nabla f_k(b_{\bullet k}) = 2b_{\bullet k} + M' \lambda = 0 \quad (3.23)$$

and

$$\hat{b}_{\bullet k} = -\frac{1}{2} M' \lambda \quad (3.24)$$

$$\lambda = -2(MM')^{-1} L_{\bullet k}, \quad k = 1, 2, 3. \quad (3.25)$$

As a result, the solution can be expressed in matrix form as

$$\hat{b} = M'(MM')^{-1}L = M^{+*}L. \quad (3.26)$$

SVD Method for Approach 1

Now consider the use of the SVD technique via Approach 1. Consider an alternate form of the SVD of the matrix M with rank $p \leq \min(m, n)$ as given by

$$M = U_M \Sigma_M V_M', \quad (3.27)$$

where

$U_M \in \mathbf{R}^{m \times p}$ is a matrix with orthogonal columns,

$V_M \in \mathbf{R}^{n \times p}$ is a matrix with orthogonal columns, and

$\Sigma_M \in \mathbf{R}^{p \times p}$ is a diagonal matrix with positive singular values (S.V.) denoted by $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$.

Let the pseudo-inverse of the matrix $M \in \mathbf{R}^{n \times m}$ be denoted by M^{++} , and from the SVD point of view we can express

$$M^{++} = V_M \Sigma_M^+ U_M', \quad (3.28)$$

where Σ_M^+ is a $n \times m$ diagonal matrix with the inverse positive singular values denoted by $1/\sigma_1 \leq 1/\sigma_2 \leq \dots \leq 1/\sigma_p > 0$. Thus by using (3.28) in (3.9), the LS solution for b satisfying (3.9) via SVD is

$$\hat{b} = M^{++}L. \quad (3.29)$$

We note, (3.29) corresponds to (3.14) and (3.26) in the calibration stage of the previously considered normal equation technique.

Then in the flight measurement stage, we have

$$\tilde{L} = \tilde{M}\hat{b} = \tilde{M}M^{++}L. \quad (3.30)$$

It is most interesting to note, that the predicted load vector in (3.30) based on the SVD technique has the same form as the predicted load vector in (3.16) based

on the normal equation technique. Indeed when $m \geq n$ (i.e., the number of loads is greater or equal to the number of gages) and when the gage measurements are quite linearly independent, the pseudo-inverse given by M^{++} in (3.29) is equal to the pseudo-inverse given by M^+ in (3.9). Thus, in those cases, either the conventional normal equation or the SVD methods will yield the same predicted load values. Of course, when the measurement values are quite linearly dependent, then the SVD approach will be better from the numerical stability point of view. As mentioned earlier, when $n > m$, the normal equation method is not applicable for Approach 1 since M^+ in (3.12) is not defined. Through the use of the concept of "minimum energy" solution, we showed that normal equation approach can be used for cases when $n \geq m$. In addition, the results of (3.28)-(3.30) under the SVD method for Approach 1 are valid in all cases including $n > m$.

3.2 Approach 2 - Linear Dependency of Gage Values on Load Values

From a physical cause and effect point of view, it is reasonable to assume that the response of the first gage to the m loads is given by

$$\begin{aligned}
 m_{\bullet 1} &= \begin{bmatrix} m_{11} \\ \vdots \\ m_{m1} \end{bmatrix} \simeq \begin{bmatrix} s_1 & s_1 y_1 & s_1 x_1 \\ \vdots & \vdots & \vdots \\ s_m & s_m y_m & s_m x_m \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \end{bmatrix} \\
 &= [L_{\bullet 1}, L_{\bullet 2}, L_{\bullet 3}] c_1 = L c_1.
 \end{aligned} \tag{3.31}$$

In (3.31), we are describing the gage measurement as a linear combination of the form

$$m_{1i} = s_i c_{11} + s_i y_i c_{12} + s_i x_i c_{13}, \tag{3.32}$$

which depends linearly on the shear, bending moment, and torque. In general, for all n gages, we have

$$M = [m_{\bullet 1}, \dots, m_{\bullet n}] \simeq L[c_1, \dots, c_n] = LC, \tag{3.33}$$

where the $3 \times n$ dependency matrix C is denoted by

$$C = [c_1, \dots, c_n].$$

In the *calibration process*, M and L are available as before. In the *flight measurement process*, we have an observed \tilde{M} given by (3.33) as

$$\tilde{M} = \tilde{L}C. \tag{3.34}$$

For the cases of $m \leq n$, normal equation and Approach 2

Let the pseudo-inverse of L be denoted by

$$L^+ = \begin{cases} (L'L)^{-1}L' & \text{for } m \geq 3 \\ L(LL')^{-1} & \text{for } m \leq 3. \end{cases} \quad (3.35)$$

By using (3.35) in (3.33), the LS solution satisfying (3.33) is

$$\hat{C} = L^+M = (L'L)^{-1}L'M. \quad (3.36)$$

In the flight stage, we have

$$\tilde{M} = \tilde{L}\hat{C} = \tilde{L}L^+M. \quad (3.37)$$

Since $m \leq n$, $(MM')^{-1}$ exists and we can multiply both sides of equation (3.37) by M' and obtain

$$\tilde{M}M' = \tilde{L}L^+(MM') \quad (3.38)$$

$$\tilde{M}M'(MM')^{-1} = \tilde{L}L^+ \quad (3.39)$$

$$\tilde{M}M'(MM')^{-1}L = \tilde{L} \underbrace{L^+L}_{I_{3 \times 3}} \quad (3.40)$$

$$\tilde{L} = \tilde{M}(MM')M'L. \quad (3.41)$$

For the case $m \geq n$, normal equation and Approach 2

Using the "minimum energy" solution of $(\tilde{L}L^+)'$ for the underdetermined system we have in the flight stage from equation (3.36) and (3.37)

$$\tilde{M} = \tilde{L}\hat{C} \quad (3.42)$$

$$\tilde{M} = (\tilde{L}L^+)M \quad (3.43)$$

$$\tilde{M}' = M'(\tilde{L}L^+)' \quad (3.44)$$

$$(\tilde{L}L^+)' = M(M'M)^{-1}\tilde{M}' \quad (3.45)$$

$$\tilde{L}L^+ = \tilde{M}[(M'M)^{-1}]'M' \quad (3.46)$$

$$\tilde{L} = \tilde{M}(M'M)^{-1}M'L, \quad (3.47)$$

where $L^+L = I_{3 \times 3}$.

SVD method for Approach 2

Now, consider solving for C in (3.33) by using the pseudo-inverse of L based on the SVD representation of L . Specifically, consider the SVD of L as given by

$$L = U_L \Sigma_L V_L', \quad (3.48)$$

where

$U_L \in \mathbf{R}^{m \times m}$ is an orthonormal matrix,

$V_L \in \mathbf{R}^{3 \times 3}$ is an orthonormal matrix, and

$\Sigma_L \in \mathbf{R}^{m \times 3}$ is a matrix of the form $diag(\sigma_{L1}, \sigma_{L2}, \sigma_{L3})$.

In particular, we note

$$\Sigma_L^+ \Sigma_L = I_3. \quad (3.49)$$

By using (3.48) in (3.34), we have

$$\tilde{M} = \tilde{L}V_L \Sigma_L^+ U_L' M. \quad (3.50)$$

Now, M can be expressed in its SVD form of

$$M = U_M \Sigma_M V_M'. \quad (3.51)$$

Then (3.50) can be written as

$$\tilde{M} = \tilde{L} \underbrace{V_L \Sigma_L^+ U_L'}_{L^+} U_M \Sigma_M V_M'. \quad (3.52)$$

For the cases of $m \leq n$, SVD and Approach 2

For $m \leq n$, $\Sigma_M^+ \in \mathbf{R}^{m \times m}$, $U_M \in \mathbf{R}^{m \times m}$, and $V_M \in \mathbf{R}^{n \times m}$. Then $V_M V_M' = I_{m \times m}$ and $U_M U_M' = U_M' U_M = I_{m \times m}$ since U_M is orthonormal.

Then equation (3.52) becomes

$$\tilde{M}(V_M \Sigma_M^+ U_M') = \tilde{L} L^+ \quad (3.53)$$

$$\tilde{L} \underbrace{L^+ L}_{I_{3 \times 3}} = \tilde{M} \underbrace{(V_M \Sigma_M^+ U_M')}_M L \quad (3.54)$$

$$\tilde{L} = \tilde{M} M^{++} L. \quad (3.55)$$

Since equation (3.55) in Approach 2 is the same as that of Approach 1, we showed that the two approaches is equivalent in this case.

For the cases of $m \geq n$, SVD and Approach 2

For $m \geq n$ we look at the expression of

$$M \approx LC.$$

The SVD solution $\hat{C} = U_C \Sigma_C V_C'$ of the above LS problem is

$$\hat{C} = L^+ M.$$

Since $C \in \mathbf{R}^{3 \times n}$ and $\hat{C} \hat{C}^+ = I_{3 \times 3}$, we can write

$$\underbrace{\hat{C} \hat{C}^+}_{I_{3 \times 3}} = L^+ M \hat{C}^+ \quad (3.56)$$

$$\hat{C}^+ = M^{++} L. \quad (3.57)$$

For the in-flight stage we have

$$\tilde{M} = \tilde{L}\hat{C} \quad (3.58)$$

$$\tilde{M} \underbrace{\hat{C}^+}_{M^{++L}} = \tilde{L} \underbrace{\hat{C}\hat{C}^+}_{I_{3 \times 3}} \quad (3.59)$$

$$\tilde{L} = \tilde{M}M^{++L}. \quad (3.60)$$

Since equation (3.60) in Approach 2 is the same as that of Approach 1, hence we showed that the two approaches yield the same solution via SVD.

3.3 Limits on the Number of Strain Gages

Due to limitations of in-flight telemetry channels, the number of gages available for the measurement of loads are less than that during the Calibration Stage. Thus we need to find the best possible combination of smaller number of gages for load estimation. Two heuristic approaches have been used to eliminate gages in such a way that the resulting increase in estimation error is not very significant [26].

T Value Method

For each gage, a T value is calculated as

$$T_i = \hat{b}_i / \epsilon_i, \quad (3.61)$$

where \hat{b}_i is the i -th element of \hat{b} and ϵ_i is the standard deviation error associated with \hat{b}_i [23]. Then the gage with the smallest T value is eliminated. Repeat the process until the desired number of gages is left.

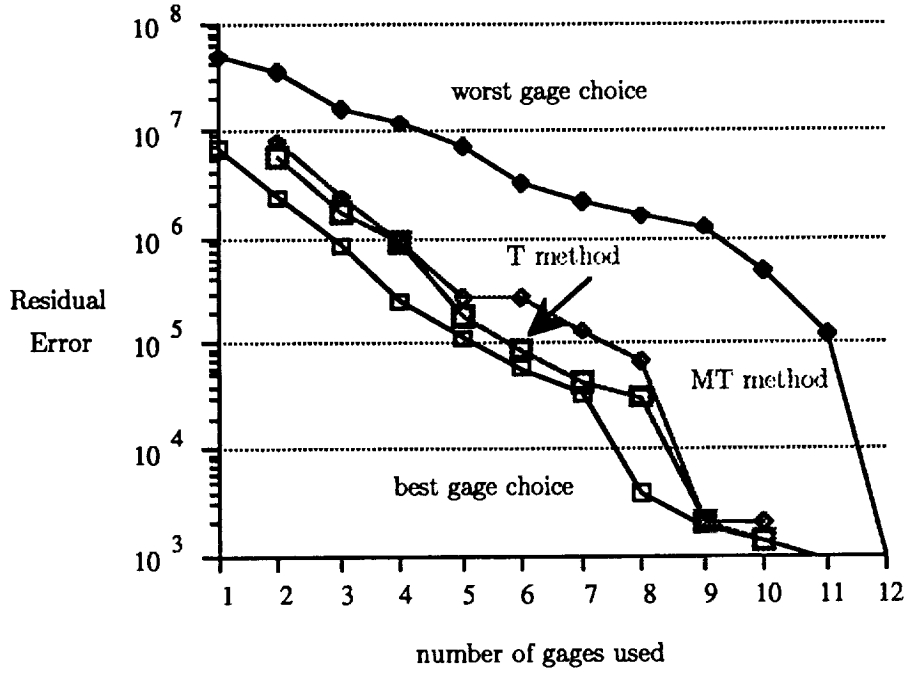


Figure 3.1: Residuals vs. number of gages used

MT Value Method

Again, for each gage, a MT value is computed as

$$MT_i = \frac{\mu_i \lambda_i \hat{b}_i}{\gamma_i \epsilon_i}, \quad (3.62)$$

where μ_i is the mean absolute response of i -th gage, λ_i is the correlation coefficient from i -th gage to all load conditions, and γ_i is the correlation coefficient from i -th gage to all other gages. The elimination procedure based on the MT value is the same as that based on the T value.

Exhaustive Search Method

This method minimizes the residual by exhaustively computing all the combinations of the desired number of gages. The combination which results in the lowest residual will be chosen as the best gage combination for load estimation.

A simulation using NASA HWTSS data is carried out to compare the difference in performance for different subset selection methods. In fig. 3.1, we observed that although MT and T value methods are heuristic in nature, it provides a more accurate solution than the solution yielded by the "worst gage" choice. Indeed, these methods perform quite close to that of the "best gage" choice method. Since the choice of which gages to be eliminated depends on the integrity of the gage data, the reduced gage choice set can be very sensitive to noisy data. Thus a more robust procedure is preferred when data matrix are corrupted with noise. Other methods that use collinearity index in trying to eliminate redundant gages are also studied in works by [30].

Chapter 4

A Bound for Noise Perturbed Systems

The performance of the estimation using least squares solution depends on the accuracy of the underlying model and the integrity of the data. This chapter will illustrate how a corrupted data matrix affects the accuracy of the model as reflected by the change in the residual of the LS problem. The effect of noise on the measurement matrix M will be shown by deriving a bound for the increase in residual as a function of the noise. The following preliminary materials are necessary for the derivation of the bound [7].

4.1 Distance Between Subspaces

Consider an over determined system where $m > n$. In specific we let $A \in \mathbf{R}^{m \times p}$ and $B \in \mathbf{R}^{m \times p}$ be two given matrices. Let the domain of A and B be \mathbf{R}^p , then let $\mathcal{F} \in \mathbf{R}^m$ and $\mathcal{G} \in \mathbf{R}^m$ denote the ranges of A and B respectively. Then we

can define

$$\begin{aligned} \cos \theta_k &= \max_{u \in \mathcal{F}} \max_{v \in \mathcal{G}} u'v = u'_k v_k, \quad k = 1, \dots, p & (4.1) \\ \|u\|_2 &= \|v\|_2 = 1 \\ u'_i &= v'_i = 0, \quad i = 1, \dots, k-1, \end{aligned}$$

where u_0 and v_0 are null vectors (i.e., for $k = 1$, the constraints $u'_0 = v'_0 = 0$ vanish). Then the set $\theta_k, k = 1, \dots, q$ satisfies $0 \leq \theta_1 \leq \theta_2 \leq \dots \leq \theta_q \leq \pi/2$. Furthermore, u_i, v_i and θ_i are called the principal vectors and the principal angles of the subspace pair of \mathcal{F}, \mathcal{G} [7]. In particular, θ_p is the maximum angle between two basis vectors in subspace \mathcal{F} and \mathcal{G} . Furthermore, the distance between \mathcal{F} and \mathcal{G} reduces to

$$\|Q_{\mathcal{F}}Q'_{\mathcal{F}} - Q_{\mathcal{G}}Q'_{\mathcal{G}}\| = \sqrt{1 - \cos^2 \theta_p} = \sin \theta_p, \quad (4.2)$$

where the columns of $Q_{\mathcal{F}}$ and $Q_{\mathcal{G}}$ are the orthonormal bases for \mathcal{F} and \mathcal{G} respectively. It can be shown [7] that

$$\cos \theta_k = \sigma_k(Q'_{\mathcal{F}}Q_{\mathcal{G}}), \quad k = 1, \dots, p. \quad (4.3)$$

where $\sigma_k(Q'_{\mathcal{F}}Q_{\mathcal{G}})$ is the k th largest singular value of the matrix $Q'_{\mathcal{F}}Q_{\mathcal{G}}$.

4.2 A Bound for the Noise Perturbed Residual

Let $[u_1, \dots, u_p]$ be a set of orthonormal basis vectors in \mathbb{R}^m that spans the subspace M of the strain gage measurements under the Calibration Stage and $[v_1, \dots, v_p]$

be a set of orthonormal basis vectors in \mathbf{R}^m that spans the subspace \bar{M} of the gage measurement with noise. Specifically, the noise E is defined by

$$\bar{M} \equiv M + E. \quad (4.4)$$

Furthermore, we denote $Q_M = [u_1, \dots, u_p]$ and $Q_{\bar{M}} = [v_1, \dots, v_p]$. Then $Q_M Q'_M$ is a projection operator onto M and $Q_{\bar{M}} Q'_{\bar{M}}$ is a projection operator onto \bar{M} .

Under ideal (noise-free) condition, the ideal LS solution \hat{b} associated with the measurement matrix M and the calibration load L_{cab} is obtained from

$$\min_b \|Mb - L_{cab}\|^2 = \|M\hat{b} - L_{cab}\|^2. \quad (4.5)$$

Then \hat{L} can be defined by

$$\hat{L} \equiv M\hat{b} = Q_M Q'_M L_{cab}. \quad (4.6)$$

Similarly, under noisy condition, the LS solution \bar{b} associated with the noisy measurement matrix \bar{M} and the calibration load L_{cab} is obtained from

$$\min_b \|\bar{M}b - L_{cab}\|^2 = \|\bar{M}\bar{b} - L_{cab}\|^2. \quad (4.7)$$

Similarly, \bar{L} can be defined by

$$\bar{L} \equiv \bar{M}\bar{b} = Q_{\bar{M}} Q'_{\bar{M}} L_{cab}. \quad (4.8)$$

Thus the difference between \hat{L} and \bar{L} becomes

$$\hat{L} - \bar{L} = (Q_M Q'_M - Q_{\bar{M}} Q'_{\bar{M}}) L_{cab} \quad (4.9)$$

and its norm squared is bounded by

$$\|\hat{L} - \bar{L}\|^2 = \|(Q_M Q'_M - Q_{\bar{M}} Q'_{\bar{M}}) L_{cab}\|^2 \leq \|L_{cab}\|^2 \sin^2 \theta_p, \quad (4.10)$$

where $\|(Q_M Q'_M - Q_{\bar{M}} Q'_{\bar{M}})\|^2 = \sin^2 \theta_p = 1 - \sigma_p^2(Q'_M Q_{\bar{M}})$ from (4.2).

For simplicity, denote

$$\sigma_p(Q'_M Q_{\bar{M}}) = \sigma_p^* . \quad (4.11)$$

Also denote the noise-free and noisy residuals by

$$\begin{aligned} \hat{\epsilon} &= L_{cab} - \hat{L} \quad \text{and} \\ \bar{\epsilon} &= L_{cab} - \bar{L}. \end{aligned} \quad (4.12)$$

Then (4.9) becomes

$$\hat{L} - \bar{L} = (L_{cab} - \bar{L}) - (L_{cab} - \hat{L}) = \bar{\epsilon} - \hat{\epsilon} = (Q_M Q'_M - Q_{\bar{M}} Q'_{\bar{M}}) L_{cab}. \quad (4.13)$$

By applying Triangle Inequality, we obtain

$$\|\bar{\epsilon}\|^2 \leq \|\hat{\epsilon}\|^2 + \|L_{cab}\|^2 \sin^2 \theta_p \quad (4.14)$$

or

$$\|\bar{\epsilon}\|^2 - \|\hat{\epsilon}\|^2 \leq \|L_{cab}\|^2 \sin^2 \theta_p = \|L_{cab}\|^2 (1 - \sigma_p^{*2}). \quad (4.15)$$

Thus (4.15) yields a bound on the noise perturbed residual.

The bound in (4.15) can further be expressed in a more useful form if σ_p^* can be expressed in terms of the perturbed matrix \bar{M} and the noise matrix E . As a result, the remaining section will be dedicated to this objective.

Suppose the matrix M and \bar{M} have QR factorizations as

$$M = Q_M R_M \quad (4.16)$$

and

$$\bar{M} = Q_{\bar{M}} R_{\bar{M}}. \quad (4.17)$$

From (4.7) we have $\bar{M} = M + E$, after applying QR factorization to M and \bar{M} , we obtain

$$Q_{\bar{M}}R_{\bar{M}} = Q_M R_M + E \quad (4.18)$$

$$\begin{aligned} Q_{\bar{M}} &= Q_M R_M R_{\bar{M}}^{-1} + E R_{\bar{M}}^{-1} \\ &= Q_M + Q_M R_M R_{\bar{M}}^{-1} (I - R_{\bar{M}} R_M^{-1}) + E R_{\bar{M}}^{-1}. \end{aligned}$$

If we denote

$$E_1 = Q_M R_M R_{\bar{M}}^{-1} (I - R_{\bar{M}} R_M^{-1}) + E R_{\bar{M}}^{-1}, \quad (4.19)$$

then (4.19) becomes

$$Q_{\bar{M}} = Q_M + E_1 \quad (4.20)$$

$$Q'_M Q_{\bar{M}} = Q'_M (Q_M + E_1) = Q'_M Q_M + Q'_M E_1 = I + Q'_M E_1.$$

Furthermore if we denote

$$E_2 \equiv Q'_M E_1, \quad (4.21)$$

then (4.21) can be expressed as

$$Q'_M Q_{\bar{M}} = I + E_2. \quad (4.22)$$

From properties of the theory of perturbed singular value (p.284 of [7]), we have

$$|\sigma_k(D + F) - \sigma_k(D)| \leq \sigma_1(F) = \|F\|_2, \quad \text{for } k = 1, \dots, p. \quad (4.23)$$

Combining (4.21) and (4.23), we have

$$|\sigma_p(Q'_M Q_{\bar{M}}) - \sigma_p(I)| = |\sigma_p(I + E_2) - \sigma_p(I)| \leq \sigma_1(E_2) = \|E_2\|. \quad (4.24)$$

$$|\sigma_p(Q'_M Q_{\bar{M}}) - 1| \leq \|E_2\|$$

$$|\sigma_p^* - 1| \leq \|E_2\|$$

$$(1 - \sigma_p^*) \leq \|E_2\|, \quad \text{for } \sigma_p^* \leq 1$$

$$\sigma_p^* \geq 1 - \|E_2\|, \quad \text{for } \sigma_p^* \leq 1.$$

From (4.21), we have

$$\begin{aligned}
E_2 &= Q'_M Q_M R_M R_M^{-1} (I - R_M R_M^{-1}) + Q'_M E R_M^{-1} \\
&= R_M R_M^{-1} (I - R_M R_M^{-1}) + Q'_M E R_M^{-1} \\
&= R_M R_M^{-1} - I + Q'_M E R_M^{-1}.
\end{aligned} \tag{4.25}$$

Taking the norm squared of E_2 and invoking Triangular Inequality, we obtain

$$\begin{aligned}
\|E_2\|^2 &= \|R_M R_M^{-1} - I + Q'_M E R_M^{-1}\|^2 \\
&\leq \|R_M R_M^{-1}\|^2 + 1 + \|Q'_M E R_M^{-1}\|^2 \\
&= \|R_M R_M^{-1}\|^2 + 1 + \|E R_M^{-1}\|^2.
\end{aligned} \tag{4.26}$$

If we apply QR factorization to M and \bar{M} in (4.19), we obtain

$$\begin{aligned}
M &= \bar{M} - E \\
Q_M R_M &= Q_{\bar{M}} R_{\bar{M}} - E \\
Q_M R_M R_M^{-1} &= Q_{\bar{M}} R_{\bar{M}} R_{\bar{M}}^{-1} - E R_M^{-1} \\
\|R_M R_M^{-1}\|^2 &= \|Q'_M Q_{\bar{M}} - Q'_M E R_M^{-1}\|^2 \\
&\leq \|Q'_M Q_{\bar{M}}\|^2 + \|E R_M^{-1}\|^2 \leq 1 + \|E R_M^{-1}\|^2
\end{aligned} \tag{4.27}$$

since $\|Q'_M\| = \|Q_{\bar{M}}\| = 1$ and $\|Q'_M Q_{\bar{M}}\|^2 \leq \|Q'_M\|^2 \|Q_{\bar{M}}\|^2 = 1$.

