The Modelling of IR Emission Spectra and Solid Rocket Motor Parameters using Neural Networks and Partial Least Squares

By Niko Hamp

Thesis presented in partial fulfilment of the requirements for the degree of Masters of Science in Chemical Engineering at the University of Stellenbosch, South Africa

Study-leaders

Professor J.H. Knoetze

Professor C. Aldrich

April 2003

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

N. Hamp 1 February 2003

Abstract

The emission spectrum measured in the middle infrared (IR) band from the plume of a rocket can be used to identify rockets and track inbound missiles. It is useful to test the stealth properties of the IR fingerprint of a rocket during its design phase without needing to spend excessive amounts of money on field trials. The modelled predictions of the IR spectra from selected rocket motor design parameters therefore bear significant benefits in reducing the development costs.

In a recent doctorate study it was found that a fundamental approach including quantum-mechanical and computational fluid dynamics (CFD) models was not feasible. This is first of all due to the complexity of the systems and secondly due to the inadequate calculation speeds of even the most sophisticated modern computers. A solution was subsequently investigated by use of the 'black-box' model of a multi-layer perceptron feed-forward neural network with a single hidden layer consisting of 146 nodes. The input layer of the neural network consists of 18 rocket motor design parameters and the output layer consists of 146 IR absorbance variables in the range from 2 to 5 μm wavelengths. The results appeared promising for future investigations.

The available data consist of only 18 different types of rocket motors due to the high costs of generating the data. The 18 rocket motor types fall into two different design classes, the double base (DB) and composite (C) propellant types. The sparseness of the data is a constraint in building adequate models of such a multivariate nature. The IR irradiance spectra data set consists of numerous repeat measurements made per rocket motor type. The repeat measurements form the pure error component of the data, which adds stability to training and provides lack-of-fit ANOVA capabilities.

The emphasis in this dissertation is on comparing the feed-forward neural network model to the linear and neural network partial least squares (PLS) modelling techniques. The objective is to find a possibly more intuitive and more accurate model that effectively generalises the input-output relationships of the data. PLS models are known to be robust due to the exclusion of redundant information from projections made to primary latent variables, similarly to principal components (PCA) regression. The neural network PLS techniques include feed-forward sigmoidal neural network PLS (NNPLS) and radial-basis functions PLS (RBFPLS). The NNPLS and RBFPLS algorithms make use of neural networks to find non-linear functional relationships for the inner PLS models of the NIPALS algorithm. Error-based neural network PLS (EBNNPLS) and radial-basis function network PLS (EBRBFPLS) are also briefly investigated, as these techniques make use of non-linear projections to latent variables.

A modification to the orthogonal least squares (OLS) training algorithm of radial-basis functions is developed and applied. The adaptive spread OLS algorithm (ASOLS) allows for the iterative adaptation of the Gaussian spread parameters found in the radial-basis transfer functions.

Over-fitting from over-parameterisation is controlled by making use of leave-one-out cross-validation and the calculation of pseudo-degrees of freedom. After cross-validation the overall model is built by training on the entire data set. This is done by making use of the optimum parameterisation obtained from cross-validation. Cross-validation also gives an indication of how well a model can predict data unseen during training.

The reverse problem of modelling the rocket propellant chemical compositions and the rocket physical design parameters from the IR irradiance spectra is also investigated. This problem bears familiarity to the field of spectral multivariate calibration. The applications in this field readily make use of PLS and neural network modelling. The reverse problem is investigated with the same modelling techniques applied to the forward modelling problem.

The forward modelling results (IR spectrum predictions) show that the feed-forward neural network complexity can be reduced to two hidden nodes in a single hidden layer. The NNPLS model with eleven latent dimensions outperforms all the other models with a maximum average R^2-value of 0.75 across all output variables for unseen data from cross-validation. The explained variance for the output data of the overall model is 94.34%. The corresponding explained variance of the input data is 99.8%. The RBFPLS models built using the ASOLS training algorithm for the training of the radial-basis function inner models outperforms those using K-means and OLS training algorithms.

The lack-of-fit ANOVA tests show that there is reason to doubt the adequacy of the NNPLS model. The modelling results however show promise for future development on larger, more representative data sets.

The reverse modelling results show that the feed-forward neural network model, NNPLS and RBFPLS models produce similar results superior to the linear PLS model. The RBFPLS model with ASOLS inner model training and 5 latent dimensions stands out slightly as the best model. It is found that it is feasible to separately find the optimum model complexity (number of latent dimensions) for each output variable. The average R^2-value across all output variables for unseen data is 0.43. The average R^2-value for the overall model is 0.68. There are output variables with R^2-values of over 0.8.

The forward and reverse modelling results further show that dimensional reduction in the case of PLS does produce the best models. It is found that the input-output relationships are not highly non-linear. The non-linearities are largely responsible for the compensation of both the DB- and C-class rocket motor designs predictions within the overall model predictions. For this reason it is suggested that future models can be developed by making use of a simpler, more linear model for each rocket class after a class identification step. This approach however requires additional data that must be acquired.

Opsomming

Die emissiespektra van die uitlaatpluime van vuurpyle in die middel-infrarooi (IR) band kan gebruik word om die vuurpyle te herken en om inkomende vuurpyle op te spoor. Dit is nuttig om die uitstralingseienskappe van ‘n vuurpyl se IR afdruk te toets, sonder om groot bedrae geld op veldtoetse te spandeer. Die gemodelleerde IR spektrale voorspellings vir ‘n bepaalde stel vuurpylmotor ontwerpsparameters kan dus grootliks bydra om motorontwikkelingskostes te bemoei.

In ‘n onlangse doktorale studie is gevind dat ‘n fundamentele benadering van kwantum-meganiese en vloeidinamika-modelle nie lewensvatbaar is nie. Dit is hoofsaaklik as gevolg van die onvoldoende vermoë van selfs die mees gesofistikeerde moderne rekenaars. ‘n Moontlike oplossing tot die probleem is ondersoek deur gebruik te maak van ‘n multilaag perseptron voorwaartse neurale netwerk met 146 nodes in ‘n enkele versteekte laag. Die laag van invoer veranderlikes bestaan uit agtien vuurpylmotor ontwerpsparameters en die uitvoerlaag bestaan uit 146 IR-absorbansie veranderlikes in die reeks golflengtes vanaf 2 tot 5 μm. Dit het voorgekom dat die resultate belowend lyk vir toekomstige ondersoeke.

Weens die hoë kostes om die data te genereer bestaan die beskikbare data uit slegs agtien verskillende tipes vuurpylmotors. Die agtien vuurpyl tipes val verder binne twee ontwerpsklasse, naamlik die dubbelbasis (DB) en saamgestelde (C) dryfmiddeltipes. Die yl data bemoeilik die bou van doeltreffende multiveranderlike modelle. Die datastel van IR uitstralingspektra bestaan uit herhaalde metings per vuurpyltipe. Die herhaalde metings vorm die suiwer fout komponent van die data. Dit verskaf stabilitieit tot die opleiding op die data en verder die vermoë om ‘n analise van variansie (ANOVA) op die data uit te voer.

In hierdie tesis lê die klem op die vergelyking tussen die voorwaartse neurale netwerk en die lineêre en neurale netwerk parsiële kleinste kwadrate (PLS) modelleringstegnieke. Die doel is om ‘n moontlik meer insiggewende en akkurate model te vind wat effektief die in- en uitvoer verhoudings kan veralgemeen. Dit is bekend dat PLS modelle meer robuus kan wees weens die weglating van oortollige inligting deur projeksies op hoof latente veranderlikes. Dit is analoog aan hoofkomponente (PCA) regressie. Die neurale netwerk PLS-tegnieke sluit in voorwaartse sigmoïdale neurale netwerk PLS (NNPLS) en radiale-basis funksies PLS (RBFPLS). Die NNPLS en RBFPLS algoritmes maak gebruik van die neurale netwerke om nie-lineêre funksionele verbande te kry vir die binne PLS-modelle van die nie-lineêre iteratiewe parsiële kleinste kwadrate (NIPALS) algoritme. Die fout-gebaseerde neurale netwerk PLS (EBNNPLS) en radiale-basis funksies PLS (EBRBFPLS) is ook weens hulle nie-lineêre projeksies na latente veranderlikes kortiliks ondersoek.

‘n Aanpassing tot die ortogonale kleinste kwadrate (OLS) opleidingsalgoritme vir radiale-basis funksies is ontwikkel en toegepas. Die aangepaste algoritme (ASOLS) behels die iteratiewe aanpassing van die verspreidingsparameters binne die Gauss-funksies van die radiale-basis transformasie funksies.

Die oormatige parameterisering van ‘n model word beheer deur kruisvalidering met enkele weglatings en die berekening van pseudo-vryheidsgrade. Na kruisvalidering word die algehele model gebou deur opleiding op die volledige datastel. Dit word gedoen deur van die optimale parameterisering gebruik te maak wat deur kruisvalidering bepaal is. Kruisvalidering gee ook ‘n goeie aanduiding van hoe goed ‘n model ongesiende data kan voorspel.

Die modellering van die vuurpyle se chemiese en fisiese ontwerpsparameters (omgekeerde probleem) is ook ondersoek. Hierdie probleem is verwant aan die veld van spektrale multiveranderlike kalibrasie. Die toepassings in die veld maak gebruik van PLS en neurale netwerk modelle. Die omgekeerde probleem word dus ondersoek met dieselfde modelleringstegnieke wat gebruik is vir die voorwaartse probleem.

Die voorwaartse modelleringsresultate (IR voorspellings) toon dat die kompleksiteit van die voorwaartse neurale netwerk tot twee versteekte nodes in ‘n enkele versteekte laag gereduseer kan word. Die NNPLS model met elf latente dimensies vaar die beste van alle modelle, met ‘n maksimum R^2-waarde van 0.75 oor alle uitvoer veranderlikes vir die ongesiende data (kruisvalidering). Die verklaarde variansie vir die uitvoer data vanaf die algehele model is 94.34%. Die verklaarde variansie van die ooreenstemmende invoer data is 99.8%. Die RBFPLS modelle wat gebou is deur van die ASOLS algoritme gebruik te maak om die PLS binne modelle op te lei, vaar beter in vergelyking met die K-gemiddeldes en OLS opleidingsalgoritmes.

Die toetse wat ‘n ‘tekort-aan-passing’ ANOVA behels, toon dat daar rede is om die geskiktheid van die NNPLS model te wantrou. Die modelleringsresultate lyk egter belowend vir die toekomstige ontwikkeling van modelle op groter, meer verteenwoordigde datastelle.

Die omgekeerde modellering toon dat die voorwaartse neurale netwerk, NNPLS en RBFPLS modelle soortgelyke resultate produseer wat die lineêre PLS model s’n oortref. Die RBFPLS model met ASOLS opleiding van die PLS binne modelle word beskou as die beste model. Dit is lewensvatbaar om die optimale modelkompleksiteite van elke uitvoerveranderlike individueel te bepaal. Die gemiddelde R^2-waarde oor alle uitvoerveranderlikes vir ongesiende data is 0.43. Die gemiddelde R^2-waarde vir die algehele model is 0.68. Daar is van die uitvoer veranderlikes wat R^2-waardes van 0.8 oortref.

Die voor- en terugwaartse modelleringsresultate toon verder dat dimensionele reduksie in die geval van PLS die beste modelle lewer. Daar is ook gevind dat die nie-lineêriteite grootliks vergoed vir die voorspellings van beide DB- en C-tipe vuurpylmotors binne die algehele model. Om die rede word voorgestel dat toekomstige modelle ontwikkel kan word deur gebruik te maak van eenvoudiger, meer lineêre modelle vir elke vuurpylklas nadat ‘n klas-identifikasiestap uitgevoer is. Die benadering benodig egter addisionele praktiese data wat verkry moet word.

Acknowledgements

At no time was I alone on the journey to completing this work as I received the love, support and encouragement every step of the way from the people to whom I dedicate this work. I thank my mother and father, Dagmar and Rainer, my sister, Elvira and last but not least my girlfriend, Lauren.

My study-leaders, Prof. Hansie Knoetze and Prof. Chris Aldrich were always willing to advise and engage in friendly discussions throughout the course of this thesis. I much appreciate the mentorship and support I received from them and I thank them for that.

The comradeship and the lengthy exchanges in knowledge and skills across numerous disciplines with fellow students and friends are experiences I shall always cherish. Many thanks to Albert Visagie, Ludwig Schwardt and Robert Fanner from the Data Signal Processing group for their time and willingness to help with modelling issues and programming. I thank JP Barnard for his tutorship and willingness to always teach me something new about modelling and computer programming. I also thank a good friend, Chris Human for his good advice and the lengthy discussions we had on experiences during our studies.

I am always amazed by the unselfish approach of some people to assist others when help is needed. I thank Prof. N.J. le Roux from the Statistics Department at the University of Stellenbosch for the numerous occasions he gave up his time when I approached him for advice on certain statistical concepts. Thank-you to Dr. Nico Prinsloo from my bursary sponsoring company, Sasol for helping me set out on the path for this thesis.

I thank my Creator for giving me the strength and the well-being to pursue my goals during the course of this thesis.

Contents

List of Figures

List of Tables

Nomenclature

Symbols

a	The number of discarded dimensions for SVD
b	Linear regression coefficient
c	RBF centre
d	Euclidean distances between data points
df	Degrees of freedom of a regression model
D_T	Throat diameter [mm]
e	Output variable residual
E_C	Expansion ratio of rocket chamber cone
ERR	Error reduction ratio
f	Input variable residual
FPE	Akaike's final prediction error
g	Weighted summation of node inputs
h	Total number of latent dimensions
H	Number of hidden nodes
j	Element of the Jacobian matrix
k	Total number of independent samples or classes/no. of principal components
K	Total number of K-means centres
l	Total number of y-variables
LD	Latent dimension
LV	Latent variable
m	Total number of x-variables or model parameters
M	Total number of parameters
MSE	Sample mean-square error
MSECV	Mean-square error of cross-validation
MSEP	Sample mean-square error of prediction
n	Number of observed measurement samples
N	Number of nearest neighbours
o	Leverage

p	Eigen vector/latent dimension loading value of X-data
P_C	Chamber pressure [MPa]
pdf	Pseudo-degrees of freedom of a regression model
PI	Confidence interval
PRESS	Predicted residual estimate sum of squares
q	Latent dimension loading value of Y-data
r	Correlation coefficient
R^2	Coefficient of determination
RMSE	Root mean-square error (overall)
RMSECV	Root mean-square error of cross-validation
s	Sample variance
SEP	Sample standard error of prediction
SS	Sum of squares
SSE	Sum-square error
SSEP	Sum-square error of prediction
SSK	Sum-square criterion for K-means clustering
SSL	Sum-square lack-of-fit
SSM	Sum-square model variance
SSP	Sum-square pure error
SST	Sum-square total
t	X-data score value or latent variable/statistic for student's t-distribution
T_C	Chamber temperature [K]
u	Y-data score value or latent variable
v	Input value for OLS regression/2nd order PCA eigenvector decomposition
w	Input weights for PLS modelling
x	Input or independent variable
y	Output or dependent variable
z	Activation of a neural node or derivative of a function

Greek Symbols

α	Gram-Schmidt orthogonalisation coefficient/statistical significance level
β	Neural network bias
χ	Index record in adaptive spread OLS algorithm
Δ	Right singular matrix

ε	Population mean-square error
Γ	Diagonal eigenvalue matrix of 2nd order PCA decomposition
$\eta 2$	Explained variance
φ	Learning rate
κ	Total number of hidden nodes considered during NNPLS training
λ	Eigenvalue
Λ	Left singular matrix
μ	Marquardt optimisation parameter or population mean
ν	Extra candidate centres in adaptive spread OLS algorithm
θ	Generalised regression parameter
ρ	RBF spread parameter
σ	Population variance
Σ	Singular value matrix
τ	Total number of training sessions
ω	Neural network weight
ξ	Standard normal deviate
ψ	Orthogonal inputs for OLS

Indices

a	Latent dimension number
D	Signifies a diagonalised matrix
e	Signifies residual or error
g	Sample number within a class
G	Signifies the class-based intra-class spread matrix
h	Independent sample number or class number
H	Signifies the class-based inter-class spread matrix
i	Measurement sample number
j	y-variable number
k	x-variable number
K	Signifies K-means
L	Signifies mean-square lack-of-fit
l	Gram-Schmidt vector index
M	Signifies set of model predicted training data

n	Nearest neighbours number
P	Signifies pure error
p	Weight index from the previous neural net layer / number of parameters
q	Weight index from the current neural net layer
R	Signifies replicates
r	Regression parameter number (also used as hidden node index)
rs	Re-substitution value applicable to MSEP
T	Signifies total random data set
t	Iteration number

Chapter 1

Introduction

The prediction of the infrared (IR) emission spectrum from the exhaust gases making up a rocket plume finds numerous applications in the strategic identification of rockets. These rocket fingerprints can be classified, thus allowing for the distinction between friend and foe. The plume radiation intensity can be reduced for stealth purposes. The lower the intensity the more difficult it becomes to track an incoming rocket. It is useful to be able to identify if a rocket has the required stealth characteristics during its design phase already, thus cutting down on the high manufacturing and live testing costs involved.

The challenge of predicting the plume radiance is one of describing the thermodynamic combustion process within the rocket chamber, the plume structure and the rocket plume composition. The factors guiding these processes are the physical rocket motor design parameters as well as the rocket motor fuel chemistry. In addition, environmental conditions have a significant impact on the plume structure and the plume chemical composition. The speed at which the rocket is travelling influences the plume shape and surface temperature profile. The atmospheric temperature, density and humidity will affect the plume chemistry and the optical scatter, which in turn affects the recording of the IR emission. These inter-relationships are illustrated in Fig. 1-1.

An attempt can now be made to model the processes in Fig. 1-1 by using the rocket motor design parameters and the rocket chemistry as inputs. The environmental conditions must also be considered in order to obtain consistent results. A robust model can therefore be used to develop rockets with the desired IR emission spectrum required for stealth purposes by iteratively changing the solid rocket design parameters and chemistry.

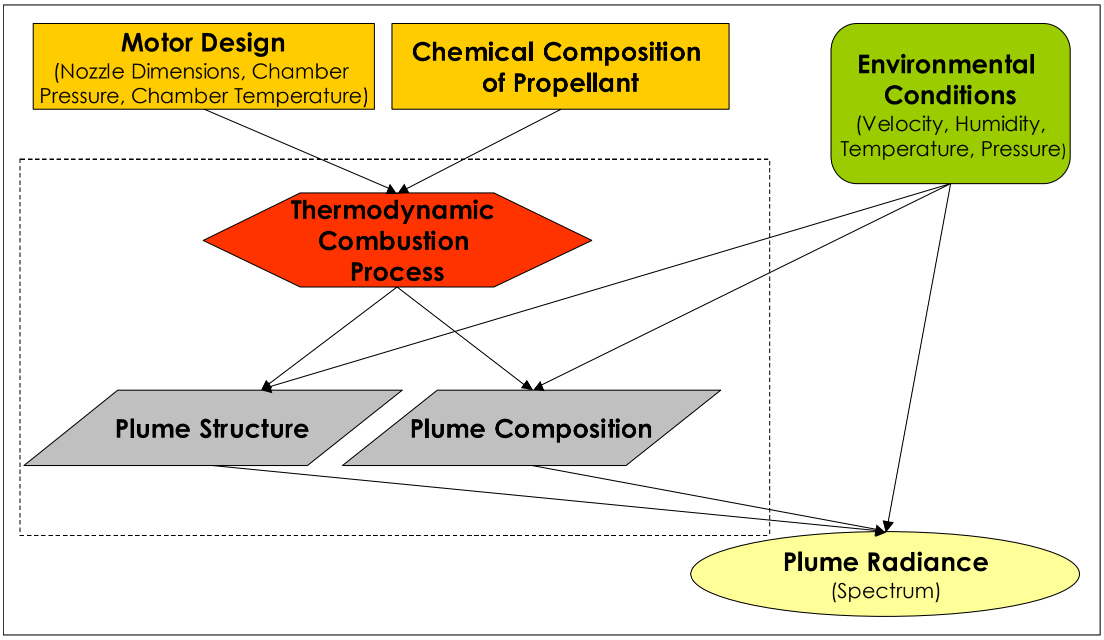
Motor Design
(Nozzle Dimensions, Chamber Pressure, Chamber Temperature)
Chemical Composition of Propellant
Environmental Conditions
(Velocity, Humidity, Temperature, Pressure)
Thermodynamic Combustion Process
Plume Structure
Plume Composition
Plume Radiance
(Spectrum)

Fig. 1-1 A flow diagram illustrating the various contributions to the rocket plume radiance (rocket fingerprint).

In the work done by Roodt [1998] it is mentioned that the point of departure is to use computational fluid dynamics (CFD) to model the two- or three-dimensional plume structure. The unsteady Navier-Stokes equations must then be solved in order to incorporate the effects of highly turbulent, local conditions within the rocket plume. Finite rate chemical reactions can be incorporated in order to obtain the chemistry of a limited number of plume species. From these models a blackbody temperature profile can be generated on a two or three-dimensional plume structure. This enables the calculation of the emission spectrum in a pre-defined IR band from quantum-mechanical principles.

At the time Roodt [1998] reported that it took a four–processor SGI Power Challenge four weeks to obtain a solution to a 600 000 grid-point model. It was further found that the results were not consistent with experimental IR measurements. The CFD approach is therefore too computationally intensive to justify its practical application, especially as it is critical that predictions can be made within short periods of time.

The use of classical statistical regression techniques becomes a daunting task, as there are numerous input variable interactions and external influences that cannot always be taken into consideration. In addition to this the available data is too sparse for this type of model.

Roodt [1998] proceeded to model the plume irradiances from the rocket motor physical design parameters and chemistry by making use of a feed-forward neural network with a single hidden layer. This ‘black-box’ method finds a functional mapping between the motor design parameters and chemistry and the output plume radiance. The advantages of this approach are the reduction in computational intensity, the high-speed predictions that can be made and the fact that no prior form of the functional relationship between the in- and outputs is required.

Roodt [1998] made use of eighteen inputs to the neural network model: fourteen elemental chemical species and four solid rocket motor design parameters. The output variables consisted of 146 spectral absorbance units at wavelengths in the middle IR band (2–5 μm wavelength). He built the neural network model on data from sixteen different rockets and tested the model on data from two unseen rockets. Repeated IR radiance measurements were obtained for each rocket and were subsequently included for building the model. The model predictions agreed well with experimental measurements for both the training and testing data.

In this work the emphasis is on comparing the neural network model to linear and neural network partial least squares (PLS) modelling techniques. The objective is to find a possibly more accurate model capable of generalising the in- and output relationships in a simpler manner.   Qualitative information on the relationship between the input and output variables can often be extracted from simpler, more intuitive models. This can be seen as adding to the value of a ‘black-box’ model. Furthermore, if an improved functional relationship can be obtained from a more intuitive model compared to a neural network what is the reasoning supporting the result?

In this work more emphasis is also put on the validation of the models used. The validation of a non-linear model is important, as there is always the problem of over-fitting a function on the given data and subsequently unseen data may be poorly predicted. The same data used by Roodt [1998] is used for building the neural network and PLS models in this work. It is evident that the data is sparse and the validation of the models becomes a more complex problem. The validation technique used in this work leads to a more critical evaluation of the performances of the models and subsequently the feasibility of any further applications of the models can be considered.

Another useful application is the prediction of the rocket motor design parameters and chemistry from the measured IR emission spectra. This is the inverse of the original problem. The 146 input wavelengths may contain numerous interactions and redundant information and therefore much room may be left for model simplification. The modelling of spectral data structures is extensively researched with numerical methods such as principal components analysis (PCA) and PLS being at the forefront of modern applications. The reverse modelling of the data is applied in this work and the use of PLS model building is derived from the backdrop of its applications in the field of chemometrics.

This work starts off with the descriptions of various numerical methods in exploratory data analysis used in the data processing and modelling applied later on. Afterwards, the detection of possible outliers is considered in a chapter where the available data is analysed and scrutinised. The two chapters following the data analysis then discuss the methodologies applied and the results obtained for the forward and reverse models built.

Chapter 2

Methods of Exploratory Data Analysis

The statistical and mathematical tools used for the processing and modelling of the solid rocket fuel data in this study are described in this chapter. The aim of this chapter is to introduce the reader to some of the most basic statistical concepts before moving on to the more advanced topics in order to provide a logical build up in understanding the concepts applied and the link in notation. The aim is to provide a better understanding as the statistical concepts are blended with mathematical formulations. The notation used in this text is explained as the various concepts are introduced.

2.1 Basic Statistical Concepts

One of the most fundamental statistical concepts is that of a population and its probability distribution. Without this concept it becomes impossible to plan, assess or understand statistical modelling.

Consider a variable, y (e.g. a measured temperature) collected over its entire population, i.e. all the possible values y can take on. There will be varying numbers of recordings of certain y-values with a subsequent clustering around maximum probability values. The histogram in Fig. 2-1 shows how the x-axis is divided up into equal intervals, Δy, and the number of y-values in each interval is counted. The frequency is then obtained by dividing the number of counts by the total number of objects in the population. As the population becomes larger and larger, Δy will approach zero and the histogram approaches a smooth continuous curve as in Fig. 2-1. The probability that a certain y will fall within an interval is the area of the interval underneath the smooth curve hence the curve is known as the probability density function (Hogg [1992] and Martens [1989]).

The concept of a population is, however idealised and must be estimated by a random sample of measurements of say n sample points for y_i, where i =1,2,…,n. The larger this random sample of n points becomes, the better the sample approaches a population estimate.

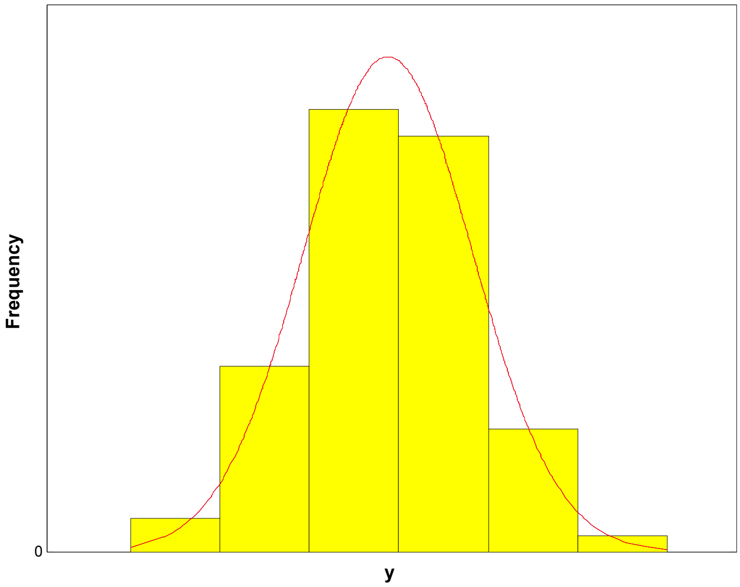

Fig. 2-1 The bar graph represents a histogram while the smooth curve represents the probability density function of the variable (y) frequencies.

2.1.1 Expectation

The expected value of y can be defined as the weighted sum of all the possible values of y as

$$E(y) = \sum_{i=1}^{n} p(y_i)\, y_i \qquad (2\text{-}1)$$

where $p(y_i)$ is the probability of y_i occurring. From the section above, it can be seen that as Δy tends towards zero the population mean becomes:

$$E(y) = \int_{-\infty}^{\infty} y\, F(y)\, dy \qquad (2\text{-}2)$$

Here *F* represents the probability density function. In practice, *E*(y) cannot be calculated but it can be estimated form a random sample set of n points by the sample mean,

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (2\text{-}3)$$

2.1.2 Variance

The expectation is a measure of location, i.e. where in the data the expected value or 'centre of gravity' can be expected. Variance is a measure of the spread in the data. The population variance is defined as the expected quadratic difference between the actual value and its expectation,

$$\sigma_{yy} = E\left[(y - E(y))^2\right] \qquad (2\text{-}4)$$

The sample variance is the estimation of the population variance and is calculated as:

$$s_{yy} = \frac{1}{n\text{-}1}\sum_{i=1}^{n} (y_i - \overline{y})^2 \qquad (2\text{-}5)$$

Here n-1 degrees of freedom are used because the mean has been used in the calculation of s_{yy}. If one subtracts the mean from each y_i to obtain a new set with each element now, y_i', then

$$\sum_{i=1}^{n} (y_i - \overline{y}) = \sum_{i=1}^{n} y_i' = 0 \qquad (2\text{-}6)$$

From this result it is evident that the nth point can be determined from the n-1 other points and therefore s_{yy} is estimated from n-1 random points in probability. However, for large data sets n degrees of freedom (df) are often used due to the insignificant change in value from n-1.

The standard deviation is the square root of the variance ($\sqrt{\sigma}$ or $\sqrt{s}$) and has the advantage of bearing the same units as the measured variable, y. However, variance is additive and for this reason it is often the preferred measure of spread for evaluating models. Usually the standard deviation is denoted by s or σ and the corresponding variance is simply the square thereof. This nomenclature has not been adopted in order to compensate for the symbolic definition of covariance (see section 2.1.3 below) and because variance plays a greater role in this text.

2.1.3 Covariance and Correlation

This concept refers to the inter-relationship between variables, such as y and x for example. The definition for the population covariance is:

$$\sigma_{yx} = E\left[(y - E(y))(x - E(x))\right] \tag{2-7}$$

The sample covariance can be estimated by:

$$s_{yx} = \frac{1}{n-1}\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) \tag{2-8}$$

The inter-relationship between x and y can be explained as follows: if an increase in y is accompanied by an increase in x the covariance is positive. If an increase in y is generally associated with a small or decreasing value of x the covariance is negative. No covariance is obtained when there is no associative trend between two variables, i.e. they are independent and the value of σ_{yx} is zero.

The covariance is, however dependent on the scale of the variables of x and y. The correlation coefficient compensates for this and can be estimated from:

$$r_{xy} = \frac{s_{yx}}{\sqrt{s_y}\sqrt{s_x}} \tag{2-9}$$

The correlation coefficient can take on values between –1 and +1. A perfect positive linear correlation relates to $r_{yx} = 1$ and for a perfect negative correlation $r_{yx} = -1$.

2.1.4 Multivariate Statistics

The statistics of expectation and variance defined above are applicable to single variables measured over random measured sample sets. It becomes tedious to deal with each variable separately when working with multiple variables. Multivariate spaces also have inter-relationships that together may contribute to the overall results as in linear regression modelling, for example. For these reasons multivariate statistics are defined using standard vector and matrix notation in order to be able to mathematically deal with the complexities and challenges posed by multivariate spaces.

The notation adopted in this text makes use of bold font lower case letters to represent vectors and bold face upper case letters to represent matrices. A scalar value can be regarded as a 1×1 vector.

The multivariate statistics definitions for theoretical populations will not be shown here. Only their estimates will be defined, as the population statistics are only of theoretical value. Their definitions are analogous to the single variable definitions given above and their multivariate sample estimates.

In this text a set of data consisting of n data sample points and m variables will be written as an n×m matrix of data elements. The columns of such a matrix thus contain the sample data of the individual variables,

$$\mathbf{X} = \left[\mathbf{x}_1 \ldots \mathbf{x}_k \ldots \mathbf{x}_m\right] \qquad (2\text{-}10)$$

For the data set, **X**, the multivariate sample mean is simply a vector of the sample means of the individual variables and is defined as (Johnson [1988] and Tabachnick [1983]):

$$\bar{\mathbf{x}}^{T} = \left[\bar{x}_1 \ldots \bar{x}_k \ldots \bar{x}_m\right] \qquad (2\text{-}11)$$

Here the superscript, T, refers to the transpose of a vector or matrix. This is because in this text vectors are defined as column vectors with dimensions n×1, where n can represent any value.

In multivariate statistics the variances and covariances of the variables are captured within the so-called covariance matrix. The sample variance-covariance matrix is defined as (Johnson [1988]):

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \cdot & \cdot & \dots & \cdot \\ s_{m1} & s_{m2} & \dots & s_{mm} \end{bmatrix} \qquad (2\text{-}12)$$

In multivariate statistics n degrees of freedom instead of n-1 are often used as large amounts of data (>30 samples) are usually processed. The difference of 1 therefore becomes insignificant.

It is usually more desirable to use a single value as a measure of a generalised variance for all m variables. According to Johnson [1988] and Tabachnick [1983] a generalised measure of variance can be the determinant of **S** or simply the sum of the diagonal elements more commonly known as the trace of **S**. The latter is more commonly used and is known as the total sample variance or generalised variance:

$$\mathrm{tr}(\mathbf{S}) = s_{11} + s_{22} + \dots + s_{mm} \qquad (2\text{-}13)$$

2.2 Linear and Non-linear Models

Consider an example where a dependent or output variable, y has to be predicted from an independent variable, x by some optimised function, *f*(x), representing the model. The function, *f*, can belong to a certain class of functions F, where $f \subseteq F$. The class, F dictates the form of the possible functions, *f*, by a fixed set of parameters, P, with parameters, $\theta_r \in \Re$. In the case of linear and non-linear models, *f* can be estimated from a set of n sample measurements of corresponding pairs (x_i,y_i) by minimising some cost function, $\mathcal{E}(\theta)$. A popular cost function is the sum-squared error (Hogg [1992], Draper [1981], Haykin [1999]),

$$\mathcal{E}(\theta) = \sum_{i=1}^{n} \left(y_i - f(x_i,\theta)\right)^2 \qquad (2\text{-}14)$$

where the parameters obtained from the minimisation are referred to as the least-square estimates.

The dependent or output variable, y can now be predicted using some estimated function, *f*, from the expression,

$$y_i = f(x_i,\theta) + e_i \tag{2-15}$$

where the residual is defined as:

$$e_i = y_i - f(x_i,\theta) \tag{2-16}$$

The residuals, e_i, are regarded as mutually independent random variables that create the scatter around the relationship, $f(x_i,\theta)$ for i = 1,2,…,n, respectively. It is also assumed that they are normally distributed, with $E(e_i) = 0$ and variance, ε ($E[(e_i - E(e_i))^2]$).

From the assumptions above, it is evident that the response in (2-15) consists of the sum of two components: the non-random component, $f(x_i,\theta)$, and the random component e_i. Hence, y_i is also a random variable and has a normal distribution with the expected mean,

$$E(y_i) = f(x_i,\theta) \tag{2-17}$$

and variance ε ($E[(e_i - E(e_i))^2]$).

2.2.1 Linear Models

A model is regarded as linear when it is linear in its parameters, b, for example for (x_i,y_i):

$$y_i = x_i b + e_i \tag{2-18}$$

Here the residual becomes,

$$e_i = y_i - \hat{y}_i \tag{2-19}$$

where the predicted value of y is

$$\hat{y}_i = x_i b \tag{2-20}$$

There is no restriction on the linearity of the independent variable. It can, for example, be raised to some power or its logarithm can be taken. The following is therefore also regarded as a linear model:

$$\hat{y}_i = x_i^2 b \tag{2-21}$$

The entire data set of n measured samples from which a model is estimated using the minimisation of the cost function can be related as:

$$\begin{bmatrix} y_1 \\ \cdot \\ y_i \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \cdot \\ x_i \\ \cdot \\ x_n \end{bmatrix} [b] + \begin{bmatrix} e_1 \\ \cdot \\ e_i \\ \cdot \\ e_n \end{bmatrix} \tag{2-22}$$

The minimisation of the sum-square error can be shown to be equivalent to obtaining $\hat{\mathbf{y}}$ by projecting **y** onto **x** (Wise [2000] and Draper [1981]). According to vector algebra the projection onto **x** will be some scalar multiple, b of **x**. The shortest distance from **x** to **y** is the vector, **y** – **x**b, which is perpendicular to **x**, thus,

$$\mathbf{x}^T(\mathbf{y} - \mathbf{x}b) = 0 \tag{2-23}$$

$$b = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} \tag{2-24}$$

and the projection of **y** onto **x** becomes:

$$\hat{\mathbf{y}} = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}}\mathbf{x} \tag{2-25}$$

The linear model illustrated above can only work for data in **x** and **y** that has zero mean as the bias or offset has not been taken into account. In order to compensate for the bias, **x** now becomes the matrix,

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ \cdot & \cdot \\ 1 & x_i \\ \cdot & \cdot \\ 1 & x_n \end{bmatrix} = [\mathbf{1}\ \ \mathbf{x}] \tag{2-26}$$

By definition in this text the bold face, **1**, represents an nx1 vector of ones as elements.

The minimisation of the sum-square error can now be generalised for matrix notation as (Draper [1981], Johnson [1988] and Wise [2000]):

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2-27}$$

It can be seen that this matrix notation bears an analogy to (2-24). The parameter, b_0 is the bias and it becomes clear that at least two sample measurements are required to find a solution to (2-27). In matrix notation:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e} \tag{2-28}$$

2.2.1.1 Multiple Linear Regression

In the situation where more than one linear independent or input variable is required the linear model can be expanded for m variables as follows (Hogg [1992]):

$$y_i = b_0 + x_{i1}b_1 + \ldots + x_{ik}b_r + \ldots + x_{im}\, b_m + e_i \tag{2-29}$$

Once again there is no restriction on the linearity of x_k and an input variable can even take on the form of an interaction variable, $x_k x_{k'}$, where $k \neq k'$. For large values of m it can be quite challenging to find all the interactions in the data that have a significant impact on the model performance.

The entire data set of n measured samples from which the model is estimated for multiple linear input variables is related by:

$$\begin{bmatrix} y_1 \\ \cdot \\ y_i \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1k} & \ldots & x_{1m} \\ \cdot & \cdot & & \cdot & & \cdot \\ 1 & x_{i1} & \ldots & x_{ik} & \ldots & x_{im} \\ \cdot & \cdot & & \cdot & & \cdot \\ 1 & x_{n1} & \ldots & x_{nk} & \ldots & x_{nm} \end{bmatrix} \begin{bmatrix} b_0 \\ \cdot \\ b_r \\ \cdot \\ b_m \end{bmatrix} + \begin{bmatrix} e_1 \\ \cdot \\ e_i \\ \cdot \\ e_n \end{bmatrix} \tag{2-30}$$

Here the least-square estimates are determined in the same manner as in equation (2-27).

2.2.1.2 Multivariate Linear regression

Multivariate linear regression is essentially a generalisation of linear and multiple linear regressions where it is possible to model more than one dependent variable. In this case the parameter matrix is extended to j = 1,…,l columns, each column containing the m+1 parameters per dependent variable.

The entire data set of n measured samples from which the model is estimated is related by (Johnson [1988]):

$$\begin{bmatrix} y_{11}\cdots y_{1j}\cdots y_{1l} \\ \cdot \quad \cdot \quad \cdot \\ y_{i1}\cdots y_{ij}\cdots y_{il} \\ \cdot \quad \cdot \quad \cdot \\ y_{n1}\cdots y_{nj}\cdots y_{nl} \end{bmatrix} = \begin{bmatrix} 1 & x_{11}\cdots x_{1k}\cdots x_{1m} \\ \cdot & \cdot \quad \cdot \quad \cdot \\ 1 & x_{i1}\cdots x_{ik}\cdots x_{im} \\ \cdot & \cdot \quad \cdot \quad \cdot \\ 1 & x_{n1}\cdots x_{nk}\cdots x_{nm} \end{bmatrix} \begin{bmatrix} b_{01}\cdots b_{0j}\cdots b_{0l} \\ \cdot \quad \cdot \quad \cdot \\ b_{r1}\cdots b_{rj}\cdots b_{rl} \\ \cdot \quad \cdot \quad \cdot \\ b_{m1}\cdots b_{mj}\cdots b_{ml} \end{bmatrix} + \begin{bmatrix} e_{11}\cdots e_{1j}\cdots e_{1l} \\ \cdot \quad \cdot \quad \cdot \\ e_{i1}\cdots e_{ij}\cdots e_{il} \\ \cdot \quad \cdot \quad \cdot \\ e_{n1}\cdots e_{nj}\cdots e_{nl} \end{bmatrix}$$

(2-31)

In matrix notation this expression becomes:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \qquad (2\text{-}32)$$

It is evident that each dependent variable can be modelled separately by using a single column from matrix, **B**:

$$\mathbf{y}_j = \mathbf{Xb}_j + \mathbf{e}_j \qquad (2\text{-}33)$$

The least-square estimates can be obtained in similar fashion as described in equation (2-27) by:

$$\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \qquad (2\text{-}34)$$

2.2.2 Non-linear Models

Non-linear models are no longer linear in their parameters. The implication of this is that there is no single solution to minimising the least squares criterion in equation (2-14). An example of a non-linear model is shown here (Draper [1981]):

$$y_i = \frac{\theta_1}{\theta_1 - \theta_2}\left(e^{-\theta_2 x_i} - e^{-\theta_1 x_i}\right) + e_i \qquad (2\text{-}35)$$

The minimisation of the sum-square error involves taking the first partial derivative of each parameter in equation (2-14). In linear models these derivatives become constants that are independent of the parameters themselves. Clearly, this is not the case here and an iterative procedure is required to approach some sum-square minimum along an error surface (where the sum-square error is a function of the model parameters). There can be multiple local minima and a single global minimum is not guaranteed.

Some of the iterative optimisation procedures include the linearisation technique (Taylor series expansion), the gradient descent method and Marquardt's compromise (Draper [1981] and Bates [1988]). Feed-forward neural networks used for non-linear regression make use of some of these optimisation techniques and will be discussed in more detail in section 2.11 below.

2.3 Singular Value Decomposition (SVD)

Any full rank n×n matrix, say **X** can be inverted by using Gaussian elimination in the row-echelon form. Non-symmetric matrices with dimensions n×m for example cannot be explicitly inverted, but a more implicit workaround can be obtained.

From equation (2-27) it is seen that **b** is obtained by projecting **y** onto the row space of **X** and the matrix that solves this is known as the pseudo-inverse of **X** (Wise [2000] and Martens [1989]):

$$\mathbf{X}^{+} = (\mathbf{X}^{T}\mathbf{X})^{-1}\mathbf{X}^{T} \qquad (2\text{-}36)$$

If however **X** is highly correlated in its row space the regression or projection may become ill conditioned in the sense that small changes in values in **X** can cause large variations in the obtained values for **b**. In order to counteract this occurrence **X** can be decomposed as follows (Haykin [1999],Wise [2000] and Martens [1989]):

$$\mathbf{X} = \Lambda\Sigma\Delta^{T} \qquad (2\text{-}37)$$

This is known as the singular value decomposition of **X** where Λ is n×n orthonormal and Δ is m×m orthonormal. The matrix Σ is an n×m diagonal matrix and with its diagonal elements arranged in descending order these are known as singular values. If there are (a) singular values in Σ that are zero or close to zero only the first (m-a) singular values are retained. This means that (m-a) rows and columns of Σ, and (m-a) columns in Λ and Δ are retained. Because both Λ and Δ are orthonormal the pseudo-inverse by SVD becomes:

$$\mathbf{X}^{+} = \Delta_{m-a}\, \Sigma_{m-a}^{-1}\, \Lambda_{m-a}^{T} \qquad (2\text{-}38)$$

SVD can be implemented in regression training algorithms where linear projections are used in order to avoid ill-conditioned results. This is especially useful for training RBF neural networks as explained in section 2.12 below. A computer can be programmed to disregard all those singular values below a predefined threshold level.

2.4 Data Pre- and Post-processing

Data is usually processed prior to further processing whether it is for filtering, feature extraction, regression or classification. In cases where variables of different dimensions are used it becomes crucial. It may also be necessary to pre-process the data so that it suits the architecture of the model being fitted.

Once data has been modelled after pre-processing it, any unseen data used for prediction must be pre-processed using the same values prior to inserting the values into the model. In order to compare model outputs with their target values the predicted values must be post-processed by using the inverse procedure of pre-processing.

2.4.1 Mean Centred Data

It may be useful to mean centre the data in order to remove the bias. This means that the new data set has mean vector, $\bar{\mathbf{x}} = \mathbf{0}$. Here $\mathbf{0}$ is an m-dimensional vector with zeros as elements. The processing of the data can thus be given by :

$$\mathbf{X}' = (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{T}) \qquad (2\text{-}39)$$

Here $\mathbf{1}$ is an n×1 vector of ones.

2.4.2 Standardised Data

After data has been mean-centred it may be scaled by dividing each variable, x_{ik}, by its standard deviation. This ensures that each variable in $\mathbf{X}$ has a standard deviation of 1. For data sets with variables of varying dimensions this is crucial so that a model fitted to the data is not biased towards the variables with the largest values. Each dimension or variable then exerts the same leverage on data fitting. Standardisation thus tends to make a data set more spherical in shape. In mathematical form standardisation becomes

$$\mathbf{X}' = (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{T})\,\mathbf{S}_{D}^{-1/2} \qquad (2\text{-}40)$$

Here $\mathbf{S}_D$ is the m×m covariance matrix of $\mathbf{X}$ with all the off-diagonal elements set to zero.

The standardisation and mean centring of data is extensively published in all introductory and advanced statistics literary sources such as Hogg [1992], Johnson [1988] or Draper [1981].

2.4.3 Normalised Scaling of Data

Data fitting models such as neural networks sometimes make use of non-linear transfer functions with threshold minimum and maximum values. These thresholds are usually 0 and 1 or –1 and 1. Neural networks can comfortably deal with bias in the data set and it is therefore not necessary to mean-centre the data. This type of data scaling also deals with data sets consisting of variables with varying dimensions. Mathematically, for each variable n×1 vector the following scales all the values for $\mathbf{x}_k$ between –1 and 1:

$$\mathbf{x}_k' = 2\,\frac{(\mathbf{x}_k - \min(\mathbf{x}_k))}{(\max(\mathbf{x}_k)-\min(\mathbf{x}_k))} - 1 \qquad (2\text{-}41)$$

2.5 Variance and Discrepancy

The question now is how well does a model fit the data it was trained or estimated from? In order to answer this question a measure of discrepancy between the desired output/dependent variable and the modelled variable is required. Once this has been done a further question is how well does the model generalise the system being modelled? An answer to the second question is only possible by first of all defining a measure of discrepancy. Measures of discrepancy are dealt with in this section. The second question is more tedious to deal with and comes down to finding a methodology for validating a model. This is discussed in section 2.6 below.

Equation (2-5) explains the total variance in a set of n random samples of the dependent variable, y_j. If the degrees of freedom are omitted the total variability can be referred to as the sum-square total (SST). In similar fashion the variability in the estimated variables can be explained by (Hogg [1992] and Draper [1981]) as,

$$SSM = \sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2 \qquad (2\text{-}42)$$

The model does not explain all the variation in the original set of n random samples. The part of the variation that is left unexplained is expressed by the residuals, e_i , and becomes a measure of discrepancy (sum-square error):

$$SSE = \sum_{i=1}^{n} e_i^2 \qquad (2\text{-}43)$$

It was mentioned previously that variance is additive and from the discussion above it follows that for linear models and n sample points used as estimation data (Hogg [1992] and Draper [1981]):

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}\right)^2 + \sum_{i=1}^{n} e_i^2 \qquad (2\text{-}44)$$

This expression holds, as for linear models $\bar{y} = \bar{\hat{y}}$. In vector notation equation (2-44) becomes (Johnson [1988]):

$$\mathbf{y}^T \mathbf{y} - n\bar{y}^2 = \hat{\mathbf{y}}^T \hat{\mathbf{y}} - n\bar{y}^2 + \mathbf{e}^T\mathbf{e} \qquad (2\text{-}45)$$

Here $n\bar{y}^2$ is the model bias. This is the offset by which the model is corrected so that any data correlation is effectively obtained from data with zero means.

The coefficient of determination can now be defined as (Draper [1981] and Hogg [1992]):

$$R^2 = \frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} = \frac{SSM}{SST} = 1 - \frac{SSE}{SST} \qquad (2\text{-}46)$$

The coefficient of determination can therefore be seen as a measure of explaining the fraction of the total variation explained by the model for a particular dependent variable and it follows that $0 \leq R^2 \leq 1$. R^2 is a summary statistic that measures how well the model fits the data. For linear models the R^2-value is equal to the square of the correlation coefficient, $r_{y\hat{y}}$ in equation (2-9).

In analogy to (2-45) the total variance for multivariate statistics can be summated as (Johnson [1988]):

$$\mathbf{Y}^T\mathbf{Y} - n\bar{\mathbf{y}}\,\bar{\mathbf{y}}^T = \hat{\mathbf{Y}}^T\hat{\mathbf{Y}} - n\bar{\mathbf{y}}\,\bar{\mathbf{y}}^T + \mathbf{E}^T\mathbf{E} \qquad (2\text{-}47)$$

The term on the left side of the equation is the same as the variance-covariance matrix, **S**, in (2-12) except that the degrees of freedom have been omitted. The diagonal elements represent the sum-square totals (SST) of the individual dependent variables, thus

$$n\mathbf{S}_T = n\mathbf{S}_M + \mathbf{E}^T\mathbf{E} \qquad (2\text{-}48)$$

The off-diagonal elements in $\mathbf{E}^T\mathbf{E}$ represent the error covariances, i.e. a positive value means that over- or under-fitting the one variable will lead to over- or under-fitting the other variable. From the additive property of the trace operator (Anton [1994]) defined in (2-13) it follows that,

$$tr\left(n\mathbf{S}_T\right) = tr\left(n\mathbf{S}_M\right) + tr\left(\mathbf{E}^T\mathbf{E}\right) \qquad (2\text{-}49)$$

The total multivariate explained variance is therefore related to the definition of R^2 given by (2-46) and can be calculated as (Wise [2000]),

$$\eta^2 = 1 - \frac{tr\left(\mathbf{E}^T\mathbf{E}\right)}{tr\left(n\mathbf{S}_T\right)} \qquad (2\text{-}50)$$

It is evident that (2-50) is a generalisation of R^2 in (2-46) because in the case where there is only a single dependent variable, η^2 becomes R^2. However, it may be required to evaluate each dependent variable individually and therefore it is convenient to distinguish between multivariate and individual variable explained variance. Here η^2 denotes the explained variance in the so-called y-block variables.

2.5.1 Mean-square Error

In the section above it was shown that the sum-square error is related to the total variability in a dependent variable, y_i and in section 2.2 it was shown that because the residual, e_i, is regarded as a random variable the same can be said about y_i. For this reason the mean-square error is the residual variance and its population estimation becomes (Martens [1989]):

$$\varepsilon = E\left[\left(y - \hat{y}\right)^2\right] \tag{2-51}$$

The mean-square error is estimated from its training/estimation sample set, using (n-df) degrees of freedom, by (Hogg [1992] and Van der Voet [1999]):

$$\text{MSE} = \frac{1}{\text{n-df}}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{2-52}$$

Here df is the number of degrees of freedom of the predicted values, $\hat{y}$ determined by the number of parameters (m+1) of the linear model plus the bias. The square root of MSE known as the root mean-square error (RMSE) is often favoured because its units are the same as y_i. This makes it a useful measure of the standard deviation of the error.

It can further be shown that ε is the sum of the variance of the residual, e, and the squared bias (Draper [1981] and Martens [1989]):

$$\varepsilon = E\left[\left\{\left(y - \hat{y}\right) - E\left(y - \hat{y}\right)\right\}^2\right] + \left[E\left(y - \hat{y}\right)\right]^2 = \sigma_e + \text{bias}^2 \tag{2-53}$$

The bias represents the offset between the actual model function and the model function obtained by regression. If this offset (systematic error) is zero the regression model used is correct, otherwise it has a value that depends on some true model. The residual variance, σ_e, is based on the random residuals (random noise) and is therefore a measure of the random error variance. For the case where the bias is zero the mean-square error thus becomes an estimate of the residual variance and $\varepsilon = \sigma_e$.

2.5.2 Pure Error

In data sets where some dependent variable measurements are repeated the data set is said to contain replicates. It must be noted that a replicate is a repeat on identical or very similar independent variables using measurements on different objects. For example, if the blood pressures of humans are to be modelled from their body masses, replicates would be obtained by measuring the weights of numerous individuals bearing the same body mass. The repeated measurement of the weight of the same individual cannot be regarded as a replicate as it supplies information on the variation of the testing method and not on the variation of blood pressure.

Replicates provide information on the variation of the individual independent variables. In equation (2-17) it is shown that the distribution about an independent prediction is assumed to be normal. In order to define this variation numerically, the following notation is defined for a data set of n total measured sample points, m independent variables and a single dependent variable, y for $g = 1,2,\ldots,n_h$ and $h = 1,2,\ldots,k$:

$y_{11}, y_{12}, \ldots, y_{1g}, \ldots, y_{1n_1}$ are n_1 replicates at $\mathbf{x}_1^T$

$y_{21}, y_{22}, \ldots, y_{2g}, \ldots, y_{2n_2}$ are n_2 replicates at $\mathbf{x}_2^T$

$y_{h1}, y_{h2}, \ldots, y_{hg}, \ldots, y_{hn_h}$ are n_h replicates at $\mathbf{x}_h^T$

$y_{k1}, y_{k2}, \ldots, y_{kg}, \ldots, y_{kn_k}$ are n_k replicates at $\mathbf{x}_k^T$

The total number of measurement samples can therefore be calculated as:

$$n = \sum_{h=1}^{k} \sum_{g=1}^{n_h} 1 = \sum_{h=1}^{k} n_h \qquad (2\text{-}54)$$

The contributions of internal variations or sum of squares for n_h sample measurements at $\mathbf{x}_h$ are summed up to form the overall sum-square pure error:

$$SSP = \sum_{h=1}^{k} \sum_{g=1}^{n_h} \left(y_{hg} - \bar{y}_h\right)^2 \qquad (2\text{-}55)$$

The pure error variance, s_P is obtained by dividing SSP by the number of replicates (degrees of freedom), $n_R = n–k$.

The pure error is therefore a measure of random variation because it represents the variation between multiple measurements on identical values of $\mathbf{x}_h^T$. For this reason it is accepted that s_P is another reliable estimator of ε (Draper [1981]).

The variation described by the pure error is built into the model and total variation. The sum-square error, SSE can be adjusted by subtracting the pure error, SSP, from it. This yields the lack-of-fit sum of squares, SSL. The mean-square lack-of-fit error, MSE_L, is obtained by dividing SSL by the degrees of freedom $n_L = n\text{-}n_R\text{-}df$. It can further be shown that SSP and SSL are additive by (Brereton [1990] and Draper [1981]):

$$SSE = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = \sum_{h=1}^{k} \sum_{g=1}^{n_h} \left(y_{hg} - \bar{y}_h\right)^2 + \sum_{h=1}^{k} n_h\left(\hat{y}_h - \bar{y}_h\right)^2 = SSP + SSL$$

(2-56)

The pure error can now be introduced into an analysis of variance. The procedure is to compare the ratio $F = MSE_L/s_P$ with the F_{crit}–value at an α-significance level of the F-distribution with n_L and n_R degrees of freedom. If the ratio is significant, i.e. $F>F_{crit}$ the regression model appears to be inadequate. If the ratio is not significant, i.e. $F<F_{crit}$ there appears to be no reason to discard the model (Draper [1981]).

Due to the fact that the pure error is built into the model and therefore into the total data variance, a data set containing replicates can never achieve an R^2-value of 1 unless $s_P = 0$. The maximum value of R^2 can therefore be obtained by making use of the minimum SSE and SST values calculated as follows:

$$\mathrm{SSE}_{\min} = \sum_{h=1}^{k} \left(\bar{y}_h - \hat{y}_h\right)^2 \qquad (2\text{-}57)$$

$$\mathrm{SST}_{\min} = \sum_{h=1}^{k} \left(\bar{y}_h - \bar{y}_h\right)^2 \qquad (2\text{-}58)$$

$$R^2_{\max} = 1 - \frac{\mathrm{SSE}_{\min}}{\mathrm{SST}_{\min}} \qquad (2\text{-}59)$$

Here the residuals are redefined and by using these residuals the maximum explained variance ($\eta^2_{\max}$) can be calculated using equation (2-50). The value of $R^2_{\max}$ can also be obtained from the square of the correlation ($r_{y\hat{y}}$) between the model predictions and the mean target values. The minimum mean-square error is calculated by dividing $\mathrm{SSE}_{\min}$ by n_L=n-n_R-df degrees of freedom.

By inspection it is evident that the first summation in equation (2-55) is simply a sum of squares or variance multiplied by its degrees of freedom. For this reason and by the definition in equation (2-12) the multivariate sum-square pure error can be defined as:

$$\mathbf{S}_P = \sum_{h=1}^{k} n_h \mathbf{S}_h \qquad (2\text{-}60)$$

Here $\mathbf{S}_h$ is the variance-covariance matrix for each set of dependent replicates measured on $\mathbf{x}_h^T$.

2.6 Model Validation

The validation and testing of a model is paramount when it comes to evaluating the integrity of a proposed model. It is therefore important to see how well the model behaves when fitting unseen data from the same system that is being modelled. Ideally there should be enough available data to be able to validate and finally test the model. The validation step of modelling determines the complexity (e.g. number of regression parameters) of the model within which the model can best generalise the trend of the available data. Data validation should therefore provide information as to whether the model is under-fitting or over-fitting the data and to give an indication of how well the model performs. The best way of judging the performance of a model is by testing it on newly generated, unseen data.

2.6.1 Internal Model Validation

Linear models can be validated internally (by the training/estimation data only) using classical techniques of inference to obtain confidence limits on the predicted values and the regression coefficients (Martens [1989], Draper [1981] and Hogg [1992]). The assumption that the regression parameters of a non-linear model are normally distributed no longer holds when it is assumed that the prediction error or residual, e_i, is normally distributed (Draper [1981]). For this reason the validation of non-linear models can be done using a Bayesian approach (Beale [2000]) or by making use of validation data sets (Geisser [1975]).

The mean-square error of estimation can be estimated when the predicted output variables are tested on the actual training/estimation data set for output variable j as in equation (2-52):

$$\mathrm{MSE}_j = \frac{1}{\text{n-df}} \left(\mathbf{e}_j^T \mathbf{e}_j \right) \qquad (2\text{-}61)$$

2.6.2 External Validation

In situations where the data sets are large enough, the data can be split up into two subsets of n' and n samples each: the training data and the validation data. The model parameters are estimated using the training data and the obtained model is subsequently tested on the validation data. The (integrated) mean-square error of prediction is calculated for n validation sample points as (Van der Voet [1999] and Martens [1989]):

$$\mathrm{MSEP}_j = \frac{1}{n} \left(\mathbf{e}_j^T \mathbf{e}_j \right) \qquad (2\text{-}62)$$

The parameter estimation or training can be repeated, each time increasing the number of parameters. Initially the value of MSEP will decrease and at the point where over-training starts taking place the MSEP will start to increase (see section 2.6.4 below). The MSE-value based on the training/estimation data will continue to decrease as it starts fitting noise in the data.

In an example presented by Baffi [1999b] the entire data set consisted of 998 samples. The data describes the process of a tank where a strong acid stream (HNO_3) is neutralised by a strong base stream (NaOH) in the presence of a buffer stream ($NaHCO_3$). The level of the tank is controlled by adjustment of the neutralised outflow stream. The pH of the tank is the modelled dependent variable and the four streams are the input variables. This data set was subsequently split into a training set of 699 samples and a validation set of 299 samples. Care was taken to ensure that the validation set is a good representative of the entire data set. This can be done by, for example, making use of the DUPLEX data splitting algorithm as described in Snee [1977].

It is interesting to point out that Snee [1977] simply refers to the data splitting method mentioned above as "data splitting" or cross-validation. Martens [1989] refers to the method as external validation. Further examples are given by Hadjiiski [1999] and Bertran [1999]. Haykin [1999] describes cross-validation as being a split of the data into a training set and a test set. The training set is then split into a validation and an estimation subset. The training is done on the estimation set and the validation set is used to find the model parameters for good generalisation. The test set is then used to verify the generalisation ability determined by the entire training set. Examples of this method can be found in Blanco [1999] and Blanco [2000].

In order to avoid confusion the data splitting method illustrated in the examples above will be referred to as external validation in this text. The data splitting method described by Haykin [1999] above can be referred to as external validation with testing.

2.6.3 Model Cross-validation

It becomes difficult to validate sparse data sets, as the omission of sample points can lead to a loss of valuable information about the variable space. This is often the case when experimental procedures are extremely expensive. The problem can be dealt with by an effective design of experiments (DOE). However, the need for cross-validation, as discussed shortly, becomes inevitable for extremely sparse data sets and no applied DOE.

The method of ‘leave-one-out’ cross-validation is described by Stone [1974] and Urban Hjorth [1994]. It is simply illustrated in Fig. 2-2. Here an entire data set of n samples is split into n subsets where each subset consists of n-1 estimation/training samples and an n^{th} point is left out for validation. Ultimately, all the samples have been treated as predicted validation samples and the mean-square error of cross-validation is calculated for the entire data set of validated samples as (Urban Hjorth [1994] and Martens [1989]):

$$\mathrm{MSECV}_j = \frac{1}{n}\left(\mathbf{e}_j^T\mathbf{e}_j\right) \tag{2-63}$$

The value of MSECV is thus calculated using the data in the far-right column of Fig. 2-2. Many texts use the so-called PRESS value (predicted residual estimate sum of squares), which is simply the MSECV value, multiplied by n:

$$\mathrm{PRESS}_j = \left(\mathbf{e}_j^T\mathbf{e}_j\right) \tag{2-64}$$

This is convenient for multivariate dependent variables as:

$$\mathrm{PRESS} = tr\left(\mathbf{E}^T\mathbf{E}\right) \tag{2-65}$$

An alternative is to leave out more than one data point for each split (Geisser [1975]). This can be done by splitting the training set into k (k < n) subsets and then training only k times, each time testing about $({}^{1}/_{k})^{th}$ part of the entire training set. This is again done in such a way that each sample point has been left out for validation prediction after full cross-validation has been completed. The advantages of leaving out more than one sample at a time include saving computational processing time and allowing for more flexibility when handpicking specific combinations of predictions when there are different classes within the data.

Generally, a distinction is made between cross-validation as seen in the context explained in section 2.6.2 above and ‘leave out more than one’ (Geisser [1975]) or ‘leave-one-out’ cross-validation (Draper [1981]). Martens [1989], Urban Hjorth [1994] and Wold [1978] refer to the ‘leave out’ methods simply as cross-validation. Thus, in order to avoid confusion, the definitions as proposed by Martens [1989] will be adopted in this text and a distinction is made between cross-validation and external validation.

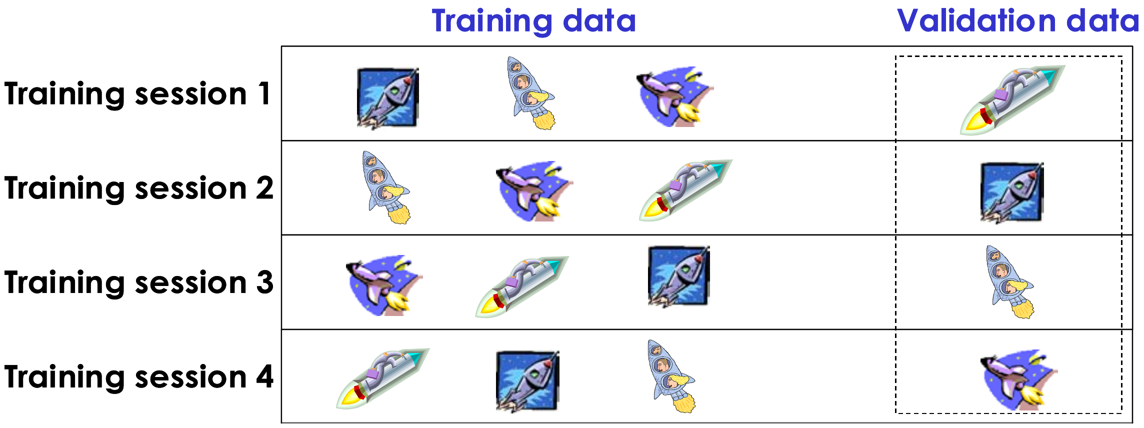

Fig. 2-2 An illustration of leave-one-out cross-validation. The far right column resembles the full set of n data sample points.

2.6.4 Over-fitting and Under-fitting

In theory the prediction error (MSEP or MSECV) is composed of two major contributions: systematic error and random measurement error (Martens [1989]). Systematic error (bias) is the result of un-modelled interference, i.e. influences from variables not yet modelled adequately due to a lack of model complexity. The random error is noise that cannot be controlled and must therefore not become part of the model. The modelling of this noise will lead to over-training.

The two contributions to the prediction error have contradicting trends as the complexity of the model increases. This is illustrated in Fig. 2-3. In practice, the systematic error will only decrease with a further increase in complexity (or number of regression parameters) if the training data is sufficiently representative of the new validation data.

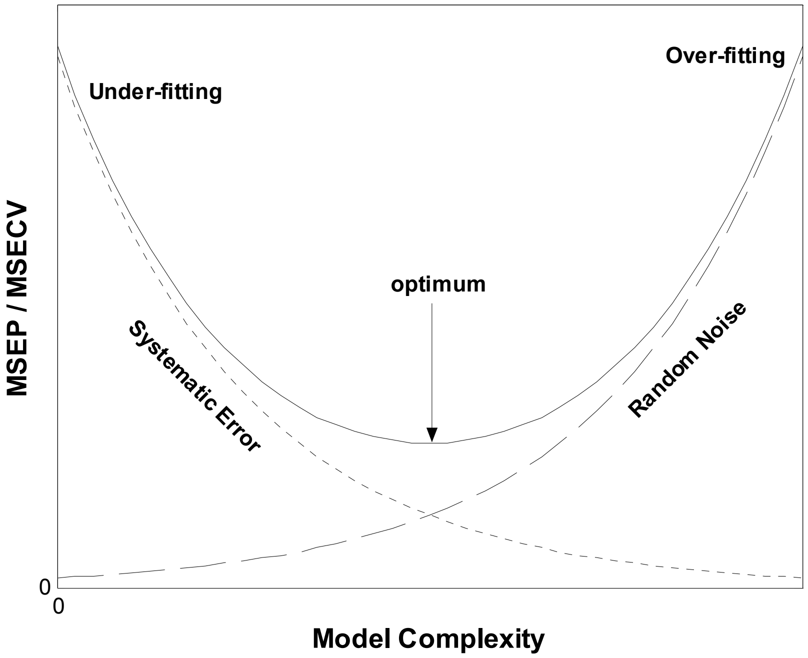

Fig. 2-3 The validation concept illustrated. The optimum point marks the trade-off between over-training and under-training.

2.7 K-means Clustering

Clustering is regarded as unsupervised learning where the aim is to group data points into classes that are more closely related by some measure than objects from other classes. This is done without prior knowledge of class membership. Numerous algorithms exist that attempt to cluster data points according to some rule based on a measure of similarity and a method involving the implementation of an algorithm.

The K-means clustering algorithm uses the method of optimal partitioning where the individual data points are partitioned into K clusters in order to optimise some criterion. For h = 1,2,…,K clusters the algorithm seeks K means, $\bar{\mathbf{x}}_h$, that represent the centres of each class. For g = 1,2,…,n_h representing the number of data points on m variables in each class the criterion to be optimised is the sum of squares (Bishop [1995]),

$$SSK = \sum_{h=1}^{K} \sum_{g=1}^{n_h} (\mathbf{x}_{hg} - \bar{\mathbf{x}}_h)^T(\mathbf{x}_{hg} - \bar{\mathbf{x}}_h) \qquad (2\text{-}66)$$

This expression has a similar form to the pure error in equation (2-55). In fact, it is the trace of $\mathbf{SS}_P$ in (2-60) for k = K. The batch version of K-means (Bishop [1995]) begins by assigning each point in **X** to one of K clusters and then calculating the mean of each initial cluster. Each point is then re-assigned to a cluster depending on which mean is closest to the particular point. This is repeated and the convergence point is reached when there is no further modification in the class memberships of the data points. After each iteration, SSK is calculated and convergence is signified by an unchanged or approximated percentage change of SSK from one iteration to the next.

2.8 Principal Components Analysis (PCA)

Due to the ever-increasing advances in computational power and capacity of computers increased amounts of data are being recorded more frequently and processed in chemical processes. This may cause a data overload, i.e. no useful information may be extracted from the data. It may become necessary to compress the data or extract only the relevant information. In IR spectrum analysis, for example, there is usually redundant information in the many wavelengths recorded. The application of principal components analysis therefore addresses the following three requirements:

- The data must be compressed such that the essential information is retained and can be more easily displayed or represented than the individual variables.
- The essential information often lies in the way variables co-vary (correlate) than in an individual variable.
- In the presence of large amounts of noise it may be necessary to make use of some sort of signal averaging.

PCA relies upon an eigenvector decomposition of the covariance matrix of the process variables. If the data in **X** has been standardised or mean centred then,

$$\mathbf{S} = \frac{1}{n}\mathbf{X}^T\mathbf{X} \qquad (2\text{-}67)$$

For eigenvectors, **p** and eigenvalues, λ there are k = 1,2,…,m decompositions:

$$\mathbf{S}\mathbf{p}_k = \lambda_k\mathbf{p}_k \qquad (2\text{-}68)$$

In PCA the eigenvectors, $\mathbf{p}_k$ are known as loading vectors and they are orthonormal (i.e. $\mathbf{p}_{k'}^T\mathbf{p}_k = 0$ for k' ≠ k and $\mathbf{p}_k^T\mathbf{p}_k = 1$). The loadings can now be used to obtain the score vectors,

$$\mathbf{t}_k = \mathbf{X}\mathbf{p}_k \qquad (2\text{-}69)$$

where $\mathbf{t}_{k'}^T\mathbf{t}_k = 0$ for k' ≠ k. This allows for the decomposition of **X** into the sum of the outer products of $\mathbf{t}_k$ and $\mathbf{p}_k$ and the residual matrix, **F** as:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \ldots + \mathbf{t}_k\mathbf{p}_k^T + \mathbf{F} \qquad (2\text{-}70)$$

Here the residual, f is defined as in equation (2-19) for the x-variables:

$$f_{ik} = x_{ik} - \hat{x}_{ik} \qquad (2\text{-}71)$$

It is therefore possible to approximate **X** by :

$$\hat{\mathbf{X}} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \ldots + \mathbf{t}_k\mathbf{p}_k^T \qquad (2\text{-}72)$$

Here k must be smaller than or equal to the smallest dimension, i.e. k ≤ min(n,m). The score vector, $\mathbf{t}_k$ is thus the linear combination of the original **X** data defined by $\mathbf{p}_k$. Another way of looking at this is that $\mathbf{t}_k$ are the projections of **X** onto $\mathbf{p}_k$. The $\mathbf{t}_k$, $\mathbf{p}_k$ pairs are then arranged according to λ_k such that $\lambda_1 > \lambda_2 > \ldots > \lambda_k > 0$. This is because each $\mathbf{p}_k$ represents a set of co-ordinates rotated relative to the original co-ordinates of $\mathbf{x}_k$ and the λ_k values are the measures of variance in the directions of $\mathbf{p}_k$. PCA ensures that the $\mathbf{t}_1$, $\mathbf{p}_1$ pair captures the greatest amount of variation in the data that is possible to capture with a linear factor. Each subsequent pair captures the greatest possible amount of variance remaining after subtracting $\mathbf{t}_k\mathbf{p}_k^T$ from **X**. The value, λ_k is therefore a measure of information explained by each principal component (PC) and defines the fraction of variance captured by

$$\eta^2 = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_k + \ldots \lambda_m} \qquad (2\text{-}73)$$

The data can therefore possibly be adequately described by using far fewer components or factors than original variables. The data overload often experienced can be solved by retaining fewer scores (weighted sums of the original variables) than original variables. This can be done with no significant loss of information. It is also often found that PCA turns up combinations of variables that are useful descriptions of particular events or phenomena.

The theory of PCA is extensively published and the approach adopted in this text is well illustrated in Johnson [1988] and Diamantaris [1996]. An alternative approach is the so-called non-linear iterative partial least squares (NIPALS) algorithm described in Martens [1989]. Here each PC is extracted iteratively by projection pursuit. The advantage of this algorithm is that the PC's are extracted in chronological order starting at the primary PC.

The scores calculated by PCA have zero mean if the original data has initially been mean-centred. These scores can therefore be used together with the eigenvalues (variances) to calculate the t-statistics. The sum of normalised squared scores, known as the Hotelling T^2-statistic, is a measure of the multivariate t-statistic over a multivariate normal distribution and is defined here as (Johnson [1988] and Wise [2000]):

$$T_i^2 = \mathbf{t}_i^T \Lambda^{-1} \mathbf{t}_i \qquad (2\text{-}74)$$

Here Λ is the k×k diagonal matrix with the eigenvalues corresponding to the k principal components on the diagonal. The vector, $\mathbf{t}_i$ is the k-dimensional vector of scores corresponding to data points, i=1,2,…,n. The confidence limits for the values of T^2 are calculated by using the *F*-distribution for degrees of freedom k and n-k and for the confidence interval of 1-α as follows:

$$T^2_{k,\,n\text{-}k,\,\alpha} = \frac{(n\text{-}1)k}{n\text{-}k} F_{k,\,n\text{-}k,\,\alpha} \tag{2-75}$$

The multivariate residual or Q-statistic for data point, i is simply the sum of squares over the ith data sample vector (Wise [2000]):

$$\text{Q-stat} = \mathbf{f}_i^T\mathbf{f}_i \tag{2-76}$$

The confidence limits for the Q-statistic for confidence interval 1-α can be calculated from the following equation (Wise [2000]):

$$\text{Q-stat}(\alpha) = \Theta_1\left[\frac{\xi_\alpha\,(2\Theta_2)^{1/2}\,\zeta}{\Theta_1} + 1 + \frac{\Theta_2\zeta(\zeta\text{-}1)}{\Theta_1^2}\right]^{1/\zeta} \tag{2-77}$$

where

$$\Theta_b = \sum_{c\,=\,k+1}^{m} \lambda_c^b \qquad \text{for } b = 1,2,3 \tag{2-78}$$

$$\zeta = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2} \tag{2-79}$$

Here ξ_α is the standard normal deviate corresponding to the upper 1-α percentile. The test statistics above are defined for normally distributed data, which means that an assumption has to be made that this is the case for the scores obtained. This assumption can be justified by the central limit theorem, which states that sums of several different groups will tend to be normally distributed, regardless of the probability distribution of the individual groups (Wise [2000]).

2.9 Class-based Principal Components Analysis

PCA removes correlations within a data set by finding primary axes or principal components in the directions of maximum variance perpendicular to each other and then transforming the original space to the new de-correlated space. Second order statistical moments can be introduced by considering class membership within the data set and by finding the principal axes in the direction of maximum variance between the classes (Fukunaga [1972]).

Class-based PCA can be divided into two separate transformations. The first transformation aims to 'whiten' the individual clusters. This transformation removes correlations within the clusters and in so doing effectively groups the class members within a class closer together. In order to achieve this the intra-class variance-covariance matrix is defined for $\mathbf{X}_h$, the $n_h \times m$ centred data belonging to a predefined class:

$$\mathbf{S}_h = \frac{1}{n_h} \mathbf{X}_h^T \mathbf{X}_h \qquad \text{for } h = 1,2,\ldots,k \qquad (2\text{-}80)$$

The average over all intra-class variance-covariance matrices can now be calculated as:

$$\mathbf{S}_G = \sum_{h=1}^{k} p(\mathbf{X}_h)\, \mathbf{S}_h \qquad (2\text{-}81)$$

Here $p(\mathbf{X}_h)$ is the probability of class h occurring. The matrix, $\mathbf{S}_G$ is known as the intra-class spread matrix.

If $\mathbf{S}_G$ can now be transformed to an identity matrix it would mean that the features in the new space, $\mathbf{S}_G'$ are not correlated. In order to achieve this it is necessary to find some $m \times m$ transformation matrix, $\mathbf{A}$ such that:

$$\mathbf{I} = \mathbf{A}^T \mathbf{S}_G \mathbf{A} \qquad (2\text{-}82)$$

Here $\mathbf{I}$ is the $m \times m$ identity matrix. It can be shown that equation (2-82) is possible by equating $\mathbf{A}$ as follows (Fukunaga [1972]):

$$\mathbf{A} = \mathbf{P}\Lambda^{-1/2} \qquad (2\text{-}83)$$

Here $\mathbf{P}$ is the $m \times m$ matrix of eigenvectors of $\mathbf{S}_G$ and Λ is the $m \times m$ diagonal matrix of eigenvalues of $\mathbf{S}_G$. The transformation in equation (2-82) has the effect of obtaining equal variances, on average, in all dimensions in the new space.

In similar fashion the inter-class variance-covariance matrix, $\mathbf{S}_H$ can be defined:

$$\mathbf{S}_H = \sum_{h=1}^{k} p(\mathbf{X}_h)\,(\bar{\mathbf{x}}_h - \bar{\mathbf{x}})(\bar{\mathbf{x}}_h - \bar{\mathbf{x}})^T \qquad (2\text{-}84)$$

Here $\bar{\mathbf{x}}$ is the mean over the entire data set, **X**. It must be noted here that for classes having equal probability of occurring, $p(\mathbf{X}_h)$ is simply equal to $^1/_k$. The matrix, $\mathbf{S}_H$ can then simply be calculated from:

$$\mathbf{S}_H = \frac{1}{k}\bar{\mathbf{X}}^T\,\bar{\mathbf{X}} \qquad (2\text{-}85)$$

Here $\bar{\mathbf{X}}$ is the k×m centred data of the k class means. $\mathbf{S}_H$ therefore gives an indication of how the class centres vary relative to one another. It is therefore used to obtain the second transformation where the principal component lies in the direction of maximum variation between the clusters.

Just like the intra-class spread matrix, $\mathbf{S}_G$ is transformed the inter-class variance-covariance matrix is transformed as:

$$\mathbf{S}'_H = \mathbf{A}^T\mathbf{S}_H\mathbf{A} \qquad (2\text{-}86)$$

Now, in order to obtain the principal axes in the directions of maximum variation between the transformed class/cluster centres, the eigenvector-eigenvalue decomposition of $\mathbf{S}_H$' leads to the matrix of eigenvectors, **V** and the m×m diagonal matrix of eigenvalues, Γ. The complete transformation for class-based PCA to a set of transformed scores is thus given by:

$$\mathbf{T} = \mathbf{X}\;\mathbf{A}\;\mathbf{V} = \mathbf{X}\;\mathbf{P}\Lambda^{-1/2}\;\mathbf{V} \qquad (2\text{-}87)$$

The transformation vectors of **AV** are commonly known as canonical variates and represent the transformed axes just as is the case in PCA. The eigenvalues in Γ can now be used to obtain the hierarchy of maximum variances between classes, similar to PCA.

If a class-based PCA transformation is successful in separating the classes in such a way that clear boundaries are visible/obtainable between classes, the classes can be classified successfully using some classification model. The classes can be completely separated using k dimensions of the transformed space. However, analogous to PCA, it may be possible to choose the primary number of dimensions h ≤ k required to obtain a clear distinction between the classes.

2.10 Non-linear Optimisation Techniques

In section 2.2.2 it was mentioned that in the case of non-linear regression the minimisation of the cost function could only be approximated using optimisation algorithms that approach the minimum. It seldom happens that a global minimum can be reached, especially with increasing complexity (large number of parameters) of the model.

Consider the functional form of some non-linear model as introduced in section 2.2:

$$\mathbf{y} = f(\mathbf{X},\theta) + \mathbf{e} \qquad (2\text{-}88)$$

The problem now becomes one of differentiating the error surface represented by the least squares cost function in equation (2-14). The solution for r = 1,2,…,M parameters and M differential equations is non-linear and various strategies as discussed below can be deployed to solve the problem.

2.10.1 The Linearisation or Taylor Series Method

If the functional form of the model in equation (2-88) is smooth and differentiable with respect to θ it can be approximated by a Taylor series (Draper [1981] and Baffi [1999a]):

$$f(\mathbf{x_i},\theta) = f(\mathbf{x}_i,\theta_0) + \sum_{r=1}^{M} \frac{\partial f(\mathbf{x}_i,\theta_{r0})}{\partial \theta_r}\left(\theta_r - \theta_{r0}\right) = \hat{y}_i|_0 + \sum_{r=1}^{M} \frac{\partial f}{\partial \theta_r}\bigg|_0 \Delta\theta_{r0} \qquad (2\text{-}89)$$

and thus for all i = 1,2,…,n,

$$\mathbf{y} - \hat{\mathbf{y}}|_0 = [\mathbf{J}\ \Delta\theta]_0 \qquad (2\text{-}90)$$

where for the Jacobian matrix, $\mathbf{J}_0$,

$$j_{ir}|_0 = \frac{\partial f(\mathbf{x}_i,\theta_{r0})}{\partial \theta_r} \qquad (2\text{-}91)$$

Equation (2-90) is in linear form and by linear regression using equation (2-27) the pseudo inverse of **J** can be calculated to obtain

$$\Delta\theta_0 = \left[(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\ \mathbf{e}\right]_0 \qquad \text{(2-92)}$$

Here the pseudo-inverse of **J** is better evaluated using SVD described in section 2.3 above, as some of the derivatives in function *f* may be correlated.

An iterative procedure can now be introduced by evaluating equation (2-92) at iteration t = 1,2,… as:

$$\theta_t = \theta_{t-1} + \Delta\theta_{t-1} \ = \ \theta_{t-1} + \left[(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\ \mathbf{e}\right]_{t-1} \qquad \text{(2-93)}$$

Convergence occurs when for an arbitrarily small δ,

$$\frac{(\theta_t - \theta_t)^T(\theta_{t-1} - \theta_{t-1})}{\theta_t^T\theta_t} \leq \delta \qquad \text{(2-94)}$$

The disadvantage of this algorithm is that divergence may occur because of "over-shooting" caused by $\Delta\theta$. Marquardt [1963] mentions that this problem can be addressed by multiplying $\Delta\theta$ by a fraction, φ. The choice of a value for φ may be cumbersome and in this text the "bold driver" method in Bulsari [1995] is used to adapt φ after each iteration:

- Increase φ by γ if the error, $[\mathbf{e}^T\mathbf{e}]_t$ is decreasing, $\varphi_t = \gamma\varphi_{t-1}$. Choose γ close to 1, e.g. 1.05.
- Decrease φ more drastically by ι if the error is increasing, $\varphi_t = \iota\varphi_{t-1}$. Choose $\iota \approx 0.5$

This method is usually used for updating the learning rate used by the gradient descent method discussed shortly. It has been applied here with great success, as the form of equation (2-93) is similar to that of the gradient descent method.

2.10.2 The Gradient Descent Method

This is one of the simplest gradient-based optimisation algorithms used for minimising the cost function (Bishop [1995], Haykin [1999] and Draper [1981]). If a non-linear function such as a sigmoidal function is used (see section 2.11) the derivative of the cost function, $\varepsilon(\theta)$ with respect to the parameters in θ can easily be determined if it is assumed that ε (θ) is smooth and differentiable for all θ.

The iterative procedure is initialised by guessing an initial θ. This set of parameters is then updated by moving a small increment in θ-space in the direction that decreases $\varepsilon(\theta)$ most rapidly, i.e. $-\nabla\theta$. This iterative process thus generates a sequence of θ values whose components are updated by:

$$\theta_r|_t = \theta|_{t-1} + \varphi \left.\frac{\partial\varepsilon(\theta)}{\partial\theta_r}\right|_{\theta_{t-1}} \qquad (2\text{-}95)$$

The choice of the learning rate φ is crucial as a value too small will make the reduction in the error too slow and a value too large will cause divergent oscillations. The prime weakness of the gradient descent method is its slow convergence and the fact that the exclusion of divergences is not guaranteed.

2.10.3 The Marquardt Method

The highly efficient Marquardt algorithm (Marquardt [1963]) addresses the weaknesses of the Taylor series and gradient descent methods. It does this by finding a compromise between the Taylor series method and the gradient descent method. It shares with the gradient descent methods the ability to converge from an initial guess outside the region of convergence. It shares with the Taylor series method the ability to rapidly close in on the converged values after the vicinity of the converged values has been reached.

The Marquardt algorithm makes use of a parameter, μ to modify the Taylor series method to:

$$\Delta\theta_0 = \left[\left(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}\right)^{-1}\mathbf{J}^T\ \mathbf{e}\right]_0 \qquad (2\text{-}96)$$

For the n×M Jacobian matrix, **J** the matrix, **I** is the r×r identity matrix.

From a series of proofs presented in Marquardt [1963] it is shown that μ allows for prompt convergence using finite arithmetic. This is achieved without the prerequisite condition that *f* is a smooth, well-behaved function as is the case for the gradient descent method. It is further shown that $\Delta\theta$ of the Taylor and gradient descent methods are at almost 90 degrees to each other. The $\Delta\theta$ term in the Marquardt algorithm rotates to close the angle between itself and the $\Delta\theta$ of the gradient descent method for each iteration as $\mu\rightarrow\infty$.

The disadvantage of the Marquardt algorithm is that it requires high and intensive memory usage. The application of this algorithm is limited to optimisation problems with relatively few parameters on personal computers with low memory resources.

2.11 Feed-forward Multi-layer Perceptron Neural Networks

Neural networks have been extensively studied and applied in practice as non-linear function approximaters. The power of neural networks lies in their architecture derived from the biological functionality of the neurons, dendrite trees and synapses of the human brain (Haykin [1999] and Zupan [1993]). This architecture can assume almost any functional form, which in turn can be used in pattern recognition and regression.

The simplest neural networks consist of two layers, the input layer and the output layer. The input layer consists of the input variables in **X**. The output layer consists of l nodes for the l predicted output variables in **Y**. A node consists of the weighted summation of inputs, which is in turn put through a transfer function, ϕ to obtain the node outputs. The functional form of a node for a two-layer network is thus:

$$\hat{y}_{ij} = \phi_j \left\{ \sum_{k=r=1}^{m} \left(\omega_{jr} x_{ik}\right) + \beta_j \right\} \qquad (2\text{-}97)$$

Here each node receives all its inputs from the previous layer without feeding any information back to it. This is known as a feed-forward neural network. A neural network can have a number of so-called hidden layers sandwiched in-between the input and output layers, each layer consisting of a number of nodes. Such a neural network becomes a so-called multi-perceptron neural network. In this text the input layer is numbered as zero with all subsequent layers numbered from 1 downwards.

Numerous different types of transfer functions can be used for ϕ. There are three generic types of transfer functions, the threshold function, the piece-wise linear function and the non–linear sigmoid functions (Haykin [1999]). In this text the tan-sigmoidal function is used as non-linear transfer function:

$$\phi(\mathrm{x}) = tanh(\mathrm{x}) = \frac{1 - exp(-2\mathrm{x})}{1 + exp(-2\mathrm{x})} \qquad (2\text{-}98)$$

The advantage of using this function is that its derivative can be written as a function of ϕ itself:

$$\phi'(\mathrm{x}) = 1 - \phi^2(\mathrm{x}) \qquad (2\text{-}99)$$

The tan-sigmoidal function has the property that its outputs are limited to the range, $-1 \leq \phi(\mathrm{x}) \leq 1$.

In this text a neural network architecture using a single hidden layer of tan-sigmoidal transfer functions is used. A special case of a piece-wise linear transfer function, the linear combiner, is used for the output nodes. This transfer resembles non other than a simple linear transformation to the output variables. The functional form of such a single layered neural network, with superscripts in brackets denoting the layer number, is given here in explicit form as

$$\hat{y}_{ij} = \sum_{r=1}^{H} \omega_{jr}^{(2)} \phi_r \left\{ \sum_{k=1}^{m} \left(\omega_{rk}^{(1)} x_{ik} \right) + \beta_r^{(1)} \right\} + \beta_j^{(2)} \qquad (2\text{-}100)$$

This functional form will be referred to as a sigmoidal neural network in this text. Schematically, the sigmoidal neural network is illustrated in Fig. 2-4.

The parameters of a neural network function are obtained using various training algorithms. The most common algorithm used for training feed-forward neural networks is the back-propagation algorithm.

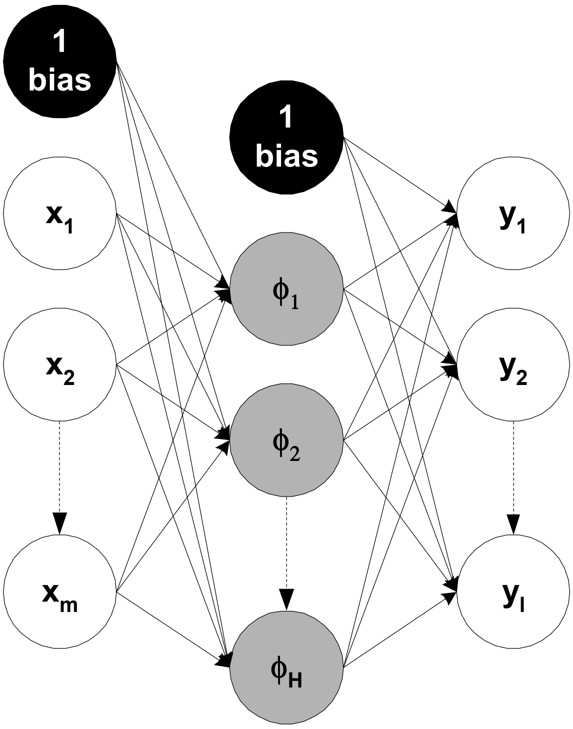

Fig. 2-4 The architecture of a feed-forward multi-perceptron neural network for multiple inputs and outputs.

2.11.1 The Back-propagation Algorithm

Most training algorithms involve the minimisation of a cost function such as the sum-square error in equation (2-14). These algorithms can be split into two distinct stages. In the first stage the derivatives of the cost function with respect to each parameter must be calculated. It is at this stage that the back-propagation algorithm, given by Rumelhart [1986], provides a computationally efficient means of calculating the derivatives by propagating the function error backwards through the neural network layers. In the second stage (discussed in 2.11.2 below) the derivatives are used to calculate the adjustments made to the parameters by some non-linear optimisation algorithm (see section 2.10).

In general each node computes a weighted sum of its inputs as discussed above:

$$g_q = \sum_p \omega_{qp} z_p \qquad (2\text{-}101)$$

Here z_p is the activation from node p in the preceding layer and ω_{pq} is the weight from the q^{th} node in the current layer associated with z_p. A bias can be included by introducing $z_0 = 1$. The summation in (2-101) is put through the non-linear activation function, ϕ to give:

$$z_q = \phi(g_q) \tag{2-102}$$

It must be noted that z_p is the same as $z_k = x_k$ if it represents the input layer and z_q is the same as $z_j = y_j$ if it represents the output layer. For each sample, i, back-propagation is initialised by a forward pass of $\mathbf{x}_i$ through the network. The set of all weights ω is initialised and the predicted outputs for $\mathbf{y}_i$ are calculated. The initialisation of the weights can be done using a random initialiser.

From the definition of a cost function in equation (2-14) the error, E_i for each data point, i over l output variables can now be defined as:

$$E_i = \sum_{j=1}^{l} \varepsilon_j(\mathbf{x}_i, \boldsymbol{\omega}) \tag{2-103}$$

The object is to find the derivative of E_i with respect to the weights, ω in order to be able to proceed with an optimisation algorithm to minimise E_i. It can be recognised that E_i depends on ω_{qp} via g_q and by the chain rule:

$$\frac{\partial E_i}{\partial \omega_{qp}} = \frac{\partial E_i}{\partial g_q}\frac{\partial g_q}{\partial \omega_{qp}} = \delta_q\, z_p \tag{2-104}$$

A useful notation has been introduced here where the nodal error is defined as

$$\delta_q = \frac{\partial E_i}{\partial g_q} \tag{2-105}$$

Equation (2-101) has been used to obtain:

$$z_p = \frac{\partial g_q}{\partial \omega_{qp}} \tag{2-106}$$

Thus, in order to calculate the derivatives in (2-104) all that is needed is to obtain the value of δ_q for each hidden and output node. From the definition in (2-104) and from (2-102) with j substituted for q the calculation of δ_j at the output layer becomes:

$$\delta_j = \frac{\partial E_i}{\partial g_j} = \phi'(g_j)\,\frac{\partial E_i}{\partial y_j} \qquad (2\text{-}107)$$

The chain rule can be used in order to calculate δ_q for the hidden node:

$$\delta_q = \frac{\partial E_i}{\partial g_q} = \sum_j \frac{\partial E_i}{\partial g_j}\frac{\partial g_j}{\partial g_q} \qquad (2\text{-}108)$$

Here the implication is that any change in g_q gives rise to a change in E_i through a change occurring in g_j. It should be noted that g_j could be from an output node or any other hidden node. Equation (2-105) can now be substituted into (2-108) and by using (2-101) and (2-102) the back-propagation formula is obtained as:

$$\delta_q = \phi'(g_q) \sum_j \omega_{jq}\delta_j \qquad (2\text{-}109)$$

A back-propagation pass is thus summarised as:

- Forward propagate an input vector, $\mathbf{x}_i$ through the network using (2-101) and (2-102).
- Evaluate δ_j for all output nodes using (2-107).
- Back-propagate all of δ_j using (2-109) and obtain δ_q for each hidden node.
- Calculate the error function derivatives using (2-104).

The derivatives of the total error function are then obtained by repeating the steps above for each data point pair $\mathbf{x}_i$,$\mathbf{y}_i$ and then summing over all the data points:

$$\frac{\partial E}{\partial \omega_{qp}} = \sum_{i=1}^{n} \frac{\partial E_i}{\partial \omega_{qp}} \qquad (2\text{-}110)$$

The back-propagation pass described above is repeated, each repeat known as an epoch.

2.11.2 Updating the Weights

During each epoch the weights are updated using a non-linear optimisation algorithm, the simplest one being that of the gradient descent method described in section 2.10.2. For batch training the weight updates are therefore obtained by:

$$\omega_{qp}|_t = \omega_{qp}|_{t-1} + \Delta\omega_{t-1} = \omega_{qp}|_{t-1} - \eta\sum_{i=1}^{n}\delta_{iq}z_{ip} \qquad (2\text{-}111)$$

This learning rule is known as the Widrow-Hoff rule.

The partial derivatives calculated from the back-propagation algorithm can also be used directly to form the Jacobian matrix, **J**. The Levenberg-Marquardt algorithm (Hagan [1994]), which is the Marquardt algorithm adapted for training neural networks can then be used for adapting the weights (see section 2.10.3). This is much more efficient than the Widrow-Hoff rule. However, the Levenberg-Marquardt method is highly memory intensive and is subsequently better suited for the training of smaller data sets. Convergence of the Levenberg-Marquardt algorithm is extremely quick and accurate.

2.12 Radial-Basis Functions (RBF)

Radial basis functions (RBF) represent another major class of the neural network model. Here the distance between the input vector and a prototype vector known as a node centre determines the activation function. Radial basis functions have their origins in the exact interpolation problem where every input vector is to be mapped exactly onto the corresponding target vector.

For exact interpolation, n basis functions of the form $\phi(||\mathbf{x} - \mathbf{x}_i||)$ are used for the mapping from an m-dimensional space in **X** onto a 1-dimensional space in **y**. Here $||\mathbf{x} - \mathbf{x}_i||$ is usually the Euclidean distance between any input vector, $\mathbf{x}^T$ and a data point, $\mathbf{x}_i^T$ from a set of n data points in **X**. The form of the non-linear function ϕ will be discussed below. The following mapping therefore represents exact interpolation (Bishop [1995]):

$$f(\mathbf{x}) = \sum_{i=1}^{n}\omega_i\phi(||\mathbf{x} - \mathbf{x}_i||) \qquad (2\text{-}112)$$

Here the ω_i for i = 1,2,…,n are the weights and can be obtained by the projection,

$$\boldsymbol{\omega} = \boldsymbol{\Phi}^{-1}\mathbf{y} \tag{2-113}$$

where $\boldsymbol{\Phi}$ is an invertible n×n matrix of elements $\phi(||\mathbf{x} - \mathbf{x}_i||)$. For this problem it has been shown that the interpolation is relatively insensitive to the form of the non-linear function ϕ (Bishop [1995]). Several forms of the basis function have been suggested, the most common one being that of the Gaussian,

$$\phi(x) = exp\left(\frac{-(x - \mu)^2}{2\sigma}\right) \tag{2-114}$$

Here σ is the parameter that controls the smoothness of the interpolation function *f* and μ is some centre. The Gaussian is a localised function with the properties, $\phi \rightarrow 0$ as $|x| \rightarrow \infty$. Another commonly used function is the thin-plate spline function,

$$\phi(x) = (x - \mu)^2\, ln(x - \mu) \tag{2-115}$$

A number of modifications to the exact interpolation problem have been introduced in order to obtain the neural network model. These modifications introduce an interpolation function in which the number of basis functions is determined by the complexity of the function rather than the number of data points present. These modifications include:

- The number of basis functions, H need not equal n, i.e. $H \leq n$. Each basis function effectively becomes a hidden node with a Gaussian transfer function.
- The centres of the basis functions are no longer restricted to the input data vectors and may be determined during the training process.
- The width/spread parameter, σ of each Gaussian transfer function can be unique.
- A bias term is included in each output node in order to compensate for differences in the data means of the target y-data and the outputs of the transfer functions or nodes.

In this text the Gaussian transfer function is used as basis function due to its popularity in literature and, in general, it was found to produce slightly better results on dummy data sets. The Gaussian can now be simplified into a form where the 2 in equation (2-114) is omitted due to the fact that the smoothness parameter is adjusted uniquely for each transfer function in the network and the presence of a constant is thus trivial. The Gaussian function is thus:

$$\phi_{ir}(\mathbf{x}_i) = exp\left(\frac{-\|\mathbf{x}_i - \mathbf{c}_r\|^2}{\rho_r^2}\right) \qquad (2\text{-}116)$$

Here ρ is known as the spread parameter and $\mathbf{c}_r$ is an m×1 centre. For output variable, j and data point, i the general form for RBF neural networks can now be written as:

$$y_{ij} = \sum_{r=1}^{H} \omega_{jr}\phi_r(\mathbf{x}_i) + \beta_j + e_i \qquad (2\text{-}117)$$

The linear parameters, ω and β are known as weights and biases respectively in order to maintain neural network conventions. For multivariate y variables and for $\phi_{ir}(\mathbf{x}_i)$ the elements of $\mathbf{\Phi}$ the general form can now be written as

$$\underset{n\times l}{\mathbf{Y}} = \underset{n\times H\times H\times l}{\mathbf{\Phi\Omega}} + \underset{n\times 1\times 1\times l}{\mathbf{1}\boldsymbol{\beta}^{T}} + \underset{n\times l}{\mathbf{E}} \qquad (2\text{-}118)$$

The RBF neural network only has a single hidden layer. This architecture is graphically illustrated in Fig. 2-5. Hartman [1990] gives a formal proof that RBF neural networks using Gaussian transfer functions and this architecture can be used for universal approximation.

A two-stage training process usually deals with the training of RBF neural networks. The first stage can be implemented using an unsupervised or a supervised RBF centre selection algorithm. The second stage is a supervised linear regression from the hidden node outputs using singular value decomposition (SVD) in order to avoid ill-conditioned or singular matrices. In this text the orthogonal least squares (Bishop [1995]) and K-means (Haykin

[1999] and Bishop [1995]) algorithms are considered as supervised and unsupervised first stage selectors of RBF centres, respectively.

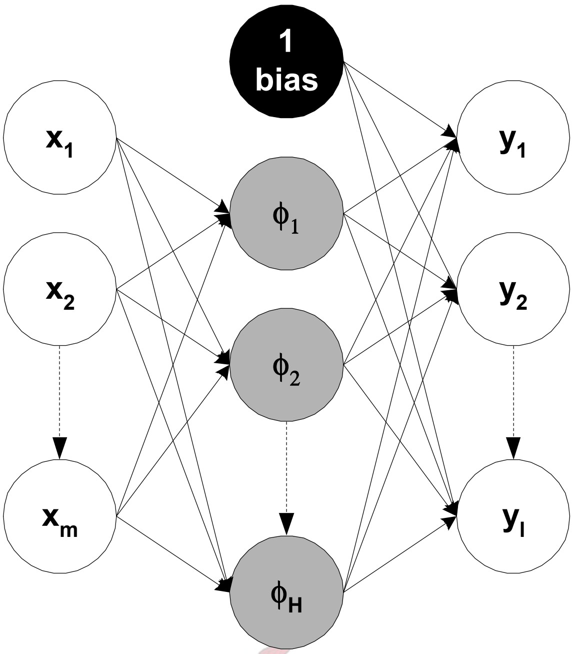

Fig. 2-5 Architecture of the RBF neural network. Each line from a hidden node to the output node represents a weight.

2.12.1 Training by K-means

The selection of suitable centres for a RBF being trained on a set of data can be done by random selection. An improvement on this is to use the K cluster means obtained from clustering the input data, **X** as described in section 2.7 above. This method is very fast and not very memory intensive, however no output data, **Y** is ever seen for the selection process. The disadvantage of this is that the result of the training is highly sensitive to the initialisation of K-means algorithm. In order to maintain consistent results a fixed method of initialisation is suggested in this text. An initialisation algorithm that aims to find the first K furthest points within **X** is used. The results from this method were found to be superior to a random initialisation of the K-centres.

Once the K centres (note: K = H) have been chosen the spread parameters, ρ need to be determined in such a way that they do not overlap too severely and that they are not too pointy or too flat. A discussion of this problem influencing a RBF network is given in Bishop [1995]. The N-nearest neighbours heuristic can be used to find a suitable solution:

$$\rho_r = \left(\frac{1}{N} \sum_{n=1}^{N} \|\mathbf{c}_r - \mathbf{c}_{rn}\|^2 \right)^{1/2} \qquad (2\text{-}119)$$

The hidden node outputs can now be calculated using equation (2-116). The pseudo inverse of Φ can now be obtained by making use of SVD on the matrix as described in section 2.3 above.

2.12.2 Training by Orthogonal Least Squares (OLS)

This method can be regarded as a more principled approach due to its supervised method of choosing RBF centres. The OLS algorithm developed by Chen [1991] chooses the RBF centres one-by-one in an iterative manner. This is done so that during each step the chosen centre maximises the increment to the explained variance of the desired output. An added advantage of this method is that the second stage of the training does not suffer from any ill-conditioned numerical problems. The chosen RBF centres do however come from the original data used for training and it may therefore be expected that large amounts of data are needed for this training algorithm.

Equation (2-117) can be reduced to a special case linear regression model for one output variable where the hidden node outputs are regarded as the regression model inputs, hence $\phi_r(\mathbf{x}_i) = v_{ir}$:

$$y_i = \sum_{r=1}^{H} v_{ir}\omega_r + e_i \qquad (2\text{-}120)$$

This is best written in matrix notation as:

$$\mathbf{y} = \mathbf{V}\omega + \mathbf{e} \qquad (2\text{-}121)$$

From linear regression analysis in section 2.2.1 above and section 2.5 it is clear that the sum-square of $\mathbf{V}\omega$ explains part of the output energy that is contributed by the regressors stemming from the chosen RBF centres. However, some of these regressors are generally correlated and it is not clear which regressor contributes to this energy. OLS now involves the transformation of **V** into a set of orthogonal vectors. This makes it possible to calculate the individual contribution by each regressor and hence the contribution by each RBF centre to the desired output energy. The objective is therefore to obtain the orthogonal set of regressors, Ψ in the same original space, **V** such that:

$$\mathbf{y} = \Psi\theta + \mathbf{e} \qquad (2\text{-}122)$$

where

$$\theta = \left(\Psi^T\Psi\right)^{-1}\Psi^T\mathbf{y} \qquad (2\text{-}123)$$

The Gram-Schmidt method can be used to orthogonalise **V**. The objective here is to select H candidate regressors out of a possible total of n. A criterion for choosing the candidates can be obtained by first of all investigating the sum-square of **y**,

$$\mathbf{y}^T\mathbf{y} = \sum_{r=1}^{H}\theta_r^2\psi_r^T\psi_r + \mathbf{e}^T\mathbf{e} \qquad (2\text{-}124)$$

If the data in **y** and Ψ is mean-centred this equation explains the model and total variance as in equation (2-45). From the expression it is therefore clear that $\theta_r^2\psi_r^T\psi_r$ explains the increment to the explained output variance influenced by ψ_r. The error reduction ratio (ERR) due to the introduction of ψ_r is now defined as

$$\mathrm{ERR}_r = \frac{\theta_r^2\psi_r^T\psi_r}{\mathbf{y}^T\mathbf{y}} \qquad (2\text{-}125)$$

This ratio can now be used to find a subset of significant regressors and thus RBF centres in a forward regression manner. The regression selection procedure is now explained in the following algorithm:

- **Step 1 (r = 1,2,…,H)**

  Gram-Schmidt orthogonalisation:

  $\psi_1^{(r)} = \mathbf{v}_r$

  Calculate ERR:

$$\theta_1^{(r)} = \frac{\left(\psi_1^{(r)}\right)^T \mathbf{y}}{\left(\psi_1^{(r)}\right)^T \psi_1^{(r)}}$$

$$\mathrm{ERR}_1^{(r)} = \frac{\left(\theta_1^{(r)}\right)^2 \left(\psi_1^{(r)}\right)^T \psi_1^{(r)}}{\mathbf{y}^T\mathbf{y}}$$

Find:

$$\mathrm{ERR}_1^{(r_1)} = \max\left\{ \mathrm{ERR}_1^{(r)} \right\}$$

Select

$\psi_1 = \psi_1^{(r_1)} = \mathbf{v}_{r_1}$ and thus $\mathbf{c}_1 = \mathbf{c}_1^{(r_1)}$

- **Step t (t = 2,3,…,H and r = 1,2,…,H where r ≠ r_1,…,r ≠ r_{t-1})**

  Gram-Schmidt orthogonalisation:

  $$\alpha_{lt}^{(r)} = \frac{\psi_l^T \mathbf{v}_r}{\psi_l^T \psi_l} \qquad \text{(for } 1 \le l < t\text{)}$$

  $$\psi_t^{(r)} = \mathbf{v}_r - \sum_{l=1}^{t-1} \alpha_{lt}^{(r)} \psi_l$$

  Calculate ERR:

  $$\theta_t^{(r)} = \frac{\left(\psi_t^{(r)}\right)^T \mathbf{y}}{\left(\psi_t^{(r)}\right)^T \psi_t^{(r)}}$$

  $$\mathrm{ERR}_t^{(r)} = \frac{\left(\theta_t^{(r)}\right)^2 \left(\psi_t^{(r)}\right)^T \psi_t^{(r)}}{\mathbf{y}^T\mathbf{y}}$$

  Find:

  $$\mathrm{ERR}_t^{(r_t)} = \max\left\{ \mathrm{ERR}_t^{(r)} \right\}$$

  Select

  $\psi_t = \psi_t^{(r_t)}$ and thus $\mathbf{c}_t = \mathbf{c}_t^{(r_t)}$

- The procedure is terminated at the Hth step when the desired number of hidden nodes is obtained or the model sum-square error has reached a threshold value.

In summary, firstly a pool of all n data points is considered as potential RBF centres. Secondly, a value for each spread parameter is calculated as will be discussed below. Thirdly, a predefined number of hidden nodes or RBF centres, H is extracted from the training data points step by step according to the OLS algorithm. Finally, a linear regression step (using SVD) mapping the training data outputs onto the hidden node outputs, including bias, completes the training. SVD may be omitted here due to the parsimonious selection of centres by the OLS algorithm.

In practice there is no need to mean-centre **y** or ψ in order to obtain the explained variance in ERR as this does not influence the maximisation process. From the OLS algorithm above it becomes clear that the H most parsimonious RBF centres to the outputs can be selected. However, much can still be said about the calculation of the Gaussian spread values.

OLS training of radial basis function networks is usually applied to smooth function interpolations where the inputs are equally incremented (e.g. time series) and a single spread value for all centres can then be calculated using equation (2-119). For better results Beale [2000] suggest a formula that ensures overlap between the RBF centres in such a way that good interpolation results can be obtained:

$$\rho_r = \frac{1}{-\mathit{ln}(0.5)}\left(\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{c}_r - \mathbf{c}_{rn}\|\right) \qquad (2\text{-}126)$$

This equation ensures that the rth RBF 'fires' a value of 0.5 and above for any points within the average distance that $\mathbf{c}_r$ is to its N nearest neighbours. The spread values influence the outcome of the hidden nodes and therefore the regressors in (2-120) as well. Because these spread values are predefined before the OLS decision algorithm is applied it is suggested in Beale [2000] that the spread values should be varied and the network retrained until a good result is obtained. This works well for input values incremented at constant intervals but what happens when this is not the case? If equation (2-126) is used to calculate the spread values of input data, which is erratically distributed, it may be found that after a certain number of hidden neurons have been chosen the spread values do not efficiently cover the input space. This is where the K-means method has an advantage. It therefore seems that for erratically spaced input data there is a trade-off between the K-means and OLS methods as far as RBF centre choice and spread value calculation are concerned.

In this text an algorithm is therefore suggested whereby the spread values can be adjusted iteratively until the input space is more evenly and therefore more efficiently covered by the Gaussian functions. The algorithm is given here for modelling the n×m input data, **X** on a single output variable, **y**:

- **Step1**

1. Consider all n data points of **X** as potential RBF centres and calculate the n×n matrix, **D**, of mutual distances between the data points with
   $d_{kk'} = ||\mathbf{x}_k - \mathbf{x}_{k'}||$ (for k = 1,2,…,n and k' = 1,2, ,…,n)
2. Use equations (2-119) or (2-126) with N predefined nearest neighbours to calculate the n×1 vector, ρ_1 of Gaussian spread parameters for each of the RBF centres.
3. Calculate the n×n matrix of hidden node outputs, $\mathbf{\Phi}_1$ using equation (2-116).
4. Use $\mathbf{V}_1 = \mathbf{\Phi}_1$ as inputs for the OLS algorithm and obtain the H candidate centres by maximum contributions towards explained variance. The value of H is pre-determined. $\mathbf{C}_1$ now becomes the H×m matrix of H RBF centres.
5. Use $\mathbf{C}_1$ to calculate the new n×H matrix, $\mathbf{V}_1$ and regress **y** on $\mathbf{V}_1$ with bias. This is done using equation (2-27) to obtain the weights, ω_{rl} in the H×l matrix, $\mathbf{\Omega}_1$ and the l×1 bias vector, β_1.
6. Calculate SSE_1 using equations (2-19) and (2-43).
7. Calculate H new, adapted spread parameters with equations (2-119) or (2-126) by using the RBF centres obtained in 4. In H×1 vector, χ_1 store the indices of the H centres chosen from the original n×m data.
8. For H < n substitute the spread parameters indexed in χ_1 into ρ_1 to obtain ρ_2.

- **Step t ≥ 2**

9. Consider all n data points of **X** as potential RBF centres as in 1.

10. Calculate the n×n matrix of hidden node outputs, Φ_t using equation (2-116) together with ρ_t.
11. Use $\mathbf{V}_t = \Phi_t$ as inputs for the OLS algorithm and obtain the H candidate centres of maximum contributions towards explained variance. $\mathbf{C}_t$ now becomes the H×m matrix of H RBF centres.
12. Use $\mathbf{C}_t$ to calculate the new n×H matrix, $\mathbf{V}_t$ and regress $\mathbf{y}$ on $\mathbf{V}_t$ with bias. This is done using equation (2-27) to obtain the weights,ω_{rl} in the H×l matrix, Ω_t and the l×1 bias vector, β_t
13. Calculate SSE_t using equations (2-19) and (2-43). Break the loop if
$$\frac{|SSE_t - SSE_{t-1}|}{SSE_t} = 0$$
14. In $\mathbf{C}_t$ find the ν_t centres that were not present in $\mathbf{C}_{t-1}$ and add their data point indices to complete the $(H+\nu_1+\ldots+\nu_{t-1}+\nu_t)\times 1$ vector, χ_t.
15. Calculate ν_t new, adapted spread parameters with equations (2-119) or (2-126) (using N predefined nearest neighbours) by considering the $(H+\nu_1+\ldots+\nu_{t-1})$ candidate centres from the t-1 previous iterations as the possible nearest neighbours.
16. For $\nu_t < n$ substitute the spread parameters into ρ_t to obtain ρ_{t+1}.
17. Go to 9

Originally, step 7 in the algorithm was used for $t \geq 2$ in the place of steps 14 and 15. This however led to oscillating SSE values. The adaptations in steps 14 and 15 ensure that SSE converges to a constant value. This convergence is due to the constant adaptation of the spread parameters from a set of candidate values. These values are returned in the full set in n×1 vector, ρ at the indices corresponding to the data points used as candidate centres. This ensures that even though changes are made to some of the spread values belonging to the candidate centres the algorithm can still see if it can do better and converge from a candidate set larger than H centres.

The algorithm above can thus be regarded as an adaptive spread OLS algorithm (ASOLS). The algorithm still requires predefined nearest neighbour values that can vary from one training session to another. These values, however need to be varied for K-means training as well and are much more intuitive to choose than the actual spread parameter values. Results from a small data set have shown that this algorithm fairs better than the K-means training method and the simple OLS method. The application of this model to the data in this text can be observed in Chapter 4 and Chapter 5.

2.13 Partial Least Squares (PLS)

In the discipline of chemometrics chemists make use of statistical methods to help them make quantitative measurements of chemical samples by use of near-infrared (NIR) spectroscopy. If it is known what the exact chemical composition of a sample is the varying percentage composition of each species can be modelled using Beer's law. Beer's law simply states that the absorbance at a specific wavelength in the NIR band is a linear function of the chemical concentration of a pure species. This application requires that pure reference samples first be scanned by NIR at varying wavelengths and that the absorptivity-path length coefficients be obtained from the measurements. For mixtures of different chemical species the peaks of the NIR spectra are influenced additively with the addition of any new chemicals.

For large amounts of data the application above can be tedious and can be dealt with in a multivariate manner by simply applying classical least squares (CLS) as described in equation (2-32) (Haaland [1988] and Lorber [1987]). Here **Y** contains the n absorbance values at l wavelengths and **X** contains the n concentrations for m chemical species. The absorptivity-path length coefficients are then obtained by projecting **Y** onto **X**. The problem with this approach is that it is assumed that the errors lie in the measurement of the spectra and not in the measurement of the concentrations. This leads to the problem that all the chemical species in **Y** must be known. The advantages, however, are that CLS is a full spectrum method (no wavelengths are left out) and qualitative information can be obtained by examining the influences of the individual parameters.

Another approach is that of inverse least squares (ILS) (Haaland [1988] and Lorber [1987]). Here **Y** contains the n concentration values for l chemical species and **X** contains the n absorbance values for the m wavelengths. The assumption here is that the measurement error lies in the concentrations. This has the advantage over CLS in that not all the chemical species need to be known in order to infer on the species of interest. The problem of ILS is that not all frequencies can be used (not a full spectrum method) as there need to be more data samples than input variables (n>m). Another problem is that there may be a large amount of correlation in the absorbance data that needs to be dealt with.

In order to combine the advantages of both the CLS and ILS methods and at the same time avoid some of their disadvantages principal components regression (PCR) was applied (Haaland [1988] and Martens [1989]). In the application of predicting the chemical compositions from the NIR absorbance values the primary dimensions of the input data are first of all retained in the scores matrix, **T**. The chemical composition data in **Y** is then projected onto the low dimensional data in **T**. PCR is thus a full spectrum method like CLS with the added advantage of ILS that not all chemical species in **Y** need to be known. PCR thus compensates for noise in both input and output data. The potential problem of PCR is that the primary dimensions best representing the spectral data may not be optimal for the concentration prediction (Haaland [1988]). It is therefore desirable to be able to obtain loadings so that more predictive information is contained within the primary dimensions. In order to achieve this the PCA NIPALS algorithm as mentioned in section 2.8 can be modified to form the linear PLS algorithm.

2.13.1 The Linear PLS Algorithm

In 1966 Herman Wold (Wold [1966]) pioneered the NIPALS algorithm for partial least squares (PLS). The NIPALS algorithm, which is thoroughly explained and illustrated in Geladi [1986], is considered as the most intuitive algorithm. This becomes evident when considering non-linear modifications to the algorithm. Numerous variations on the linear PLS algorithm exist. The so-called PLS1 algorithm for single dependent variable modelling is given in Lorber [1987], Martens [1989] and Haaland [1988]. For multivariate model outputs the PLS2 (Martens [1989]) algorithm is closely related to the NIPALS algorithm. A computationally more efficient PLS algorithm is given by Lorber [1987].

The objective of the NIPALS algorithm is to project in- and output matrices **X** and **Y** onto a subset of latent variables (with dimension h), **T** and **U**, which are referred to as the in- and output scores, respectively. The output scores can now be fitted to the input scores by linear least-squares regression in order to obtain the so-called inner linear relationship coefficients, b_a for a = 1,2,…,h:

$$\mathbf{u}_a = \mathbf{t}_a b_a + \mathbf{e}_a \tag{2-127}$$

Here the h primary latent dimensions explaining the majority of the model variance are retained in a similar manner to PCA. The decompositions of **X** and **Y** can be defined, as in PCA, using the loading vectors **p** and **q** such that:

$$\mathbf{X} = \sum_{a=1}^{h} \mathbf{t}_a \mathbf{p}_a^T + \mathbf{F} \tag{2-128}$$

$$\mathbf{Y} = \sum_{a=1}^{h} \hat{\mathbf{u}}_a \mathbf{q}_a^T + \mathbf{E} \tag{2-129}$$

Here $\hat{\mathbf{u}}$ denotes the predicted scores of **u** in (2-127). The matrices, **F** and **E** are the resulting residual matrices when a model with h ≤ min(n,m) latent dimensions is used for the approximation of **X** and the prediction of **Y**. Usually the remaining latent dimensions explain the random noise in the data (see discussion in section 2.6). The NIPALS algorithm lies at the heart of PLS. It iteratively extracts a latent variable pair as a linear combination of the input and output variables. In the following step the inner relationship of scores is obtained and the remaining information in **X** and **Y** is used to extract the next latent dimension. The complete algorithm is thus:

0. Mean centre or standardise the inputs and outputs, **X** and **Y**. Initialise the algorithm by setting the output scores, **u** equal to a column of **Y**. For each latent dimension, a = 1,2,…h follow steps 1 to 12 below:

1. Calculate the input weights, **w**, by regressing **X** on **u**:

$$\mathbf{w}^{T} = \frac{\mathbf{u}^{T}\mathbf{X}}{\mathbf{u}^{T}\mathbf{u}}$$

2. Normalise **w** to unit length:

$$\mathbf{w} = \frac{\mathbf{w}}{||\mathbf{w}||}$$

3. Calculate the input scores:

$$\mathbf{t} = \frac{\mathbf{X}\,\mathbf{w}}{\mathbf{w}^{T}\mathbf{w}}$$

4. Calculate output loadings by regressing **Y** on **t**:

$$\mathbf{q}^{T} = \frac{\mathbf{t}^{T}\,\mathbf{Y}}{\mathbf{t}^{T}\,\mathbf{t}}$$

5. Normalise **q** to unit length:

$$\mathbf{q} = \frac{\mathbf{q}}{||\mathbf{q}||}$$

6. Calculate new output scores **u**:

$$\mathbf{u} = \frac{\mathbf{Y}\,\mathbf{q}}{\mathbf{q}^{T}\mathbf{q}}$$

7. Check for convergence on **w** for some δ arbitrarily small:

$$\frac{\left|\,||\mathbf{w}_{t-1}|| - ||\mathbf{w}_{t}||\,\right|}{||\mathbf{w}_{t}||} \le \delta$$

   If yes go to the next step, else start iteration t+1 at step 1.

8. Calculate the input loadings, **p** by regressing **X** on **t**:

$$\mathbf{p}^{T} = \frac{\mathbf{t}^{T}\mathbf{X}}{\mathbf{t}^{T}\,\mathbf{t}}$$

9. Calculate the inner linear regression coefficient b:

$$b = \frac{\mathbf{t}^{T}\,\mathbf{u}}{\mathbf{t}^{T}\,\mathbf{t}}$$

10. Calculate the input residual matrix:

$$\mathbf{F} = \mathbf{X} - \mathbf{t}\,\mathbf{p}^{T}$$

11. Calculate the output residual matrix:

$$\mathbf{E} = \mathbf{Y} - \hat{\mathbf{u}}\,\mathbf{q}^{T}$$

12. If additional PLS latent dimensions are required replace **X** and **Y** with **F** and **E** respectively and return to step 1 for calculation of latent dimension a+1.

In Baffi [1999a] it is further shown that the n×h scores matrix, **T** can be related to the input matrix, **X** by

$$\mathbf{T} = \mathbf{XW'} \tag{2-130}$$

where **W'** is obtained from:

$$\mathbf{W'} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1} \tag{2-131}$$

The practicality of this expression lies in the fact that **T** can be expressed in terms of **W** without having to breakdown **X** into its residuals for each latent dimension. Equation (2-130) can now be used to obtain the overall regression coefficients for PLS by

$$\hat{\mathbf{Y}} = \hat{\mathbf{U}}\mathbf{Q}^T = \mathbf{XW'BQ}^T = \mathbf{XB}_{PLS} \tag{2-132}$$

where

$$\mathbf{B}_{PLS} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{BQ}^T \tag{2-133}$$

It can be shown further that by retaining all latent dimensions, h=k, the linear PLS regression model converges towards the multiple/multivariate linear regression model where

$$\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{2-134}$$

The overall PLS procedure can thus be seen as being two linear outer mappings between the in- and output variables and their corresponding scores, and a linear inner mapping between each subsequent pair of latent variables. The relationship between the outer and inner mappings is illustrated in Fig. 2-6.

In the past, PLS has shown to be a powerful regression technique for problems where the data is noisy, highly correlated and where there are only a limited number of observations. The power of PLS lies in its decomposition of a multivariate regression problem into a number of uncorrelated univariate regression analyses.

Each pair of input-output latent variables accounts for a certain amount of variability in both the input and output data sets, **X** and **Y** and as mentioned earlier the higher order latent dimensions are usually associated with random noise. An appropriate number of latent dimensions can therefore be identified from cross-validation as described in section 2.6. Here the addition of each latent dimension adds to the overall model complexity.

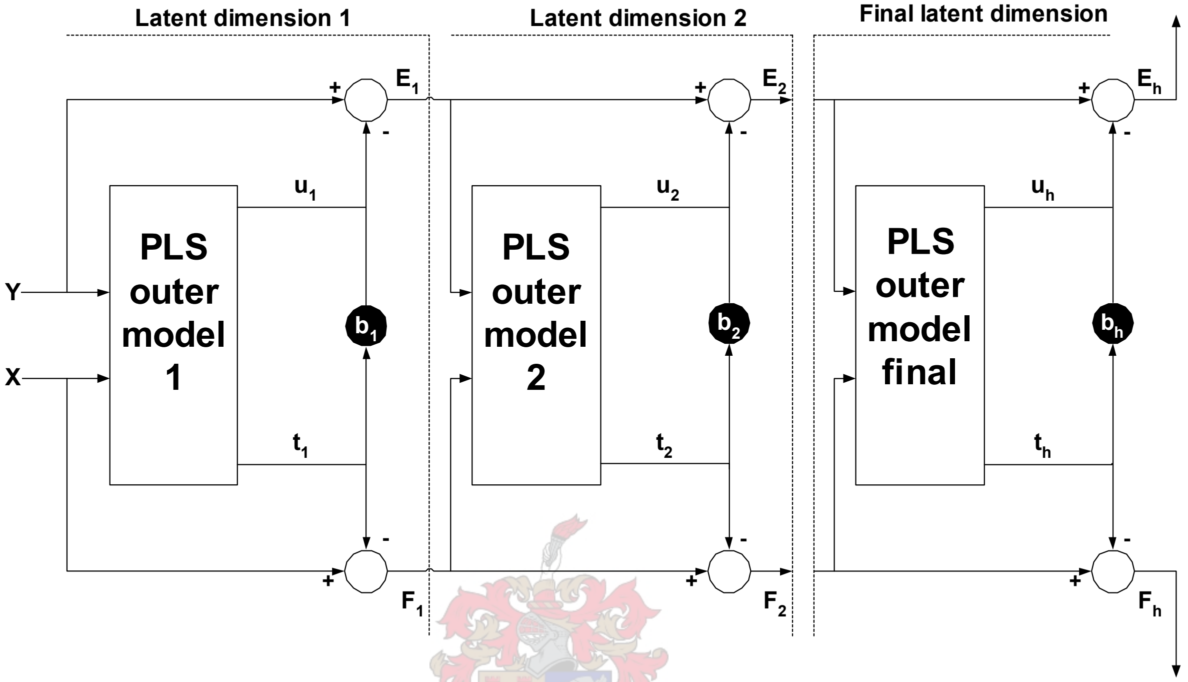

Fig. 2-6 A schematic representation of the PLS model. It illustrates the relationship between the outer and inner models.

2.13.2 The Non-linear Neural Network PLS Approach

The application of linear PLS to inherent non-linear data relationships may lead to problems regarding the omission of the higher order (or minor) latent dimensions. These minor latent dimensions may contain significant information about the non-linear nature of the problem. Non-linear structures may be modelled using a combination of latent variables, which include information from both higher and lower order latent dimensions calculated from linear PLS. It therefore becomes important to be able to retain the maximum amount of relevant information during each latent extraction in order to optimise the choice of latent dimensions to be used in the final model.

A number of non-linear PLS algorithms have been proposed but the seminal work by Wold [1989], which led to the development of the algorithm by Baffi [1999a], is the more conducive approach. Wold [1989] proposed that a quadratic polynomial be used to fit the functional relation between each pair of latent variables (QPLS). The NIPALS algorithm in section 2.13.1 can thus be modified by replacing equation (2-127) with:

$$\mathbf{u}_a = b_{a0} + \mathbf{t}_a b_{a1} + \mathbf{t}_a^2 b_{a2} + \mathbf{e}_a \qquad (2\text{-}135)$$

Step 9 in the NIPALS algorithm now becomes the multiple regression projection as explained in section 2.2.1.1.

The main drawback of merging non-linear inner regression models within the linear PLS outer framework is that the use of a non-linear function influences both the inner and outer mappings of the PLS algorithm. If the inner mapping is highly non-linear this approach may no longer be acceptable. This problem was addressed by Wold [1989] by updating the weights in **w** using a complicated, non-intuitive Taylor series linearisation technique. Wold [1978] did however conclude that weight updating could be omitted if the inner relationship is slightly non-linear.

A drawback with using a quadratic function for the inner mapping is that a non-linear functional form is assumed for the inner relationship. Qin [1992] suggested a generic non-linear PLS approach (NNPLS) using a feed-forward neural network with a single sigmoidal hidden layer for the inner relationship of each pair of latent variables. The advantage of the neural network is that it is a universal approximater and does not require prior knowledge of a functional form. For a sigmoidal neural network the functional relationship, for $\mathbf{t}_a$ set equal to $\mathbf{X}_a\mathbf{w}_a$, in its explicit form becomes (see equation (2-100)):

$$\hat{u}_{ia} = f(\mathbf{x}_{ia}\mathbf{w}) = \sum_{r=1}^{H} \omega_{2r}\phi_r\left(\omega_{1r}\mathbf{x}_{ia}^T\mathbf{w}_a + \beta_{1r}\right) + \beta_2 \qquad (2\text{-}136)$$

Here the first index number on the left of the neural network weights indicates the layer number. This simplified notation is made possible because there are only single input and output variables involved. The latent dimension index, a=

1,2,…h can be omitted because it can be recognised that this equation, within the weight updating algorithm, applies to one latent dimension at a time. Step 9 of the NIPALS algorithm in section 2.13.1 now becomes a neural network-training step to obtain the neural network weights, ω. Qin [1992] did not use weight updating of **w**, which may be a drawback when dealing with highly non-linear inner mappings.

Two other neural network PLS approaches are those of Holcomb [1992] and Malthouse [1997]. Their approaches, however only compare qualitatively with the generic PLS algorithm and essentially assume an overall neural network structure.

Baffi [1999a] proposed an error-based input weights updating procedure using a Taylor series expansion, which is an improvement on the weight updating procedure originally suggested by Wold [1989]. In Baffi [1999b] this weight updating procedure was then later also applied using the neural network (EBNNPLS) of Qin [1992] as inner mapping function. The error-based procedure assumes that the non-linear function used to fit the inner model is smooth and differentiable with respect to the input weights, w. The algorithm will be developed here with an inner mapping using a sigmoidal neural network with r nodes in a single hidden layer as in equation (2-136).

Equations (2-89) to (2-91), in conjunction with equation (2-99) can be applied to equation (2-136) in order to obtain the derivatives (Baffi [1999b]):

$$j_{ik} = \frac{\partial f(\mathbf{x}_i^T \mathbf{w})}{\partial w_k} = \sum_{r=1}^{H} \omega_{2r}\omega_{1r}\left[1 - \phi^2\left(\omega_{1r}t_i + \beta_{1r}\right)\right] x_{ik} \qquad (2\text{-}137)$$

Equation (2-92) is used to obtain the error-based weight updating equation:

$$\Delta\mathbf{w} = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\,\mathbf{e} \qquad (2\text{-}138)$$

The NIPALS algorithm can now be modified in such a way that the input weights are updated at each iteration within the algorithm:

0. Mean centre or standardise the inputs and outputs, **X** and **Y**. Initialise the algorithm by setting the output scores, **u** equal to a column of **Y**. For each latent dimension, a = 1,2,…h follow steps 1 to 19 below:

1. Calculate the input weights, **w**, by regressing **X** on **u**:

$$\mathbf{w}^T = \frac{\mathbf{u}^T\mathbf{X}}{\mathbf{u}^T\mathbf{u}}$$

2. Normalise **w** to unit length:

$$\mathbf{w} = \frac{\mathbf{w}}{||\mathbf{w}||}$$

3. Calculate the input scores:

$$\mathbf{t} = \frac{\mathbf{X}\,\mathbf{w}}{\mathbf{w}^T\mathbf{w}}$$

4. Train the sigmoidal neural network on **t** and **u** to obtain the neural network weights:

$(\omega_{1r},\omega_{2r},\beta_{1r},\beta_2)$ for r = 1,2,…H

5. Calculate the non-linear prediction of **u**:

$$\hat{u}_i = \sum_{r=1}^{H} \omega_{2r}\phi_r\left(\omega_{1r}\mathbf{x}_i^T\mathbf{w} + \beta_{1r}\right) + \beta_2$$

6. Calculate output loadings by regressing **Y** on $\hat{\mathbf{u}}$:

$$\mathbf{q}^T = \frac{\hat{\mathbf{u}}^T\,\mathbf{Y}}{\hat{\mathbf{u}}^T\hat{\mathbf{u}}}$$

7. Normalise **q** to unit length:

$$\mathbf{q} = \frac{\mathbf{q}}{||\mathbf{q}||}$$

8. Calculate new output scores, **u**:

$$\mathbf{u} = \frac{\mathbf{Y}\,\mathbf{q}}{\mathbf{q}^T\mathbf{q}}$$

9. Calculate the Jacobian, **J** and the weight updates, Δ**w** in equations (2-137) and (2-138):

10. Calculate the updated weights:

$\mathbf{w}_t = \mathbf{w}_{t-1} + \Delta\mathbf{w}$

11. Normalise $\mathbf{w} = \mathbf{w}_t$ to unit length:

$$\mathbf{w} = \frac{\mathbf{w}}{||\mathbf{w}||}$$

12. Calculate the new input scores, **t**:

$$\mathbf{t} = \frac{\mathbf{X\,w}}{\mathbf{w}^T\mathbf{w}}$$

13. Check for convergence on **w** for some δ arbitrarily small:

$$\frac{\big|\, ||\mathbf{w}_{t-1}|| - ||\mathbf{w}_t||\, \big|}{||\mathbf{w}_t||} \le \delta$$

If yes go to next step, else start iteration t+1 at step 4.

14. Calculate the input loadings, **p** by regressing **X** on **t**:

$$\mathbf{p}^T = \frac{\mathbf{t}^T\mathbf{X}}{\mathbf{t}^T\,\mathbf{t}}$$

15. Train the sigmoidal neural network on **t** and **u** to obtain the neural network weights:

$(\omega_{1r},\omega_{2r},\beta_{1r},\beta_2)$ for $r = 1,2,\ldots H$

16. Calculate the non-linear prediction of **u**:

$$\hat{u}_i = \sum_{r=1}^{H} \omega_{2r}\phi_r\left(\omega_{1r}\mathbf{x}_i^T\mathbf{w} + \beta_{1r}\right) + \beta_2$$

17. Calculate the input residual matrix:

$$\mathbf{F} = \mathbf{X} - \mathbf{t}\,\mathbf{p}^T$$

18. Calculate the output residual matrix:

$$\mathbf{E} = \mathbf{Y} - \hat{\mathbf{u}}\,\mathbf{q}^T$$

19. If additional PLS latent dimensions are required replace **X** and **Y** with **F** and **E** respectively and return to step 1 for calculation of latent dimension a+1.

In this text two small modifications are made that enhance the speed of convergence of the error-based weight-updating algorithm by Baffi [1999b]. The first modification is to replace steps 0 to 3 by steps 0 to 7 of the linear NIPALS algorithm for the specific latent dimension. In most cases this method produced a slightly better estimate for the initial set of in- and output scores, **t** and **u**. This modification is feasible as the linear PLS algorithm is not computationally intensive. The second modification is to include the learning rate, φ in equation (2-138) as is described in section 2.10.1.

The NNPLS and EBNNPLS algorithms are illustrated in Fig. 2-7. The two algorithms differ in their outer model mappings as seen in the algorithms above.

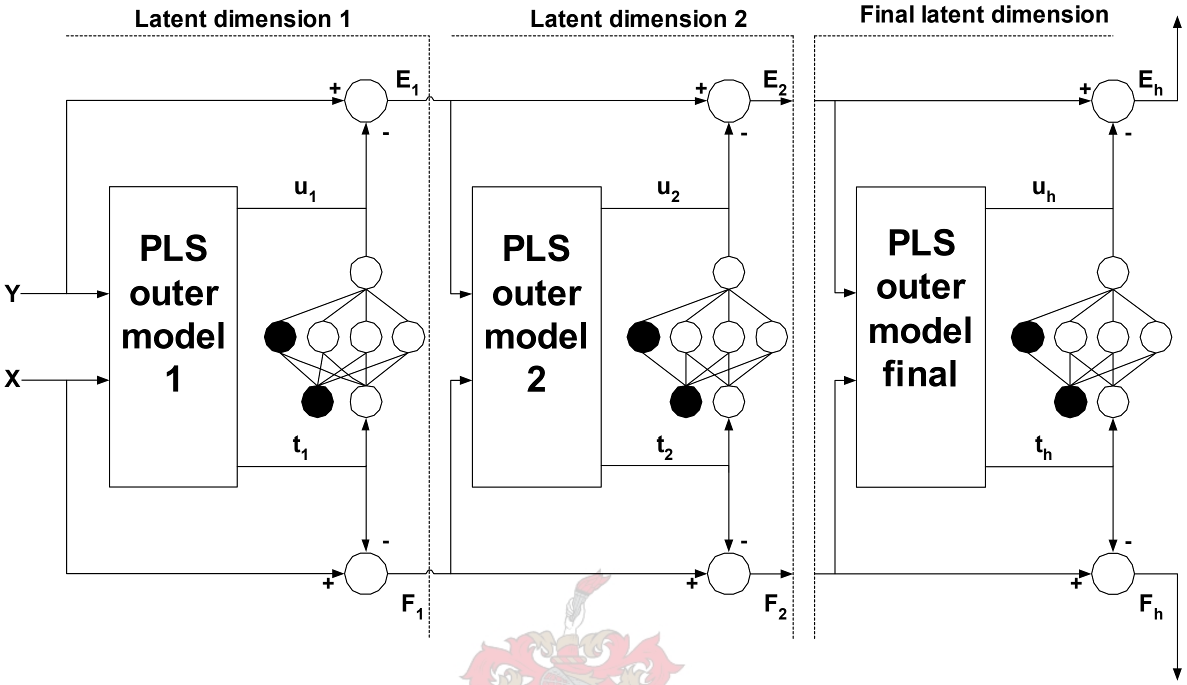

Fig. 2-7 The inner and outer mappings of the NNPLS and EBNNPLS algorithms illustrated.

The choice of training algorithm for the sigmoidal neural network is the Levenberg-Marquardt algorithm (see section 2.11.2). The training data sets are very small due to the single in- and output variables being trained on. Convergence is therefore quick and accurate.

Baffi [1999b] also applied a radial basis function network to the inner PLS mapping (RBFPLS or EBRBFPLS). For a RBF equation (2-136) becomes,

$$\hat{u}_{ia} = f(\mathbf{x}_{ia}\mathbf{w}) = \sum_{r=1}^{H} \omega_r \, exp\left(\frac{-||\mathbf{x}_{ia}^T\mathbf{w}_a - c_r||^2}{\rho_r^2}\right) + \beta \qquad (2\text{-}139)$$

The derivatives for latent dimension, a=1,2,…,h of this function become:

$$j_{ik} = \frac{\partial f(\mathbf{x}_i^T \mathbf{w})}{\partial w_k} = \sum_{r=1}^{H} \omega_r \, exp\left(\frac{-||t_i - c_r||^2}{\rho_r^2}\right)\left(-2\,\frac{||t_i - c_r||^2}{\rho_r^2}\right)\left(\pm\, x_{ik}\right) \qquad (2\text{-}140)$$

for which the following conditions apply:

$+x_{ik}$ if $(t_i - c_r) > 0$

$-x_{ik}$ if $(t_i - c_r) < 0$

Steps 4 and 15 in the error-based weight-updating algorithm above can now be replaced by obtaining $(c_r, \rho_r, \omega_r, \beta)$ from training a RBF on **t** and **u**.

The drawback of building NNPLS or RBFPLS models is that of selecting the number of hidden nodes to be used in the neural network architecture. Baffi [1999b] suggests using an exhaustive search over the number of hidden nodes. Akaike's final prediction error (FPE) can be used to identify the best network architecture (Ljung [1987]):

$$\mathrm{FPE} = \frac{1 + {}^{M}/_{n}}{1 - {}^{M}/_{n}}\,\mathrm{MSEP} \qquad (2\text{-}141)$$

Here the total number of neural network parameters, M can be set equal to 3H+1. The total number of data points, n is chosen from the training data. The mean-square error of the validation data set, MSEP is calculated from the actual and predicted output scores, $\hat{\mathbf{u}}$ and $\mathbf{u}$ respectively. The neural network model with the number of hidden neurons, H corresponding to the minimum FPE-value is used to model the inner relationship of the particular latent dimension. Akaike's FPE penalises models with increased complexity. This provides a conservative, more robust measure of determining the adequate model complexity so as to avoid over-training the model.

In order to be able to make predictions on unseen data the in- and output scores need to be evaluated for the validation data. The scores for the validation set can be calculated as follows:

$$\mathbf{t}_a^{val} = \mathbf{F}_{a-1}^{val}\,\mathbf{w}_a \qquad (2\text{-}142)$$

$$\mathbf{u}_a^{val} = \mathbf{E}_{a-1}^{val}\, \mathbf{q}_a \qquad (2\text{-}143)$$

Here the values of **F** and **E** are calculated from:

$$\mathbf{F}_a^{val} = \mathbf{F}_{a-1}^{val} - \mathbf{t}_a^{val}\, \mathbf{p}_a^T \qquad (2\text{-}144)$$

$$\mathbf{E}_a^{val} = \mathbf{E}_{a-1}^{val} - \hat{\mathbf{u}}_a^{val}\, \mathbf{q}_a^T \qquad (2\text{-}145)$$

Qin [1992] went on to prove that the NNPLS (and therefore also EBNNPLS) algorithm is equivalent to a multi-perceptron feed-forward neural network but is specifically trained by the NNPLS (or EBNNPLS) algorithm. He showed that the NNPLS algorithm using single hidden-layered sigmoidal feed-forward neural networks as inner models could be transformed into the global functional form a single hidden-layered sigmoidal neural network. The number of hidden nodes of this model subsequently equals the total number of hidden nodes added up over all inner models of the latent dimensions obtained from NNPLS-training. An analogous relationship cannot be obtained for RBFPLS (or EBRBFPLS) and a RBF neural network.

2.13.3 The Advantages of Building PLS Models

Feed-forward multi-layer perceptron neural networks are extensively applied in practice due to their ability to approximate almost any non-linear relationship. Furthermore, it may be perceived that because an iterative gradient–related training algorithm is usually used to optimise the cost function the neural network is unaffected by correlations in the input data. Baffi [1999b] explains that in practice neural networks are affected by correlations in the input data and may fail to provide a robust solution. The failure of the network to provide an optimal solution is due to the fact that correlated variables lead to numerous combinations of network weights that can minimise the cost function and still give virtually the same answer. In these circumstances neural networks therefore have the tendency to enlarge the noise in predictions. In chemical processes it is often the case that some variables are simply different reflections of the same event.

A solution to the problem of dealing with highly correlated input data is to make use of PCA in order to reduce the input data to a lower dimensional set of uncorrelated scores. These features are then used as inputs for building a neural network model. The problem with PCA is that the directions of greatest variability within the input data are identified and applied as independent dimensions without considering the correlations that exist between the input and output variables. In practice, some variables may show large variability within the input data but are weakly correlated with the output variables. As a consequence some variables that are highly correlated with the output variables may be more significant in the lower order principal components, which may then be discarded. PLS focuses on extracting principal components or latent dimensions by maximising the correlations between in- and output data. In this way it is certain that the bulk of the information will be retained in the primary latent dimensions.

The use of neural networks within the PLS framework bears numerous benefits over the direct neural network approach. A multiple input-multiple output (MIMO) network regression is reduced to numerous single input–single output (SISO) regressions. This means that each neural network model within the PLS framework makes use of much fewer weights and subsequently the error surface consists of fewer local minima. These two factors avoid the over-parameterisation problem that can affect neural network models built from multiple in- and outputs. The NNPLS and EBNNPLS methods are thus less sensitive to dealing with small or scarce data sets.

The overall effect of NNPLS is therefore increased robustness and smaller prediction variance as a result of reduced sensitivity to correlations in the input data and the SISO regressions used in the inner mapping of the PLS framework.

2.14 Pseudo-degrees of Freedom

In section 2.5.1 it was shown how the sample mean-square error can be calculated by dividing the SSE by the degrees of freedom, n-df. Here df is the number of degrees of freedom of the regression model. For the case of linear models this is simply the number of regression parameters (the bias included) or alternatively the number of input variables to the model (m+1).

In the case of non-linear models the total number of parameters is commonly used as an estimate for the model degrees of freedom. It is an estimate as the SSE of the training data is not a global minimum (see section 3.1.3) and the regression parameters are obtained by an iterative procedure as opposed to Gaussian elimination. For sparse data sets and complex models this estimate may not be accurate or feasible for that matter as n-df approaches zero.

For h latent dimensions in PLS modelling the choice of df has most commonly been (Martens [1989]):

$$df = h + 1 \tag{2-146}$$

This approach is quoted from the work done by Van der Voet [1999] as being 'naïve.' Van der Voet [1999] subsequently went about finding an expression for calculating the pseudo degrees of freedom by first deriving some useful relationships under linear modelling conditions. He makes use of the definition of leverage, where the leverages are the diagonal elements of the following expression:

$$\mathbf{O} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{2-147}$$

The mean leverage is denoted by:

$$\bar{o} = \frac{m}{n} \tag{2-148}$$

Van der Voet [1999] then derived an expression whereby the mean leverage can be approximated by:

$$\bar{o} \approx 1 - \sqrt{\frac{MSEP_{rs}}{MSECV}} \tag{2-149}$$

The value for $MSEP_{rs}$ is defined as the mean-square error of re-substitution per output variable. This is simply equation (2-62) applied to the training data set of n independent observed measurements. It is recommended that MSECV be calculated from a leave-one-out cross-validation on all n data points (Van der Voet [1999]).

For linear models df = m and by making use of equation (2-148) the pseudo-degrees of freedom per output variable, y_j are therefore defined as:

$$\text{pdf} = n\left[1 - \left(\sqrt{\frac{\text{MSEP}_{rs}}{\text{MSECV}}}\right)\right] \qquad (2\text{-}150)$$

This generalisation is intended for all models that assume identically and normally distributed errors as pointed out in section 2.2 for general regression.

2.15 Confidence Intervals

Any model of a physical system has error associated with its predictions due to the uncontrollable or unobservable influences on the system's outputs. For any regression technique a measure for assessing the reliability of the prediction is required. A popular measure of this reliability is obtained by calculating prediction intervals. These intervals consist of the upper and lower confidence limits of the predicted values. The larger the magnitude of the intervals the less precise the prediction becomes.

Several methods for calculating confidence intervals for linear and non-linear exist. The mathematically more precise methods include the likelihood, the lack-of-fit, jack-knife and bootstrap approaches (Donaldson [1987]). Chryssolouris [1996] developed an approach for calculating intervals for predictions from multi-perceptron feed-forward neural networks. The approach is based on the Taylor series approximation and the Jacobian matrix of the functional mapping between the in- and output variables. This approach is less accurate, but it has shown to give acceptable results with minimal computational effort. Baffi [2002] used this approach and extended it to the linear PLS and the non-linear PLS algorithms. In this text all confidence intervals are calculated using this method pursued by Chryssolouris [1996] and Baffi [2002] and a brief description of the mathematical procedure follows below.

For a general mapping function, *f* between a set of input variables in **X** and an output variable in **y**, the prediction output variable is written as:

$$\bar{y} = f(\mathbf{X},\theta) \qquad (2\text{-}151)$$

The Jacobian matrix, **J** is defined, as in section 2.10.1, as the collection of partial derivatives:

$$j_{ir} = \frac{\partial f(\mathbf{x}_i,\theta)}{\partial \theta_r} \qquad (2\text{-}152)$$

The matrix, **J**, calculated using the training data, therefore consists of n rows and a number of columns equal to the total number of model parameters, M. The confidence interval is now calculated, given a new input sample, $\mathbf{x}^*$ corresponding to output y^*, as follows:

$$\mathrm{PI}(\hat{y}^*,\alpha) = \hat{y}^* \pm t_{n\text{-}df,1-\alpha/2}\sqrt{\mathrm{MSE}\left(1 + \mathbf{j}^{*T}(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{j}^*\right) - s_P} \qquad (2\text{-}153)$$

Here $t_{n\text{-}df,1-\alpha/2}$ is the critical value for the student's t-distribution at the α-significance level and n-df degrees of freedom. Chryssolouris [1996] included the term, s_P as an estimate of variance of the measurement noise, which is the difference between the measured observation and it's true value. For noiseless measurements this term is omitted. The vector, $\mathbf{j}^*$ is calculated using equation (2-152) for input sample point, $\mathbf{x}^*$. The model degrees of freedom, df can be replaced by the pseudo degrees of freedom, pdf, as discussed in section 2.14.

For output variable, y_j the functional mapping of a feed-forward neural network with a single hidden layer, using the hyperbolic tan-sigmoidal transfer function in the hidden layer and a simple linear weighting function in the output layer, can be written as:

$$\hat{y}_j = \sum_{r=1}^{H} \omega_{jr} \mathit{tanh}\left(\sum_{k=1}^{m} \omega_{rk}x_k + \beta_r\right) + \beta_j \qquad (2\text{-}154)$$

For each data point and output variable, y_j the partial derivatives with respect to the hidden layer weights can be calculated as follows:

$$\frac{\partial \hat{y}_j}{\partial \omega_{rk}} = \omega_{jr}\left[1 + tanh^2\left(\sum_{k=1}^{m} \omega_{rk}x_k + \beta_r\right)\right]x_k \qquad (2\text{-}155)$$

The hidden layer weights will therefore contribute m×H columns (number of parameters) to **J**. The partial derivatives with respect to hidden layer bias parameters will contribute H columns to **J** as follows:

$$\frac{\partial \hat{y}_j}{\partial \beta_r} = \omega_{jr}\left[1 + tanh^2\left(\sum_{k=1}^{m} \omega_{rk}x_k + \beta_r\right)\right] \qquad (2\text{-}156)$$

The output layer weights and bias parameters will contribute H columns and one column to **J**, respectively:

$$\frac{\partial \hat{y}_j}{\partial \omega_{jr}} = tanh\left(\sum_{k=1}^{m} \omega_{rk}x_k + \beta_r\right) \qquad (2\text{-}157)$$

$$\frac{\partial \hat{y}_j}{\partial \beta_j} = 1 \qquad (2\text{-}158)$$

For PLS models Baffi [2002] shows how the Jacobian matrix can be expanded into three sets of partial derivatives:

$$\mathbf{J} = [\mathbf{J}_P\ \mathbf{J}_\theta\ \mathbf{J}_Q] \qquad (2\text{-}159)$$

Here each set can be decomposed into simpler subsets of partial derivatives computed for each latent dimension and each data sample point of the PLS model:

$$\mathbf{J}_P\text{: } \frac{\partial \hat{y}_j}{\partial p_{ka}} = \frac{\partial \hat{y}_j}{\partial u_a}\ \frac{\partial u_a}{\partial t_a}\ \frac{\partial t_a}{\partial p_{ka}} \qquad (2\text{-}160)$$

$$\mathbf{J}_\theta\text{: } \frac{\partial \hat{y}_j}{\partial \theta_a} = \left[\frac{\partial \hat{y}_j}{\partial u_a}\ \frac{\partial u_a}{\partial \theta_a}\right] \qquad (2\text{-}161)$$

$$\mathbf{J}_Q\text{: } \frac{\partial \hat{y}_j}{\partial q_{aj}} \qquad (2\text{-}162)$$

The sets, $\mathbf{J}_P$ and $\mathbf{J}_Q$ thus comprise the derivatives relative to the input and output loadings of the outer PLS model. The partial derivatives of set, $\mathbf{J}_\theta$ are written with θ in vector notation, as these parameters have varying architectures depending on the PLS inner model used per latent dimension. The matrix, $\mathbf{J}_P$ comprises a number of columns equal to h×m. The matrix, $\mathbf{J}_\theta$ comprises the summed M_a columns for each latent dimension where a = 1,2,…,h. For sigmoidal neural networks and radial-basis functions (see section 2.13.2), M = 3H + 1 and for linear PLS models, M = 1. Finally, $\mathbf{J}_Q$ comprises h columns. The partial derivatives for a single data point and the j^th output variable using a linear PLS model are therefore calculated as follows:

$$\mathbf{J}_P\text{:}\ \frac{\partial \hat{y}_j}{\partial p_{ka}} = q_{aj}\, b_a\, x_k \qquad (2\text{-}163)$$

$$\mathbf{J}_\theta\text{:}\ \frac{\partial \hat{y}_j}{\partial b_a} = q_{aj}\, t_a \qquad (2\text{-}164)$$

$$\mathbf{J}_Q\text{:}\ \frac{\partial \hat{y}_j}{\partial q_{aj}} = u_a \qquad (2\text{-}165)$$

The exploratory data analysis techniques presented in this chapter can now be applied in the following chapters. In Chapter 3 the data is first analysed for dimensionality, degrees of freedom, correlations, outliers and classifications. PCA and classed-based PCA become important tools for dealing with these analyses. In the chapters following Chapter 3 the modelling techniques of feed-forward multi-layer perceptron neural networks, RBF networks and PLS are applied and their results are discussed. In the modelling applications important statistical concepts such as degrees of freedom, model complexity and model validation are addressed. These are common to all the models applied and are important in order to maintain statistical integrity regarding the performances of the candidate models.

Chapter 3

Data Analysis and Outlier Detection

In this chapter the origins and structures of the data used are initially described and analysed. This is followed by an investigation into the correlations and similarities within the data structure. These two investigations lead up to the detection of potential outliers, investigating the expected clustering of the data and developing a *modus operandi* for building 'black-box' models.

3.1 Defining the Origin and Structure of the Data

The data, which is divided into two parts, has been adopted from the work done by Roodt [1998]. The independent part with no measurement noise contains the rocket design parameters and chemistry known as the rocket design features. The IR emission spectra data is the dependent part containing the measurement noise.

3.1.1 The Rocket Motor Design Feature Data

In Chapter 1 the data structure to be modelled was introduced. Two different classes of solid propellants were used in the preparation of the 18 different rockets fired for data sample collection. The double-base (DB) propellants are the first production propellants to be implemented. The development of polymers as binders made it possible for the production of composite (C) propellants. Emission spectra were measured using 6 different DB rocket motors and 12 different C rocket motors.

The DB propellants form a homogeneous propellant grain consisting of nitrocellulose dissolved in nitroglycerine with small amounts of additives sometimes added. Both nitro-ingredients contain carbon and hydrogen as fuels and oxygen as oxidiser. This means they both act as combined fuels and oxidisers.

The C propellants form a heterogeneous grain where the oxidiser and powdered fuel are held together in a matrix of synthetic rubber or some other plastic material. The conventional oxidiser and fuel are ammonium perchlorate (NH_4ClO_4) and aluminium powder respectively. The binder is usually a natural rubber such as polybutadiene. Solid oxidisers such as ammonium nitrate (NH_4NO_3) and potassium perchlorate ($KClO_4$) can also be used.

In general, elements in groups 1 to 4 on the periodic table, such as C, H, Li, etc., act as fuels. The elements in groups 6 and 7, such as Cl, O, F, S, etc., act as oxidisers. A more detailed discussion is given in Roodt [1998].

There are a large variety of different chemical ingredients and propellant formulations that have been synthesised and tested in experimental rocket motors. A typical propellant may have 5 to 15 different ingredients. These have a significant influence on the propellant characteristics and are prepared specifically for each rocket motor. The rocket must be designed in such a way that the desired burn rate, physical and thermal properties, IR emission spectra and overall performance characteristics are obtained. The rocket propellant must also be safe to manufacture and must have a reasonable shelf life.

One of the primary challenges when it comes to using 'black-box' modelling techniques is to find appropriate input variables for building the model to predict the IR emission spectra. It is mentioned in Roodt [1998] that past studies have shown that the spectra are best modelled using the thermodynamic properties, propellant chemical composition, the conical rocket nozzle dimensions and the gas mass flow rate through the nozzle. It was found that the thermodynamic properties are all governed by the rocket chamber temperature (T_C) and pressure (P_C). The expansion ratio (E_C) through the nozzle and the rocket throat diameter (D_T) were subsequently chosen for the nozzle dimensions (see Fig. 3-1). The gas flow rate was left out. Table 3-1 shows how the rocket motor design parameters have been scaled and in what dimensions they are presented by Roodt [1998]. The same values have been adopted here, as it was found that the scaling has no effect on the model predictions if the data processing methods as described in section 2.4 are applied.

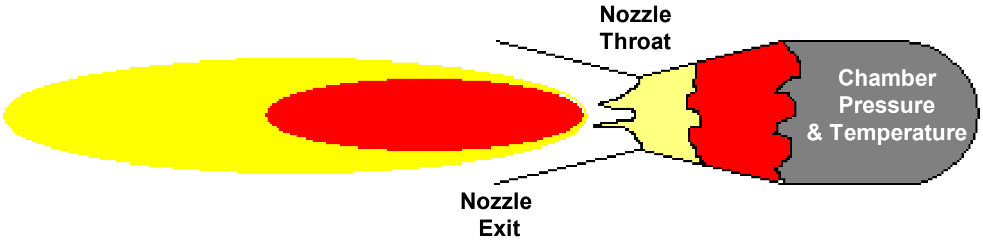

Fig. 3-1 A schematic representation of the rocket chamber and the nozzle dimensions.

Table 3-1 The physical rocket motor design parameters as they are used for building the models.

Rocket Motor Design Parameter	Symbol	Unit
Chamber Pressure	P_C	[MPa]
Chamber Temperature	T_C	[K]/1000
Conic Expansion Ratio	E_C	[-]/10
Throat Diameter	D_T	[mm]

Initially, the idea was to use a thermo-chemical computer program to compute the equilibrium chemical composition after combustion by making use of Gibbs free energy minimisation. These compositions would then be used as part of the input vector. However, the model predictions did not correspond well with the IR measurements made. The computer program has a composition cut-off value below which it discards all chemical species when doing calculations. It is difficult to determine the optimum molar fraction below which the program should not consider the corresponding molecular species. Different rocket motors will produce different combustion products leading to a data set containing many zeros. The next step was to use the propellant molecular composition as part of the input vector. This ensured that there would be no loss in chemical information. However, once again it was found that there are too many zeros in the data due to the fact that different propellants consist of different ingredients.

A large contribution to the insight presented by Roodt [1998] included ‘collapsing’ the propellant molecular chemical compositions into their elemental compositions. This was done by summation of a selected elemental composition over all the molecular species. These calculations were done on a 100kg basis of propellant. For example, for the chemical compound ‘NC25’ ($C_6H_{7.579}O_{9.833}N_{2.416}$) used in the rocket motor propellant DB1 the carbon contribution can be calculated from (Roodt [1998]):

$$C_{kmol} = \frac{\text{NC25 \%} \times n_C}{n_C \times M_r(C) + n_H \times M_r(H) + n_O \times M_r(O) + n_N \times M_r(N)} \qquad (3\text{-}1)$$

$$= \frac{42.8 \times 6}{(6 \times 12.011) + (7.579 \times 1.008) + (9.833 \times 15.999) + (2.416 \times 14.007)}$$

$$= \underline{0.948 \text{ kmol}}$$

The advantages here are that there is no loss in chemical information and because the elements are common to most molecular species the data set contains relatively few zeros. This denser data set was found to be better suited for building a neural network model and the predictions compared well with IR measurements.

The final rocket feature set is shown in Table 3-2 for both DB and C rocket motors. The first 14 variables are the elemental compositions and the last 4 are the rocket motor design parameters. This data also contains a fair amount of zeros. This is because F only has a single entry and S and Fe have two and three entries respectively. These form three dimensions, which may not contribute significantly to the overall models that will be built. It is, however, interesting to see if these elements have a significant effect on the model predictions, as the maximum information possible needs to be used to find the required stealth characteristics.

Table 3-2 The rocket design features consisting of 18 variables (14 variables form the rocket fuel elemental chemistry and the last 4 variables are the physical motor design parameters).

	C	H	O	N	Al	K	F	Cu	Pb	S	Cl	Si	Ti	Fe	T_C	P_C	E_C	D_T
DB1	1.869	2.658	3.362	1.335	0	0	0	0.007	0.008	0	0	0	0.008	0	2.92	4.90	6.2	13.0
DB2	2.008	2.985	3.321	1.277	0	0	0	0.011	0.005	0	0	0	0	0	2.53	4.40	8.7	11.0
DB3	2.440	3.364	3.357	0.839	0	0	0	0	0.009	0	0	0	0	0	1.82	3.10	14.5	8.5
DB4	1.964	2.863	3.347	1.184	0	0.045	0	0.01	0.005	0	0	0	0	0	2.56	4.10	8.7	11.0
DB5	1.733	2.387	3.270	1.124	0.015	0.121	0.062	0.007	0.008	0	0	0	0.008	0	2.86	5.00	10.5	10.0
DB6	1.733	2.655	3.269	1.392	0	0.059	0	0.007	0.008	0	0	0	0.008	0	2.78	7.40	10.5	10.0
C1	0.997	3.908	2.407	0.604	0.593	0	0	0	0	0	0.596	0	0	0	3.27	2.70	6.2	13.0
C2	0.997	3.839	2.377	0.587	0.593	0	0	0	0	0	0.579	0	0	0.025	3.26	2.80	4.7	15.0
C3	0.997	3.839	2.385	0.587	0.593	0.023	0	0	0	0.011	0.579	0	0	0	3.26	2.50	6.2	13.0
C4	0.938	4.349	2.960	0.738	0	0	0	0	0	0	0.732	0.025	0	0	2.93	6.20	6.2	13.0
C5	0.938	4.281	2.938	0.721	0	0.023	0	0	0	0.011	0.715	0.025	0	0	2.87	2.20	6.2	13.0
C6	0.893	4.141	2.812	0.711	0.148	0	0	0	0	0	0.698	0	0	0	3.07	3.40	4.7	15.0
C7	0.924	4.334	2.966	0.740	0	0	0	0	0	0	0.734	0.025	0	0	2.90	3.00	7.3	12.0
C8	0.997	3.908	2.407	0.604	0.593	0	0	0	0	0	0.596	0	0	0	3.30	4.00	7.3	12.0
C9	0.894	4.203	2.855	0.708	0.148	0	0	0	0	0	0.702	0	0	0.009	2.98	1.06	1.7	25.0
C10	1.115	4.399	2.769	0.695	0.148	0	0	0	0	0	0.681	0	0	0	2.81	1.29	4.5	15.0
C11	1.047	4.382	2.796	0.701	0.148	0	0	0	0	0	0.689	0	0	0	2.86	1.00	4.5	15.0
C12	0.924	4.327	2.963	0.738	0	0	0	0	0	0	0.732	0.025	0	0.003	2.90	3.00	6.2	13.0

3.1.2 The IR Emission Spectra Data

The IR emission data consists of numerous repeat radiometer absorbance observations made for each of the eighteen different types of rockets considered. These repeat observations or data sample points are recorded at 146 wavelengths in the middle IR band (2 to 5.5 μm wavelength). Furthermore, it is assumed at this stage that the repeat measurements are true replicates as described in section 2.5.2. The number of measurements made per rocket range from 4 to 44 (see Table 3-3). The recorded IR emission spectra are shown for all 18 rockets in Fig. 3-2 to Fig. 3-4. The tabulated values of the data are given in Roodt [1998]. The exact incremented wavelength values are not given and therefore the wavelength numbers are numbered using unit integers from 1 to 146.

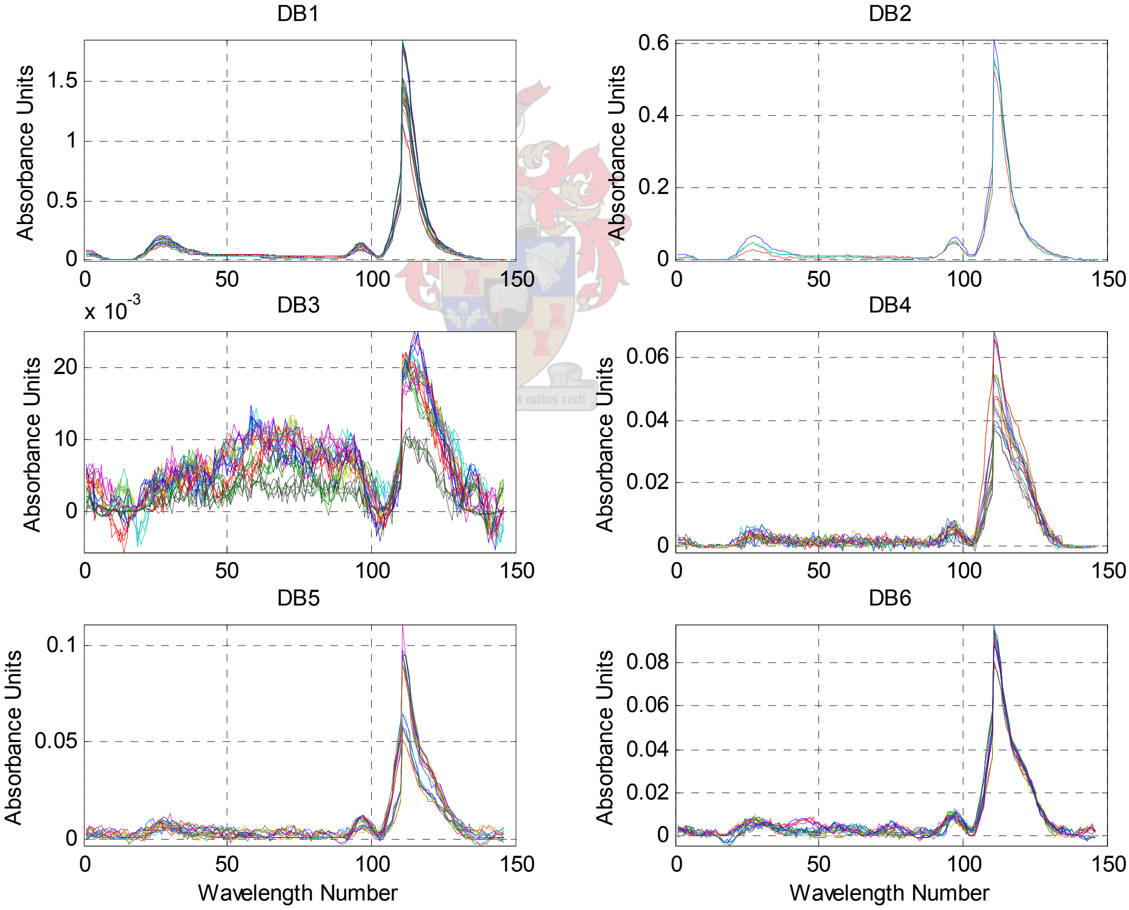

Fig. 3-2 The measured IR emission spectra for the six different rockets with double base solid propellants at 146 wavelengths from 2 to 5.5 μm.

Table 3-3 The numbers of IR emission measurements for each rocket motor.