Now $R_{\bar{M}}$ is an upper triangular matrix, by defining μ we have

$$\mu \equiv \|R_{\bar{M}}^{-1}\|^2 \tag{4.28}$$

$$= \|\text{diag}(1/r_1, \dots, 1/r_p)\|^2 \tag{4.29}$$

$$= \max_{i=1, \dots, p} 1/r_i^2, \tag{4.30}$$

where r_i for $i = 1, \dots, p$ are the diagonal element of the matrix $R_{\bar{M}}$.

Then combining (4.27) and (4.28), we have

$$\|E_2\|^2 \leq 1 + \|ER_{\bar{M}}^{-1}\|^2 + 1 + \|ER_{\bar{M}}^{-1}\|^2 \quad (4.31)$$

$$\begin{aligned} &= 2(1 + \|ER_{\bar{M}}^{-1}\|^2) \\ &\leq 2(1 + \mu\|E\|^2). \end{aligned} \quad (4.32)$$

$$(4.33)$$

In summary, we have the following results

$$\|\bar{\epsilon}\|^2 - \|\hat{\epsilon}\|^2 \leq \|L_{cab}\|^2(1 - \sigma_{p^*}^2) \quad (4.34)$$

$$\sigma_{p^*} \geq 1 - \|E_2\|, \quad \text{for } \sigma_{p^*} \leq 1 \quad (4.35)$$

$$\|E_2\| \leq \sqrt{2(1 + \mu\|E\|^2)}. \quad (4.36)$$

$$(4.37)$$

As a result, the bound can be expressed as

$$\|\bar{\epsilon}\|^2 - \|\hat{\epsilon}\|^2 \leq \|L_{cab}\|^2 \{1 - [1 - \sqrt{2(1 + \mu\|E\|_2^2)}]^2\}. \quad (4.38)$$

Thus, we have derived a bound for the change in residual due to the error matrix E . It is interesting to see that the bound depends on the value of μ (which is the largest value of the diagonal elements of matrix $R_{\bar{M}}$ found from the QR factorization of the noisy data matrix \bar{M}) and the largest singular value of the noise matrix E .

Now since $\|\bullet\|_2 \leq \|\bullet\|_F$, equation (4.38) can be expressed as

$$\|\bar{\epsilon}\|^2 - \|\hat{\epsilon}\|^2 \leq \|L_{cab}\|^2 \{1 - [1 - \sqrt{2(1 + \mu\|E\|_F^2)}]^2\}. \quad (4.39)$$

Thus we have related the norm of the noisy M residual $\bar{\epsilon}$ to the norm of the

noiseless M residual $\hat{\epsilon}$ and the norm of the matrix error E . This bound provides an analytical relation between the norm of the noiseless and noisy residuals (corresponding to M) thereby allowing us to estimate the norm of the noise in the data matrix when the other two terms in (4.39) are determined or estimated. Although it is only an analytical bound whereas the actual behavior of the system is not accurately determined, it allows us to have an understanding of how these quantities are related.

Chapter 5

Noise Sensitivity Analysis for the Flight Stage Load Estimation

5.1 Effects of Errors on Load Prediction in Calibration and Flight Stage

During the calibration process, the gage measurement observed can be represented as a matrix of

$$\tilde{M}_c \equiv M_c + E_c, \quad (5.1)$$

where $M_c \in \mathbf{R}^{m \times n}$ is the true gage measurements and $E_c \in \mathbf{R}^{m \times n}$ is a matrix with its elements as the error associated with the gage measurement. In order to estimate the flight load, the estimation parameter derived from the noisy

calibration matrix M_c can be written as

$$\tilde{b} \equiv \hat{b} + b_E = (M_c + E_c)^+ L_C, \quad (5.2)$$

where $\hat{b} \in \mathbf{R}^n$ is the noiseless estimation parameter, b_E is the estimation parameter error and $L_C \in \mathbf{R}^m$ is the calibration load vector. During the flight stage, the observed noisy output gage measurement can be expressed as a matrix of

$$\tilde{M}_o = M_o + E_o, \quad (5.3)$$

where $E_o \in \mathbf{R}^n$ is a matrix with its elements as the error associated with the output gage measurement defined as $M_o \in \mathbf{R}^n$.

Thus the final noisy estimated load taking into consideration of the calibration and flight stage can be expressed as

$$\tilde{M}_o' \tilde{b} = (M_o + E_o)'(\hat{b} + b_E). \quad (5.4)$$

The resulting estimated load can also be expressed as

$$\begin{aligned} \tilde{L}_o &= (M_o + E_o)'(\hat{b} + b_E) = M_o' \hat{b} + M_o' b_E + E_o' \hat{b} + E_o' b_E \\ &= \hat{L}_o + M_o' b_E + E_o' \hat{b} + E_o' b_E \\ &\equiv \hat{L}_o + \epsilon_{OC}, \end{aligned} \quad (5.5)$$

where \hat{L}_o is the noiseless load estimate and is the estimate that we wish to obtain. Similarly ϵ_{OC} is the corrupting error term which reduces the accuracy of the load estimate.

If we examine the variance of ϵ_{OC} , we could gain some insight on how to reduce the effect of this noise on the \hat{L} . The variance of ϵ_{OC} can be expressed as

$$\text{var}\{\epsilon_{OC}\} = \text{var}\{M_o' b_E + E_o' \hat{b} + E_o' b_E\}. \quad (5.6)$$

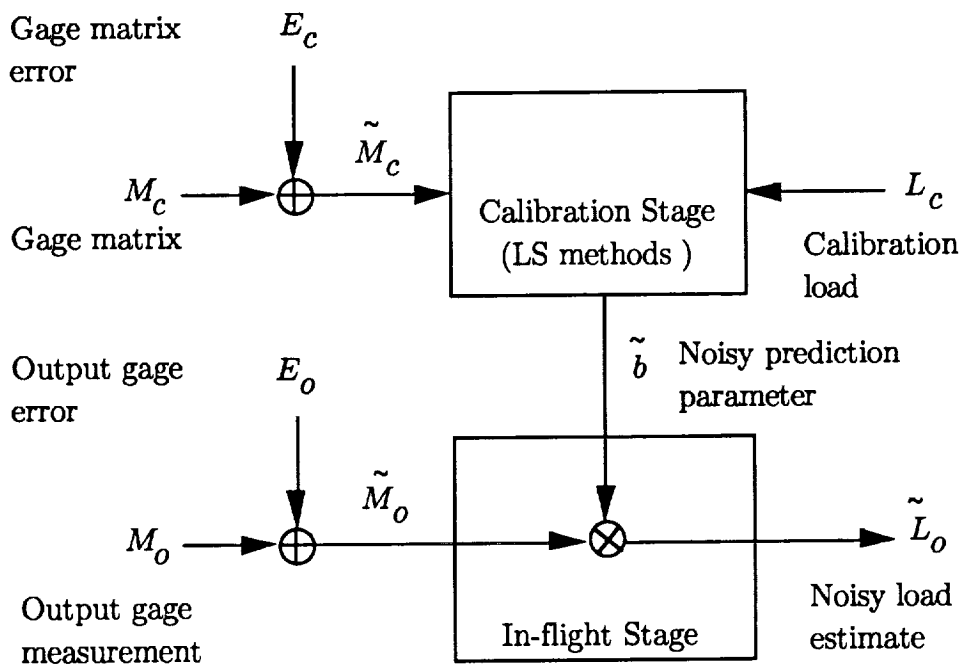


Figure 5.1: Combined Noise Effects in Calibration and In-flight Stage

If we assume that the output error E_o and b_E are uncorrelated, which is reasonable since input and output noise are usually uncorrelated, then

$$\text{var}\{E_o' b_E\} = 0. \quad (5.7)$$

As a result, (5.6) can be written as

$$\text{var}\{\epsilon_{OC}\} = \text{var}\{M_o' b_E\} + \text{var}\{E_o' \hat{b}\}. \quad (5.8)$$

(5.6) indicated that if we want to minimize the $\text{var}\{\epsilon_{OC}\}$, we can minimize the first and second term in (5.8) independently. In other words, we could reduce the combined effects by independently reducing the noise variance during the calibration and flight stage.

5.2 Properties of Flight Stage Noise and Load Estimate Error

In the last section, we have shown that the overall noise variance can be reduced by independently reducing the noise in the calibration and flight stage. For simplicity, we now assume the noise in the calibration is zero (i.e. $E_c = 0$). Then we will study the effects of noise on the load estimation accuracy during the flight stage. As mentioned before, the calibration load L_c and the gage measurement M are related by the system of linear equations,

$$L_c \simeq M_c b. \quad (5.9)$$

The estimating parameter $\hat{b} = M_c^+ L_c$ will then be used to predict the amount of load during the flight stage. The flight stage noise, which can be modeled

as an additive noise $E_o = [e_1, \dots, e_n]'$ as in (5.3) on the gage readings $M_o = [m_1, \dots, m_n]$, is generally present in the measurement system. As a result, the noisy flight stage load estimate \tilde{L}_o can be expressed as

$$\tilde{L}_o \equiv \tilde{M}_o \hat{b} = (M_o + E_o) \hat{b} \equiv \hat{L}_o + \epsilon_o, \quad (5.10)$$

where \hat{L}_o is the noiseless load estimate and $\epsilon_o = E_o \hat{b}$ is the excessive load estimate error due to the noise E_o .

In order to study the corrupting effects of the load estimate error ϵ_o , we need to examine some of its statistical properties.

If we assume that the mean of the flight stage noise E_o is zero, the expected value of ϵ_o can be expressed as

$$E\{\epsilon_o\} = E\{E_o \hat{b}\} = 0. \quad (5.11)$$

Then the variance of ϵ_o is

$$\begin{aligned} \text{Var}\{\epsilon_o\} &= E\{(\epsilon_o - E\{\epsilon_o\})^2\} = E\{\epsilon_o^2\} = E\{(E_o \hat{b})^2\} \\ &= E\{\hat{b}' E_o E_o' \hat{b}\} = \hat{b}' E\{E_o E_o'\} \hat{b} = \hat{b}' \Sigma_{EE} \hat{b}, \end{aligned} \quad (5.12)$$

where $\Sigma_{EE} \equiv E\{E_o E_o'\} = E \left\{ \begin{bmatrix} e_1 e_1 & \dots & e_1 e_n \\ \vdots & \ddots & \vdots \\ e_n e_1 & \dots & e_n e_n \end{bmatrix} \right\}.$

It is often desirable to reduce the variance of ϵ_o since a small variance suggests that the corrupting effect of noise on the load estimate will be less severe in the flight stage. However the variance of ϵ_o depends only on the amount of noise in the flight stage and the estimator \hat{b} obtained from the calibration process, it is impossible to reduce the variance of ϵ_o without making changes on the vector

\hat{b} . Nevertheless, the accuracy of the predicted load will also be affected when changes are made on \hat{b} . In other words, the reduction of the noise sensitivity of the estimate is at the expense of the model accuracy. Therefore, it is important to optimize the amount of tradeoff between noise sensitivity and model accuracy when noise sensitivity reduction is absolutely necessary. The following sections will examine two possible solutions to this problem.

5.3 The Sensitivity Measure for Noise Perturbed Systems

Since the variance of ϵ_o is an indicator for the intensity of the noise, it can be interpreted as an intensity index of noise sensitivity. If white gaussian noise is a reasonable assumption (i.e $var\{\epsilon_o\} = \hat{b}'\hat{b}\sigma_{\epsilon_o}^2 = \|\hat{b}\|^2\sigma_{\epsilon_o}^2$), the sensitivity index can be defined as

$$S_{\epsilon_o} = \frac{\sigma_{\epsilon_o}^2}{\sigma_{E_o}^2} = \|\hat{b}\|^2, \quad (5.13)$$

where $\sigma_{E_o}^2$ is the variance of the flight stage noise E_o and $\sigma_{\epsilon_o}^2$ is the variance of ϵ_o .

5.4 The Trade-off between Model Accuracy and Noise Sensitivity

In order to examine the effects of the reduction in noise sensitivity on the model accuracy, we need to look at the noisy load estimate from (5.10) such as

$$\tilde{L}_o = \tilde{M}_o' \hat{b} = (M_o + E_o)' \hat{b} = \hat{L}_o + \epsilon_o. \quad (5.14)$$

5.5 Single Coefficient Noise Sensitivity Reduction(SCNR)

In theory, we would like to have a small valued noise sensitivity index. As a result, the reduction of $S_{\epsilon_o} = \|\hat{b}\|^2$ in (5.13) requires that the value of the norm of the estimator vector $\hat{b} = [\hat{b}_1, \dots, \hat{b}_n]'$ to be reduced. It can be achieved by changing the values of one or more elements of the vector \hat{b} . The SCNR method discussed in this section only changes one element of \hat{b}_i such that the new estimator vector becomes

$$\hat{b}_{SCNR} = [\hat{b}_1, \dots, c_i \hat{b}_i, \dots, \hat{b}_n]', \quad (5.15)$$

where $c_i < 1$. Then the decrease of the noise sensitivity ΔS can be defined as

$$\Delta S \equiv \|\hat{b}\|^2 - \|\hat{b}_{SCNR}\|^2 = (1 - c_i^2)\hat{b}_i^2 \geq 0. \quad (5.16)$$

Now, we define the noiseless new load estimate (when the output stage noise $E_o = 0$) as

$$\hat{L}_{o,SCNR} \equiv M_o' \hat{b}_{SCNR} = m_1 \hat{b}_1 + \dots + m_i c_i \hat{b}_i + \dots + m_n \hat{b}_n \quad (5.17)$$

and the new load estimate obtained from the noisy measurement \tilde{m}_o as

$$\tilde{L}_{o,SCNR} \equiv \tilde{M}_o' \hat{b}_{SCNR} = \tilde{m}_1 \hat{b}_1 + \dots + \tilde{m}_i c_i \hat{b}_i + \dots + \tilde{m}_n \hat{b}_n. \quad (5.18)$$

The change in load prediction accuracy ΔA with respect to the noiseless load estimate \hat{L}_o can then be expressed as

$$\Delta A \equiv \tilde{L}_o - \tilde{L}_{o,SCNR} = (1 - c_i)\hat{b}_i m_i. \quad (5.19)$$

Combining (5.16) and (5.19), we have

$$\Delta A = (\hat{b}_i - \sqrt{\hat{b}_i^2 - \Delta S})m_i$$

and

$$\Delta S = \hat{b}_i^2 - \left(\frac{\Delta A}{m_i} - \hat{b}_i\right)^2 = \Delta A \left(2\frac{\hat{b}_i}{m_i} - \frac{\Delta A}{m_i^2}\right). \quad (5.20)$$

From (5.20), if we fix the change in model accuracy ΔA , the decrease in the noise sensitivity ΔS can be maximized by scaling the i -th gage prediction vector \hat{b} to $c_i \hat{b}_i$ such that $\hat{b}_i^2 - \left(\frac{\Delta A}{m_i} - \hat{b}_i\right)^2$ in (5.20) is the largest among gages $i = 1, \dots, n$.

The maximum ΔS for a fixed $\Delta A = \delta A$ attained can then be expressed as

$$\Delta S_{SCNR} = \max_{0 < i < n} \left\{ \hat{b}_i^2 - \left(\frac{\Delta A}{m_i} - \hat{b}_i\right)^2 \right\}. \quad (5.21)$$

$$\Delta A = \delta A$$

Figure 5.2 below shows a plot of the normalized ΔS_{SCNR} versus ΔA for the NASA Wing data, where the definitions of

$$\Delta S_{SCNR} = \frac{\Delta S_{SCNR}}{\|\hat{b}\|} \quad \text{and} \quad (5.22)$$

$$\Delta A = \frac{\Delta A}{\hat{L}_o}. \quad (5.23)$$

It indicates the amount of improvement in noise sensitivity ΔS_{SCNR} at a given level of accuracy deteriorating ΔA .

5.5.1 Sub-optimal Single Coefficient Noise Reduction for practical applications

The SCNR described in the previous section requires the knowledge of m_i in order to maximize ΔS with respect to all gages. However, only $\tilde{m}_i = m_i + \epsilon_i$ is observable at the flight stage, the exact value of m_i is not known. As a result, we can only use \tilde{m}_i in maximizing ΔS , such that (5.22) becomes

$$\Delta S_{SCNR} = \max_{0 < i < n} \Delta S = \max_{0 < i < n} \hat{b}_i^2 - \left(\frac{\Delta A}{\tilde{m}_i} - \hat{b}_i\right)^2. \quad (5.24)$$

$$\Delta A = \delta A$$

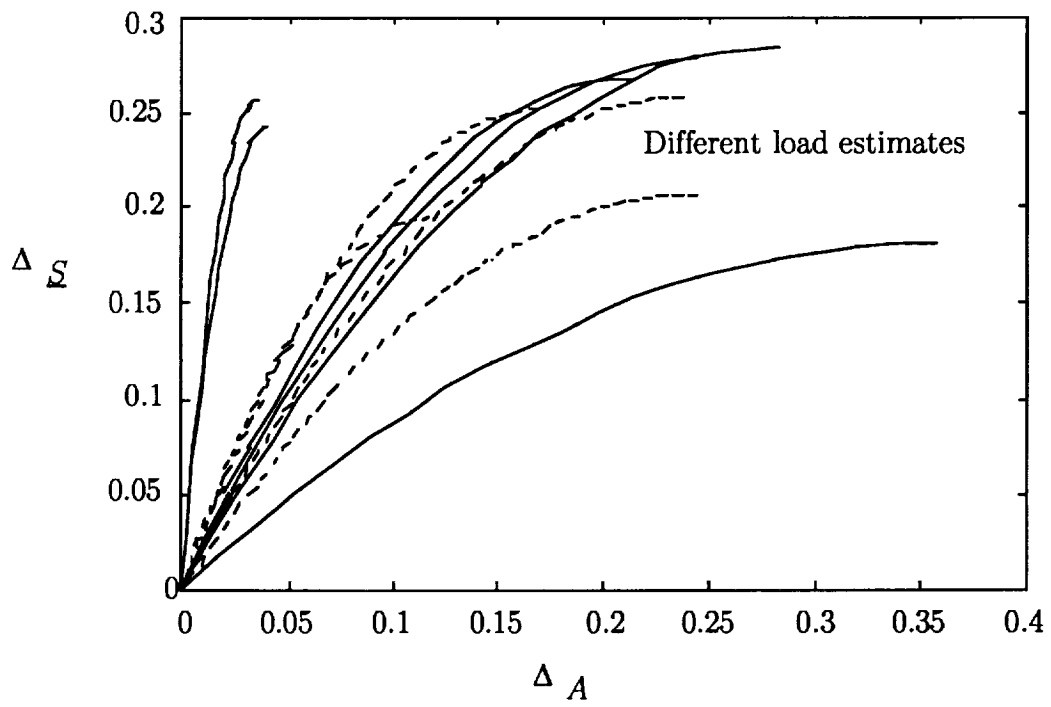


Figure 5.2: ΔS_{SCNR} vs. ΔA for NASA Data

Since $\tilde{m}_i = m_i + e_{oi} \Rightarrow m_i = \tilde{m}_i - e_{oi}$, (5.25) becomes

$$\Delta S_{SCNR} = \max_{0 < i < n} \Delta S = \max_{0 < i < n} \hat{b}_i^2 - \left(\frac{\Delta A}{\tilde{m}_i - e_{oi}} - \hat{b}_i \right)^2. \quad (5.25)$$

$$\Delta A = \delta A$$

A possible sub-optimal method can be used by restricting the amount of noise ϵ_i to the $r\sigma_{e_o}$ criteria. The max-min criteria can then be expressed as

$$\max_{0 < i < n} \left[\min_{-r\sigma_{e_o} < e_{oi} < r\sigma_{e_o}} \{ \Delta S | r\sigma_{e_o} \text{ criteria} \} \right]. \quad (5.26)$$

The above equation (5.26) can be interpreted as the maximum of the minimum of the ΔS under the $r\sigma_{ee}$ criteria. By setting $\epsilon_i = r\sigma_{ee}$, the max-min condition in (5.26) becomes

$$\begin{aligned} \Delta S_{SCNR-SO} = \max_{0 < i < n} \Delta S &= \max_{0 < i < n} \hat{b}_i^2 - \left(\frac{\Delta A}{\tilde{m}_i - r\sigma_{e_o}} - \hat{b}_i \right)^2 \\ &= \hat{b}_i^2 - \left(\frac{\Delta A}{\tilde{m}_i * -r\sigma_{e_o}} - \hat{b}_i \right)^2. \end{aligned} \quad (5.27)$$

Thus we have proposed a sub-optimal method for the SCNR procedure(SCNR-SO), which will not perform as good as the theoretical SCNR method. However, the analysis of SCNR and the subsequent SCNR-SO methods gives us an understanding of how the model accuracy and the noise sensitivity are related. It also provides an analytical bound for the region of operation in which SCNR-SO will perform better the original method.

5.5.2 Performance Analysis of the SCNR procedure

In order to have a better understanding of the characteristic of the new procedure, it is necessary to compare the performance of the proposed procedure and the original method of load prediction. As stated before, the presence of normal

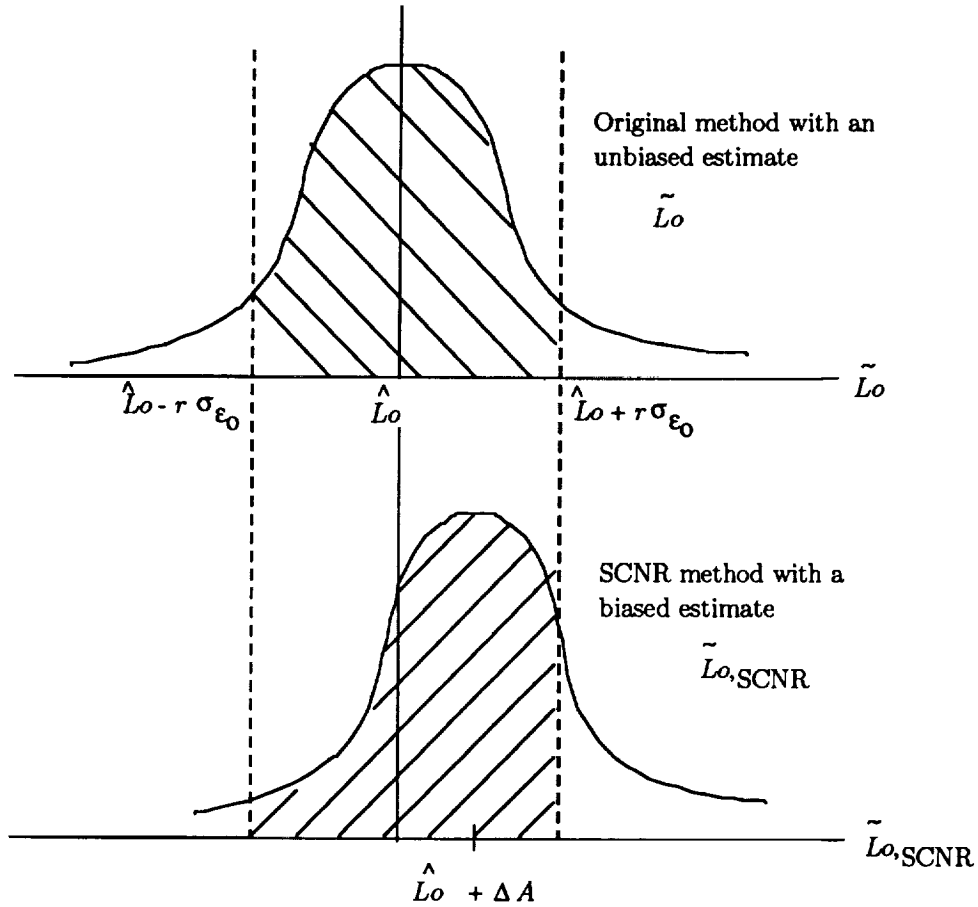


Figure 5.3: Probability Distribution for load estimate \tilde{L}_o

gaussian noise at the flight stage is assumed. One way of comparing their performance is by evaluating the probability that the noisy estimate \tilde{L}_o lies within a specific range for the original and the new SCNR method. Using general statistical methods for establishing confidence interval in normally distributed noise model, we can use the interval of $\pm r\sigma_o$. (For example, a 99.73 % confidence interval corresponds to $r=3$) Figure 5.3 shows the areas for evaluating the probabilities for the two methods.