DB1	DB2	DB3	DB4	DB5	DB6	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
39	4	31	20	20	22	15	24	24	17	18	26	44	15	25	14	23	22

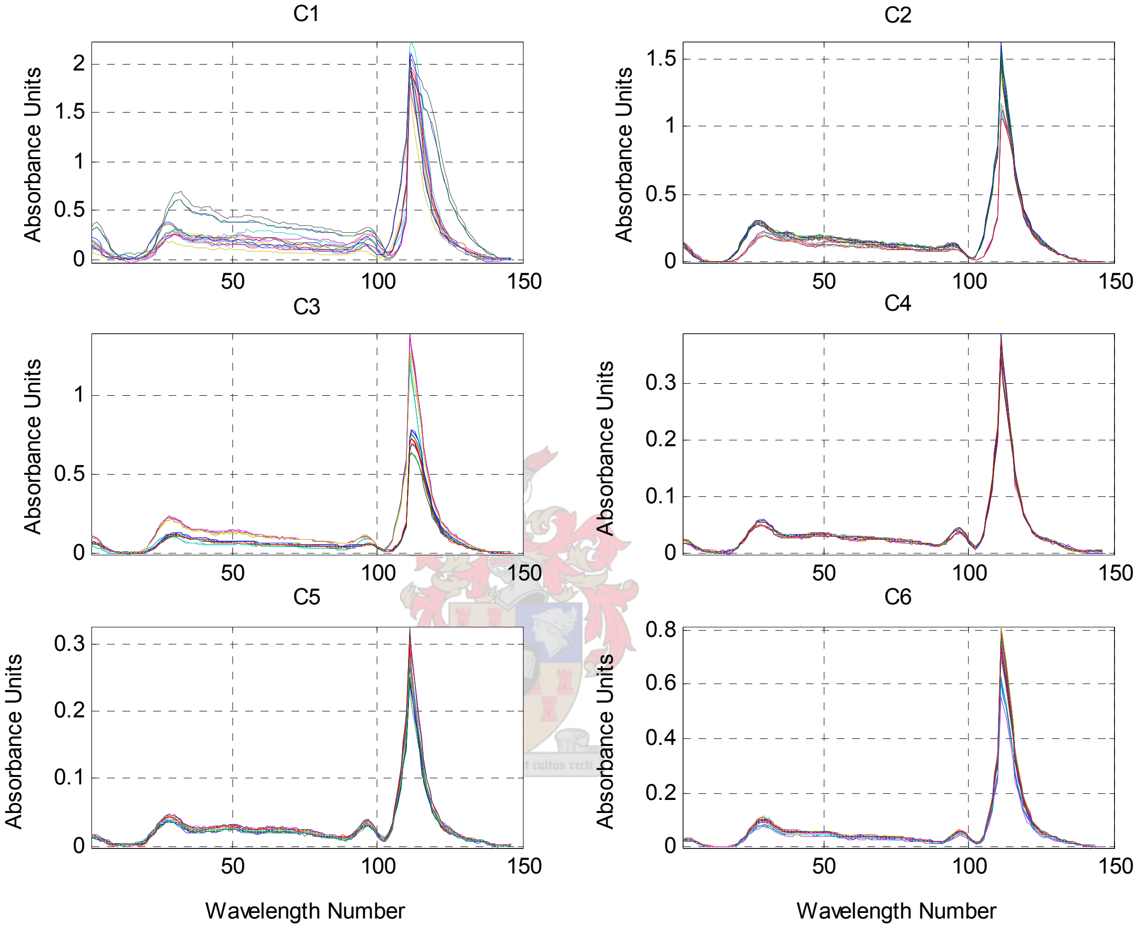

Fig. 3-3 The measured IR emission spectra for the first six different rockets with composite solid propellants at 146 wavelengths from 2 to 5.5 µm.

The emission data was recorded by Roodt [1998] over a six-year period between 1992 and 1998. The spectra were recorded using a spectral radiometer at varying distances from the rockets that were fired. These distances were 500m, 350m, 250m and 200m. For this reason the data had to be pre-processed by using a scaling factor in order to compensate for the varying absorbance pathlengths.

Due to the varying atmospheric conditions such as humidity, temperature, atmospheric pressure, line of sight range and aerosol concentrations some form of atmospheric correction was required. This was achieved by making use of Bouguer's law as described in Roodt [1998]. The two data pre-treatments mentioned above were thus used to compensate, as far as possible, for environmental influences as illustrated in Fig. 1-1.

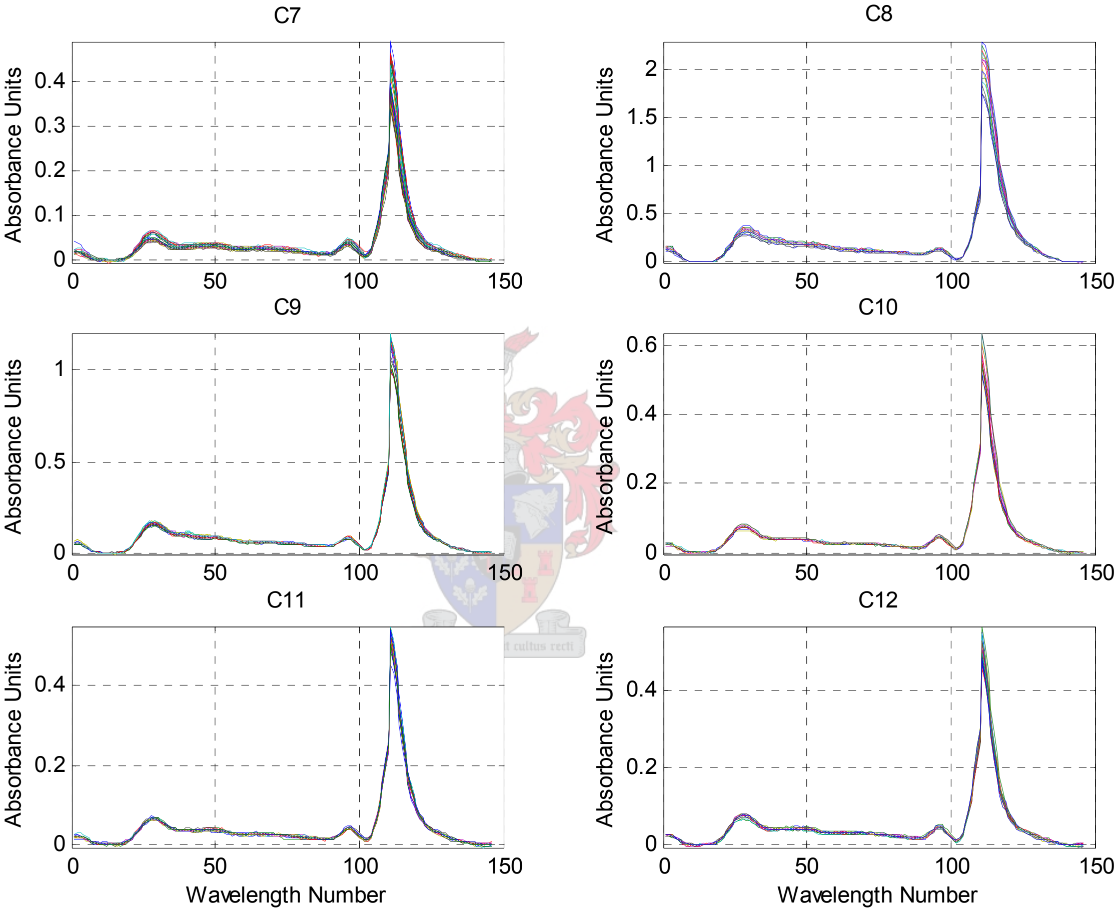

Fig. 3-4 The measured IR emission spectra for the last six different rockets with composite solid propellants at 146 wavelengths from 2 to 5.5 μm.

3.1.3 Nature of the Data Structure and the Modelling Thereof

At this stage it is evident that the data set available is sparse and the rocket features and the emission spectra are multivariate. There are a total of 403 measurements made over all 18 rocket motors tested. However, there are only 18 predictions to be made. Hence there are h = 1,2,…,18 rocket features and mean IR emission spectra for which there are n_h emission spectra replicates for each rocket motor (see section 2.5.2). From statistical theory presented in section 2.2 each prediction made by a regression model is the expected value over a normal probability distribution of possible predictions for an independent data point. If a linear regression model were to be used on this data, care would have to be taken that there are more independent data points than there are model parameters. In the case where the simplest possible linear model is used, where the input variables are in linear form, there would be 19 required model parameters (including bias). Clearly, the data set is not large enough for the simplest linear regression model.

The fact that there is a varying number of repeat spectral measurements for each rocket motor does not imply that predictability is biased towards those rocket motors with the most repeat measurements. This is due to the geometry of projecting the output space onto the input space (see section 2.2.1 and Bates [1988]) and the fact that the cost function is minimised over the entire data set. The model predictabilities for all independent data points are influenced by the residual variance, ε, which is also the variance of the normal distributions around the expected values, $f(\mathbf{x}_i,\theta)$. This variance is the same for all data points (see section 2.2) and is calculated using the entire sample set. The model results in section 4.2 support these arguments.

The repeat measurements form the pure error component of the data set, which becomes an estimate of the population residual variance, ε. The more data points there are to form the pure error component the better the population estimate. Certain repeat measurements could be omitted for the purpose of obtaining more accurate predictions. This can be achieved by a reduction in residual variance, which would reduce the confidence interval around a prediction. This can be misleading to believe a model is more accurate than it really is. However, if it is determined that some of the data points are outliers it is essential that the data sample points be removed (see section 3.3).

The emission spectra to be predicted consist of $n = 403$ data points in total of which there are 385 replicates. According to statistical theory (Brereton [1990] and Draper [1981]) it is recommended to have 3 to 5 lack-of-fit (residual) degrees of freedom (n_L) left in order to build a linear model that is built from an over-determined system. The appropriate number of model inputs can thus be calculated from $m = n – n_R – n_L - 1$. It is clear that the number of linear model inputs should thus be 14 or smaller (see Table 3-4).

Table 3-4 The degrees of freedom involved in building a linear model for the prediction of the IR emission spectra.

Total number of sample measurements	n	403
Bias	1	1
Total number of replicates	n_R	385
Lack-of-fit degrees of freedom	n_L	3
Number of linear model inputs	m	14

From the previous discussion it is shown that the simplest multivariate linear regression model is not an option for building a model here. For neural networks the pseudo-dimension is limited to m+1 per node where m is the number of inputs to the node. This means that at least m+1 independent data points are needed in order to 'shatter' the data set for a radial-basis function (Schmitt [2001]) or a sigmoidal node (Sontag [1998]). The pseudo dimension is the Vapnik-Chervonenkis (VC) dimension (Haykin [1999]) for continuous function approximation by neural networks. Unless a significant dimensional reduction could be obtained it would therefore also seem futile at this stage to use a feed-forward neural network to build an adequate model on the data.

Lawrence [1997] showed that for data consisting of 200 independent data sample points the optimum number of hidden nodes in the single hidden layer was between 30 and 40 for a network with 20 input variables and a single output variable. At 30 nodes in the hidden layer there are therefore 661 parameters in the network as a whole, which is significantly higher than the 200 data points available. A total of 5×10^5 epochs and no regularisation such as early stopping during training were used and yet the oversized neural network managed to provide generalisation.

Lawrence [1997] further showed that the rules defined by the degrees of freedom discussed above are unreliable for neural networks, as there are other factors that came into play such as the quality of the solution found, the amount of noise, any bias in the algorithm and the nature of the function being approximated. The quality of the solution refers to the fact that no global optimum is obtained. It is further stated that if the global optimum were to be found the rules of the degrees of freedom would be relevant. Lawrence [1997] did however state that it is desirable to find solutions with the smallest number of parameters. He also emphasised the importance of the number of epochs used during training, as the larger the number of epochs used the closer the algorithm gets to a global minimum. From the discussion above he also claimed the VC dimension to be somewhat conservative in estimating the lower bound for the number of data points.

A further implication from the discussions above is therefore that by over-training a feed-forward neural network, as far as the number of epochs is concerned, the model complexities can be controlled almost entirely by the number of model parameters (network weights). This means it would be possible to obtain similar predictabilities from two different network architectures built using the same data set. However the one network may have relatively few hidden nodes with over-trained epochs and the other one may have more hidden nodes but a regularisation technique is used to avoid using too many epochs during training. Lawrence [1997] shows how the larger neural network architectures contain more weights that do not significantly contribute towards the final approximation.

There appears to be enough reason to proceed with the building of models using neural networks. The neural network built by Roodt [1998] may seem extremely oversized with 146 nodes used in the hidden layer. This amounts to a total of 24236 parameters for the 16 independent data sample points used to build the network. However, he only used approximately 200 training epochs in an attempt to avoid over-training the network.

The limit on the data available is determined by economic implications. The firing of each rocket is extremely expensive and the use of an efficient design of experiment is not possible when at the time the work by Roodt [1998] was published there were only 12 rocket motor designs in common use. As mentioned previously safety is an important criterion when applying a new rocket motor design and not just any design will be acceptable. The large number of repeat measurements may not have been necessary but the original point of departure was to use a more fundamental approach combining CFD and quantum-mechanical principles.

The fact that the data is so sparse makes it very difficult to validate any model, as the loss of just one independent data point has a significant effect on the degrees of freedom available. In the case of linear modelling, internal validation could be used but as discussed previously linear modelling is not possible here. For non-linear models the use of validation data is essential in order to measure up the generalisation ability of the model. The omission of just one or two rocket motors for testing is the only option left. However, a neural network could be retrained until both the training set and the omitted validation data are approximated well. For this reason ‘leave-out’ cross-validation as described in section 2.6.3 is the best option for obtaining a more reliable method of validating the ability of the model to approximate the functionality between the rocket features and the IR emission spectra.

All the repeat measurements can be included for the forward mapping from the independent rocket motor features to the measured IR emission spectra. The reverse problem of predicting the rocket features from the spectral measurements forms part of an extensively researched field in chemometrics. Here Beer’s law and the data dimensionality reduction methods using PCA and PLS can be used as discussed in section 2.13. However, the means of the emission spectra have to be used as independent input data. This reduces the data set to 18 input and output measurement samples. This is necessary in order to keep within statistical modelling theory as discussed in section 2.2.

3.2 Exploring Data Proximities and Correlations

In the previous section it was shown that the available data is far from ideal for the purpose of ‘black-box’ modelling, however there is no justification to completely disregard it at this stage. It may be possible to deal with the apparent lack of information in the data by examining it more closely as far as correlations and dimensionality are concerned. Any information obtained from this scrutiny would aid in obtaining qualitative information required to make decisions on the ‘black-box’ model type and structure.

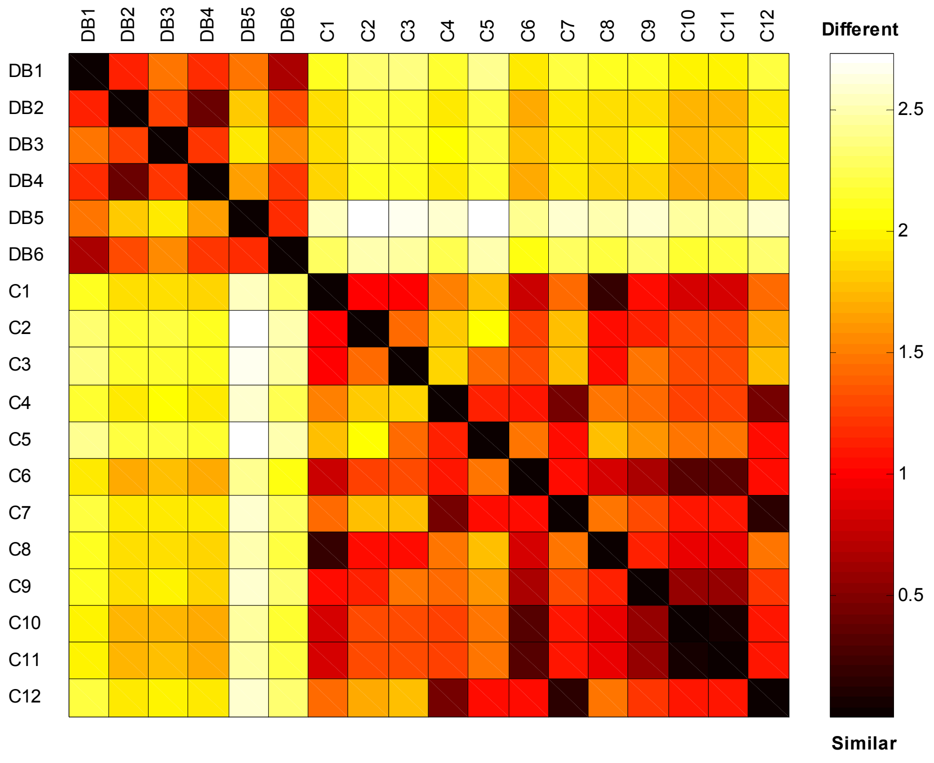

Fig. 3-5 A map of Euclidean distances between the data sample points of the rocket design feature set.

A look at the Euclidean distances between data sample points can give an idea as to whether there are distinguishable classes within a data set. The

Euclidean distances between the normalised rocket features mapped in Fig. 3-5 show that a great deal of similarity is evident and that there are two distinguishable classes. The existence of two classes within the data was already known but this result shows that the distinction is picked up in the data and therefore provides greater confidence in the choice of the rocket feature variables chosen.

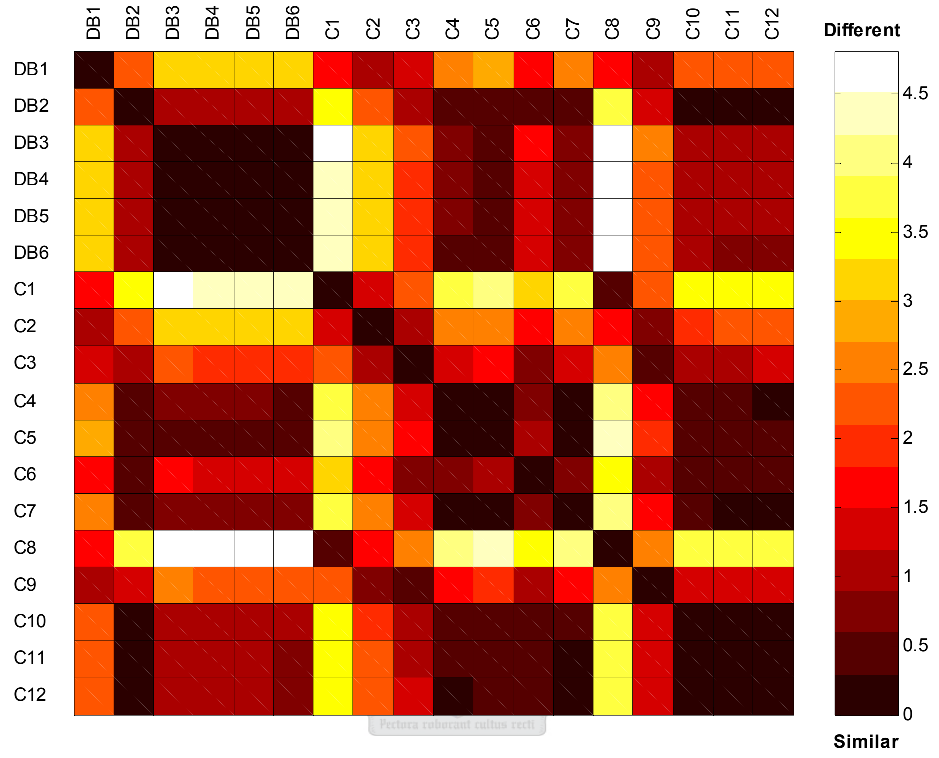

Fig. 3-6 A map of Euclidean distances between the data sample points of the eighteen mean IR emission spectra.

The Euclidean distances map for the mean emission spectra in Fig. 3-6 contains no pattern indicating possible class memberships. This shows that the problem of modelling the data may be complex. The information that can be extracted from the map is that there are strong similarities in irradiance spectra between rockets DB3, DB4, DB5, and DB6 and also between C10, C11 and C12. There also seems to be a crossover between classes where DB2 has similar irradiance spectra to C10, C11 and C12. This information becomes useful when evaluating the model results in later chapters.

A correlation map (see section 2.1.3) has been drawn up, initially, in order to see if there are any linear correlations between the rocket design feature variables. The variables have been normalised to values between 0 and 1 in order to compare all variable on the same scale. Fig. 3-7 shows that there are no prominent patterns, which is expected as the mixing of ingredients and the rocket motor design parameters for a particular rocket are not necessarily obtained by some underlying common process.

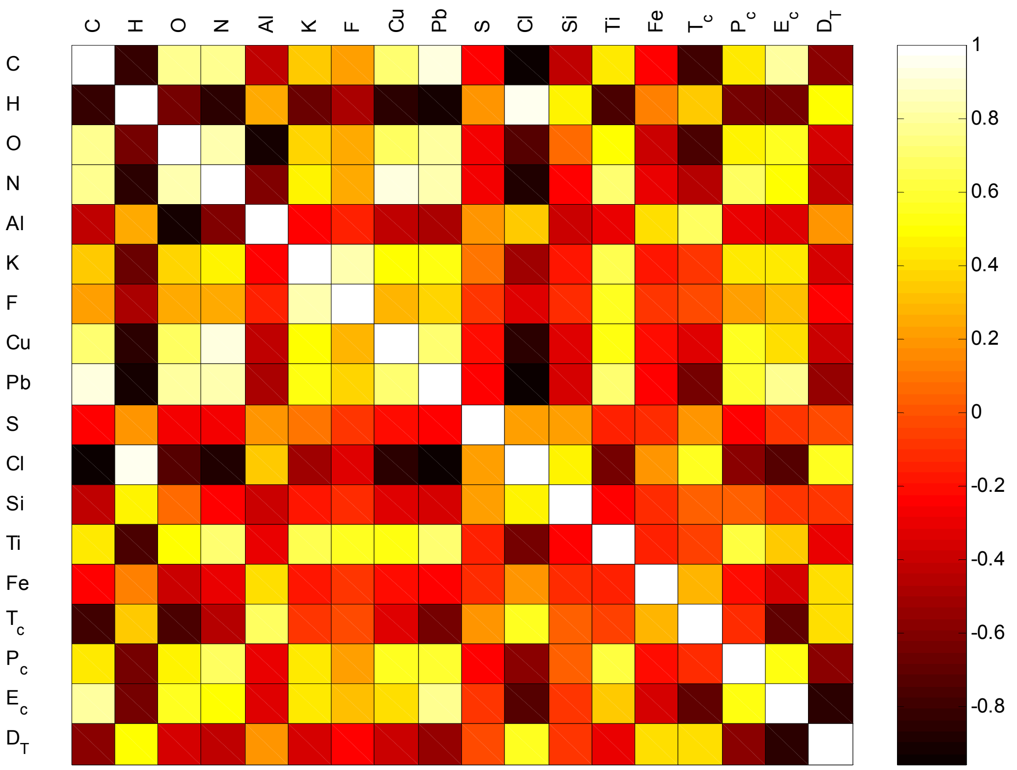

Fig. 3-7 A correlation map of the rocket motor design parameters and chemistry to investigate the possibility of redundant information being present.

The PLS modelling techniques make use of latent dimensions similar to PCA (see section 2.13) and therefore it can be expected that some redundant information, usually found in highly correlated data can be excluded for the final input-output mapping. For this reason it is useful to investigate if any dimensional reduction can be expected, if the rocket features are used as input data set. At this stage it seems like there is a fairly even spread of high and low correlations between the variables as seen in Fig. 3-7. A radical reduction in input space dimensionality is therefore not expected.

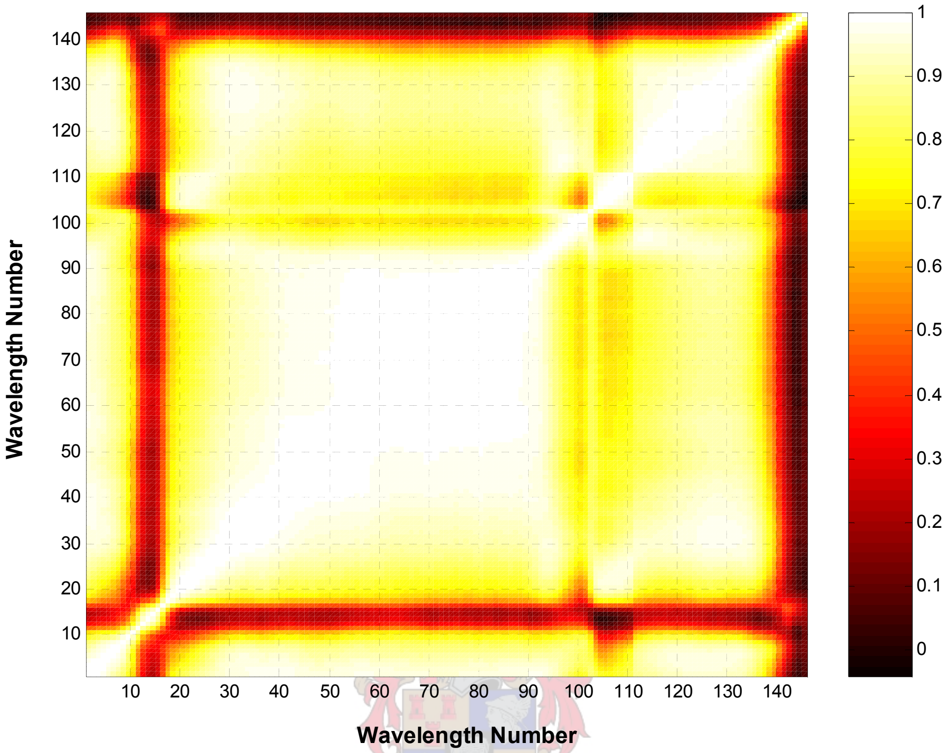

Fig. 3-8 A correlation map of the IR emission spectral absorbance values to investigate the possibility of redundant information being present.

Fig. 3-8 reveals a different picture to that of the rocket feature data. It is clear that there is a great deal of correlation in the regions spanning from wavelength numbers 1 to 10, 20 to approximately 95 and then from approximately 110 to 140. The regions of no or very little correlation appear to be those bearing the large CO_2 peaks between wavelength numbers 100 and 120 (at approximately 4.4 μm wavelength) and the smaller, prominent H_2O peaks between wavelength numbers 20 to 40 (at approximately 2.7 μm). The fringe to the far right of the spectrum also seems to bear very little correlation. The relevance of this region is not clear at this stage. Roodt [1998] mentions that the region from approximately wavelength number 20 to 95 is extremely important for pattern recognition and the identification of stealth characteristics. However, this region may be adequately represented by only a few latent dimensions or within other prominent latent dimensions due to its large degree of correlation between variables.

3.3 Outlier Detection

In the previous section it was shown that a class distinction between DB and C rocket motor designs could be observed by inspection of the rocket motor feature data set. However, this distinction is not so easily observed when analysing the emission spectra data. This may be a result of the high degree of correlations and therefore possibly redundant information within the spectral data. It is useful to be able to investigate if there really is redundant information and if it is possible to extract features of lower dimensionality from the spectral data.

PCA becomes a useful tool for analysing data if the first two to three principal components explain a large amount of the total variance within the data (see section 2.8). The principal scores can then easily be plotted and provide a visual means of investigating the data. They contain the relevant information about the distinguishing characteristics of the data, enhancing the ability to observe patterns within the data without being polluted by noise or redundant information.

In section 3.1 it was shown that there are only 4 measured sample points for the spectral data of DB2. Seventeen additional points have been generated randomly around the mean in such a way that the original mean and standard deviation have been maintained. This will clearly not have an effect on the regression results, as a regression model seeks the expected value of a data point within an assumed normal probability distribution (see section 2.2). The additional data has thus been generated in order to aid with the visualisation of the data during data analysis. The spectral data set now has a total of 420 data points.

The explained variances of a principal components analysis on the full spectral data set are shown in Table 3-5. Here the spectral data has been standardised (see section 2.4.2) prior to the analysis in order to centre the data and therefore obtain unit standard deviation for each variable. It is shown that the bulk of the explained variance (86.7%) is captured within the first two principal components. This is ideal for the visualisation of the data and supports the notion in the previous section that there may be redundant information in the data.

The scores from the first two principal components have been plotted against each other in Fig. 3-9. It is evident that the DB class rocket data is clustered towards one side of the plot. This is a promising result as this means that just two principal components are required to start visualising class distinctions. This is also positive for the building of models, as PCA is not a pattern recognition technique but merely a feature extractor. The use of such features can make the models more robust, especially because no prior knowledge of class membership is required to extract the features. The PLS models are related to PCA and are regarded as more parsimonious (as discussed in section 2.13.3).

Table 3-5 The variance explained by each of the first 12 principal components (PCA) out of 146 possible dimensions of the standardised spectral data.

principal component	explained variance	cumulative variance
1	80.55%	80.55%
2	6.16%	86.71%
3	4.10%	90.81%
4	3.25%	94.06%
5	2.32%	96.38%
6	1.34%	97.72%
7	0.50%	98.22%
8	0.34%	98.56%
9	0.23%	98.79%
10	0.17%	98.96%
11	0.12%	99.08%
12	0.10%	99.18%

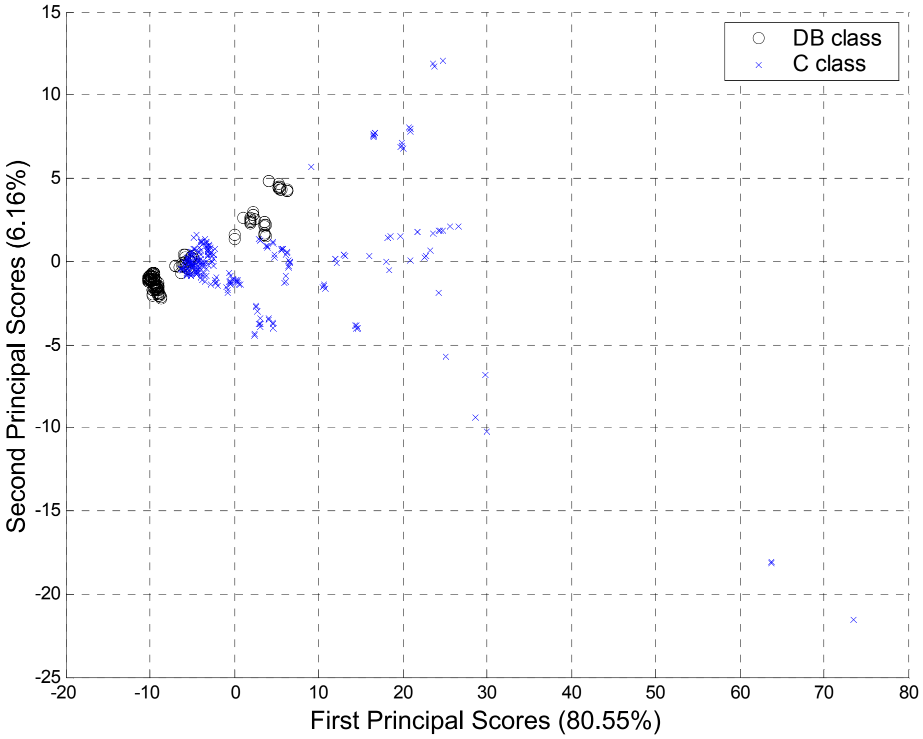

Fig. 3-9 A plot of the two primary principle component (PCA) scores from the standardised spectral data.

It is useful to make use of PLS model building and then to use the input scores to calculate leverage and residual values, which are used to infer potential outliers on some significant statistical α-level. However, neural networks will be compared to the PLS models, which makes it desirable to detect potential outliers prior to data modelling. For this reason the PCA scores will be used to investigate the presence of outliers here.

In Fig. 3-9 three points can be seen isolated at the bottom far-right corner of the plot. These three points have been marked and identified in Fig. 3-10 as being data points 160, 161 and 162. These points belong to data of rocket motor C1. Every bit of information is important in an already sparse data set and therefore potential outliers must be carefully scrutinised before leaving them out. The 3-dimensional plot in Fig. 3-11 provides supporting evidence that data points 160 to 162 are possible outliers by noting that their irradiance levels are very high in comparison to the rest of the data.

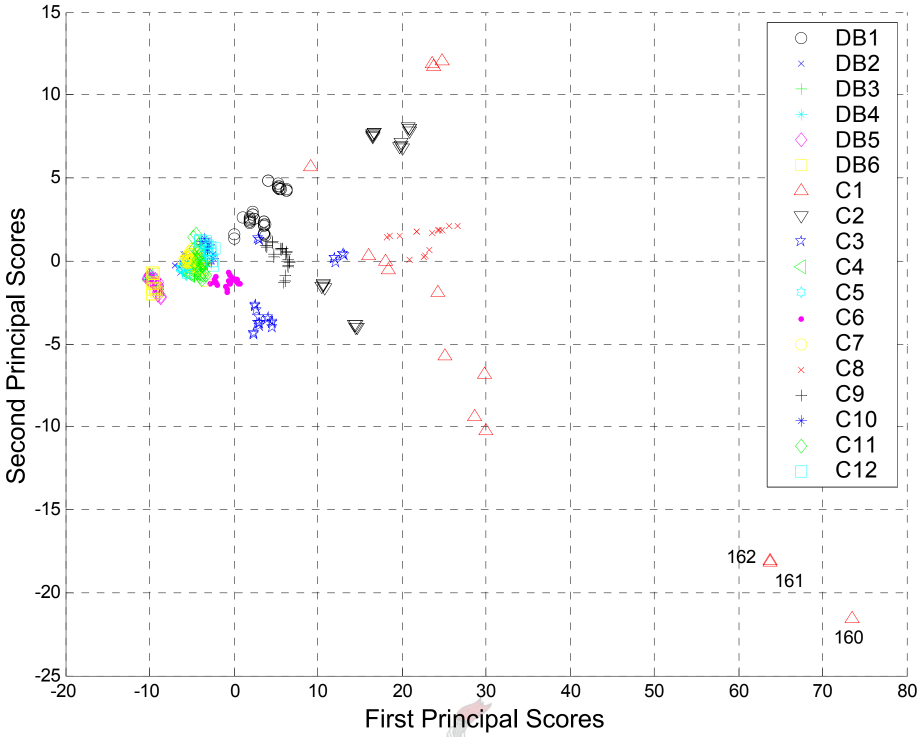

Fig. 3-10 The primary principal components plotted to identify any distinct groupings of data. Data points 160 to 162 are labelled to mark them as potential outliers.

Table 3-2 shows that the rocket chemistries of C1 and C8 are identical but their motor design parameters differ. Fig. 3-6 shows that the Euclidean distances of the mean spectral data are similar between these two rocket motors. For this reason it is useful to plot all the spectra of C1 and C8 together and investigate if there are possible irregularities. Fig. 3-12 shows that data points 7 to 9 of C1 corresponding to points 160 to 162 of the whole data set are much larger than the rest of the measured data and do not follow the same tendencies, especially at the trailing end of the CO_2 absorbance peak.

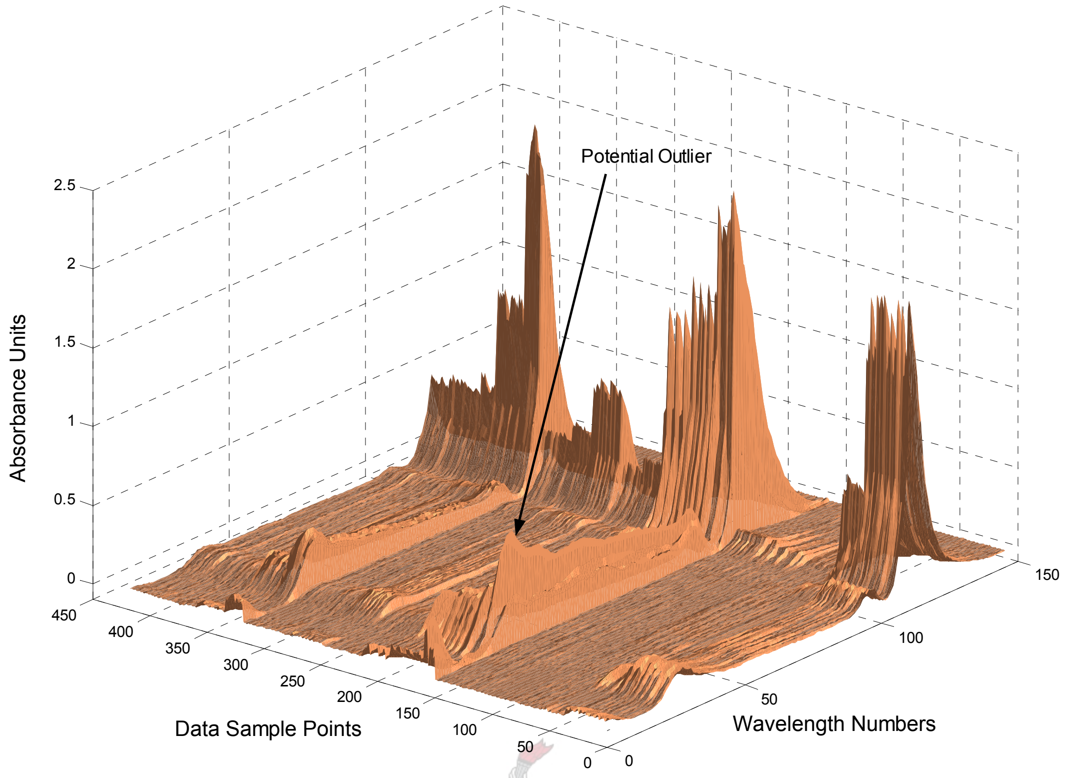

Fig. 3-11 A 3D plot of the spectral data arranged from DB1 to C12 over all 420 data sample points and all wavelength numbers.

At this stage a statistical test is in order to investigate any further justification to leave out the three data points under suspicion. This is done using Hotelling's T^2-statistic and the Q-statistic as described in section 2.8. The plot in Fig. 3-13 is obtained using the first two principal components to calculate the statistics. The figure shows that the sample data points 160 to 162 have extremely large leverage (T^2-statistic) on the outcome of the PCA model, although their discrepancy (Q-statistic) value is low. This shows that the low discrepancy could be as a result of the PCA model loadings being so largely influenced by these points that they are well compensated for within the model.

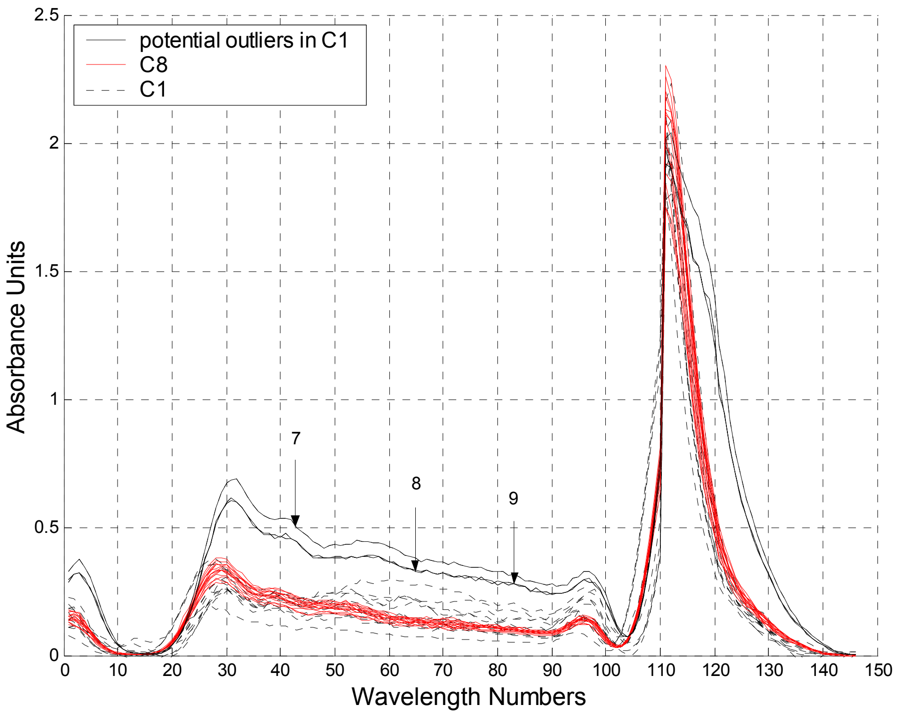

Fig. 3-12 The IR emission spectra of C1 and C8 plotted together. The potential outliers correspond to data points 7,8 and 9 of C1.

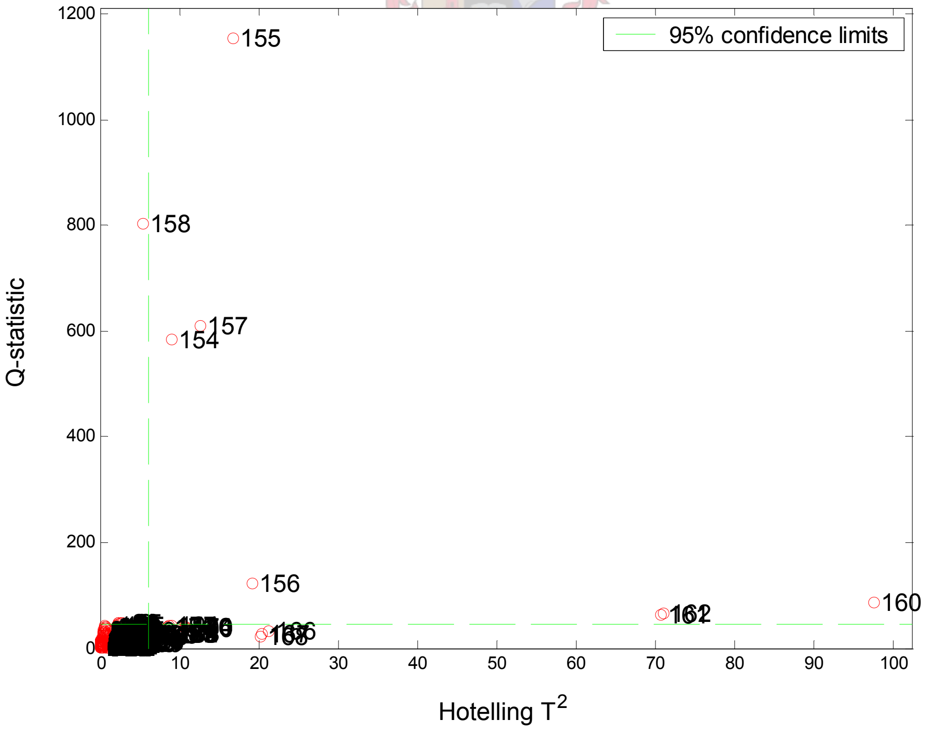

Fig. 3-13 The discrepancy and leverage of the IR emission spectra data points calculated using two principal components (86.7% variance).

Strictly speaking the sample data points 154 to 157 (data points 1 to 4 of C1) should also be regarded as outliers, as they lie outside the confidence limits (Fig. 3-13). Their leverages are not as extreme and thus have a lesser effect on the model parameters. From analysis of the spectra in Fig. 3-12 and because the statistics used here are based on normal distributions (which may not be the case here) it may be sensible to keep the data sample points outside the confidence limits for modelling purposes. The idea is also to keep as much information as possible for building the model in order to enhance robustness.

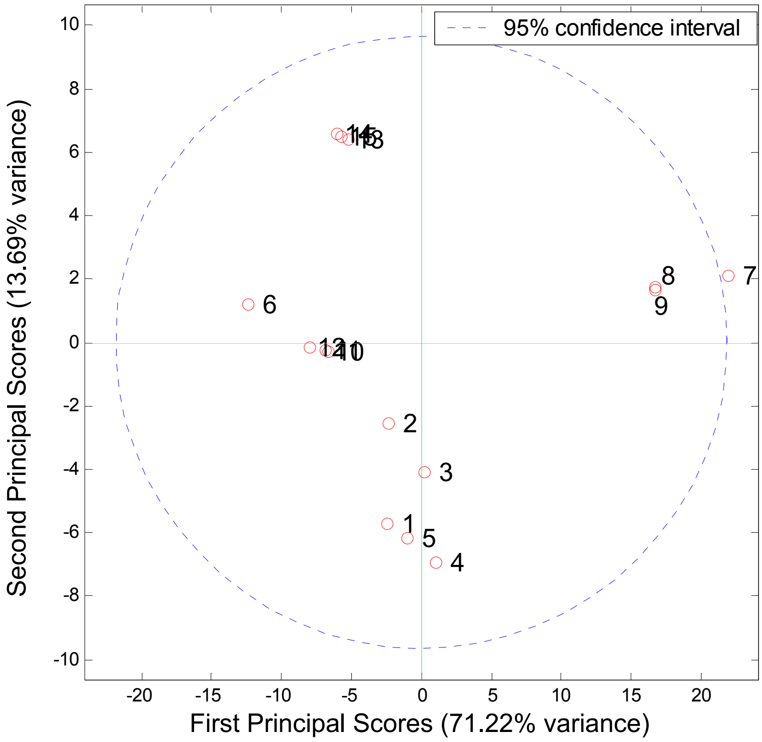

Fig. 3-14 The confidence interval calculated using the first two principal component scores of rocket motor C1.

Fig. 3-14 shows that only data sample point 7 (data point 160 of total data set) of rocket motor C1 is outside the 95% confidence interval. The confidence interval is calculated using Hotelling's T^2-statistic on the principal component scores (see section 2.8). Because a normal distribution is assumed and because there are only 15 spectral measurement samples the parameters of the multivariate normal probability distribution are influenced by the presence of the potential outliers. Data points 8 and 9 are close to the limits and will therefore still be regarded as outliers.

3.4 A Final Word on the Spectral Data Structure

It can now be concluded, after extensively analysing the data, that the three data sample points, 7,8 and 9 of rocket motor C1 can be omitted for further modelling purposes. This reduces the total set of IR emission spectral data considered for modelling to 417 data points. A principal components analysis is done on this data set and Fig. 3-15 shows that the two rocket motor classes are now better distinguishable using a plot of the first two principal components. Table 3-6 shows that the first two principal components capture 85.25% of the total variance in the data.

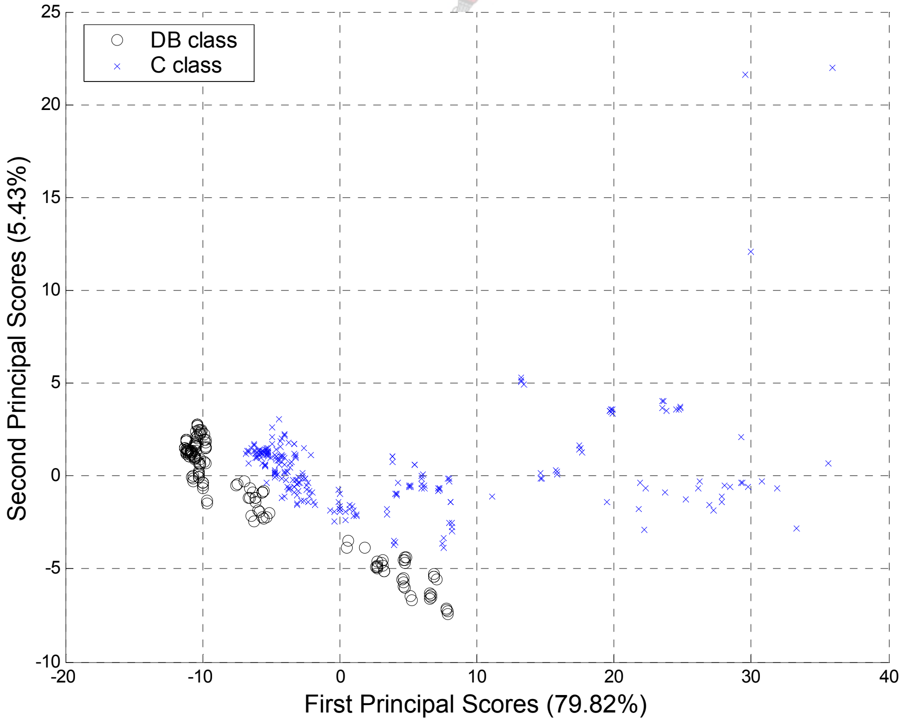

Fig. 3-15 The principal component scores plotted for the spectral data set where the outliers are removed shows a clearer distinction between classes.

Fig. 3-16 has been included here to illustrate the proximity of the spectral data of rocket motors DB3 to DB6 and the relative separation of DB1 and DB2. The same results are found in section 3.2 above where the total Euclidean distances are mapped in Fig. 3-6. The principle scores have been calculated here by excluding the C spectral data.

Table 3-6 The variance explained by each of the first 12 principal components for the data where the potential outliers are removed.

principal component	explained variance	cumulative variance
1	79.82%	79.82%
2	5.43%	85.25%
3	4.80%	90.05%
4	3.29%	93.34%
5	2.70%	96.04%
6	1.42%	97.46%
7	0.55%	98.01%
8	0.32%	98.33%
9	0.24%	98.57%
10	0.18%	98.75%
11	0.14%	98.89%
12	0.13%	99.02%

A second PCA was done on the C spectral data only and the results are shown in Fig. 3-17. It can be seen that similar to the results observed in Fig. 3-6 the rocket motors C1 and C8 are in proximity, as are the rocket motors C4, C5, C6, C7, C10, C11 and C12. The rocket motors C2, C3 and C9 also appear in close proximity to each other.

The results above further testify to the fact that the full spectral data contains redundant data. This is due to the possibility of obtaining the same results from the dimensionally reduced spectral data. The significance of these analyses is that these results can be compared to model predictions and if a test data point is predicted within its proximity family the model can be regarded as being reliable for interpolation and to some degree generalisation.

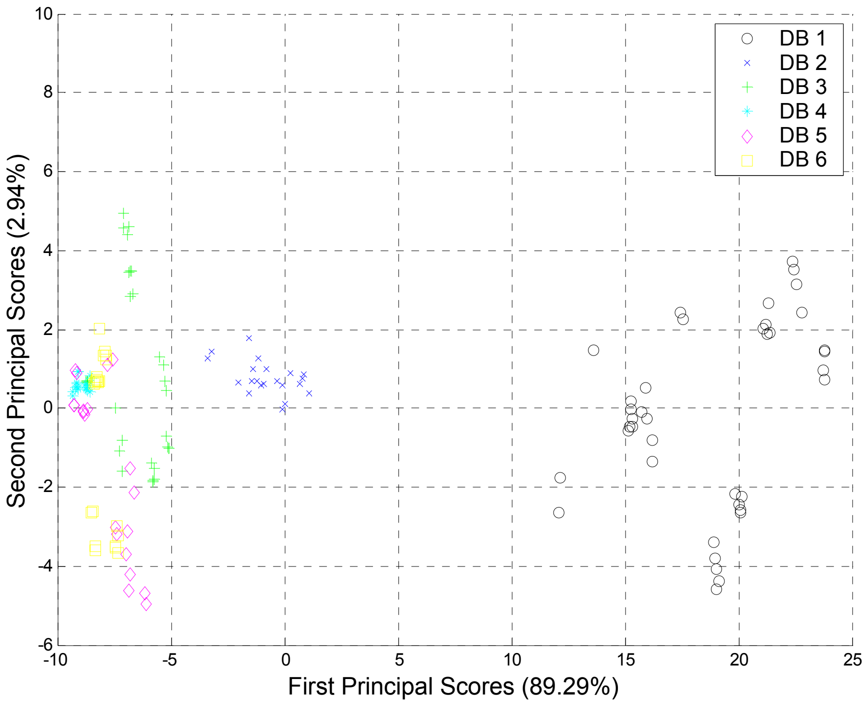

Fig. 3-16 The principal component scores calculated on the standardised DB spectral data show the proximity of the rocket motors to one another.

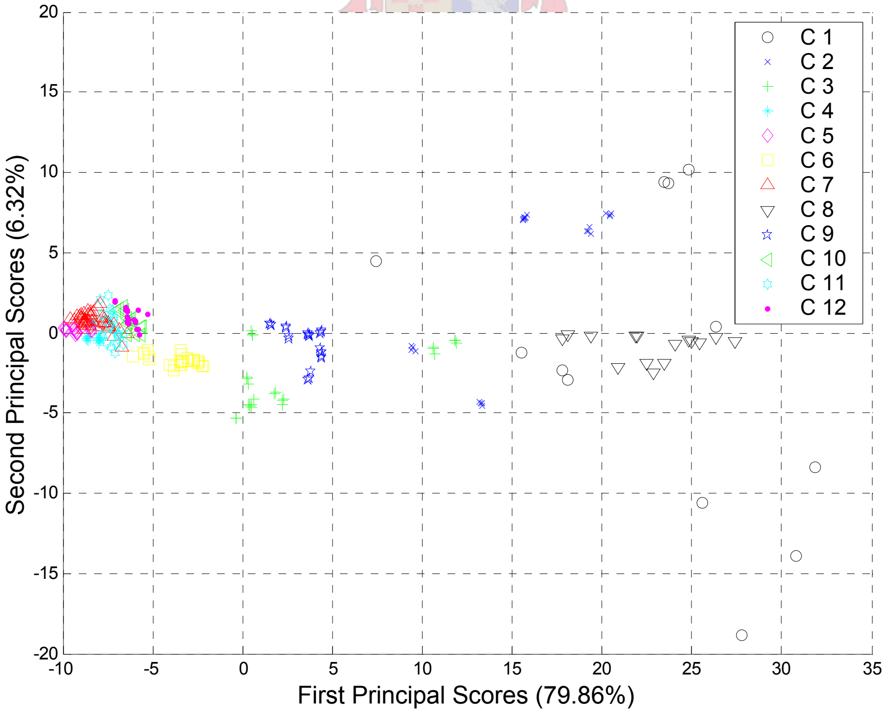

Fig. 3-17 The principal component scores calculated on the standardised C spectral data show the proximity of the rocket motors to one another.

3.5 The Explicit Clustering of the Spectral Data

The data analysis on the spectral data, using PCA, thus far shows that the reduced dimensional features produce data structures in which class memberships have a tendency to cluster together. In Fig. 3-10 it can be seen that the spectral data is slightly clustered, but there is some overlap between the rocket motor types. In higher dimensions these clusters will be better defined, however for the purpose of classification PCA is not efficient enough.

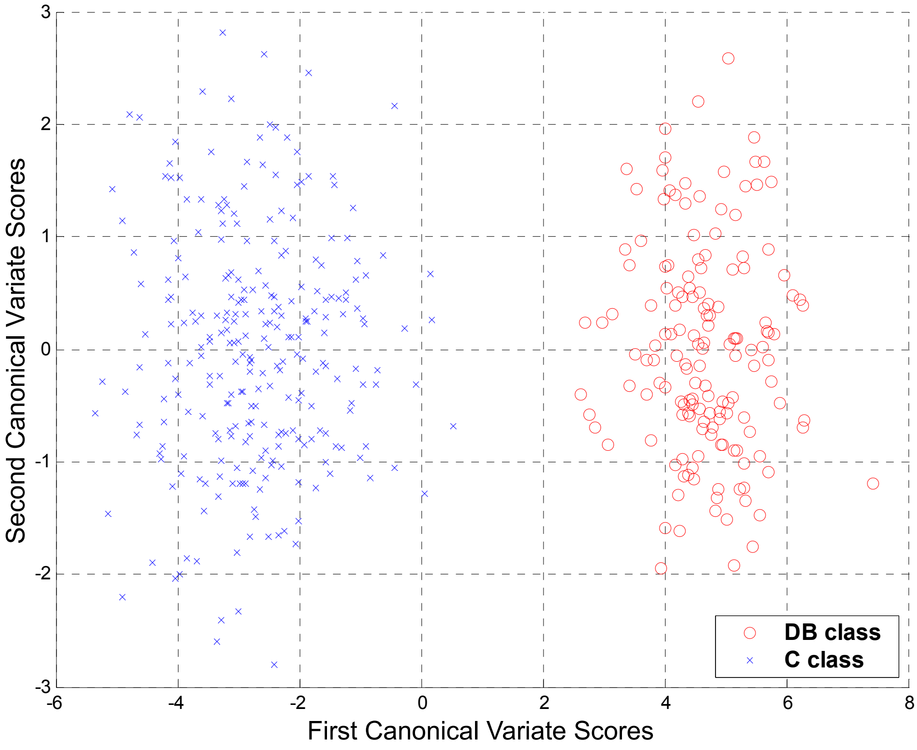

Fig. 3-18 The spectral data projected onto the canonical variates using class-based PCA.

Class-based PCA (see section 2.9) uses prior knowledge of class memberships to build features that separate the classes more efficiently. The canonical variates lie in the direction of maximum variance between the cluster centres. This is evident in Fig. 3-18 where only a single canonical variate (x-axis) is needed to be able to completely separate members into each class. Class-based PCA could therefore be used to classify a rocket motor type from its measured IR emission spectrum prior to predicting its rocket motor design parameters and chemistry.

In Chapter 1 it is mentioned that the objective of modelling the rocket IR emission spectra from the rocket design features is to be able to design rocket motors that emit the required IR spectrum for stealth characteristics. Fig. 3-19 and Fig. 3-20 show how these rocket motors can be designed to produce a spectrum that falls within one of the clusters after transformation of the standardised data by class-based PCA. The use of 2 or 3 canonical variates show good separation but it may be necessary to use more. It should therefore be possible to classify the IR emission spectra using a simple classification model or even by visual inspection. It seems the best results will be obtained by first recognising which class of fuel (C or DB) is obtained.

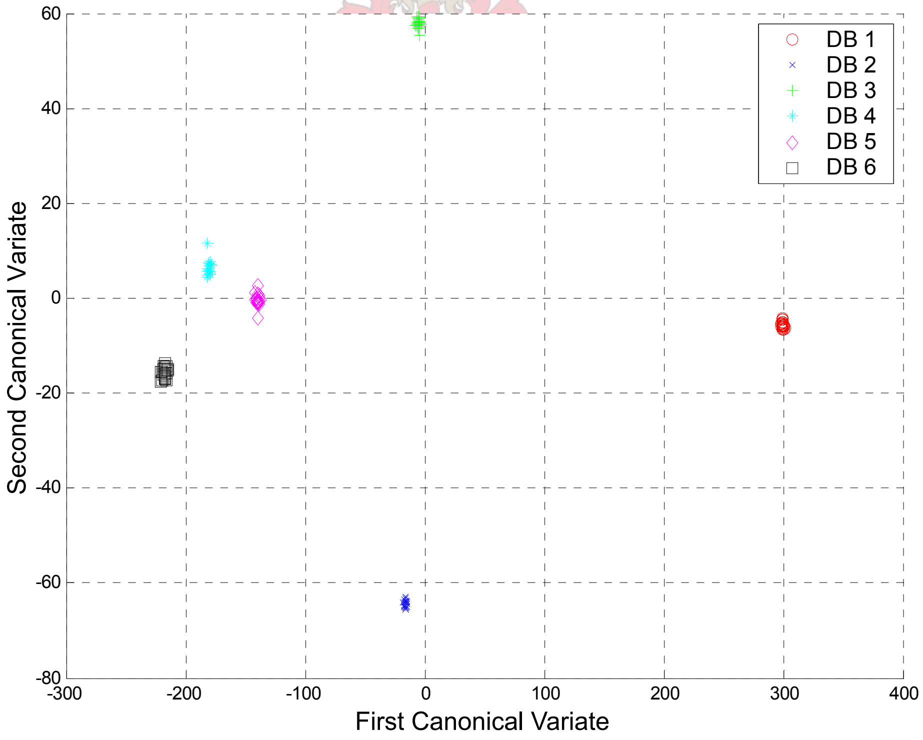

Fig. 3-19 The standardised IR spectra for the DB-class rocket motors clustered using class-based PCA.

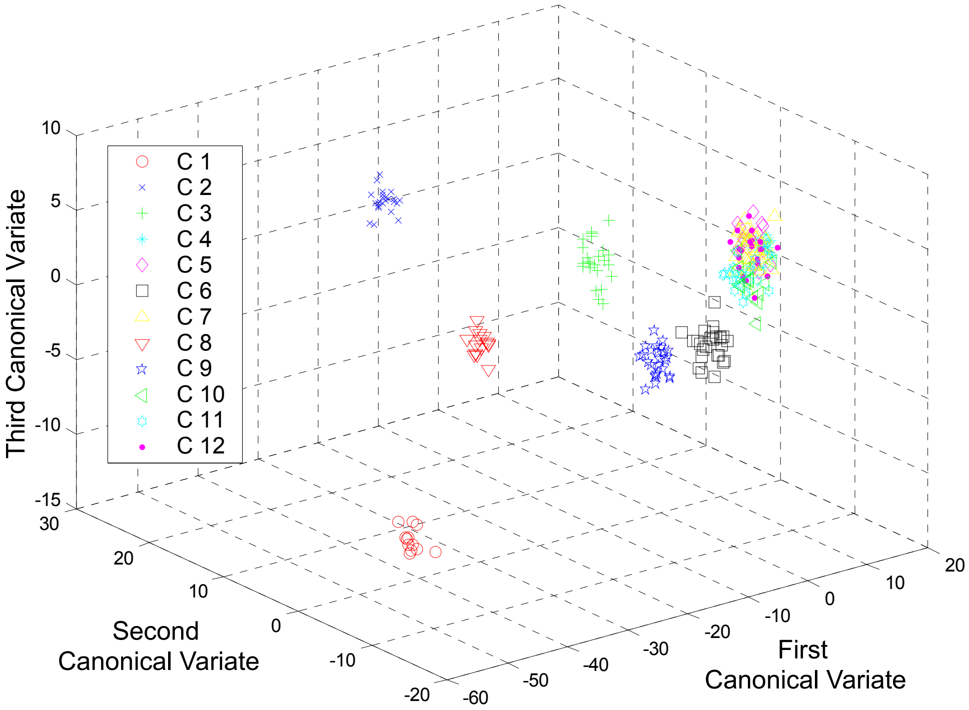

Fig. 3-20 The standardised IR spectra for the C-class rocket motors clustered using class-based PCA.

Chapter 4

Predictions of Emission Spectra

The prediction of the solid rocket motor emission spectra from their design parameters as independent predictors is known as the ‘forward’ modelling problem in this text. The reason for this is that the forward problem represents a true data set for regression purposes in the sense that the independent variables are fixed without errors and the dependent variables are measurements that are subject to random noise. For this data set the forward problem proves to be the more complex problem compared to its reverse mapping.

4.1 Methods of Model-building and Validation

The aim of this section is to motivate a methodology for the modelling and validation of the forward problem. It stands to reason that different model-fitting techniques can best be compared when they are validated on test data. This is especially the case where more complex non-linear models are involved and statistical inference techniques become more difficult to apply. Validation is also important to make sure that over-fitting is avoided. It is further important to consider the objectives of the modelling, the structure of the data and the possible limitations of the data.

The objectives of the modelling are stated in more detail in Chapter 1. Structurally, each rocket motor has 18 rocket motor design and chemical parameters as input variables and the 146 IR-absorbance values as output variables. In summary, the most parsimonious functional relationship needs to be established between the multivariate input and output variables in such a manner that the IR irradiance spectrum of a rocket motor can be predicted by simply feeding the function with the 18-dimensional feature vector of a rocket motor. The data available consists of data points from only 18 different rocket motor designs as described in section 3.1.3, which is a major limitation on the degrees of freedom that can be used to model the data. Together with the replicates in the data there are 417 data points in total. It is found that greater stability and accuracy in the predictions can be obtained when including the replicates during model building.

In this work the algorithms are programmed using the software package, MATLAB® Release 12 by Mathworks, Inc. In the applications where multi-perceptron feed-forward neural networks are trained the Mathworks Neural Network Toolbox is used.

4.1.1 Data-fitting Techniques

The aim here is to explore viable data-fitting techniques by beginning with a more simplistic approach. For this a linear regression technique is a popular choice. From such an initial investigation non-linear regression approaches can be explored in order to see if an improved model can be obtained, thus pointing to the presence of a non-linear correlation structure between the in- and output data. Due to the multivariate nature of the data and the lack of knowledge about the system used here, "black-box" data-fitting techniques seem a viable choice to find the input-output relationships.

In Chapter 2 the basic principles and algorithms are described for building models using linear PLS. Linear PLS reduces to straightforward multiple or multivariate linear regression when the number of latent dimensions equals the input variable space. Linear PLS could therefore give an indication as to whether there is dimensional redundancy present in the input space. Dimensional redundancy is desired, as there is not enough data to build an adequate model from just 18 independent predictors. If cross-validation were to produce the best validation predictions using a linear model it would point to the fact that there is an inherent linear correlation between the in- and output spaces. These models are usually more robust and therefore it is more desirable to obtain such a model.

A popular and practical choice for non-linear regression is the feed-forward multi-perceptron neural network with back-propagation (see section 2.11). Neural networks are universal approximaters and therefore do not require prior knowledge of functional relationships. This is proven by the universal approximation theorem (Haykin [1999]). The back-propagation algorithm provides a computationally efficient means of converging towards a minimum value for the cost function, in this case the residual sum of squares. The ‘traditional’ optimisation-training algorithm used is the gradient descent method, however the resilient propagation (RPROP) algorithm (Riedmiller [1993]) will be used here. This method proved to be more efficient regarding computational time and validation results. The neural networks are trained using a single hidden layer, as this was found to produce the best results in the work done by Roodt [1998]. The hyperbolic tan-function is used as transfer function in the hidden layer and the output layer (146 nodes) is linear (section 2.11).

In section 2.13.3 it was mentioned that PLS reduces the overall regression problem to a series of SISO (single input-single output) regressions. This makes PLS conducive to the problem of building a model on effectively 18 or fewer data points (or degrees of freedom). An improvement on linear PLS may be obtained by introducing feed-forward multi-layer perceptron neural networks into the PLS inner model as described in section 2.13.2. Here the NNPLS or the EBNNPLS algorithms may be useful and both methods will be applied here. The feed-forward neural network is trained using the Levenberg-Marquardt optimisation algorithm, which is ideal for the small data sets and less complicated architectures of the inner models (see section 2.10.3).

The sigmoidal neural networks are initialised using the Nguyen-Widrow initialisation algorithm (Nguyen [1990]). This method generates initial weight and bias values for each layer so that the active regions of the layer's neurons will be distributed as evenly as possible over the input space. The advantages of this method over generating random weights and biases are that few neurons are wasted since all the neurons are in the input space and training works faster as each area of the input space has neurons.

In addition to NNPLS and EBNNPLS, the radial-basis function network as described in section 2.12 are applied within the PLS inner model (RBFPLS and EBRBFPLS). The RBFPLS and EBRBFPLS models can then be compared to the NNPLS and EBNNPLS algorithms for efficiency. The advantage of working with RBF networks is that the modelling results are reproducible. This is because there is no random initialisation of the weights involved, as is the case for the sigmoidal neural networks. A brief, additional comparison will be made between using the K-means and adaptive spread OLS (ASOLS) methods of training a RBF. The ASOLS algorithm is a novel modification to the OLS algorithm and its application must therefore be justified.

It must be mentioned that from the analysis made in section 3.2 it can be expected that most latent dimensions extracted from the 18-dimensional input space are expected to be relevant to the PLS model. This is due to the lack of correlations observed in the data (see Fig. 3-7).

4.1.2 Validation of a Model

The aim here is to explore and evaluate how well the solid rocket motor data can be modelled using the different model-building techniques. In order to obtain legitimate comparisons of the different techniques it is important to obtain the optimum number of parameters for each model by adequate model validation. In section 3.1.3 it was discussed that due to the shortage of degrees of freedom available for building a model it is evident that the most appropriate method of validation is that of model cross-validation as explained in section 2.6.3. It must be said that the emphasis here is on finding the best of the candidate models, even if the predictions on unseen data are not always accurate. After cross-validation a single, overall model can be built by training on the entire data available and by using the optimum number of parameters determined from cross-validation.

The complete data set of all eighteen rockets includes 417 spectral data samples (see section 3.3), which may create the misconception that there is enough data to be split into a large training subset and a large test set. The data set, however, consists of 399 replicates (see section 3.1.3) and according to Snee [1977] these replicates should be removed for model validation. This is motivated by the fact that splitting the data along replicates does not change the correlation structure between the target values and the prediction values. A prediction data set selected in this way would be a poor check on the model. The entire block including all spectral replicates corresponding to a specific rocket motor must be regarded as an independent 'data point'. A validation data set must therefore consist of one or more blocks of spectral data corresponding to one or more rocket motor features.

How should the data be split for cross-validation? From the discussion above there are 18 independent samples to be considered. The leave-one-out cross-validation discussed in section 2.6.3 would require eighteen separate training sessions. In the case of 'leave-out-more-than-one' cross-validation a heuristic is to divide the complete data set into 10 subsets of training and validation data (Brereton [1990]). For the rocket data a close approximation to this heuristic would be to split the data into 9 subsets, each subset containing 16 training rockets and 2 validation rockets. In comparison to the leave-one-out method this allows for a reduction in the computation time required to complete a complete cross-validation run. The 'leave-out-two' cross-validation also provides a good idea of the model complexity required for a model required to make predictions over two different classes of rockets. This is only possible if for many of the two rockets left out for validation each of the two rockets represents a different class of rockets.

For the leave-out-two cross-validation procedure there are numerous methods for choosing the combination of data points to be selected for each training and validation step. Three of the methods described in Wise [2000] have been considered here, the contiguous block method, the Venetian blinds method and random selection until each point has been removed once for validation. The contiguous block method implies leaving out the first two samples for the first training step, the next two for the next training step and so forth. For training session τ the Venetian blinds method uses sample points, τ and $\tau + 9$ for validation. The random method speaks for itself. See Table 4-1 for a detailed illustration of how the entire rocket data set is split into nine training-validation pairs.

From Table 4-1 it is evident that the contiguous block method will include three training steps consisting of validation on the DB class rockets and six training steps on C class rockets only. This may cause greater bias towards a certain class within the training sets and because there are only 6 rockets within the DB class leaving out two of them at a time may lead to an unacceptable loss in information for the training set. The Venetian blinds method guarantees the most distributive profile relative to the two classes of DB and C rocket motors.

Table 4-1 The 18 rockets split up into their 9 training-validation sets for leave-out-two cross-validation (each rocket is assigned to a set number as shown).

Rocket type	Contiguous Blocks	Venetian Blinds
DB1	1	1
DB2	1	2
DB3	2	3
DB4	2	4
DB5	3	5
DB6	3	6
C1	4	7
C2	4	8
C3	5	9
C4	5	1
C5	6	2
C6	6	3
C7	7	4
C8	7	5
C9	8	6
C10	8	7
C11	9	8
C12	9	9

In section 2.6.4 it is shown how the PRESS (or MSECV) values must be plotted against the model complexity values in order to find the optimum number of model parameters required to find the most generalised functional relationship. In the case of linear PLS each latent dimension contains a single linear regression parameter between the input and output scores. An increase in model complexity thus represents an increase in the number of latent dimensions applied and for this reason it is appropriate to plot PRESS-values against monotonically increasing latent dimension values. In the case of building a sigmoidal feed-forward neural network with a single hidden layer each consecutive addition of a hidden node represents a uniform increase in complexity. The PRESS-values can subsequently be plotted against the number of hidden nodes.

The first attempts at modelling with cross-validation were done using the leave-two-out Venetian blinds method. The results were compared to the leave-one-out method and it was found that leave-one-out cross-validation produced the lower PRESS-values (see section 2.6.3) when plotted against model complexity. This implies that leave-one-out cross-validation produced the better-predicted IR spectra on unseen data. A reason for this is that more information is lost per training session due to the 16 independent sample points (rocket motor features) used as opposed to the seventeen used for leave-one-out cross-validation. This supports the view in section 3.1.3 that the available data is sparse and therefore it is essential that training be done on the maximum number of independent data points possible during each session of cross-validation. Leave-one-out cross-validation is subsequently chosen as the method of cross-validation to be considered.

In the discussion above the method of cross-validation has been determined. In addition the methods of evaluating cross-validation by plotting PRESS vs. model complexity for linear PLS and feed-forward neural networks have been described. In the case of the NNPLS and RBFPLS related algorithms the evaluation is more complex. This is due to the fact that varying numbers of hidden nodes are obtained for the inner model of each extracted latent dimension. The implication is therefore that the PRESS-value can be plotted against increasing latent dimension or the total number of hidden nodes used counted over all the latent dimensions extracted. This is due to the fact that a single hidden-layered sigmoidal feed-forward neural network can be obtained from the transformation of a NNPLS (or EBNNPLS) model (see section 2.13.2).

A further complication when using the NNPLS (or EBNNPLS) and RBFPLS (or EBRBFPLS) algorithms arises when finding the optimum number of hidden nodes required for each inner model of a latent dimension. This issue is discussed in section 2.13.2 whereby Akaike's final prediction error (FPE) is calculated for a validation data set for each latent dimension. Each training session within leave-one-out cross-validation has only a single independent data set (rocket feature from one rocket type) available for validation. Each one of the 18 rocket features covers a unique but important part of the input space to be modelled. During a particular training session the optimum number of nodes per latent dimension is decided upon based on the minimum FPE-value corresponding to the single validation point. Subsequently, the result of leave-one-out cross-validation is that this method leads to a varying number of nodes obtained for the same corresponding latent dimension across the numerous modelling sessions. It therefore becomes difficult to determine a single optimum number of hidden nodes for a latent dimension.

It can be argued that leave-one-out cross-validation gives an idea of how many hidden nodes are required per latent dimension. This implies that the average number of hidden nodes obtained over all training sessions for a specific latent dimension could be used to determine a final overall optimum model. The average may however not be an accurate measure of the overall performance.

The fact that the decision, as to how many hidden nodes are retained per inner model of a latent dimension, is based on the minimum FPE-value of a particular validation set leads to a fair amount of bias towards that validation set. This means that by the time all 18 latent dimensions have been built up during one of the training sessions the final sum-square error of the validation point is at its minimum since the training has taken place in the direction that favours the prediction of the single point. After all 18 sessions have been completed the overall PRESS-value will therefore be at an overall minimum due to the individual bias during each training session. The result is that there is no characteristic 'dip' where the PRESS value goes through a minimum at some latent dimension, $a \leq 18$. This phenomenon will be illustrated later in this chapter. Furthermore it becomes evident that each training session is customised for its single validation point (see section 4.2.3). This type of cross-validation using NNPLS or RBFPLS can therefore strictly–speaking not be used to compare PRESS values to those obtained for the linear PLS and feed-forward neural network models.

A solution to the problem is to cross-validate each latent dimension one at a time. This means that all 18 training-validation sets are trained together up to latent dimension, a, in which an MSECV-value for each number of hidden nodes used is calculated. These MSECV-values are based on the output scores of the particular latent dimension. The number of hidden nodes to be used for the inner model of the particular latent dimension then corresponds to the minimum MSECV-value for that latent dimension. The algorithm for each latent dimension using NNPLS is thus as follows:

0. Begin the algorithm by setting indices a=1, b=1, c=1. Here index a is the number of the current latent dimension, index b is the current number of hidden nodes and index c is the current training-validation session number.
1. Obtain the in- and output scores, $\mathbf{t}_{ab}^{(c)}$ and $\mathbf{u}_{ab}^{(c)}$ as well as the inner model neural network parameters, $\omega_{1b}^{(c)}$, $\omega_{2b}^{(c)}$, $\beta_{1b}^{(c)}$, $\beta_{2b}^{(c)}$, for a=1 by using the NNPLS or EBNNPLS algorithms described in section 2.13.2.
2. Calculate the corresponding in- and output scores for a and b, $\mathbf{t}_{val}^{(c)}$ and $\mathbf{u}_{val}^{(c)}$ of the validation data by using equations (2-142) to (2-145).
3. Calculate the sum square error calculated on the validation predicted output scores: $\mathrm{SSE}_{ab}^{(c)} = \left(\hat{\mathbf{u}}_{val}^{(c)} - \mathbf{u}_{val}^{(c)}\right)^{T}\left(\hat{\mathbf{u}}_{val}^{(c)} - \mathbf{u}_{val}^{(c)}\right)$
4. Go to step 5 if τ = c otherwise repeat steps 1 to 3 for training-validation set c+1.
5. Calculate $\mathrm{MSECV}_{ab} = \frac{1}{n}\sum_{c=1}^{\tau} \mathrm{SSE}_{ab}^{(c)})$

6. Go to step 7 if $\varpi = b$ for some predefined number of hidden nodes, ϖ, otherwise repeat steps 1 to 5 for b+1 hidden nodes in the NNPLS inner model.
7. Find the number of hidden nodes where MSECV is a minimum, i.e.:
   H = b where MSECV = min $\{MSECV_{ab}, b = 1,2,\ldots,\varpi\}$
8. Stop training when h = a, otherwise repeat steps 1 to 7 for latent dimension a+1.

It should be noted that the value of n in step 5 of the algorithm is equal to 417. This is not correct but the choice of the value of n is of no consequence, as qualitatively a PRESS-value could be used instead of the MSECV-value.

The value of ϖ in the algorithm can be determined according to some criterion within the programmed algorithm. In this work further training with increased numbers of hidden nodes is stopped once the MSECV-value for a latent dimension is at a maximum for the last four iterations of increasing number of hidden nodes.

4.2 Results and Discussion of Candidate Models

The cross-validation results for the various model-building techniques are evaluated first. The results from the cross-validation are used to build a model on the entire data set (all 417 data points). This is done by using the optimum number of parameters as determined by the cross-validation. This model can be referred to as the 'overall' or 'training data' model.

The dependent and independent data sets of each training data set within cross-validation is standardised as described in section 2.4.2. The corresponding validation data is then centred and scaled using the means and standard deviations of the training set. All statistics scores, unless stipulated otherwise, are calculated on target and predicted values after they have been post-processed back to their original scale.

In the work done by Roodt [1998] it was found that the neural network model predicted spectral responses inversely proportional to the chamber pressure, P_C. This is in contrast to expectation since an increase in chamber pressure is expected to lead to a rise in chamber temperature for a fixed fuel composition and rocket shell design. The chamber pressure values were thus inverted for model-building in order to obtain a directly proportional response in the spectral predictions. All the models built here include this amendment made to the chamber pressure.

The values of SSEP in this text are defined by the following equation:

$$\mathrm{SSEP} = tr(\mathbf{E}^T\mathbf{E}) \tag{4-1}$$

SSEP is therefore the trace of the diagonal of $\mathbf{E}^T\mathbf{E}$. Here $\mathbf{E}$ is the $n\times l$ matrix of residuals calculated using the predictions made by the overall model after cross-validation has been completed. The dimensions of $\mathbf{E}$ are thus n=417 and l=146. It is also evident that SSEP is the summation over all the SSE-values corresponding to each of the 146 output variables. From section 2.5.2 SSEP_{min} is therefore the summation over all SSE_{min}-values for each output variable as defined by equation (2-57). In this case n=18 for $\mathbf{E}_{min}$.

The PRESS-values are defined by equation (2-65) in section 2.6.3. Here the replicates are built into $\mathbf{E}$ and therefore n=417. However, in order to calculate the MSECV_j-value for each output variable in equation (2-63) the value of n must be 18. This is in agreement with the statement made by Snee [1977] (see section 4.1.2) and also due to the fact that a leave-one-out cross-validation for the 18 rocket motor designs allows for 18 degrees of freedom. This justifies the definition of PRESS_{min} for which the residuals are calculated using the mean irradiance spectra for each rocket motor (n=18) and each of the 18 predictions made during cross-validation.

The $\mathrm{MSEP}_{rs,j}$-value defined in section 2.14 is therefore calculated using the $\mathrm{SSE}_{min,j}$-value divided by n=18 for each output variable. This makes it possible to legitimately use equation (2-150) for n=18 and therefore calculating the pdf-value for each output variable.

Due to the fact that all the output variables consist of the same units it is sensible to evaluate the various model complexities by using the SSE-values summed over all the output variables, as described for PRESS and SSEP. It is further found that due to the high degree of correlations in the output data (see Fig. 3-8) a change in model complexity qualitatively affects all output variables in a similar manner. It is also not feasible to separately find the optimum model complexity for each of 146 output variables.

4.2.1 Linear PLS

The linear PLS model is evaluated over all 18 latent dimensions, however in Fig. 4-1 the PRESS-values for latent dimensions 14 to 18 have been left out in order to maintain the scale of the graph. In Appendix A the omitted PRESS-values are shown in Table A-2. From Fig. 4-1 it is evident that the minimum PRESS-value lies at latent dimension 2 (LD 2). However, another very low PRESS-value occurs at LD 11. The graph therefore does not have the customary dipping shape as discussed in section 2.6.4. The SSEP-values show the expected continuous decline due to the increase in model complexity.

Fig. A-1 has been included in order to illustrate the similarity in structure as far as the PRESS-values are concerned. The $SSEP_{min}$-values for LD 17 and LD 18 are zero (see Table A-2). As far as LD 18 is concerned this result is expected, as there are 18 data points (all 18 rocket motors) and 18 input variables. The regression problem reduces to a unique solution of 18 variables in 18 equations as would be the case if multiple linear regression were applied. This result supports the discussion in section 3.1.3 on the sparseness of the data and the need to find dimensional redundancy in order to obtain lack-of-fit degrees-of-freedom. The result for LD 17 is not expected unless the last latent dimension contributes negligibly towards the overall model variance. This is very likely, especially as the seventh input variable, Fluorine (F) only contains a single data entry.

From the results in Fig. 4-1 and Fig. A-1 it is evident that further analysis is required in order to decide how many latent dimensions are required to build the most feasible model. The uncharacteristic shape of the PRESS-curve can be attributed to the fact that the latent variables (in- and output scores) are possibly non-linearly related. The hyper-planes picking up the primary dimensions after LD 2 seem to miss valuable information and then pick it up again at LD 11. After LD 11 there is no more valuable information. At this stage it could therefore be argued that although PRESS at LD 11 is slightly higher than at LD 2 there may still be too much information missing at LD 2 and that LD 2 merely averages out the PRESS-value much better. The PRESS-value used here is after all the summation over all 146 output variables.

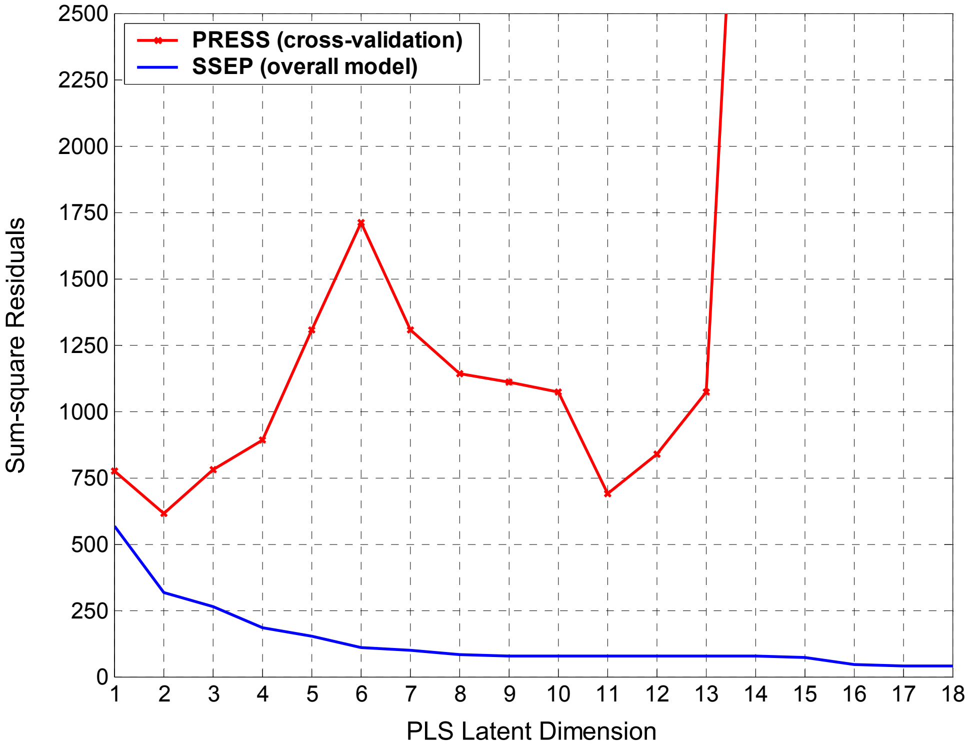

Fig. 4-1 The sum-squared residuals obtained from building a linear PLS model.

An Inspection of Fig. 4-2 shows the expected incremental increases in explained variances for both the input and output data. This is typical for PLS regression as opposed to PCA regression where it may be possible for incremental decreases to occur. Table A-1 shows the explained variance increments for the standardised data on which the models are built.

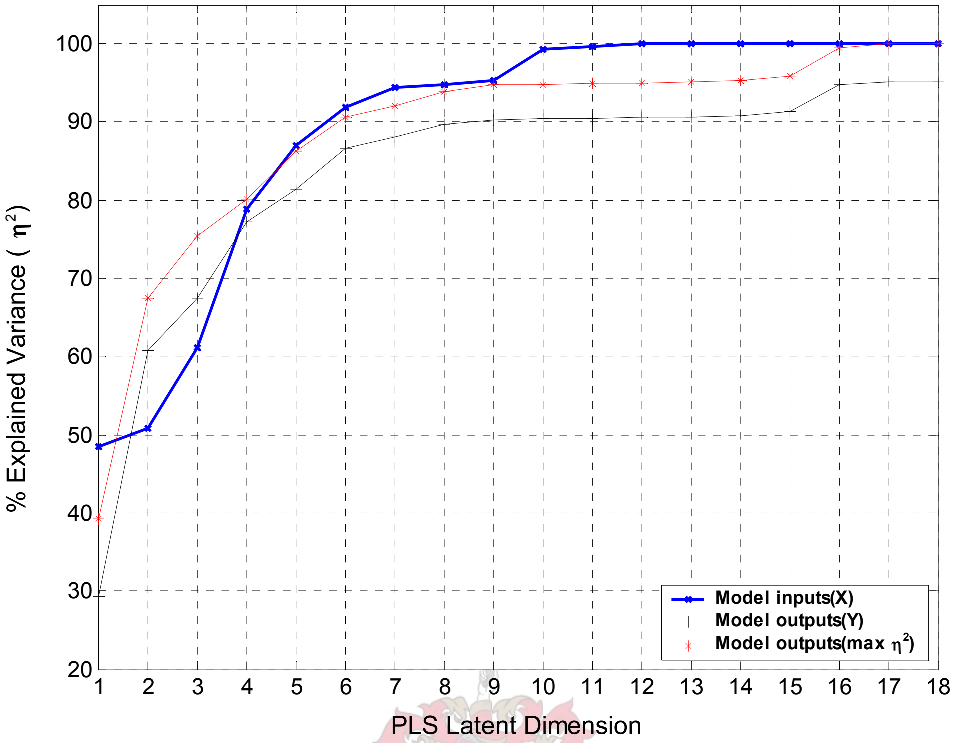

Fig. 4-2 The characteristic curves of increasing explained variances for the overall linear PLS model (built on all 18 rocket motors).

At LD 2 only 50.76% (see Table A-2) of the X-block data variance is explained by the model. It is not impossible for such a model to be feasible, but it is not very likely. The explained variance of the Y-block is fairly low at 60.86%. From the above observations it seems that LD 10 or LD 11 (99.7% and 90.5% X-block and Y-block η^2) are the more feasible choices for a model. A very low dimension is not expected when considering the low degree of correlation found in the X-block data (see Fig. 3-7). The maximum explained variance (max η^2) for the Y-block shows that 100% is achieved at LD 17 and LD18. This is expected, as the SSEP-values are zero here. The 'normal' explained variance for the Y-block cannot reach 100% due to the replicates built into the calculation (see section 2.5.2).

In sections 2.6.3 and 4.1.2 it was explained how all the validation sets during cross-validation are re-combined to form a 417×146 cross-validation data set (CV) of predicted outputs. In Fig. 4-3 the squared correlations ($R^2 = (r_{y\hat{y}})^2$) between the predicted cross-validation model outputs and the target output variables are plotted. The region of the spectrum resulting from IR absorbance due to water lies approximately between wavelength numbers 20 and 45 (2.7 to 3.3 μm). Fig. 4-3 shows that the predicted variables from unseen data are best correlated to the target values in this region. The dip just before the 'water'-peak does not feature in any of the model complexities because these absorbance values are zero for all rocket motors. The CO_2-peak between wavelength numbers 100 and 140 (4.3 to 5μm) appears to be predicted less accurately. It is evident that LD 11 produces most of the highest R^2- values in the CO_2-peak region.

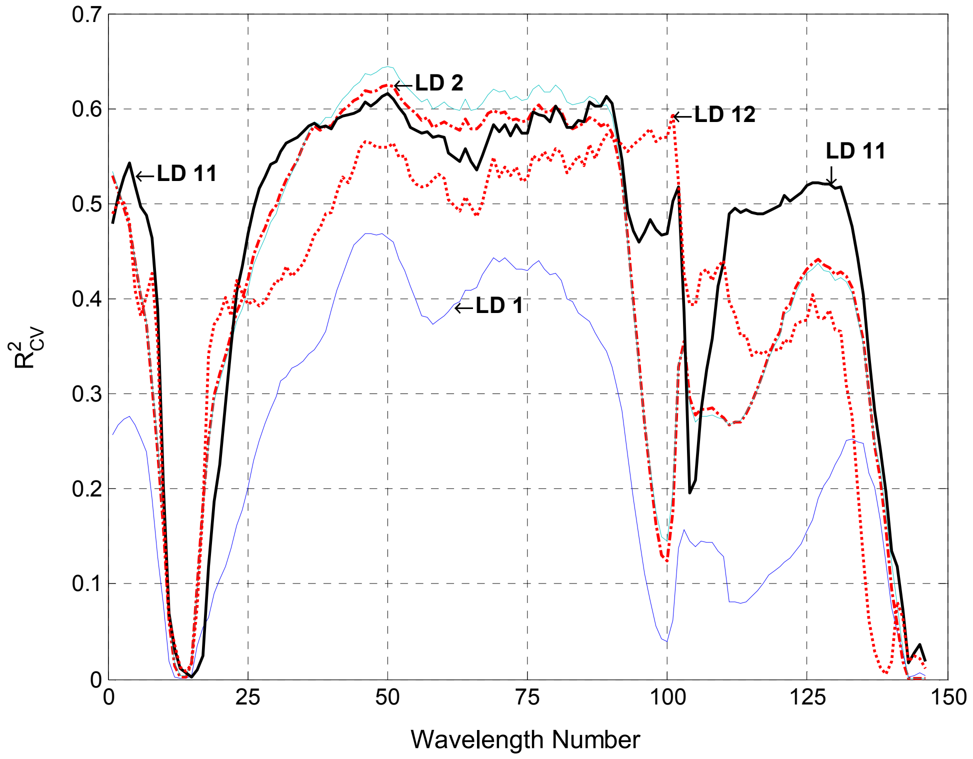

Fig. 4-3 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their targets for linear PLS.

Fig. A-2 has been included to illustrate that the $R^2_{cv(max)}$ –values show the same trends and that the maximum obtainable squared correlations exceed 0.6.

From the discussions above it appears that a linear PLS model using 10 to 12 latent dimensions is feasible. In particular, the model with 11 latent dimensions performs the best on cross-validation data. If it were possible to obtain a larger data set approaching the population representation a slightly different number of latent dimensions may have been optimal. For the purpose of this study the best results for each model are required to find the maximum potential of the models given the sparse data. From Table A-2 it becomes evident that the best predictions on cross-validation data and good predictions on the overall model are generally obtained at LD 11. Further, it can be seen that the average R^2_{cv} –value for LD 2 (0.431) is slightly lower than that of LD 11 (0.461).

In Fig. 4-4 the R^2-values for all 146 output variables are plotted using the overall model with 11 latent dimensions. Similar tendencies to the cross-validation data results can be observed. There are lower correlations in the CO_2 absorbance region and (as expected) almost no correlations in the dip before the water-peak.

The pseudo degrees of freedom (see section 2.14) have been calculated for each of the 146 output variables using 11 latent dimensions for the overall model. For any model the model degrees of freedom (df) are independent of the output variables. Van der Voet [1999] shows that his method of calculating pseudo degrees of freedom (pdf) also exhibits this independence. However, as the pdf is an approximation there are deviations among output variables. Fig. 4-5 shows how the pdf-values vary for the different output variables. The largest deviations occur at the CO_2 absorbance region and the region before the water absorbance peak.

The average pdf-values for each LD have been plotted together with the df-values of an equivalent multiple linear regression model with increasing number of input variables in Fig. 4-6. From this figure it is evident that linear PLS generally includes more model degrees of freedom with increasing number of LD's than the multiple regression equivalent. This can be attributed to the outer model projections made in the algorithm and the subsequent increased amount of information captured by each latent dimension. The pdf-value for LD 17 is 18, but it remains at 18 for LD 18. This is due to the lack of information carried by LD 18 as discussed above.

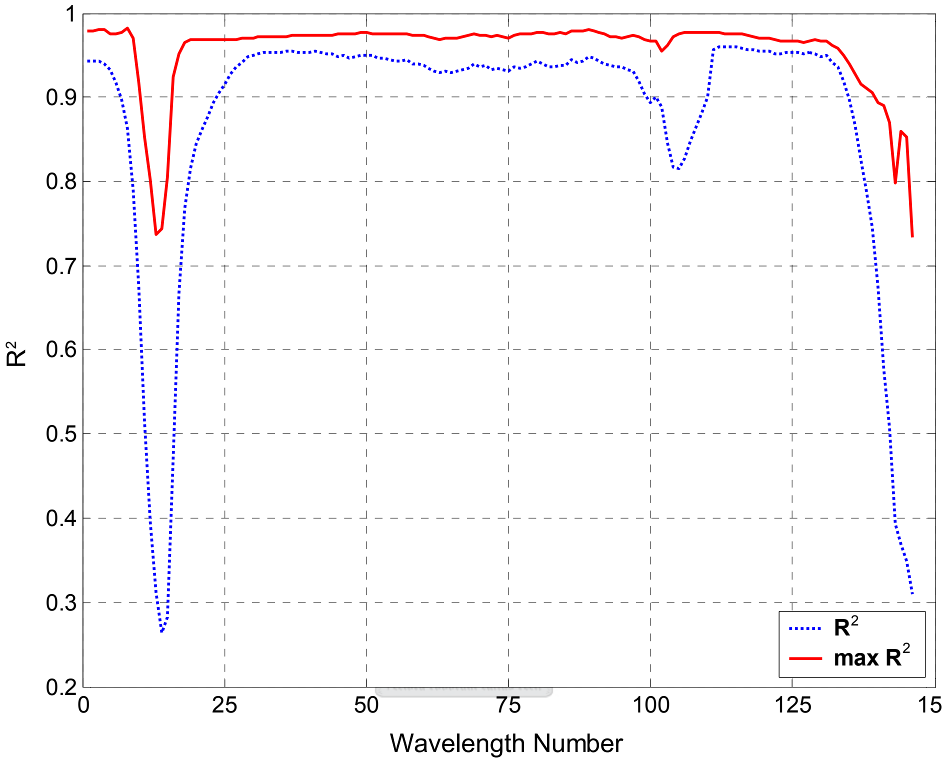

Fig. 4-4 The squared correlations between the 146 overall model output irradiance absorbance values and their targets for 11 latent dimensions.

Table 4-2 shows the total number of linear PLS parameters cumulating with increasing number of latent dimensions. These parameters include the in- and output loadings, which are customarily not included in the determination of the model degrees of freedom, as was shown by the naïve formula in equation (2-146). From the calculations of the pseudo-degrees of freedom it is obvious that by using the total number of parameters to calculate the model degrees of freedom would be a grossly over-valued estimate.

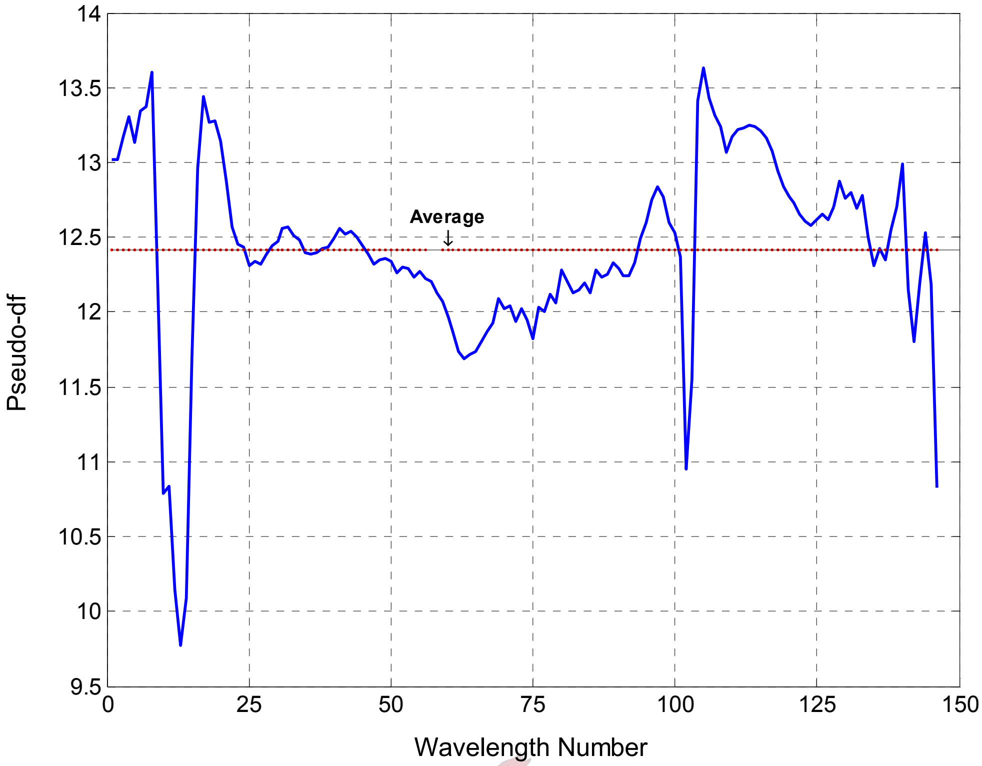

Fig. 4-5 The number of pseudo-degrees of freedom (pdf) for each of the 146 output variables using the overall linear PLS model with 11 latent dimensions.

The performance scores thus far have no bearing on how well the model managed to predict the individual irradiance spectra. In Fig. 4-8 to Fig. 4-10 the individual unseen validation set predictions from leave-one-out cross-validation for each rocket motor type are shown. The predictions are presented together with the mean target spectrum and the band of two standard deviations for the repeat measurements of the particular rocket motor. Confidence intervals were not calculated, as they require model pseudo-degrees of freedom to be calculated. This requires that the training sets of each cross-validation session must be cross-validated themselves in order to satisfy equation (2-150). This is an extensive exercise requiring large amounts of computational time, especially for the non-linear models, which require repeat cross-validation runs in order to validate the results (see next section).

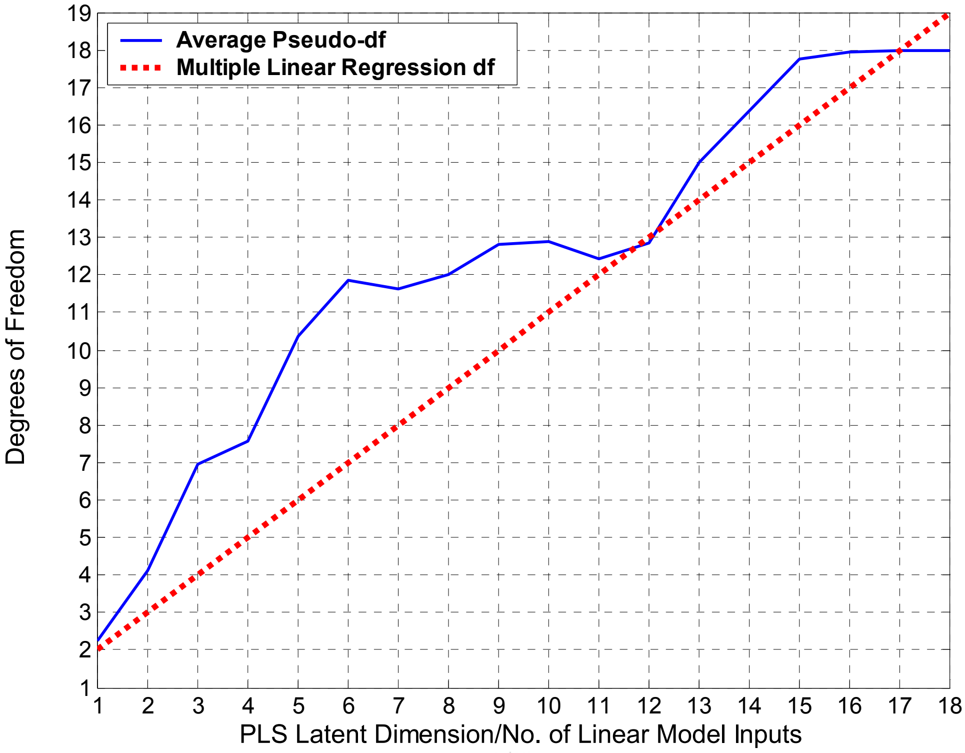

Fig. 4-6 The average pdf over all output variables for linear PLS built with increasing number of latent dimensions compared to multiple regression df.

Table 4-2 The total number of parameters per output variable using linear PLS.

LD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Parameters Per Output Variable Cumulated	20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360

The graphs (Fig. 4-8 to Fig. 4-10) show that, in general, the linear PLS models are able to predict the C-class rocket motors fairly well with the exception of C5. The predictions of DB3 to DB6 and C5 are failures, especially as these predictions, except DB6, predict negative spectra. Realistically this means that the model effectively predicts zero transmittance, as a negative spectrum is not possible. The inaccurate prediction for C5 may be due to the fact that C5 contains potassium (K) and no aluminium (Al). The linear model is upset by the apparent similarity the rocket motor design parameters has with the majority of the DB-class predictions. The prediction for the emission spectrum of DB1 is too low. This could be attributed to the model ‘seeing’ the DB1 as a DB-class rocket motor, whose irradiance spectra are generally lower than the C-class motor designs. This distinction is especially possible as the DB-class rockets contain Cu and Pb, which are not present in the C-class designs. The above results therefore show that 12 out of the 18 (66.7%) unseen predictions are successfully predicted.

In Fig. 4-11 to Fig. 4-13 the overall model (see definition in section 4.1.2) predictions are presented together with their 95% confidence intervals (see section 2.15). Negative values for prediction intervals are rounded to zero, as negative values are not realistic predictions. The predictions look more promising, especially for motor designs DB3, DB5 and DB6. Although these predictions appear highly accurate the confidence intervals are extremely large. This lack of precision is confirmed by the poor cross-validation predictions. The predictions for C5 and DB4 remain poor. The predictions for the remaining rockets are highly accurate. If one considers all predictions with accurate predictions as well as realistic confidence intervals the prediction efficiency is 13 out of 18 (72.2%). The significant difference in prediction performances between the overall model and the unseen data points towards the fact that the linear PLS model up to LD 11 may be slightly over-trained. This is because the redundant, noisy components of the input space may be included in the higher dimensions, producing the poorer unseen predictions (see section 2.6.4).

It is interesting to observe the linear relationships between the in- and output scores (latent variables) as shown in Fig. A-3 and Fig. A-4 for LD 1 to 12. The orders of magnitude of the in- and output scores remain constant up to LD 10 after which there is a slight decline. From these graphs it is also evident how large the variation is in repeat measurements on some of the motor designs. The first 6 data points for the graph of LD 1 are from the 6 motor designs of the DB-class. The data points lying far off the prediction lines are therefore not necessarily of the DB-class. The shape of the prediction line and the relationship of the in- and output scores are dependent on prior latent dimensions. The overall effects of the predictions are the combined effects over all the latent dimensions. See section 4.3 for further discussion.

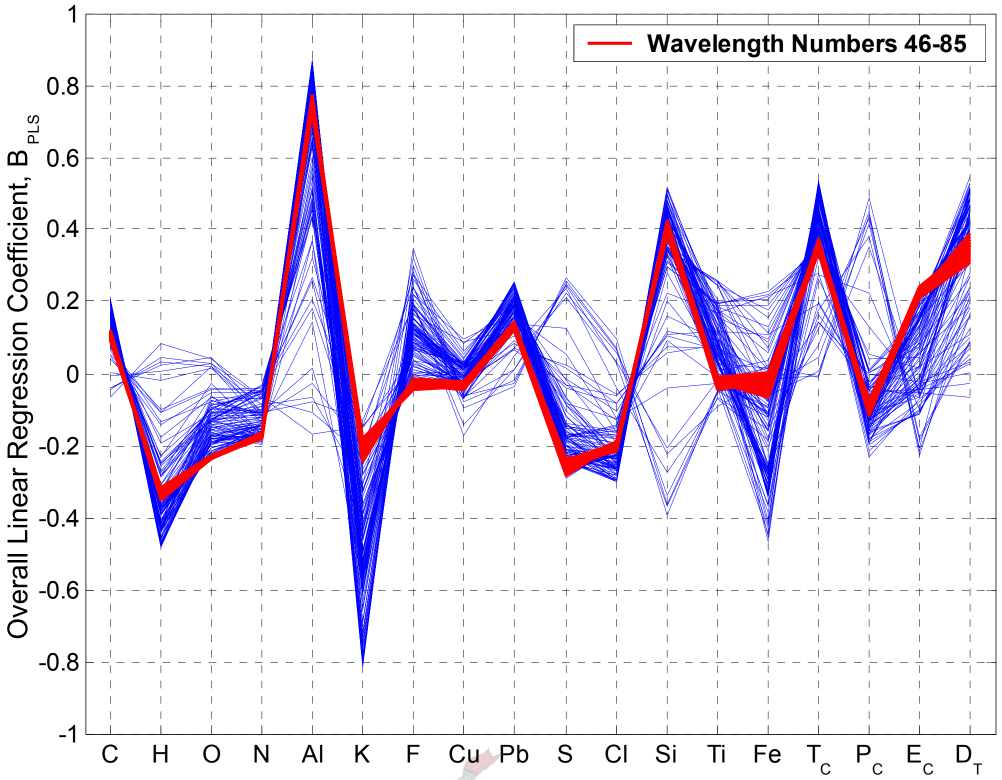

Fig. 4-7 A plot of the linear regression coefficients of the linear PLS model for all 146 output variables.

A convenient feature of linear PLS is that once a model has been built the functional relationship to the output variables can be expressed in terms of linear regression coefficients. The derivation leading to equation (2-133) is given in section 2.13.1. A plot of the regression coefficients is shown in Fig. 4-7 where the effect of the coefficient of each input variable can be seen. Aluminium (Al) and potassium (K) clearly have strong influences on the predictions. The range from wavelength numbers 46 to 85 has been highlighted as this region shows the same effect on the predictions across all variables where the R^2-values are high. Furthermore the Al-coefficient has a strong influence in this band, whereas the K-coefficient has a lesser influence. The variables Si, Fe, T_C, and D_T also have significant influences on the model predictions for most output variables. From these results it seems that the strategic addition of additives can ‘mould’ the emission spectra in predetermined regions as required.

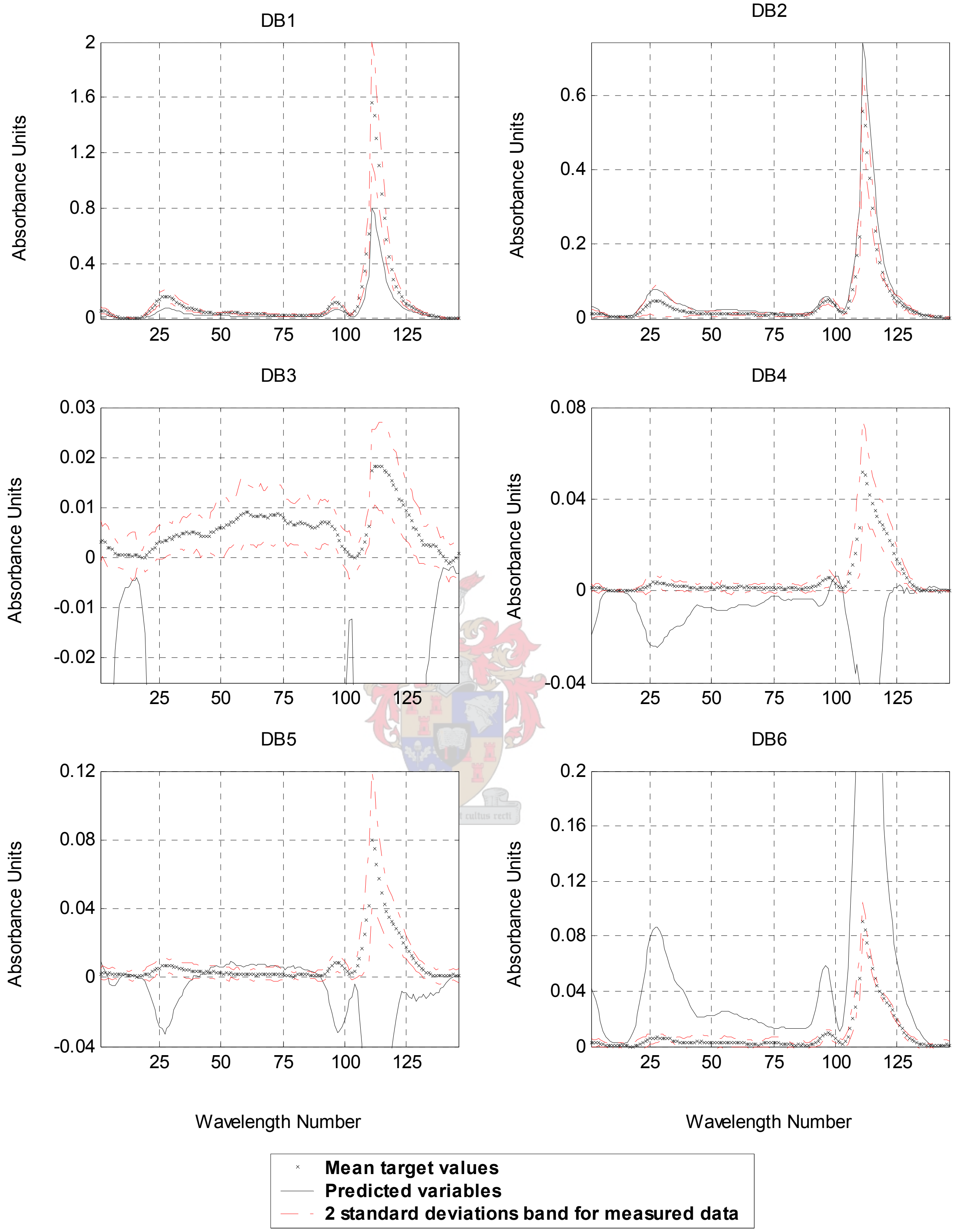

Fig. 4-8 The plume irradiance predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of linear PLS.

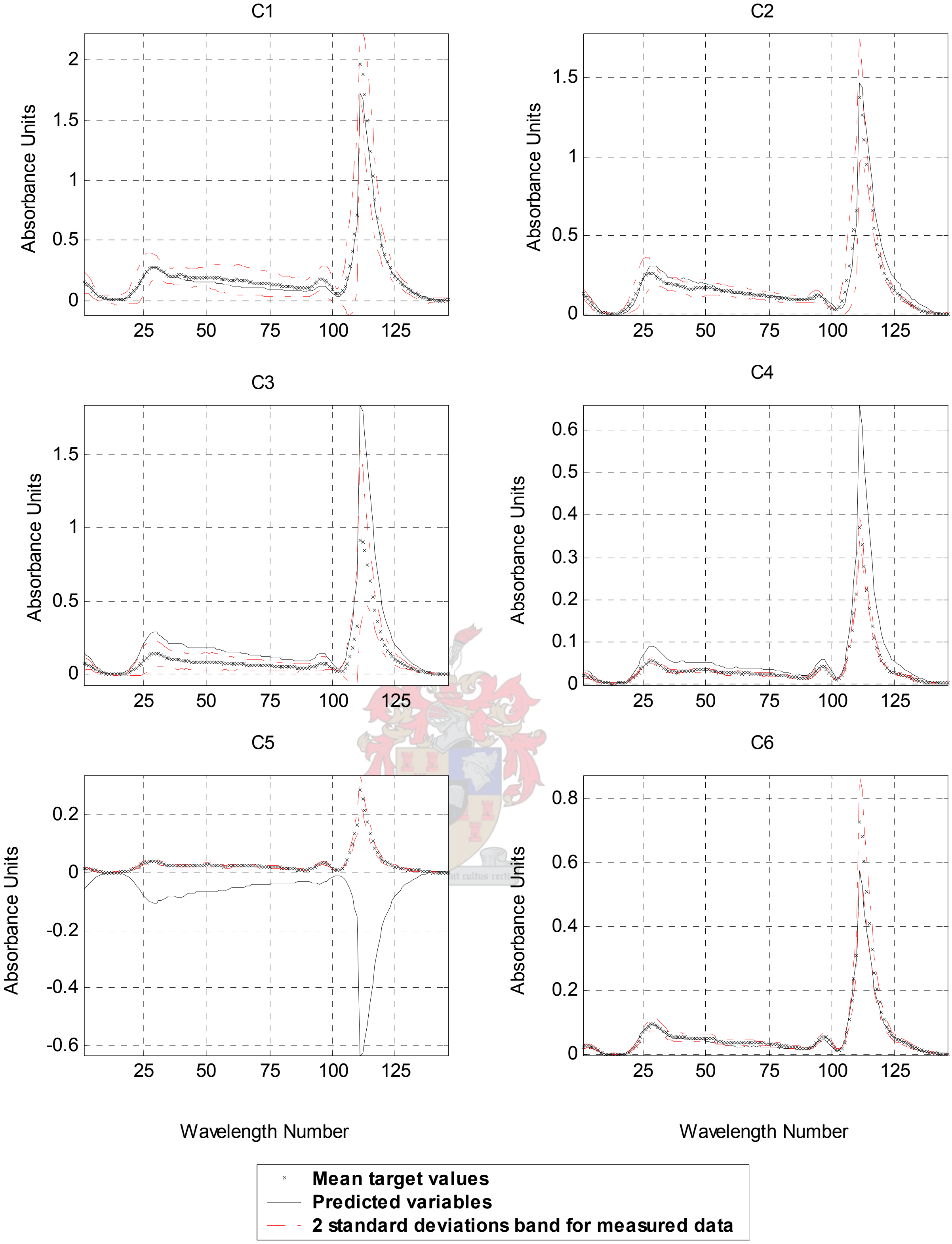

Fig. 4-9 The plume irradiance predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of linear PLS.

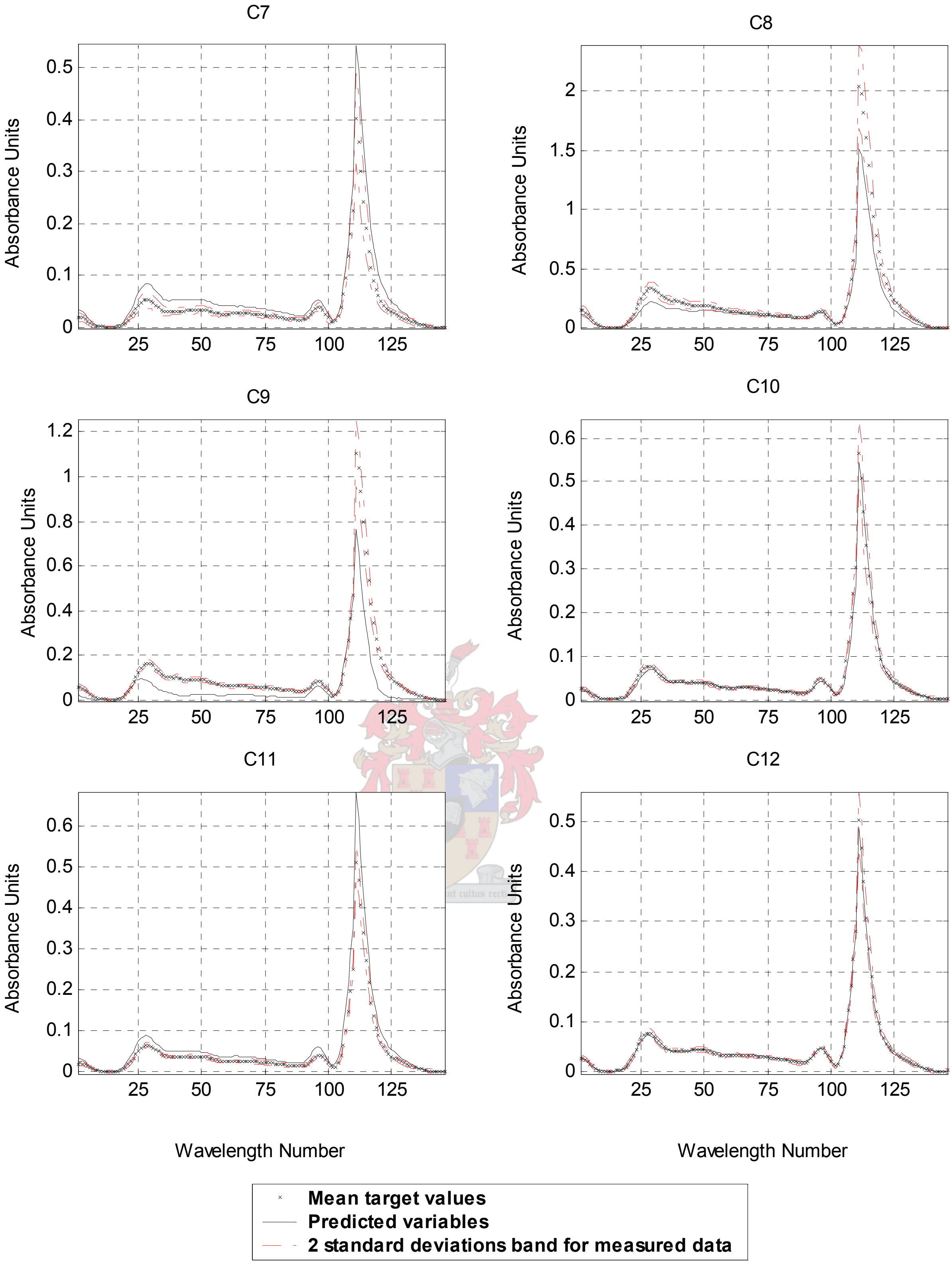

Fig. 4-10 The plume irradiance predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of linear PLS.

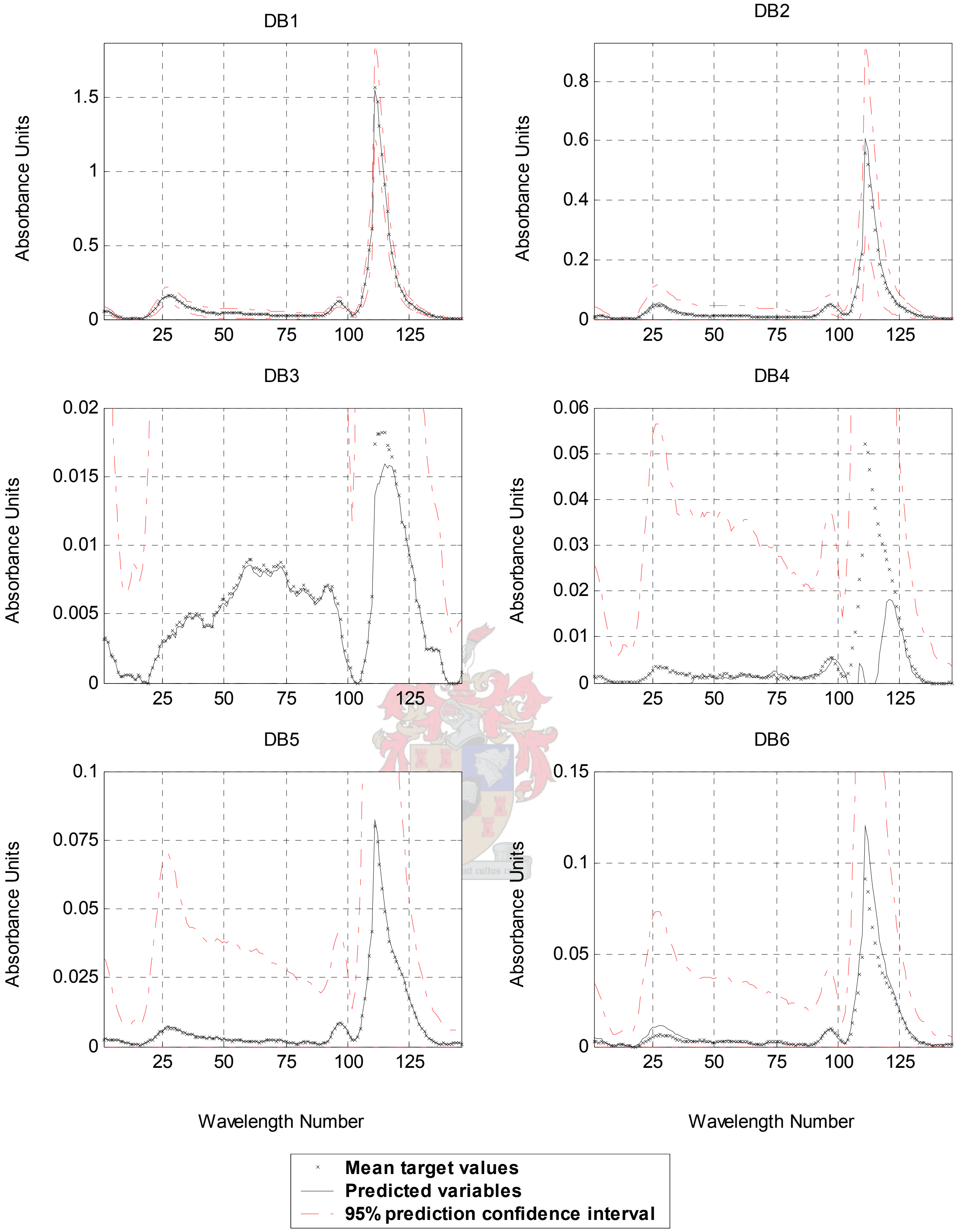

Fig. 4-11 The plume irradiance predictions for rocket motors DB1 to DB6 obtained for the overall linear PLS model.

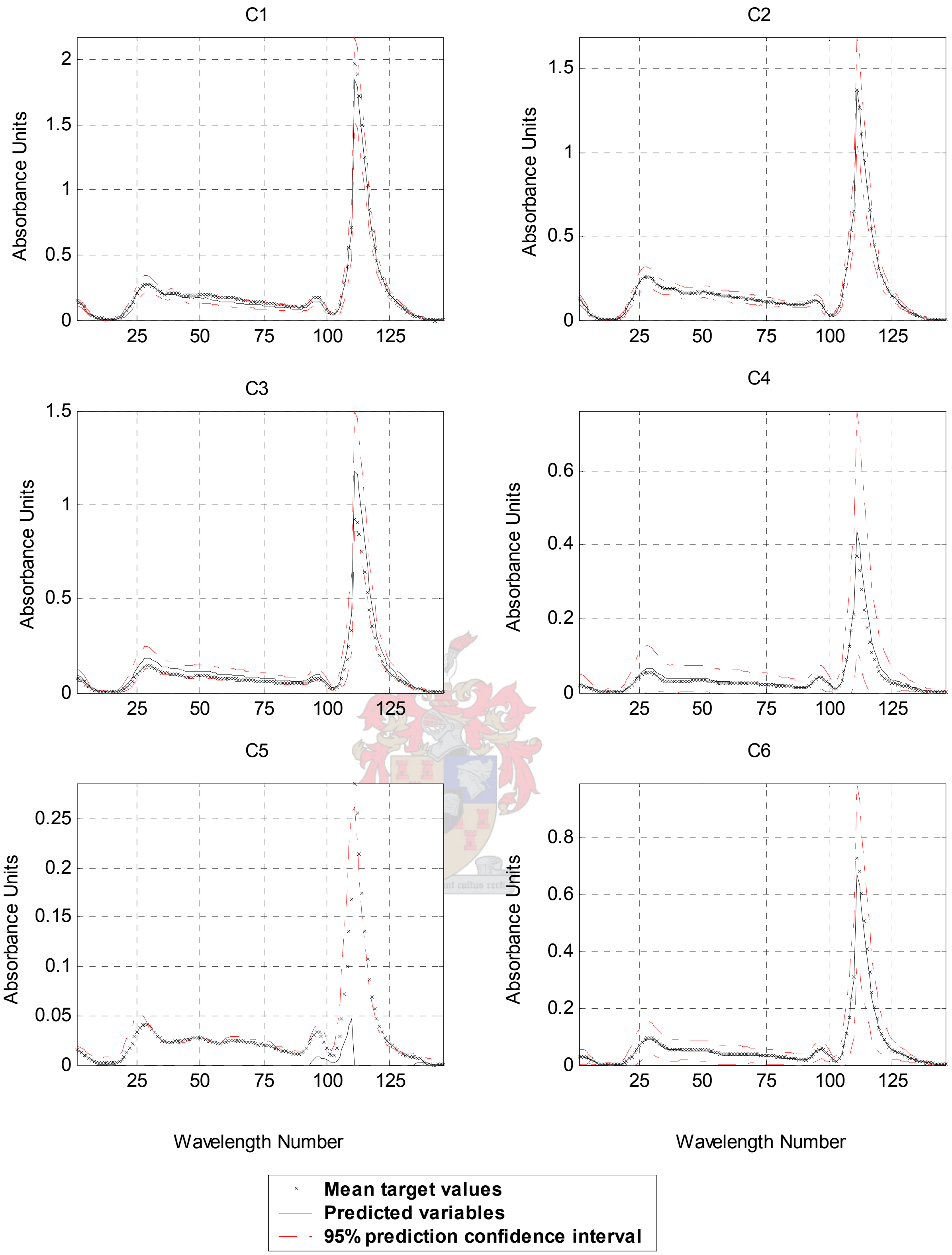

Fig. 4-12 The plume irradiance predictions for rocket motors C1 to C6 obtained for the overall linear PLS model.

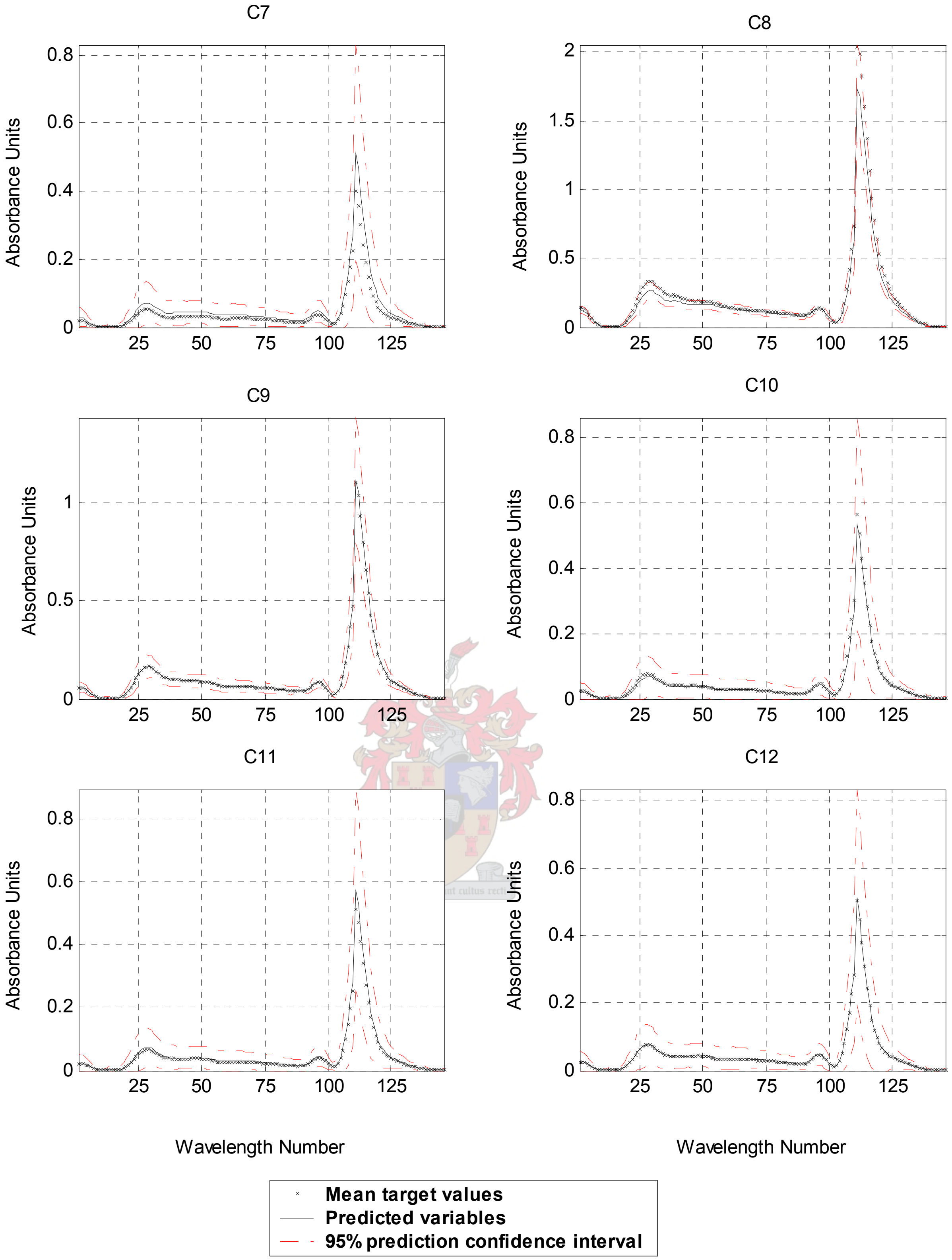

Fig. 4-13 The plume irradiance predictions for rocket motors C7 to C12 obtained for the overall linear PLS model.

4.2.2 Feed-forward Neural Network

A varying number of hidden nodes from 1 to 146, as illustrated in Fig. 4-14, have been used to find the optimum neural network architecture. Each cross-validation session was repeated ten times for each network complexity level in order to obtain an overall performance as shown in Fig. 4-14. Here a box-and-whisker plot diagram indicates the range of PRESS-values between the first and last quartile, including the median. The whiskers show the extent to which the rest of the data varies. The exact values of the repeated runs are shown in Table A-3. The SSEP-values for the overall model are presented in Table A-4 for 5 repeated runs. It is evident that the repeat runs produce very similar results and a box-and-whisker plot has therefore not been included here.

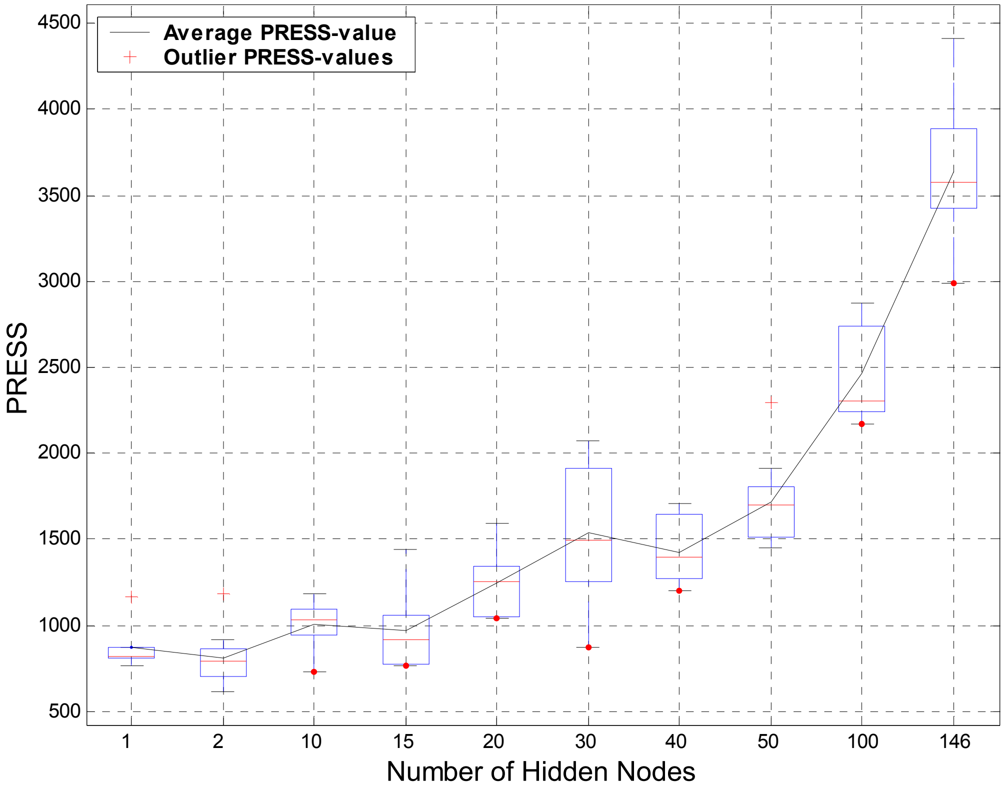

Fig. 4-14 A box-and-whisker plot of PRESS-values obtained from 10 cross-validation repetitions with increasing neural network complexity.

From Table A-3 the best run is chosen according to the lowest PRESS-value obtained in order to process further results from cross-validation. The best run for the overall model is chosen from Table A-4. The various neural network architectures were deliberately over-trained as shown in Fig. 4-15 for run 1 of the overall model. This was done to ensure that the performances of the various model complexities are based on the number of parameters and not so much on the extent of training. It also ensures that the sum-square error is minimised as far as possible. This method is addressed in the work done by Lawrence [1997] as discussed in section 3.1.3.

From the discussion above, by inspection of Fig. 4-14 and the average values in Table A-3 it is evident that two nodes in the hidden layer is the optimum neural network architecture. The best run corresponding to this result is run 2 for the PRESS-values and run1 for the SSEP-values. The statistics scores calculated for Table A-5 are thus obtained from these runs.

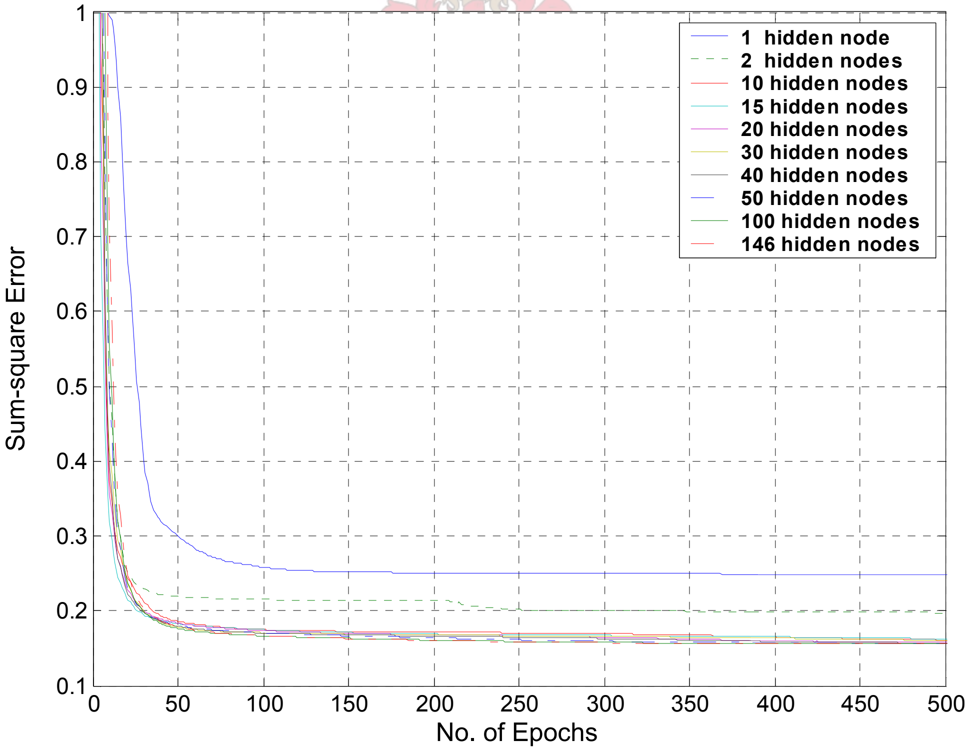

Fig. 4-15 An illustration of how the various neural network models have been over-trained by observing results from training run 1 of the overall model (evaluated on standardised values).

The explained Y-block variances for the chosen overall model are shown in Table A-5. There is a sharper increase (79.8% to 93.4%) in η^2 from a single node to double node model complexity (88.9% to 98.3% max η^2). From there the explained variance tapers out (94.8% to 95.1%) and points towards over-parameterisation. The neural network with 2 hidden nodes achieves higher η^2-values for the outputs of the overall model than the linear PLS model with 11 latent dimensions. It would therefore appear at this stage that the neural network is the better model, especially as it addresses the potential problem of non-linearities being present (discussed in section 4.2.1).

In Fig. 4-16 the squared correlations ($R^2_{CV} = (r_{y\hat{y}})^2$) between the predicted cross-validation model outputs and the target output variables are plotted. The region just before the spectrum resulting from IR absorbance due to water (wavelength numbers 20 and 45) shows the same expected zero correlations as in the case for the linear PLS model (Fig. 4-3). Here it is evident that the gradient from the region of zero correlation to the maximum at approximately wavelength number 40 is less steep than is the case for linear PLS. It therefore appears that linear PLS better predicts this region of the spectrum. The CO_2 absorbance region (wavelength numbers 100 and 140) also generally seems to be better predicted by the linear PLS model.

In Fig. 4-16 the highly irregular and 'noisy' R^2_{CV} –values for 10 hidden nodes and more can be attributed to over-parameterisation. In the work done by Roodt [1998] similar irregular, noisy predictions were obtained for the neural network consisting of 146 nodes in the hidden layer. The predictions were subsequently smoothened by using a 5-point moving average filter.

A comparison of the PLS model with 11 latent dimensions and the neural network with 2 hidden nodes shows that the linear PLS model appears to better predict unseen data. This is evident from the plots in Fig. 4-3 and Fig. 4-16, as well as the average R^2_{CV} –value of 0.417 for the neural network and that of 0.461 for linear PLS. This tendency is, however not reflected by the lower

PRESS-value obtained for the neural network. In linear regression the sum-square error is minimised to a global minimum and this corresponds to maximising the correlation between target and predicted outputs. In non-linear regression the SSE is not usually a global minimum and therefore may not correspond to a maximum correlation.

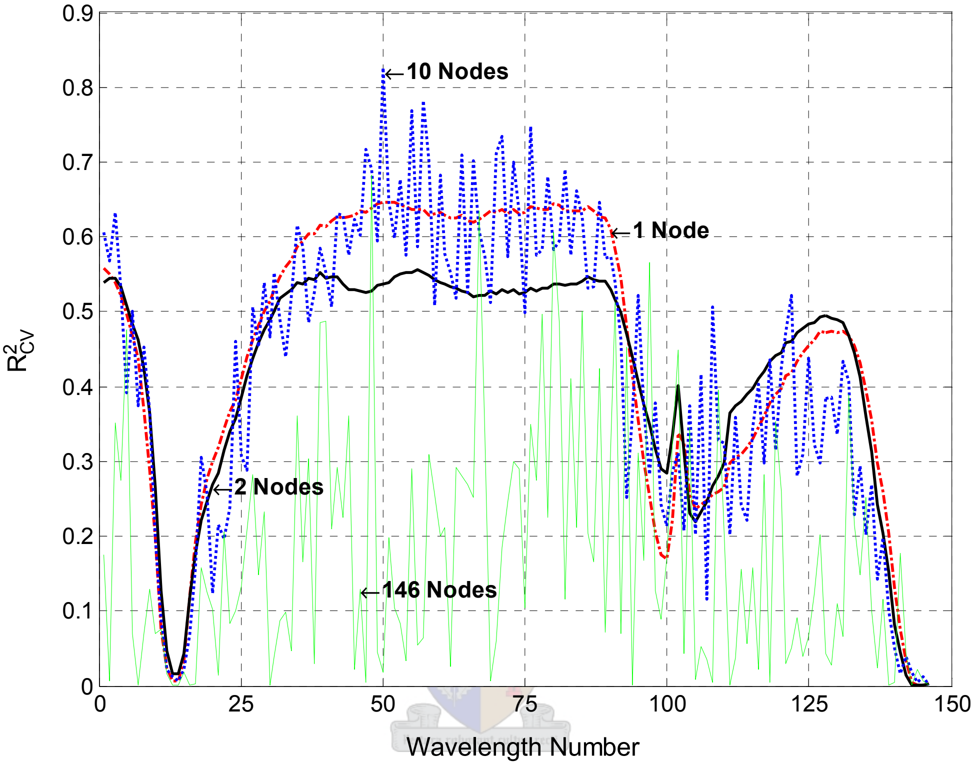

Fig. 4-16 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their targets.

From the correlation results above it therefore seems that, overall, a neural network with a single hidden node best predicts unseen data. The model with 2 hidden nodes however better predicts the CO_2-region of the spectrum. This would explain the lower PRESS-values for 2 hidden neurons compared to the single hidden node.

Fig. A-5 has also been included to illustrate that the $R^2_{cv(max)}$ –values show the same trends and that the maximum obtainable squared correlations exceed 0.6.

In Fig. 4-17 the R^2-values for all 146 output variables are plotted using the overall neural network model with 2 hidden nodes. Similar tendencies to the cross-validation data results can be observed. There are lower correlations in the CO_2 absorbance region and almost no correlations in the dip before the water-peak. These results are similar to those obtained for the linear PLS model, except for the reduced correlations obtained for the tail end of the spectrum.

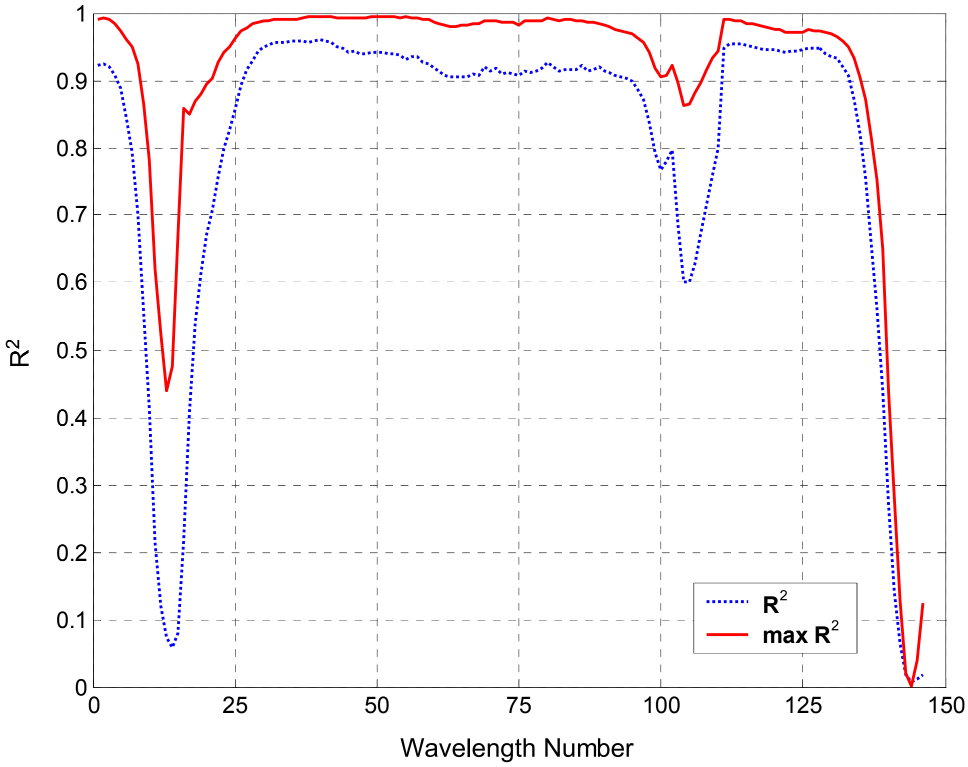

Fig. 4-17 The squared correlations between the 146 overall model output irradiance absorbance values and their targets for the neural network with 2 hidden nodes.

The pseudo degrees of freedom (see section 2.14) have been calculated for each of the 146 output variables using 2 hidden neurons for the overall model. As was discussed and shown in section 4.2.1 the pdf-values calculated and presented in Fig. 4-18 show deviations among output variables. The largest deviations occur at the CO_2-peak region, the region before the water-peak and at the tail end of the spectrum. Table A-5 shows that the average pdf-value for all 146 output variables is 12.91. This is slightly higher compared to the value obtained for linear PLS (12.42).

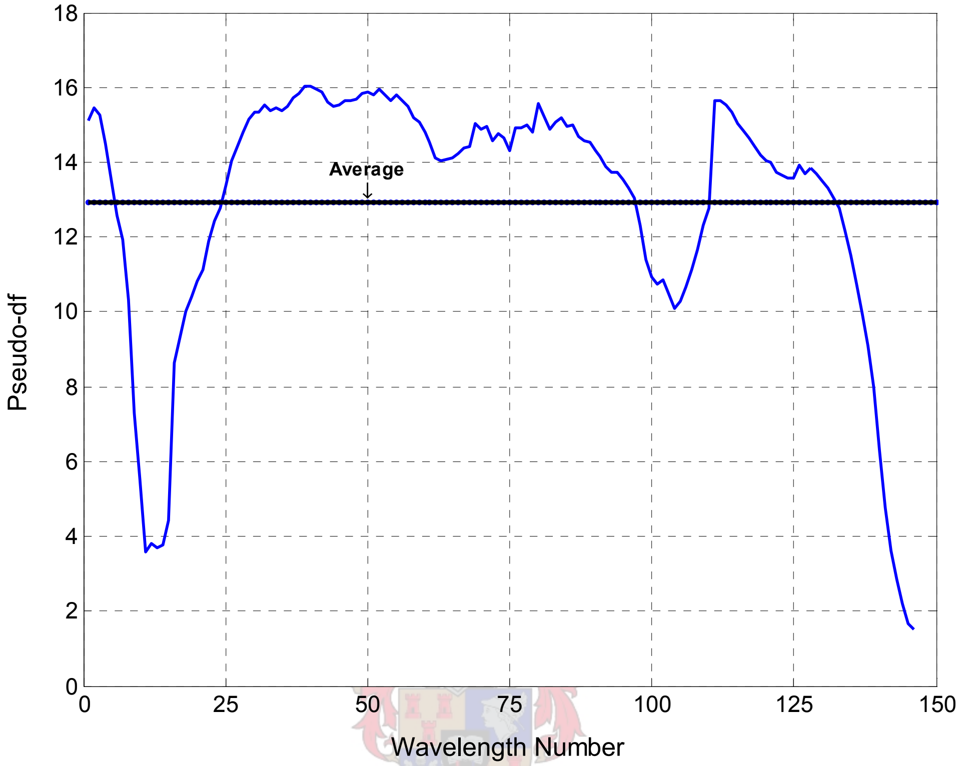

Fig. 4-18 The number of pseudo degrees of freedom (pdf) for each of the 146 output variables using a neural network model with 2 hidden nodes.

Table 4-3 shows the total number of neural network parameters cumulating with increasing number of hidden nodes per output variable. During training a neural network seeks to minimise the SSE summed over all output variables (SSEP). For this reason the total number of parameters over all output variables have been included in the table. This is a better reflection on the total complexity the training algorithm has to deal with and could indicate a possible model weakness.

In Fig. 4-19 to Fig. 4-20 the individual unseen validation set predictions from leave-one-out cross-validation for each rocket motor type are shown. The predictions are presented together with the mean target spectrum and the band of two standard deviations for the repeat measurements of the particular rocket motor. Confidence intervals were not calculated, as discussed in section 4.2.1.

Table 4-3 The total number of parameters cumulating with increased feed-forward neural network complexity.

No. of hidden nodes	1	2	10	15	20	30	40	50	100	146
Parameters per output variable cumulated	21	41	201	301	401	601	801	1001	2001	2921
Parameters for all output variables cumulated	311	476	1796	2621	3446	5096	6746	8396	16646	24236

The graphs (Fig. 4-19 to Fig. 4-20) show that the predictions for the irradiance spectra of the C-class rocket motors are fair. The predictions for C6, C7 and C11 are not desirable as their predictions fall slightly outside the 2 standard deviations band. The predictions for C5, C9, DB3, DB4, DB5 and DB1 are failures, as these predictions generally differ by more than 50% from the target values in most absorbance values across the spectral bands. The irradiance prediction for DB6 is not desirable, but is not necessarily a failure as it portrays the desired trend. It is especially encouraging to note that DB4 and DB6 tend to pick up the correct, lower order of magnitude in the irradiance of the DB-class. Rocket motor C5 again (see section 4.2.1) shows negative predictions, not shown here as realistically this means that the model effectively predicts zero transmittance. The prediction for the emission spectrum of DB1 is again too low which could be attributed to the model ‘seeing’ DB1 as a DB-class rocket motor. The above results show that 12 out of the 18 (66.7%) unseen predictions are predicted with a fair amount of accuracy.

In Fig. 4-22 to Fig. 4-23 the overall model predictions are presented together with their 95% confidence intervals (see section 2.15). In contrast to the linear PLS model the predictions look more promising for the DB-class rocket motor designs. The confidence intervals also appear much smaller, which points towards an increase in accuracy of the predictions made for the DB-class of rocket motor designs. This increase in precision confirms the tendency of the unseen DB-rocket motors to be predicted with an improved order of magnitude. Rocket motor C5 is better predicted using the neural network model.

If the predictions of C5 and DB3 are regarded as failures the prediction efficiency is 16 out of 18 (88.9%). This is much higher than that obtained for the unseen data. This supports the fact that there is not really enough data to build an adequate neural network model and that the data does not necessarily cover a comprehensively large input space obtainable from a design of experiments (DOE). The neural network model appears to be able to compensate for the DB-class rocket motor designs through its non-linearity.

In comparison to the predictions presented in the work by Roodt [1998] the predictions using 2 hidden nodes here are smooth. They show no definite signs of over-parameterisation and therefore do not require any post-processing by a moving average filter. An improvement in accuracy for the prediction of the DB-class rocket motor designs appears to go hand-in-hand with a loss in accuracy for the predictions of the C-class rocket motors. This is in agreement with the findings made by Roodt [1998]. Neural networks can predict training data well, but usually require large amounts of available data in order to build a reliable model. For this reason it may be necessary to further explore possible more robust non-linear methods for building a model.

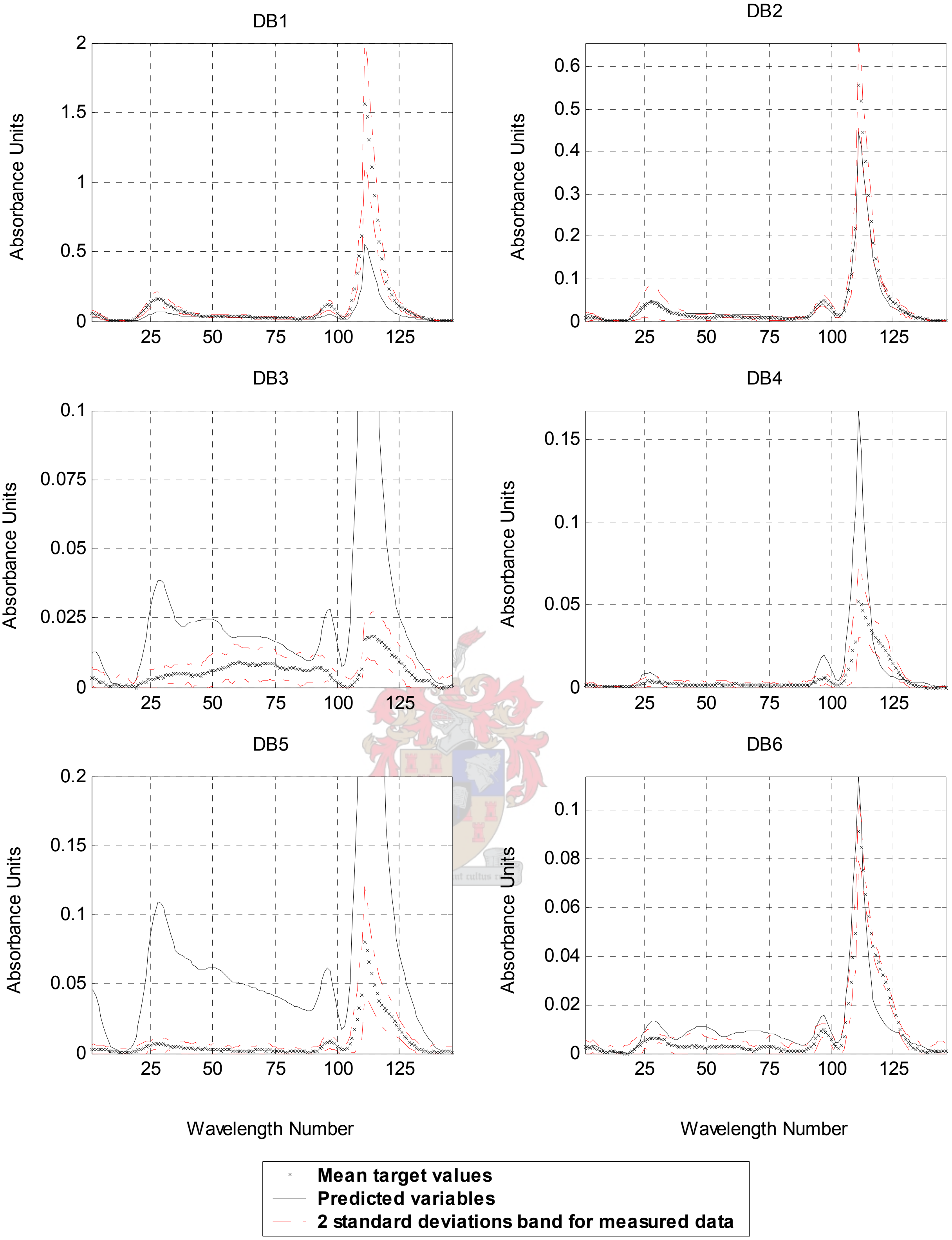

Fig. 4-19 The plume irradiance predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of a neural network with 2 hidden nodes.

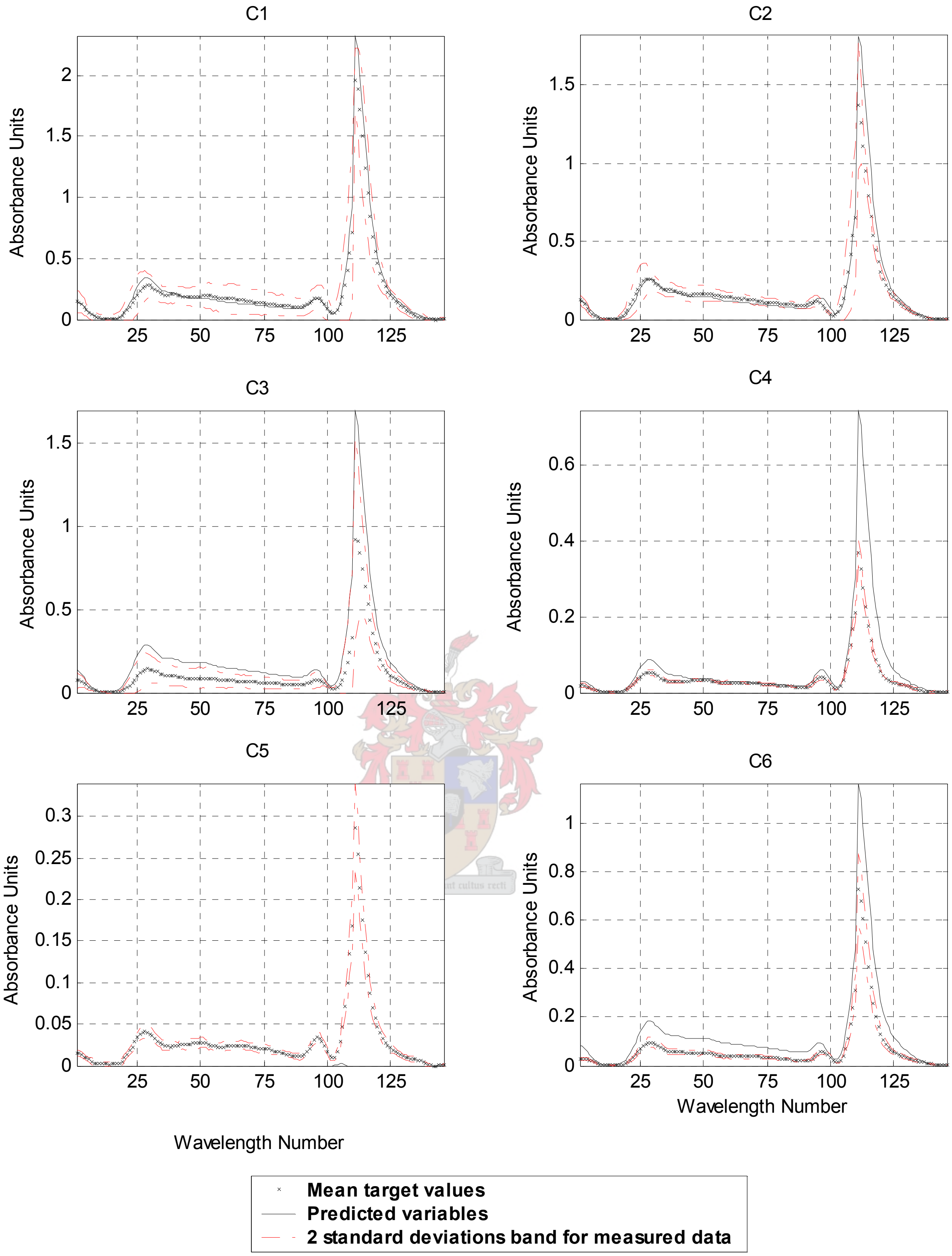

Fig. 4-20 The plume irradiance predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of a neural network with 2 hidden nodes.

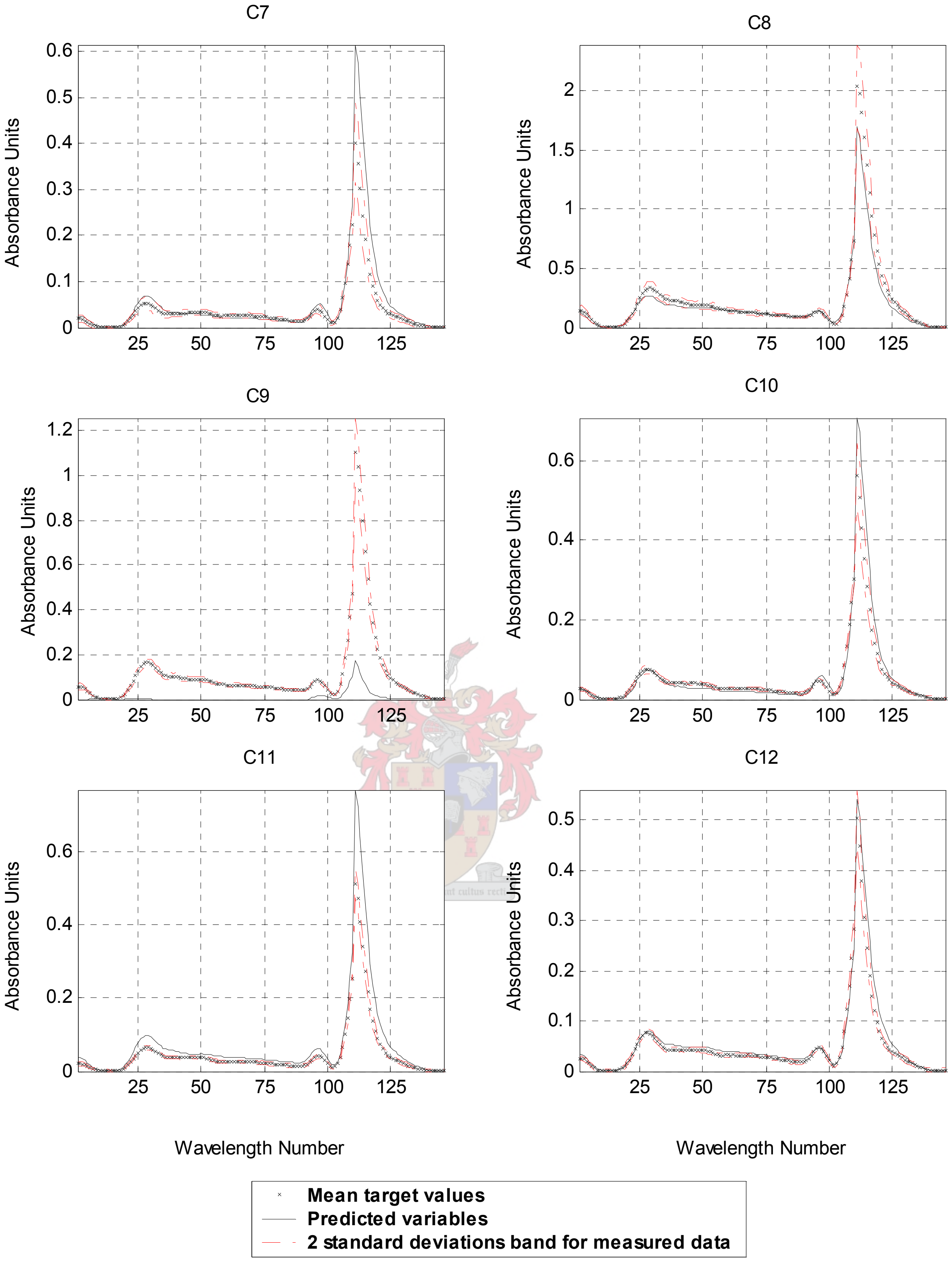

Fig. 4-21 The plume irradiance predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of a neural network with 2 hidden nodes.

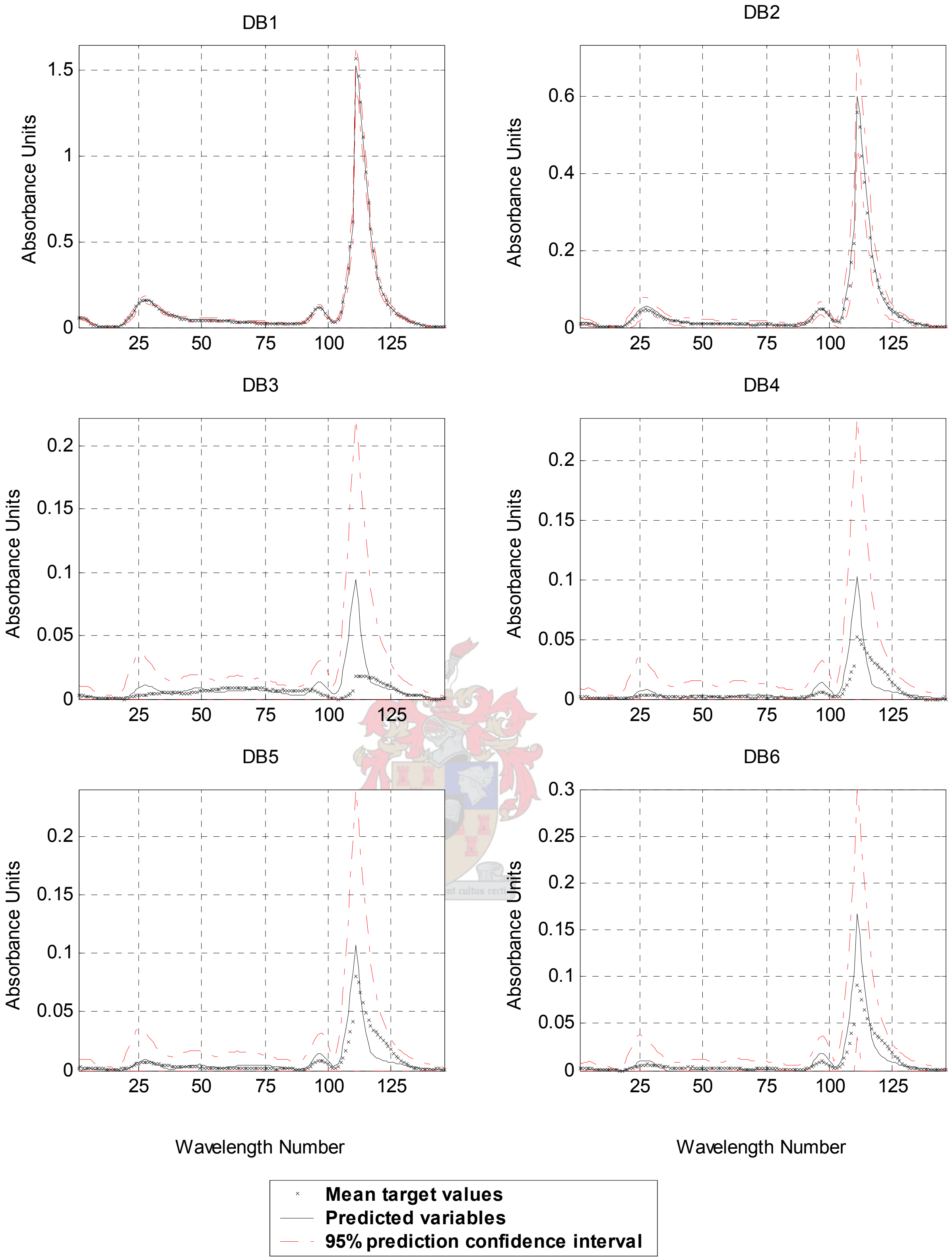

Fig. 4-22 The plume irradiance predictions for rocket motors DB1 to DB6 obtained for the overall neural network model (2 hidden nodes).