Now, the probability that the noisy estimate \tilde{L}_o lies within the range using the original method is

$$P\{original\} \equiv P\{\hat{L}_o - r\sigma_{\epsilon_o} < \tilde{L}_o < \hat{L}_o + r\sigma_{\epsilon_o}\} = \frac{1}{2\pi} \int_{\hat{L}_o - r\sigma_{\epsilon_o}}^{\hat{L}_o + r\sigma_{\epsilon_o}} \exp\left(-\frac{x^2}{2}\right) dx, \quad (5.28)$$

where σ_{ϵ_o} is the standard deviation of the flight stage load error ϵ_o in (5.10) and r is an integer depending on the level of confidence. Similarly, the probability that the new SCNR method noisy estimate $\tilde{L}_{o,SCNR}$ lies within the same range is

$$P\{SCNR\} \equiv P\{\hat{L}_o - r\sigma_{\epsilon_o} < \tilde{L}_{o,SCNR} < \hat{L}_o + r\sigma_{\epsilon_o}\} = \frac{1}{2\pi} \int_{r_{1*}}^{r_{2*}} \exp\left(-\frac{x^2}{2}\right) dx, \quad (5.29)$$

where

$$\begin{aligned} r_{1*} &= \frac{-r\sigma_{\epsilon_o} - \Delta A}{\sigma_{\epsilon_o,SCNR}} \\ &= -\frac{\|\hat{b}\|}{\|\hat{b}_{SCNR}\|} - \frac{\Delta A}{\|\hat{b}_{SCNR}\|\sigma_{\epsilon_o}} \end{aligned} \quad (5.30)$$

and

$$\begin{aligned} r_{2*} &= \frac{r\sigma_{\epsilon_o} - \Delta A}{\sigma_{\epsilon_o,SCNR}} \\ &= \frac{\|\hat{b}\|}{\|\hat{b}_{SCNR}\|} - \frac{\Delta A}{\|\hat{b}_{SCNR}\|\sigma_{\epsilon_o}}. \end{aligned} \quad (5.31)$$

Then if the condition $P\{SCNR\} > P\{original\}$ in (5.28) and (5.29) is satisfied, the probability that the SCNR noisy estimate lies within the specified range is higher than that of the original method. In these cases, the new method is much superior than the old method. In other words, the noisy estimate is more likely to be confined in the specified range when using the new method. Figure 5.4 below, which is generated from NASA data, indicates that the SCNR method performs better in a significantly large region of operation and especially when large flight

stage noise variance σ_{e_o} is present. From (5.27) and (5.28), the probability for the SCNR-sub-optimal method can be written as

$$P\{SCNR - SO\} = \frac{1}{2\pi} \int_{r_{1**}}^{r_{2**}} \exp(-\frac{x^2}{2}) dx, \quad (5.32)$$

where

$$r_{1,2**} = \mp \frac{\|\hat{b}\|}{\|\hat{b}_{SCNR-SO}\|} - \frac{\Delta A}{\|\hat{b}_{SCNR-SO}\| \sigma_{e_o}} \quad (5.33)$$

and

$$\begin{aligned} \|\hat{b}_{SCNR-SO}\|^2 &= \|\hat{b}\|^2 - \Delta S_{SCNR-SO} \\ &= \|\hat{b}\|^2 - [\hat{b}_i^2 - (\frac{\Delta A}{\tilde{m}_i * -r\sigma_{e_o}} - \hat{b}_i)^2]. \end{aligned} \quad (5.34)$$

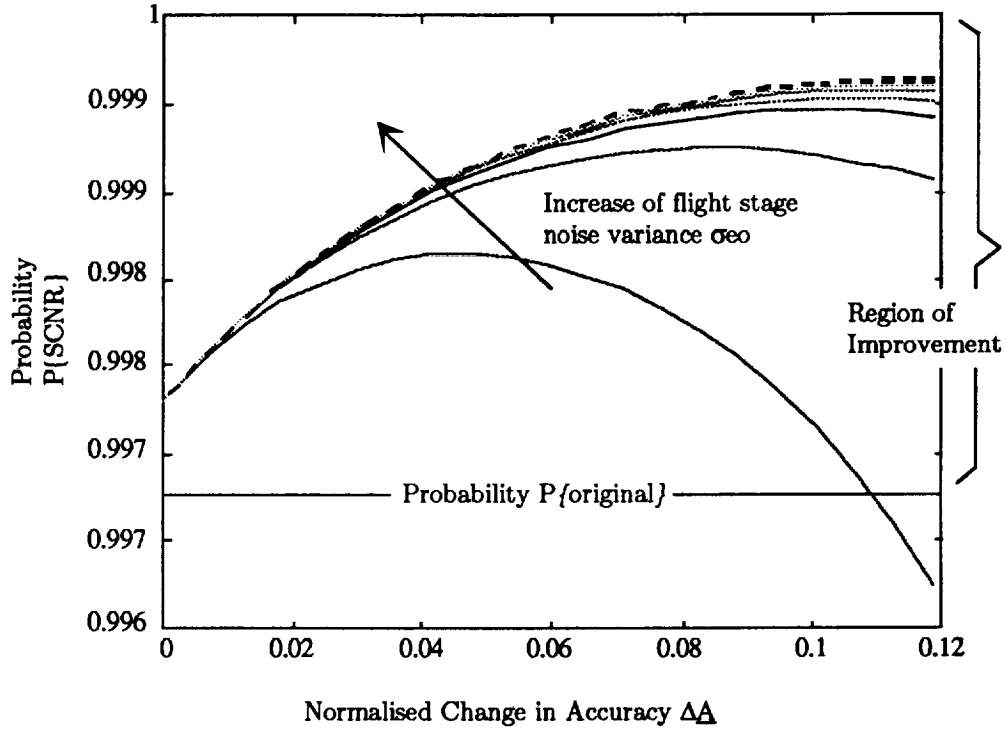


Figure 5.4: *Probability vs ΔA for NASA data*

In summary, the new procedure essentially generates a biased load estimate with a reduced noise variance whereas the original method gives a non-biased estimate and a larger variance. Since both of the two methods have desirable characteristic in the load estimation, one has to decide how to make trade-off between the noise sensitivity(variance) and the model accuracy(bias). Generally if a large amount of noise is expected, the new SCNR could be used to reduce the corrupting effects of the flight stage noise by confining the deviation of the load estimate.

5.6 Generalized Method for Reduction of Noise Sensitivity(GNR)

In (5.15), the variance of the error ϵ_o is reduced by changing \hat{b}_i to $c_i \hat{b}_i$. Now we will generalize the method by using the scaled vector \hat{b}_{GNR} such that

$$\hat{b}_{GNR} = Bc, \quad (5.35)$$

where $B = \text{diag}(\hat{b}_1, \dots, \hat{b}_n)$ and $c = [c_1, \dots, c_n]'$, $c_i \leq 1$. Then the new load estimate using GNR is

$$\hat{L}_{o,GNR} = M_o' \hat{b}_{GNR} = M_o' Bc. \quad (5.36)$$

From (5.16) and using (5.35), the decrease in noise sensitivity is

$$\Delta S = \|\hat{b}\|^2 - \|\hat{b}_{GNR}\|^2 = (1 - c)' B B (1 - c). \quad (5.37)$$

Also from (5.19) and (5.36), the change in the noiseless load prediction accuracy can be written as

$$\Delta A = \hat{L}_o - \hat{L}_{o,GNR} = M_o' B \cdot 1 - M_o' Bc = M_o' B(1 - c). \quad (5.38)$$

In order to reduce the noise sensitivity by using the GNR method, we need to maximize the change in noise sensitivity ΔS with respect to a fixed ΔA such that

$$\begin{aligned} \max_c \Delta S &= \min_c c' B B c. \\ M_o' B(1 - c) &= \Delta A \end{aligned} \quad (5.39)$$

(5.39) can be solved by the Lagrangian method with

$$G(c) = c' B B c + \lambda [M_o' B(1 - c) - \Delta A]. \quad (5.40)$$

Taking the gradient $G(c)$ with respect to c and set it to zero gives

$$\nabla G(c^*) = 2BBc^* + \lambda'BM_o(-1) = 0 \quad (5.41)$$

and

$$c^* = -\frac{\lambda}{2}B^{-2}BM_o = \frac{\lambda}{2}B^{-1}M_o. \quad (5.42)$$

Putting c^* into the constraint equation gives

$$\begin{aligned} M_o'B(1 - c) &= \Delta A, \\ M_o'B(1 + \frac{\lambda}{2}B^{-1}M_o) &= \Delta A, \\ \lambda &= 2\frac{(\Delta A - M_o'B1)}{M_o'BB^{-1}M_o} = 2\frac{(\Delta A - \hat{L}_o)}{\|M_o\|^2}. \end{aligned} \quad (5.43)$$

Then c^* can be expressed as

$$c^* = \left(\frac{\hat{L}_o - \Delta A}{\|M_o\|^2}\right)B^{-1}M_o. \quad (5.44)$$

Therefore the change in the noise sensitivity ΔS_{GNR} can be written as

$$\Delta S_{GNR} = \frac{1}{\sigma_{\epsilon_o}^2}(\sigma_{\epsilon_o}^2 - \sigma_{\epsilon_o,GNR}^2) = \|\hat{b}\|^2 - \frac{(\hat{L}_o - \Delta A)^2}{\|M_o\|^2}. \quad (5.45)$$

5.6.1 Sub-optimal Generalized method for Noise Reduction

As stated in the previous section, the value of M_o in (5.45) is not known. Therefore, we need to tolerate a sub-optimal procedure in practical applications. The change in noise sensitivity ΔS using GNR is

$$\Delta S_{GNR} = \|\hat{b}\|^2 - \frac{(\hat{L}_o - \Delta A)^2}{\|M_o\|^2}. \quad (5.46)$$

Similarly from (5.45), the sub-optimal GNR using the "Maximum-minimum Criteria" gives

$$\Delta S_{GNR-SO} = \max_c \min_{-r\sigma_{e_o} < \epsilon_o < r\sigma_{e_o}} \Delta S. \quad (5.47)$$

Applying the $r\sigma_{e_o}$ Criteria, (5.47) becomes

$$\Delta S_{GNR-SO} = \|\hat{b}\|^2 - \frac{(\bar{L}_o + r\sigma_{e_o}\|\hat{b}\| - \Delta A)^2}{\|\bar{M}_o\|^2 - nr^2\sigma_{e_o}^2}. \quad (5.48)$$

5.6.2 Performance Analysis of the GNR procedure

The probability that the GNR method noisy estimate $\bar{L}_{o,GNR}$ lies within the $r\sigma_{e_o}$ interval can be expressed as

$$P\{GNR-SO\} = P\{\hat{L}_o - r\sigma_{e_o} < \bar{L}_o < \hat{L}_o + r\sigma_{e_o}\} = \frac{1}{2\pi} \int_{r_{1*}}^{r_{2*}} \exp -\frac{x^2}{2} dx, \quad (5.49)$$

where from (5.30)

$$r_{1,2*} = \mp \frac{\|\hat{b}\|}{\|\hat{b}_{GNR}\|} - \frac{\Delta A}{\|\hat{b}_{GNR}\|\sigma_{e_o}} \quad (5.50)$$

and

$$\hat{b}_{GNR} = \frac{(\hat{L}_o - \Delta A)}{\|\bar{M}_o\|}. \quad (5.51)$$

Similarly for suboptimal GNR method, the interval for the evaluation of the probability in (5.49) can be expressed as

$$r_{1,2**} = \mp \frac{\|\hat{b}\|}{\|\hat{b}_{GNR-SO}\|} - \frac{\Delta A}{\|\hat{b}_{GNR-SO}\|\sigma_{e_o}}, \quad (5.52)$$

where

$$\|\hat{b}_{GNR-SO}\| = \frac{(\bar{L}_o + r\sigma_{e_o}\|\hat{b}\| - \Delta A)^2}{\|\bar{M}_o\|^2 - nr^2\sigma_{e_o}^2}. \quad (5.53)$$

5.7 Conclusion and Remarks

This section provided a study of how the trade-off between variance(or noise sensitivity ΔS) and bias(or accuracy ΔA). We also presented the new methods for an effective reduction of noise variance when a given amount of accuracy deterioration ΔA is allowed. This problem is also studied in a different setting in the field of statistical analysis. In Ridge Regression [7], the problem of constrained LS is solved. It solves the LS problem of the $\min_x \|Ax - b\|$ with the constraint that $\|x\| \leq \alpha$. However, in this problem the idea relating the prediction accuracy and sensitivity is not explicitly used. Thus our study provided an alternate approach for analyzing how variance and bias are related to each other.

Chapter 6

Total Least Squares and Correspondence Analysis

The Total Least Squares(TLS) method was popular in signal processing applications recently, while the correspondence analysis technique was created over twenty years ago and used in applied statistical and data analysis. These two seemingly distinct topics developed independently among their own researchers. In this section, we provided the basic reformulation and analytical and geometrical tools to prove the equivalency of these two basic and useful methods.

In this section, we consider the total least squares estimation method well known in numerical analysis and modern signal processing as well as the correspondence analysis technique encountered in applied data and clustering analysis. Due to historical reasons of development, each of these two subjects have generally been formulated with its own notations and solutions. However, upon more detailed consideration, both optimization problems reduce to the application of SVD technique for the respective solution. Indeed, upon appropriate

pre-processing operation of translating the centroids of the data matrices to the origin, these two methods are equivalent.

6.0.1 Total Least Squares(TLS)

Least squares techniques based on the statistical assumption that only error is present in the vector to be estimated. Total least squares, which is introduced by Golub and Van Loan, attempts to generalize the solutions by allowing both observation matrix and estimator vector to be corrupted by errors. This assumption allows us to have a more consistent solution since errors are usually presented in all data. The problem of TLS can be stated as follows. For an overdetermined system of equations of the form $Ax \approx B$, errors occurred in both A and B . The TLS solutions for x solves the perturbed problem of

$$(A + \Delta A)x = B + \Delta B,$$

with the constraint that the F norm $\|\Delta A, \Delta B\|_F$ is minimized. It can be seen that the classical LS problem solves the same problem above except that no error is assumed in the matrix A (or $\Delta A = 0$).

6.0.2 Correspondence Analysis(CA)

In statistical and data analysis, there are myraids of analytical, graphical, and intuitive methods for performing data reduction, clustering and display of statistical properties. The CA techniques was proposed by Benzecre [3] and studied by Lebart [19], Greenacre [9] and others. This technique was clearly motivated by several standard statistical multivariate techniques of principal component analysis and discriminant analysis. It provides an analytical method of displaying of

a large centered matrix in low dimensional spaces such as the two dimensional space.

Graphical and Numerical Examples

For a given data matrix $X \in \mathbf{IR}^{m \times n}$, a basic problem of data reduction is to determine and eliminate the set of redundant columns(or rows). If two vectors are strictly independent, one of the vectors can be eliminated. However, the presence of noise will make the redundant vector strictly independent. The concept of collinearity in [25] quantify the amount of linear dependencies for a given column(or row) vector relative to all other columns(or row) vectors of X . The collinearity index of matrix X can be defined as

$$\kappa_j = \|x_{\text{ldot}j}\| \| [X^+]_j \text{ th row} \|, \quad j = 1, \dots, n, \quad (6.1)$$

where X^+ is the pseudo-inverse of X .

We now look at a physical load problem when the 18×12 strain gage measurement matrix X is given by $X =$

12.5	-13.8	11.6	4.2	11.8	11.9	11.3	23.4	36.9	50.3	89.4	128.7
21.4	-2.9	19.2	12.8	17.9	21.5	16.5	40.0	42.1	73.7	58.2	68.3
30.8	9.1	23.4	23.6	20.7	36.9	28.8	67.4	28.9	38.3	40.3	27.7
42.5	29.7	29.1	43.6	28.2	64.4	24.6	37.2	20.9	20.4	24.8	8.9
56.6	61.4	41.4	76.8	25.4	37.2	21.7	23.1	15.8	10.8	15.3	-2.7
77.8	126.3	41.0	54.7	24.4	21.5	19.8	12.3	8.2	2.2	3.5	-15.2
27.0	-19.2	26.3	8.5	29.2	15.3	23.6	25.8	64.1	29.6	132.8	82.1
45.4	-2.3	35.8	18.0	32.9	20.2	31.6	27.7	59.0	25.6	98.8	56.8
62.3	14.0	45.2	24.2	37.1	21.5	37.8	27.1	45.3	21.3	72.4	34.2
85.8	34.3	55.7	30.0	42.8	21.2	41.1	26.5	36.4	17.0	50.2	13.9
110.7	55.2	68.0	33.4	44.1	19.6	43.4	22.8	28.9	11.7	29.5	-3.0
131.4	82.0	66.6	44.2	39.4	21.2	37.3	19.1	13.9	6.5	5.4	-18.9
46.3	-22.6	41.9	11.9	38.1	14.7	40.1	16.6	83.4	0.9	160.6	63.0
64.6	-3.5	49.0	18.3	41.8	13.7	43.4	16.6	69.3	4.6	129.5	48.9
138.4	51.2	80.8	19.3	53.1	9.4	52.9	20.9	30.8	10.8	35.1	-2.7
163.3	62.6	82.7	30.7	49.4	19.0	49.1	21.9	20.0	6.8	8.7	-18.9
91.4	15.5	65.1	18.6	50.8	4.2	51.0	6.8	58.0	3.7	96.9	31.7
114.4	35.9	74.6	18.9	53.1	1.7	54.8	14.8	45.8	10.5	68.2	16.4

These data are obtained from a NASA/Ames hypersonic wing test structure (HWTSS) load measurement experiment. The columns represent the output responses of 12 strain gages located at the wing and fuselage region of the aircraft when 18 known input load conditions are applied on different locations of the wing

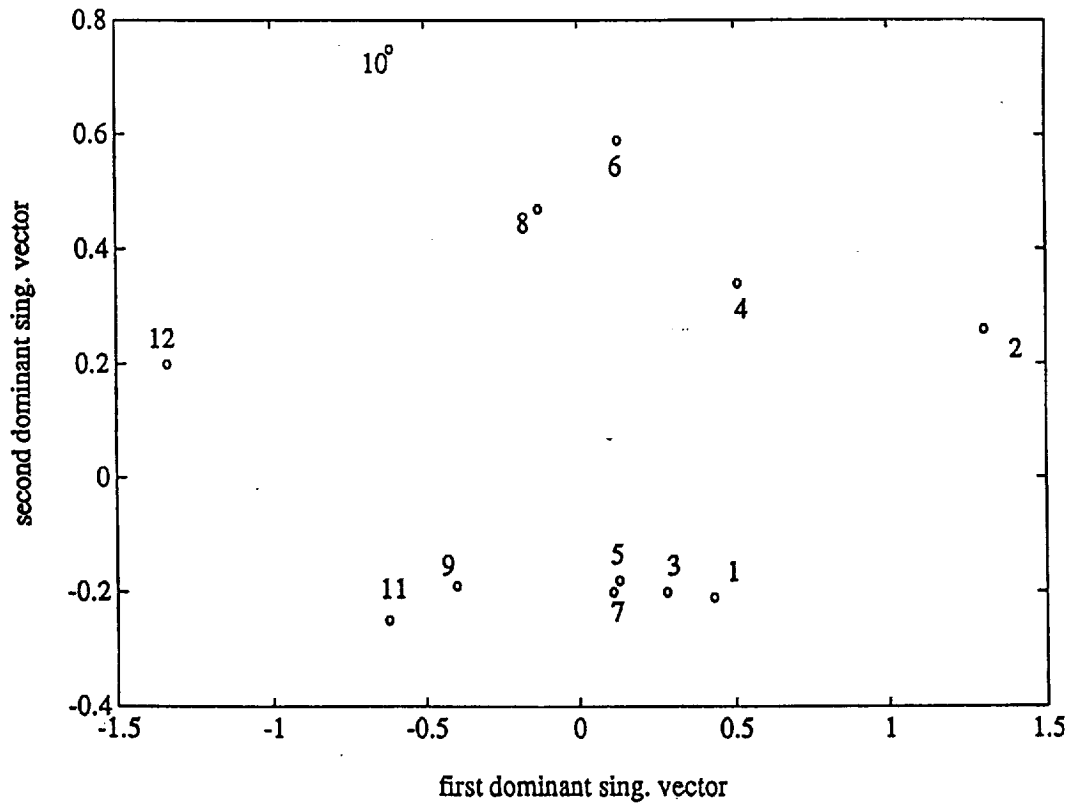


Figure 6.1: Correspondence Analysis of NASA HWTSS load data

structure in the calibration stage. In practice, the number of gages available during a flight is usually much less than that available during the calibration. Thus, from the data in matrix X , we are motivated to determine the redundant gages that can be eliminated. Evaluations Collinearity indices are computed as $\kappa = (24, 4.3, 52, 7.8, 34, 6.2, 29, 7.5, 32, 8.3, 32, 9.5)$. Applying the CA technique gives $S = \text{diag}(.61, .31, .18, .18, .083, .067, .051, .031, .011, 6.6 \cdot 10^{-3}, 5.6 \cdot 10^{-3}, 7.2 \cdot 10^{-17})$. The coefficients of expansion of the 12 column vectors for the two dominant singular vectors are shown in Fig. 6.1.