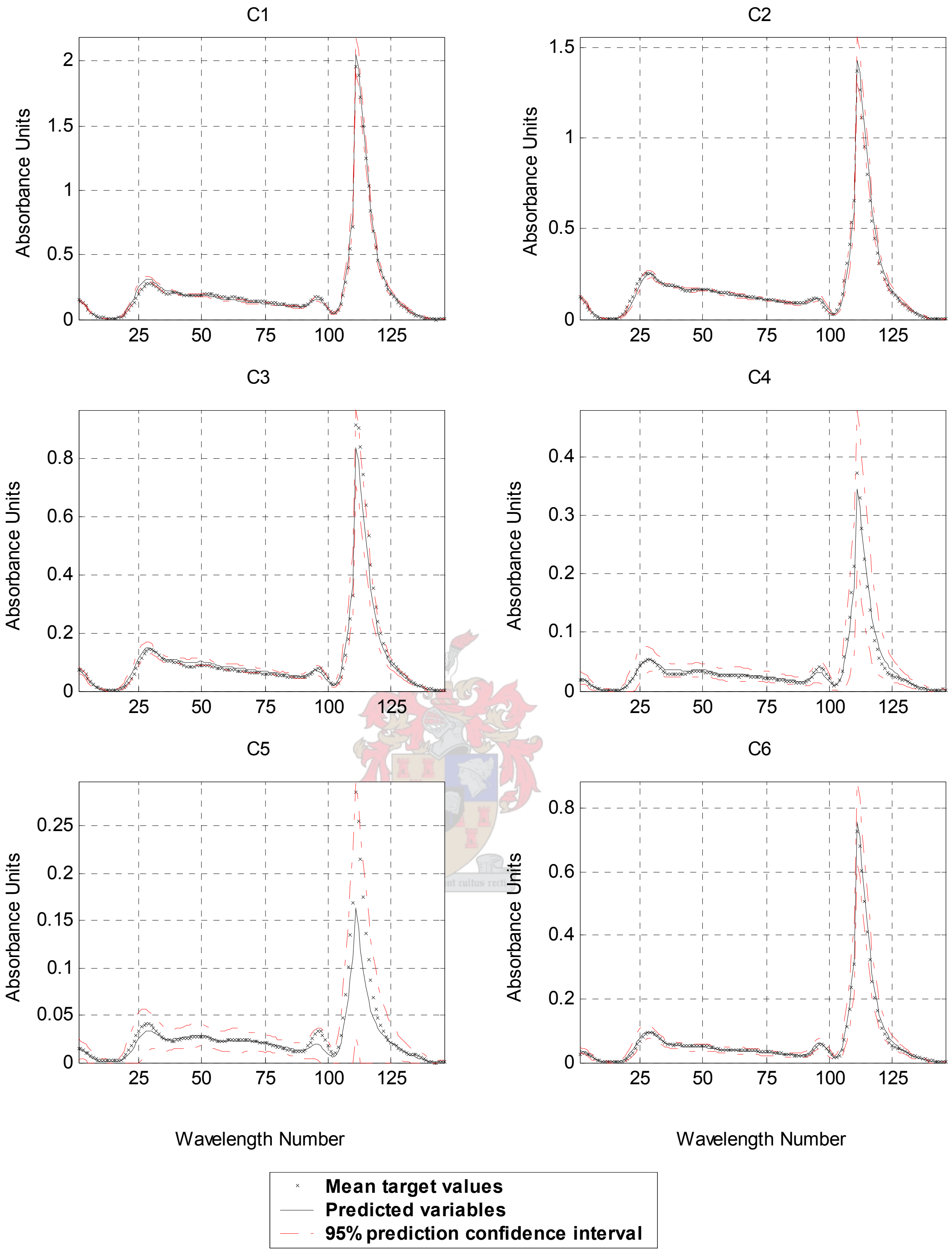

Fig. 4-23 The plume irradiance predictions for rocket motors C1 to C6 obtained for the overall neural network model (2 hidden nodes).

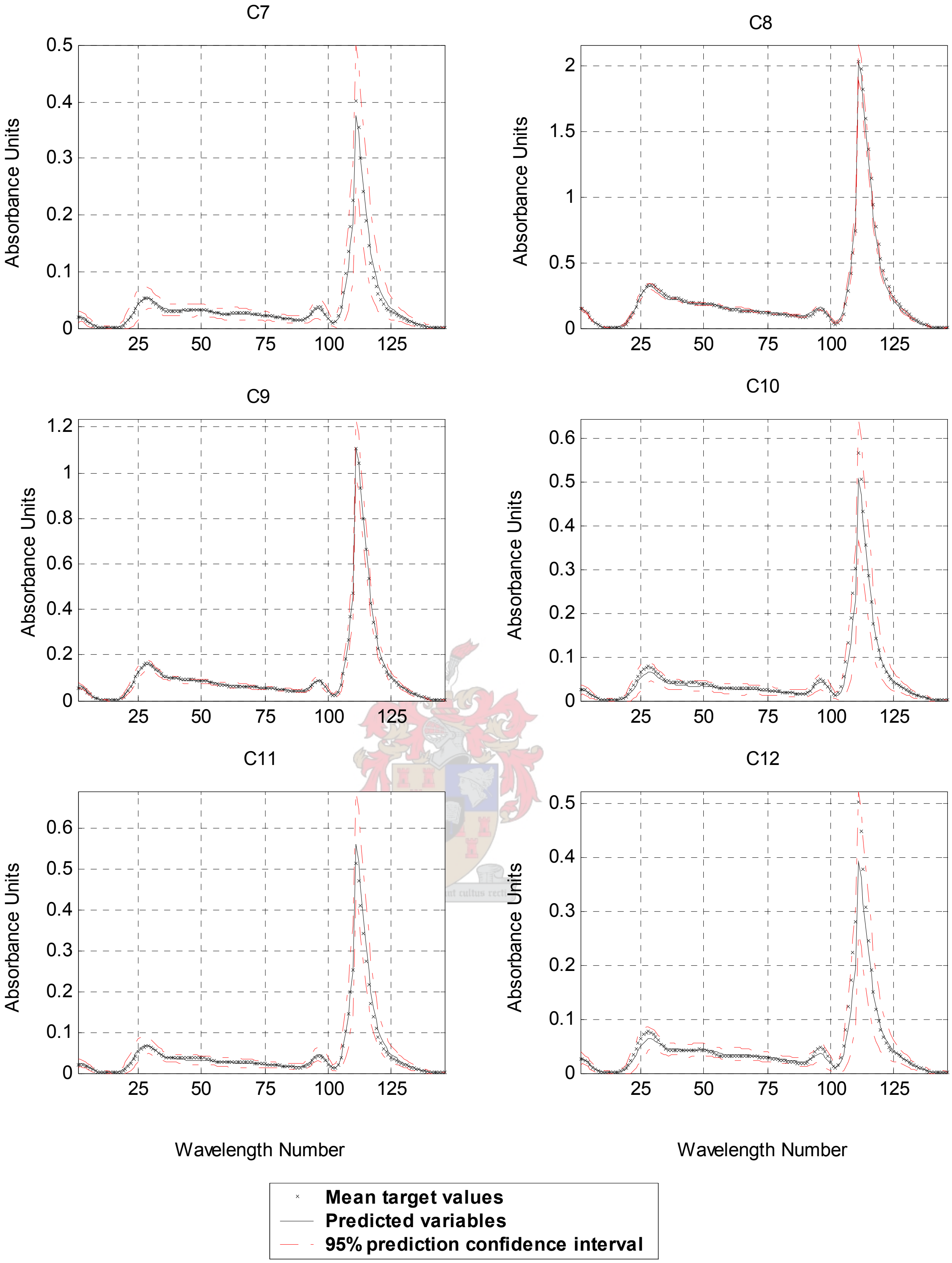

Fig. 4-24 The plume irradiance predictions for rocket motors C7 to C12 obtained for the overall neural network model (2 hidden nodes).

4.2.3 Neural Network PLS

The first attempts at building a neural network PLS model were done using the error-based sigmoidal neural network (EBNNPLS) algorithm (see section 2.13.2). The inner neural network models were trained using 80 epochs for the Levenberg-Marquardt algorithm, which has the ability to converge quickly. For the single input and single output (SISO) models this is sufficient to ensure over-training. The performances of the models are therefore again dependent on parameterisation and not so much on the extent of the iterative training of the models.

Following the discussion in section 4.1.2 and the subsequent modification to the PLS cross-validation algorithm the results of the original cross-validation algorithm are included in Table A-6. It is evident that there is no characteristic ‘dip’ in the PRESS-value. Although the lowest PRESS-value at LD 18 is impressive it simply points to fact that the model has been optimised to compensate for the left out data point. This no longer renders the left out point as unseen. The number of hidden nodes per inner model for each latent dimension cannot be shown in the table, as there are different sets of varying numbers of hidden nodes per LD for each cross-validation session.

Table 4-4 shows the modified cross-validation (PRESS) and overall model (SSEP) results for single runs using the EBNNPLS algorithm. The PRESS-value does go through the characteristic ‘dip’ as expected and a set of hidden nodes used per inner model is obtained to train the overall model. From the large PRESS-values compared to those obtained for linear PLS and the feed-forward neural network models it is clear that there is no reason to further evaluate the EBNNPLS results. The minimum PRESS-value for EBNNPLS is achieved at LD 2 already and the SSEP-value drops sharply for the first three latent dimensions. This shows how efficient the algorithm is at finding a global optimum. However, due to the high degree of non-linearity in extracting latent dimensions the EBNNPLS model seems to over-train and the predictions for unseen data are poor.

Table 4-4 Modified cross-validation and overall model-building results for single runs using EBNNPLS.

LD	PRESS	SSEP	Hidden nodes per inner model
1	2047.8	160.33	6
2	1597.6	52.69	4
3	1762.8	46.90	5
4	1988.0	44.99	4
5	1905.5	44.01	6
6	2272.2	42.89	3
7	2265.4	42.01	2
8	2189.4	41.58	4
9	2239.7	41.35	3
10	2233.9	40.61	4
11	2257.5	40.39	10
12	2282.0	40.28	1
13	2289.4	40.18	3
14	2292.1	40.09	5
15	2221.1	39.99	5
16	2207.8	39.98	13
17	2230.5	39.87	7
18	2223.5	39.38	14

Each modified cross-validation session was repeated ten times for each latent dimension using the NNPLS algorithm in order to obtain the box-and-whisker plot in Fig. 4-25. For each run a certain number of hidden nodes per latent dimension inner model are obtained and presented in Table A-9. The average PRESS-values in Table A-8 and Fig. 4-25 show that the optimum predictions are achieved at LD 11. From Table A-8 it can further be seen that the best results are obtained from runs 3, 4 and 10 and that the minimum PRESS-value here is lower than the linear PLS and feed-forward neural network optimums. It is interesting to note that these three runs have the exact same set of PRESS-values and therefore the same set of hidden nodes for each latent dimension. It appears that if the initial conditions for the neural network training of LD 1 are the same or similar, such that 6 hidden nodes are chosen, the rest of the algorithm follows the same path.

The hidden node configuration corresponding to runs 3, 4 and 10 is chosen to build the overall NNPLS model. The training of the overall model was repeated five times and the box-and-whisker results in Fig. 4-26 show that there is little variation in the SSEP-values for the higher latent dimensions. From Table A-7 run 3 is chosen as the best overall model for further evaluation.

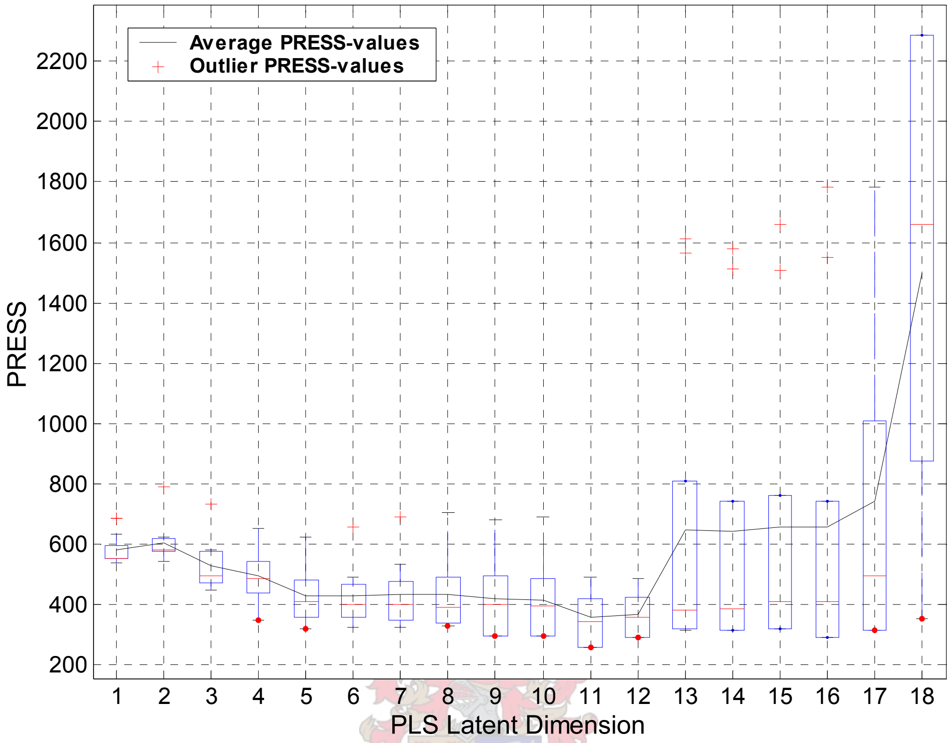

Fig. 4-25 A box-and-whisker plot of PRESS-values obtained for NNPLS from 10 cross-validation repetitions.

Inspection of Fig. 4-27 shows the expected incremental increases in explained variances for both the input and output data . In comparison to the linear PLS model (Fig. 4-2) the curves appear smoother and the explained variance of the Y-block is much higher at LD 1. The first latent dimension already captures 75.6% (maximum 83.3%) of the Y-block data (see Table A-10). The explained variance of the X-block follows a similar trend compared to that of linear PLS. This shows that the X-block latent structure may be close to that of linear PLS. The non-linearity of the inner model allows for more information to be extracted in the earlier latent dimensions. The maximum explained Y-block variance does not quite reach 100% at LD 18 due to the non-linear nature of the training algorithm.

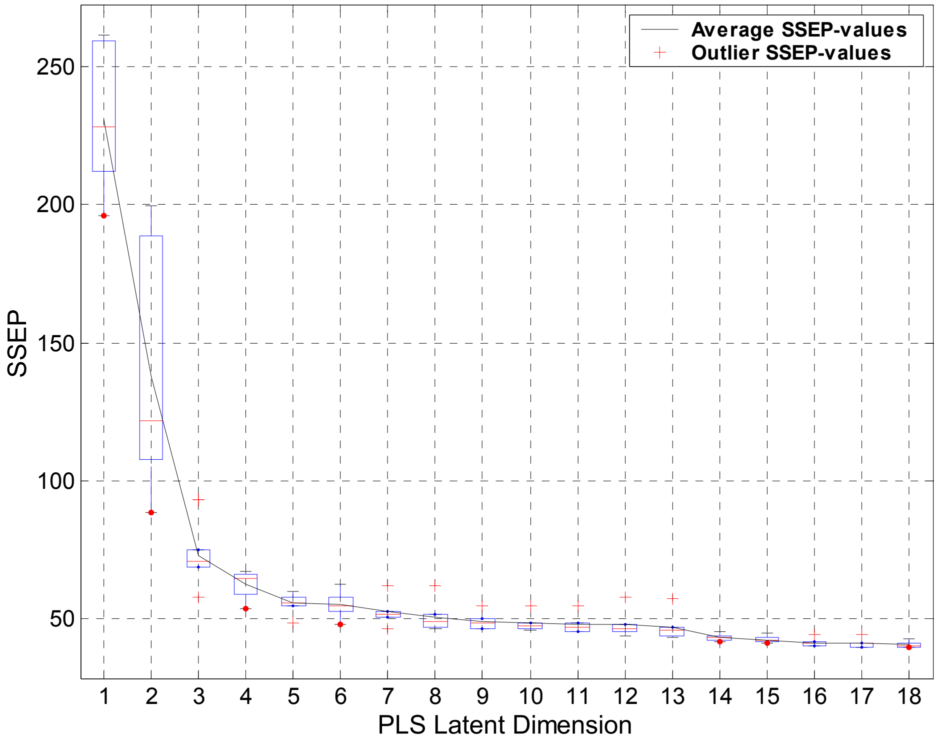

Fig. 4-26 A box-and-whisker plot of SSEP-values obtained for the overall NNPLS model from 5 repetitions.

Due to the expected 'dipping' shape of the PRESS-curve there are no discrepancies as to where the optimum latent dimension lies. From Fig. 4-27 a NNPLS model with 11 latent dimensions or even less seems feasible.

In Fig. 4-28 the squared correlations ($R^2_{CV} = (r_{y\hat{y}})^2$) between the predicted cross-validation model outputs and the target output variables are plotted. The same tendencies for prediction accuracies in the H_2O and CO_2 absorbance regions are obtained as for linear PLS and the feed-forward neural network. The obvious differences, however are the significantly higher R^2_{CV} –values achieved by NNPLS modelling. A few of the R^2_{CV} –values are even above 0.8 in the H_2O absorbance region and $R^2_{CV(max)}$ exceeds 0.9 at places (see Fig. A-6). The predictions for LD 10 appear to be better than those of LD 11 in the region between 20 and 95 wavelength numbers. LD 11 has the better correlations in the CO_2 absorbance region. This would explain the lower PRESS-values for

LD 11, as the absorbance units in the CO_2-region are large compared to the rest of the spectrum. Due to the dominant feature of the CO_2-peak it is therefore better to build a model using 11 latent dimensions.

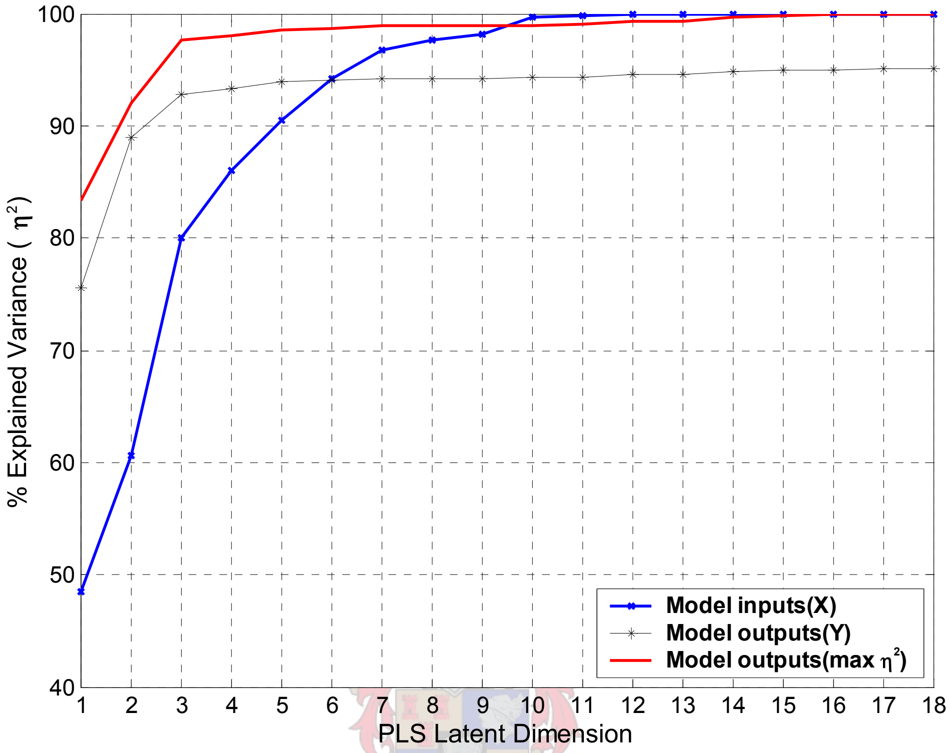

Fig. 4-27 The characteristic curves of increasing explained variances for the overall NNPLS model (built on all 18 rocket motors).

The average squared correlation for cross-validation using 11 latent dimensions is significantly higher at 0.63 (0.75 maximum) as seen in Table A-10. The highest average values however lie at LD 9 and LD 10. This can only be attributed to the fact that the broad band from the H_2O absorbance region to before the CO_2 absorbance region has high R^2_{CV} –values, which influence the averages.

In Fig. 4-29 the R^2-values for all 146 output variables are plotted using the overall model with 11 latent dimensions. The shapes of the curves are similar to those of the previous models and bear similar, less dramatic tendencies as observed for cross-validation. In comparison to linear PLS (Fig. 4-4) the correlations in the pre-H_2O and CO_2 absorbance regions are lower. The average R^2-value at LD 11 is 0.83 (maximum 0.95), which is lower than the value of 0.88 (maximum 0.96) obtained for linear PLS.

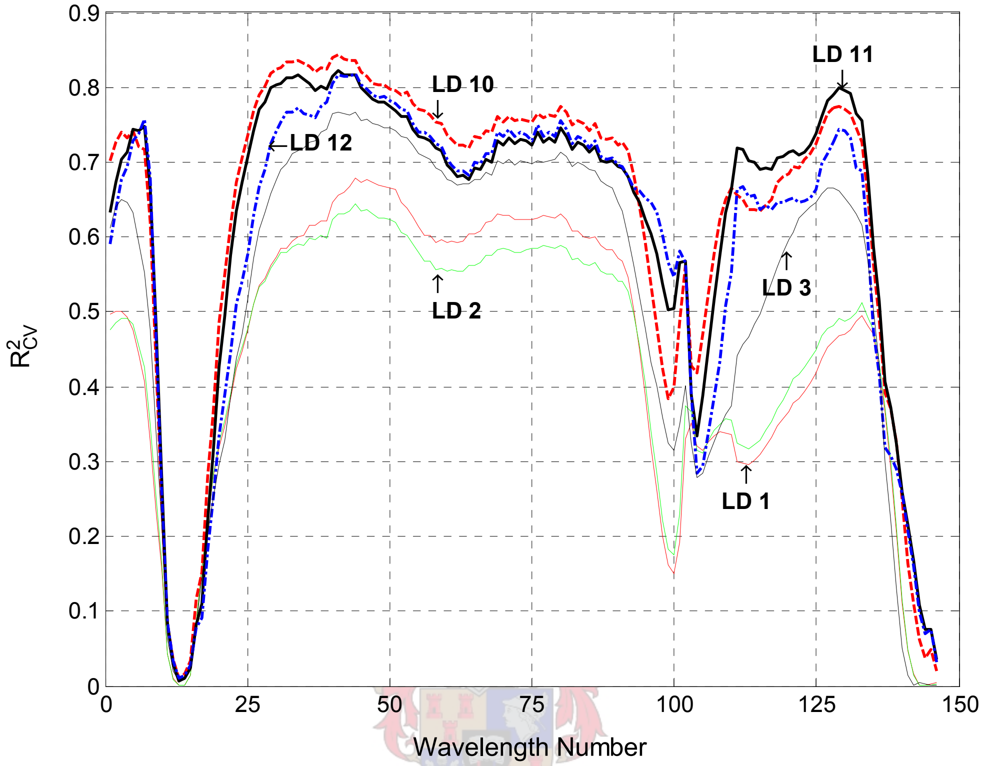

Fig. 4-28 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their targets for NNPLS.

The pseudo degrees of freedom (see section 2.14) have been calculated for each of the 146 output variables using 11 latent dimensions for the overall NNPLS model. Fig. 4-30 shows how the pdf-values vary for the different output variables. The deviations appear to be more random and not mainly at the absorbance regions for CO_2 and before H_2O as is the case for linear PLS. The average pdf value of 12.33 for LD 11 compares well with the averages obtained for optimum overall linear PLS (12.42) and feed-forward neural network (12.91) models.

The average pdf-values (see Table A-10) for each latent dimension are plotted together with the df-values of an equivalent multiple linear regression model with increasing number of input variables in Fig. 4-31. In comparison to linear PLS (Fig. 4-6) the model degrees of freedom appear higher at the earlier latent dimensions. This corresponds to the larger explained variance achieved at the lower latent dimensions. A similar 'dipping' trend towards LD 12 is observed as in linear PLS. This is an indication that the increase in model complexity no longer contributes significantly to the overall input space leverage. The higher latent dimensions in fact decrease the overall input space leverage, which is indicative of over-parameterisation.

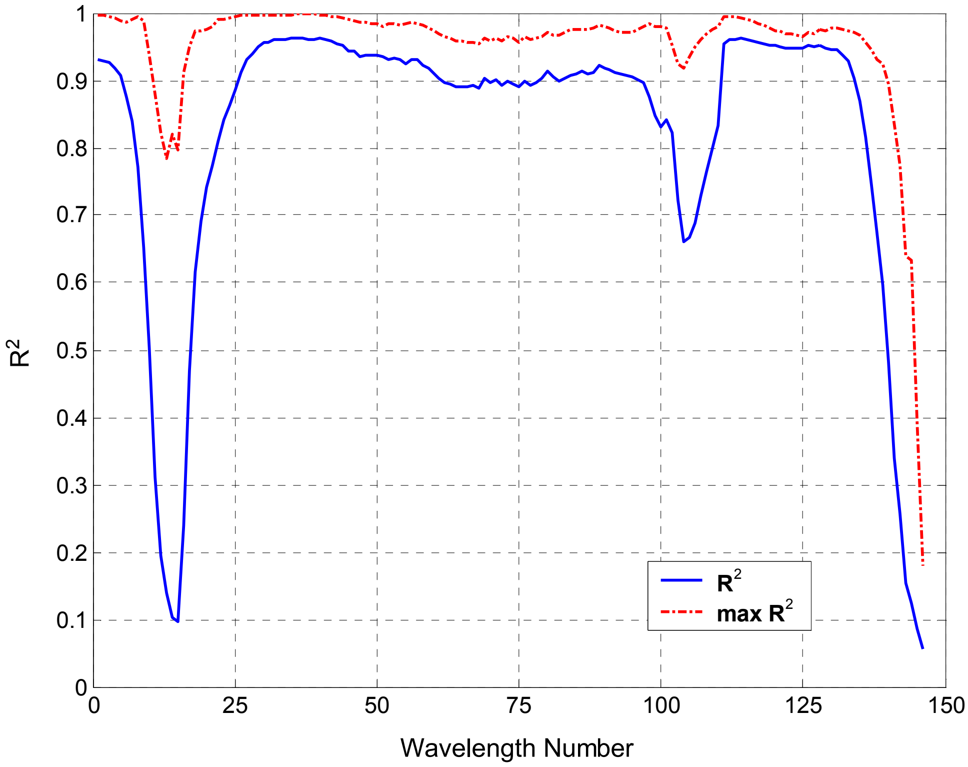

Fig. 4-29 The squared correlations between the 146 overall model output irradiance absorbance values and their targets for NNPLS up to LD 11.

Table 4-5 shows the total number of NNPLS parameters per output variable cumulating with increasing number of latent dimensions. These parameters include the in- and output loadings as discussed in section 4.2.1. The cumulated numbers of inner model parameters are also presented here. This is to distinguish the inner model parameters from the in- and output loadings, which are also counted as parameters. For LD 11 the 98 inner model parameters are significantly more than the 11 for the linear PLS model. Yet the statistics scores generally appear to be better for the NNPLS model. In comparison to the feed-forward neural network model with 2 hidden nodes (41 parameters) the overall NNPLS model with LD 11 (307 parameters) appears to have many more parameters per output variable. However, if one compares the total number of parameters used for training the whole neural network (476 parameters) to those used to train the NNPLS model the NNPLS model has fewer parameters. Due to the nature of the different training algorithms it is difficult to compare the different models on a parameter basis. For this reason the calculation of the pseudo degrees of freedom to determine the input space leverage is a better choice.

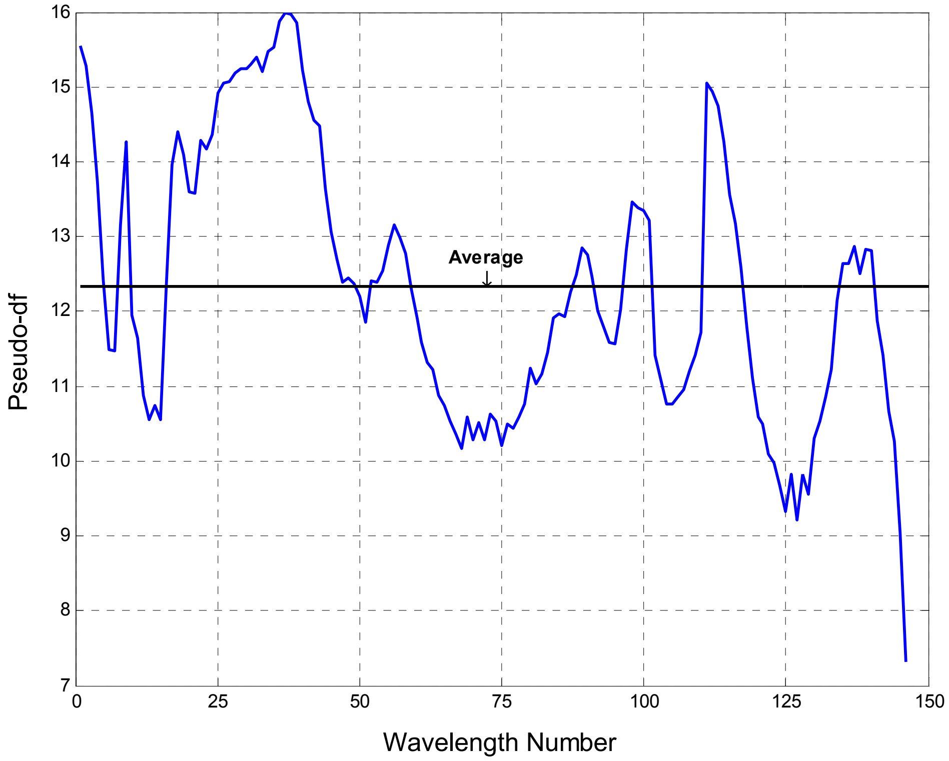

Fig. 4-30 The number of pseudo degrees of freedom (pdf) for each of the 146 output variables using the overall NNPLS model with 11 latent dimensions.

In Fig. 4-32 to Fig. 4-34 the individual unseen validation set predictions from leave-one-out cross-validation for each rocket motor type are shown. The predictions are presented together with the mean target spectrum and the band of two standard deviations for the repeat measurements of the particular rocket motor. Confidence intervals are not calculated, as discussed in section 4.2.1.

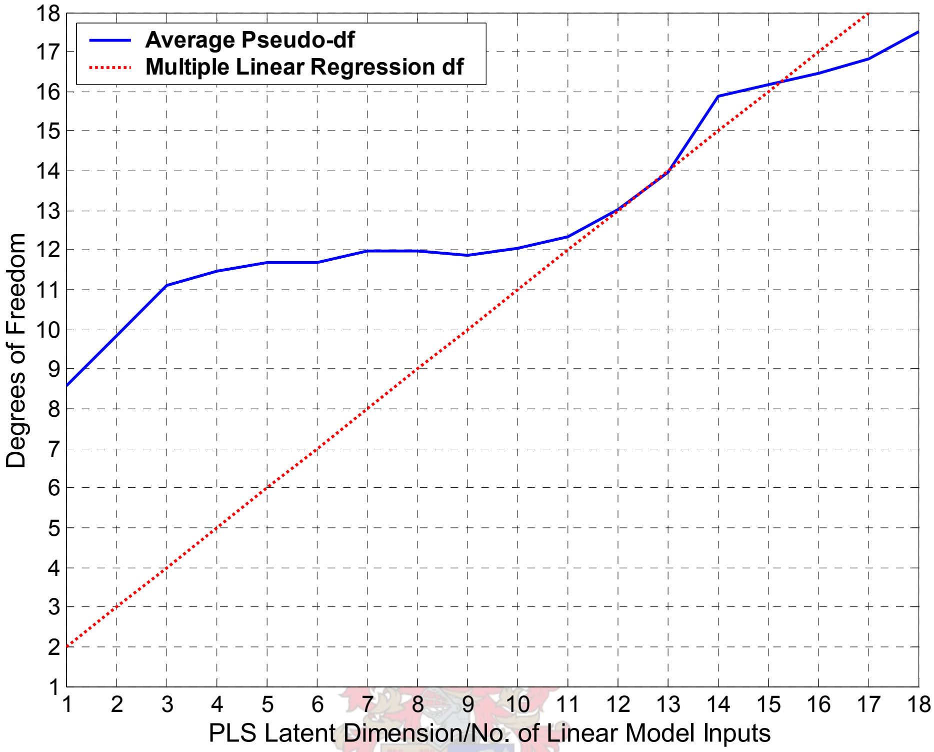

Fig. 4-31 The average pdf over all output variables for the overall NNPLS model with increasing number of latent dimensions compared to multiple regression df.

The graphs (Fig. 4-32 to Fig. 4-34) show good predictions for the C-class rocket motors. The predictions for DB1 and DB3 to DB6 can be regarded as failures although the tendency for the models to predict the correct order of magnitude for the DB-class rocket motors is there. A vast improvement is that rocket motor C5 is correctly predicted. The prediction for the emission spectrum of DB1 is again too low. The above results show that 13 out of the 18 (72.2%) unseen predictions are predicted with a fair amount of accuracy.

In Fig. 4-35 to Fig. 4-37 the overall model predictions are presented together with their 95% confidence intervals (see section 2.15). The NNPLS model is able to better predict the DB-class rocket motor spectra than the feed-forward neural network. The confidence intervals are smaller compared to the optimum linear PLS model, which supports the idea that there is an increase in accuracy of the spectral predictions made for the DB-class of rocket motor designs. This increase in precision confirms the tendency of the unseen DB-rocket motors to be predicted with an improved order of magnitude.

Table 4-5 The total number of parameters per inner model and output variable for the optimum NNPLS model.

LD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Total number of inner model paramters	19	32	45	58	62	66	76	83	90	94	98	102	139	167	204	235	266	270
Total number of parameters per output variable*	38	70	102	134	157	180	209	235	261	284	307	330	386	433	489	539	589	612

If the prediction for DB4 is regarded as a failure the prediction efficiency is 17 out of 18 (94.4%). The overall NNPLS model is able to compensate for the DB-class rocket motor designs through its non-linearity but the unseen predictions remain poor. This can be attributed to the lack of data and the fact that the data does not necessarily cover a comprehensively large input space obtainable from a design of experiments (DOE). The spectral predictive ability of the C-class rocket motor designs does not appear to be significantly compensated for by an increase in predictive ability for the DB-class. This is the case for the feed-forward neural network.

The fact that the NNPLS model performs much better than the EBNNPLS can only be attributed to the fact that the latent dimensions are more linear in nature. In section 2.13.2 it is mentioned and referenced to external sources that the NNPLS algorithm is better suited for building models on data with slight non-linear tendencies. The fact that the linear PLS models achieves such accurate predictions on unseen data for the C-class rocket motors is indicative of linearities in the input-output relationships.

The target and predicted output scores versus the input scores for each inner NNPLS overall model up to LD 12 have been plotted in Fig. A-7 and Fig. A-8. The target in- and output scores of the first latent dimension are, as expected, identical to those of the overall linear PLS model. From Fig. A-7 it is evident that the functional relationship is the most non-linear in LD 1. This non-linearity manages to capture 46.3% more explained variance of the outputs compared to linear PLS. From LD 2 the scores calculated for NNPLS differ from using linear PLS due to the systematic subtraction of the latent variable space from the overall data during PLS training. This explains why the initial conditions in the first latent dimension are important in determining the structures for the rest of the latent dimensions. It also explains why it is possible for 3 repeat training runs to be identical. It appears that the inner models for each latent become more linear in nature with increasing latent dimension.

In section 2.13.2 it is shown that there are 3H+1 parameters per neural network inner model. This means that if there are more than 5 hidden nodes in the inner model there are more parameters than the 18 independent data points. In Table A-9 it is shown that for run 10 only LD 1 uses more than 5 hidden nodes in the inner model. It is discussed in section 3.1.3 that it is not necessarily crucial for non-linear models to have more data points than parameters, however it is desired. This may be a contributing factor to why the NNPLS model performs better than the feed-forward neural network.

The reductions in input dimension achieved by linear PLS and NNPLS make it possible to build more robust models. The dimensional problem as discussed in section 3.1.3 can be overcome. Alternatively stated, if the data size is fixed the only remaining option is to manage the dimensions of the input space.

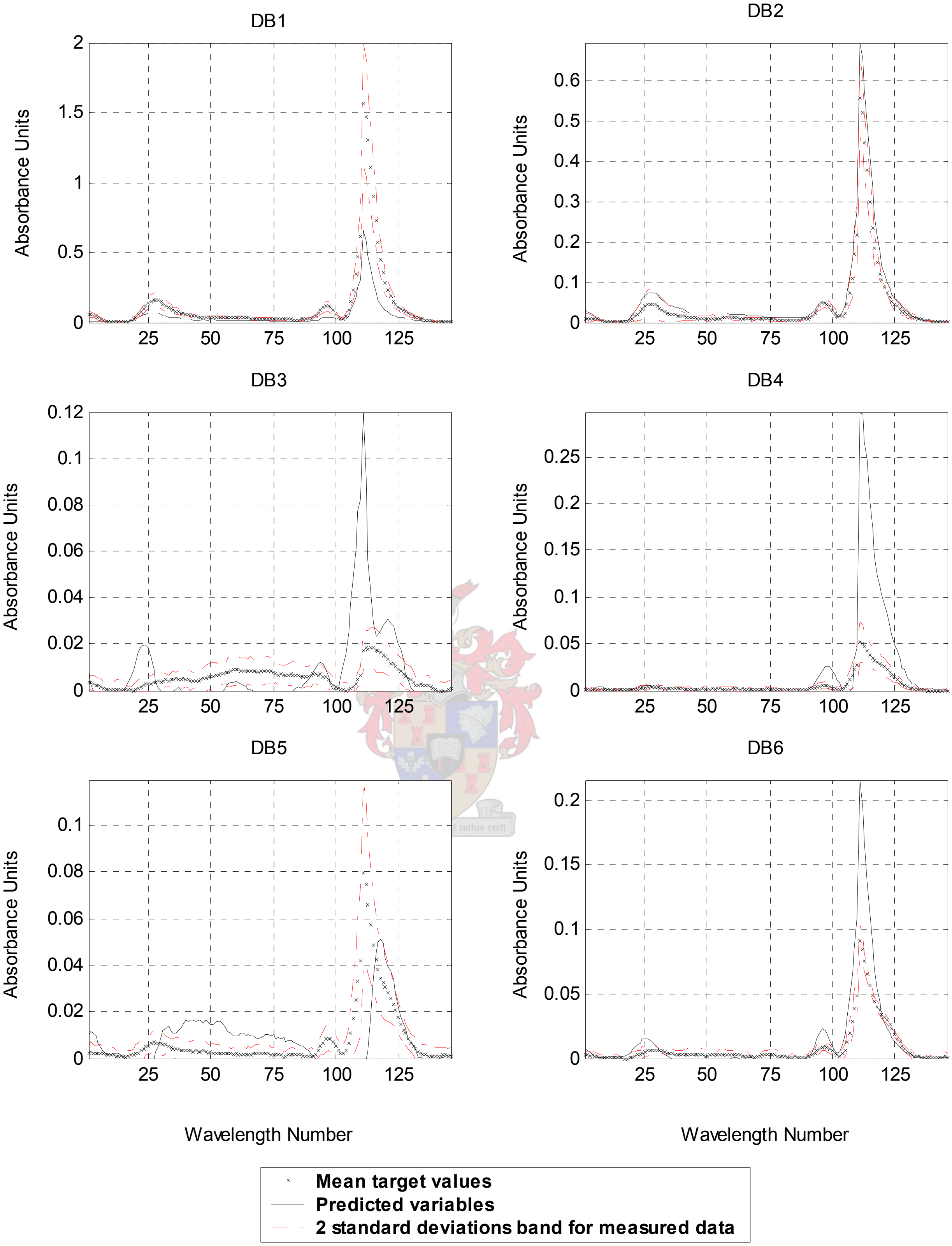

Fig. 4-32 The plume irradiance predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of NNPLS with 11 latent dimensions.

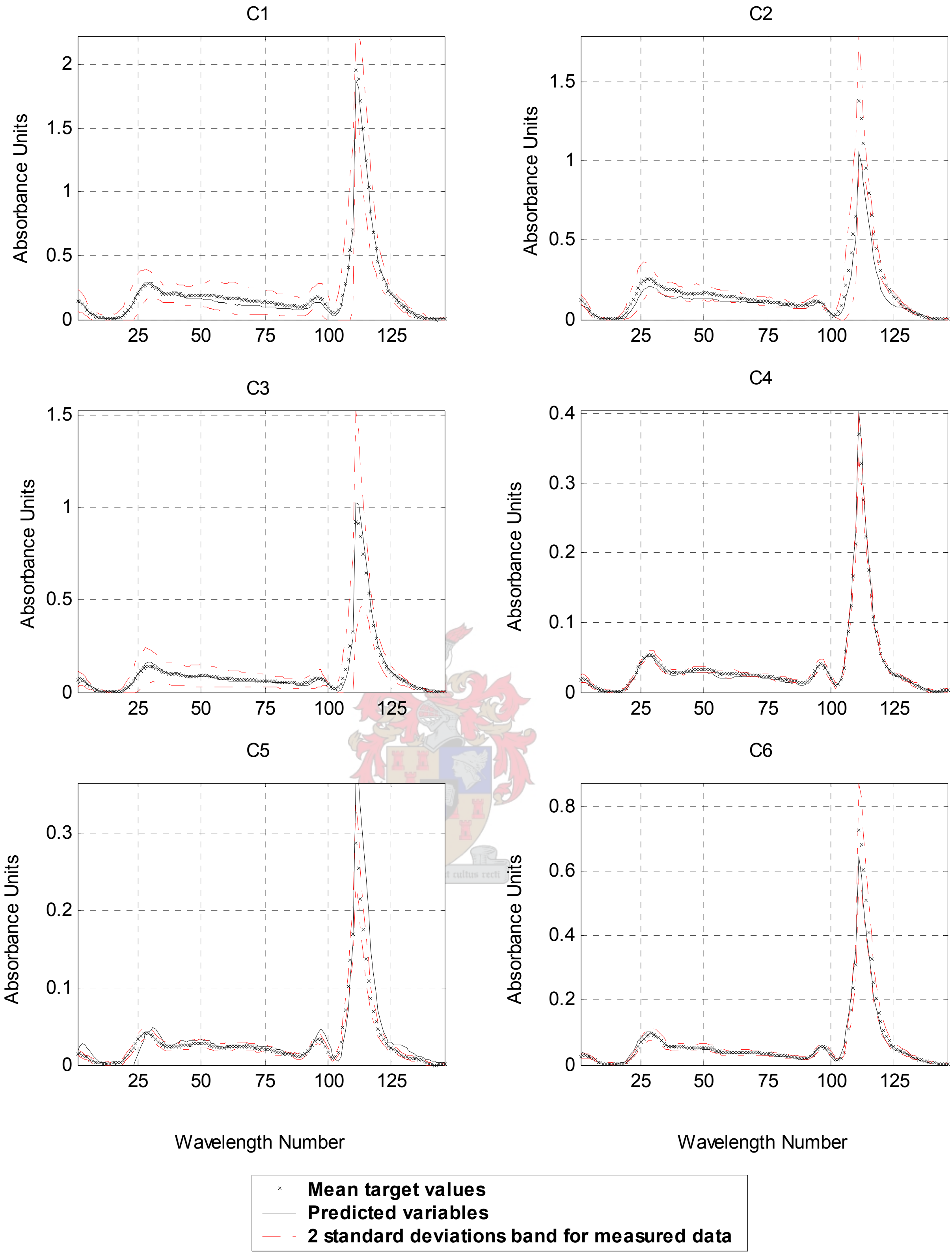

Fig. 4-33 The plume irradiance predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of NNPLS with 11 latent dimensions.

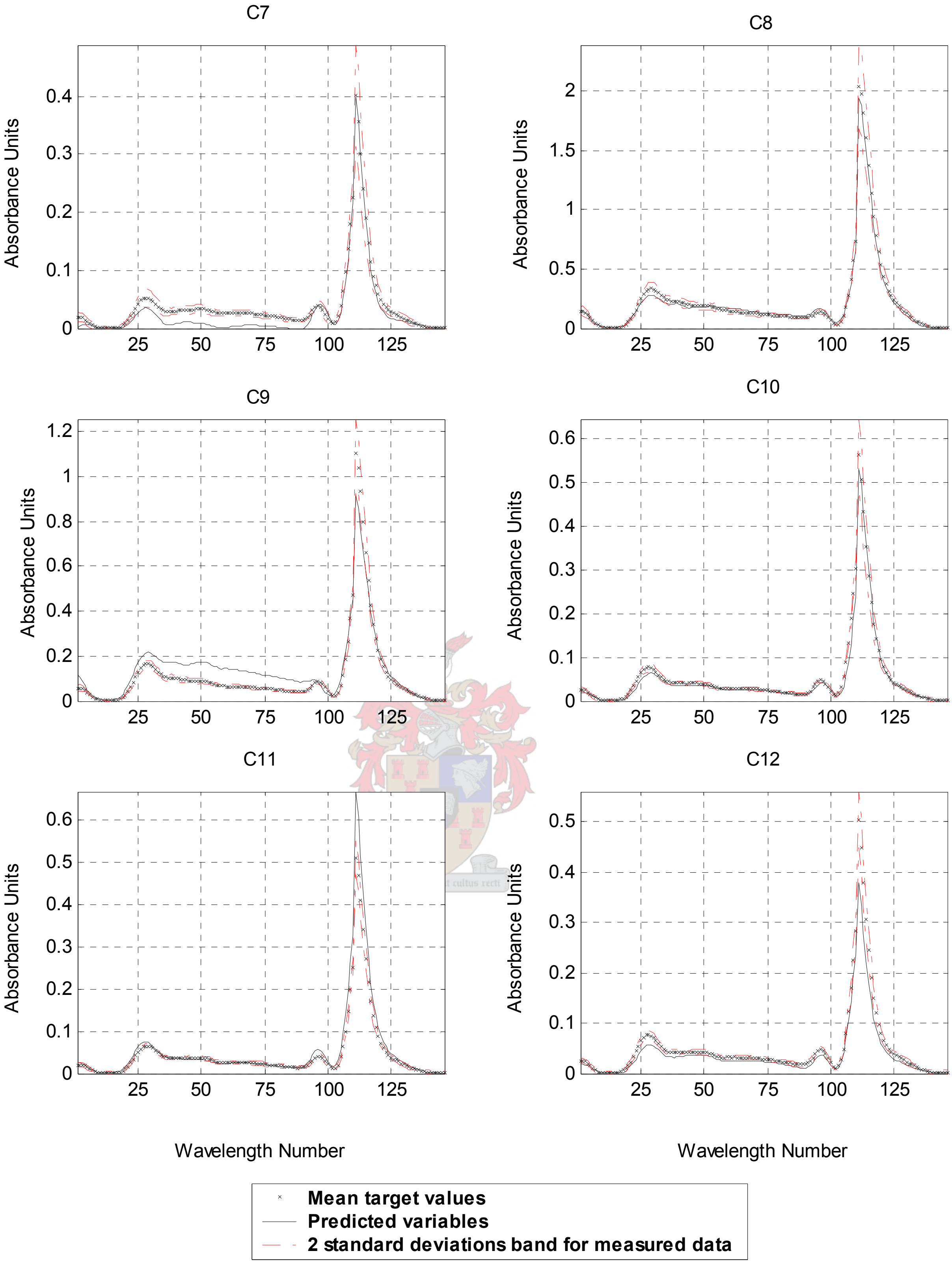

Fig. 4-34 The plume irradiance predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of NNPLS with 11 latent dimensions.

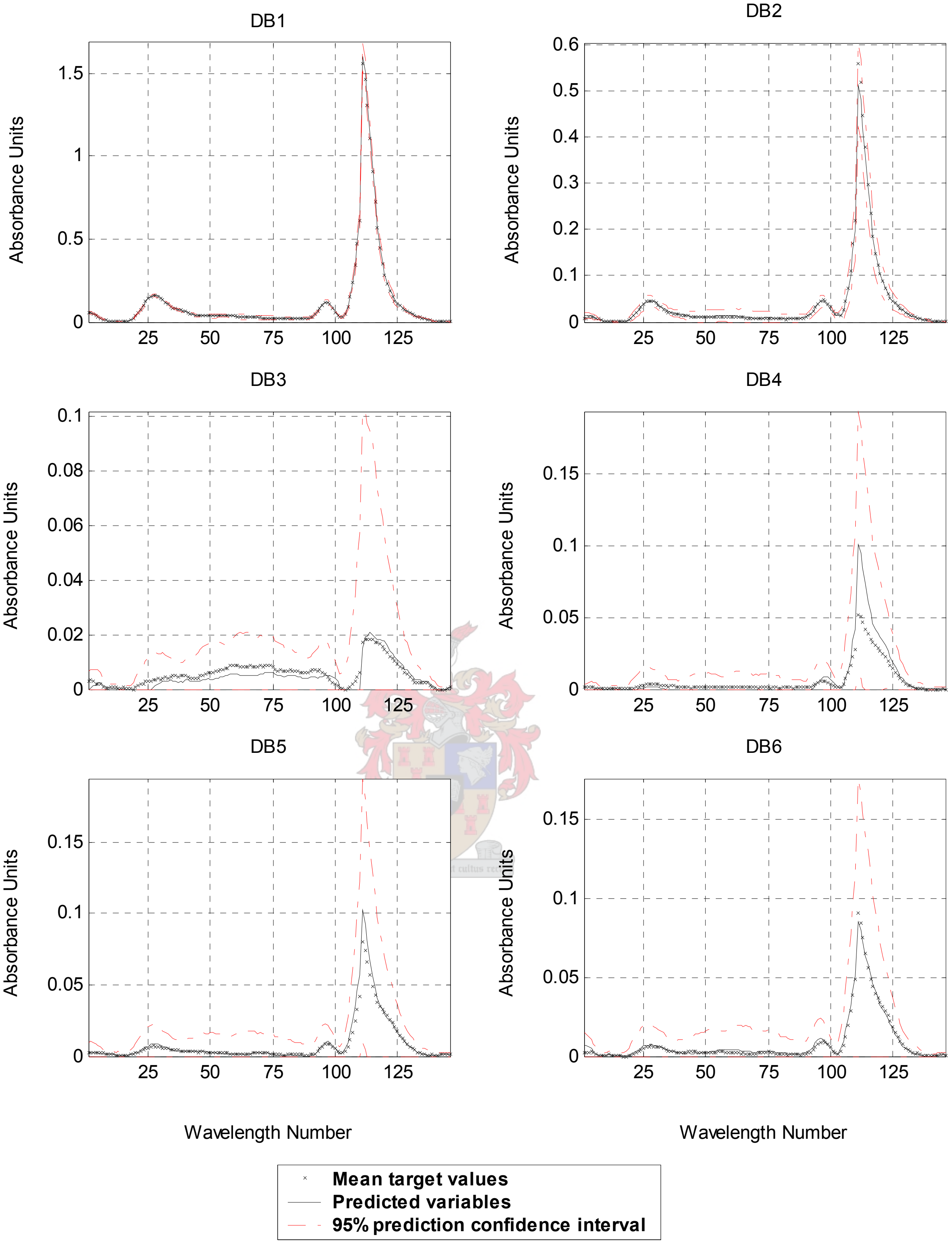

Fig. 4-35 The plume irradiance predictions for rocket motors DB1 to DB6 obtained for the overall NNPLS model using 11 latent dimensions.

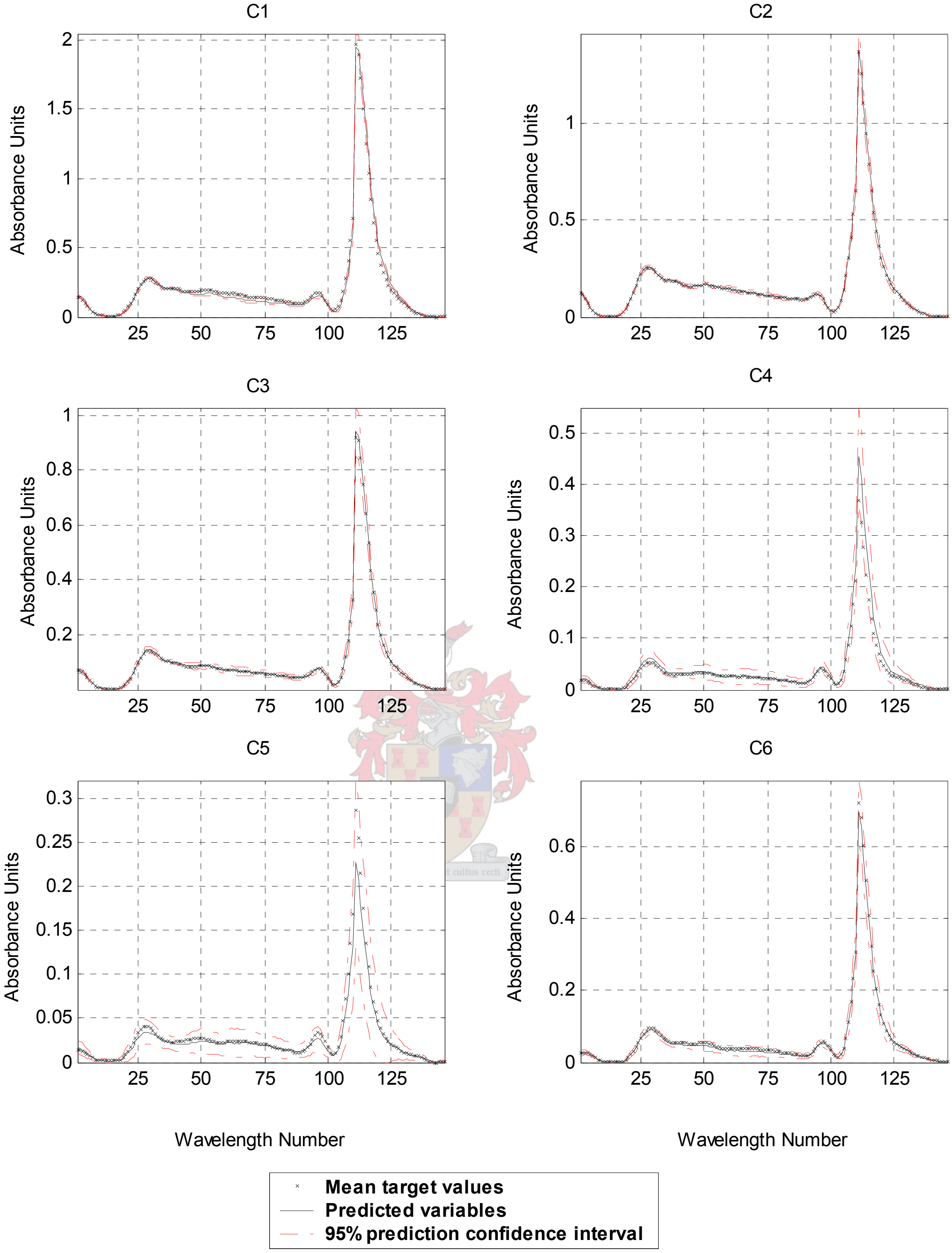

Fig. 4-36 The plume irradiance predictions for rocket motors C1 to C6 obtained for the overall NNPLS model using 11 latent dimensions.

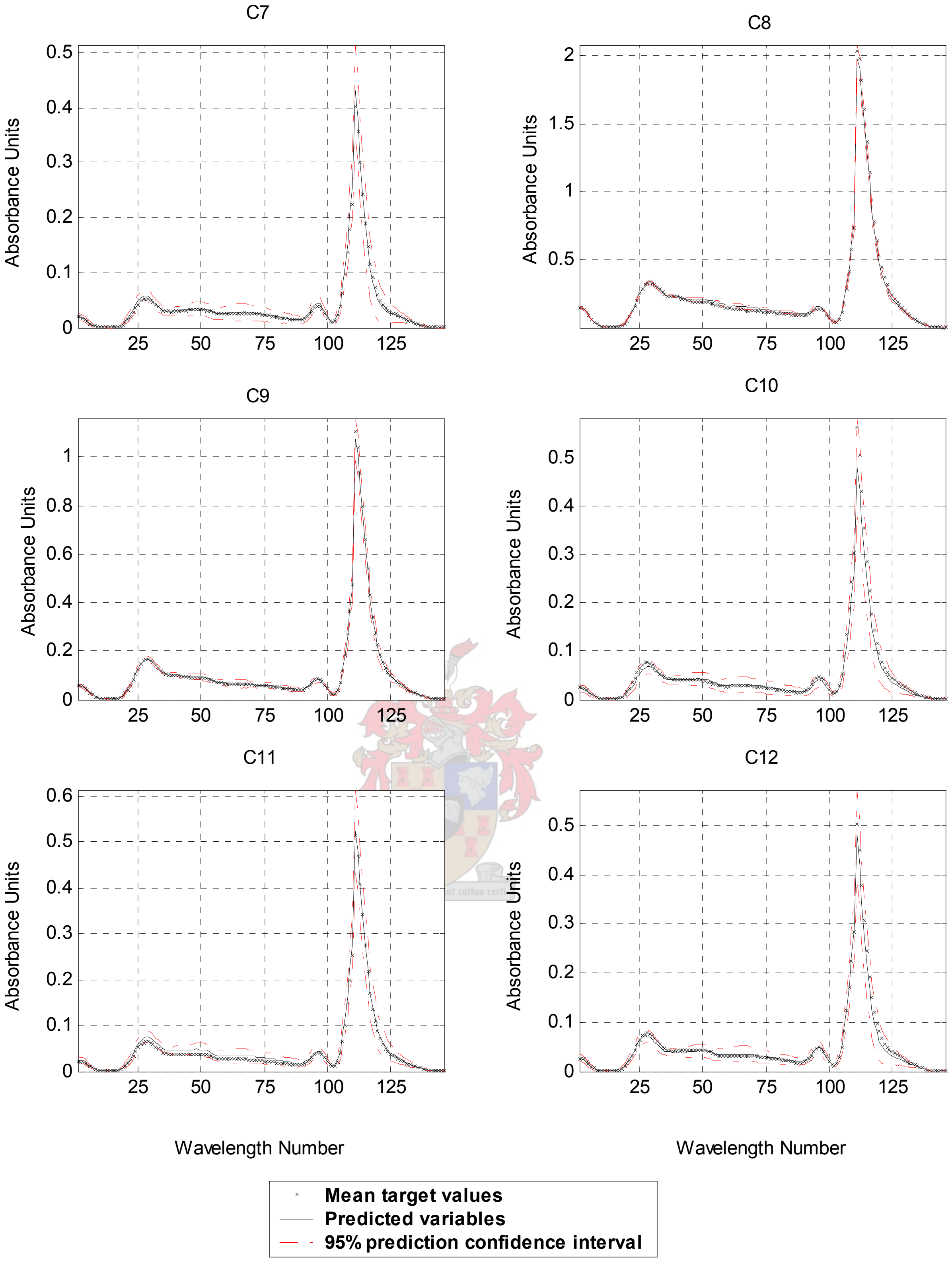

Fig. 4-37 The plume irradiance predictions for rocket motors C7 to C12 obtained for the overall NNPLS model using 11 latent dimensions.

4.2.4 Radial-basis Function PLS

The use of RBFPLS as an alternative to NNPLS is discussed in section 2.13.2. A motivation for applying this algorithm is that the RBF training algorithms discussed in section 2.12 allow for repeatable training results. The RBF neural networks form a part of a unique class of neural networks as far as model training and functional relationships are concerned. The characteristic feature is that the hidden node outputs are based on the proximities of the input data to pre-selected centres. For some data sets this philosophy may produce good results, depending on the nature of the input-output relationship. The training of a RBF network by K-means clustering is achieved much more quickly than an equivalently sized sigmoidal feed-forward neural network. Due to the results obtained in section 4.2.3 the EBRBFPLS algorithm is not considered here.

In Table A-11 the modified cross-validation and overall model results are shown for the RBPLS models built using the K-means RBF training method (see sections 2.7 and sections 2.12.1). The initialisation of the K-means algorithm has a significant effect on the model performance. The K-means method is applied using two different initialisation methods. The first is that of finding the first K furthest points within the RBF model inputs. The second method makes use of the primary principal component scores of the input space. The data is then split into two clusters along the centre of the principal component axis. The principal components of each of these clusters are then again used to further split each cluster into two. This is repeated until there are K-clusters. The centroids (means) of each cluster are used as the K initial centres. This is the so-called PCA binary split method. The splitting is stopped once there are K-clusters, else it would only be possible to have 2^n clusters, for n = 1,2,…,etc.

The OLS method for training the RBF within the RBFPLS model requires that the spread parameter be calculated according to some heuristic (see section 2.12). The use of equation (2-126) generally produces the lower residuals compared to equation (2-119). These results are shown for 80 and 120 nearest neighbours in Table A-12.

A comparison of the RBFPLS models built using the RBF algorithms discussed above reveals similar results. The PRESS-values obtained from the K-means methods appear to go through the characteristic minimum at lower model complexities. It is interesting to note that the number of hidden nodes for each inner model do not exceed the value of 5. This shows that there are no more than 16 parameters per inner model (see discussion in section 4.2.3). The PRESS-values obtained from the OLS methods do not appear to go through distinct, characteristic minima at lower model complexities. Many of the inner model radial-basis functions do consist of more than 5 hidden nodes.

The RBFPLS PRESS-values using the proposed ASOLS training algorithm for each inner model are shown in Table A-14 for 5 runs. Each run uses 80, 120, 160, 110 and 130 nearest neighbours, respectively for each inner model to initialise the ASOLS algorithm (see steps 1 and 2 of the ASOLS algorithm in section 2.12.2). The ASOLS algorithm further uses 2 nearest neighbours in steps 7 and 15 of the algorithm. This is due to the single input-single output architecture of the inner models and no more than 2 nearest neighbours are therefore required. A few of the inner model training results are shown in Table A-13 to illustrate how the ASOLS algorithm converges using a tolerance of zero in step 13 of the algorithm. There are reductions in SSE for dimensions 1, 4 and 6, which indicate an improvement on the OLS algorithm without modification.

The corresponding SSEP-values (from ASOLS training) obtained from training on the overall model are shown in Table A-15. The second run, using 120 nearest neighbours by far produces the lowest PRESS-values. This is evident in Fig. 4-38 where the PRESS-values are compared to the average PRESS-values over all 5 runs. The minimum PRESS-value for run 2 lies at latent dimension 10. From the average PRESS-values it is evident that, generally, there is no optimum minim value for model complexities at latent dimensions below 18. The minimum at LD 10 for run 2 is quantitatively not as convincing as the minimum value obtained at LD 11 for NNPLS. Inner models with 6 and 10 hidden nodes are however present (see Table A-16).

It seems safe to assume that the optimum model therefore lies somewhere from LD 10 to LD 18 for the RBFPLS model with ASOLS training. Low model complexities are always desirable and for this reason it appears reasonable to accept 10 latent dimensions as the optimum model. In Fig. 4-39 it can be seen how the Y-block explained variance already tapers out by LD 10 and upwards.

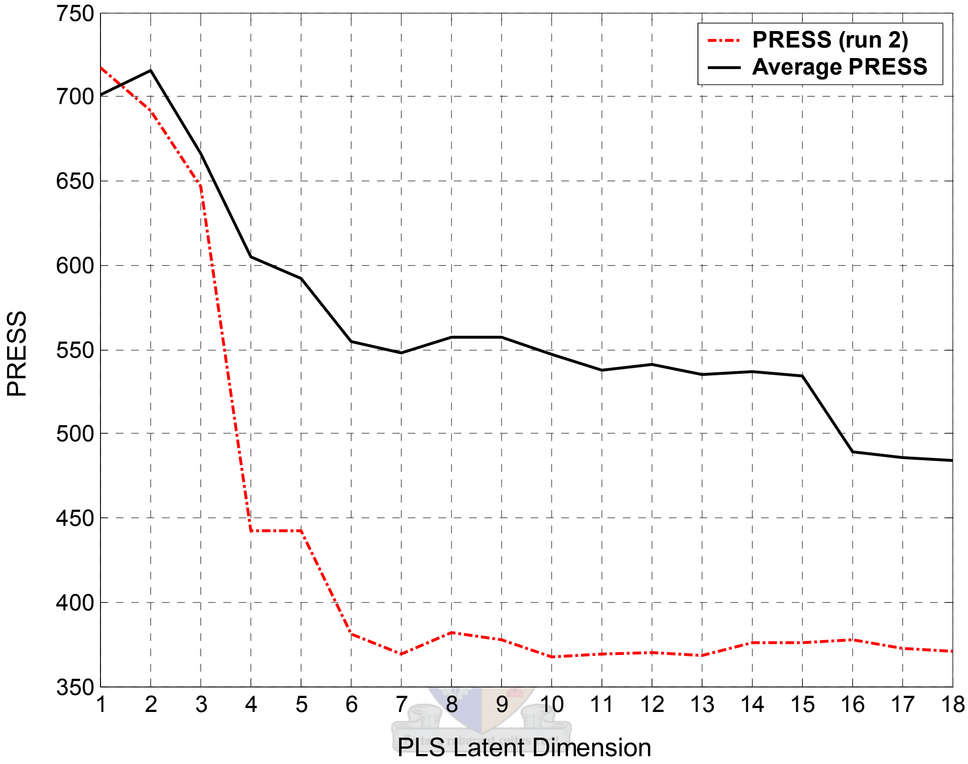

Fig. 4-38 A plot of PRESS-values using the ASOLS training algorithm for each inner model within RBFPLS (see Table A-14).

In comparison to the previous models an elaborate discussion is not included here due to the similarity in structure and patterns of results to the NNPLS model. The squared correlations between the predicted and mean target spectra for all output variables are plotted in Fig. A-9. The basic trends are similar to those of the previous models. The predictions across the entire spectrum band for the RBFPLS model fairs worse in comparison to the NNPLS model. The RBFPLS model is able to better predict the CO_2 absorption band compared to the linear PLS and feed-forward neural network models, however the band between the H_2O and CO_2 absorption regions is predicted more poorly with $R^2_{CV(max)}$ –values between 0.5 and 0.6.

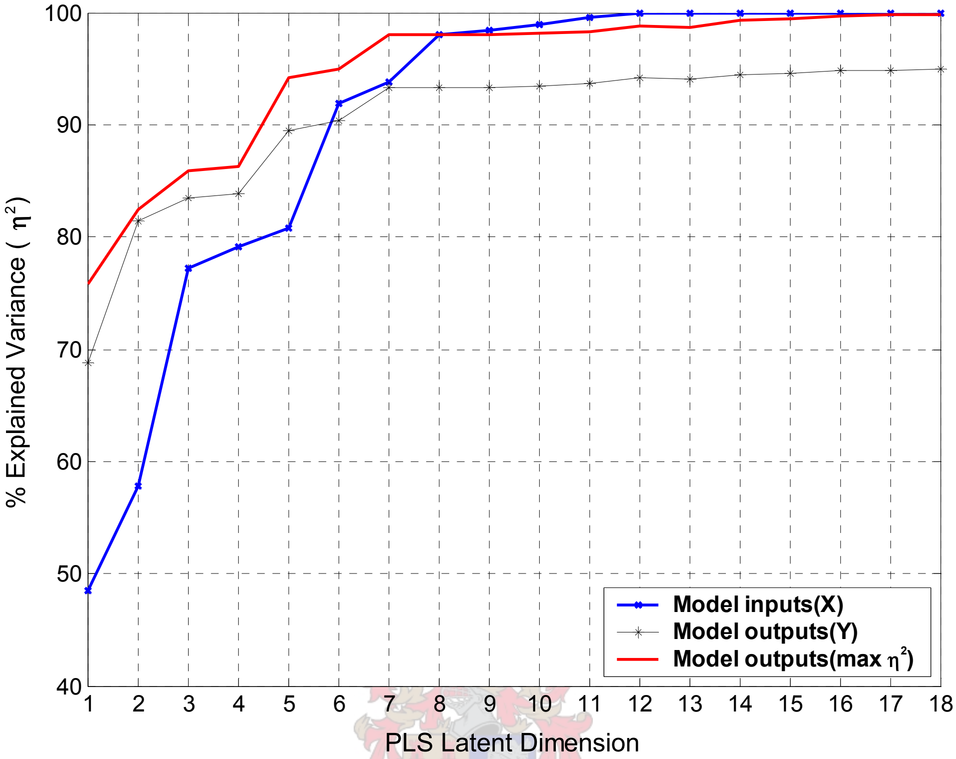

Fig. 4-39 The increasing explained variance of the overall RBFPLS model with ASOLS training of the inner models.

In comparison to the other PLS models a similar tapering trend towards LD 11 and 12 for the average pseudo degrees of freedom is shown in Fig. 4-40. This supports the idea that the optimum model complexity lies somewhere in the region of 10 to 12 latent dimensions. Beyond LD 13 there seem to be no significant increases in model degrees of freedom. This strongly suggests significant over-parameterisation in the higher model complexities.

The cumulated total number of parameters per output variable and the number of parameters per inner model are tabulated in Table 4-6. At LD 10 there are more model parameters cumulated in comparison to the NNPLS model. The NNPLS model is thus more efficient in extracting the equivalent input-output relationships in the data. The RBFPLS model contains more inner models with over 5 hidden nodes per inner model neural network, which might make the model less desirable than the NNPLS model.

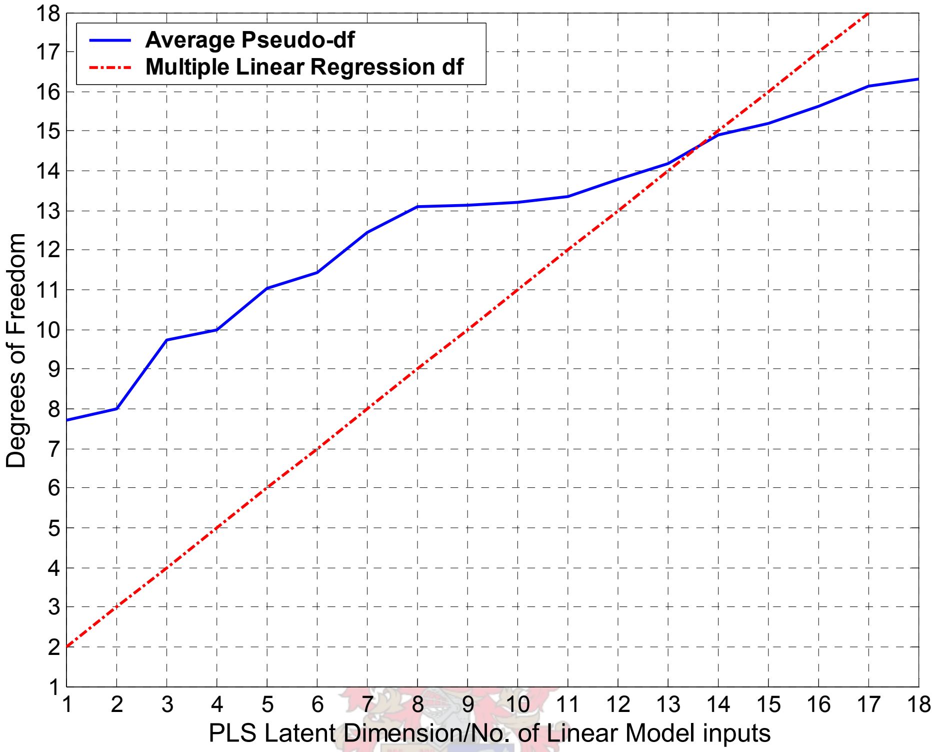

Fig. 4-40 The average pdf over all output variables for the overall RBFPLS model with increasing number of latent dimensions compared to multiple regression df.

The predictions of the unseen irradiance spectra as shown in Fig. 4-41 to Fig. 4-43 indicate that most of the rocket motor predictions are not accurate to within two standard deviations of the measurement variances for each rocket motor design. Negative predictions are again given default values of zeros, as negative absorbances are not practically tangible. Except for DB1 and DB2, the model seems to have no affinity for the DB-class rocket motor spectral predictions. The C-class rocket motor spectral predictions are generally not predicted so accurately compared to the other models. The orders of magnitude of the spectra are however predicted accurately. The region between the CO_2 and H_2O absorption peaks is generally poorly predicted. This explains the lower $R^2_{CV(max)}$ –values obtained in this region.

The fair predictions made on the unseen data support the fact that the average R^2_{CV} –value at LD 10 (see Table A-17) matches the value of 0.417 obtained for the feed-forward neural network using two hidden nodes. The lower optimum PRESS-value for RBFPLS compared to that of the linear PLS and feed-forward neural network models is indicative of the fact that there are no completely negative spectral predictions made. This is the case for the predictions made for C5 for example.

Table 4-6 The total number of parameters per inner model and output variable for the optimum RBFPLS model with ASOLS training.

LD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Total number of inner model paramters	16	20	27	46	56	87	106	119	123	130	137	150	172	176	189	208	212	228
Total number of parameters per output variable*	35	58	84	122	151	201	239	271	294	320	346	378	419	442	474	512	535	570

The rocket motor spectral predictions together with their 95% confidence intervals are shown in Fig. 4-44 to Fig. 4-46. Except for DB3 to DB6 the predictions are accurate. The orders of magnitude of the predictions for DB3 to DB6 are correct, which is encouraging. The confidence interval bands are larger compared to the NNPLS model.

The target and predicted output scores versus the input scores for each inner RBFPLS overall model up to LD 6 have been plotted in Fig. A-10. The functional relationship is the most non-linear in LD 1. This non-linearity manages to capture 39.5% more explained variance of the outputs compared to linear PLS. Shapes of the prediction curves for LD 1 and LD 2 appear similar to those obtained for the overall NNPLS model. The inner models for each latent become seem to become more linear in nature with increasing latent dimension.

From the results for the RBFPLS model with ASOLS training it is evident that the model performs slightly better overall compared to the linear PLS and feed-forward neural network models. However, there are certain areas where its performance is weaker, such as the predictions in the absorption band between the CO_2 and H_2O absorption peaks. The NNPLS model performs better than the RBFPLS model overall.

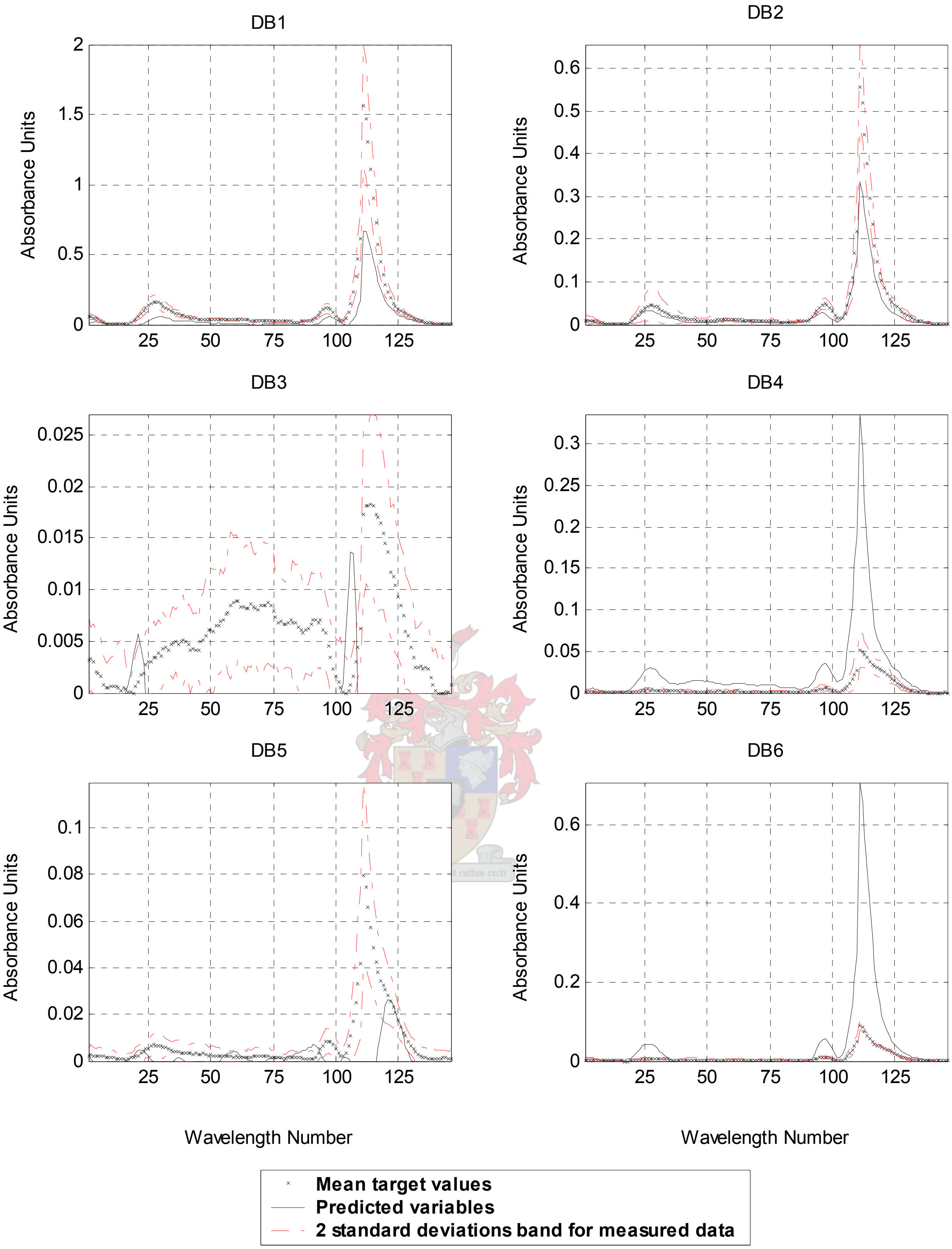

Fig. 4-41 The plume irradiance predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of RBFPLS (ASOLS training) with 11 latent dimensions.

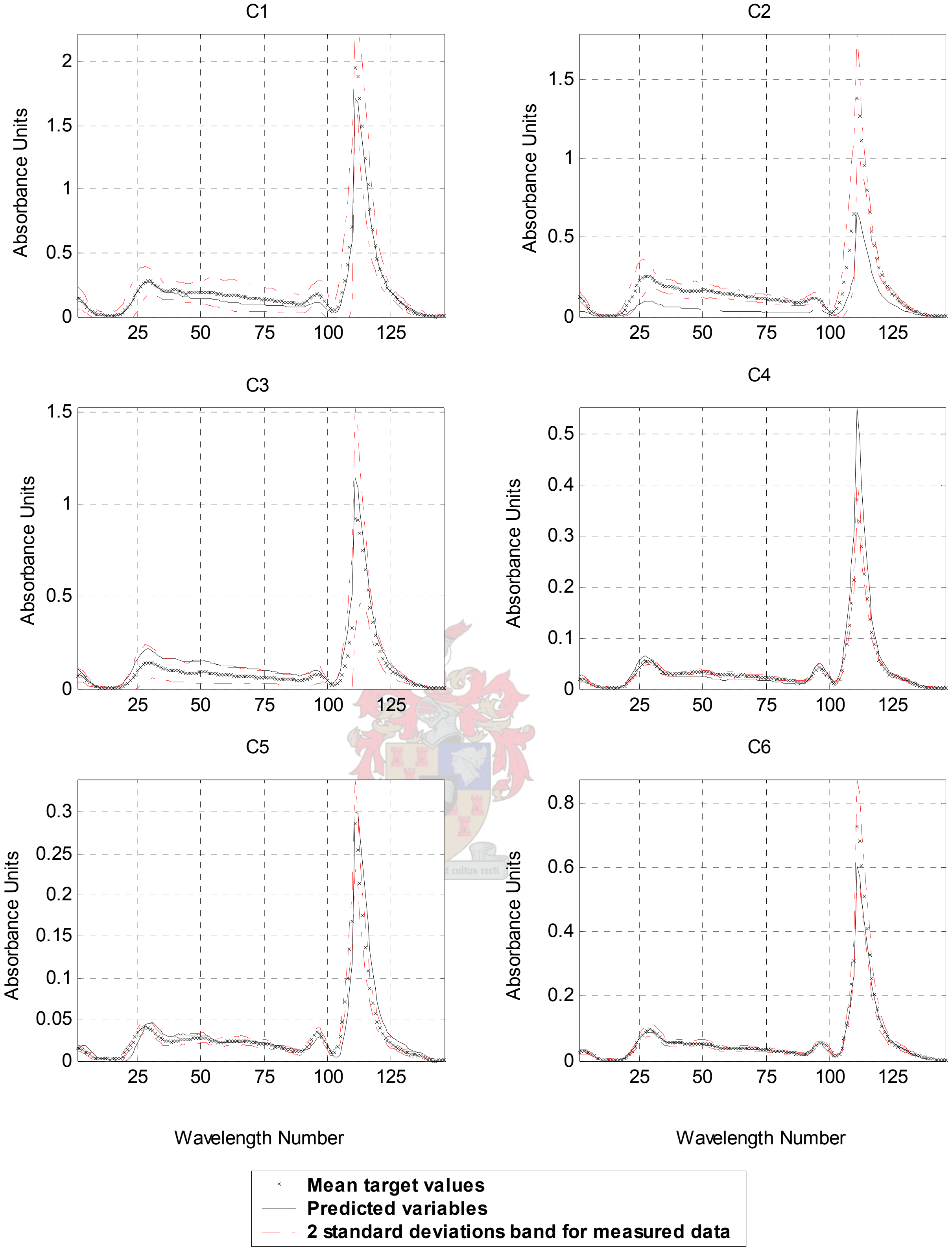

Fig. 4-42 The plume irradiance predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of RBFPLS (ASOLS training) with 11 latent dimensions.

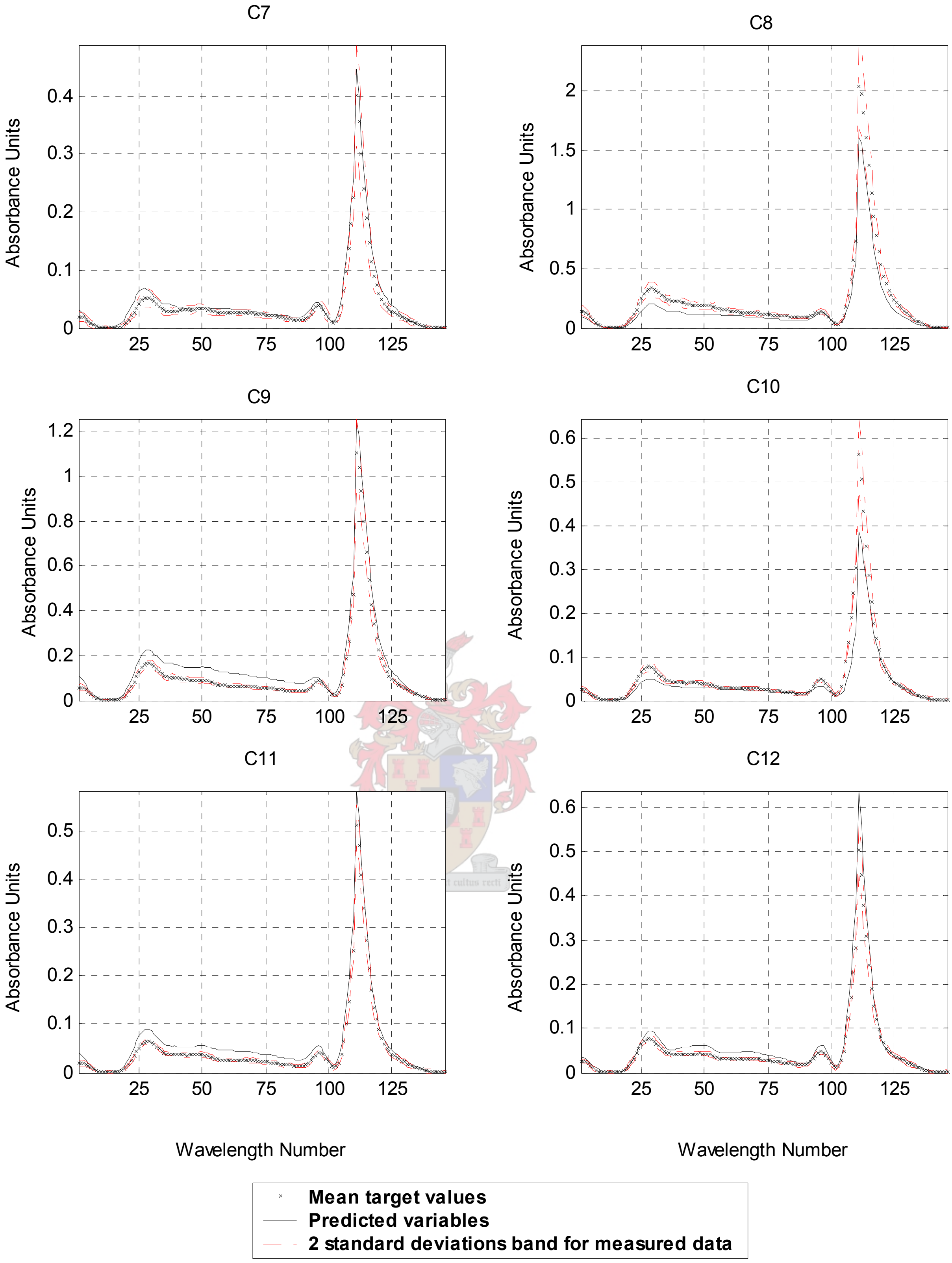

Fig. 4-43 The plume irradiance predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of RBFPLS (ASOLS training) with 11 latent dimensions.

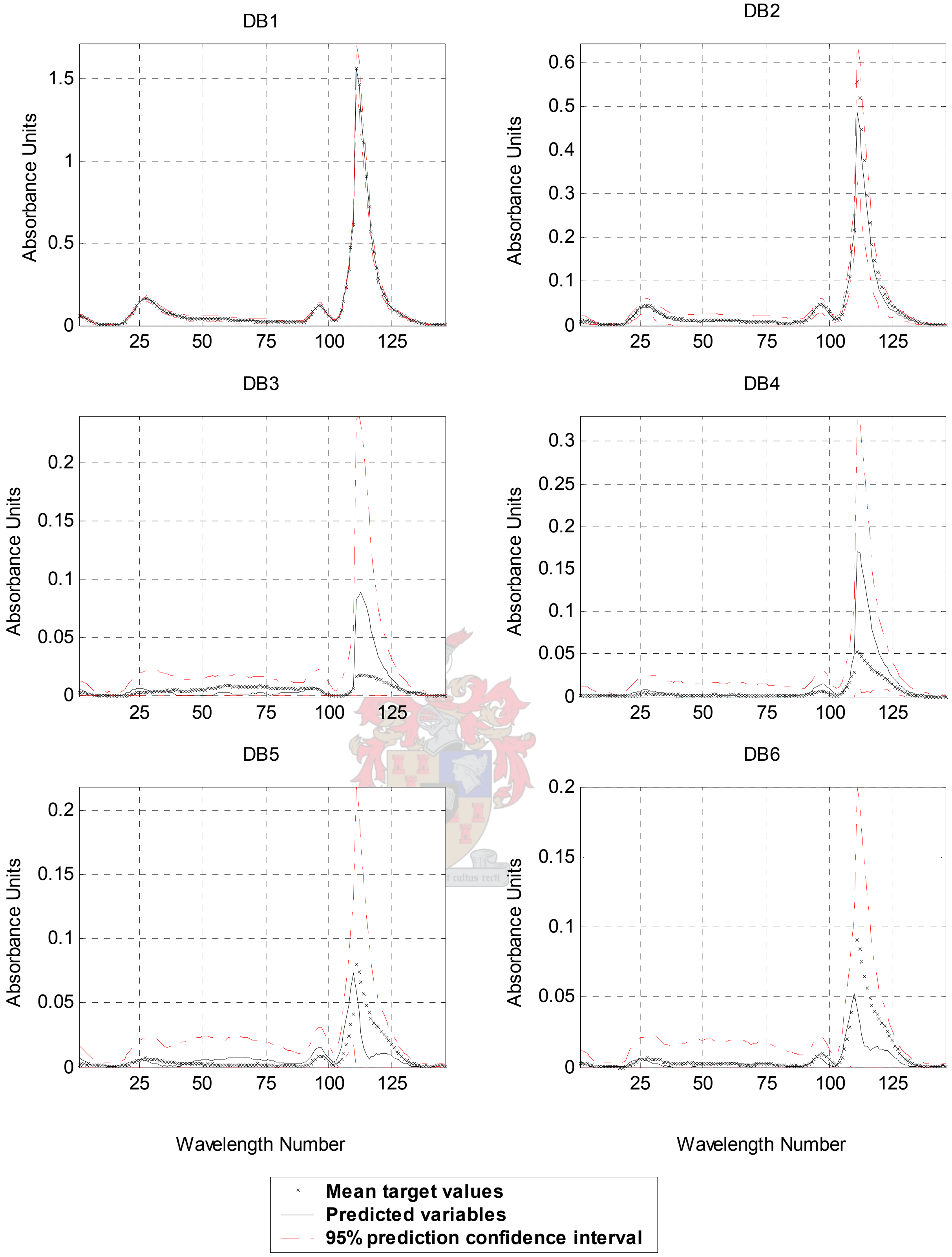

Fig. 4-44 The plume irradiance predictions for rocket motors DB1 to DB6 obtained for the overall RBFPLS (ASOLS training) model using 11 latent dimensions.

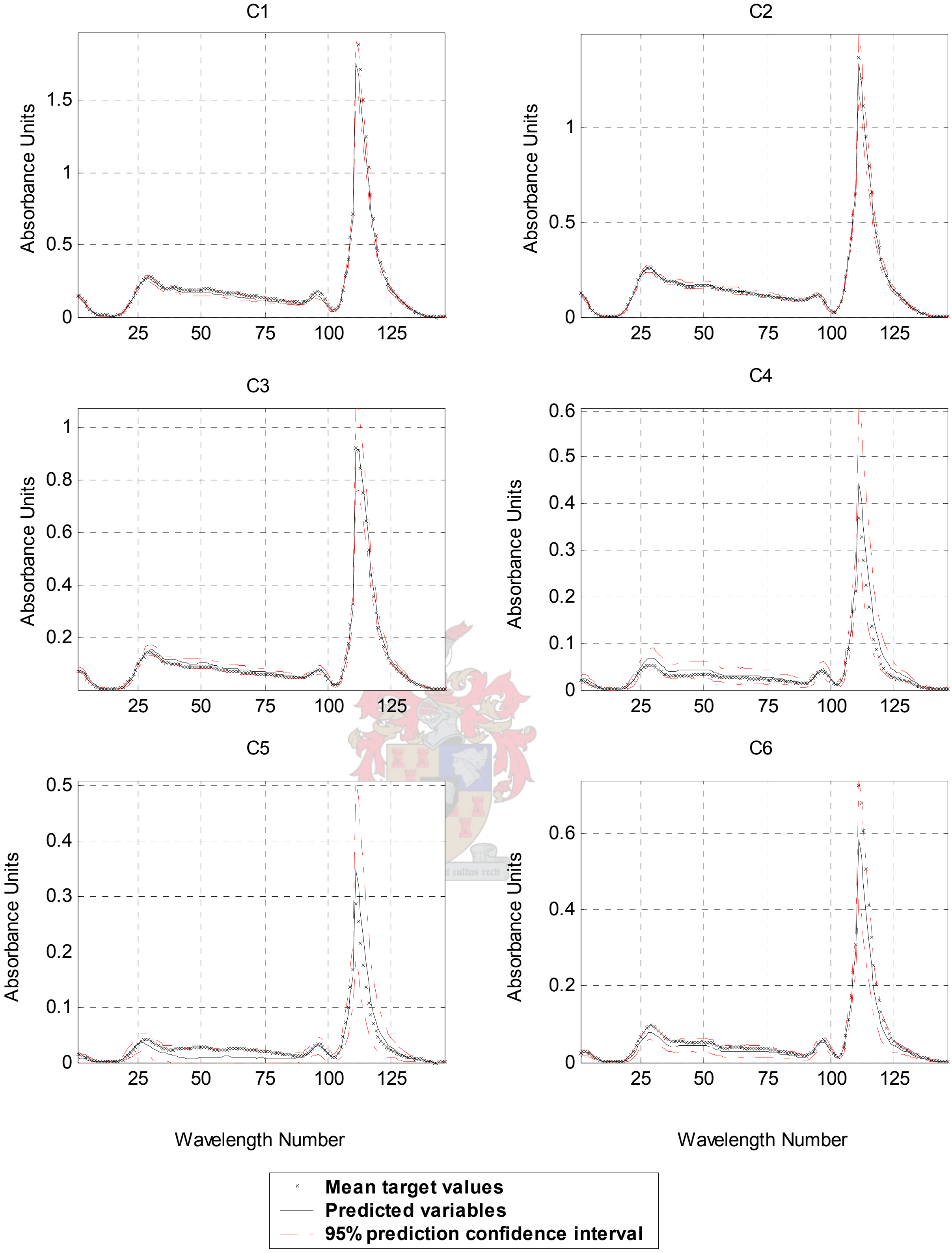

Fig. 4-45 The plume irradiance predictions for rocket motors C1 to C6 obtained for the overall RBFPLS (ASOLS training) model using 11 latent dimensions

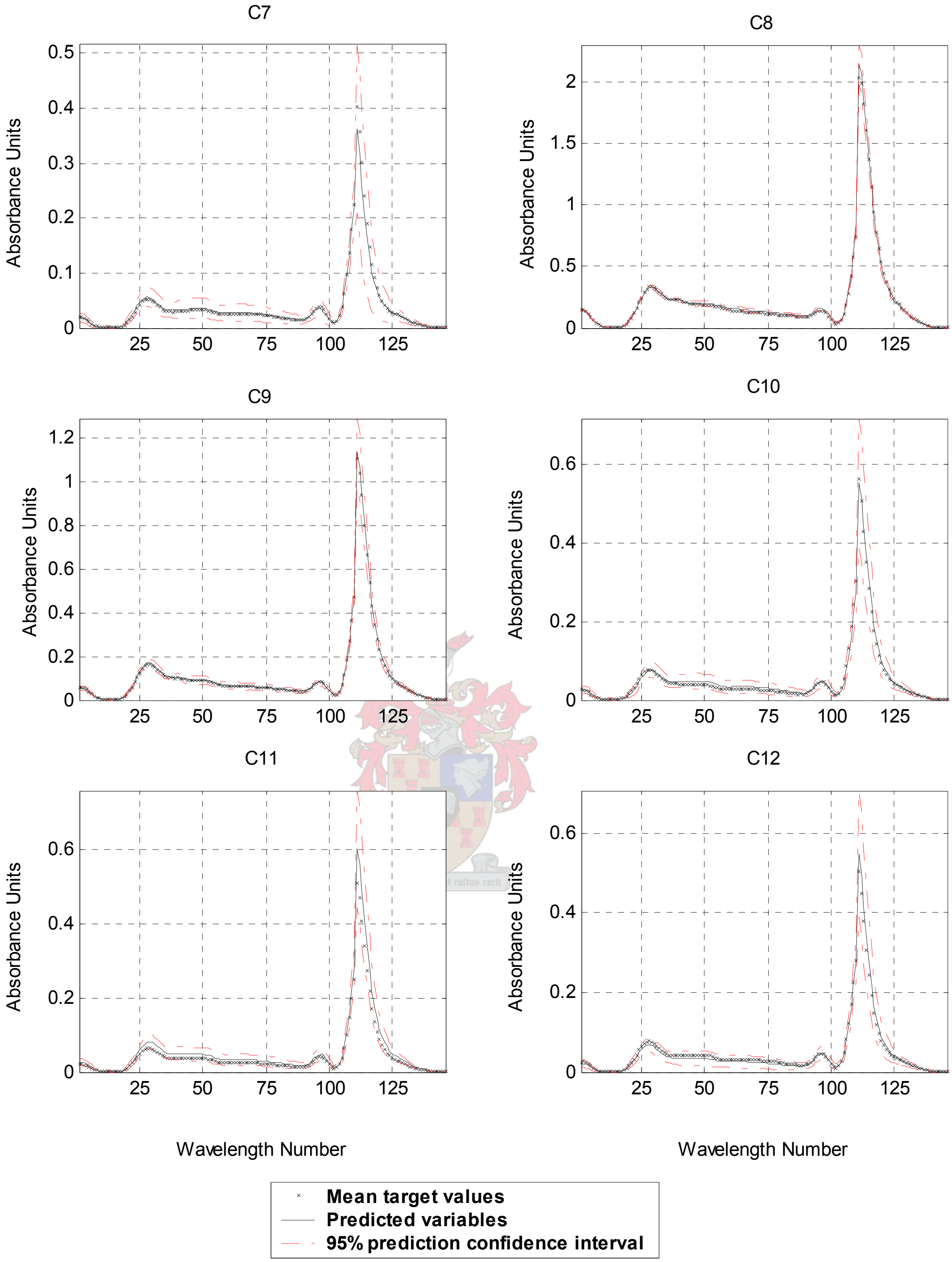

Fig. 4-46 The plume irradiance predictions for rocket motors C7 to C12 obtained for the overall RBFPLS (ASOLS training) model using 11 latent dimensions

4.3 Discussion of the Candidate Models

At this stage it is necessary to consolidate the detailed discussion presented in section 4.2. A comparison summary is presented in Table 4-7. The aim here is not only to draw final conclusions on the comparative model performances but also to find a generalisation on the modelling ability and structure of the data. The EBNNPLS and EBRBFPLS results have not been included due to poor performances on the unseen cross-validated data, as discussed in section 4.2.

The results from section 4.2 and in Table 4-7 show that the most efficient model is that built by the NNPLS model. Only the linear PLS model is able to perform better on the R^2 - and R^2_{max} –values calculated for the overall model. The reason for this is the fact that most of the C-class rocket motor irradiance spectra, except for C5 are accurately predicted using the linear PLS model. The 95% confidence intervals calculated using the higher MSE- and therefore SSEP-values are however extremely large and the calculation of the R^2 – values does not take this into account. If it were not for the failure to predict C5 the overall linear PLS model may even be a promising candidate model. The NNPLS model maintains the linear latent projections and therefore introduces a subtle non-linearity in the inner models to compensate for the shortcomings of the linear PLS algorithm.

In section 2.13.2 it is mentioned that Qin [1992] showed how the NNPLS model could be transformed into a feed-forward neural network architecture. It is therefore possible to represent the predictive function in the form of a single layer neural network with 29 hidden nodes using tan-sigmoidal transfer functions and an output layer of 146 nodes with purely linear functions. This result shows that the back-propagation algorithms are not necessarily the best methods for training a feed-forward neural network when dealing with multivariate outputs. The result further supports the idea by Lawrence [1997] that it is possible to over-parameterise a neural network and still achieve the desired results.

It is interesting to note that the optimum models for all four candidate models produce similar average pseudo degrees of freedom. The three PLS models also produce their optimum models at similar numbers of latent dimensions. The implication of these results shows that the optimum results are obtained at a certain MSECV/MSEP$_{rs}$-ratio. This further implies that the PRESS to SSEP ratio is of importance and that the different PLS models satisfy this ratio at similar model complexities. Other non-linear model complexities like the feed-forward neural network can now be compared to the PLS models by virtue of the pseudo degrees of freedom.

Table 4-7 A summary of performance scores for each optimum model. The Y-block variances are calculated on the overall optimised models.

	linear PLS	**Feed-forward NN**	**NNPLS**	**RBFPLS (ASOLS)**
Complexity	11 LD	2 H	11 LD	10 LD
PRESS	688.50	613.23	258.10	367.56
SSEP	76.28	52.74	45.41	53.06
Y-block %η^2	90.52	93.43	94.34	93.39
Y-block max %η^2	94.98	98.31	98.98	98.09
Ave R^2_{cv}	0.461	0.417	0.626	0.417
Ave $R^2_{cv(max)}$	0.541	0.548	0.746	0.527
Ave R^2	0.876	0.803	0.825	0.818
Ave R^2_{max}	0.957	0.907	0.954	0.946
Ave pdf	12.42	12.91	12.33	13.21
Parameters	220	476	307	346

The model optima of the candidate models have therefore been determined by the trade-offs between cross-validation and the overall model training. The optimum predictive ability for the data is further validated by the comparison of the optima from the different model architectures and algorithms. It may be possible to find an improved model architecture and algorithm, but the expectation would therefore be to find the optimum at comparable model complexities. This optimum is therefore limited by the amount and nature of the available data.

It is desirable to obtain PRESS-values comparable to the SSEP-values. In all the models applied the PRESS-values seem high relative to the corresponding

SSEP-values. The best way of testing the equivalence between the cross-validation residual variance and that of the overall model is by comparing the mean-square errors, MSECV and MSE (overall model). A more detailed discussion on residual variances follows in section 4.4.

4.4 Analysis of the Residual Variances

An analysis of model residuals is standard practice when evaluating the legitimacy of a model. From the results obtained thus far it is shown that the NNPLS model appears to be the more promising candidate model. For this reason the analysis of residuals and variance is limited to the NNPLS model only.

It is a daunting task to plot the residuals for all 146 output variables against their data sample point number. For this reason the Euclidean distances between the predicted and mean target irradiance spectra are plotted using the overall NNPLS model in Fig. 4-47. The Euclidean distances for the standardised data and the re-scaled data are plotted here as a check on the results of the re-scaled data.

The Euclidean distances for C1, C2 and C8 are high due to the fact that their spectral absorbance values are the highest compared to the other rocket motor irradiance spectra. The values of the Euclidean distances are spread out randomly. This is expected, as the NNPLS model is essentially approximated using a universal approximation function. This means that the function will not suffer from inherent functional mismatch, as can be the case for linear models. In such a case it would be possible to identify a trend in the residual plots due to the fact that the wrong functional form for the mapping function is chosen.

Due to the presence of repeat measurements in the output data set it is possible to do a lack-of-fit test described in section 2.5.2. Draper [1981] maintains that the usual analysis of variance (ANOVA) F-test for regression and the lack-of-fit F-test are generally not valid for non-linear models. This is due to the fact that the assumption of normally distributed model parameters no longer holds when it is assumed that the model residuals are normally distributed. However Bates [1988] readily uses the lack-of-fit statistic to evaluate the adequacy of a non-linear model. The motivation for using the right-tailed lack-of-fit F-test is that the F-ratio for ANOVA is calculated using the pure error and lack-of-fit mean-square error. The two variables can therefore be regarded as independent and each follows its own χ^2(chi-squared)-distribution.

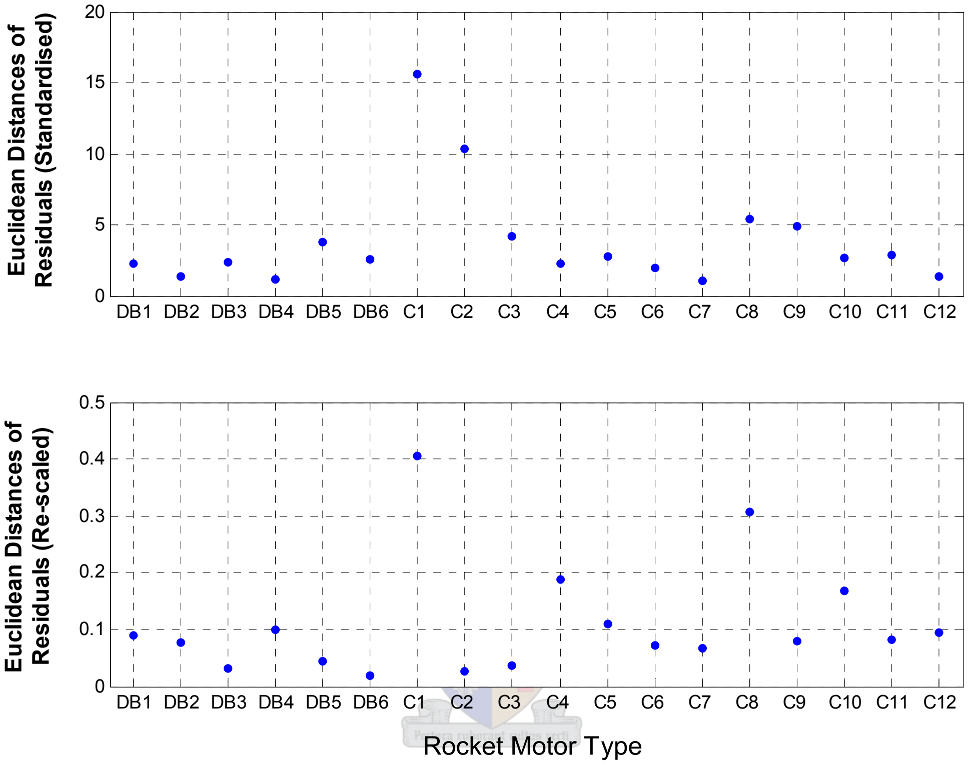

Fig. 4-47 The Euclidean distances between the target and predicted spectra from the overall NNPLS model (LD 11) plotted against their data points.

The multivariate analysis of variance (MANOVA) was attempted for the 146 outputs but it was found that the determinants for calculating Wilks' lambda of the residual matrix could not be calculated due to an ill-conditioned matrix. For this reason the lack-of-ANOVA is done for each individual output variable, j and the results are plotted in Fig. 4-48. The lack-of-fit mean-square error uses 18-pdf degrees of freedom and the pure error uses 399 degrees of freedom. The pdf-values vary as shown in Fig. 4-30. The maximum critical F-statistic of all the output variables and all the model complexities shown at a 1% α-level is indicated.

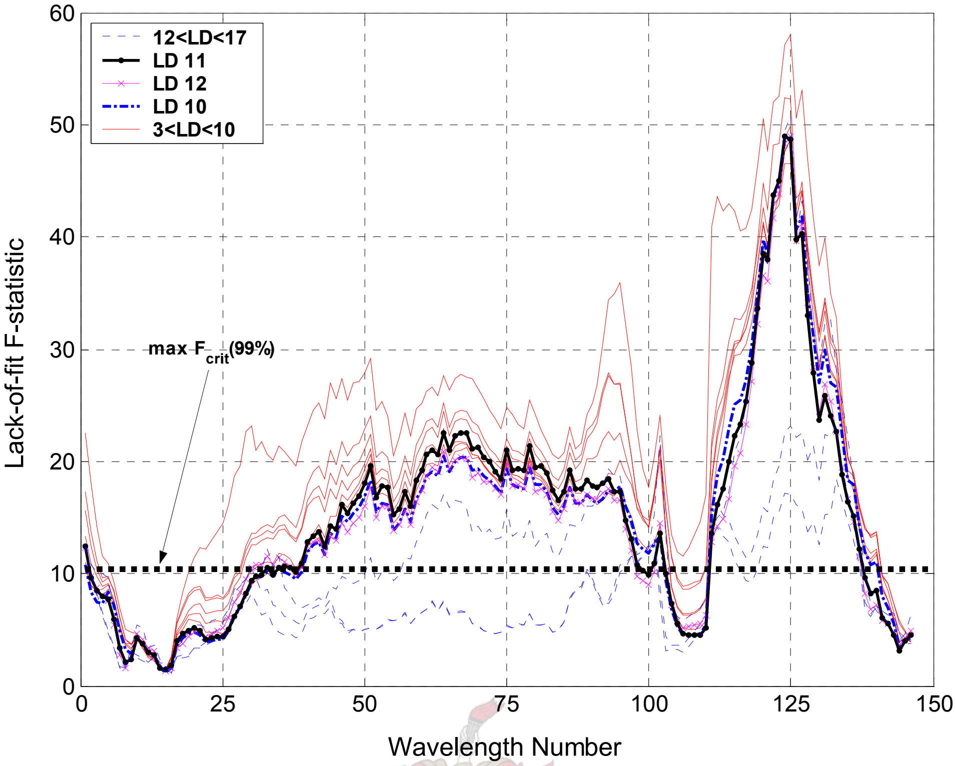

Fig. 4-48 The lack-of-fit ANOVA results for the overall NNPLS model at various model complexities. The maximum critical F-statistic equals 10.45.

The plot in Fig. 4-48 shows that the majority of the predictions across the spectral band indicate significant lack-of-fit, i.e. their lack-of-fit F-ratios are higher than the maximum critical value of 10.45. Only at the higher model complexities do the F-ratios start dipping below the maximum critical value for most of the spectrum. The NNPLS models from LD 13 to 18 are however shown to be over-parameterised in section 4.2.3. The CO_2-absorbance region of the spectrum (wavelength numbers 100 to 140) shows the most lack-of-fit.

In practice the sum-square pure error and lack-of-fit sum-square error are pooled together to calculate the overall MSE (per variable, j) if insignificant lack-of-fit is found. The assumption was made in section 4.2 that the model is adequate and for this reason the MSE is used to calculate the confidence intervals of the output predictions. A plot of the mean-square residuals, MSE, MSE_L, MSE_{min} and s_P is shown in Fig. 4-49. These are the estimates of the residual variance, ε and it is evident that MSE_L does not compare well with the other estimates.

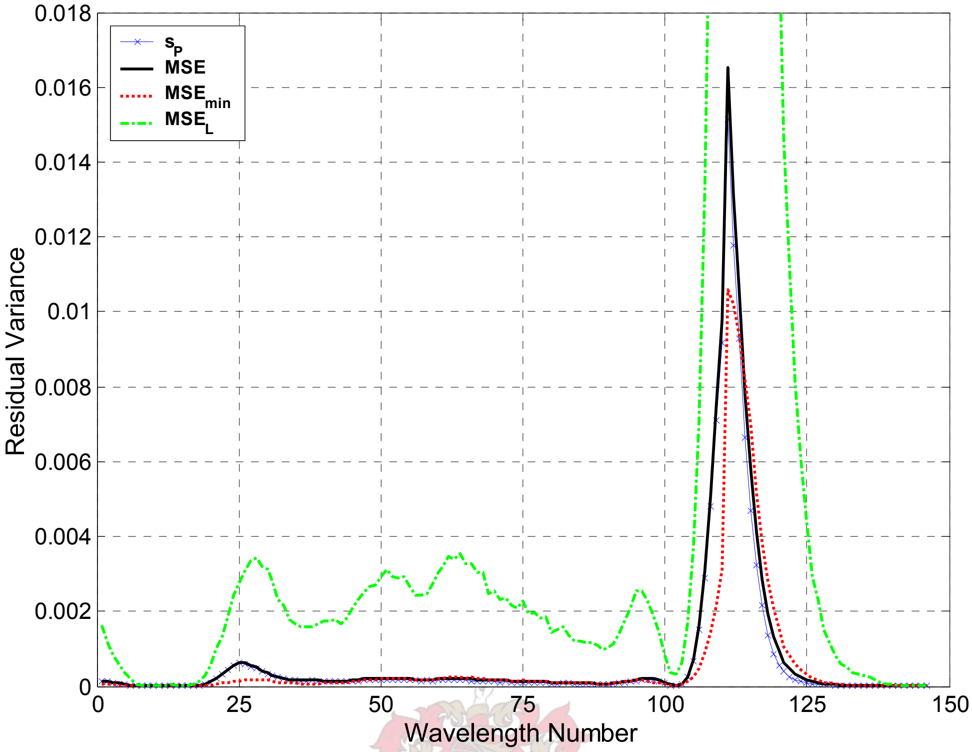

Fig. 4-49 A comparison of the various estimates of the residual variance calculated using the overall NNPLS model with 11 latent dimensions.

The F-statistics in Fig. 4-50 are calculated in order to test for the null hypothesis that the mean-square error (MSE for overall model) and the pure error (s_P) are equivalent. The critical F-statistics are therefore evaluated at 417-pdf and 399 degrees of freedom, respectively. A small region in the band between the H_2O (wavelength numbers 20 to 45) and CO_2 absorption peaks, and a larger band in the CO_2 absorption peak fall outside the critical 99% confidence band. The null hypothesis is therefore rejected for these regions, however on average the F-statistic remains within the confidence band.

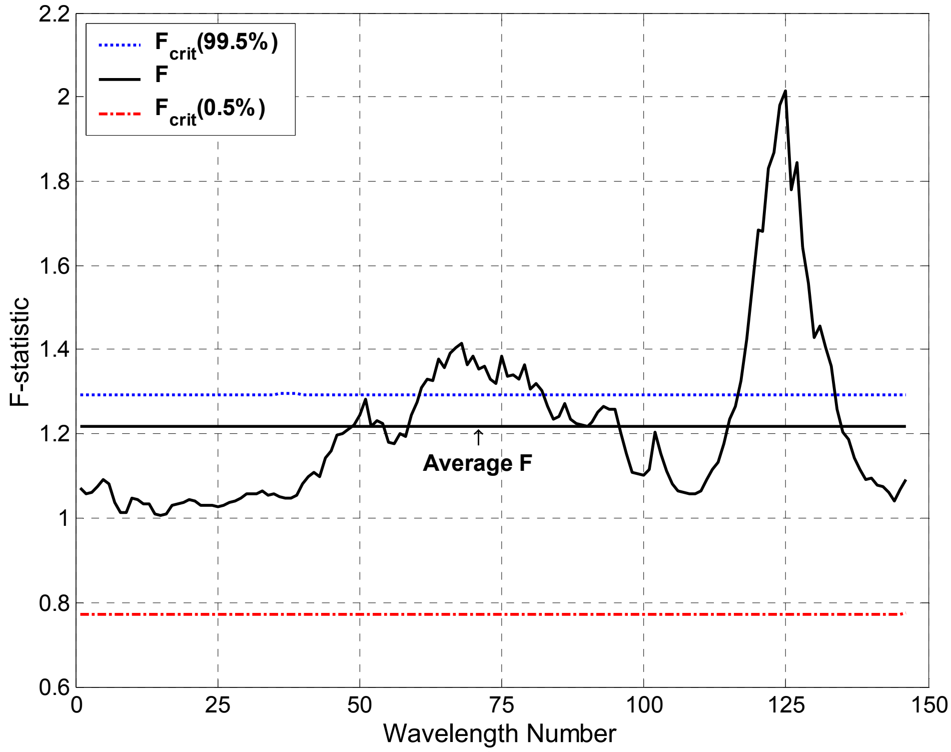

Fig. 4-50 The ratio, F = MSE/s_P is compared to the critical F-statistics at n–pdf and n_R = 399 degrees of freedom for all 146 output variables using the overall NNPLS model with 11 latent dimensions.

The mean-square error (MSE for overall model) is further tested for the null hypothesis of equivalence with the minimum mean-square error (MSE_{min}) as shown in Fig. 4-51. The critical F-statistics are therefore calculated using 417-pdf and 18-pdf degrees of freedom, respectively. The irregular upper bound of the critical 99% confidence band can be attributed to the large difference between the degrees of freedom used by each estimate of residual variance. The plot shows that the null hypothesis can be accepted across virtually the entire spectrum. This result is significant as it motivates the use of MSE_{min} as an estimate of MSE and hence the residual variance, ε.

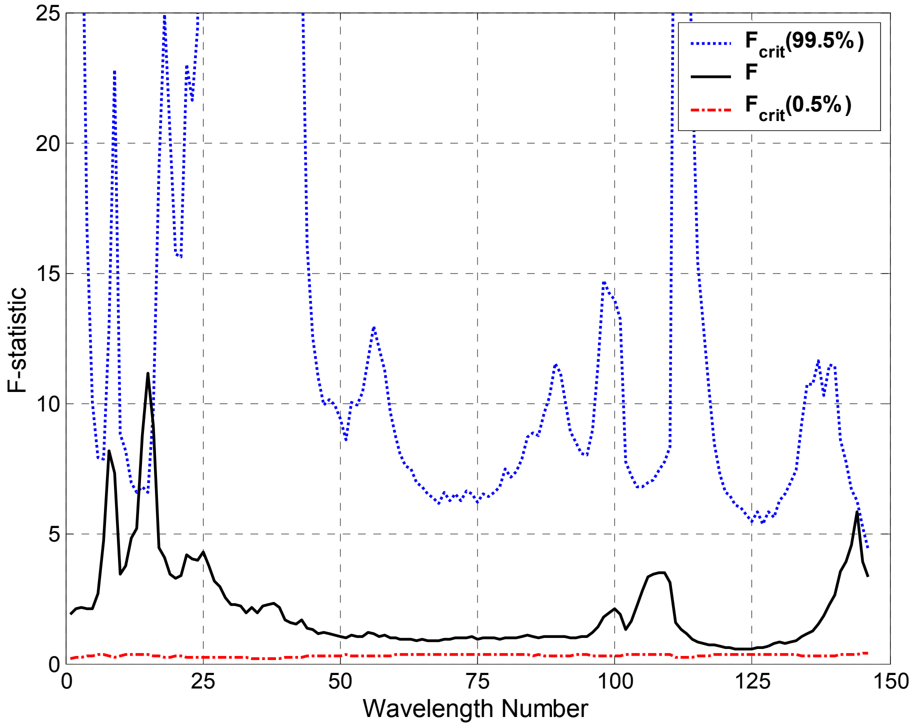

Fig. 4-51 The ratio, F = MSE/MSE$_{min}$ is compared to the critical F-statistics at 417–pdf and 18-pdf degrees of freedom for all 146 output variables using the overall NNPLS model with 11 latent dimensions.

From the predominantly significant lack-of-fit obtained for the best candidate model in Fig. 4-48 there appears to be reason to doubt the adequacy of the overall models to fit the data. In Fig. 4-52 the F-statistics are plotted for the null hypothesis that MSECV and MSE (overall model) are equivalent. The mean-square error of cross-validation is calculated using the PRESS$_{min,j}$-values for each individual output variable (see sections 2.6.3 and 4.2). The critical F-statistics are evaluated at 18 and 417-pdf degrees of freedom, respectively. Most of the F-statistics are outside the critical 99% confidence bands, thus the null hypothesis can be rejected for most of the output variables. This result places doubt on the ability to be able to build a model on the given data that can make accurate unseen predictions.

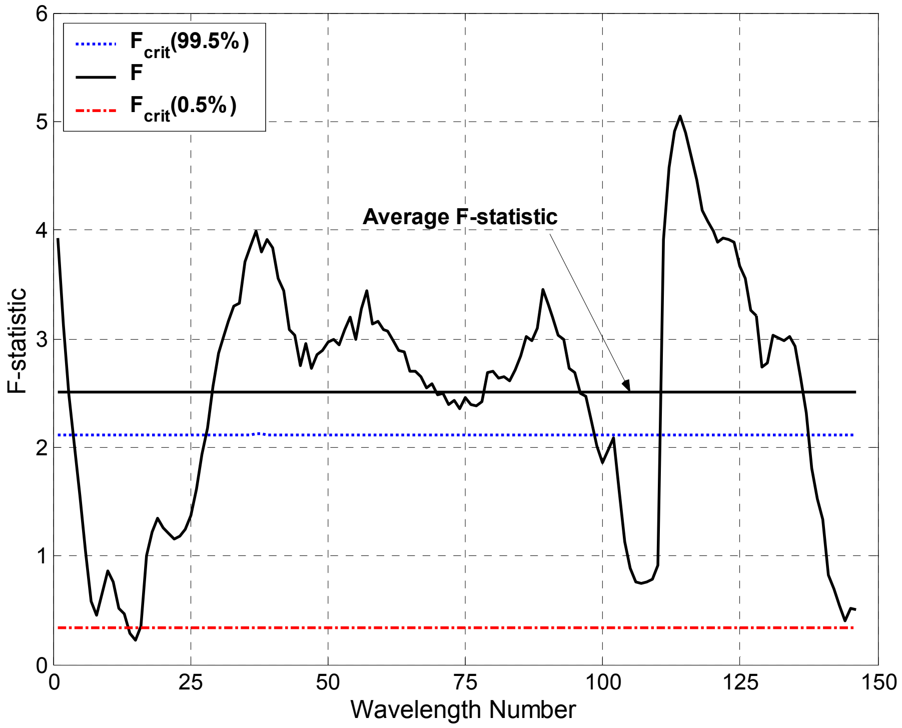

Fig. 4-52 The ratio, F = MSECV/MSE is compared to the critical F-statistics at 18 and 417-pdf degrees of freedom for all 146 output variables using the overall NNPLS model with 11 latent dimensions.

4.5 Sensitivity Analysis

From the statistical tests even the best candidate model, the overall NNPLS model, may not be ideal. A sensitivity analysis is required in order to see if the model is able to qualitatively predict the expected behaviour from perturbations made on the input variables. The sensitivity analysis done here has been adopted from the work done by Roodt [1998]. The reason for this is that the NNPLS model can be compared to the feed-forward model in the referenced work. The sensitivity analysis for this data requires careful planning, as a change in rocket chemistry represents simultaneous changes to multiple elements such as carbon (C), hydrogen (H) and oxygen (O) for example.

Initially, the model responses are tested on changes made to the following key inputs as suggested by Roodt [1998]:

- *The chamber pressure (P_C) and temperature (T_C):* These changes go hand-in-hand and an increase in spectral irradiance is expected.
- *The throat diameter (D_T):* An increase allows more gas to flow per unit time and the amount of cooling is reduced (temperature increases). The optical depth also increases and an increase in irradiance is expected.
- *The expansion ratio (E_C)*: A drop in spectral irradiance is expected due to a drop in temperature when the ratio is increased.
- *The amount of aluminium*: An increase in Al causes an increase in the solids in the plume and therefore an increase in optical density, which increases the irradiance.
- *The amount of flame suppressant*: Potassium (K) acts as a flame suppressant and therefore decreases the spectral irradiance.

The changes made to rockets DB1, C1, C9 and C10 corresponding to the list above are shown in Table A-18. The original model predictions are shown relative to the predictions after the modifications have been made in Fig. 4-53. Except for the changes made to the expansion ratio and the throat diameter the modified predictions are as expected. It must be added that Al has a high heat of combustion and therefore this may be a contributing factor towards the model picking up an increase in irradiance. The decrease in expansion ratio leaves the prediction for DB1 virtually unchanged. This result is not impossible, as a change in expansion ratio may not have a significant as expected effect on the plume temperature (and therefore irradiance intensity). It is encouraging to see the clear decrease in irradiance prediction by the addition of potassium to C10. This prediction is more pronounced compared to that made by Roodt [1998].

It is interesting to note in Fig. 3-7 that the expansion ratio is directly correlated with the chamber pressure and inversely correlated with the throat diameter. The throat diameter is also inversely correlated to the chamber pressure. It is therefore difficult to draw conclusions based on the individual perturbations and their combined effects have to be considered. This is especially the case for the throat diameter, as a decrease in the throat diameter increases the chamber pressure for fixed rocket motor chemistries. This could explain why the spectral irradiance for C9 increases with a throat diameter increase, which is contrary to expectation. These interactions further motivate the use of dimensional reduction of the data to latent structures.

A further analysis is made by making global changes to the rocket, C11. This analysis is more appropriate following the preceding discussion. The modifications here include doubling (C11-A) and the halving (C11-B) the aluminium content whilst maintaining the chemical balance and by re-calculating a chamber temperature and pressure using the same thermo-chemical program mentioned in section 3.1.1. Further changes include changing the rocket motor binder, HTPB/Isophoron to CTPB/Isophoron (C11-C) and finally to polyester (C11-D). These modifications and the effects on the re-calculated temperatures and pressures are shown in Table A-19. The overall changes to the input features are shown in Table 4-8.

The results from the global changes made to the input feature of C11 are shown in Fig. 4-54. In C11-A the increase in Al causes an increase in calculated chamber temperature to 2980K as expected. The spectral irradiance is therefore expected to increase, as predicted. In C11-B the halving of the aluminium composition leads to a decrease in chamber temperature of 2820K. This is a small decrease of 40K and a less dramatic decrease in spectral radiance is expected and also predicted by the model.

The binder changes in C11-C and C11-D effectively change ratios of the elements relative to each other and therefore the combustion products in the plume. The use of binder CTPB does not bring about large changes in expected chamber temperature and compositions. The spectral irradiance is therefore not expected to change significantly. The prediction matches this expectation. This is an improvement on the result achieved by the feed-forward neural network model built by Roodt [1998]. The use of polyester (C11-D) has a greater effect on the elemental ratios and an even greater effect on the temperature, which increases to 3050K (see Table 4-8). The amount of Al is only slightly increased. The expectation is a slight increase in spectral irradiance. The prediction matches the expectation, which is also an improvement on the result achieved by Roodt [1998] where a decrease is predicted.

Table 4-8 The overall changes to the input feature of rocket C11 after making global changes to the motor design.

	C11	C11-A	C11-B	C11-C	C11-D
C	1.047	1.0474	1.0474	1.0528	0.8551
H	4.382	4.382	4.4505	4.3833	3.9587
O	2.796	2.66	2.8642	2.792	2.9775
N	0.701	0.6672	0.7183	0.7013	0.6894
Al	0.148	0.2965	0.0741	0.1482	0.1482
K	0	0	0	0	0
F	0	0	0	0	0
Cu	0	0	0	0	0
Pb	0	0	0	0	0
S	0	0	0	0	0
Cl	0.689	0.6553	0.7064	0.6894	0.6894
Si	0	0	0	0	0
Ti	0	0	0	0	0
Fe	0	0	0	0	0
T_c	2.86	2.98	2.82	2.87	3.05
P_c	1	1.23	1.23	1.23	1.23
E_c	4.5	4.5	4.5	4.5	4.5
D_T	15	15	15	15	15

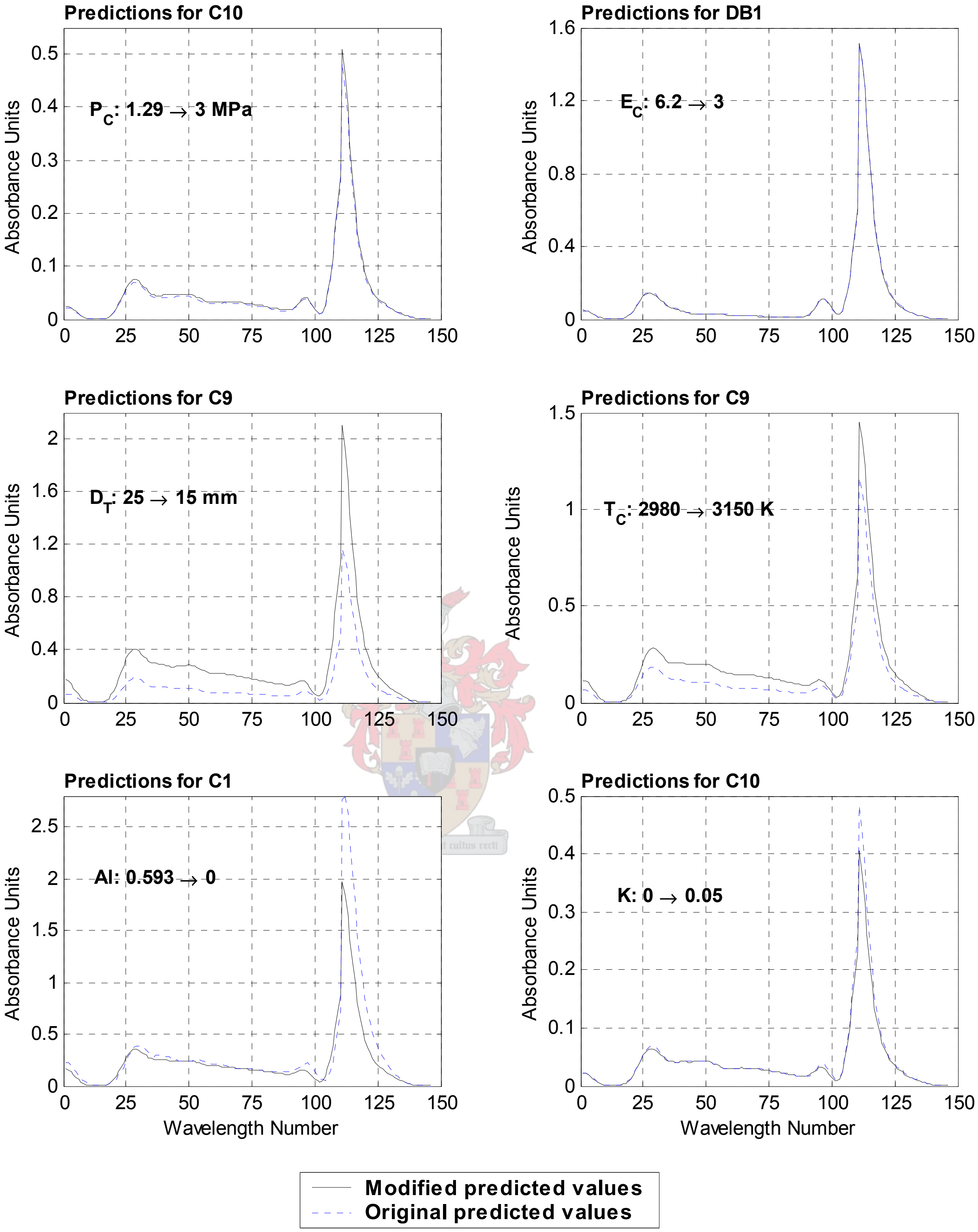

Fig. 4-53 The resulting predictions from the overall NNPLS model after the indicated modifications have been made to the input features.

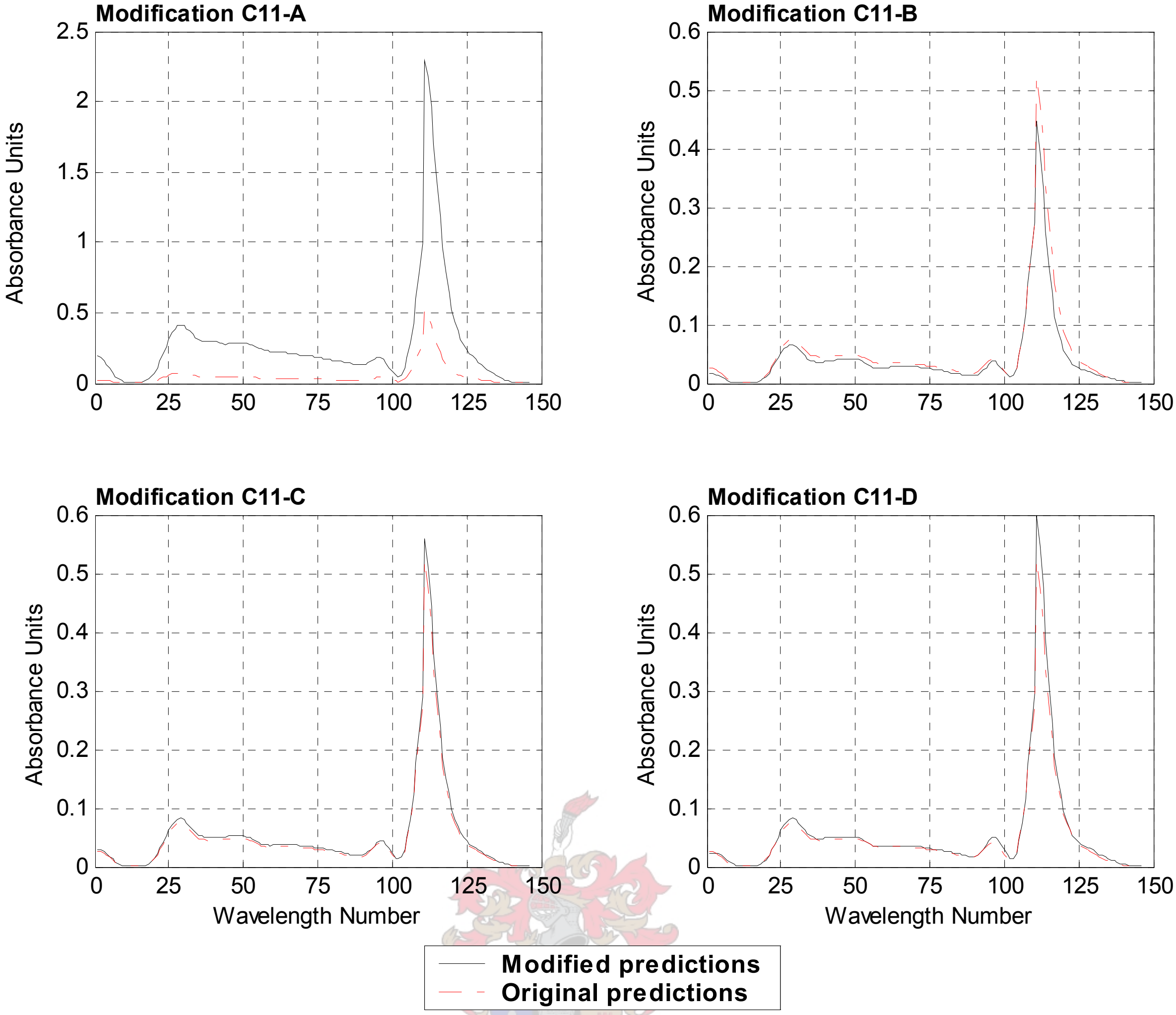

Fig. 4-54 The predictions made from making global changes to C11 using the overall NNPLS model.

Chapter 5

Predictions of the Rocket Motor Design Parameters

The prediction of the design parameters from the rocket irradiance emission spectra as independent predictors has already been defined as the 'reverse' modelling problem in this text. The reverse problem poses the problem that the measurement noise is contained within the input data set. This makes it difficult to identify the correct representative of an independent data point as model input. The application of dimensional reduction, using PLS and PCA regression, has been extensively applied for such related problems. These techniques are especially applied in various spectrometric fields, such as near infrared (NIR: 500 nm to 2000 nm) spectrometry, in order to enhance robustness for calibrating the measuring equipment.

5.1 Methods of Model-building and Validation

The developments of the applications of principal components regression and partial least squares (PLS) from classical least squares are described in section 2.13.1. From the myriad of modelling applications these techniques are especially suited for calibrating measured spectra in the NIR absorption band of a solution for specie identification. The applications of these techniques to NIR modelling, as well as neural networks and PCA neural networks have been published in the works done by Blanco [1999], Blanco [2000], Martens [1989] and Brereton [1990]. The use of the modelling techniques used for the forward modelling problem can therefore justifiably be extended to the reverse problem.

In the NIR absorption band there are a large number of correlations between the absorbances for the various wavelengths. In addition, there is a significant amount of measurement noise in the data. The data analysis in Chapter 3 shows that these effects are also found for the emission spectra measured in the middle IR band for the fired rocket motors. The high degree of correlations between the spectral measurements is well illustrated in Fig. 3-8.

It is therefore expected that there is a significant amount of dimensional reduction in the input space. This is an important motivation to proceed with the application of the above modelling techniques. The reason is that the data, which is to be modelled using 146 input variables and only 18 independent data points, would otherwise be grossly sparse and insignificant. The models above allow for the possibility of reducing the dimension of the input space significantly enough in order to be able to build a model with adequate remaining degrees of freedom for lack-of-fit (see section 3.1.3).

In the reverse problem the actual dependent variables become the independent variables. The repeat spectral measurements do not add any significant meaning to the input space from a modelling point of view. This is because for each spectrum from a certain rocket motor type the corresponding output is identical to the others. This is more synonymous to a pattern recognition problem. The pure error component of the residual variance is zero, thus there is no difference in the residual variance and model predictive ability. The value predicted for data point, y_i is the expected value ($E(y_i)$) for the point (see section 2.2). The multiple predictions made for such a point if the repeat measurements are included in the input set do not satisfy this. The repeat measurements can be used to observe the deviations in the predictions made, which is a form of model validation.

The repeat measurements of the emission spectra are therefore removed and a single representative spectrum for each rocket motor is used in the input space for building a model. The eighteen representative spectra are chosen to be the mean absorption values over all repeat measurements for a specific rocket motor type for all wavelengths in the IR band.

The linear PLS model, neural network PLS and radial-basis function PLS techniques are applied to the reverse modelling of the data. The discussion in section 4.1.1 motivates the applications of these data-fitting techniques, as well as the methods of implementation. All PLS models are built using 20 latent dimensions. The inner neural network training algorithms are the same as discussed in section 4.1.1. The discussion in section 2.13 explains the superiority of the PLS methodology over the PCA regression methodology. The PCA regression technique is therefore not investigated here.

The feed-forward neural network is trained using the same resilient propagation algorithm (RPROP) discussed in section 4.1.1. The Levenberg-Marquardt algorithm is often the preferred back-propagation training algorithm. However, it is too memory intensive for the standard personal computer of today when training such a largely multivariate data set. The transfer functions are the hyperbolic-tan and purely linear functions for the hidden and output layers respectively.

The discussion in section 4.1.2 motivates a method of cross-validation. Leave-one-out cross-validation is also applied to the reverse modelling problem. The disadvantage of cross-validation is that most of the data points left out during a training session may have one or more variables outside the range of values used for training. If the values outside the training ranges do not greatly differ from the closest value in the training range the ability of the model to predict the variable(s) is not too significantly affected.

The modified cross-validation algorithm motivated and proposed in section 4.1.2 for the neural network PLS algorithm is applied to the reverse modelling problem. From the modelling results in section 4.2.3 it is sensible not to compare the modified method to the original cross-validation.

5.2 Results and Discussion of Candidate Models

The statistics scores of PRESS and SSEP are evaluated in the same manner for the 18 unseen rockets from 18 cross-validation sessions and the overall model, as for the forward modelling problem in Chapter 4 . There are only n=18 total number of training data points for cross-validation and the overall model. There are therefore no repeat measurements to calculate PRESS$_{min}$ (or SSEP$_{min}$) and R^2_{max} . Due to the varying nature and dimensions of the output variables it becomes sensible to separately evaluate the optimum model complexity for each variable.

The root mean-square error (RMSE) and the square root of MSECV (RMSECV) are calculated for each individual output variable in order to obtain a measure of the standard deviation of the error of prediction.

In the forward problem the chamber pressure is inverted in order to obtain the correct qualitative tendencies in the spectral responses. The reverse modelling problem is evaluated without this manipulation.

5.2.1 Linear PLS

The input and output variables are standardised prior to training. The statistics scores are all calculated using the re-scaled output and input variables. It is found that it is not feasible to test the PLS model beyond 20 latent dimensions.

The PRESS-values up to latent dimension 15 are plotted in Fig. 5-1. The PRESS-values become exceedingly high at higher latent dimensions. Fig. 5-1 and Table B-1 show that 18 latent dimensions is the maximum achievable dimension, even though the input space consists of 146 dimensions. This can be attributed to the fact that the data row space only consists of 18 dimensions (or 17 dimensions in the case of each cross-validation session).

Fig. 5-1 further shows that overall the optimum model complexity lies at 1 or 2 latent dimensions. This represents a large dimensional reduction in the input space, which is expected when considering the high degree of correlation in the input space (see Fig. 3-8). The low dimension is the desired result, as this usually leads to a lower model degree of freedom and therefore a more credible model for the sparse data set. The average pdf-value for 2 latent dimensions is 3.75 (see Table B-1). This value is encouraging, as in theory it means there are about 14 lack-of-fit degrees of freedom left to check the validity of the model.

The average correlation coefficient between the unseen predicted outputs from cross-validation and the target values (R^2_{cv}) for 2 latent dimensions is low at 0.183. There appears to be no relationship between the PRESS-values and the average R^2_{cv} –values with increasing LD. A trend of decreasing R^2_{cv} –value with increasing PRESS-value is expected, as is the case for the forward problem. A reason for this is that the optimum complexities for the different output variables lie at different latent dimensions.

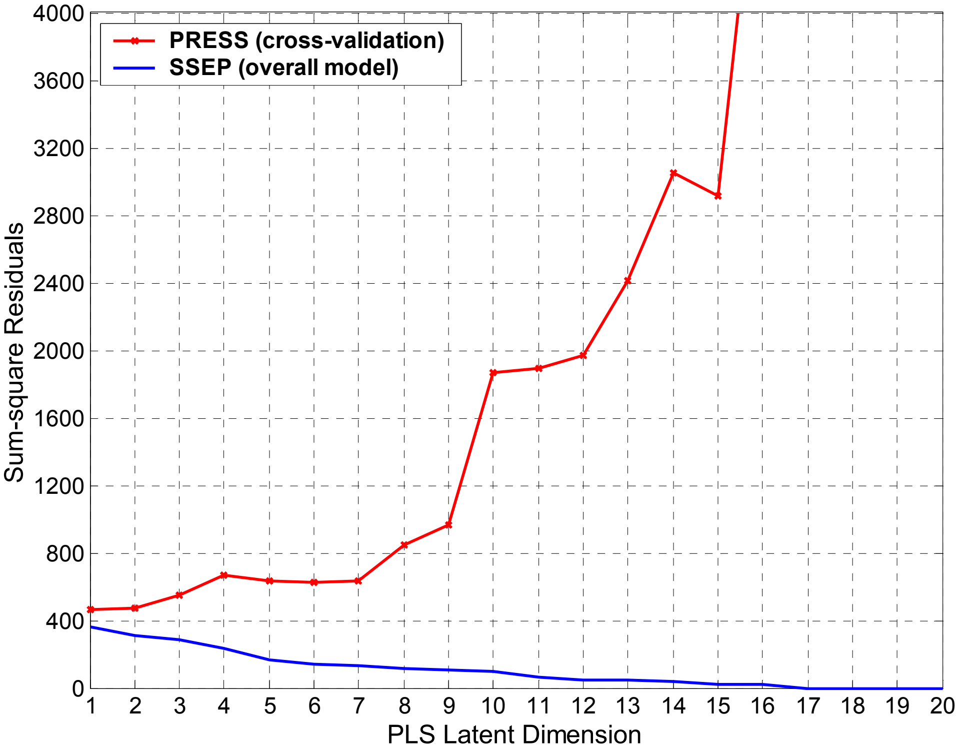

Fig. 5-1 The sum-squared residuals obtained from building a linear PLS model for the reverse modelling problem.

The X-block explained variance for the overall model is high at 95.25% for 2 latent dimensions, however the Y-block variance is poor at 26.85% (see Table B-1). A large amount of information is captured by the model in the input space, however it appears the model is not able to adequately capture the overall output variance.

The PRESS- and SSEP-values shown thus far represent the pooled performances across all 18 output variables. In contrast to the forward problem in Chapter 4 the output variables here are not all of the same units and there is not such a high degree of correlations between them (see Fig. 3-7). A significant degree of variability in optimum model complexity is therefore expected for each output variable. For this reason the optimum set (OPT) of statistics scores and predictions is obtained for each output variable. The optimum model complexity for each output variable, y_j is determined by evaluating at which LD its minimum $PRESS_j$-value lies. These optima are further chosen in such a way that the minimum $PRESS_j$-values be obtained at latent dimensions where the pdf-values do not exceed 15. The summary of these results for the overall models is shown in Table B-2. The results show that the optimum latent dimensions for the output variables vary greatly. The average values for the number of parameters, pdf, latent dimensions and correlations are comparable to an overall model of 3 to 4 latent dimensions (if it were possible).

In Fig. 5-2 the R^2_{cv} –values are plotted for the optimum set and the models using 1 to 3 latent dimensions output variables pooled together. The optimum set shows a general improvement in the correlation scores, however the low values for most of the output variables are not desirable. The model does not seem to be able to capture the variances in the hydrogen (H), silicone (Si), chamber pressure (P_C) and the throat diameter (D_T) at all.

The R^2 -values for the overall models are plotted in Fig. 5-3. An interesting result is that the overall model does not seem to be able to capture the variance of sulphur (S) at all, but that the correlation for the unseen data (cross-validation) is high. This can be attributed to the model not being able to capture the relationship between the inputs and S and that at low LD the model predicts the overall mean (bias) value for S.

The predictions for fluorine (F), titanium (Ti), iron (Fe), Si and S are expected to be inconsistent due to the large amounts of zero-entries. These inconsistencies may be the reason for the lack of correlation between increasing PRESS and decreasing R^2_{cv} –values. The lack of predictability for P_C and D_T is not surprising, as these are physical properties, which may not necessarily significantly influence the emission spectra. The lack of predictability for H is however not expected, as it is contained within the combustion product, H_2O for which there is a defined peak in the spectrum. The fair degree of predictability for potassium (K) is encouraging, as this is the important element responsible for depressing the irradiance intensity. This characteristic is important for determining the stealth characteristics of the rocket.

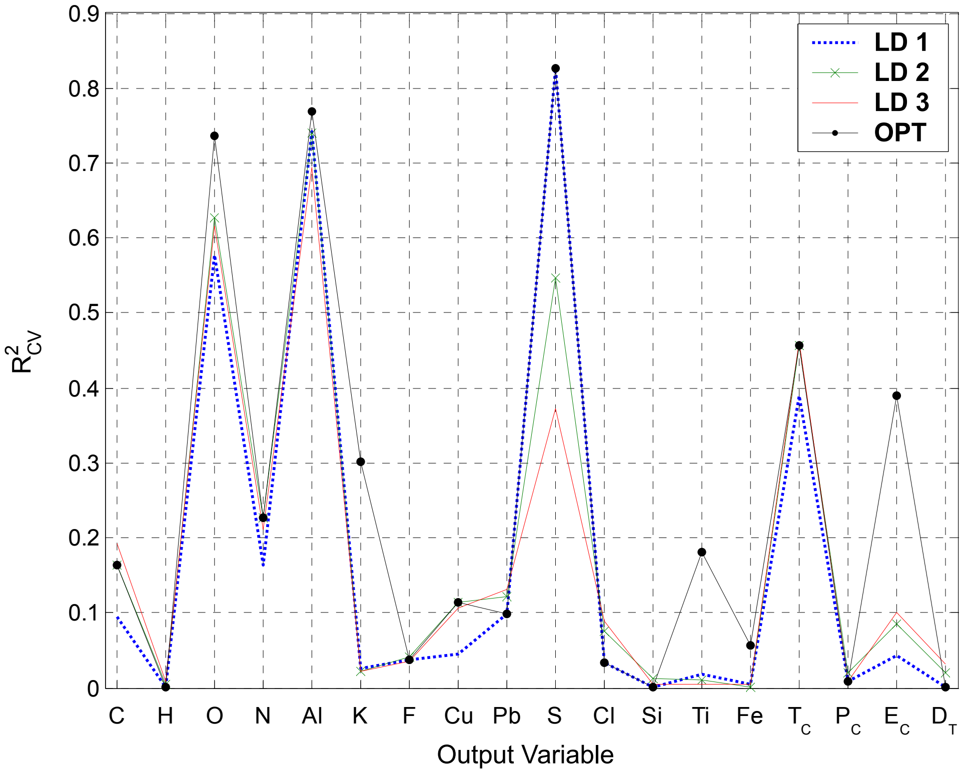

Fig. 5-2 The squared correlations between the unseen predicted rocket motor design parameters from cross-validation and their targets for linear PLS.

The pseudo degrees of freedom plotted for each output variable in Fig. 5-4 vary somewhat for a certain number of latent dimensions (all output variables pooled). The increase in predictability for oxygen (O), aluminium (Al), Ti and Fe using the optimum set with higher LD explains the sharp rise in pdf-values.

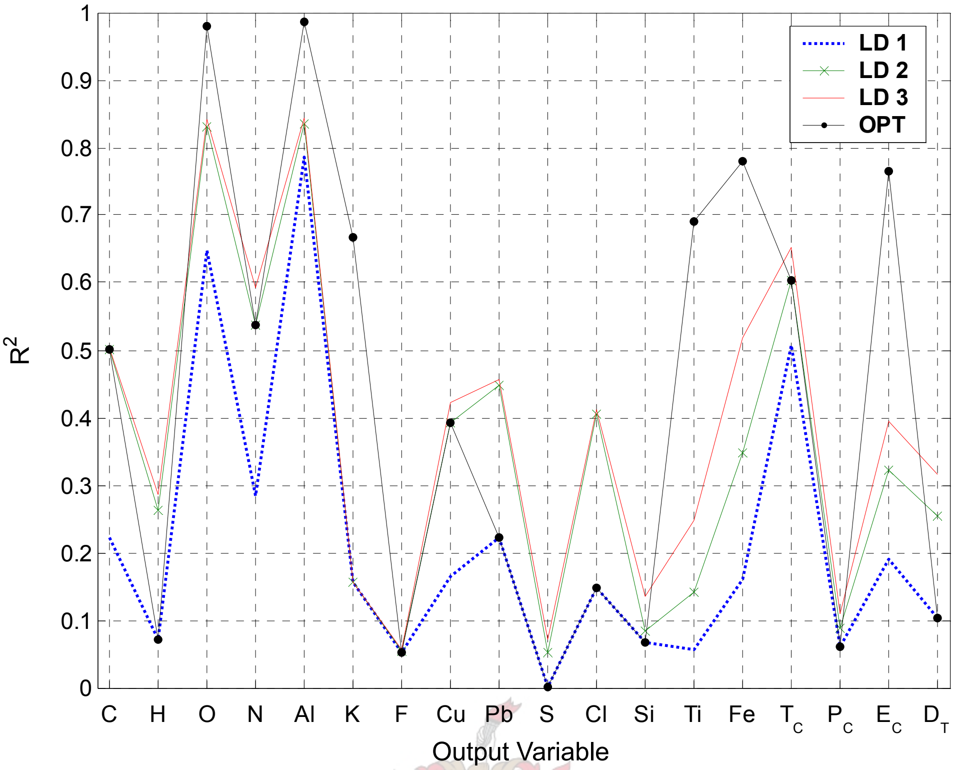

Fig. 5-3 The squared correlation coefficients between the predicted rocket motor design parameters for the overall model and their targets for linear PLS.

In Fig. 5-5 to Fig. 5-7 the unseen data point predictions are plotted for each rocket motor design together with the RMSECV band for each output variable. These predictions are plotted using the optimum set. Confidence intervals were not calculated, as they require model pseudo-degrees of freedom to be calculated. This requires that the training sets of each cross-validation session must be further cross-validated in order to satisfy equation (2-150).

The results show that the model predictions favour the C-class rocket motor designs. This is evident from the fact that there are predictions made for chlorine (Cl), where the DB-class rocket motors do not contain any Cl. The predictions for H are generally poor relative to the neighbouring variables. The E_C-value for DB1 is extremely poorly predicted. The RMSECV band for the last three output variables, i.e. the physical properties (P_C, E_C, D_T), is much broader compared to the other variables. This is due to the fact that the variables' ranges consist of higher values. Except for the chamber temperature (T_C), the other physical parameters are expected to be poorly predicted by the model following the R^2 –results obtained. For C10 and C11 the P_C-predictions are failures.

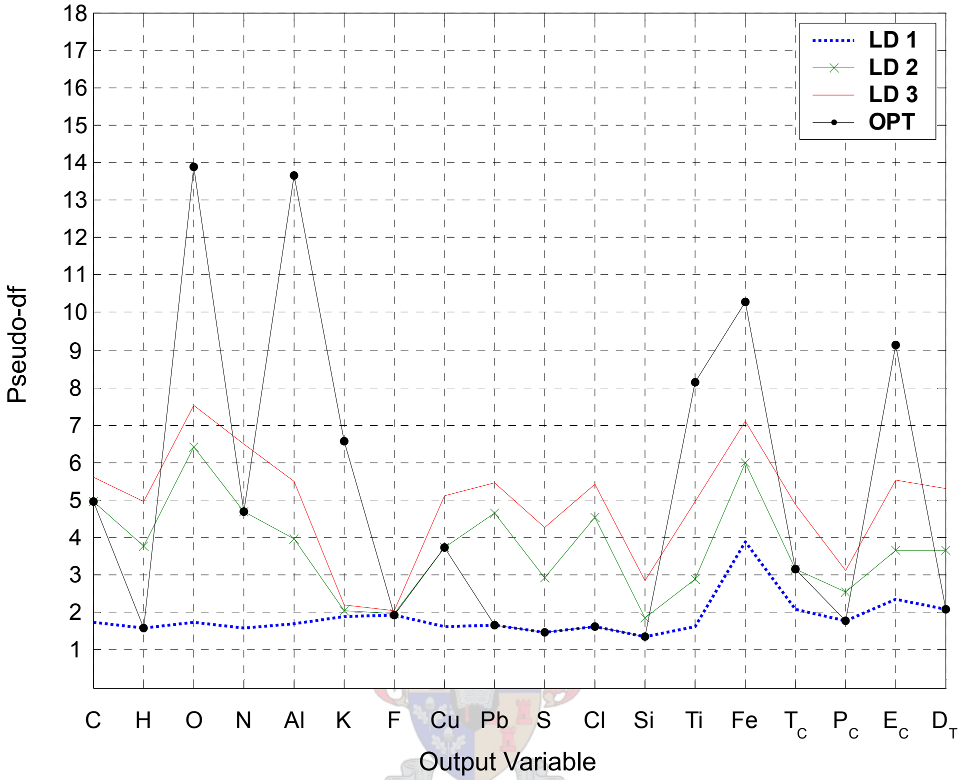

Fig. 5-4 The pseudo degrees of freedom of the linear PLS model calculated for each output variable.

The predictions as well as the 95% confidence intervals for the optimum set overall models are shown in Fig. 5-8 to Fig. 5-10. The broad confidence intervals for the last three physical design parameters and H coincide with the results obtained for the unseen predictions from cross-validation. The DB-class rocket motor design parameters are predicted more poorly compared to the C-class rocket motors. There is also the unwanted prediction of Cl for the DB-class rocket motors.

The predictions in Fig. 5-5 to Fig. 5-10 appear to be fairly good. However it must be noted that most of the output variables, especially from nitrogen (N) to

Fe have very low ranges and there are many zero entries. What appears to be a small deviation from the target value is in fact large and therefore the R^2 -values are low and the Y-block explained variance is unacceptably low.

The output scores (latent variables) for the first 6 latent dimensions are plotted against the input scores in Fig. B-1. Due to the linear relationships there are numerous target points that are far from the prediction line. Non-linear inner models could therefore possibly improve on this.

From the results above it is clear that it is difficult to decide on a definite model complexity as a representative model. From the average scores obtained for the optimum set it appears that a model with 3 or 4 latent dimensions for all output variables would be a reasonable generalisation. The optimum set can be used. Inconsistencies, however, may occur as observed in the case where the R^2 –values for the overall model and cross-validation are compared for S above. In this text the optimum set is used to make the comparisons, as it is difficult to make an obvious choice to decide on the best number of latent dimensions overall.

The plots of the linear PLS regression coefficients as described by equation (2-133) in section 2.13.1 are shown in Fig. B-2 and Fig. B-3. The plots show that the weightings at the various wavelengths are virtually mirror images for the various output variables. For example, the coefficients for Al and H show positive weightings in the band for wavelength numbers 46 to 85 in contrast to the negative weightings for C, N and O. The strongest influences on predictions appear to come from the tail end of the spectral region and the region just before the water absorbance peak (wavelength numbers 20 to 45). Ironically the forward models poorly predict these two regions. The CO_2-absorption band (wavelength numbers 100 to 140) and the intermediate band (wavelength numbers 46 to 85) bear significant weight on the predictions.

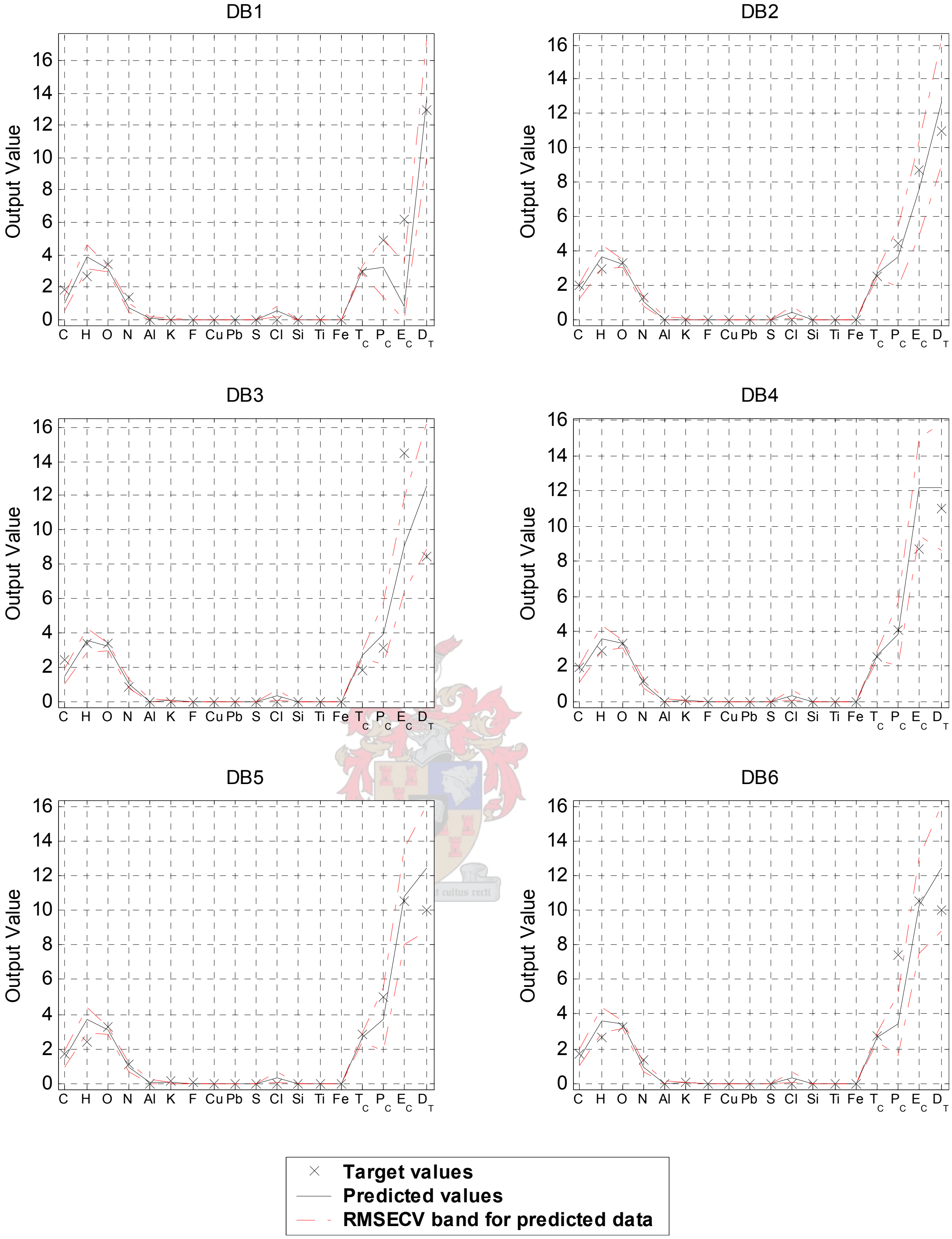

Fig. 5-5 The rocket motor parameter predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of linear PLS.

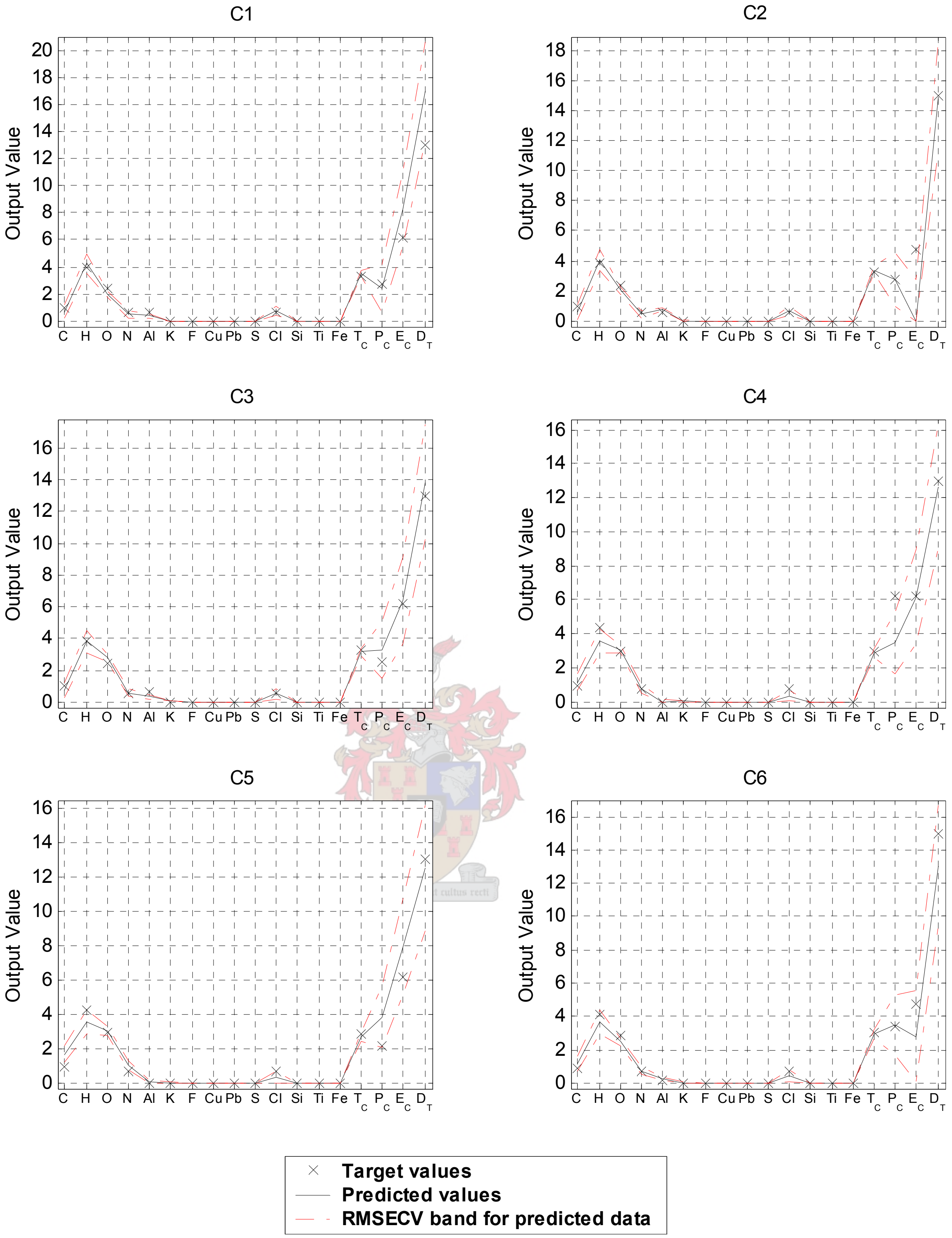

Fig. 5-6 The rocket motor parameter predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of linear PLS.

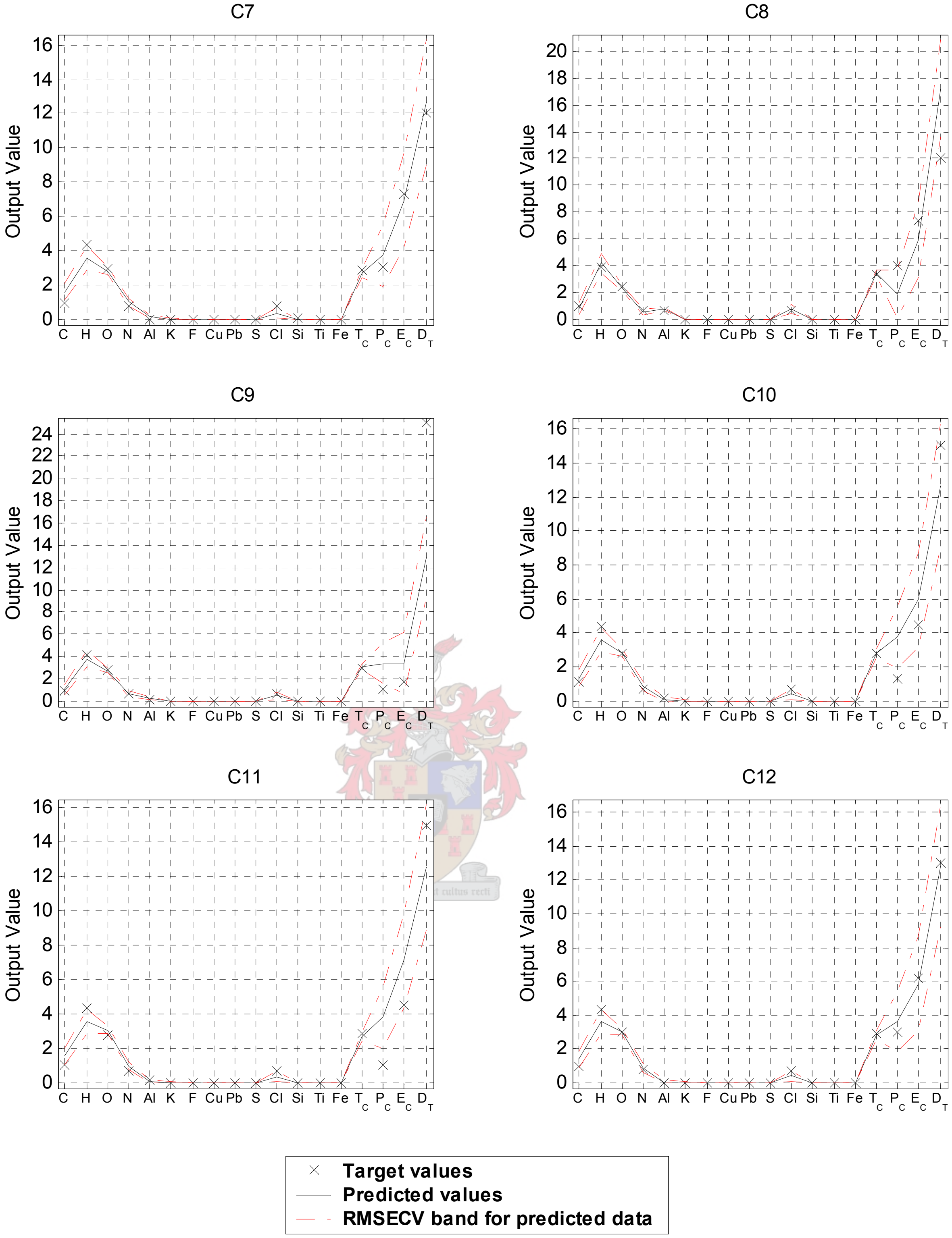

Fig. 5-7 The rocket motor parameter predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of linear PLS.