From the collinearity index κ , we observed that all the odd numbered column data vectors are fairly collinear. From Fig. 6.1, we see that these vectors are grouped closely together. However, the even-numbered column data vectors are

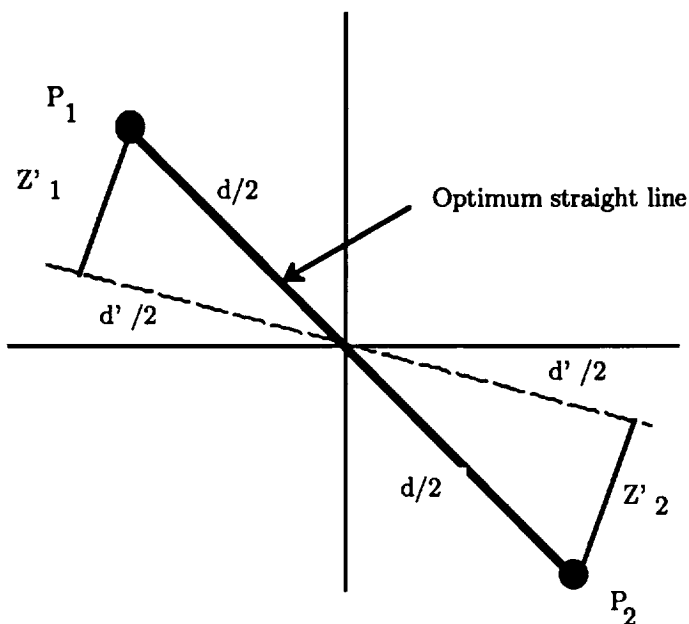


Figure 6.4: Mean translated data vectors

is zero. In general, the above condition need not be satisfied for arbitrarily P_1 and P_2 . Now, consider the column zero mean translated version of A and b .

$$\bar{A} = \begin{bmatrix} (a_1 + a_2)/2 \\ (a_1 + a_2)/2 \end{bmatrix}, \bar{b} = \begin{bmatrix} (b_1 + b_2)/2 \\ (b_1 + b_2)/2 \end{bmatrix}, \quad (6.5)$$

$$\text{and } \tilde{A} = A - \bar{A} = \begin{bmatrix} (a_1 - a_2)/2 \\ (a_1 - a_2)/2 \end{bmatrix}, \tilde{b} = b - \bar{b} = \begin{bmatrix} (b_1 - b_2)/2 \\ (b_1 - b_2)/2 \end{bmatrix}$$

$$\tilde{P}_1 = \begin{bmatrix} (a_1 - a_2)/2 \\ (b_1 - b_2)/2 \end{bmatrix}, \tilde{P}_2 = \begin{bmatrix} (a_2 - a_1)/2 \\ (b_2 - b_1)/2 \end{bmatrix}. \quad (6.6)$$

After mean translation, \tilde{P}_1 and \tilde{P}_2 are anti-symmetric. That is, $\tilde{P}_1(1) = -\tilde{P}_2(1)$ and $\tilde{P}_1(2) = \tilde{P}_2(2)$. Clearly, the optimum straight line runs from \tilde{P}_1 to \tilde{P}_2 , with $Z_1^2 + Z_2^2 = 0$ and $d = 2\|\tilde{P}_1\|^2 = 2\tilde{P}_2^2$. For any other alternate straight

line, $Z_1'^2 + Z_2'^2 > 0$, and

$$d'^2 = (d'/2 + d'/2) = \sqrt{(d/2)^2 - Z_1'^2} + \sqrt{(d/2)^2 - Z_2'^2} < d/2 + d/2 = d. \quad (6.7)$$

Thus, any alternate straight line yields inferior TLS and CA results.

6.1.2 TLS and CA Criterion for general dimensions of data matrix

Correspondence analysis can be used to represent data collected in matrix form in a more compact format by projecting the raw data onto a lower dimensional subspace.

Let $A = (a_{ij}) \in \mathbf{R}^{m \times n}$ with $m > n$ the original data matrix. For simplification purposes, which will become clear later, let us assume that $\sum_{i=1}^m a_{ij} = 0$ (i.e. A is "centered") and let the expression of $A = U\Sigma V'$ be its singular value decomposition.

Now, let $Q = [q^{(1)}, \dots, q^{(j)}] \in \mathbf{R}^{n \times J}$ be a matrix with $J \leq n$ orthonormal columns and $H = AQ = (h_{ij})$, with $h_{ij} = a_i \cdot q^{(j)}$, the matrix with its elements as the the projections of A onto Q .

In this context, Q represents the basis-matrix of a J -dimensional subspace. For the purposes of correspondence analysis, it is reasonable to vary Q so that the projection coefficients h_{ij} are as separated as possible, with respect to each basis vector $q^{(j)}$.

In order to satisfy this requirement, one can perform the maximization of the following quantity over the space spanned by the columns of Q such that

$$D \equiv \sum_{j=1}^J \sum_{i=1}^m \sum_{k=1}^m (h_{ij} - h_{kj})^2$$

$$\begin{aligned}
&= \sum_{j=1}^J \sum_{i=1}^m \sum_{k=1}^m (h_{ij}^2 - 2h_{ij}h_{kj} + h_{kj}^2) \\
&= m \sum_{j=1}^J \sum_{i=1}^m h_{ij}^2 - 2m \sum_{j=1}^J \sum_{i=1}^m \sum_{k=1}^m h_{ij}h_{kj} + m \sum_{j=1}^m \sum_{k=1}^m h_{kj}^2 \\
&= 2m \sum_{j=1}^J \sum_{i=1}^m h_{ij}^2 - 2 \sum_{j=1}^J \sum_{i=1}^m h_{ij} \sum_{i=1}^m h_{kj} \\
&= 2m \sum_{j=1}^m \sum_{i=1}^m (h_{ij}^2 - \sum_{k=1}^m \frac{h_{kj}^2}{m}) \\
&= 2m \sum_{j=1}^J \sum_{i=1}^m (h_{ij} - \bar{H}_j)^2,
\end{aligned}$$

where $\bar{H}_j = \sum_{i=1}^m h_{ij}/m$. Since A is "centered", it can be shown that $\bar{H}_j = 0, \forall j$.

Therefore

$$\max D = 2m \max \sum_{j=1}^J \sum_{i=1}^m h_{ij}^2 \quad (6.8)$$

$$= 2m \max \|H\|_F^2, \quad (6.9)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

The above maximization can be solved by the use of the singular value decomposition of A. In fact,

$$\begin{aligned}
\|H\|_F^2 &= \|AQ\|_F^2 = \|U\Sigma V'Q\|_F^2 \\
&= \|U\Sigma\tilde{V}'\|_F^2 \\
&= \|U\Sigma[\tilde{v}_1, \dots, \tilde{v}_J]\|_F^2 \leq \sum_{i=1}^J \sigma_i^2.
\end{aligned}$$

The equality is obtained when $Q = V_J = [v_1, \dots, v_J]$, hence

$$\tilde{V} = Q'V = [I_J, 0_{J \times (n-J)}] \quad (6.10)$$

and $H = AQ = AV_J = U\Sigma V'V_J = [\sigma_1 u_{.1}, \dots, \sigma_J u_{.J}]$. This results are known in the area of correspondence analysis.

Now, let $[Q, Q^\perp] \in \mathbf{R}^{n \times n}$ be an orthonormal expansion of Q . Then we have

$$\begin{aligned}
\|H\|_F^2 &= \sum_{j=1}^J q^{(j)'} A' A q^{(j)} \\
&= \sum_{i=1}^n q^{(i)'} A' A q^{(i)} - \sum_{j=J+1}^n q^{(j)'} A' A q^{(j)} \\
&= \sum_{j=1}^n q^{(j)'} V \hat{\Sigma}^2 V' q^{(j)} - \|A Q^\perp\|_F^2 \\
&= \sum_{j=1}^n \sum_{k=1}^n \sigma_k^2 (q^{(j)'} v_{\cdot k})^2 - \|A Q^\perp\|_F^2 \\
&= \sum_{k=1}^n \sigma_k^2 - \|A Q^\perp\|_F^2,
\end{aligned}$$

where $\hat{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n) \in \mathbf{R}^{n \times n}$. As a result, the maximization of D is equivalent to the minimization of

$$\min \|A Q^\perp\|_F^2 = \min \|A(I_n - Q Q')\|_F^2 = \min \|A - \hat{A}\|_F^2 = \min \|\Delta \hat{A}\|_F^2, \quad (6.11)$$

where $\hat{A} = H Q'$ and $\Delta \hat{A} = A - \hat{A}$.

The maximization problem stated above (or minimizing D) is therefore minimizing the energy of the perturbation imposed on the original data matrix A . This perturbation reduces the rank of A to $\text{rank}(\hat{A}) \leq J$ when the vector X in the orthogonal subspace spanned by Q^\perp solves the equation $\hat{A}X = 0$.

Case for $\bar{H}_j \neq 0$

Now consider the case when A is not "centered", $\bar{H}_j \neq 0$ or $\sum_{i=1}^m a_{ij} \neq 0$ (which is a more general case), then we can write

$$\begin{aligned}
\max_Q D &= 2m \max \|H - \frac{1}{m} \mathbf{1}_m H\|_F^2 \\
&= 2m \max \|(I_m - \frac{1}{m} \mathbf{1}_m) H\|_F^2 \\
&= 2m \max \|H_o\|_F^2
\end{aligned}$$

$$\begin{aligned}
&= 2m \max \|PH\|_F^2 \\
&= 2m \max \|PAQ\|_F^2,
\end{aligned}$$

where $\mathbf{1}_m$ is a square matrix of all ones, $P = I_m - \frac{1}{m}\mathbf{1}_m$ and $H_o = PH$. Thus we can still carry out the same data reduction by using $A_o = PA$, a "centered" form of A .

Case for shifted H criteria

In order to justify the equivalence of TLS and CA, we need to look at a shifted H criteria. Now we consider the shifted projection of A onto the subspace spanned by columns of Q such that

$$H = AQ + B. \quad (6.12)$$

Instead of maximizing D with $H = AQ$ as before, we maximize the shifted $H = AQ + B$ as described above. Therefore we can write

$$\begin{aligned}
\max_Q D &= 2m \max \sum_{j=1}^J \sum_{i=1}^m (h_{ij} - \hat{H}_j)^2 \\
&= 2m \max \|P(AQ + B)\|_F^2 \\
&= 2m \max \|(A_o, B_o) \begin{bmatrix} Q \\ -I_J \end{bmatrix}\|_F^2 \\
&= \sqrt{2}m \max \|(A_o, B_o)\tilde{Q}\|_F^2,
\end{aligned}$$

where A_o and B_o are centered form of A and B and $\tilde{Q} = \frac{1}{\sqrt{2}}(Q', -I_J)' \in \mathbf{IR}^{(n+J) \times J}$ and $\tilde{Q}'\tilde{Q} = I_J$.

From the above, the form of the minimizing with the cases before is the same as that of the shifted version which we are now minimizing the expression of

$$\min \|(A_o, B_o)(I_{n+J} - \tilde{Q}\tilde{Q}')\|_F^2 = \min \|\Delta(\hat{A}_o, \hat{B}_o)\|_F^2. \quad (6.13)$$

Thus the results can be summarized as:

Correspondence Analysis

For the shifted version of $H = AQ+B$, the minimizing problem is equivalent to that of

$$\min \|\Delta(\hat{A}_o, \hat{B}_o)\|_F^2,$$

or finding the vector X in the orthogonal subspace spanned by $\tilde{Q} = \frac{1}{\sqrt{2}}(Q', -I_J)'$ which satisfies the equation $(\hat{A}_o, \hat{B}_o)X = 0$.

Total Least Squares

The total least squares problem $A\mathcal{X} \approx B$ can be stated as follows: find the minimum norm vector $X = [\mathcal{X}', -\mathcal{I}]'$ which solves exactly the perturbed equation $(\hat{A}, \hat{B})X = 0$, where $(\hat{A}, \hat{B}) = (A, B) - \Delta(\hat{A}, \hat{B})$. The Frobenius norm of the perturbation applied to the appended matrix (\hat{A}, \hat{B}) , namely $\|\Delta(\hat{A}, \hat{B})\|_F$, has to be minimized. Therefore both TLS and CA approaches are equivalent in this formulation.

Assumptions and Observations

For the TLS and CA equivalence to hold, we have the following assumptions.

- Both matrix A and B have to be centered as stated above.
- In Correspondence Analysis, the rank j of the subspace on which the matrix A is projected can be chosen freely. However, the rank of the TLS problem are usually reduced to the numerical rank of A itself.

Chapter 7

Neural Networks and its applications

7.1 Introduction

The development of neural networks are motivated by the theory of human neural systems. The theory of human neural systems involves the study of the behavior of the interconnected processing elements called neurons. Neurons are processing or decision making units by which information are passed from one region to another region. In human nervous system, a vast number of interconnected neurons are responsible for processing and relaying information and commands from one part of the human body to the other. This complex chain of commands and decision making process motivated the development of Artificial Neural Network(NN) which mimic the behavior of the human nervous system. The basic elements in neural networks called neurons is modeled as that of the human nervous system. Signals processed by the neurons travels to adjacent neurons such that a specific

command or signal can be passed to the intended destination. In recent years, thanks to the availability of high speed computers, much attention has been given to the development of Artificial Neural Network with applications ranging from pattern recognition, function approximation, image processing, system identification to dynamical system problems. It is illustrated in [20] [12] [21] that a large number of problems can be handled by some forms of neural networks without prior assumption of the system model. This robust nature makes neural networks a good methodology for solving a wide range of problems. Before demonstrating the usefulness of neural networks in solving our load measurement problem, the following section will give a brief overview of the theory and development of neural networks by presenting some practical examples and applications.

7.2 An Overview of Neural Networks

Since the theory of neural networks is still not quite mature, there are still some unsolved issues such as the type of architecture of the neural networks, the type of training methods and the design procedures. Recently, much works have been done on the applications of neural networks. There are two main categories in neural networks. The first category is called the dynamic neural networks as the system parameters is dependent on time or the network possess "memory". The second category is called static neural networks since the system parameters are static in time or "memoryless".

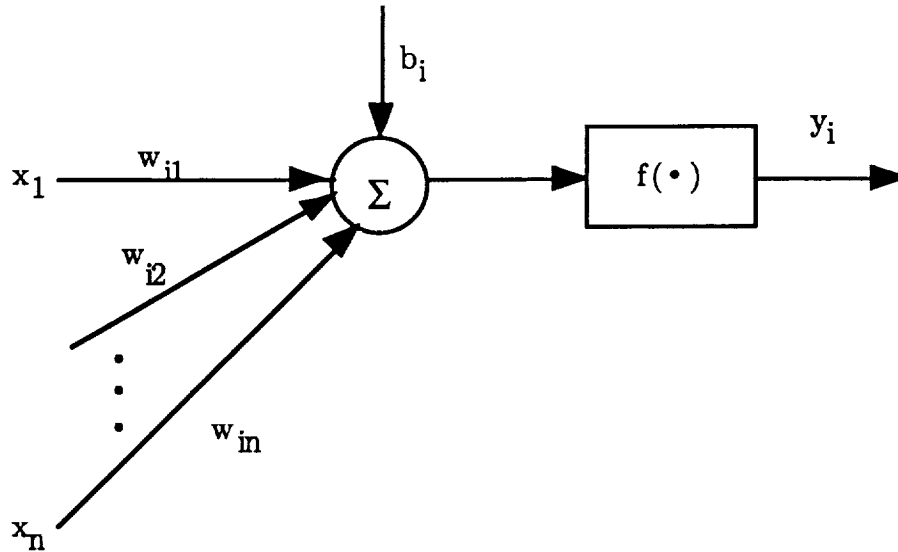


Figure 7.1: A basic neuron

7.2.1 Basic Neural Units or Neurons

Neural Networks consist of basic processing units called neurons. It can be viewed as input/output devices where the output is determined by a specific rule or a functional relation. The figure (7.1) shows a neuron with multiple inputs $x = [x_1, \dots, x_n]' \in \mathbb{R}^n$ and a single output y_i . In addition, the neuron is defined by its activation function $f(\cdot)$, the bias b_j and the weights w_{ij} .

Thus, the output of the i -th neuron can be written as

$$y_i = f\left(\sum_{j=1}^n x_j w_{ij} + b_i\right), \quad (7.1)$$

where n is the total number of input to the i -th neuron.

The activation function $f(\cdot)$ is an important element of a neural network for the nature of the function introduces nonlinearity into the NN. Such nonlinearities, which we will explore later, allows the NN to handle nonlinear problems. The type of activation function used is known as the sigmoid function which usually take

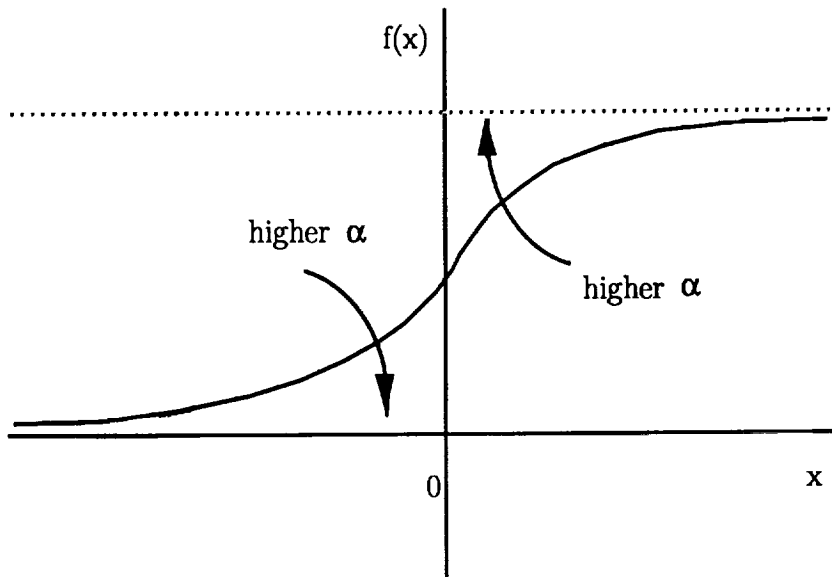


Figure 7.2: A sigmoid function

the form of a step function, tangent, logarithm or sometimes even linear function. An examples of such a function is shown in fig (7.2) where the functional relation can be expressed as,

$$f(x) = \frac{1}{1 + e^{-\alpha x}}. \quad (7.2)$$

Furthermore, the bias b_i and the parameter α determine the shape of the sigmoid function which we can select during the design phase of a neural network.

7.2.2 Basic Neural Networks

A typical neural network consists of interconnected neurons. Different types of architecture are commonly used in NN. Neurons can be fully connected(i.e. each neuron is connected to the other neurons via a weigh). However, most popular NN requires only neurons to be connected to adjacent layers of neurons which we will subsequently discuss in detail. Besides the architecture of the NN, NN can be

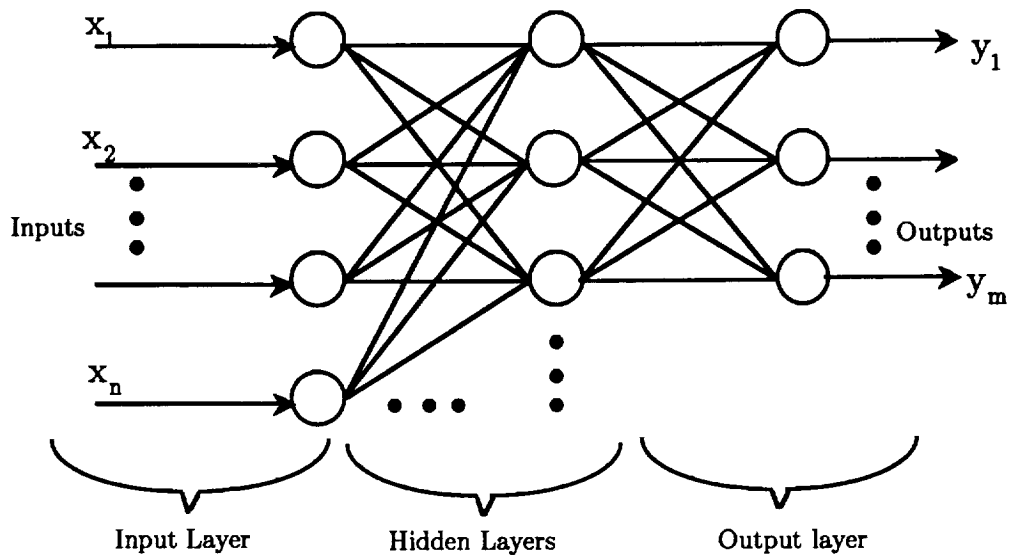


Figure 7.3: A Multiple Layer Perceptron Neural Network

classified as the "feedforward static NN" and "feedback/recurrent /dynamic NN". Feedforward NN allows signals to flow in one direction. One of the important types of NN is called Multiple Layer Perceptron(MLP). The second class of NN is characterized by the feedback network such as the famous Hopfield Network.[21] [32] In chapter 8, NN will be used to solve the load measurement problem. The type of NN to be used for this application will be in the form of a static NN or MLP.

7.2.3 Multiple Layer Perceptron (MLP)

A neural network can be made up of several layers of neurons. In MLP, only adjacent layers are connected to each other. Fig. (7.3) shows a fully connected MLP Neural Network.

A NN can be partitioned into the input layer, the hidden layers, and the output layer. In fig (7.3), the number of hidden layers shown can be selected for some specific problems. However, the exact number of layers to be chosen for a specific problem is still quite arbitrary. It usually requires experimentations or trial-and-error methods in determining the number of layers for possible optimal solutions. Works in this area shown in [20] proved that two hidden layers NN can approximate any types of function if enough number of neurons are being used. However, additional layers might reduce the number of required neurons and thereby greatly reduces the complexity of the problem. In chapter 9, we will discuss this aspect of the problem and suggest a procedure of designing a neural network by selecting the number of neurons in a layer using our load estimation problem as an example.

7.2.4 Training Techniques for Neural Networks

In order for the Neural Network to act as a function approximation, system identification or pattern recognition network, we need to train the neural network by adjusting the weights and bias iteratively until the desired objective or error criterion is achieved. The training procedures can be classified into two types. One type is called the supervised learning and the other type is called the unsupervised learning. Supervised learning [12] involves the use of target values or patterns. Targets and input pairs are presented to the neural network for training at each step until an error criterion has been reached. For unsupervised learning, the neural network is only presented with inputs without the aid of the targets.

7.3 Training by Back Propagation(BP) Technique

Due to the relative ease of use and the stability in the solution, Back Propagation became one of the most popular training techniques since its introduction [27]. For enhancing the convergence properties of the BP technique, modifications had been made such as the addition of momentum terms in the direction of update. BP is essentially based on the optimization theory using first order gradient descent methods. The success of the convergence of the BP relies on the direction of the search. In addition, the BP attempts to solve the optimization problem which is equivalent to the search of the global minimum to an energy or error function defined as E . As a result, BP sometimes give solution of local minimum and may not always arrive at the global minimum. Other methods using a momentum term (or a second order gradient method) such as conjugate gradient method as studied in [17] attempts to avoid the solution to settle on a local minimum.

7.3.1 Notations and Definitions

Here are some of the notations and definitions used in deriving the Back Propagation method:

- E – Error or energy function for minimization.
- L – Last layer of the neural network.
- N_L – Number of neurons in the L -th (last) layer.

- N_l – Number of neurons in the l -th layer.
- m – m -th step of the iterative updating process.
- t_i – i -th element of the training vector.
- $w_{ij}(l)$ – The weight connecting j -th neuron's output in $(l - 1)$ -th layer to i -th neurons in l -th layer.
- $a_i(l)$ – Output of the i -th neuron in the l -th layer.
- $u_i(l)$ – Input to the i -th neuron's activation function in the l layer.
- μ – Learning rate for the back propagation training.
- $b_i(l)$ – Bias of the i -th neuron in the l -th layer.

7.3.2 Derivation

Using the notations defined previously, the input $u_i(l)$ and output $a_i(l)$ to the i -th neuron in the l layer can be expressed as

$$\begin{aligned} u_i(l) &= \sum_{j=1}^{N_i} w_{ij}(l) a_j(l-1) + b_i(l), \quad \text{and} \\ a_i(l) &= f(u_i(l)). \end{aligned} \tag{7.3}$$

In Back Propagation, the weights are updated according to the expression as

$$w_{ij}^{(m+1)}(l) = w_{ij}^{(m)}(l) + \Delta w_{ij}^{(m)}(l), \tag{7.4}$$

where m is the iteration step and $\Delta w_{ij}^{(m)}(l)$ is the change or updates in weights for step m .