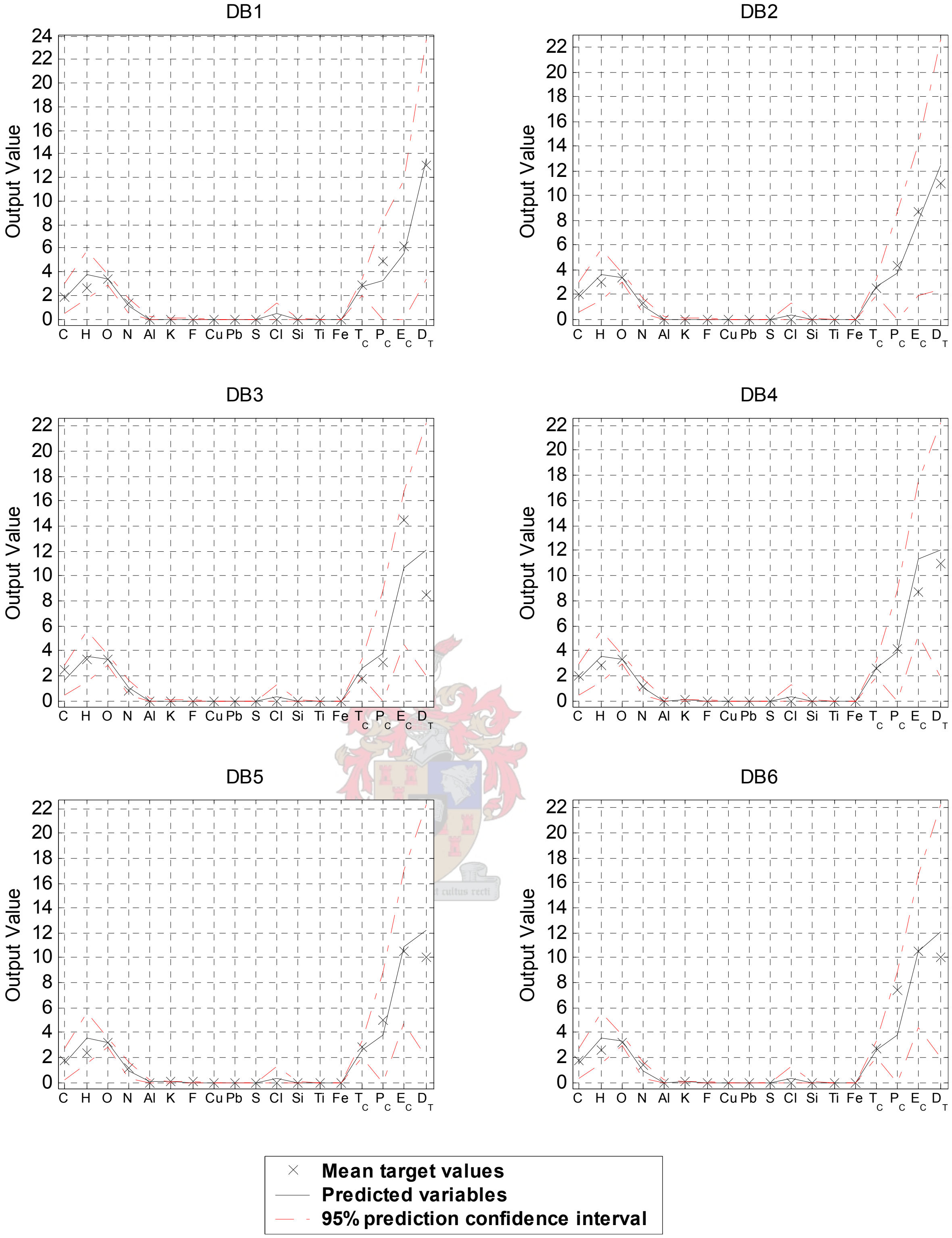

Fig. 5-8 The rocket motor parameter predictions for rocket motors DB1 to DB6 obtained for the overall linear PLS model.

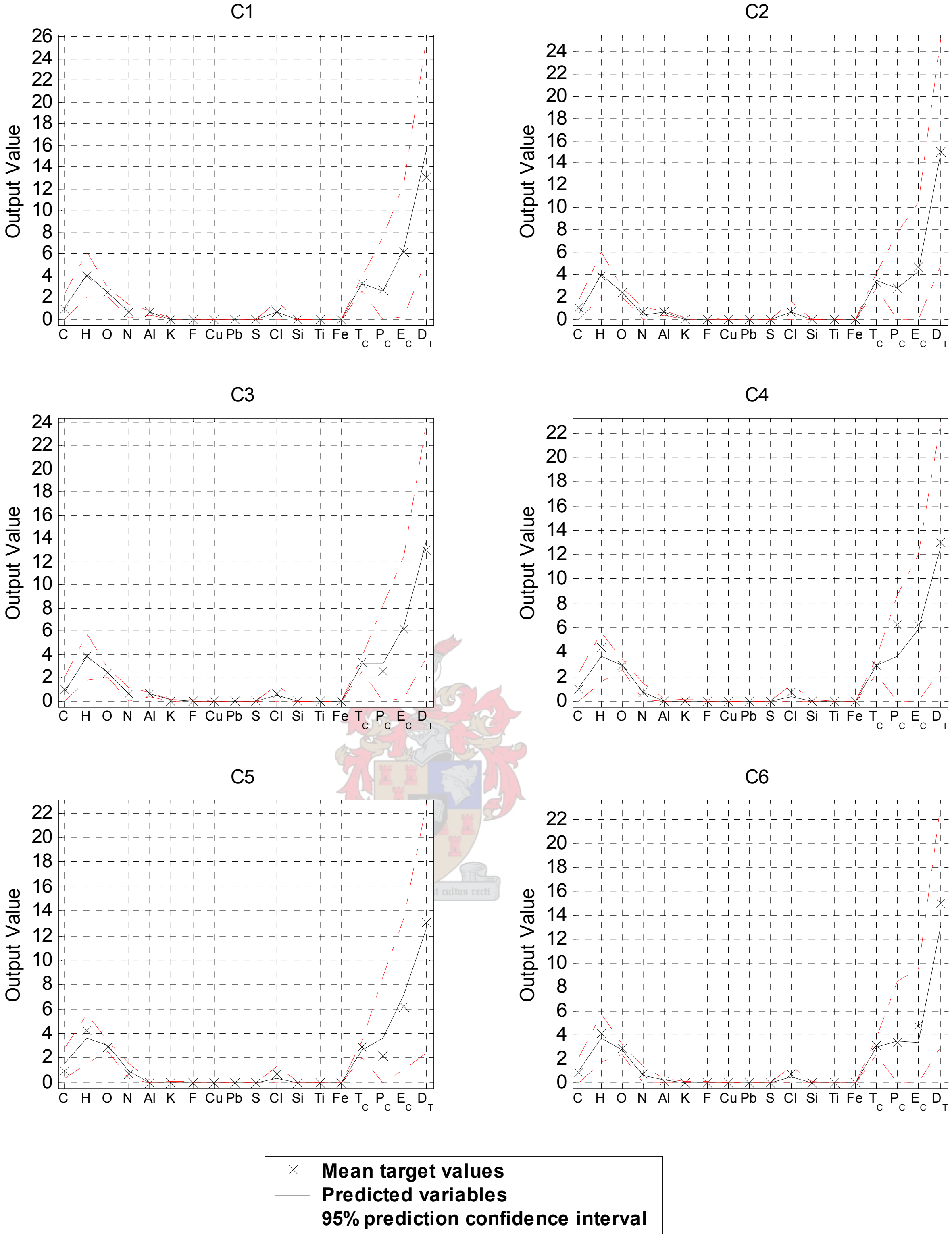

Fig. 5-9 The rocket motor parameter predictions for rocket motors C1 to C6 obtained for the overall linear PLS model.

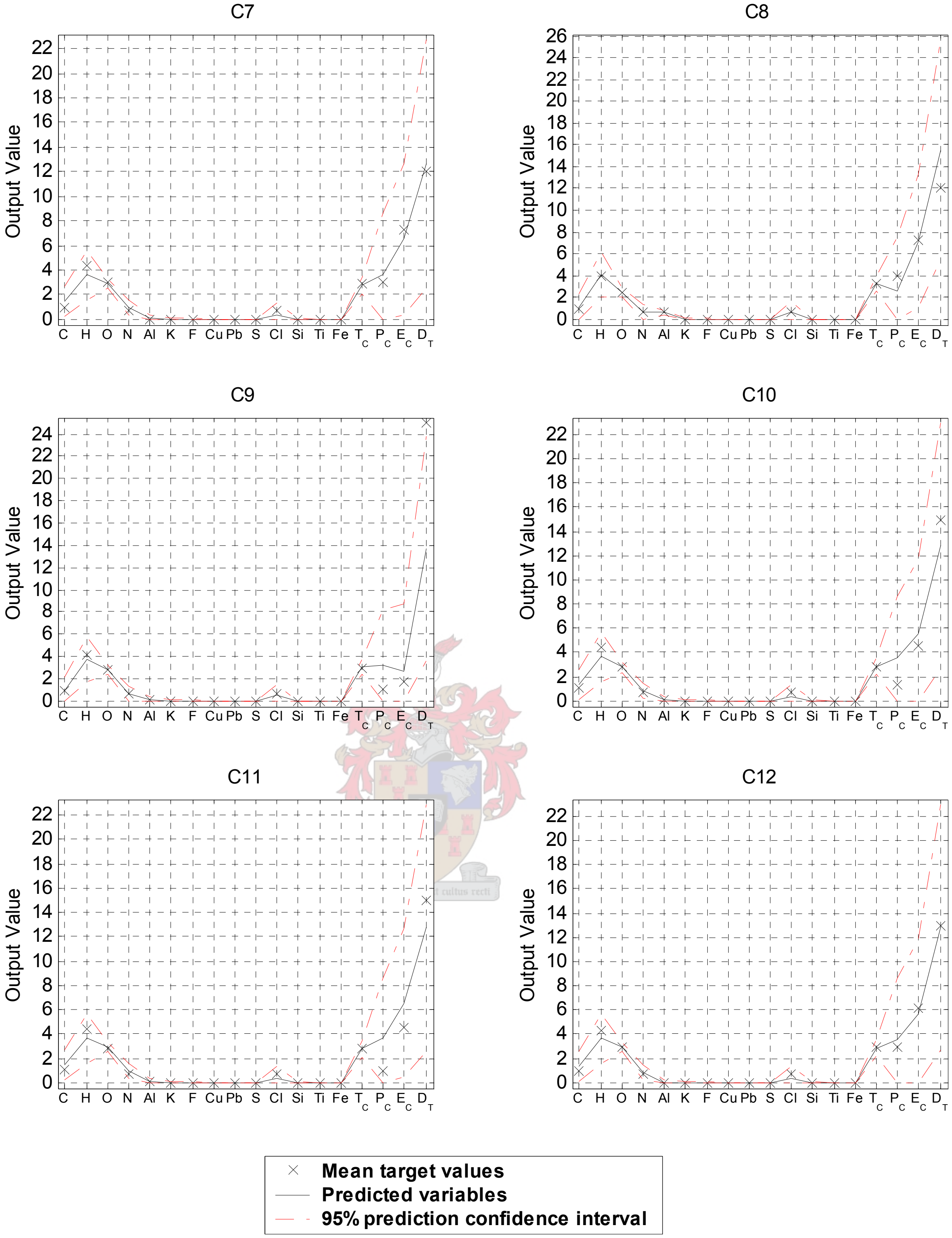

Fig. 5-10 The rocket motor parameter predictions for rocket motors C7 to C12 obtained for the overall linear PLS model.

5.2.2 Feed-forward Neural Network

The neural network was trained using a single hidden layer with 1, 2, 4, 6, 8, 10, 15, 20, 30 and 40 hidden nodes for cross-validation and the overall models. It is found that no scaling of the inputs and normalised scaling of the outputs (as in section 2.4.3) produce the best results for training the data. The box-and-whisker plot in Fig. 5-11 is plotted using the results from 10 repeat runs as shown in Table B-4. The plot shows that the optimum model complexity for all eighteen output variables overall is a single node or two nodes in the hidden layer. The PRESS-values here therefore show that, as is the case for linear PLS, the feed-forward neural network favours very low model complexities.

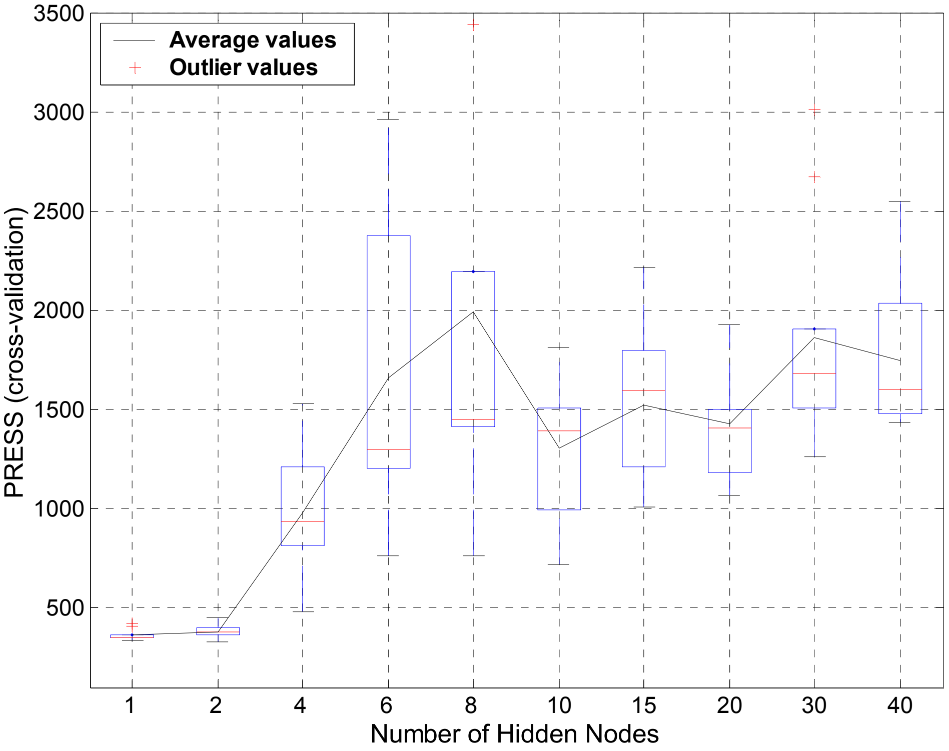

Fig. 5-11 A box-and-whisker plot of PRESS-values obtained from 10 cross-validation repetitions with increasing neural network complexity.

The first cross-validation run is chosen to be the best run on which further statistics scores are calculated. The aim here is to compare the various models by their best performances in order to extract the maximum potentials of each modelling technique for considering further modelling applications on this data. In section 4.2.2 it is mentioned that the aim is also to compare model complexities and therefore the feed-forward neural networks are over-trained in order to compare complexities based on the parameterisation.

It is found that there is a significant difference in over-training limits (flattening of the sum-square error vs. epochs plot) as the model complexities increase. For example, for 1 hidden node over-training already occurs after 500 epochs and for 10 to 40 hidden nodes over-training can occur anywhere between 5000 and 10000 epochs. For this reason each training session for cross-validation and the overall models is trained using 15000 epochs to ensure over-training. The training of the overall models for all 10 candidate model complexities were repeated five times (see Table B-5). Run 3 is chosen to be the best run and the training plots for this run are shown in Fig. 5-12. The plots for 15 to 40 hidden nodes are omitted, as they are not distinguishable from the plot for 10 hidden nodes.

The summary of pooled results for all output variables is shown in Table B-6 using the results form the best runs. The minimum PRESS-value at 2 hidden nodes corresponds to the maximum average R^2_{cv} –value of 0.351. The Y-block explained variance is higher at 41.49% compared to the linear PLS model with 2 or 3 latent dimensions. The average pdf-value of 7.07 for all output variables is also higher. It is interesting to note that the feed-forward neural network now performs considerably better compared to the linear PLS model. This was not the case for the forward model. A reason for this is that there are considerably fewer output variables to deal with during training of the reverse model. This is reflected by the relatively smaller difference in total number of parameters per neural network complexity and the total number of parameters per output variable per network (see Table B-6).

The optimum set (OPT) is obtained in similar fashion to that obtained for linear PLS. The minimum $PRESS_j$-value for each individual output variable is used to determine the optimum model complexity for the particular variable. The results are presented in Table B-7. The results show that the optimum numbers of hidden nodes for most of the output variables are 1, 2 and 6. The total PRESS-, SSEP-, average R^2_{cv} – and average R^2 –values show little improvement over the model with 2 hidden nodes pooled. The average number of hidden nodes of 2.11 verifies that there is little improvement in finding the best model complexity for each individual variable as opposed to the overall analysis of 2 hidden nodes per output variable.

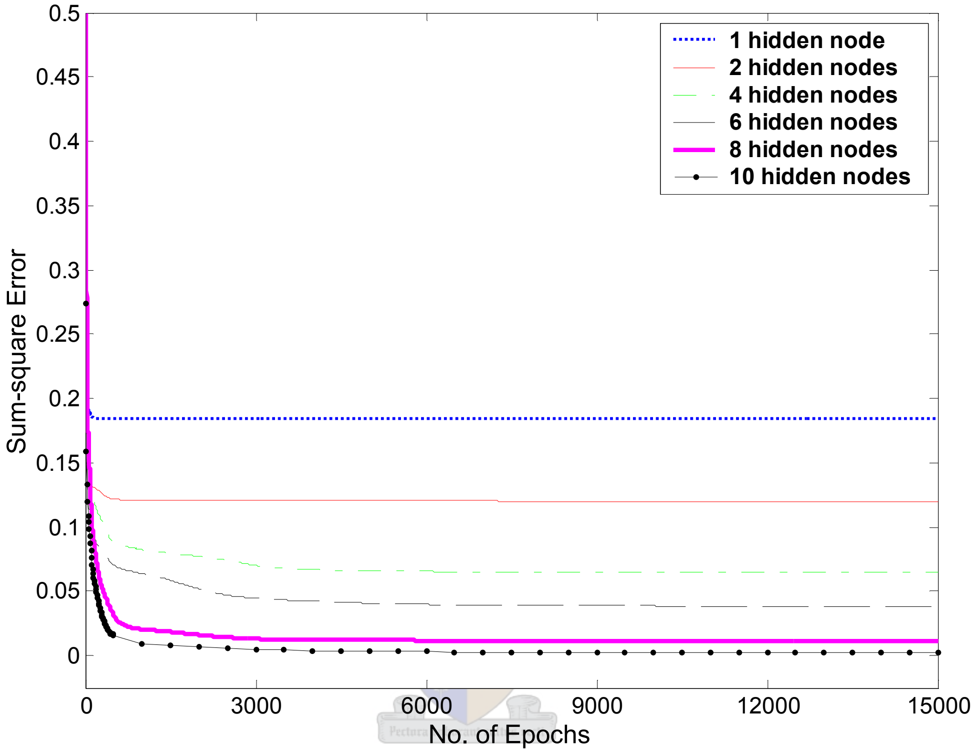

Fig. 5-12 An illustration of how the various neural network models have been over-trained by observing results from training run 3 of the overall model (evaluated on scaled values).

The fact that the model with 2 hidden nodes best predicts most of the unseen output variables can be seen in Fig. 5-13. The predictabilities of S, Fe and F are still virtually zero. This can be expected due to the large number of zero entries for the different rocket motors. It is encouraging to see that H and Cl are now much better predicted with R^2_{cv} –values above 0.5. These results prove that there are definite non-linearities in the functional relationships between the input and output data sets. The predictabilities for Si, P_C, E_C, and D_T also show significant improvements. There are 7 out of the 18 variables with R^2_{cv} –values above 0.5 as opposed to the 3 for linear PLS.

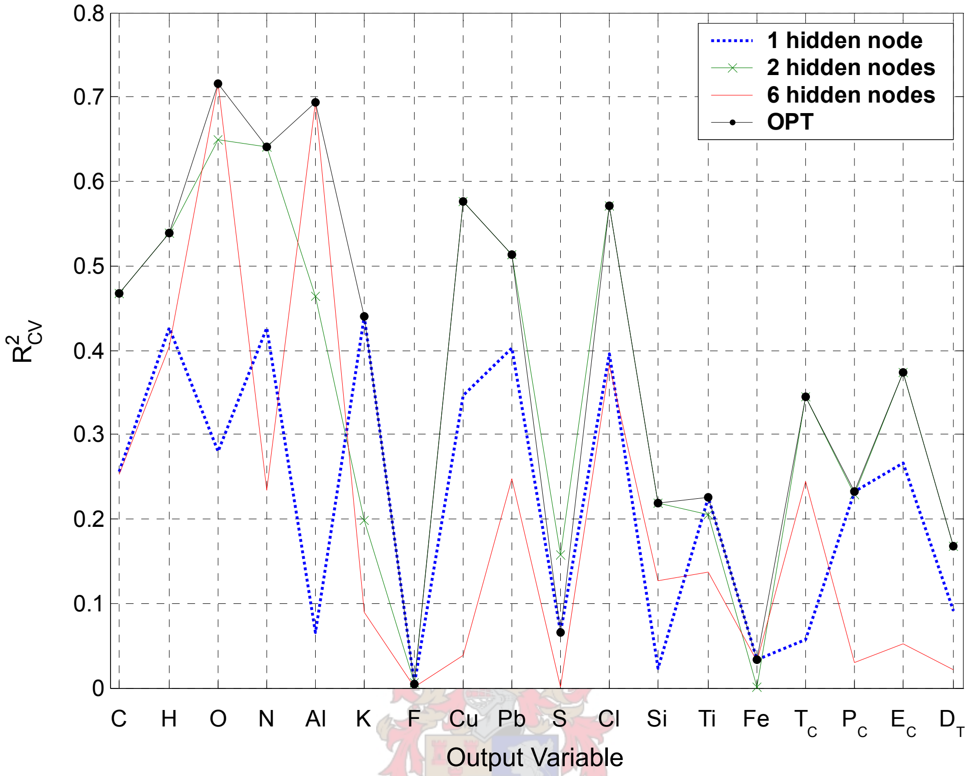

Fig. 5-13 The squared correlations between the unseen predicted rocket motor design parameters from cross-validation and their targets for feed-forward neural networks.

The overall model R^2 –values are plotted in Fig. 5-14. The predictabilities for the various output variables follow the same pattern as for the unseen data from cross-validation (R^2_{cv}). The results for C, H, O, N, Al, Pb, Cl and Si are desired with R^2 –values above 0.8. It is also encouraging to see that the physical properties, P_C, E_C, and D_T are better predicted with E_C reaching the R^2 –value of 0.5. The improved predictabilities for Pb, Cu and Cl make it possible for the model to better compensate for both the DB- and C-class rocket motors, as Pb and Cu are limited to the DB-class rockets only. The extremely low R^2 –values for S, F and Fe have a large effect on the average R^2 –values and it is therefore more useful to judge the performance of the models by the individual output variables.

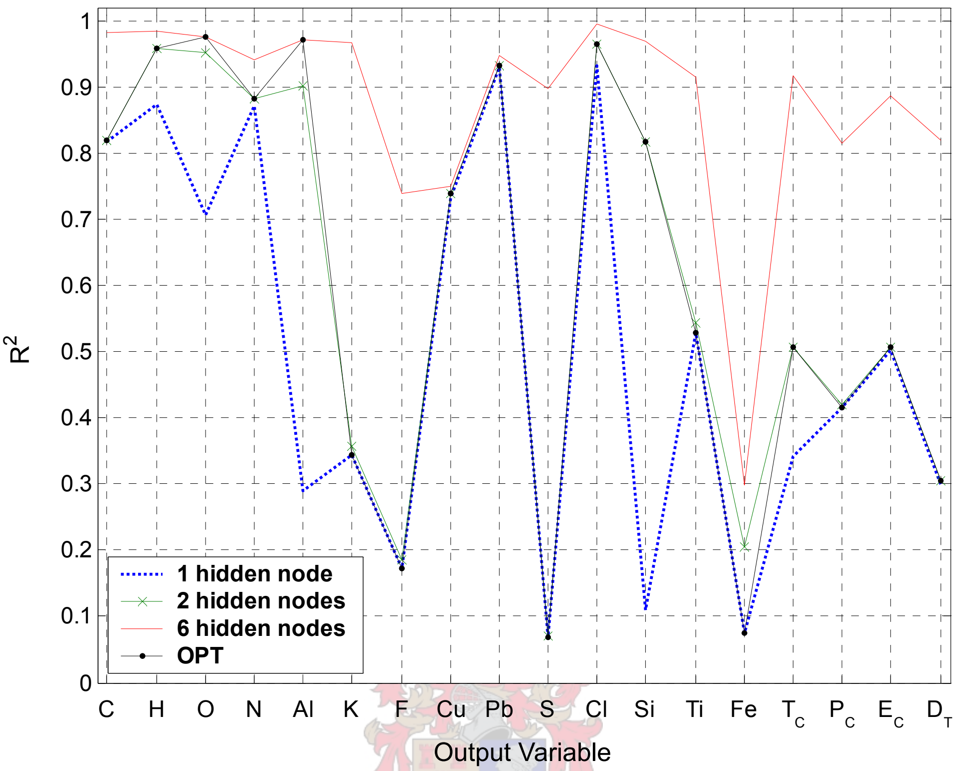

Fig. 5-14 The squared correlation coefficients between the predicted rocket motor design parameters for the overall model and their targets for feed-forward neural networks.

There is greater variability in the pdf-values (all below the value of 15) among the output variables when compared to linear PLS. In Fig. 5-15 it can be seen how pdf can vary between zero and 14 for a certain neural network architecture. The high pdf-values explain the increased predictabilities obtained for H, O, Al, Pb and Cl. The high degree of variability can be attributed to the non-linearity of the model capturing different functional relationships for the various output variables. In Table B-6 it can be seen that the maximum average pdf-value of 18 is achieved with between 20 and 30 hidden nodes in the neural network model.

In Fig. 5-16 to Fig. 5-18 the unseen data point predictions are plotted for each rocket motor design together with the RMSECV band for each output variable. These predictions are plotted using the optimum set (OPT). The predictions do not appear as biased towards the C-class rocket motors as is the case for linear PLS. It can be observed that except for DB1 the predictions for Cl in the DB-class rocket motors are virtually zero. The improvements in predictions for the DB-class rocket motors are expected following the discussion for the R^2_{cv} – values regarding Pb, Cu and Cl above. The predictions for H and the physical properties, P_C, E_C, and D_T also appear to be improved.

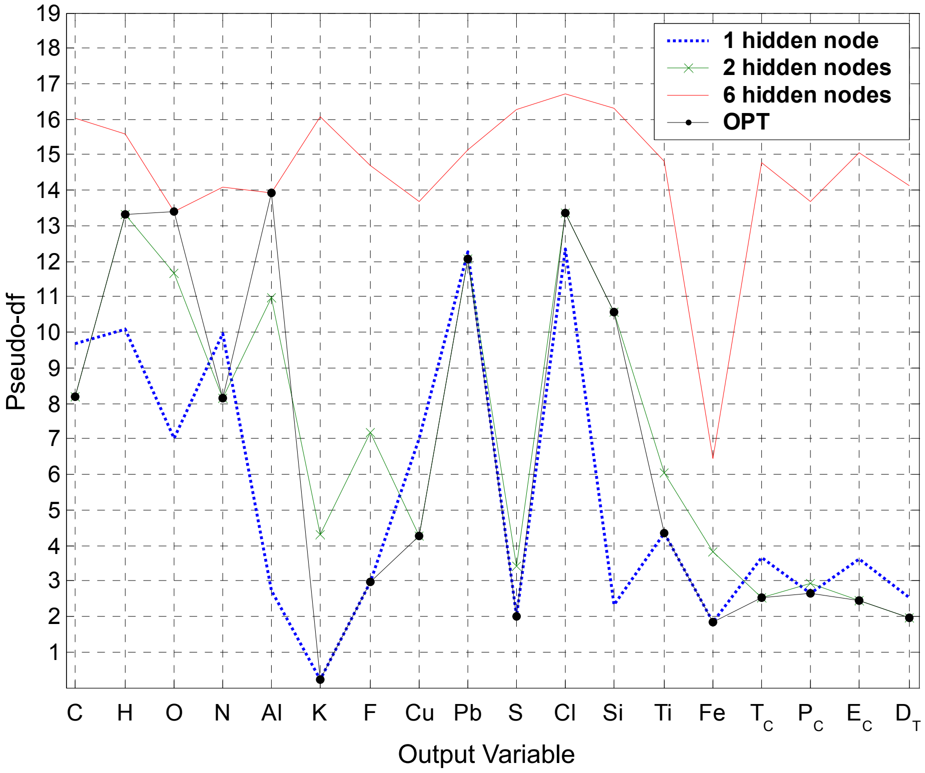

Fig. 5-15 The pseudo degrees of freedom of the feed-forward neural network models calculated for each output variable.

The predictions as well as the 95% confidence intervals for the optimum set overall models are shown in Fig. 5-19 to Fig. 5-21. The fact that the models predict almost zero Cl in the DB-class and the narrower confidence intervals prior to P_C, E_C, and D_T is encouraging. The non-linearity in the neural network model clearly introduces an improvement over the linear PLS model.

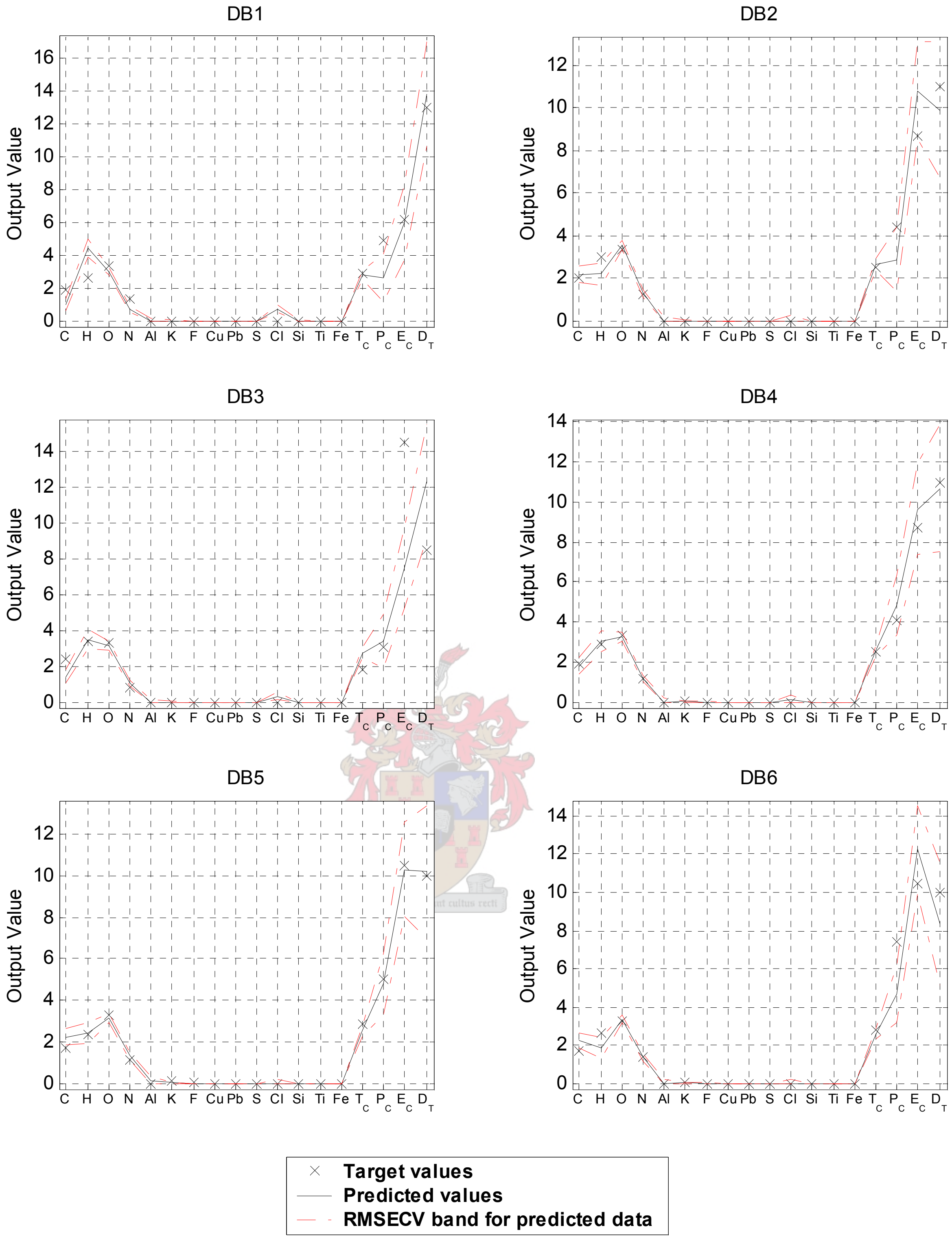

Fig. 5-16 The rocket motor parameter predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of the optimum set feed-forward neural networks.

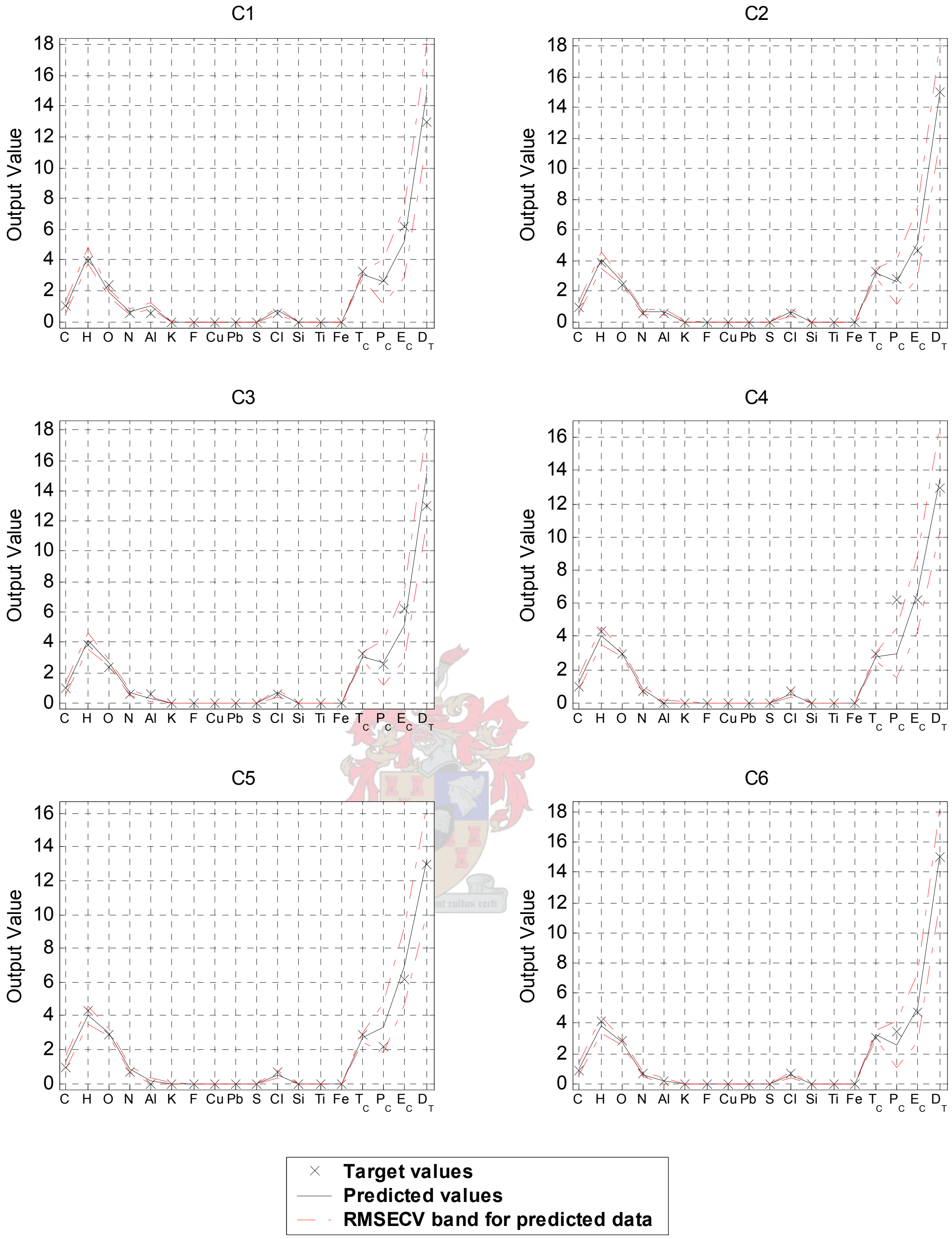

Fig. 5-17 The rocket motor parameter predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of the optimum set feed-forward neural networks.

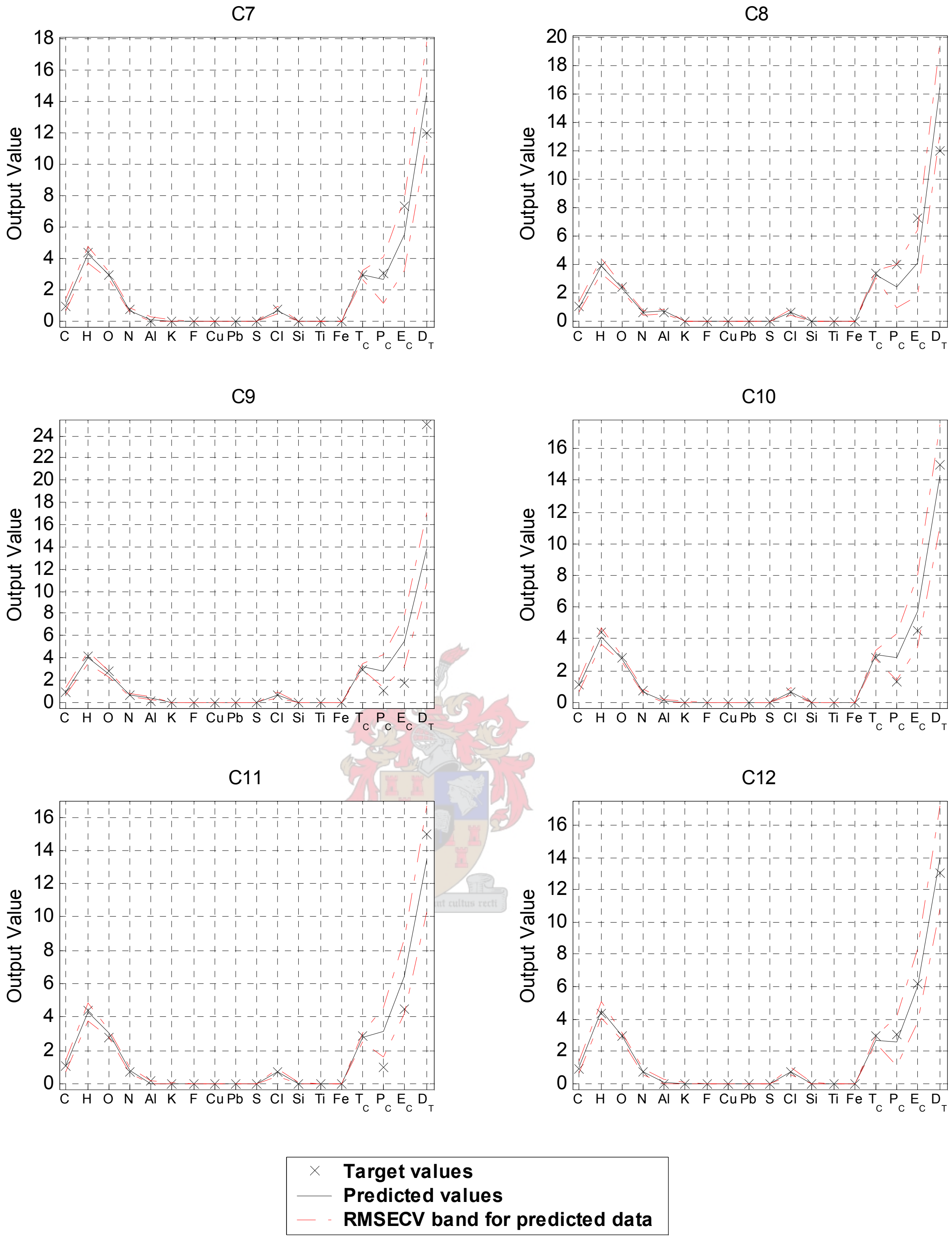

Fig. 5-18 The rocket motor parameter predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of the optimum set feed-forward neural networks.

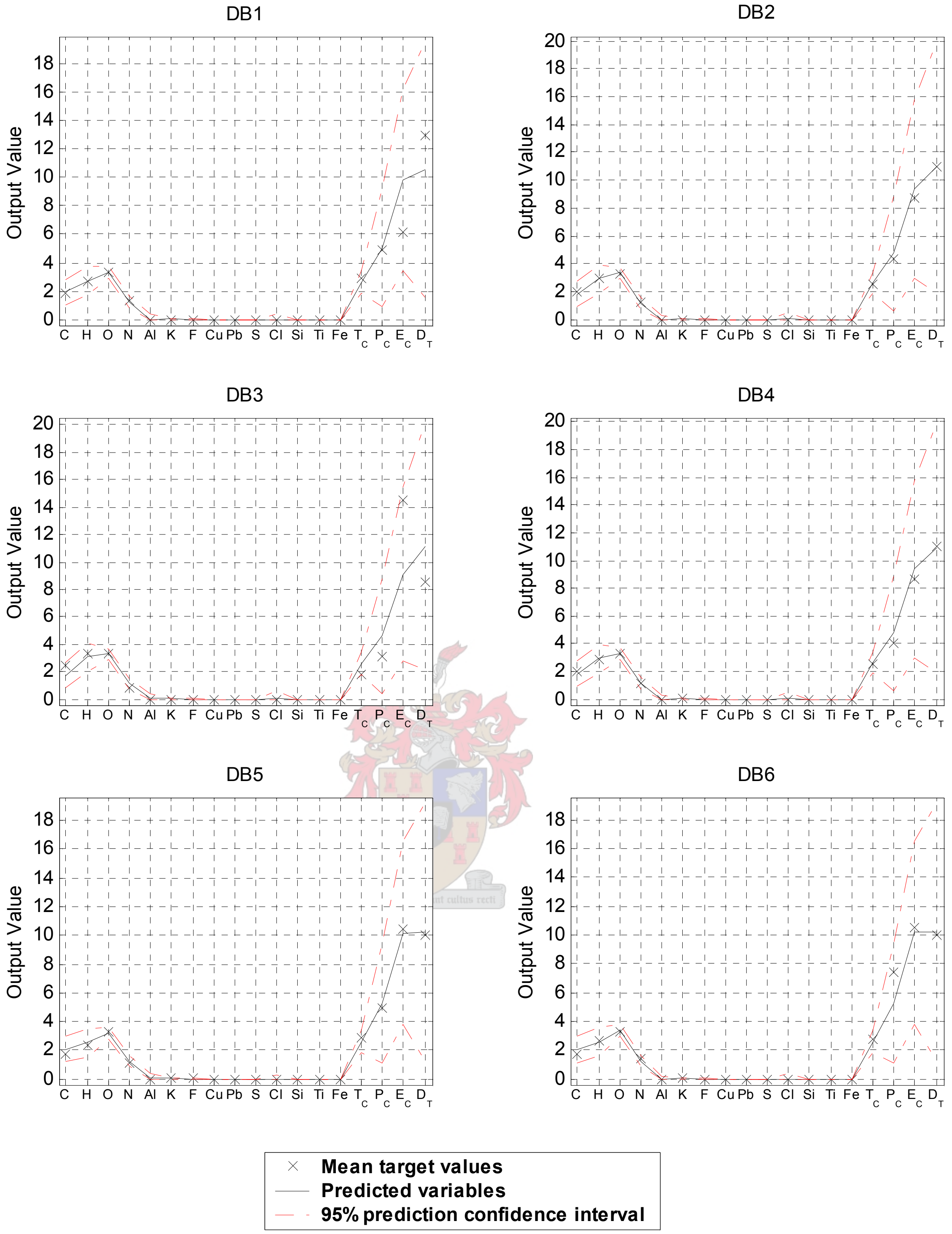

Fig. 5-19 The rocket motor parameter predictions for rocket motors DB1 to DB6 obtained for the overall optimum set feed-forward neural network models.

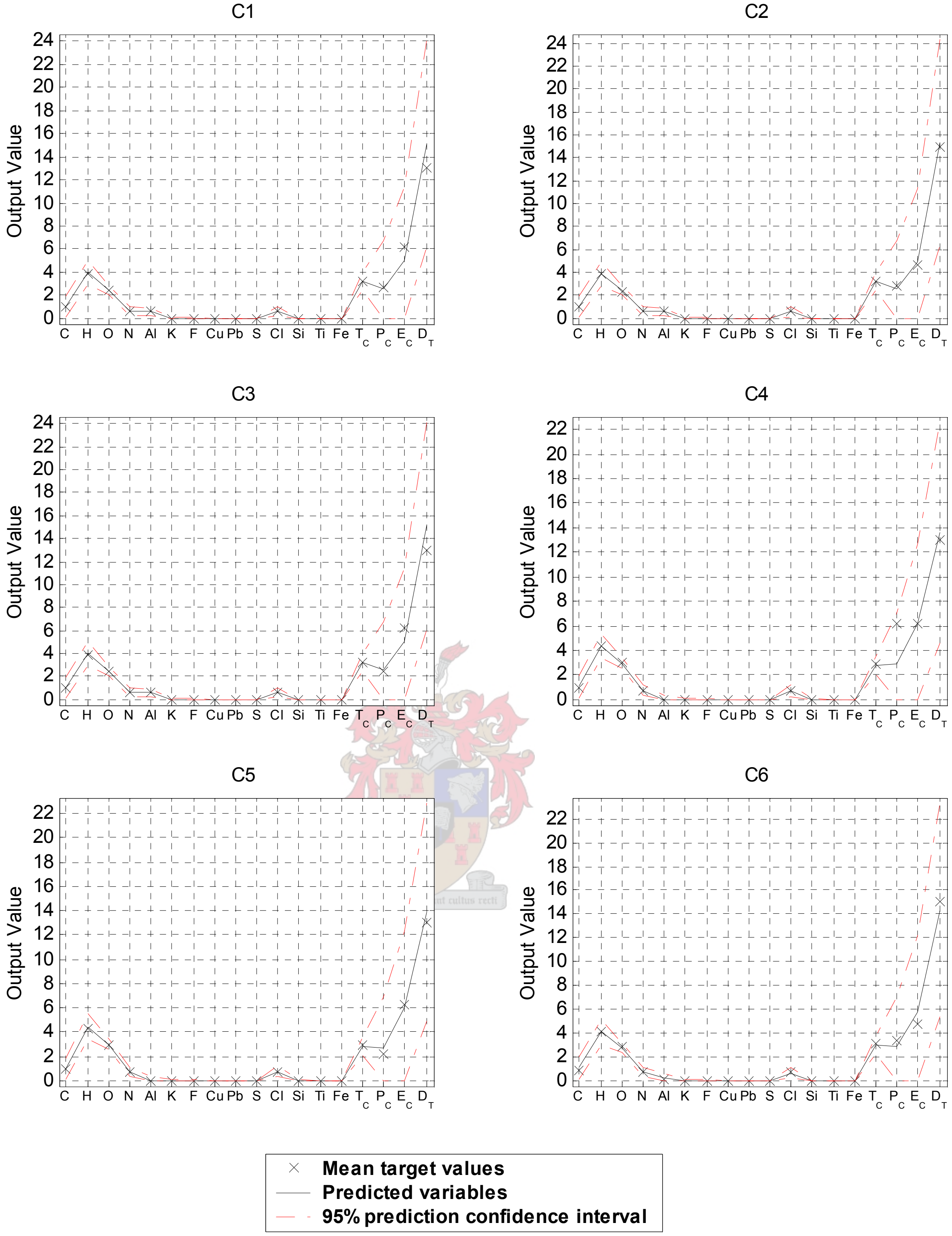

Fig. 5-20 The rocket motor parameter predictions for rocket motors C1 to C6 obtained for the overall optimum set feed-forward neural network models.

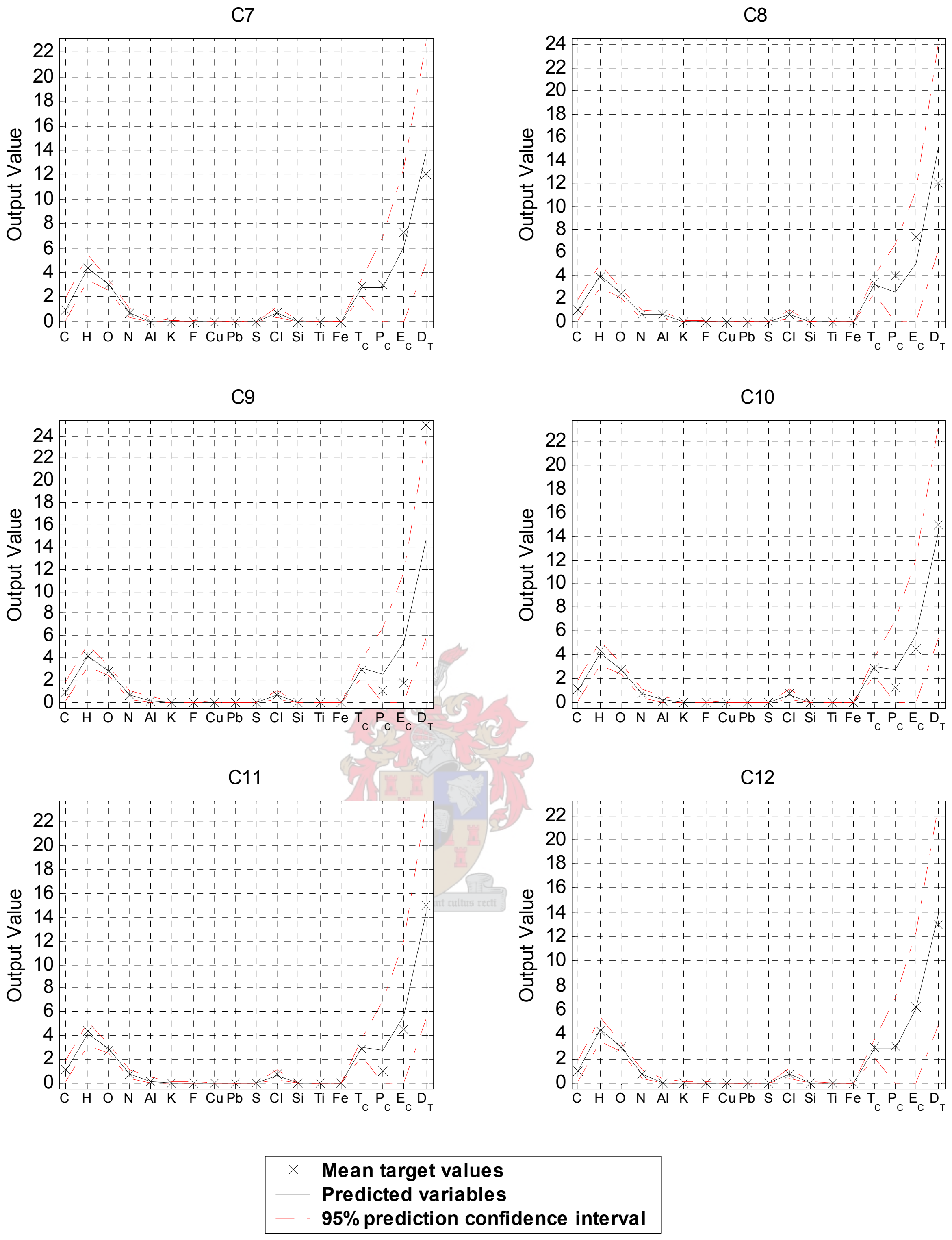

Fig. 5-21 The rocket motor parameter predictions for rocket motors C7 to C12 obtained for the overall optimum set feed-forward neural network models.

5.2.3 Neural Network PLS

The same approach used for the forward model was used to build the reverse NNPLS model. The first attempts at building a neural network PLS model were done using the error-based sigmoidal neural network (EBNNPLS) algorithm (see section 2.13.2). The inner neural network models were trained using 80 epochs for the Levenberg-Marquardt algorithm. This ensured sufficient over-training for the SISO (single input-single output) system. The best results are obtained by centring the input data and the standardising the output data.

The original cross-validation is not applied here following the undesired results illustrated in section 4.2.3. Only the modified cross-validation algorithm described in section 4.1.2 is therefore applied here.

Table 5-1 shows the modified cross-validation (PRESS) and overall model (SSEP) results for single runs using the EBNNPLS algorithm. The minimum PRESS-value at LD 1 is considerably high and is therefore not expected to perform better than the linear PLS model. For this reason the model is not evaluated any further. At this stage it therefore seems that the latent variables are better described using linear projections for both the forward and the reverse problems. The reason for this can also be attributed to the fact that the data is sparse and, as seen thus far, the models with lower complexities perform the best on unseen data.

Each modified cross-validation session was repeated ten times for each latent dimension using the NNPLS algorithm in order to obtain the box-and-whisker plot in Fig. 5-22. For each run a certain number of hidden nodes per latent dimension inner model are obtained and presented in Table B-9. The average PRESS-values in Table B-8 and Fig. 5-22 show that the optimum model for all eighteen output variables pooled lies at 3 latent dimensions. Run 8 is chosen as the best run from which further statistics are evaluated. The number of hidden nodes per latent dimension corresponding to run 8 is used to train the overall model. The training of the overall model is repeated 5 times and run 3 in Table B-10 is chosen as the best run for further evaluation.

Table 5-1 Modified cross-validation and overall model-building results for single runs using EBNNPLS.

LD	PRESS	SSEP	Hidden nodes per inner model
1	649.5	231.20	2
2	690.9	218.48	1
3	904.8	206.23	1
4	1035.3	202.39	1
5	1420.5	180.73	5
6	6358.8	180.18	2
7	7043.2	173.82	8
8	8132.3	65.01	10

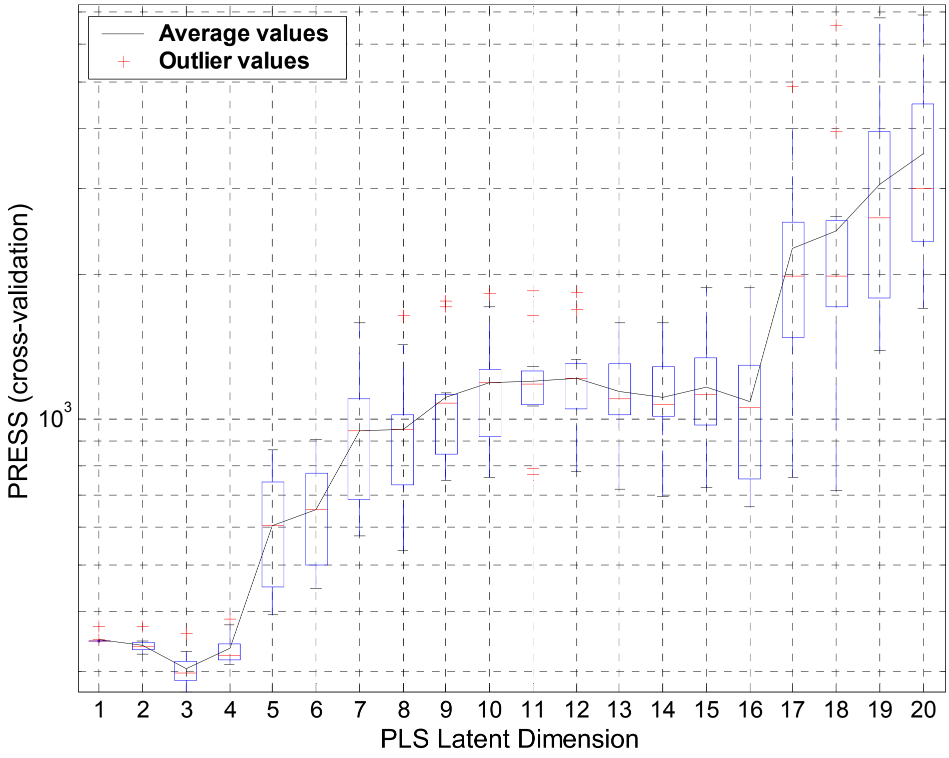

Fig. 5-22 A box-and-whisker plot of PRESS-values obtained for the reverse model using NNPLS from 10 cross-validation repetitions.

The average R^2_{cv} –value for 3 and 4 latent dimensions in Table B-11 is at a maximum corresponding to the minimum PRESS-values. The average R^2_{cv} –

value of 0.322 is slightly lower compared to the feed-forward neural network with 2 hidden nodes. The Y-block explained variance for the overall model is however significantly higher at 59.22%. The model is therefore capable of making more accurate predictions across all output variables or that certain key output variables are considerably better predicted. The average pdf-value is lower at 6.19 compared to the feed-forward neural network model with 2 hidden nodes. The PRESS- and SSEP- values at LD 3 are also significantly lower.

From the lower average pdf-value at LD 3 it appears the NNPLS model is less complex overall compared to the neural network model with 2 hidden nodes. The NNPLS model uses more parameters (462) per output variable compared to the neural network (297). However, the cumulative number of inner model parameters of 21 is relatively low. This means that 441 parameters are derived from the latent projections using NNPLS. From these criteria it is evident that it is difficult to compare model complexities by the number of total parameters. The pdf-values are therefore more useful for comparing model complexities.

The optimum set (OPT) of results for each individual output variable is shown in Table B-12. The overall results show slight improvements. The average number of latent dimensions can be rounded to 4. The number of latent dimensions for the various output variables varies greatly between 1 and 9.

The individual R^2_{cv} –values for the output variables are plotted in Fig. 5-23. The results are similar to those obtained for the feed-forward neural network. The predictabilities for F, S and Fe show the same low values when comparing the optimum (OPT) sets. The predictabilities for H, K, Cu, Si, Ti and P_C are lower. The predictabilities for Pb and Cl are higher and therefore the ability to compensate for both C-class and DB-class rocket motors can be expected. The predictabilities for D_T and C are also higher compared to the neural network optimum set. There are 7 output variables in the optimum set with R^2_{cv} –values above 0.5. The NNPLS model generally seems to better predict the output variables with values above zero for all 18 rocket motor types and poorer predictabilities can be expected for the variables with many zero entries.

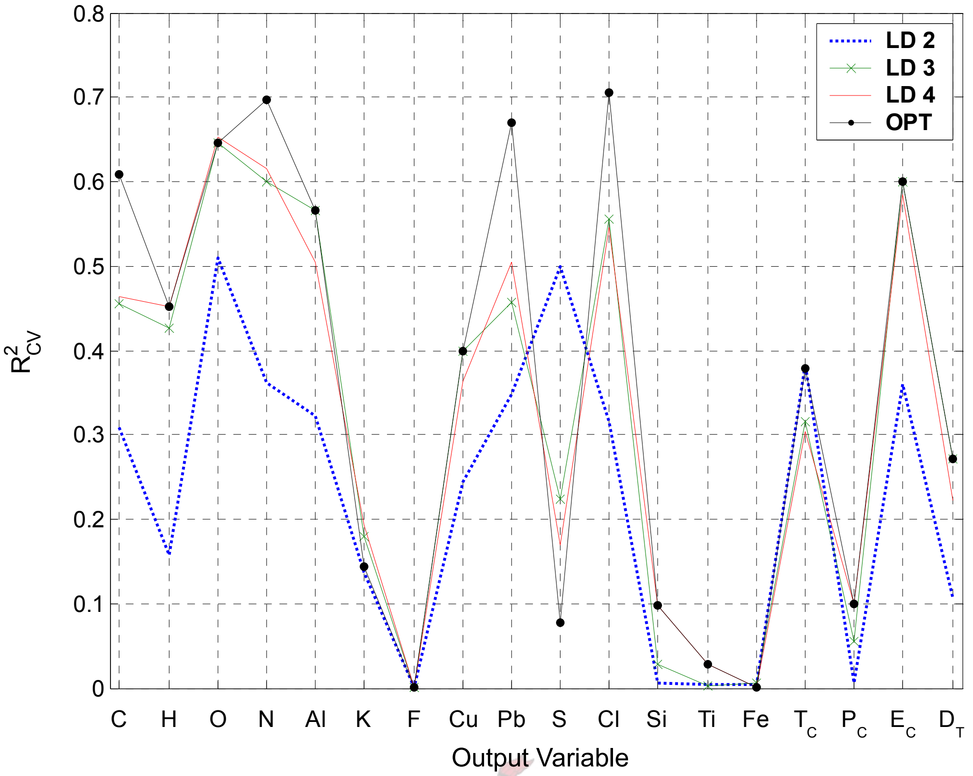

Fig. 5-23 The squared correlations between the unseen predicted rocket motor design parameters from cross-validation and their targets for NNPLS.

The optimum set (OPT) R^2 –values for the overall model in Fig. 5-24 appear better compared to the feed-forward neural network model. The predictabilities for S, E_C and D_T are all significantly higher, which explains the overall increase in Y-block explained variance (η^2).

There is great variability in the pdf-values (all below the value of 15) among the output variables. In Fig. 5-25 it can be seen how pdf can vary between 2 and 13 for LD 2. The high degree of variability can be attributed to the non-linearity of the model capturing different functional relationships for the various output variables. In Table B-11 it can be seen that the maximum average pdf-value of 18 is not completely achieved and yet 100% of the X-block variance is achieved up to LD 7.

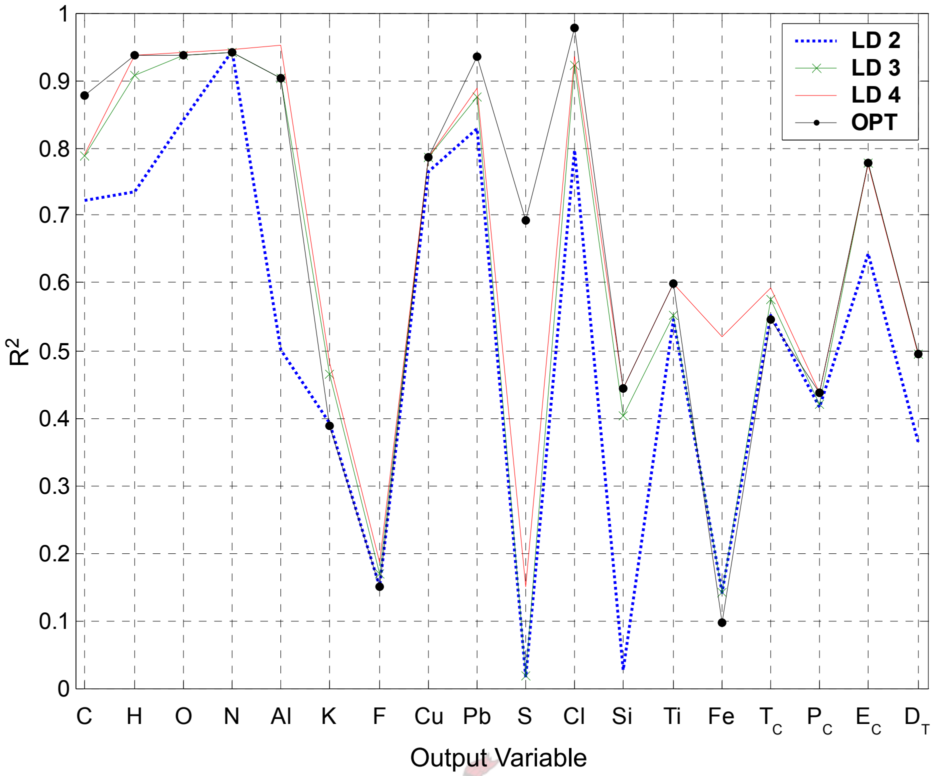

Fig. 5-24 The squared correlation coefficients between the predicted rocket motor design parameters for the overall model and their targets for NNPLS.

In Fig. 5-26 to Fig. 5-28 the unseen data point predictions are plotted for each rocket motor design together with the RMSECV band for each output variable. These predictions are plotted using the optimum set (OPT). The predictions do not appear as biased towards the C-class rocket motors as is the case for linear PLS. Except for DB1 and DB2 the predictions for Cl in the DB-class are virtually zero. Overall, the predictions are comparable to those of the feed-forward neural network. The predictions for P_C, E_C and D_T are especially poor for rocket motors DB3, DB4, C9, C10 and C11.

The predictions as well as the 95% confidence intervals for the optimum set overall models are shown in Fig. 5-29 to Fig. 5-31. The Cl predictions for all DB-class rocket motors are virtually zero as is the case for the feed-forward neural network. The predictions for E_C and D_T are slightly improved compared to the neural network models. The worst predictions for P_C, E_C and D_T are again found for rocket motors DB3, DB4, C9, C10 and C11.

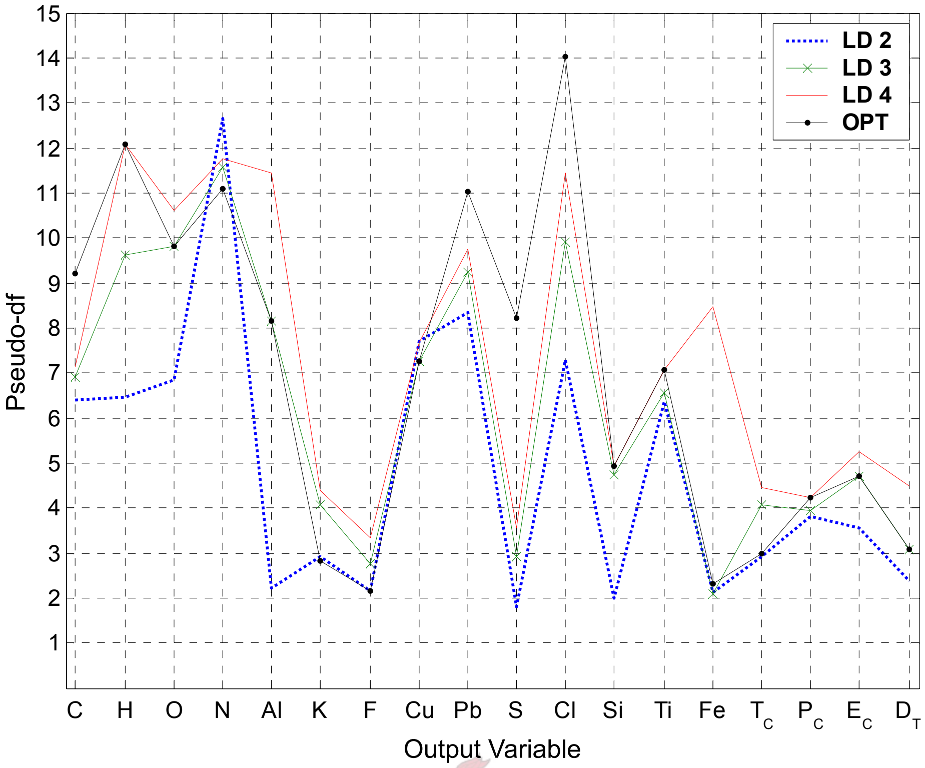

Fig. 5-25 The pseudo degrees of freedom of the NNPLS models calculated for each output variable.

The results from the discussion around the R^2 –values and the prediction plots above show that the feed-forward neural network and NNPLS models produce comparable predictions on the unseen data from cross-validation. However, the overall NNPLS models appear to perform slightly better.

The output scores (latent variables) for the first 6 latent dimensions are plotted against the input scores in Fig. B-4. The curved prediction lines are more sensible generalisations of the data, which explains the improvement in predictions by non-linear models. The curve for LD 2 has an interesting ‘dip’, which is characteristic of the trend in the data. This illustrates the strength of using a neural network as inner model in PLS. If it were not for the dip in the curve for LD 2, the prediction curves for LD 2 to LD 4 all have near sigmoidal-shapes.

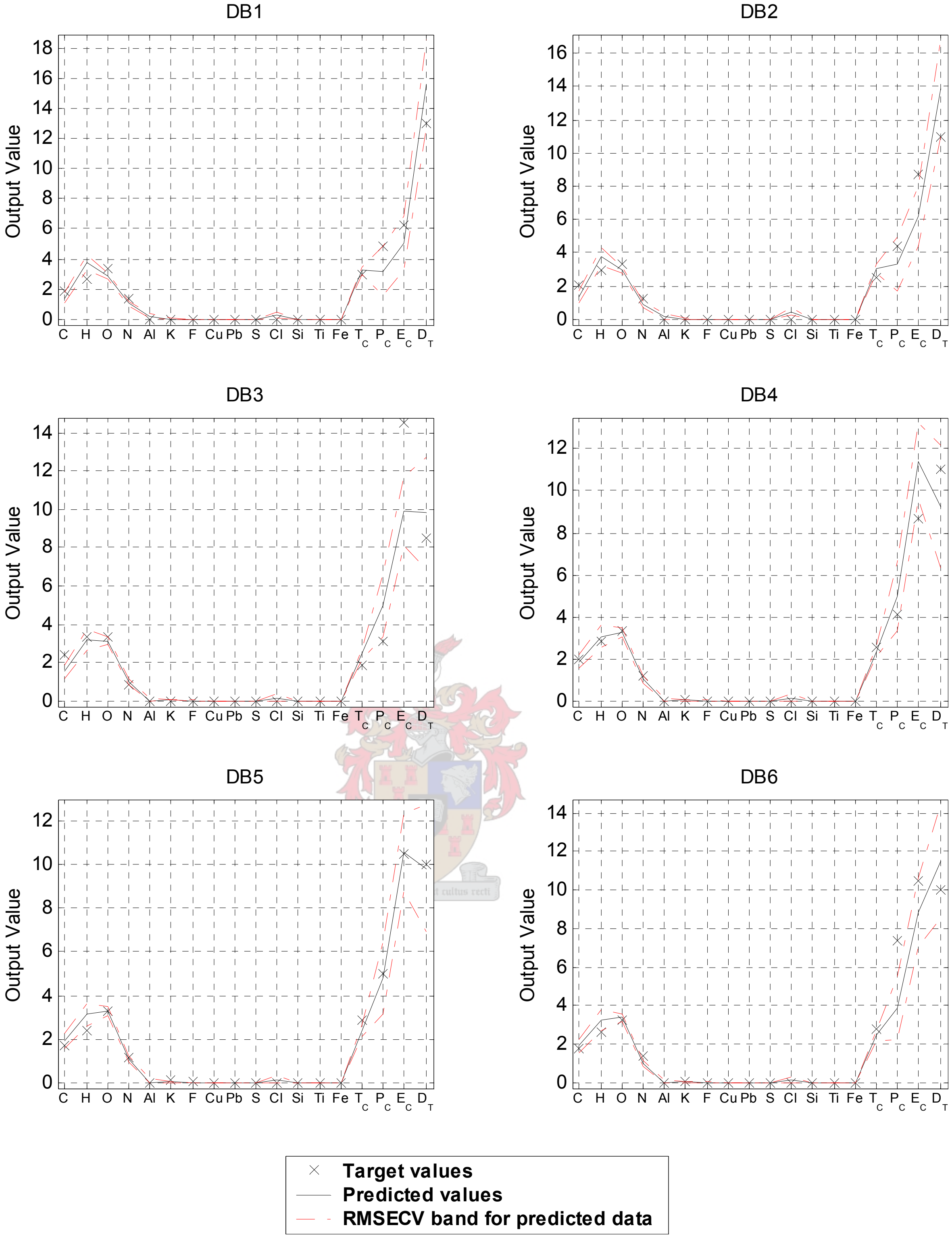

Fig. 5-26 The rocket motor parameter predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of the optimum set NNPLS models.

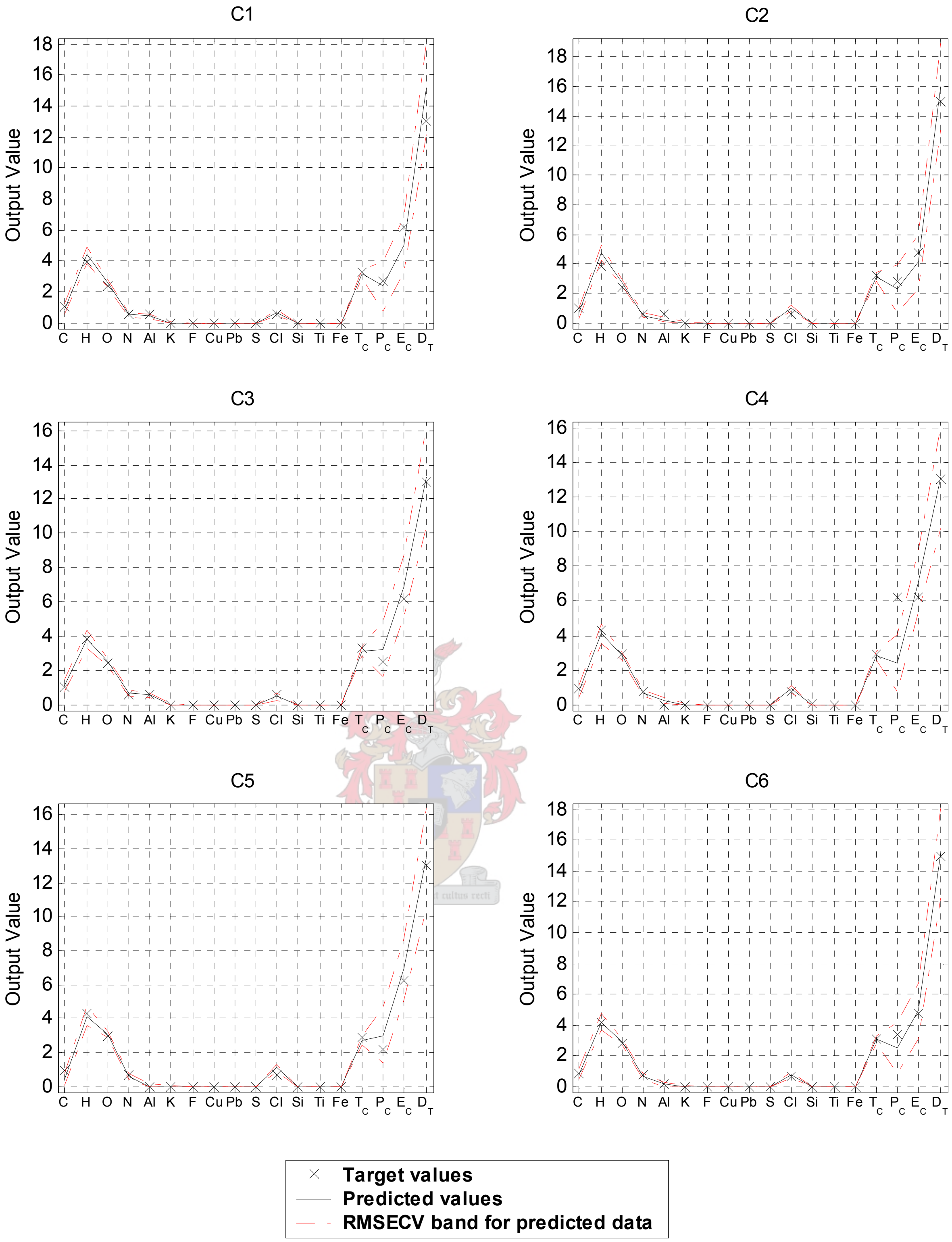

Fig. 5-27 The rocket motor parameter predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of the optimum set NNPLS models.

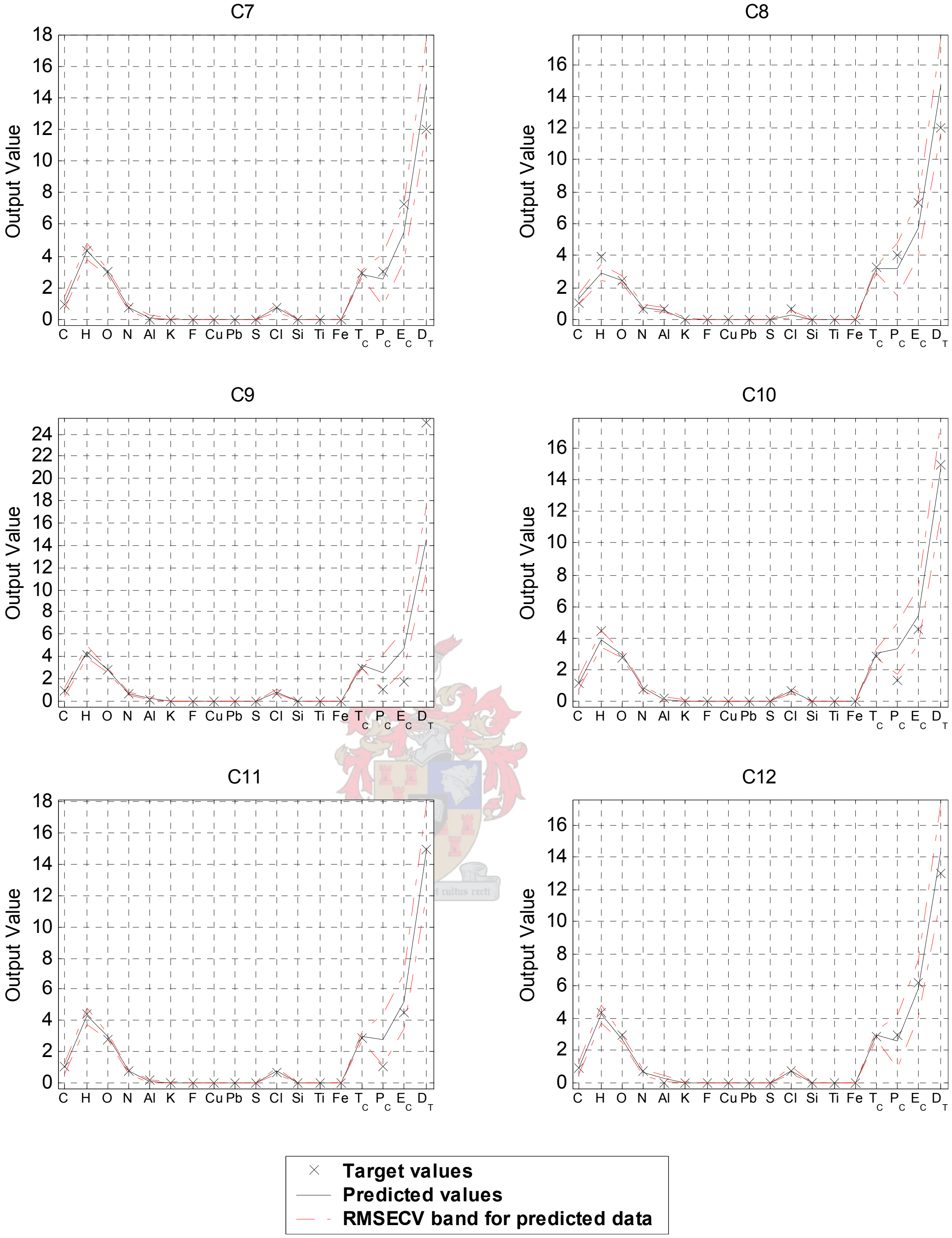

Fig. 5-28 The rocket motor parameter predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of the optimum set NNPLS models.

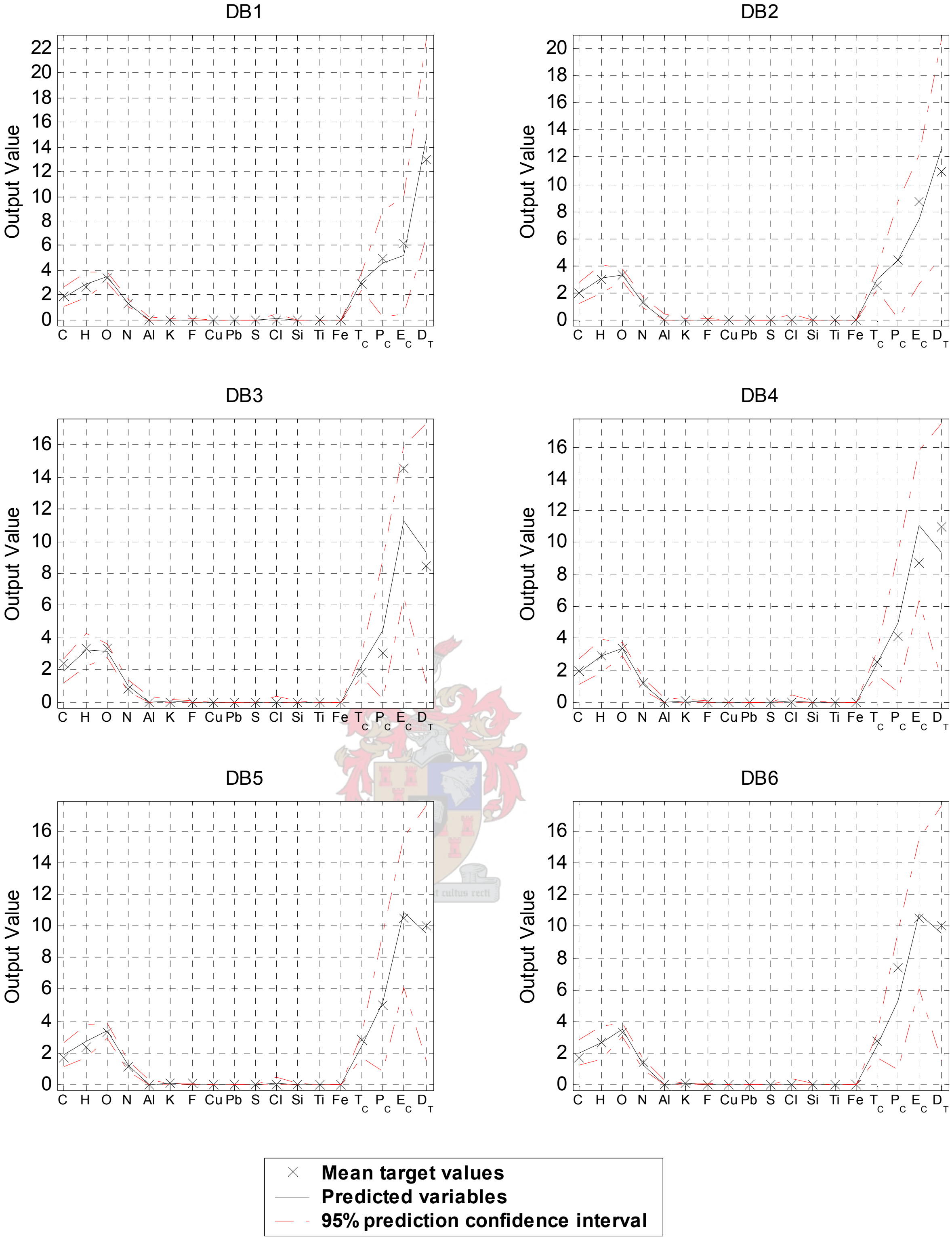

Fig. 5-29 The rocket motor parameter predictions for rocket motors DB1 to DB6 obtained for the overall optimum set NNPLS models.