The weight updating process is according to a first order gradient descent method such that the updating direction is in the negative gradient of the energy or error function E as expressed by

$$E = \frac{1}{2} \sum_{m=1}^{M_p} \sum_{i=1}^{N_L} [t_i^{(m)} - a_i^{(m)}(L)]^2, \quad (7.5)$$

where M_p is the total number of training patterns and N_L is the number of neurons in the last layer (or the number of output elements).

Using chain rule, the updating step can be expressed as

$$\Delta w_{ij}^{(m)}(l) = -\mu \frac{\partial E}{\partial w_{ij}^{(m)}(l)} \quad (7.6)$$

$$= -\mu \frac{\partial E}{\partial a_i^{(m)}(l)} \frac{\partial a_i^{(m)}(l)}{\partial w_{ij}^{(m)}(l)} \quad (7.7)$$

$$= -\mu \delta_i^{(m)} \frac{\partial a_i^{(m)}(l)}{\partial u_i^{(m)}(l)} \frac{\partial u_i^{(m)}(l)}{\partial w_{ij}^{(m)}(l)}. \quad (7.8)$$

From equation (7.3) we have,

$$\frac{\partial a_i^{(m)}(l)}{\partial u_i^{(m)}(l)} = f'(u_i^{(m)}(l)) \quad \text{and} \quad (7.9)$$

$$\frac{\partial u_i^{(m)}(l)}{\partial w_{ij}^{(m)}(l)} = a_j^{(m)}(l-1). \quad (7.10)$$

From equation (7.8) and (7.10), we have

$$\Delta w_{ij}^{(m)}(l) = \mu \delta_i^{(m)}(l) f'(u_i^{(m)}(l)) a_j^{(m)}(l-1). \quad (7.11)$$

For the last layer, the partial derivative $\delta_i^{(m)}(L)$ can be expressed as

$$\delta_i^{(m)}(L) = \frac{\partial E}{\partial a_i^{(m)}(L)} = t_i^{(m)} - a_i^{(m)}(L). \quad (7.12)$$

Again using chain rule the derivative can be expanded as

$$\delta_i^{(m)}(l) = \frac{\partial E}{\partial a_i^{(m)}(l)} = - \sum_{j=1}^{N_{l+1}} \frac{\partial E}{\partial u_j^{(m)}(l+1)} \underbrace{\frac{\partial u_j^{(m)}(l+1)}{\partial a_i^{(m)}(l)}}_{w_{ij}^{(m)}(l+1)} \quad (7.13)$$

$$= - \sum_{j=1}^{N_{l+1}} \frac{\partial E}{\partial a_i^{(m)}(l+1)} \frac{\partial a_i^{(m)}(l+1)}{\partial u_j^{(m)}(l+1)} w_{ij}^{(m)}(l+1) \quad (7.14)$$

$$= - \sum_{j=1}^{N_{l+1}} \delta_j^{(m)}(l+1) f'(u_j^{(m)}(l+1)) w_{ij}^{(m)}(l+1), \quad (7.15)$$

where $l = L - 1, \dots, 1$.

Then the weight updating can be written as

$$\Delta w_{ij}^{(m+1)}(l) = w_{ij}^{(m)} + \mu \delta_i^{(m)}(l) f'(u_i^{(m)}(l)) a_i^{(m)}(l-1) \quad (7.16)$$

Now,

$$\delta_i^{(m)}(l) = - \sum_{j=1}^{N_{l+1}} \delta_j^{(m)}(l+1) f'(u_j^{(m)}(l+1)) w_{ij}^{(m)}(l+1) \quad (7.17)$$

$$\delta_i^{(m)}(L) = t_i^{(m)} - a_i^{(m)}(L). \quad (7.18)$$

Then we can evaluate,

$$\delta_i^{(m)}(L-1) = - \sum_{j=1}^{N_{L-1}} \delta_j^{(m)}(L) f'(u_j^{(m)}(L)) w_{ij}^{(m)}(L) \quad (7.19)$$

$$\delta_i^{(m)}(L-2) = - \sum_{j=1}^{N_{L-2}} \delta_j^{(m)}(L-1) f'(u_j^{(m)}(L-1)) w_{ij}^{(m)}(L-1) \quad (7.20)$$

\vdots

$$\delta_i^{(m)}(1) = - \sum_{j=1}^{N_1} \delta_j^{(m)}(2) f'(u_j^{(m)}(2)) w_{ij}^{(m)}(2). \quad (7.21)$$

Since $\delta_i^{(m)}(l)$ can be recursively computed starting from $\delta_i^{(m)}(L) = t_i^{(m)} - a_i^{(m)}(L)$ at the last layer, the term Back Propagation is used in the sense that the error is propagated from the last layer.

Update Frequency

In the last section, the update of the weights are done when each input target pair or patterns are presented to the network. This type of update is suitable for some on-line problems when real-time training is necessary. However, the update can be done when all the patterns have been presented to the network. This type of update is called Block Update and is known to be more robust because the weights are updated when the training patterns are averaged. Thus for the load estimation when real-time training is not necessary, Block Update method will be used. The Block Update method can be generalized by the following modification to the update equation in equation (7.16) as

$$\Delta w_{ij}^{(k+1)}(l) = w_{ij}^{(k)}(l) + \mu \sum_{m=1}^{M_p} \delta_i^{(m)}(l) f'(u_i^{(m)}) a_j^{(m)}(l-1), \quad (7.22)$$

where k is the block update step and M_p is the total number of training patterns.

Chapter 8

Load Estimation Problem - Linear and Non-linear Least Squares Approach

In this chapter, we will discuss the use of nonlinear least square approach in solving the load estimation problem. This problem is motivated by the NASA F-111 data set. The objective is to use the calibrated data to predict the load experienced by the wing structure during the in-flight stage.

Load Conditions

Here we will define the notion of loading conditions in the calibration and in-flight stage of the load estimation problem. During calibration, loads are applied on different parts of the wing while the corresponding set of n gages' readings are recorded. If we assume that the number of different loading condition is m , we can define the q -th loading condition as the set of all individual point loads as $\mathcal{L}C^{(q)}$, where $1 \leq q \leq m$.

Now the individual load applied on the wing structure $l = (l_1^{(q)}, \dots, l_{J^{(q)}}^{(q)})' \in \mathbb{R}^{J^{(q)} \times 3}$ can be written as

$$l_i^{(q)} = [s_i, s_i x_i, s_i y_i], \quad i = 1, \dots, J^{(q)}, \quad (8.1)$$

where (x_i, y_i) is the relative position of the load on the wing structure and $J^{(q)}$ is the total number of individual load points for the q -th load condition $\mathcal{L}C^{(q)}$.

Single Equivalent Load

As discussed in chapter 2, the first, second and third elements of the load vector l_i represents the shear, the moment and the torque. In order to simplify the set up of the system of linear equations, the notion of equivalent single load is used. Thus the equivalent single load $\mathbf{L}^{(q)} \in \mathbb{R}^{1 \times 3}$ corresponding to the q -th load condition $\mathcal{L}C^{(q)}$ can be defined as

$$\mathbf{L}^{(q)} = \sum_{i=1}^{J^{(q)}} l_i \quad (8.2)$$

$$= \sum_{i=1}^{J^{(q)}} [s_i, s_i x_i, s_i y_i]. \quad (8.3)$$

For simplicity, we will concentrate on the shear s element of the load vector since the use of the other elements will lead to similar procedures and results. Now the shear load of $\mathcal{L}C^{(q)}$ can be written as

$$L^{(q)} = \sum_{i=1}^{J^{(q)}} s_i. \quad (8.4)$$

Then the calibration stage can be summarized by the following steps.

Calibration Stage Procedure

- Known loads, which are defined as a specific load condition $\mathcal{L}C^{(q)} = \{l_i^{(q)} | 1 \leq q \leq J^{(q)}\}$, are placed on different parts of the wing structure as shown in

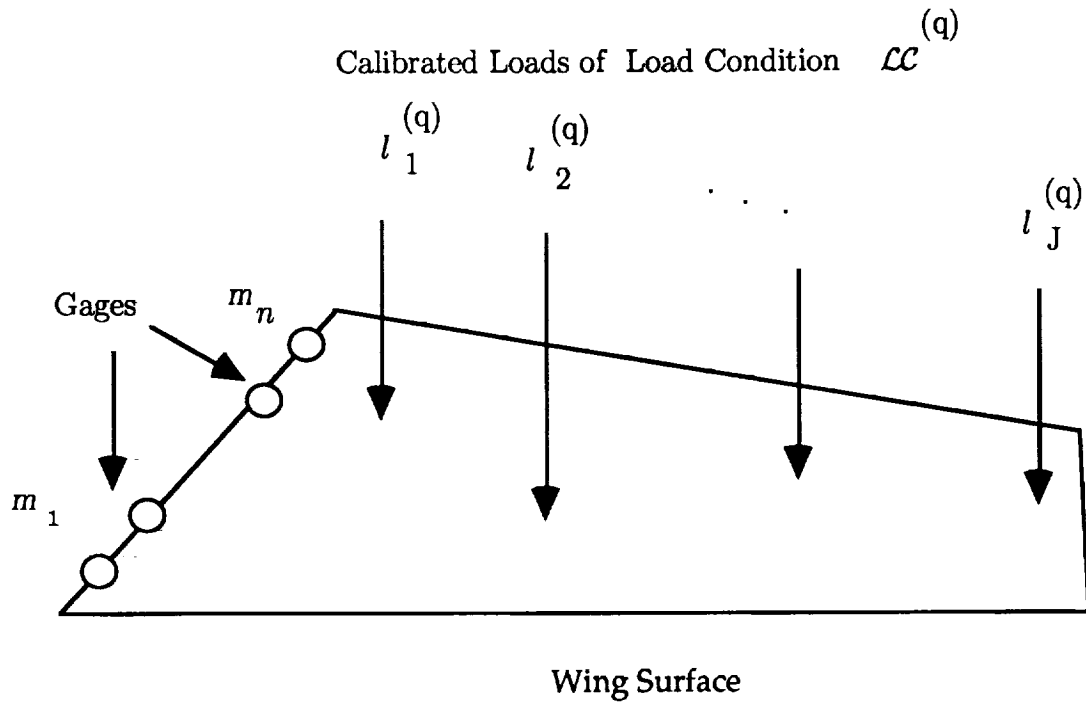


Figure 8.1: Wing Structure in Calibration Stage of Load Condition $\mathcal{LC}^{(q)}$

fig. (8.1). Usually, the specific choice of $l^{(q)}$ are due to actual loading conditions of interest (such as the loads on the leading edge of the wing is more pronounced).

- The magnitude of the known calibrated loads are simultaneously increased (we denote $L_j^{(q)}$ as the equivalent load as described in equation (8.4) for $1 \leq j \leq k$ where k is the total number of loads of varying magnitude) and the corresponding gage readings are recorded (in the F111 Calibration data example, an approximate $k = 50$ points are recorded). Fig 8.2 shows a typical gage vs. load response curve obtained from a specific load condition $\mathcal{LC}^{(q)}$.
- Loads are relocated so that a new set of data corresponding to a different loading condition $\mathcal{LC}^{(q+1)}$ can be recorded.

- For the F111 data case, there are $m = 9$ load conditions while each sets contains 50 data points.

As seen in fig. 8.2, high degree of nonlinearities are present in the data. As a result, the conventional linear methods are inadequate for accurate estimation. However, we will still discuss the linear approach in estimating the load as a comparison to the neural network approach. In order to understand the load problem, the load estimation procedure can be summarized in fig (8.3).

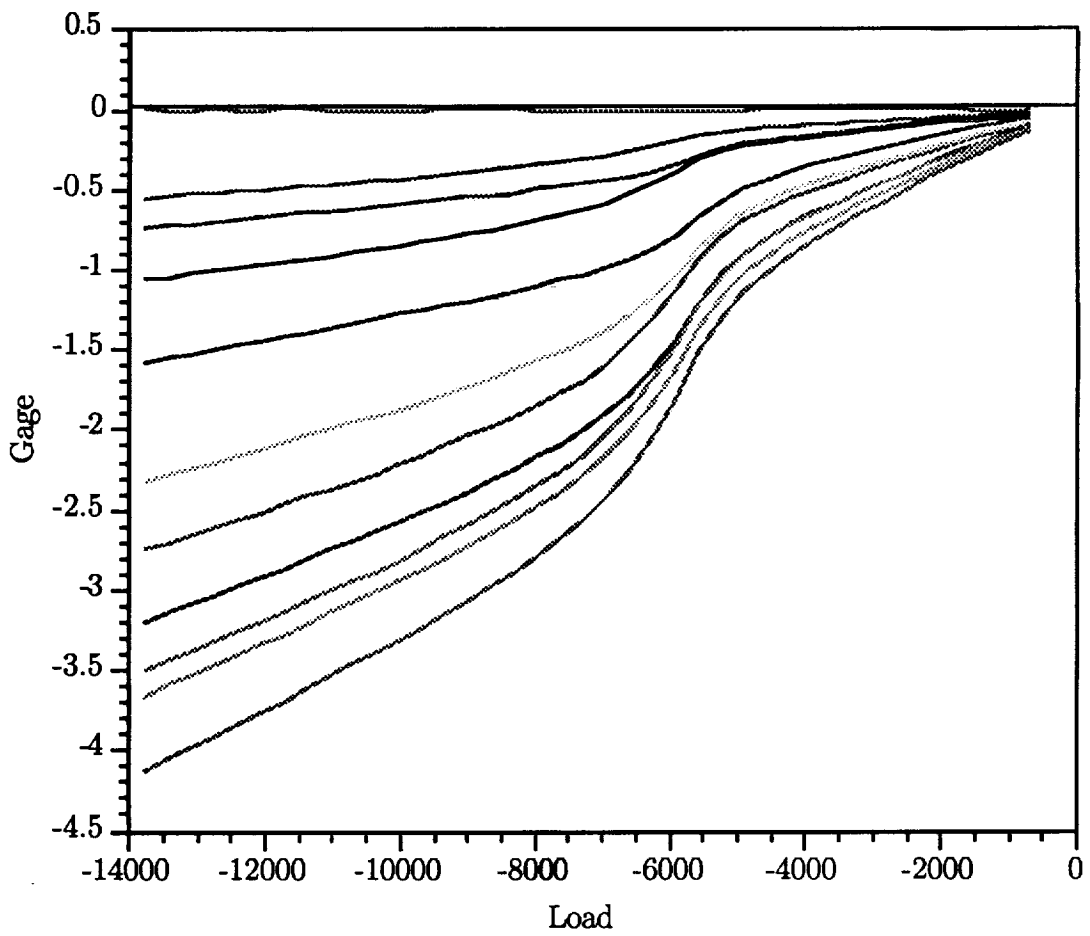


Figure 8.2: Gage vs. Load measurement from F-111 Data in Calibration Stage

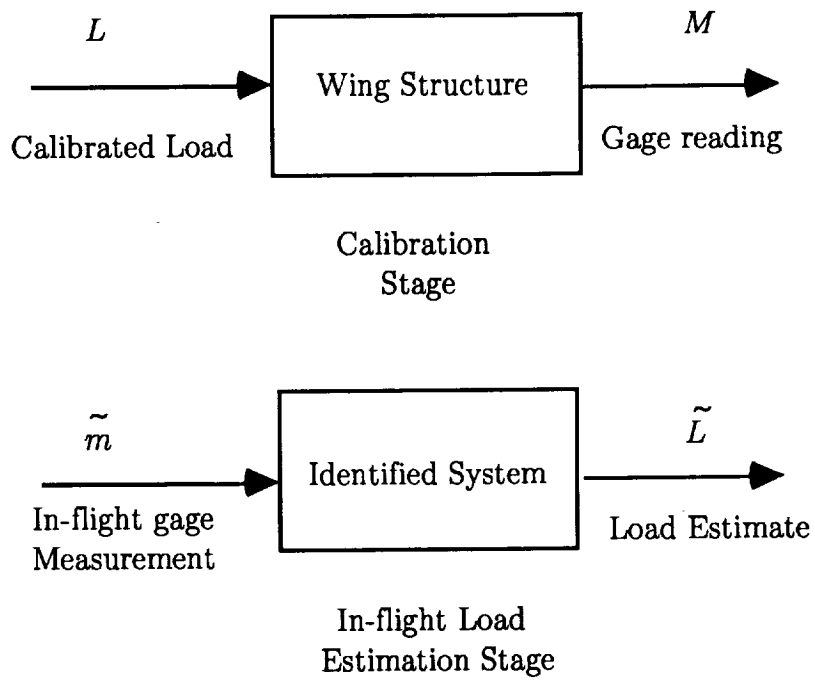


Figure 8.3: The Load Estimation Problem

8.1 Linear Approach

The conventional linear approach attempts to find a relationship between the gage and load data ($Mb \approx L$) in a linear least square sense. It solves for the minimizer \hat{b} of the system of linear equations such that

$$\min_{\hat{b}} \|Mb - L\| = \|M\hat{b} - L\|. \quad (8.5)$$

Since the gage matrix $M \in \mathbf{R}^{m \times n}$ and the load vector $L \in \mathbf{R}^m$, the system of equations can be written as

$$Mb \approx L$$

$$\begin{bmatrix} m_{11}^{(1)} & m_{12}^{(1)} & \cdots & m_{1n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k1}^{(1)} & m_{k2}^{(1)} & \cdots & m_{kn}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i1}^{(m)} & m_{i2}^{(m)} & \cdots & m_{in}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k1}^{(m)} & m_{k2}^{(m)} & \cdots & m_{kn}^{(m)} \end{bmatrix} b \approx \begin{bmatrix} L_1^{(1)} \\ \vdots \\ L_k^{(1)} \\ \vdots \\ L_1^{(m)} \\ \vdots \\ L_k^{(m)} \end{bmatrix},$$

where

$$M' = \begin{bmatrix} m_{11}^{(1)} & \cdots & m_{k1}^{(1)} & \cdots & m_{11}^{(m)} & \cdots & m_{k1}^{(m)} \\ m_{12}^{(1)} & \cdots & m_{k2}^{(1)} & \cdots & m_{12}^{(m)} & \cdots & m_{k2}^{(m)} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{1n}^{(1)} & \cdots & m_{kn}^{(1)} & \cdots & m_{1n}^{(m)} & \cdots & m_{kn}^{(m)} \end{bmatrix}, \quad (8.6)$$

$$L' = \underbrace{[L_1^{(1)} \cdots L_k^{(1)}]}_{\mathcal{L}^{(1)}} \cdots \underbrace{[L_1^{(m)} \cdots L_k^{(m)}]}_{\mathcal{L}^{(m)}}, \quad (8.7)$$

m is the number of loading conditions,

k is the number of points per loading condition, and n is the number of gages.

The solution \hat{b} to the linear system can be expressed as

$$\hat{b} = M^+L, \quad (8.8)$$

where M^+ is the Moore-Penrose pseudo inverse of M .

The corresponding load estimate \tilde{L} can then be computed as

$$\tilde{L} = \tilde{m}'M^+L, \quad (8.9)$$

where \tilde{m} is a $n \times 1$ vector from the in-flight gage reading.

8.2 Non Linear Least Squares Approach

Since nonlinearities are present in the data set, the load problem of solving the nonlinear system($f(M) \approx L$) can be written as

$$\min_b ||f(M) - L||, \quad (8.10)$$

where $f(M, b)$ is a nonlinear function of M and b .

8.2.1 Neural Network Approach

In solving the problem $\min_b ||f(M) - L||$ where $f(\cdot)$ is a nonlinear function, the neural network approach can be used as an approximator of the function $f(\cdot)$.

First, the structure of the NN has to be chosen for this problem. Since there is no definite approach in choosing the number of layers or the number of neurons for the network, we need to use a trail-and-error approach to determine a possible

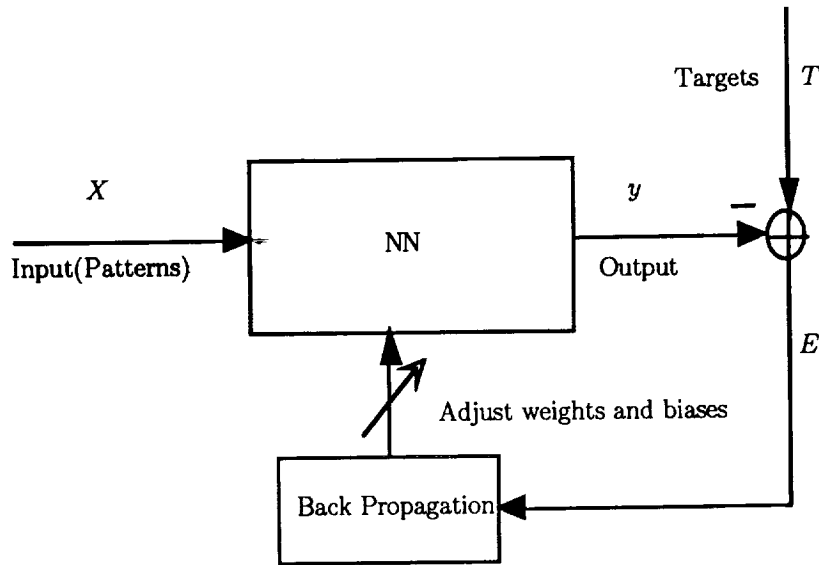


Figure 8.4: The Back Propagation Training Process

optimal selection. In chapter 9, we will discuss some aspects of determining the number of neurons in a hidden layer. In our simulations, two hidden layers are used for it is found to be adequate in most cases. The Back Propagation(BP) training procedure, which can be illustrated in fig. (8.4), is used in the learning process.

Once the Network is properly trained(i.e. when an error criterion has been reached), the NN can be used to estimate the wing load during the in-flight stage. The prediction process is fast enough for real-time application since all the time consuming computation or training can be done in the ground calibration phase. Fig (8.5) shows that a load estimate can be obtained by passing the in-flight gage data to the trained Neural Network.

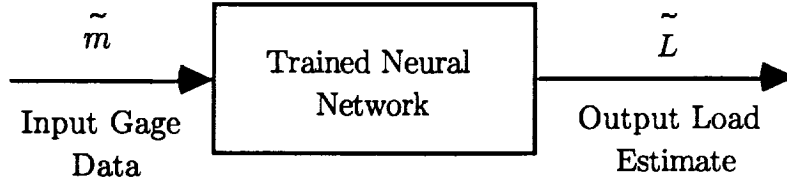


Figure 8.5: The Load Prediction by a Trained Neural Network

8.3 Simulation Results and Observations

In order to evaluate the performance of the trained Neural Network, the NN has to be tested using existing calibration data. The sigmoid function used for this example is defined in (7.2), where we set $\alpha = 1$. Although we have $m = 9$ sets of loading conditions, we only used $m = 5$ sets for calibration and training. The remaining load conditions are used to validate and quantify the performance of the network. After the training is completed, the NN is tested with three different sets of loading conditions. In this simulation, the performance of the NN and the linear methods are compared by computing the relative error which is denoted by

$$\bar{e}_{NN} = \left\| \frac{L_{true} - L_{NN,estimate}}{L_{true}} \right\| \quad (8.11)$$

and similarly the relative error of the linear approach can be denoted as

$$\bar{e}_{LS} = \left\| \frac{L_{true} - L_{LS,estimate}}{L_{true}} \right\|, \quad (8.12)$$

where

L_{true} is the true calibrated load used as a verification,
 $L_{NN,estimate}$ is the estimated load using Neural Network approach, and
 $L_{LS,estimate}$ is the estimated load using Linear approach.

Three performance analysis plots in fig. 8.6, 8.7 and fig. 8.8 show \bar{e}_{NN} and \bar{e}_{LS} vs. load test samples.

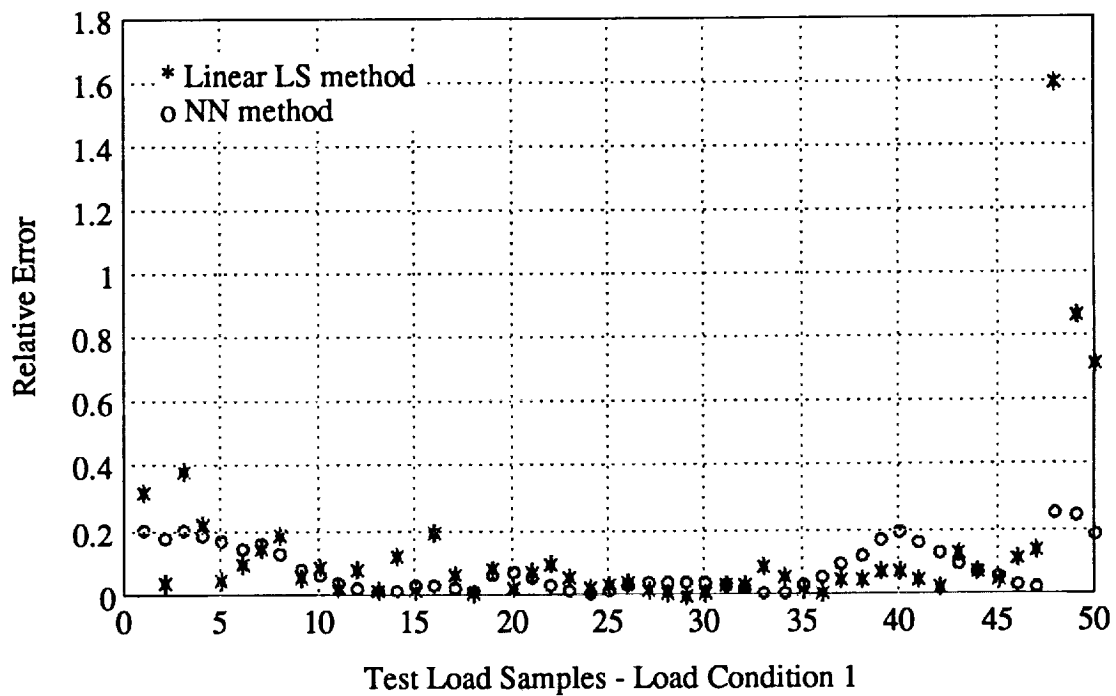


Figure 8.6: Relative Error vs. Test Load Samples for NN and LS approach(Load Condition 1)

C.J.

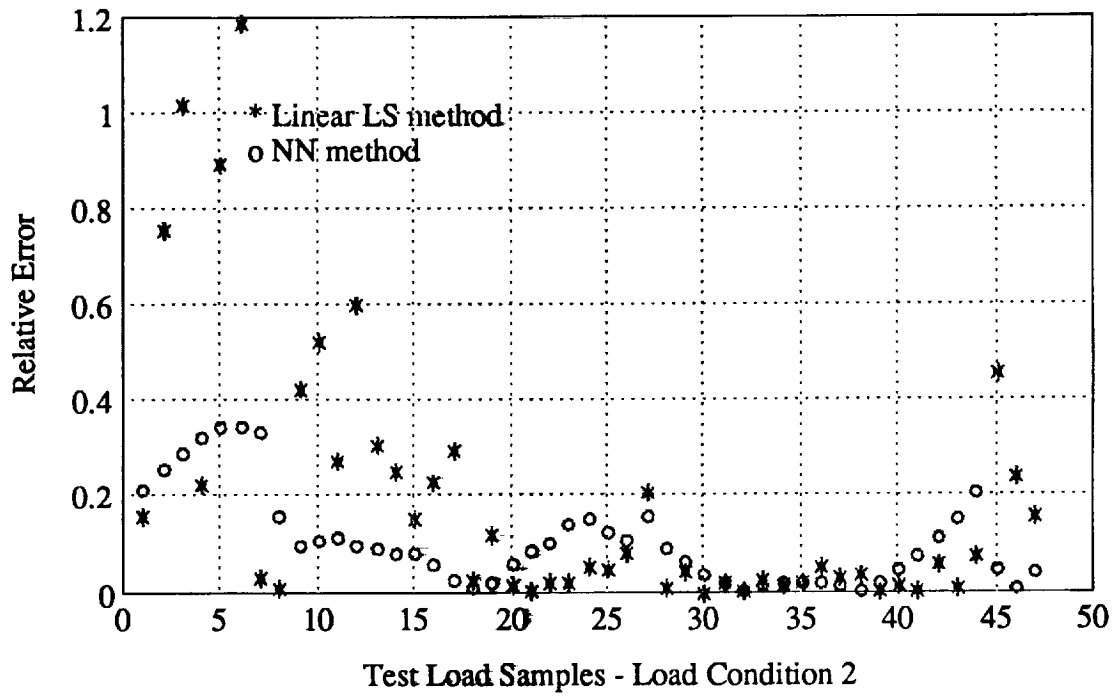


Figure 8.7: Relative Error vs. Test Load Samples for NN and LS approach(Load Condition 2)

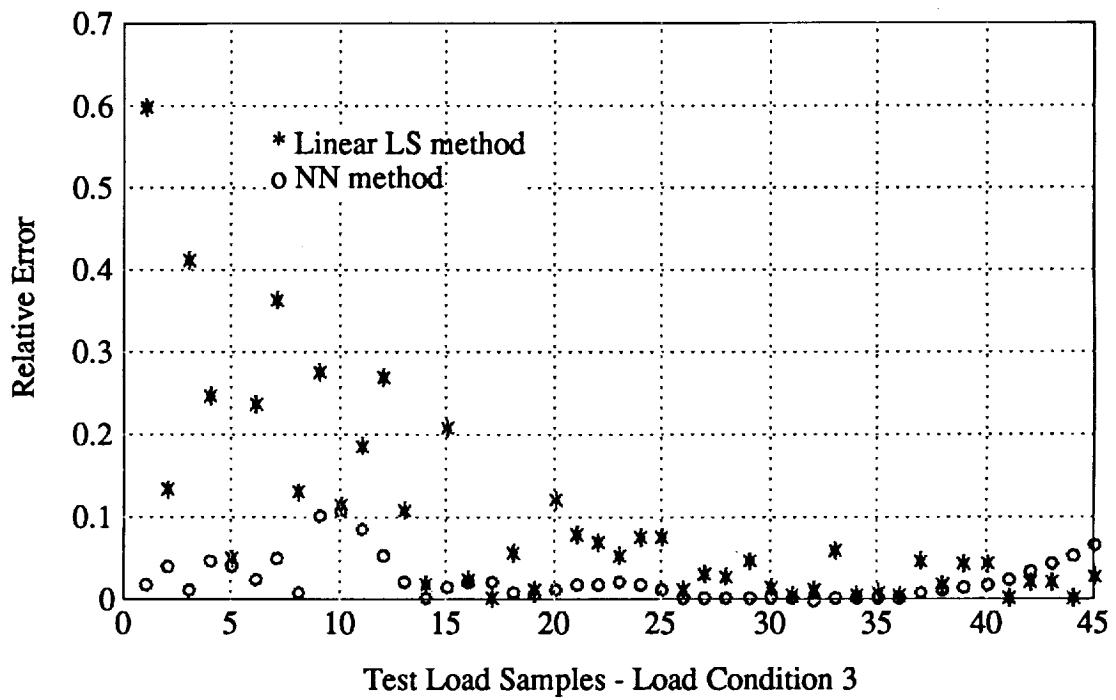


Figure 8.8: Relative Error vs. Test Load Samples for N and LS approach(Load Condition 3)

As observed in fig. 8.6,8.7, and fig. 8.8, the simulation results from the Neural Network approach have better performance in most sample cases. In addition large errors occurred in using linear LS method for some specific data samples, whereas the errors from the NN Approach are all relatively small. In general, the relative error of NN are more or less confined in the range of 20 ~ 30% or below. In addition, for the linear LS method, large error occurred in some specific examples. This seems to be a verification of the general superiority of the NN to the LS method when nonlinearities are present in the system. Moreover, NN approach performance will not deteriorate even when the data is actually linear for NN can handle both linear and nonlinear systems.

8.4 Distributed Load Estimation Problem

In the past section 8.2.1, the solution for the single load estimate are presented. Single load estimate is the equivalent load estimate which consists of a single load point $L = (S, Sx, Sy)$, where S denote the equivalent shear load and (x_i, y_i) denote the relative location of the equivalent load on the wing surface. It provides only the location and the magnitude of the equivalent load (or equivalently the shear, moment and torque components of the Load L). In real life situations, loads are usually distributed on different locations of the wing structure. Thus the knowledge of the single equivalent point load is not adequate in providing information as how the load is distributed on the wing. This lack of information might becomes undesirable in dangerous situations such as when a large load is concentrated on a specific region of the wing structure. As a result, the estimate of the distributed load is highly relevant and useful since it might provide an "early warning signal" for the presence of structure overloading.

Fig. 8.9 shows the difference between the single load and the distributed load representation. It also shows how the wing structure can be partitioned into regions of interest.

Distributed load estimate not only provides the estimate of the single equivalent load, but also provides the positions and magnitudes of all the forces in specific regions of the wing.

Distributed Load Problem The distributed load problem selects the right pattern $q = q^*$ or $\mathcal{L}^{(q^*)} = \{l_j^{(q^*)} | 1 \leq j \leq J^{(q^*)}\}$ from the set $(\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \dots, \mathcal{L}^{(m)})$ which represents the pattern set of all possible flight conditions.

If we can solve this general problem, we will be able to identify the magnitude

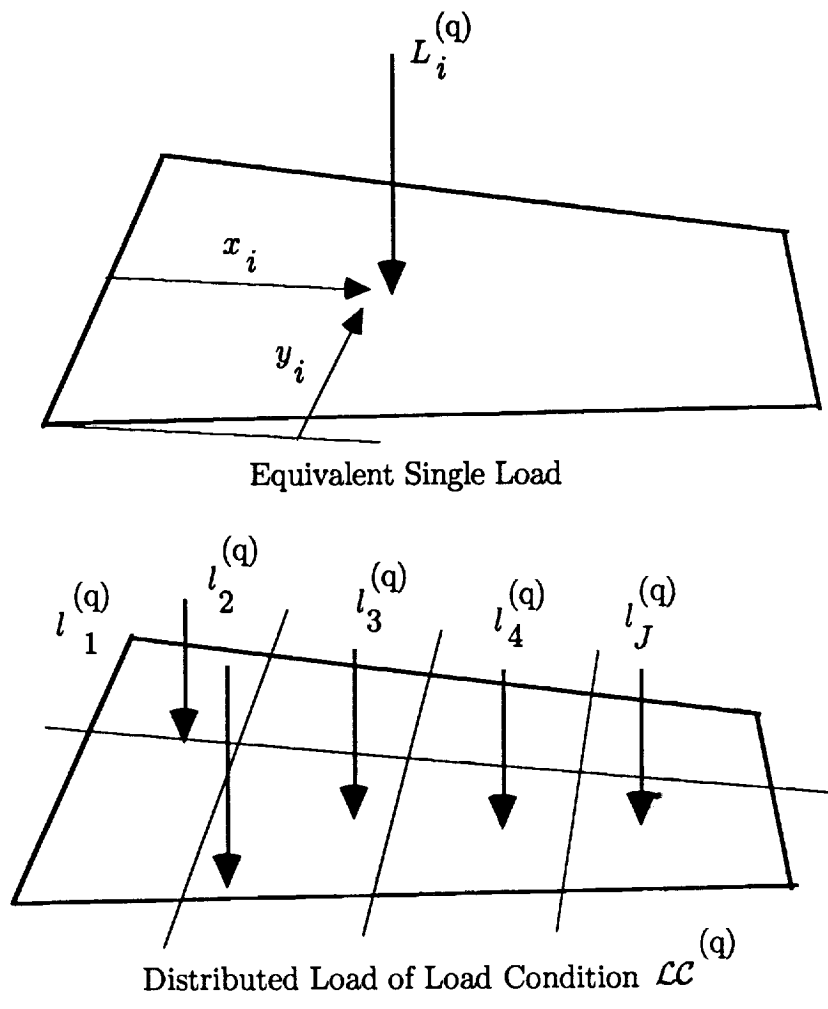


Figure 8.9: Single Equivalent Load and Distributed Load on a Wing Surface

of the specific force acting on a specific small region of the wing. As a result, the solution might provide valid information such as early warning signals for local overloading of the wing structure. However, such a task seems to be too ambitious. Alternatively, a sub-optimal approach can be adopted in order to retrieve as much information as possible for the in-flight data analysis. Since we only have a limited number of load patterns obtained in the calibration stage, the best we can do is to utilize those sets of data to achieve our goal. If each patterns in the calibration stage are valid in-flight characteristic loading conditions, we can use them as the training patterns. Once the neural network has been properly trained, we could(with some degree of accuracy) be able to identify which previously trained load pattern(or loading condition) the wing is experiencing during the in-flight session. Such a procedure seems useful since if we train the network with some specific conditions that represent a dangerous overloading situations, "early warning signals" might be available once this pattern is encountered during flight. Before apply the neural networks approach to the distributed load problem, the next section will discuss some basics of pattern recognition by neural networks.

8.5 Pattern Recognition by Neural Network

Pattern recognition by neural networks can be described as using nonlinear classifier to separate classes or patterns by drawing partitions between classes in the pattern space. Fig (8.10) shows a typical pattern space and how nonlinear separable cases can be partitioned into decision regions using nonlinear classifier such as NN.

Non Linear Separable Patterns

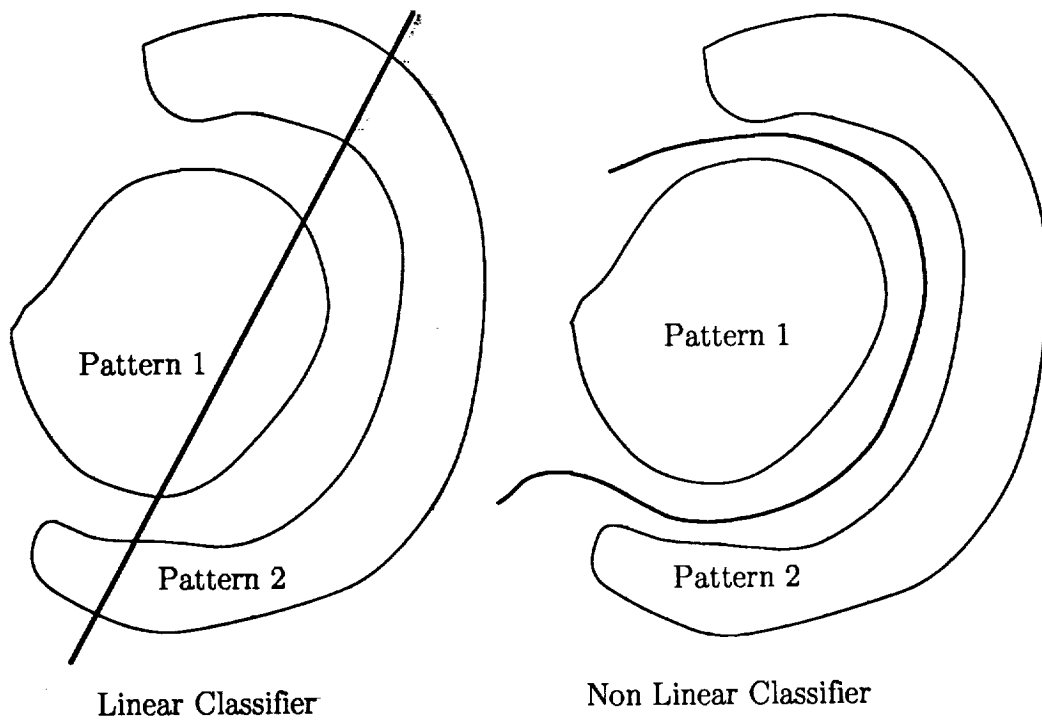


Figure 8.10: Classifier in Nonlinear Separable Patterns

The training input matrix and target matrix for this neural network are the gage matrix M and the target matrix T where

$$M' = \begin{bmatrix} m_{11}^{(1)} & \cdots & m_{k1}^{(1)} & \cdots & m_{11}^{(m)} & \cdots & m_{k1}^{(m)} \\ m_{12}^{(1)} & \cdots & m_{k2}^{(1)} & \cdots & m_{12}^{(m)} & \cdots & m_{k2}^{(m)} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{1n}^{(1)} & \cdots & m_{kn}^{(1)} & \cdots & m_{1n}^{(m)} & \cdots & m_{kn}^{(m)} \end{bmatrix} \quad \text{and} \quad (8.13)$$

$$T = \begin{bmatrix} 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{bmatrix}. \quad (8.14)$$

The neural network approach in solving the problem of identifying the patterns of loading can be summarized as follows:

- The structure of the neural network such as the number of layers and the number of neurons are chosen by in some(ad-hoc) manner(chapter 9 will discuss the issue of the selection of the number of neurons).
- M' and T are presented to the neural network and trained by back propagation(using block update frequency as described in the last chapter) until the maximum number of steps has exceeded or the the error criteria ϵ_f is is met as given by

$$E = \|T - \hat{T}\| \leq \epsilon_f, \quad (8.15)$$

where \hat{T} is the output of the neural network.

- A noisy version of the gage matrix which we define as \check{M}' and the target T are presented to the Neural Network. Then the second phase of training

continues until the error criteria is met or the maximum number of steps has been exceeded. The noisy version of the gage matrix can be expressed as

$$\check{M}' = [M', \check{M}'_1, \check{M}'_2], \quad (8.16)$$

$$\check{M}'_1 = [M' + e'_1], \quad \text{and} \quad (8.17)$$

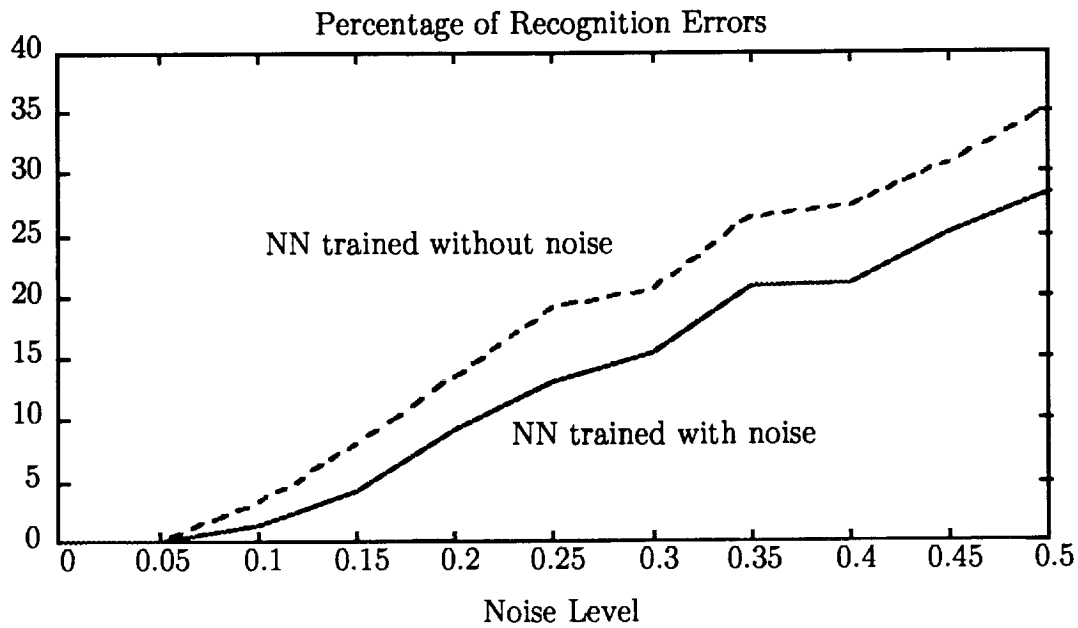
$$\check{M}'_2 = [M' + e'_2], \quad (8.18)$$

where e_1 and e_2 are $mk \times n$ normal random noise matrix with two different amount of noise variance.

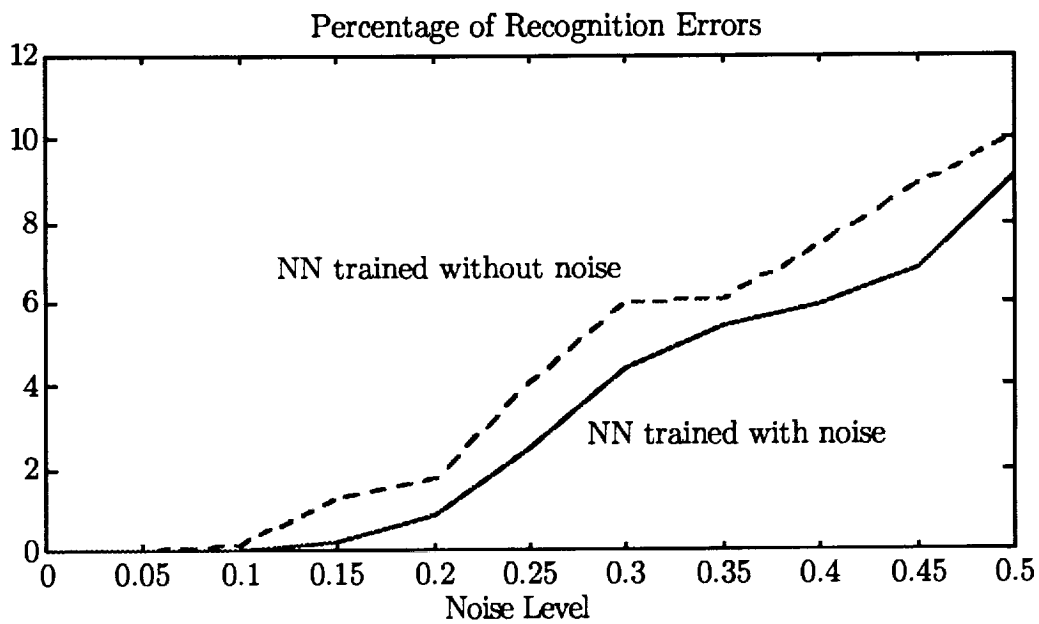
- The Neural Network is again trained with the noiseless matrix M' and T to guarantee accurate prediction of the patterns for noiseless cases.
- Once the NN has been trained, the in-flight gage data \tilde{m} will be passed into the NN. A "one" in the i -th position of the output target \hat{T} indicates the presence of the i -th load pattern.

8.6 Results and Observation

Results shown in fig. (8.11) and (8.12) shows reasonably good recognition performance even for noisy situations. For 10% noise level, a recognition error of less than 1% is observed.

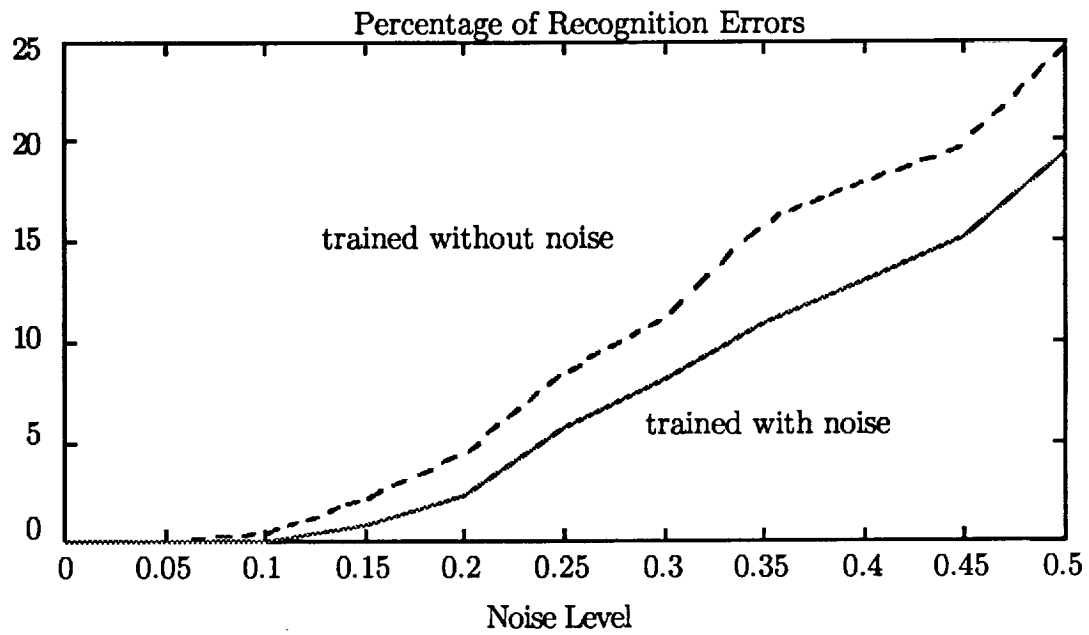


1 hidden layer NN for 3 classes recognition with 40 neurons

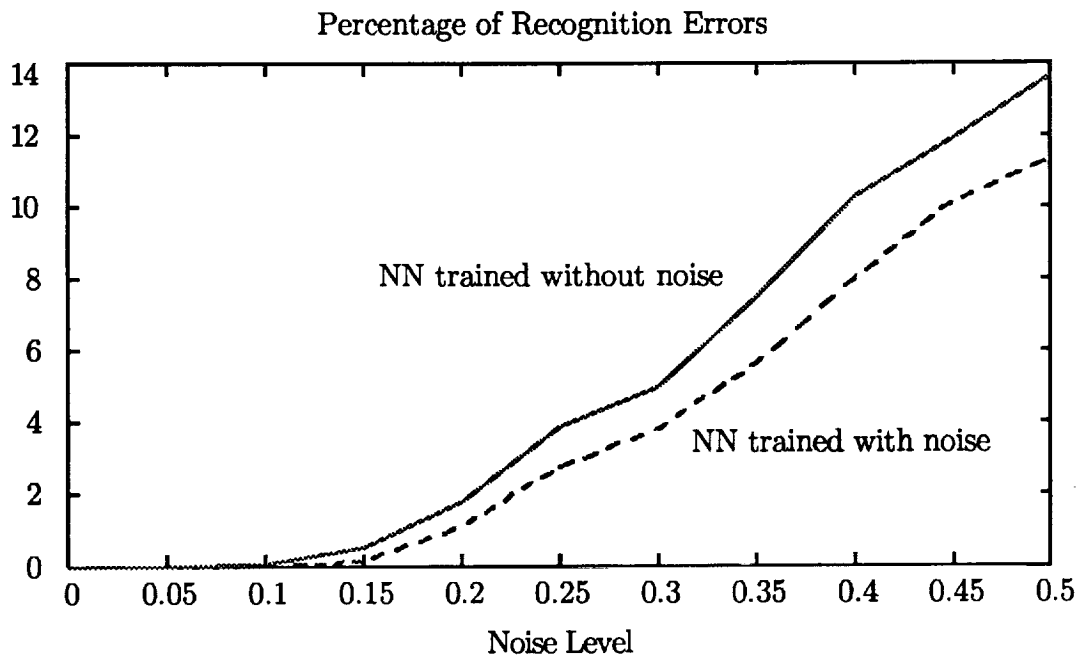


2 hidden layer NN for 2 classes recognition with 20, 2 neurons in layer 1, 2

Figure 8.11: Neural Network for Load Pattern Recognition



2 hidden layer NN for 3 classes recognition and 24, 4 neurons in layers 1, 2



2 hidden layer NN for 4 classes recognition 32, 8 neurons in layer s1, 2

Figure 8.12: Neural Network for Load Pattern Recognition

Chapter 9

Performance Analysis and Design Criteria of Neural Networks

In this chapter, we will discuss the issue of design criteria and the performance aspects of neural networks. There are a number of parameters determining the performance (accuracy) of the neural networks in estimating the targets (loads). The parameters can be listed as follows.

- $\epsilon \equiv$ error criteria for stopping the training process.
- $n \equiv$ number of iteration required for convergence.
- $p \equiv$ number of steps or time needed per iteration n .
- $\sigma_e \equiv$ the noise variance for the model.
- $\alpha \equiv$ the degree of nonlinearities in the sigmoid function.

- $n_d \equiv$ total number of nodes in the neural network.
- $n_l \equiv$ number of hidden layers in the neural network.

The two basic important terms which quantify the performance of the neural network can be defined as the prediction accuracy p_f and the time t required for training.

- $t = n * p \equiv$ total time needed for the training process.
- $p_f \equiv$ the load prediction accuracy.

The total time for training can be related as a function of the basic parameters as

$$t = f(\epsilon, \sigma_e, \alpha, n_d, n_l). \quad (9.1)$$

Similarly, the performance of the neural network can also be related as

$$p_f = g(\epsilon, \sigma_e, \alpha, n_d, n_l). \quad (9.2)$$

Fig. 9.1 shows the simulation results of the F-111 data set (the gage and load data are listed in Appendix B) in which the number of neurons in the first hidden layer is varied. The number of neurons in the second layer are fixed at 2. Since excessive neurons often causes overfitting of the model and insufficient neurons causes underfitting of the model, we expect that the errors will be high in these cases. Indeed from Fig. 9.1, we observed that the relative errors are both high in these cases. Thus we can conclude from this example, the optimum choice seems to lie in the vicinity of 14 neurons in the first hidden layer.

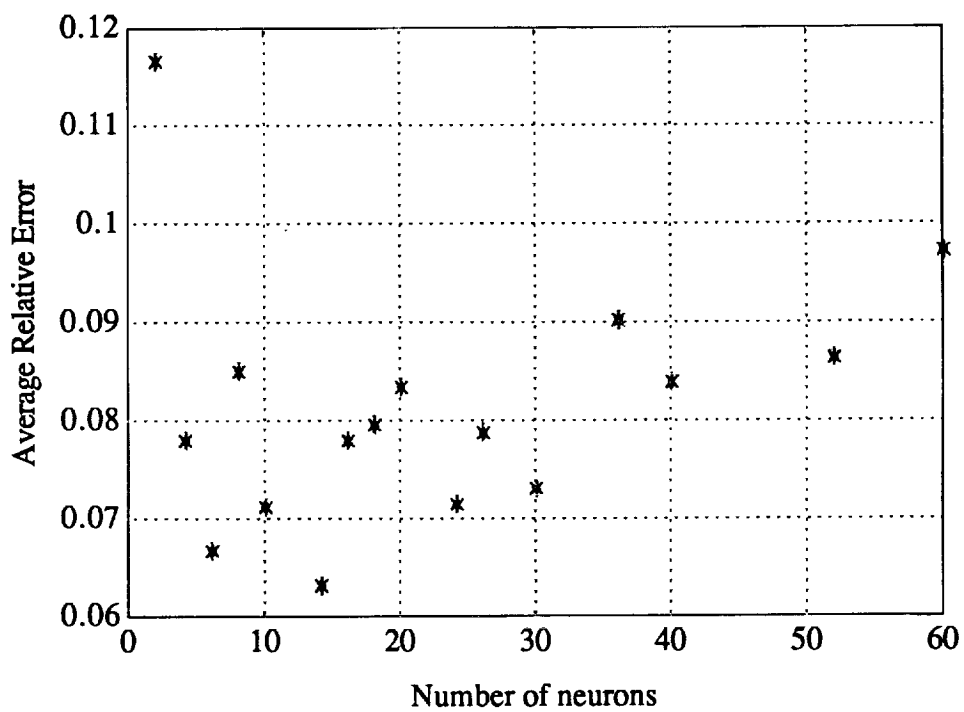


Figure 9.1: Average Relative Error vs. number of neurons in the first hidden layer

9.0.1 Generalization and Training Accuracy

The ability of the NN to estimate the targets with untrained data is sometimes referred to the term generalization performance [17]. The error criteria ϵ can also be interpreted as the training accuracy of the NN. Thus the performance of the NN depends more on the generalization performance rather than the training accuracy of the network. The issue of generalization and training accuracy can be viewed as an analogy to the issue of noise sensitivity and model accuracy as discussed in chapter 5. Although a large number of neurons can increase the training accuracy, the problem of overfitting might cause the network to have a poor generalization performance. Thus we need to select the right number of neurons such that the effects of overfitting can be avoided or reduced. In other words, we need to increase the number of neurons only if it is necessary.

In order to quantify the generalization performance of the network, an approach called "Cross Validation", in which calibration data are separated into two parts, can be used. One part of the data are used for training and the other part are used for validation. Thus from the validation results, one can quantify the generalization performance of the network.

9.0.2 SVD, CA and Collinearity Method for Reducing Neurons in a Layer

Recently, Xue in [29] proposed the use of SVD in determining the number of neurons in a hidden layer. By determining the effective rank of the weight matrix, one can effectively determine the number of neurons needed in a hidden layer.

Let the output of the i -th hidden layer be $A \in \mathbf{R}^m$, the input of the i -th layer

be $P \in \mathbf{R}^n$, where m is total number of neurons in the i -th hidden layer, and n is the total number of inputs from the i -th hidden layer, and $W \in \mathbf{R}^{m \times n}$ is the weight matrix of the i -th hidden layer.

Thus the output of the i -th layer can be written as,

$$A = WP, \quad (9.3)$$

where each row of W represent the weights of a neuron. If two rows of W are nearly identical, the output vector A will have two identical elements. Thus if two neurons give the same output for every input vector A , reducing one neuron from these two will give the same results. Based on this observation, if two rows of W are nearly linearly dependent, one can eliminate the extra neuron without affecting the performance of the network.

However in practice, the determination of the effective rank of a matrix is usually a sensitive issue. Xue in [29] proposed the use of an error criteria in determining the effective rank of the weight matrix W . However, the use of this method is not straightforward and clear. By using correspondence analysis(CA) and collinearity index(CO), we can provide an alternate and simple method to determine the rank of W and thus reduce the number of neurons in a layer.

Correspondence analysis and collinearity method are applied to our simulation example as shown in fig.(9.1). From fig. (9.2) and fig. (9.3) we can group the nearby point to form one single point. Using CA, the effective rank of the weight matrix W is 15 while using CO gives an effective rank of 13. Thus this simple example seems to agree with our simulation shown in fig. (9.1) that the vicinity of 16 neurons is in the region of the optimum choice of neurons. Although the use of the CA and CO is not straightforward as the grouping of points requires the right criteria of closeness. Different criteria of closeness between points sometimes

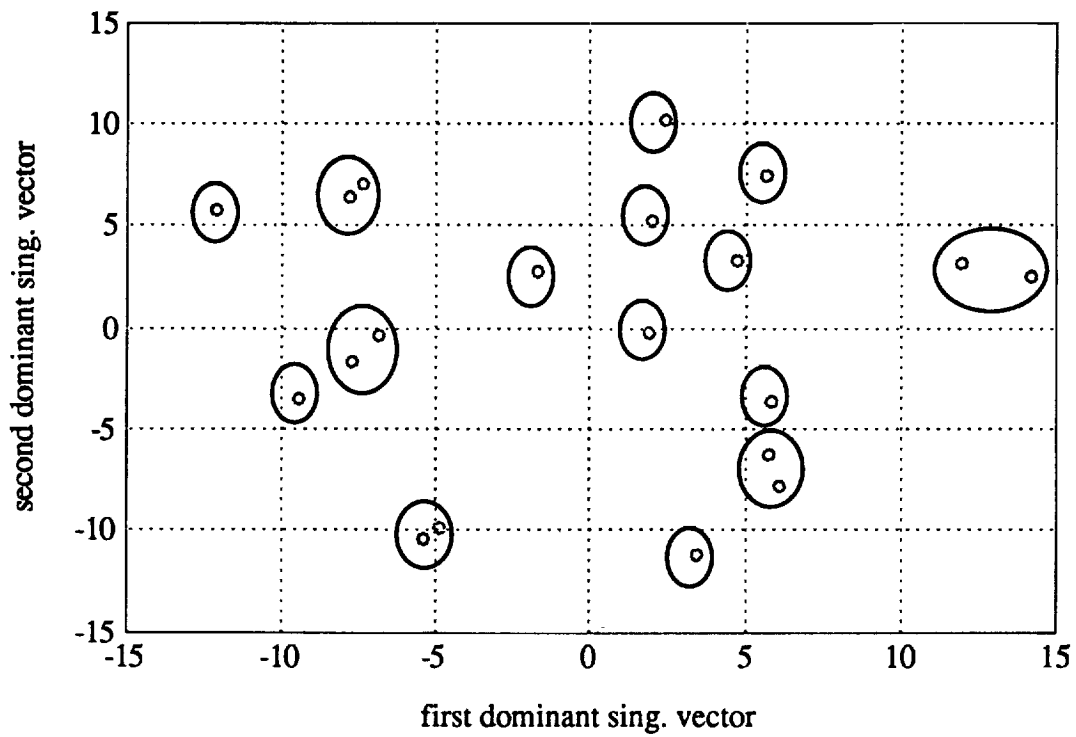


Figure 9.2: Correspondence Analysis of the weight matrix W

render different results. However we provided a practical and simple approach of combining SVD, CA, and CO in determining the possible optimal number of neurons in a layer.

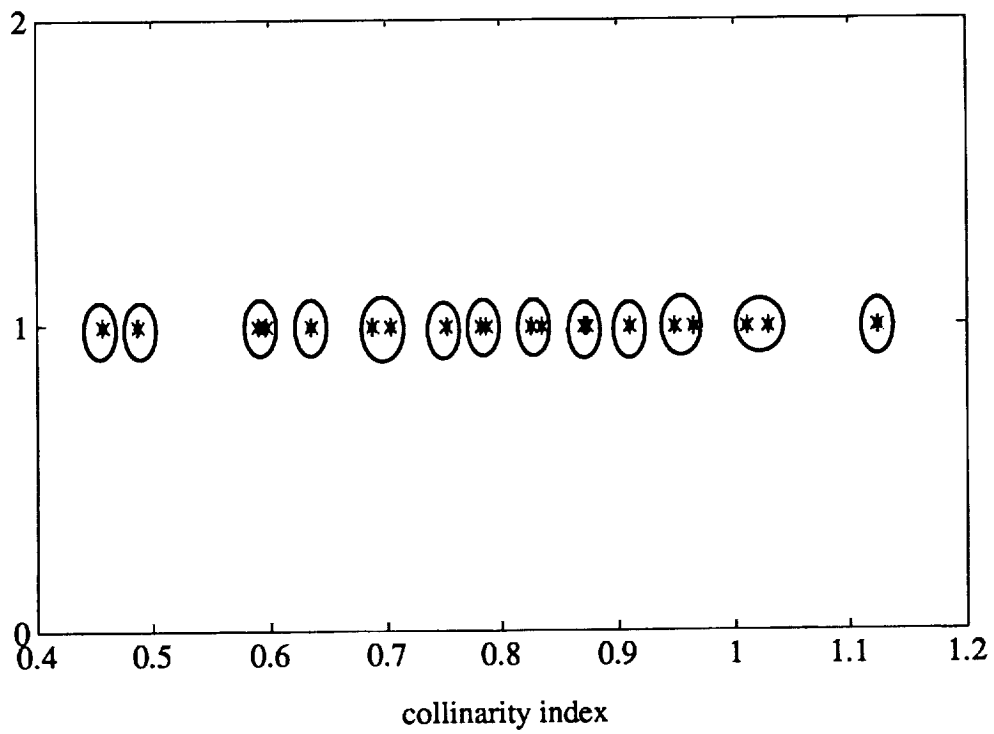


Figure 9.3: Collinearly Index of the weight matrix W

Chapter 10

Conclusions and Future Work

This thesis provided a study of parameter estimation using linear and nonlinear LS approach. We have discussed the issue of model accuracy/noise sensitivity and derived a bound for the change in the model accuracy (represented by its residual) as a function of the noise. The problem of the trade-off between model accuracy and noise sensitive is a major issue. This problem is also analogous to the issue of generalization and training accuracy of the neural network. We have also studied and derived the equivalency of TLS and CA. These two seemingly different topics in the field of signal processing and statistical analysis are shown to be equivalent with proper scaling and centering of the data matrix. In practice, the data matrix are sometimes nonlinear in nature and the use of linear LS methods is insufficient for accurate parameter estimations. Motivated by the nonlinearities in the NASA F-111 data matrix, we studied the possible use of nonlinear LS methods. Conventional nonlinear least square methods and spline approximation require the exact modeling of the functional relationship. The use of neural network provides a practical and robust approach in dealing

with highly nonlinear data. Using the NASA data as the training data, we simulated the estimation procedure by using neural network approaches. Although the computational effort in using neural networks is very high, the results we obtained are both useful and encouraging. Compared to the conventional linear LS approach, the neural network approach yields significantly better performance in load estimation. We also showed the ability of neural network in estimating the load conditions. The recognition of the trained load conditions during flight provided a possible method to provide "early warning signal" for the wing structure once a dangerous overloading condition is encountered.

We also studied the aspects of designing neural networks. The selection of the number of neurons in a hidden layer is crucial in designing a neural network with good generalization and estimation accuracy. Underfitting and overfitting yields poor performing network if the number of neurons is not correctly used. Work done by Xue in [29] proposed the use of effective rank of the weight matrix determined by SVD as a way to reduce the number of neurons in a layer. We found that Correspondence Analysis and Collinearity Index methods appeared to be equally effective in determining the effective rank of the weight matrix and thereby effectively reduces the number of neurons in a hidden layer.

We have studied both the linear and nonlinear methods for solving LS problem. As we expected, the linear methods are more analytically tractable while nonlinear methods need to be based on heuristic techniques. Thus more works is needed to fully understand the nonlinear methods including the neural network approaches.

Some of the unsolved problem and possible extension to our work include:

- Other types of NN other than MLP can be used for the Load Problem.

- Analytical analysis and design aspects of NN need to be considered.
- The use of a mix of NN types to recognize complex decision regions such as the "exclusive or(XOR)" operation(disconnected pattern space),which is not possible for MLP type NN.
- The use of basis function in defining the pressure distribution can be used instead of point loads.
- Computer simulated structural analysis can simulate gage-load relation but lack the ability to model the actual structure itself.
- A combination of computer analysis and structural testing(calibration) can be used to design a better and more accurate flight load estimations system.

Appendix A

Computer Codes for Neural Network Load Estimation

This appendix lists the matlab [6] computer codes for the single load estimation procedure. The procedures are summarized below:

1. `datreduct.m` — Reduces the number of points in a load condition;
2. `datred.m` — Select the load conditions to be reduced;
3. `datsetup.m` — Set up the gage/load data for training;
4. `trainNW.m` — Train the NN with noise;
5. `trainNN5.m` — Train the NN without noise;
6. `testset.m` — Test the NN performance.

```

datreduct.m
res = 1;
while (res==1),
    clg;
    hold off;
    Leff=b;
    mm=a;
    ss=size(a);ss=ss(1);ss=ss/2-3;ss=round(ss);
    Leff=Leff(1:ss);
    mm=mm(1:ss,:);
    plot(Leff,mm,'*');
    hold on;
    plot(Leff,mm);
    s=size(Leff);s=s(1);
    step=(max(Leff)-min(Leff))/s(1);
    %npoint=input('please input number of points ');

    disp('the maximum number of points you can input is ');
    disp(s);
    disp('please input the Leff value to be picked for your reduced
    data ');
    disp('e.g. input [-2000 -4000 -5000 -8000 -11000] for 5 points,
    but more than 1 point ');
    disp('or the name of the vector, e.g. v ');
    Lin=input('please enter now ');
    np=size(Lin');
    npoint=np(1);
    mmr=zeros(npoint,12);Leffr=zeros(npoint,1);
    for k=1:npoint,
        temp = 1;
        for i=1:ss,
            temp1 = Leff(i) - Lin(k);
            if sign(temp1) == -sign(temp),
                mmr(k,:)=mm(i,:);
                Leffr(k)=Leff(i);
                temp = Leff(i) - Lin(k);
            % disp(temp);
            else
                temp = Leff(i) - Lin(k);
        end;
    end;

    end;
    end;
    plot(Leffr,mmr);
    plot(Leffr,mmr,'o');
    hold off;
    gagemat=mmr;
    gagemat=-gagemat';

    % gagematrix normalization

```



```
mm=max(max(gagemat));

gagemat=gagemat/mm;
targets=Leff(i);
% targets normalization
targets=-targets';
tt=max(targets');
targets=targets/tt;

disp('enter 1 if you want to fit better ');
res=input('enter 0 if you are satisfied >>');

end,
```

datred.m

```
disp('load condition 1.3 ----- LC 1 ');
disp('load condition 11.3 ----- LC 2 ');
disp('load condition 13.3 ----- LC 3 ');
disp('load condition 17.2 ----- LC 4 ');
disp('load condition 21.4 ----- LC 5 ');
disp('load condition 24.8 ----- LC 6 ');
disp('load condition 28.4 ----- LC 7 ');
disp('load condition 9.3 ----- LC 8 ');
disp('load condition 91011 -----LC 9 ');

t(1,:)='dat01'; tr(1,:)='rdat01';
t(2,:)='dat02'; tr(2,:)='rdat02';
t(3,:)='dat03'; tr(3,:)='rdat03';
t(4,:)='dat04'; tr(4,:)='rdat04';
t(5,:)='dat05'; tr(5,:)='rdat05';
t(6,:)='dat06'; tr(6,:)='rdat06';
t(7,:)='dat07'; tr(7,:)='rdat07';
t(8,:)='dat08'; tr(8,:)='rdat08';
t(9,:)='dat09'; tr(9,:)='rdat09';

disp('enter the load conditions to be used for training ');
disp('For example [1 2 3 4 5] for selecting LC1 to LC5 ');
kk=input('enter ');

s=size(kk);s=s(2);

for i=1:s,
j=kk(i);
eval(t(j,:));
datereduct;
c=setstr(['rdat',int2str(j)]);
if j==1,
save rdat01 mnr /ascii /double;
save rdat01_L Leffr /ascii /double;
elseif j==2,;
save rdat02 mnr /ascii /double;
save rdat02_L Leffr /ascii /double
elseif j==3,
save rdat03 mnr /ascii /double;
save rdat03_L Leffr /ascii /double
elseif j==4,
save rdat04 mnr /ascii /double;
save rdat04_L Leffr /ascii /double
elseif j==5,
save rdat05 mnr /ascii /double;
save rdat05_L Leffr /ascii /double
elseif j==6,
save rdat06 mnr /ascii /double;
save rdat06_L Leffr /ascii /double
```

```
elseif j==7,  
save rdat07 mmr /ascii /double;  
save rdat07_L Leffr /ascii /double  
elseif j==8,  
save rdat08 mmr /ascii /double;  
save rdat08_L Leffr /ascii /double  
elseif j==9,  
save rdat09 mmr /ascii /double;  
save rdat09_L Leffr /ascii /double  
end;  
  
end;  
end.
```

datsetup.m

```
disp('Enter the number Load conditions to be included for
training');
disp('For example, enter [1 2 3 4 5] --- LC1 to LC 5 is used for
training');
temp3=input('enter>>');
temp4=size(temp3);
temp4=temp4(2);
gmmr=[];
gLeffr=[];
for i = 1:temp4,
ttt=temp3(i);
if ttt==1,
load rdat01;
load rdat01_L;
mmr=rdat01; Leffr=rdat01_L;
elseif ttt==2,;
load rdat02;
load rdat02_L;
mmr=rdat02; Leffr=rdat02_L;
elseif ttt==3,
load rdat03;
load rdat03_L;
mmr=rdat03; Leffr=rdat03_L;
elseif ttt==4,
load rdat04;
load rdat04_L;
mmr=rdat04; Leffr=rdat04_L;
elseif ttt==5,
load rdat05;
load rdat05_L;
mmr=rdat05; Leffr=rdat05_L;
elseif ttt==6,
load rdat06;
load rdat06_L;
mmr=rdat06; Leffr=rdat06_L;
elseif ttt==7,
load rdat07;
load rdat07_L;
mmr=rdat07; Leffr=rdat07_L;
elseif ttt==8,
load rdat08;
load rdat08_L;
mmr=rdat08; Leffr=rdat08_L;
elseif ttt==9,
load rdat09;
load rdat09_L;
mmr=rdat09; Leffr=rdat09_L;
end;

gmmr=[gmmr',mmr']';
```

```
gLeffr=[gLeffr',Leffr']';  
end;
```

trainNW.m

```
%           Train pattern recognition problem
%           load conditions are represented as patterns
%
%
% TRAIN WITH NOISE
%
%           This file trains a two hidden layer log-sigmoid/log-sigmoid
%           network to classify loading conditions patterns
%
%
%           The network is first trained on noise free loads.
%           It is then trained on loads with noise.
%           the network is then again trained on noise free loads.
%           The result is a network which can properly estimate
%           noise free loads and does a good job of estimating
%           loads with noise.
% LOAD PROBLEM
%=====
%
% INITIALIZE NETWORK ARCHITECTURE
%=====
[R,Q] = size(gagemat);
S1 =20;
S2 =2;

disp('the number of 1 st hidden layer neurons is ');
disp(S1);
res=input('enter y if you want to change it, otherwise push enter
>> ','s');
if res=='y',
S1=input('enter no of 1st layer neurons >>');
end;
disp('the number of 2 nd hidden layer neurons is ');
disp(S2);
res=input('enter y if you want to change it, otherwise push enter
>>','s');
if res=='y',
S2=input('enter no of 2nd layer neurons >>');
end;

[S3,Q] = size(targets);

[W1,B1] = nwlog(S1,R);
W2 = rands(S2,S1)*0.01;
B2 = rands(S2,1)*0.01;
W3=rands(S3,S2)*.01;
B3=rands(S3,1)*.01;
% TRAIN THE FUNCTION
%=====
```

```

% PROBLEM
% Get gage data to be trained
P = gmmr';
T = gLeffr';
% Normalize the data
nmm=max(max(abs(P)));
ntt=max(abs(T));
P=-P./nmm;
T=-T/ntt;

% TRAINING PARAMETERS
disp_freq = 200;
max_epoch = 16000;
err_goal = 0.01;
lr = 0.05;
lr_inc = 1.05;
lr_dec = 0.7;
momentum = 0.95;
err_ratio = 1.04;
% TRAINING WITHOUT NOISE
TP = [disp_freq max_epoch err_goal lr lr_inc lr_dec momentum
err_ratio];
[W1,B1,W2,B2,W3,B3,xx]=trainbpx(W1,B1,'logsig',W2,B2,...
'logsig',W3,B3,'logsig',P,T,TP);
% SAVE NETWORK TRAINED WITHOUT NOISE
=====
save pr1_w1 W1 /ascii /double
save pr1_b1 B1 /ascii /double
save pr1_w2 W2 /ascii /double
save pr1_b2 B2 /ascii /double
save pr1_w3 W3 /ascii /double
save pr1_b3 B3 /ascii /double
% TRAINING PARAMETERS
max_epoch = 3000;
err_goal = 0.06;
rand('uniform');
TP = [disp_freq max_epoch err_goal lr lr_inc lr_dec momentum
err_ratio];
% TRAINING WITH NOISE
for pass = 1:5

    fprintf('Pass = %.0f\n',pass);
    [dd1,dd2]=size(gagemat);
    %avegagval=norm(gagemat)/dd1/dd2;
    P = [gagemat, gagemat, ...
(gagemat + gagemat.*rand(R,Q)*0.1), ...
(gagemat + gagemat.*rand(R,Q)*0.2)];
    T = [targets targets targets targets];
    %size(T);

[W1,B1,W2,B2,W3,B3,xx]=trainbpx(W1,B1,'logsig',W2,B2,...

```

```

'logsig',W3,B3,'logsig',P,T,TP);
    end
rand('uniform');
%    PROBLEM
P    = gagemat;
T    = targets;
%    TRAINING PARAMETERS
max_epoch = 16000;
err_goal = 0.05;
%    TRAINING WITHOUT NOISE AGAIN
TP = [disp_freq max_epoch err_goal lr lr_inc lr_dec momentum
err_ratio];
[W1,B1,W2,B2,W3,B3,xx]=trainbpx(W1,B1,'logsig',W2,B2,...
'logsig',W3,B3,'logsig',P,T,TP);
%    SAVE NETWORK TRAINED WITH NOISE
%=====
save pr2_w1 W1 /ascii /double
save pr2_b1 B1 /ascii /double
save pr2_w2 W2 /ascii /double
save pr2_b2 B2 /ascii /double
save pr2_w3 W3 /ascii /double
save pr2_b3 B3 /ascii /double
%    SUMMARIZE RESULTS
%=====
A    = logsig(W3*logsig(W2*logsig(W1*P,B1),B2),B3);
SSE = sumsq(A-T);
fprintf('Final sum squared error without noise: %g.\n',SSE);

```


trainNN5.m

```
% Train Neural Networks problem
% load conditions are represented as patterns
% Using gage matrix of dataset
% This file trains a two hidden layers log-sigmoid/log-sigmoid
% network to classify loading conditions patterns
%
%
% The network is first trained on noise free loads.
% The result is a network which can properly classify
% noise free loading conditions and does a good job of
classifying
% loads with noise.
% LOAD PROBLEM
%=====
%

% INITIALIZE NETWORK ARCHITECTURE
%=====
[R,Q] = size(gagemat);
S1 =20;
S2 =2;

disp('the number of 1 st hidden layer neurons is ');
disp(S1);
res=input('enter y if you want to change it, otherwise push enter
>> ','s');
if res=='y',
S1=input('enter no of 1st layer neurons >>');
end;
disp('the number of 2 nd hidden layer neurons is ');
disp(S2);
res=input('enter y if you want to change it, otherwise push enter
>>','s');
if res=='y',
S2=input('enter no of 2nd layer neurons >>');
end;

[S3,Q] = size(targets);

[W1,B1] = nwlog(S1,R);
W2 = rands(S2,S1)*0.01;
B2 = rands(S2,1)*0.01;
W3=rands(S3,S2)*.01;
B3=rands(S3,1)*.01;
% TRAIN THE FUNCTION
%=====
% PROBLEM
% Get gage data to be trained
```

```

P = gmmr';
T = gLeffr';
% Normalize the data
nmm=max(max(abs(P)));
ntt=max(abs(T));
P=-P./nmm;
T=-T/ntt;

% TRAINING PARAMETERS
disp_freq = 10;
max_epoch = 16000;
err_goal = 0.005;
lr = 0.05;
lr_inc = 1.05;
lr_dec = 0.7;
momentum = 0.95;
err_ratio = 1.04;
% TRAINING WITHOUT NOISE
TP = [disp_freq max_epoch err_goal lr lr_inc lr_dec momentum
err_ratio];
[W1,B1,W2,B2,W3,B3,xx]=trainbpx(W1,B1,'logsig',W2,B2,...
'logsig',W3,B3,'logsig',P,T,TP);
% SAVE NETWORK TRAINED WITHOUT NOISE
%=====
save pr1_w1 W1 /ascii /double
save pr1_b1 B1 /ascii /double
save pr1_w2 W2 /ascii /double
save pr1_b2 B2 /ascii /double
save pr1_w3 W3 /ascii /double
save pr1_b3 B3 /ascii /double
% TRAINING PARAMETERS

%      SUMMARIZE RESULTS
%=====
A      = logsig(W3*logsig(W2*logsig(W1*P,B1),B2),B3);
SSE = sumsqr(A-T);
fprintf('Final sum squared error without noise: %g.\n',SSE);

```

```

testset.m
% This is to test the NN and Normal equation approach
% For load condition 1 to 9 the relative error is displayed
% The relative error is stored in errn=[error-Normal
Equation,error-NN,error-NN-trained with noise]
% error-NE ----relative error using normal equation approach
% error-NN ----relative error using NN and trained without noise
% error-NN-WN ---relative error using NN and trained with noise

disp('enter the Load condition to be tested');
in=input('enter>>');

for k=1:9
if k==in,
    eval(['dat0',int2str(in)]);
end;

end;

disp('enter 1 if the NN is trained with noise');
disp('enter 0 if the NN is trained without noise');
temp9=input('enter>>');

% Test with no noise trained network
load pr1_w1; W1=pr1_w1;
load pr1_b1; B1=pr1_b1;
load pr1_w2; W2=pr1_w2;
load pr1_b2; B2=pr1_b2;
load pr1_w3; W3=pr1_w3;
load pr1_b3; B3=pr1_b3;

if temp9 == 1,
load pr2_w1;
load pr2_b1;
load pr2_w2;
load pr2_b2;
load pr2_w3;
load pr2_b3;
end;

% Linear solutions
AA=P';
x=pinv(AA)*T';
res=norm(AA*x-T');
res=res*res;
sz1=size(b);
sz1=sz1(1);
sz1=round(sz1(1)/2);
a=a(1:sz1,:);

```

```

b=b(1:sz1,:);

%a=a(1:50,:);
%b=b(1:50,:);

Pt=-a'/nmm;
targetss=-b'/ntt;

%P=n2gf11;
%targetss=n2L11;

% A is the output of the NN network--or estimated load using NN
% P is the test gage values
% T is the targets or true load for testing

A = logsig(W3*logsig(W2*logsig(W1*Pt,B1),B2),B3);
if temp9 ==1,
A2 =
logsig(pr2_w3*logsig(pr2_w2*logsig(pr2_w1*Pt,pr2_b1),pr2_b2),pr2_b
3);
end;
error=A-targetss;

YY=Pt'*x;
errorN=YY'-targetss;

errorNn=errorN./targetss;
errorn=error./targetss;
errn=[errorNn',errorn'];

if temp9==1,
error2=A2-targetss;
errorn2=error2./targetss;
errn=[errorNn',errorn',errorn2'];
end;

%plot(abs(error),'*');

plot(abs(errn),'*');
grid;
end;

```

Appendix B

Calibration Data from NASA

F-111 Load Measurement

The reduced load vector $L \in \mathbb{R}^{50}$ and the corresponding gage matrix $M \in \mathbb{R}^{50 \times 12}$ are listed below.

-3,571	-0.23	-0.26	-0.12	-0.12	-0.45	-0.60	-0.45	-0.63	-0.69	-0.75	-0.75	-0.88
-5,561	-0.52	-0.46	-0.24	-0.23	-0.96	-1.25	-0.91	-1.22	-1.45	-1.54	-1.46	-1.71
-4,710	-0.34	-0.26	-0.16	-0.15	-0.66	-0.85	-0.59	-0.78	-0.99	-1.03	-0.96	-1.13
-7,010	-0.82	-0.78	-0.39	-0.39	-1.56	-2.02	-1.47	-1.99	-2.33	-2.49	-2.36	-2.75
-2,833	-0.32	-0.18	-0.15	-0.11	-0.67	-0.91	-0.53	-0.76	-1.11	-1.13	-0.98	-1.20
-3,544	-0.50	-0.27	-0.22	-0.16	-1.00	-1.33	-0.80	-1.13	-1.61	-1.67	-1.42	-1.74
-2,087	-0.17	-0.19	-0.08	-0.09	-0.39	-0.55	-0.39	-0.60	-0.65	-0.74	-0.76	-0.90
-2,653	-0.23	-0.25	-0.10	-0.12	-0.50	-0.72	-0.51	-0.78	-0.85	-0.98	-0.98	-1.16
-3,554	-0.15	-0.31	-0.08	-0.13	1.46	-0.45	-0.41	-0.58	0.00	-0.58	-0.67	-0.74
-4,571	-0.21	-0.43	-0.11	-0.19	1.46	-0.62	-0.57	-0.80	0.00	-0.81	-0.92	-1.01
-1,727	-0.10	-0.13	-0.06	-0.06	-0.21	-0.28	-0.22	-0.31	-0.32	-0.35	-0.37	-0.43
-1,755	-0.12	-0.14	-0.06	-0.06	-0.22	-0.30	-0.22	-0.32	-0.34	-0.37	-0.37	-0.43
-2,973	-0.19	-0.21	-0.10	-0.10	-0.37	-0.48	-0.37	-0.52	-0.56	-0.61	-0.62	-0.72
-4,191	-0.27	-0.31	-0.14	-0.15	-0.53	-0.71	-0.54	-0.75	-0.82	-0.90	-0.89	-1.04
-5,314	-0.39	-0.41	-0.19	-0.20	-0.75	-0.99	-0.74	-1.01	-1.14	-1.25	-1.20	-1.40
-7,019	-0.80	-0.91	-0.39	-0.42	-1.53	-2.02	-1.55	-2.12	-2.30	-2.53	-2.49	-2.88
-9,795	-1.03	-1.15	-0.50	-0.56	-1.94	-2.57	-1.96	-2.68	-2.94	-3.23	-3.15	-3.63
-11,669	-1.14	-1.29	-0.57	-0.63	-2.18	-2.88	-2.20	-3.02	-3.30	-3.65	-3.56	-4.11
-1,704	-0.10	-0.09	-0.04	-0.05	-0.21	-0.28	-0.20	-0.27	-0.31	-0.34	-0.32	-0.38
-1,719	-0.10	-0.10	-0.05	-0.05	-0.21	-0.27	-0.19	-0.26	-0.31	-0.33	-0.32	-0.38
-4,119	-0.28	-0.23	-0.13	-0.12	-0.54	-0.70	-0.50	-0.66	-0.82	-0.86	-0.82	-0.96
-5,210	-0.43	-0.31	-0.20	-0.17	-0.81	-1.04	-0.71	-0.94	-1.21	-1.25	-1.15	-1.36
-5,952	-0.70	-0.58	-0.32	-0.30	-1.31	-1.68	-1.20	-1.59	-1.95	-2.05	-1.91	-2.24
-8,565	-0.96	-0.94	-0.46	-0.47	-1.81	-2.36	-1.73	-2.34	-2.70	-2.92	-2.78	-3.23
-11,199	-1.15	-1.12	-0.55	-0.58	-2.17	-2.83	-2.07	-2.80	-3.25	-3.51	-3.33	-3.87
-13,116	-1.26	-1.26	-0.62	-0.64	-2.40	-3.13	-2.30	-3.13	-3.61	-3.91	-3.72	-4.32
-671	-0.07	-0.05	-0.03	-0.02	-0.15	-0.20	-0.11	-0.18	-0.24	-0.25	-0.23	-0.28
-1,397	-0.14	-0.09	-0.06	-0.05	-0.30	-0.41	-0.24	-0.36	-0.50	-0.51	-0.45	-0.55
-2,293	-0.24	-0.15	-0.11	-0.08	-0.51	-0.70	-0.41	-0.61	-0.85	-0.88	-0.77	-0.98
-3,191	-0.40	-0.22	-0.18	-0.13	-0.81	-1.09	-0.65	-0.92	-1.33	-1.37	-1.17	-1.43
-3,887	-0.63	-0.39	-0.28	-0.21	-1.25	-1.67	-1.03	-1.44	-2.00	-2.08	-1.80	-2.18
-4,660	-0.82	-0.62	-0.36	-0.32	-1.64	-2.20	-1.43	-2.00	-2.62	-2.75	-2.45	-2.94
-6,129	-0.98	-0.81	-0.44	-0.41	-1.99	-2.70	-1.76	-2.49	-3.22	-3.40	-3.06	-3.67
-7,045	-1.07	-0.89	-0.48	-0.45	-2.18	-2.98	-1.93	-2.74	-3.56	-3.76	-3.36	-4.04
-391	-0.02	-0.03	-0.01	-0.01	-0.06	-0.08	-0.06	-0.10	-0.09	-0.12	-0.13	-0.15
-384	-0.03	-0.03	-0.01	-0.01	-0.07	-0.09	-0.07	-0.12	-0.11	-0.13	-0.14	-0.17
-779	-0.06	-0.07	-0.03	-0.03	-0.14	-0.19	-0.14	-0.22	-0.23	-0.26	-0.28	-0.33
-1,718	-0.14	-0.16	-0.07	-0.07	-0.32	-0.45	-0.32	-0.50	-0.53	-0.61	-0.62	-0.75
-2,465	-0.21	-0.23	-0.09	-0.11	-0.47	-0.67	-0.48	-0.73	-0.78	-0.90	-0.91	-1.08
-3,216	-0.29	-0.32	-0.13	-0.15	-0.63	-0.91	-0.65	-0.98	-1.07	-1.23	-1.23	-1.45
-4,721	-0.76	-0.86	-0.34	-0.39	-1.56	-2.16	-1.58	-2.29	-2.50	-2.82	-2.77	-3.25
-5,655	-0.85	-0.96	-0.39	-0.44	-1.75	-2.43	-1.77	-2.60	-2.83	-3.19	-3.16	-3.70
-739	-0.03	-0.05	-0.02	-0.02	1.46	-0.09	-0.08	-0.12	0.00	-0.13	-0.13	-0.14
-737	-0.03	-0.05	-0.02	-0.03	1.45	-0.09	-0.08	-0.12	0.00	-0.13	-0.13	-0.14
-1,173	-0.05	-0.08	-0.03	-0.04	1.46	-0.14	-0.12	-0.17	0.00	-0.18	-0.19	-0.22
-2,870	-0.12	-0.23	-0.07	-0.10	1.46	-0.36	-0.32	-0.45	0.01	-0.46	-0.53	-0.58
-4,240	-0.19	-0.38	-0.10	-0.17	1.46	-0.56	-0.51	-0.72	0.00	-0.72	-0.82	-0.95
-5,587	-0.31	-0.67	-0.16	-0.29	1.46	-0.93	-0.86	-1.21	0.00	-1.22	-1.37	-1.52
-8,317	-0.72	-1.12	-0.36	-0.51	1.46	-1.91	-1.62	-2.24	0.00	-2.43	-2.56	-2.88
-10,025	-0.85	-1.27	-0.43	-0.59	1.46	-2.21	-1.87	-2.57	0.00	-2.82	-2.94	-3.31

Bibliography

- [1] T.W. Andersen. *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, Inc., New York, 1984.
- [2] D. Bates and D. Watts. *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons, Inc., New York, 1988.
- [3] J.P. Benzecri. *L'Analyse des Donnees*. Dunod, 1976.
- [4] S. Chatterjee. *Sensitivity Analysis in Linear Regression*. John Wiley & Sons, New York, 1987.
- [5] R.D. Cook and Sanford Weisberg. *Residuals and Influence in Regression*. Chapman and Hall Ltd., New York, 1982.
- [6] H. Demuth and M. Beale. *Neural Network Toolbox*. The MathWorks, Inc., Massachusetts, 1992.
- [7] G.H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1985.
- [8] B. Green. "The Orthogonal Approximation of an Oblique Structure in Factor Analysis". *Psychometrika*, 17:429-440, 1952.

- [9] M.J. Greenacre. *Theory and Applications of Correspondence Analysis*. Chapman and Hall Ltd., Acad. Press, 1984.
- [10] S. Van Huffel and J. Vandewalle. "The Total Least Squares Techniques: Computation, Properties and Applications". In E.F. Deperettere, editor, *SVD and Signal Proc.* Elsevier, 1988.
- [11] S. Van Huffel and J. Vandewalle. "Algebraic Connections Between the Least Squares and Total Least Squares Problems". *Numer. Math*, 1989:
- [12] D.R. Hush and B.G. Horne. "Progress in Supervised Neural Networks". *IEEE Signal Processing Magazine*, pages 8-39, Jan 1993.
- [13] Michel Jambu. *Exploratory and Multivariate Data Analysis*. Academic Press, Inc., Boston, 1991.
- [14] J.M. Jenkins, A.E. Kuhl, and A.L. Carter. "Strain Gage Calibration of a Complex Wing". *Journal of Aircraft*, 14:1192-1196, Dec 1977.
- [15] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Inc., New Jersey, 1992.
- [16] B. Kosko, editor. *Neural Networks for Signal Processing*. Prentice-Hall, Inc., New Jersey, 1992.
- [17] S.Y. Kung. *Digital Neural Networks*. PTR Prentice Hall, New Jersey, 1993.
- [18] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Inc., New Jersey, 1974.

- [19] L. Lebart. *Multivariate Descriptive Statistical Analysis*. John Wiley & Sons, 1984.
- [20] R. P. Lippmann. "An Introduction to Computing with Neural Nets". *IEEE ASSP Magazine*, pages 4–22, April 1987.
- [21] C.Y. Maa and M.A. Shanblatt. "Linear and Quadratic Programming Neural Network Analysis". *IEEE Transaction on Neural Networks*, 3, No.4:580–594, July 1992.
- [22] E. Oja. *Subspace Method of Pattern Recognition*. John Wiley & Sons, Inc., New York, 1984.
- [23] B Ostle. *Statistics in Research*, chapter 8,9. The Iowa State University Press, 1954.
- [24] T.H. Skopinski, W.S. Aiken Jr., and W.B. Huston. "Calibration of Strain Gage Installations in Aircraft Structures for Measurement of Flight Loads". Technical report, NASA Rept. 1178, 1954.
- [25] G.W. Stewart. *Collinearity and Least-Squares Regression*. Statistical Science, 1987.
- [26] M.H. Tang and R.G. Sheldon. "A modified T-value Method for Selection of Strain Gages for Measuring Loads on A Low Aspect Ratio Wing". Technical report, NASA Tech. Paper - 1748, 1980.
- [27] P.J. Werbos. "Back Propagation Through Time: What It Does and How To Do It". *Proc. IEEE*, 78, No. 10:1550–1560, October 1990.

- [28] G.B. Wetherill. *Regression Analysis with Applications*. Chapman and Hall Ltd., New York, 1985.
- [29] Q. Xue, Y. Hu, and W. Tompkins. "A Neural Network Weight Pattern Study with ECG Pattern Recognition". *11 th Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 1989.
- [30] K. Yao. "SVD-based Data Reduction and Classification Techniques". In V. Poor, editor, *Advances in Statistical Processing - Theory and Applications*. JAI Press, 1991.
- [31] K. Yao, F. Lorenzelli, and J. Kong. "The Equivalence of Total Least Square and Correspondence Analysis". *Proc. IEEE Conference on ICASSP*, pages 585-588, March 1992.
- [32] Jacek M. Zurada. *Introduction to Artificial Neural Systems*. West Publishing Co., St. Paul, 1992.