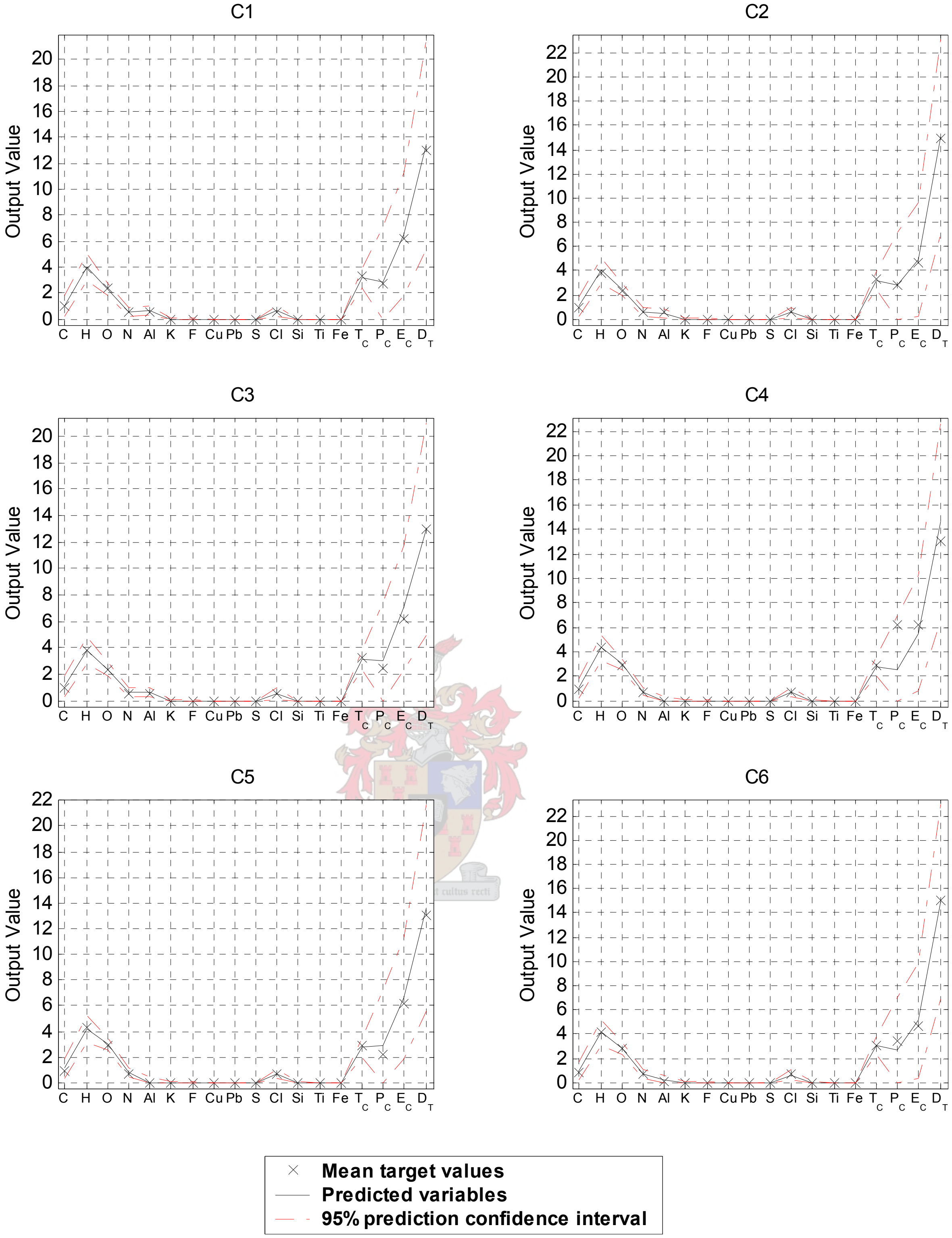

Fig. 5-30 The rocket motor parameter predictions for rocket motors C1 to C6 obtained for the overall optimum set NNPLS models.

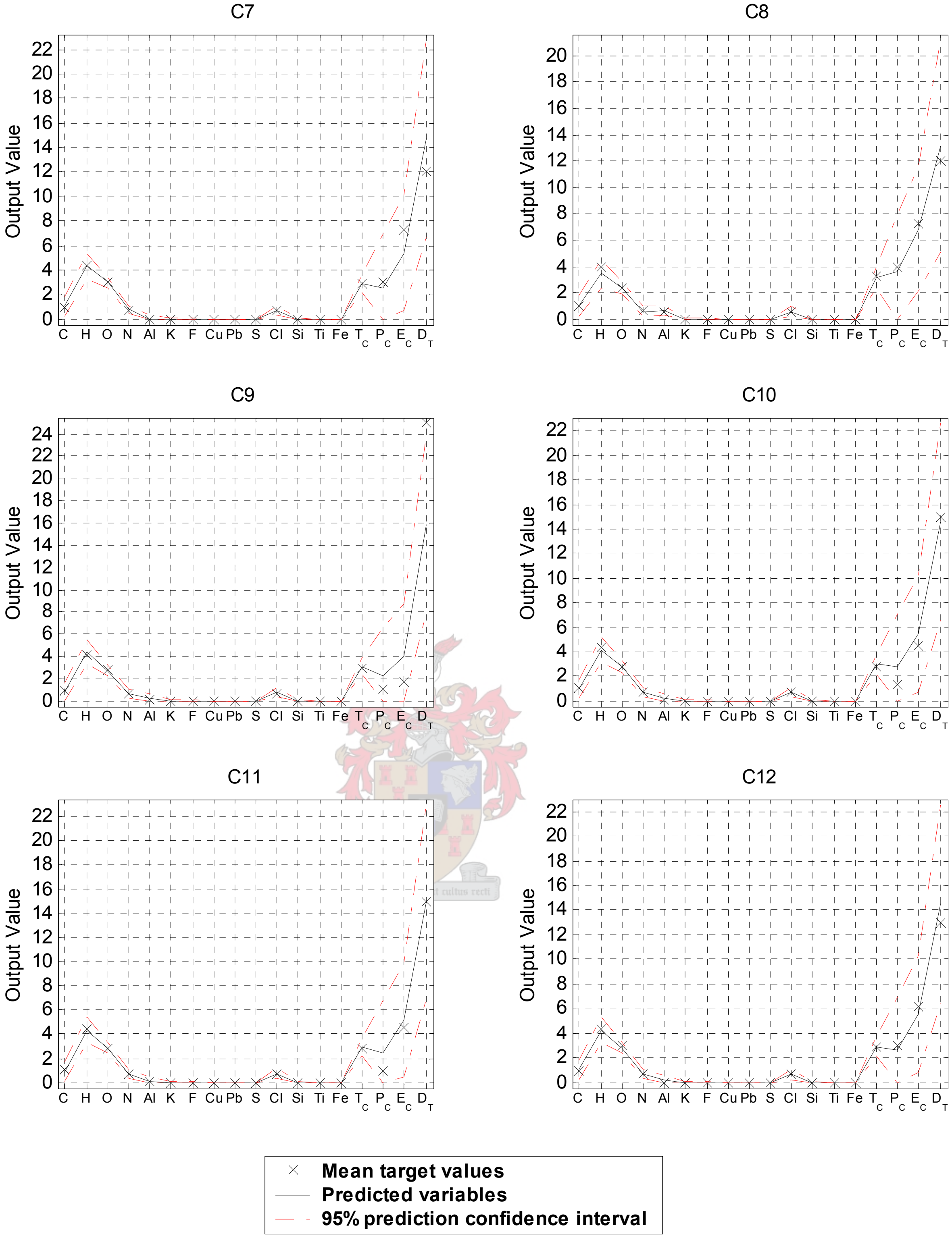

Fig. 5-31 The rocket motor parameter predictions for rocket motors C7 to C12 obtained for the overall optimum set NNPLS models.

5.2.4 Radial-basis Function PLS

The use of RBFPLS as an alternative to NNPLS is discussed in section 2.13.2 and 4.2.4. A motivation for applying this algorithm is that the RBF training algorithms discussed in section 2.12 allow for repeatable training results. Due to the undesired results obtained in sections 4.2.3 and 5.2.3 the EBRBFPLS algorithm is not considered here. The best results are obtained by centring the input data and the standardising the output data.

In Table B-13 the modified cross-validation and overall model results are shown for the RBPLS models built using the K-means RBF training method (see sections 2.7 and sections 2.12.1). The same two initialisation methods as discussed in section 4.2.4 are used here. The initialisation of the K-means algorithm has a significant effect on the model performance.

The OLS method for training the RBF within the RBFPLS model requires that the spread parameter be calculated according to some heuristic (see section 2.12). The nearest neighbours heuristic of equation (2-126) is applied here, as it performs better compared to equation (2-119). The results are shown for 3 and 4 nearest neighbours in Table B-14.

The ASOLS training algorithm is used for training the inner RBFPLS models and the PRESS-values from cross-validation are shown for three repeat runs in Fig. 5-32. The repeat runs represent different initialisations using 2, 3 and 4 nearest neighbours for runs 1,2 and 3 respectively (see steps 1 and 2 of the ASOLS algorithm in section 2.12.2). The ASOLS algorithm uses 2 nearest neighbours in steps 7 and 15 of the algorithm.

The PRESS-values in Fig. 5-32 show similar irregular trends as obtained for the forward problem (no of characteristic ‘dipping’ shape). This appears to be due to the nature of the training algorithm, as it is also observed for RBFPLS with OLS inner model training. The third run produces the lowest PRESS-values of all the RBFPLS models. An overall minimum occurs at LD 5 making it easier to determine the best model complexity for all 18 output variables pooled. The

SSEP-values for each run are shown in Table B-17. In Table B-16 it can be seen that LD 2 uses 10 hidden nodes. This is a significantly large number of hidden nodes and may explain the improved performance of the model relative to the other runs.

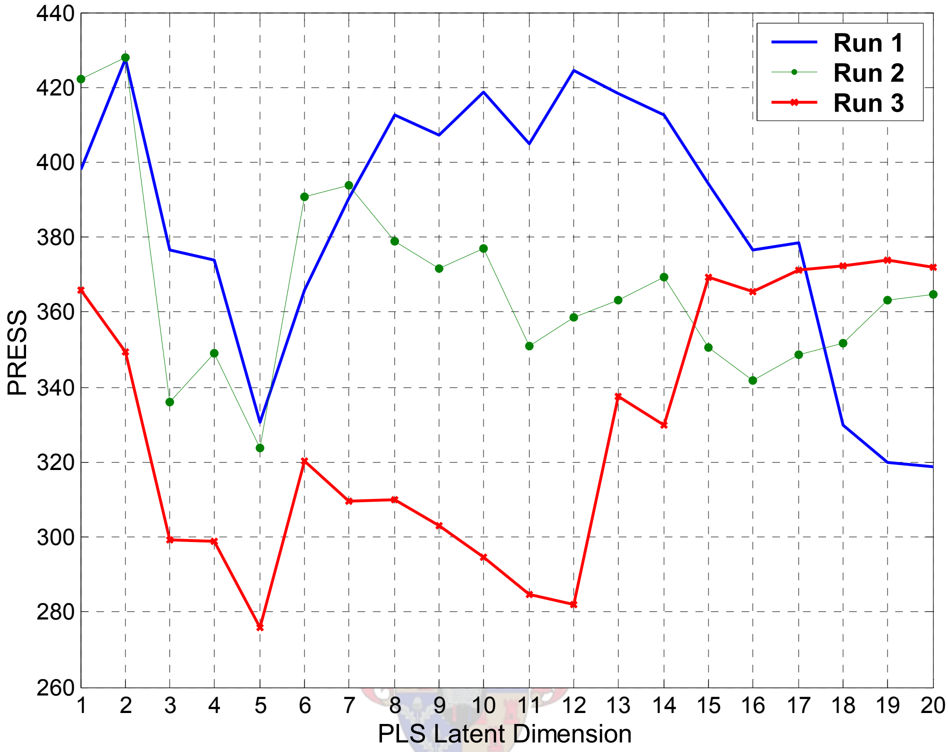

Fig. 5-32 A plot of PRESS-values using the ASOLS training algorithm for each inner model within RBFPLS (see Table B-15).

The PRESS- and, SSEP-values of 276 and 178 respectively for LD 5 in Table B-18 compare well with the values obtained by the NNPLS model for LD 3. The average pdf-value of 6.82 for the overall model shows that the model complexity is similar to that of the NNPLS model with an average pdf-value of 6.19. The Y-block explained variance of 57.55 % for the overall RBFPLS model with 5 latent dimensions is also similar to that of the NNPLS model (59.22% at LD 3). The average R^2_{cv} – and R^2 –values are slightly higher at 0.383 and 0.669. The RBFPLS model however requires 71 inner model parameters compared to the 21 required by the NNPLS model to obtain the similar overall model complexity.

The optimum set (OPT) of results for each individual output variable is shown in Table B-19. The overall results show significant improvements compared to the NNPLS and the feed-forward neural network. The average number of latent dimensions is 6. The average number of latent dimensions for NNPLS did not differ much from the best model for all output variables pooled. Similarly the feed-forward neural network had an average of 2.11 hidden nodes for the optimum set and 2 nodes for the best model for all outputs pooled. The number of latent dimensions for the various output variables varies even more greatly compared to NNPLS between 1 and 14. This is coupled with a slight increase in the average pdf-value of 7.27, which means that the slight increases in model complexities require larger amounts of model parameters. The average R^2_{cv} –value of 0.427 is higher compared to the NNPLS and feed-forward neural network models.

The individual R^2_{cv} –values for the output variables are plotted in Fig. 5-33. The predictabilities for F, Ti and Fe show the same low values as NNPLS when comparing the optimum (OPT) sets. Only the predictabilities for T_C and Pb are significantly lower compared to NNPLS. The R^2_{cv} –value for S shows the same high value as for linear PLS with its optimum at only 2 latent dimensions. There are 9 output variables in the optimum set with R^2_{cv} –values above 0.5. The high R^2_{cv} –values for H, N and Cl between 0.7 and 0.8 are encouraging results. The predictabilities for Cu and Cl are higher compared to NNPLS and the feed-forward neural networks and therefore the ability to distinguish between C-class and DB-class rocket motors can be expected. The predictability for K is higher compared to NNPLS but lower than the neural network optimum.

The optimum set R^2 –values for the overall model in Fig. 5-34 appear similar compared to NNPLS. The predictabilities for D_T are E_C are significantly higher with R^2 –values around 0.9. The overall RBFPLS model also better predicts Si. The predictions for F, Fe and S are as expected to remain inaccurate. The predictability for K remains fairly poor, which is not desired as K is an important component for suppressing the flame temperature of a fired rocket motor.

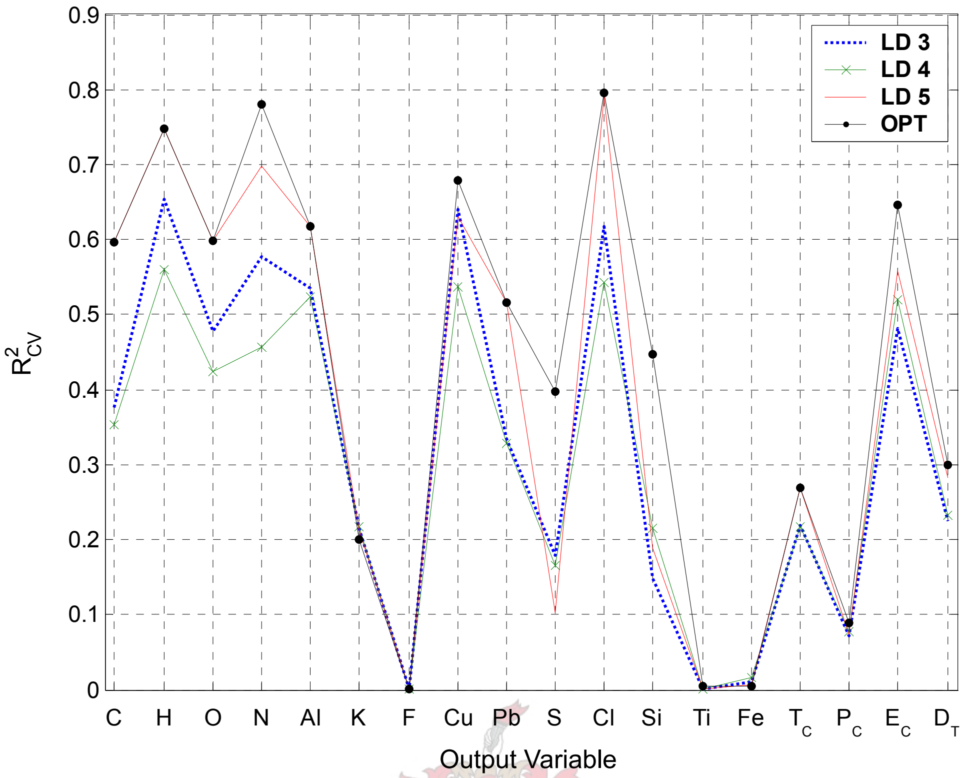

Fig. 5-33 The squared correlations between the unseen predicted rocket motor design parameters from cross-validation and their targets for RBFPLS.

In Fig. 5-35 it can be seen how pdf can vary between 1 and 15 for the RBFPLS overall models. The high degree of variability is more extreme for the RBFPLS models and can be attributed to the large variability in optimum latent dimensions (model complexities) for the different output variables. In Table B-18 it can be seen that a maximum average pdf-value of 15.51 is achieved and yet 100% of the X-block variance is achieved up to LD 7. Higher latent dimensions are not considered as any pdf above 15 is not considered viable from the discussion in section 3.1.3.

In Fig. 5-36 to Fig. 5-38 the unseen data point predictions are plotted for each rocket motor design together with the RMSECV band for each output variable. These predictions are plotted using the optimum set (OPT). The predictions do not appear as biased towards the C-class rocket motors as is the case for linear PLS. Only DB3 shows a significant deviation from zero for the Cl-prediction. The predictions for E_C and D_T appear slightly better compared to NNPLS. The predictions for P_C, E_C and D_T are especially poor for rocket motors C9, C10 and C11.

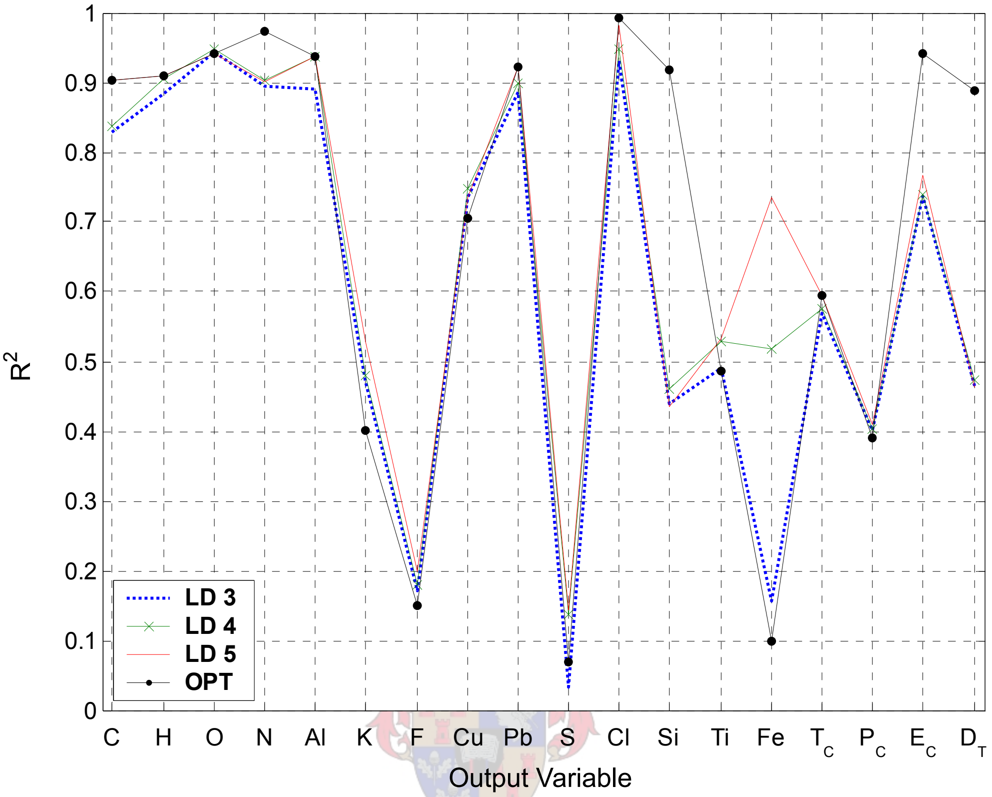

Fig. 5-34 The squared correlation coefficients between the predicted rocket motor design parameters for the overall model and their targets for RBFPLS.

The predictions as well as the 95% confidence intervals for the optimum set overall models are shown in Fig. 5-39 to Fig. 5-41. The Cl predictions for all DB-class rocket motors are virtually zero as is the case for the NNPLS and feed-forward neural network models. The predictions for E_C and D_T are slightly improved compared to the neural network models. The worst predictions for P_C, E_C and D_T are again found for rocket motors C9, C10 and C11. These predictions however appear slightly improved compared to the NNPLS and feed-forward neural network optimum predictions. The 95% confidence intervals remain high for P_C, E_C and D_T and therefore the precision of predictions in this region is not desirable. The predictions as well as the confidence intervals for C, H, O, N and Al remain satisfactory and comparable to the NNPLS and feed-forward neural network predictions.

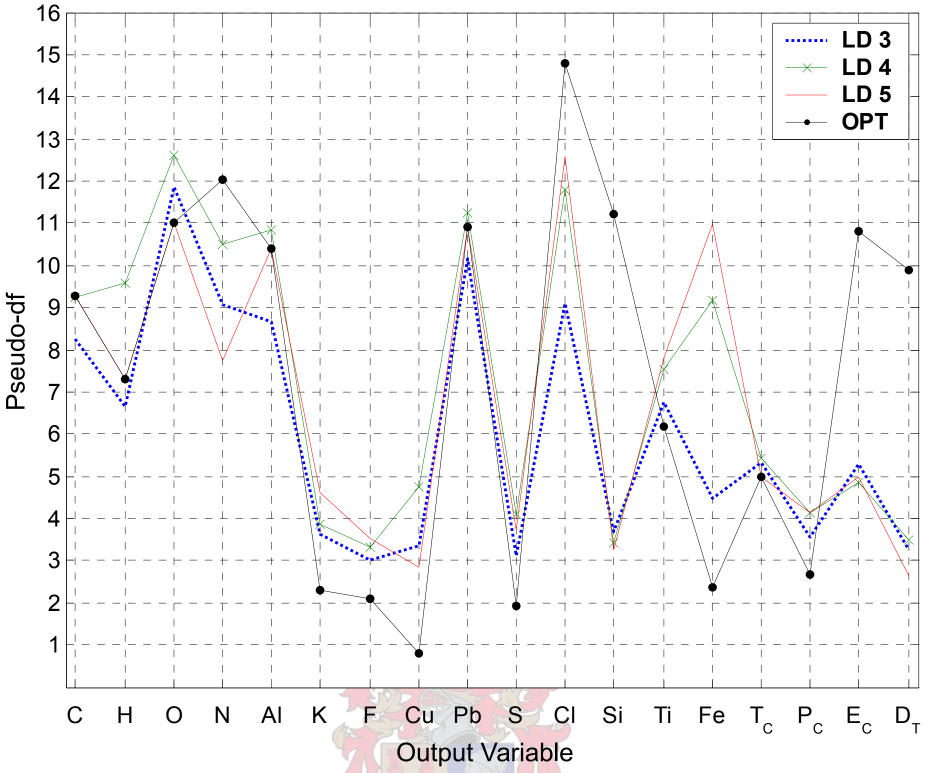

Fig. 5-35 The pseudo degrees of freedom of the NNPLS models calculated for each output variable.

The output scores (latent variables) for the first 6 latent dimensions are plotted against the input scores in Fig. B-5. The curved prediction lines are again more sensible generalisations of the data. The shapes of the prediction curves for the first 4 latent dimensions are similar to those obtained for the NNPLS overall model. The curve for LD 1 has a small spike, which is the major deviation form the NNPLS curve for LD 1. The near sigmoidal-shaped curves for LD 3 and LD 4 are especially similar to their NNPLS counterparts.

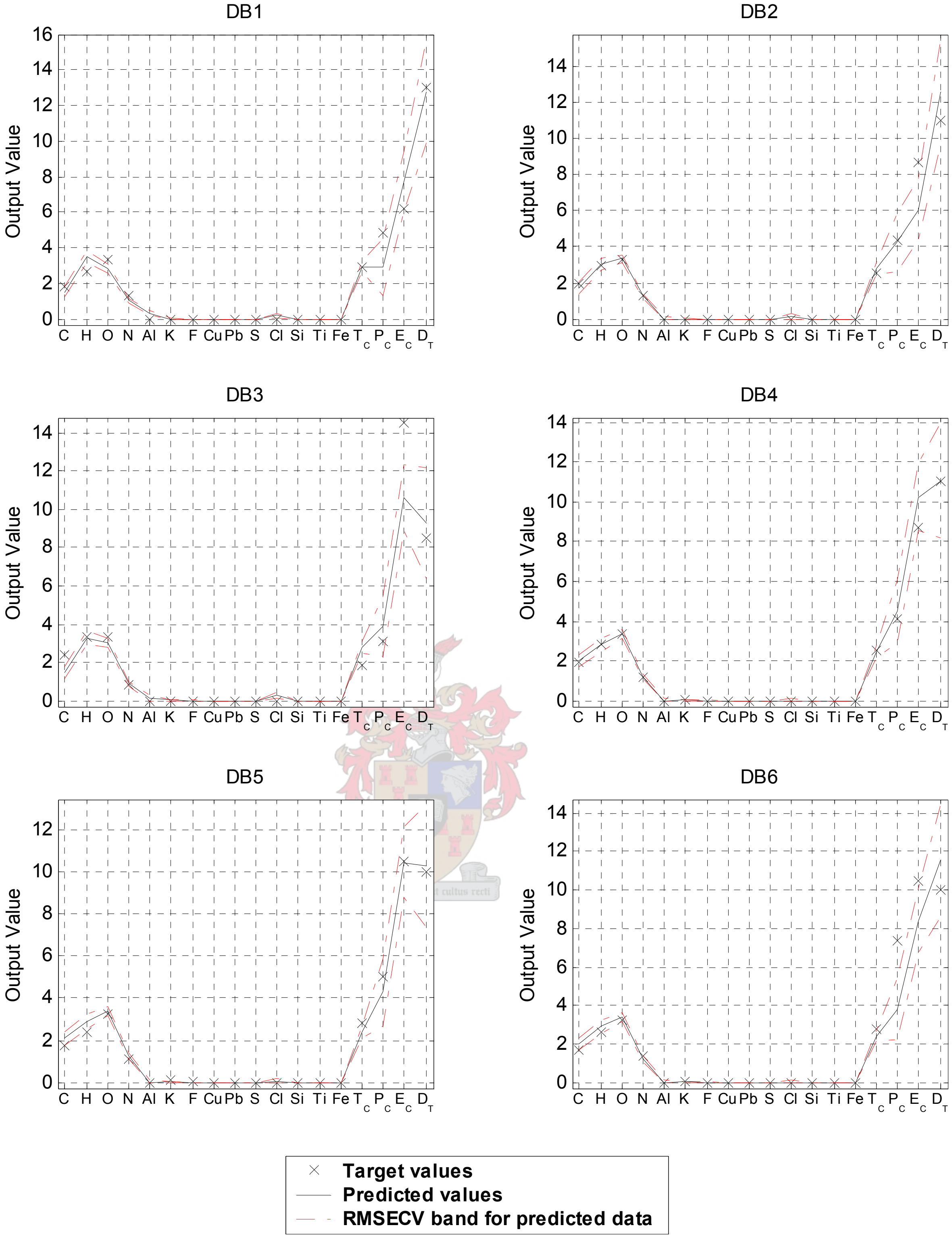

Fig. 5-36 The rocket motor parameter predictions for unseen rocket motors DB1 to DB6 obtained during leave-one-out cross-validation of the optimum set RBFPLS models.

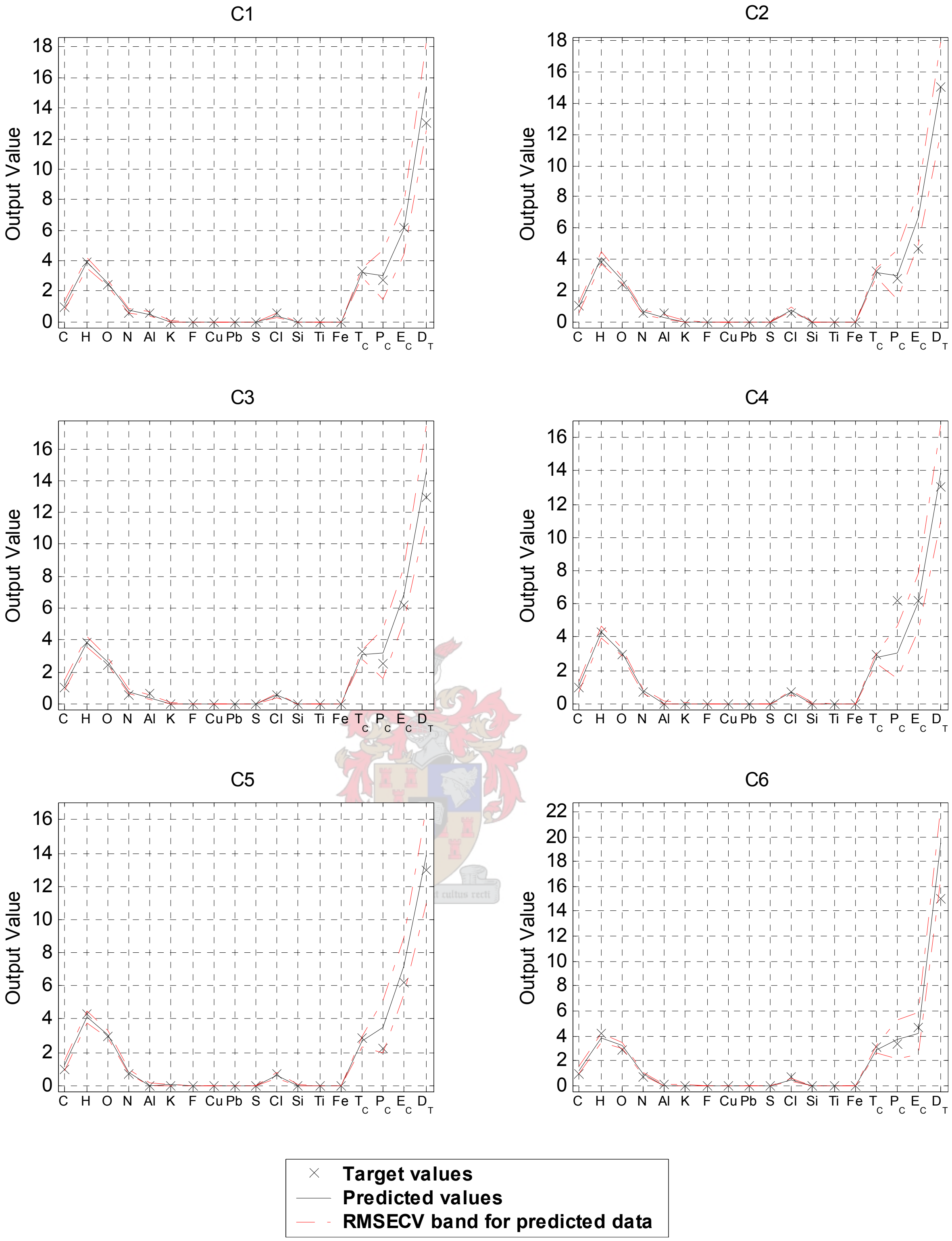

Fig. 5-37 The rocket motor parameter predictions for unseen rocket motors C1 to C6 obtained during leave-one-out cross-validation of the optimum set RBFPLS models.

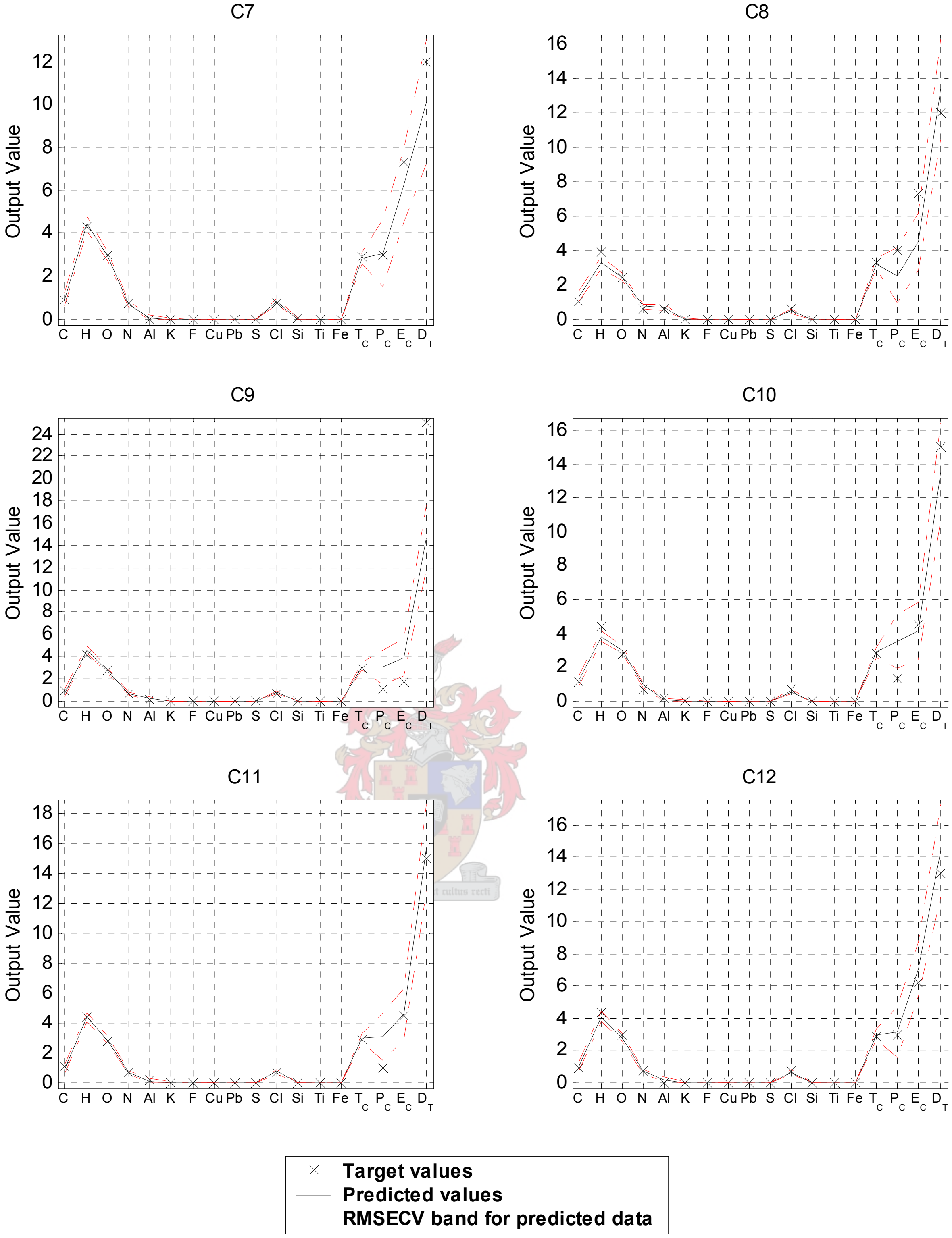

Fig. 5-38 The rocket motor parameter predictions for unseen rocket motors C7 to C12 obtained during leave-one-out cross-validation of the optimum set RBFPLS models.

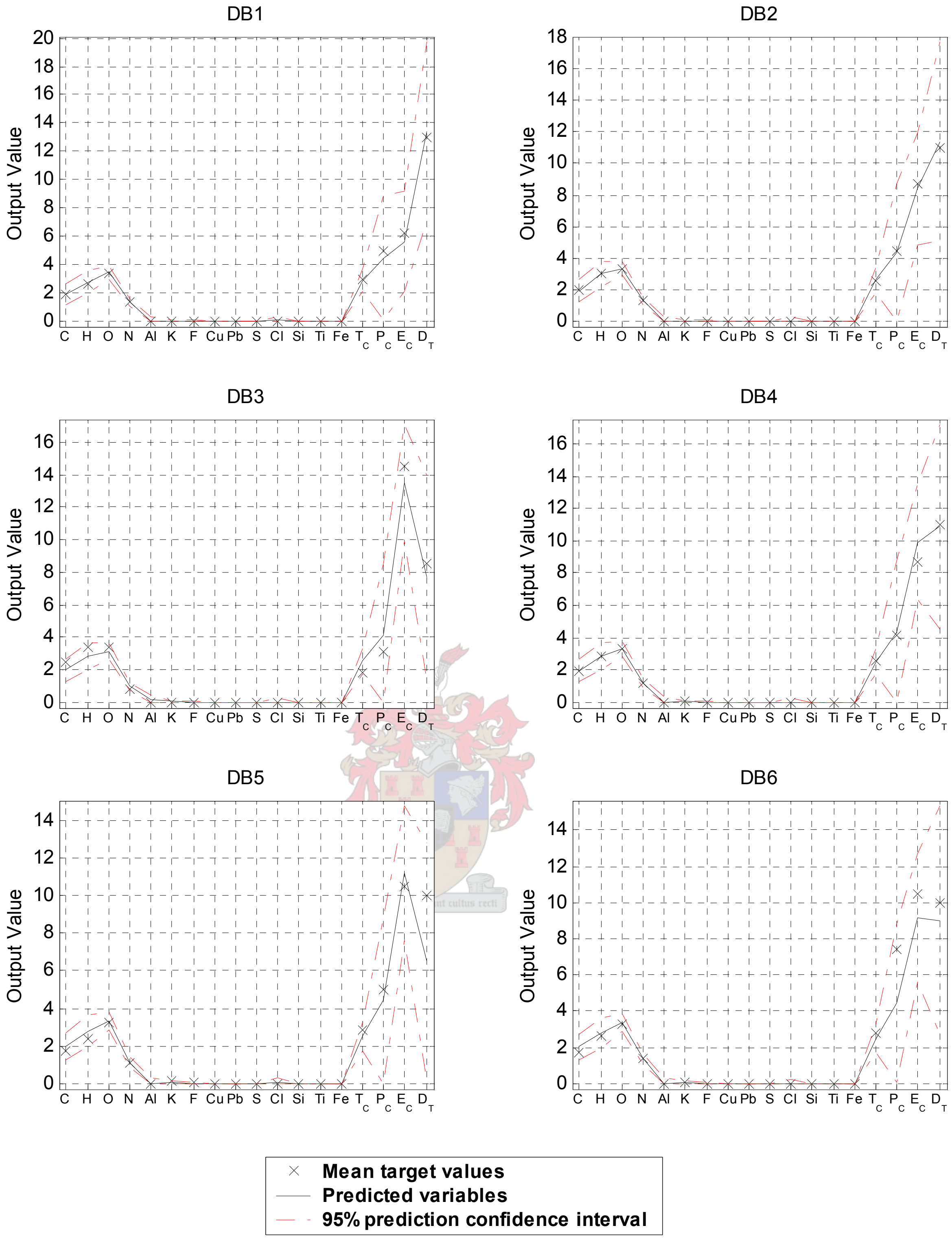

Fig. 5-39 The rocket motor parameter predictions for rocket motors DB1 to DB6 obtained for the overall optimum set RBFPLS models.

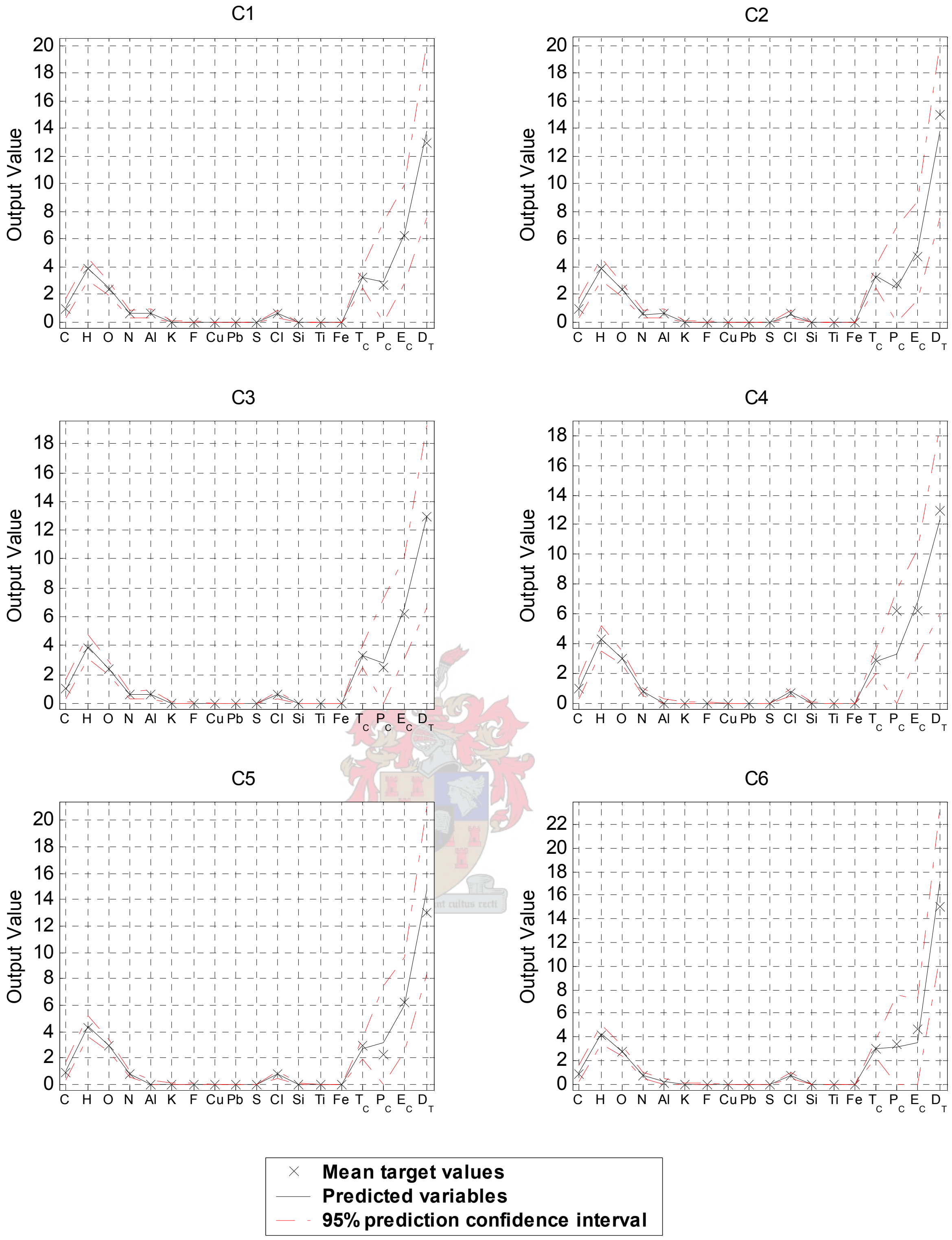

Fig. 5-40 The rocket motor parameter predictions for rocket motors C1 to C6 obtained for the overall optimum set RBFPLS models.

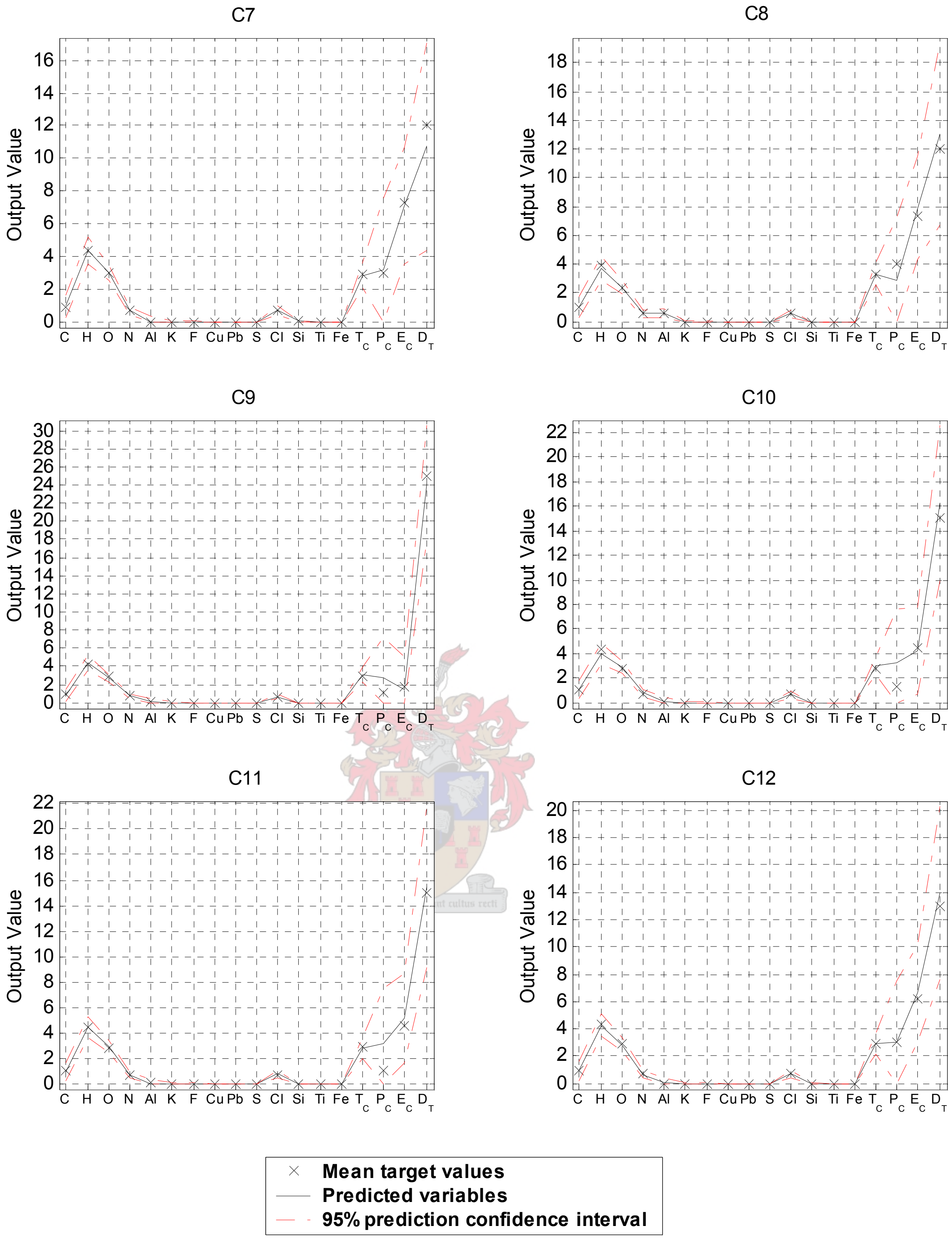

Fig. 5-41 The rocket motor parameter predictions for rocket motors C7 to C12 obtained for the overall optimum set RBFPLS models.

5.3 Discussion of the Candidate Models

A consolidation of the results in the more detailed discussion of section 5.2 is also included here for the reverse problem. The comparison summary for the results of all output variables pooled is shown in Table 5-2. The summary for the optimum sets as discussed in section 5.2 is presented in Table 5-3.

The summary for the pooled results shows that there is very little difference in the results between the NNPLS and RBFPLS models. If only the average R^2 – values are considered it can even be concluded that the feed-forward neural network model compares well to the NNPLS ad RBFPLS models.

Table 5-2 A summary of performance scores of each optimum model for the output variables pooled together. The Y-block variances are calculated on the overall optimum models.

	linear PLS	Feed-forward NN	NNPLS	RBFPLS (ASOLS)
Complexity	2 LD	2 H	3 LD	5 LD
PRESS	475.6	329.1	273.8	276.1
SSEP	307.08	245.61	171.18	178.18
Y-block $\%\eta^2$	26.85	41.49	59.22	57.55
Ave R^2_{cv}	0.183	0.351	0.322	0.383
Ave R^2	0.351	0.615	0.615	0.669
Ave pdf	3.75	7.07	6.19	6.82
Parameters	296	297	462	806

There are significant differences between the models concerning the predictabilities of the individual output variables. The best that each model can perform on unseen data is therefore reflected in the optimum set. Here the RBFPLS model with the ASOLS training algorithm for the inner PLS mappings stands out slightly. The feed-forward neural network best predicts the unseen data of output variables, K, Ti and P_C. This suggests that the optimum results may therefore be attainable using the best modelling technique for each individual output variable. This is a tedious task, but the desired predictabilities may be best achieved in this way.

In section 5.2 it is observed that the predictions for unseen as well as training data appear fairly good for the output variables, C, H, O, N, Al and T_C. The relatively tight 95% confidence intervals indicate that the predictions in this region can be made with a fair degree of precision. It must be noted that the ranges for these output variables are very small in magnitude. This implies that even when a value appears to be closely predicted the margin may not be small enough for it to be an accurate prediction.

The predictions for E_C are observed to be fairly accurate, however the 95% confidence interval indicates a lack of expected precision. The predictions for P_C and D_T are not accurate and this is reflected in the broad confidence interval band.

Table 5-3 A summary of performance scores of the optimum sets (OPT).

	linear PLS	Feed-forward NN	NNPLS	RBFPLS (ASOLS)
Complexity	3.61 LD	2.11 H	3.78 LD	6 LD
PRESS	451.7	327.8	271.9	254.9
SSEP	279.0	245.7	168.3	73.9
Ave R^2_{cv}	0.244	0.379	0.358	0.427
Ave R^2	0.424	0.610	0.662	0.679
Ave pdf	5.09	6.57	6.95	7.27
Parameters	534.4	313.4	588.6	958.8

Due to the low magnitudes of the variables from K to Fe it is difficult to observe the confidence intervals in the prediction plots. The predictabilities of these variables are better determined by the R^2-values. From these values it is evident that Cu, Pb and Cl are fairly accurately predicted and that mostly due to a lack of data the other variables, F, S, Si, Ti and Fe are not worth evaluating. The predictabilities for K are generally not as desired but there is a small amount of explained variance due the fact that there is more data compared to F, S, Si, Ti and Fe.

It is interesting to note from Table 5-2 that the three neural network based candidate models produce similar average pseudo degrees of freedom. This tendency is also observed for the forward problem. The result confirms the notion that the optimum results are obtained at a certain MSECV/MSEP$_{rs}$-ratio and that the different models complexities can be compared by virtue of the pseudo degrees of freedom.

It is difficult to determine exactly where the optimum model for linear PLS lies (see discussion in section 5.2.1). From the optimum set results the conclusion is that the pooled optimum might even be at 3 or 4 latent dimensions. This corresponds to 4.91 and 7.45 average pseudo degrees of freedom, which is closer to the average pdf-values obtained for the neural network based models.

It is desirable to obtain PRESS-values comparable to the SSEP-values. An analysis is done in section 5.4 on the MSECV- and MSE-values of the individual output variables in order to test for equivalence.

5.4 Analysis of the Residual Variances

The results thus far do not indicate a candidate model that can be regarded as irrefutably the best model. The RBFPLS model with ASOLS inner model training does slightly outshine the other models as far as the overall pooled and optimum set results are concerned. For this reason the analyses in this section are limited to the results from this model.

The Euclidean distances between the predicted and target rocket motor design parameters for the overall RBFPLS model with 5 latent dimensions are plotted in Fig. 5-42. The plot shows that rocket motor C9 has the largest deviation between predicted and target values altogether. The reason for this is that the value for D_T is significantly larger compared to the other values.

The values of the Euclidean distances are spread out randomly. This means that the function will not suffer from inherent functional mismatch, as can be the case for linear models.

There are no replicate measurements available for a lack-of-fit F-test and from the discussion in section 4.4 the regression F-test is not feasible. However, the MSECV$_j$- and MSE$_j$-values can be tested for the null hypothesis of equivalence. Equivalence indicates that the overall model is adequate in making predictions of the quality analysed for the model on unseen data within the range used for training.

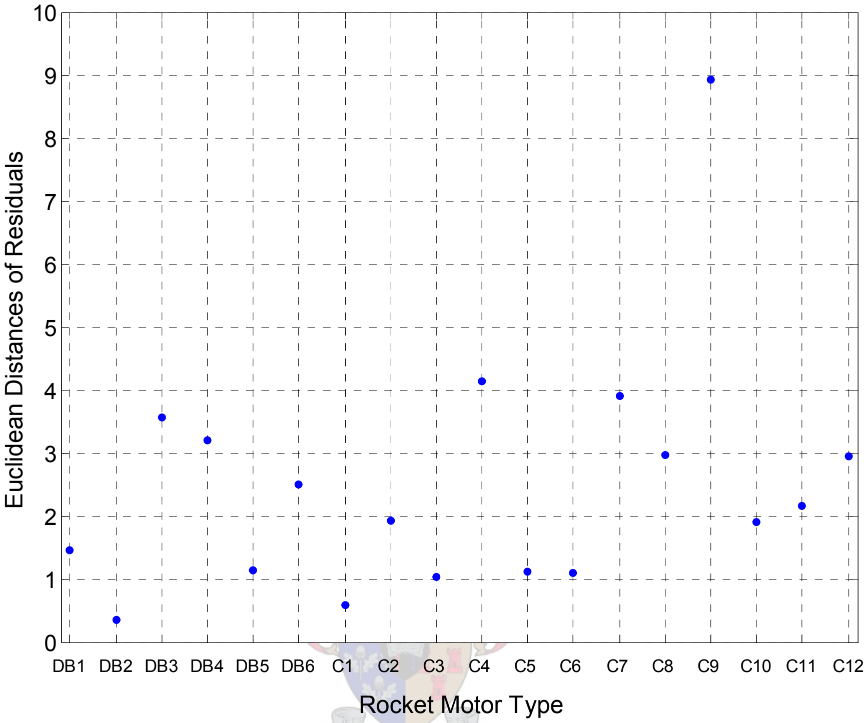

Fig. 5-42 The Euclidean distances between the target and predicted spectra from the overall RBFPLS model (LD 5) plotted against their data points.

In Fig. 5-43 the null hypotheses of equivalence for MSECV$_j$ and MSE$_j$ at an α-level of 1 % are tested for each output variable using the RBFPLS model with 5 latent dimensions. In Fig. 5-44 the same null hypotheses of equivalence are tested using the results from the optimum set models for RBFPLS. In both cases it is evident that the models can be trusted, as the F-statistics for all output variables are within the minimum and maximum ranges of the two-tailed 99% confidence bands. The minimum-maximum ranges for the optimum set are generally larger compared to those for the pooled results set. This result is expected due to the optimisation for each individual output variable.

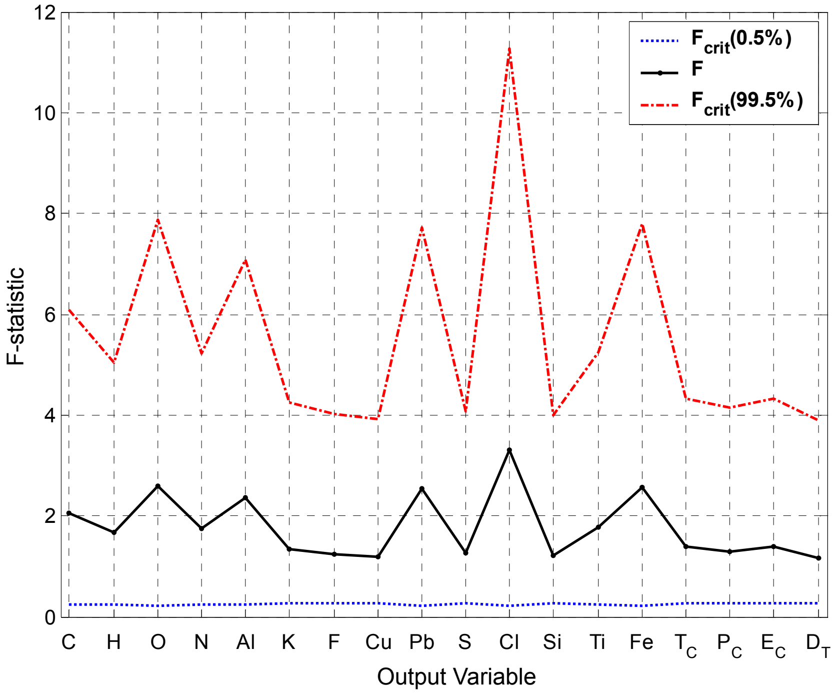

Fig. 5-43 The ratio, F = MSECV/MSE is compared to the critical F-statistics at 18 and 18-pdf degrees of freedom for all 18 output variables using the overall RBFPLS model with 5 latent dimensions.

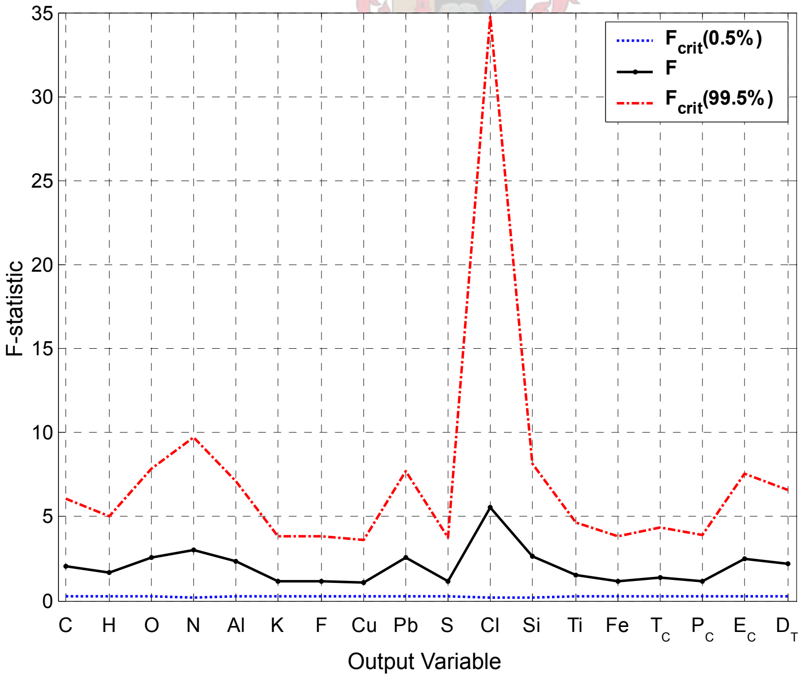

Fig. 5-44 The ratio, F = MSECV/MSE is compared to the critical F-statistics at 18 and 18-pdf degrees of freedom for all 18 output variables using the overall optimum set of RBFPLS models.

5.5 Sensitivity Analysis

A sensitivity analysis for the reverse modelling problem is not an obvious procedure. It is not feasible to manipulate each of the 146 input absorbance values. All the input variables are recorded in the same units. It is difficult to exactly define an area of significant influence. Even if it were possible it is difficult to determine by how much each variable within that spectrum must be manipulated in order to have a realistic influence on the output variables.

The sensitivity analysis in section 4.5 consists of two parts, the first part tested the model responses to individual perturbations in the input set and the second is a to test qualitatively the responses to global changes in the input set. The model predictions from the latter analysis can be used as inputs to the reverse model. The desired result is that the outputs qualitatively move in the direction of the original global changes made to rocket design parameters.

In section 4.5 the global changes are made to rocket motor C11, where the aluminium content is halved and doubled and the rocket motor binder is replaced with a different material. The newly calculated chamber temperatures and pressures are included as global changes. The set of global changes comprise C11-A to C11-D (see Table 4-8). The model outputs in Fig. 4-54 can now be used as inputs to the NNPLS overall model with 3 latent dimensions. Fig. 5-45 shows that the chamber pressure (P_C) and the conic expansion ratio (E_C) are not successfully predicted at all. These variables are unchanged in the original changes made to obtain C11-A to C11-D. The predicted values are shown in Table B-20 and should be compared to the target values in Table 4-8. The predictions for Cl, T_C and D_T compared to the target values remain stable and satisfactory. An interesting result is that the presence of Fe is predicted for C11-A to C11-C. The predictions for Al are quantitatively inaccurate although the qualitative trend from the original C11 composition is correct. The qualitative trends for C, H, O and N are somewhat erratic, although quantitatively the predictions appear stable.

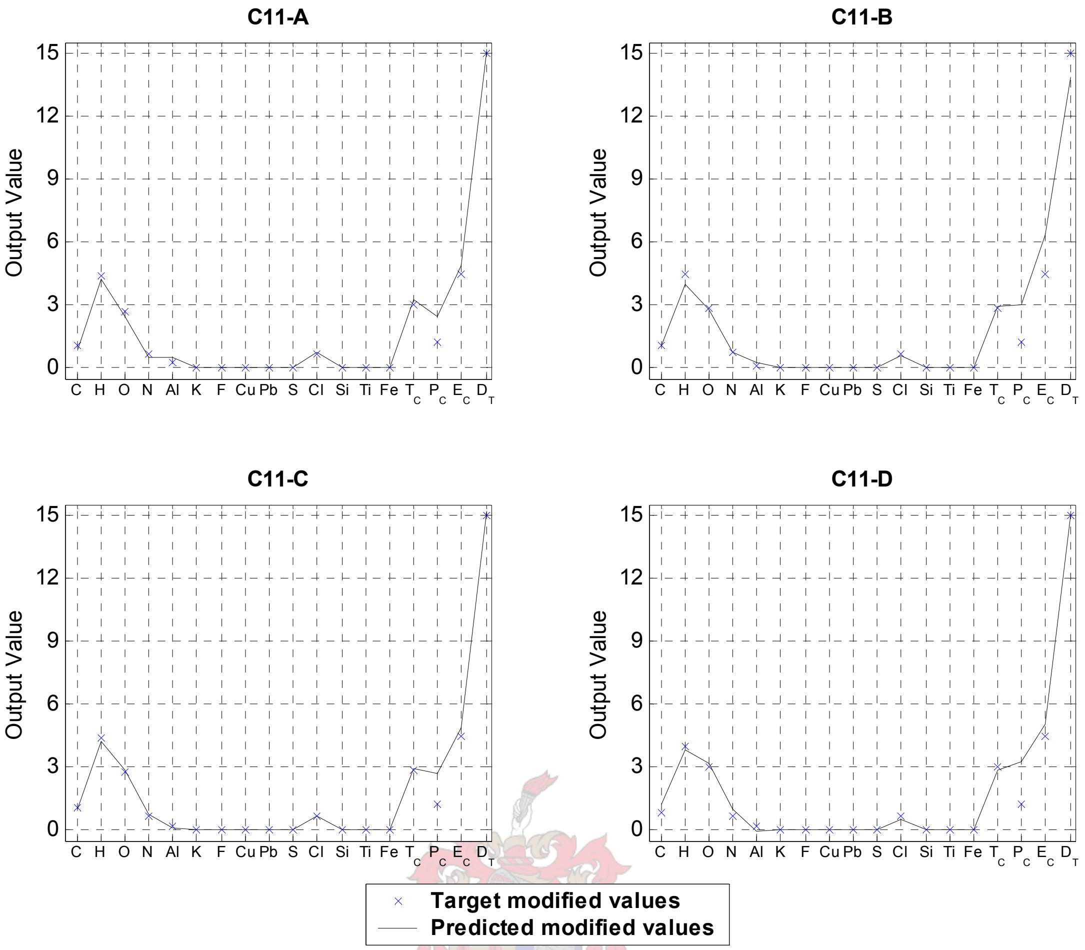

Fig. 5-45 The predictions made from making global changes to C11 using the reverse overall NNPLS model with 3 latent dimensions.

In Fig. 5-46 the principal input loadings of the overall NNPLS model with 3 latent dimensions are plotted to highlight the leverage provided by each input variable towards their latent variables (scores, t_k). For all three latent dimensions the CO_2 absorption region has the largest and most complex influence on the latent variables. The H_2O absorption region of the spectrum makes a smaller contribution, mainly for the first and third latent dimensions. Significant leverage from the region between the H_2O and CO_2 peaks is only observed for the second latent dimension. The results therefore show that significant leverages comes from all 3 regions as discussed, including the small area in the front end from wavelength numbers 1 to approximately 10.

The best results for NNPLS were obtained by only centring the input data. The fact that there is no scaling accentuates the magnitude of the larger absorbance values relative the lower values. The implication of this and the higher loadings seen in the CO_2 region is that the influence of the CO_2 absorbance region is the most important for carrying across information to make accurate predictions.

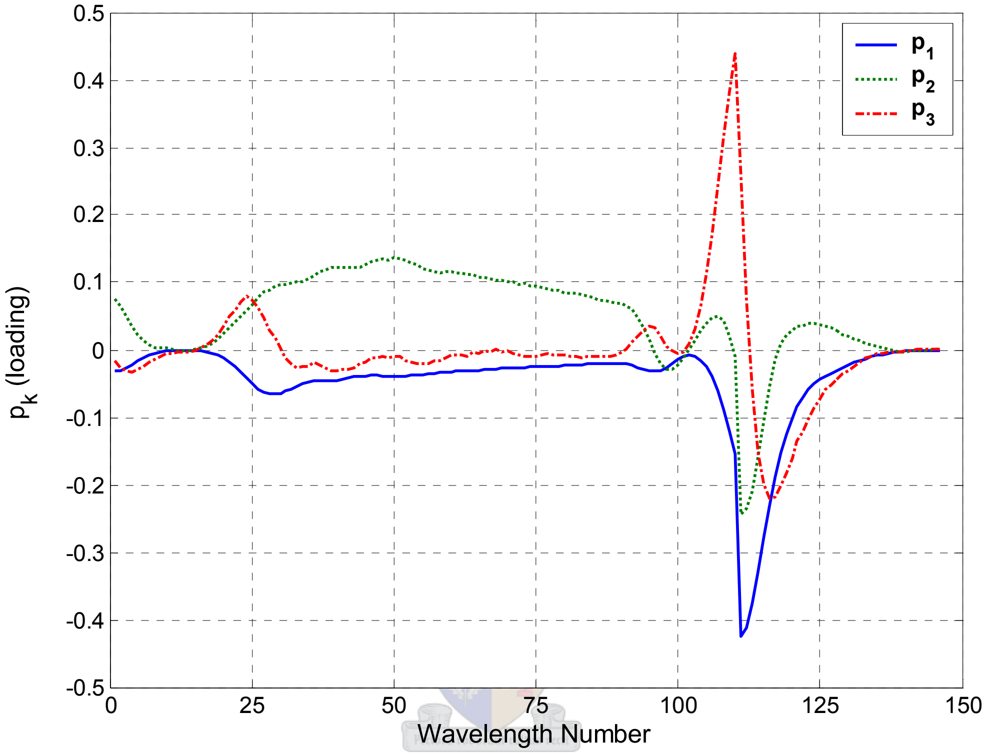

Fig. 5-46 The principal input loadings for the overall NNPLS model with 3 latent dimensions.

The graphs in Fig. 5-47 to Fig. 5-49 are included to show the range of predictions made from the original repeat measurements for each rocket motor type for the optimum set of overall RBFPLS models. The predictions appear well within the 95% confidence intervals when compared to Fig. 5-39 to Fig. 5-41. The precisions of the predictions are comparable to the RMSECV band of the unseen predictions. It must be noted that the RMSECV- and RMSE-values obtained for each model can be seen to have similar values. This is supported by the equivalence of the variances, MSE and MSECV as tested in section 5.4.

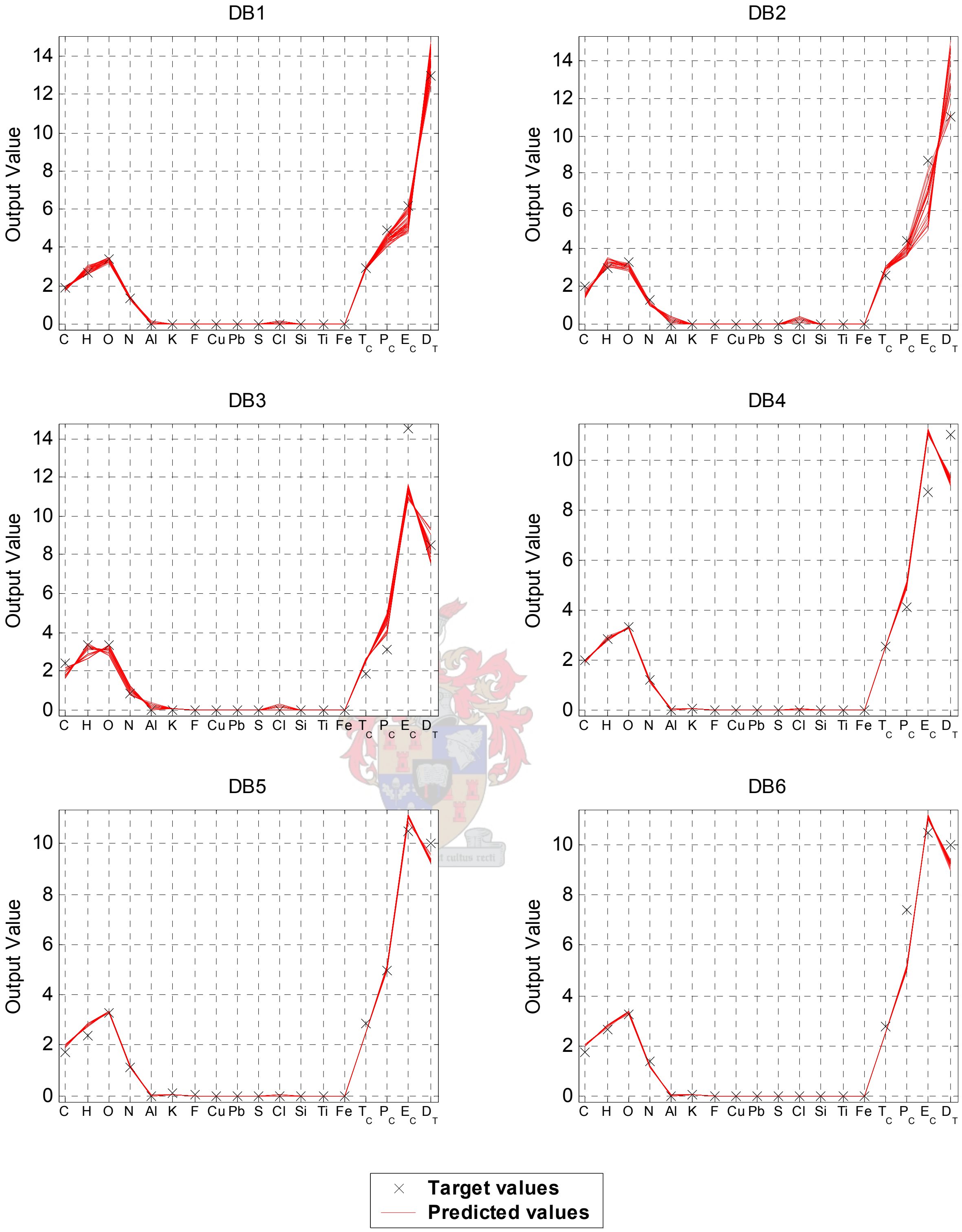

Fig. 5-47 Predictions made from the repeat measurements of input spectra using the optimum set of the overall RBFPLS models for rocket motors DB1 to DB6.

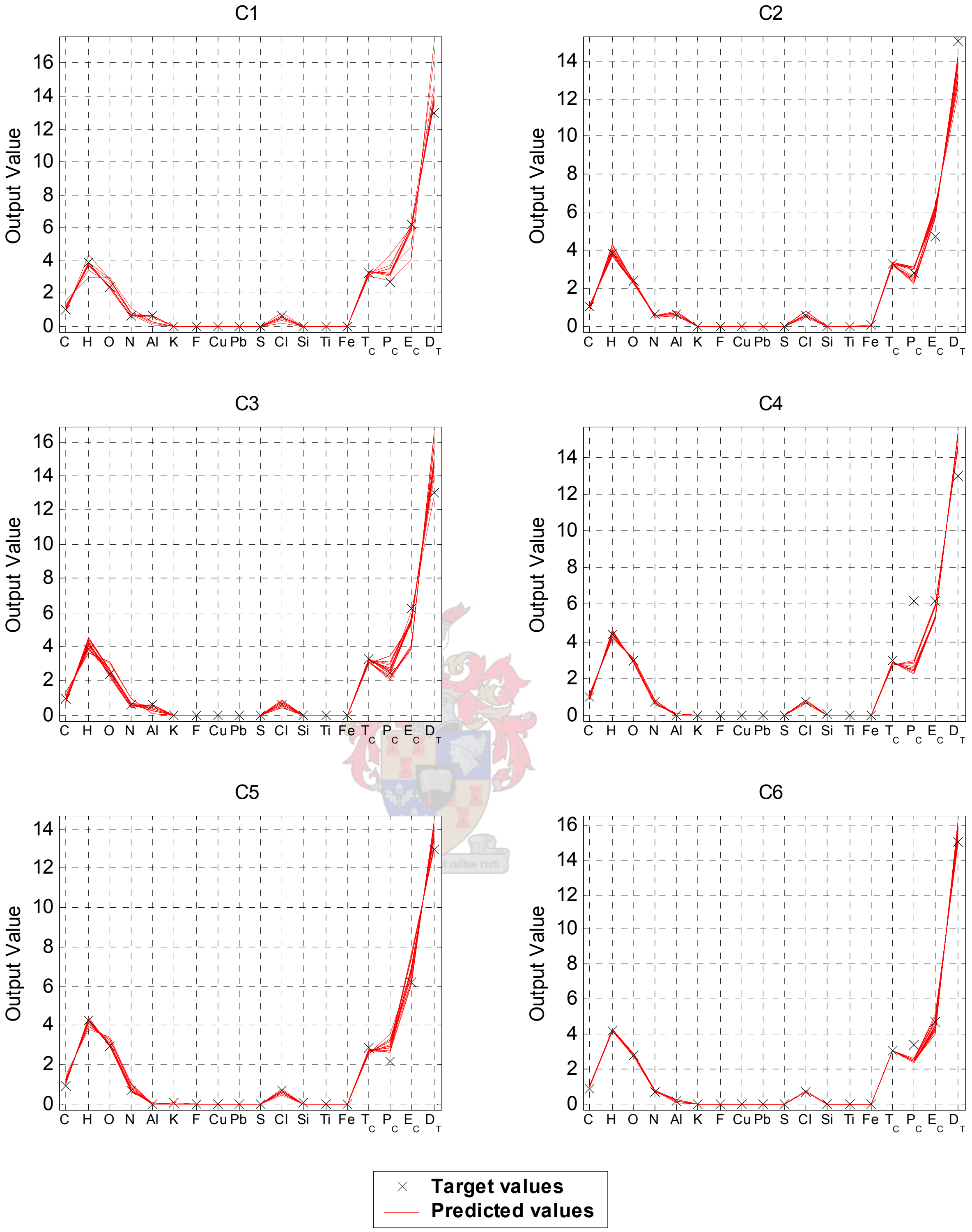

Fig. 5-48 Predictions made from the repeat measurements of input spectra using the optimum set of the overall RBFPLS models for rocket motors C1 to C6.

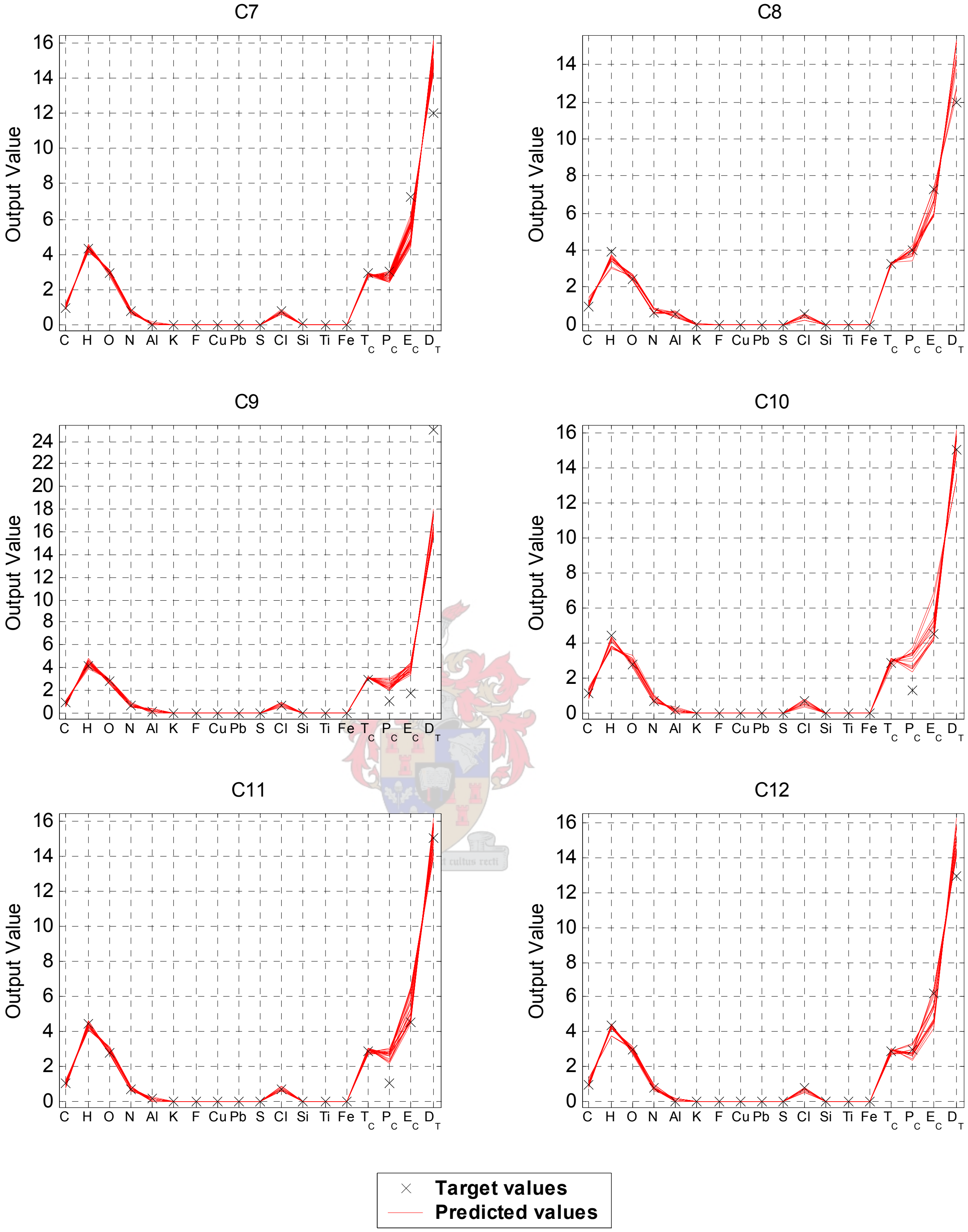

Fig. 5-49 Predictions made from the repeat measurements of input spectra using the optimum set of the overall RBFPLS models for rocket motors C7 to C12.

Chapter 6

Conclusions and Recommendations

The objective of finding a possibly simpler, more intuitive model compared to the feed-forward neural network with 146 nodes in a single hidden layer (forward model by Roodt [1998]) has come with constraints. The first and foremost constraint is that the available data is sparse with only 18 available rocket motor designs (independent data points). In addition the in- and output data are highly multivariate with 18 rocket motor design parameters and 146 spectral wavelengths in the middle IR band. One advantage is that the IR spectral measurements are repeated a number of times (4 to 44 repeats per rocket motor). For pattern recognition problems the data may be sufficient, but for modelling purposes there is a fairly severe lack of dimensional proportionality in the data. Due to these constraints a snowball effect is obtained whereby it becomes increasingly important to statistically keep a check on the model integrity whilst the statistics become more complex to deal with for ill-conditioned data.

The approach of building a model is one of basic statistical linear or non-linear regression theory. The convenience of non-linear black-box modelling often sets the trap of model over-parameterisation. It is found that there are basically 18 lack-of-fit degrees of freedom to work with from the 18 different rocket motor types when building a model. In order to make a prediction the model uses a certain number of these degrees of freedom, at least 3 to 5 lack-of-fit degrees of freedom should then be available as a check on the generalisation ability of the model.

The general perception is that there must be more independent data points than model parameters in order to avoid over-parameterisation. This train of thought is rooted in basic linear modelling theory. With reference to the highly credible research by Lawrence [1997] and from the results in this dissertation it is found that this is not necessarily true for non-linear modelling. Over-parameterisation can lead to optimum model complexities (see section 2.6.4) for making predictions on unseen data due to the nature of the iterative optimisation algorithms. The problem is therefore one of finding a suitable measure of model degrees of freedom for these non-linear models. This has been addressed by the pseudo-degrees of freedom calculated by making use of model cross-validation and overall model training.

Outlier detection shows that three of the repeated IR spectral measurements from rocket motor C8 can be removed. This is due to their significant deviations from the sample and their large leverages on the input space.

It is found that there is a significant degree of correlation between the IR absorbance values in the irradiance spectra. Linear PCA shows that just 3 principal components are required to explain 90% of the standardised spectral data variance (η^2). By contrast the rocket motor design parameters show little correlation with each other. From correlations and class-based PCA it is also shown that there is a clear class distinction between the DB- and C-class rocket motor designs.

Leave-one-out cross-validation is found to be the best method of getting an idea as to what the overall model performance is on predictions made for unseen data. In addition to this, cross-validation gives an indication of what the optimal model complexity is in order to achieve a model capable of effectively generalising the input-output data relationships. Over-parameterisation is therefore avoided in this way and the pseudo-degrees of freedom are used to check on the overall model complexity and lack-of-fit.

Leave-one-out cross-validation for the neural network PLS modelling techniques is not so straight-forward. This stems from the fact that a number of hidden nodes need to be decided upon for the inner model of each latent dimension. Each latent dimension inner model within PLS has to be fully cross-validated to avoid the situation where training occurs in the direction that favours the minimisation of the sum-squared error for the particular data point left out of a particular training session within cross-validation. This leads to the false conclusion that generalisation can be achieved at high model complexities. An analogy to this would be to train a feed-forward neural network on 17 data points and stopping the training epochs when the 18th, left out data point is best predicted. The resulting model could be over-parameterised and it says nothing about the specific model's generalising ability. The 'modified' cross-validation for neural network PLS is successfully applied in this work with PRESS-values going through the characteristic 'dip' indicating the best model complexity for generalisation.

A methodology of validating and building models with generalising abilities has thus been developed to test the candidate models, linear PLS, the feed-forward neural network, NNPLS (or EBNNPLS) and RBFPLS (or EBRBFPLS). In the process of applying these modelling techniques the OLS method of training radial-basis function neural networks has been modified to allow for the Gaussian spread parameters to be updated. This is done using an adaptive technique known here as the adaptive spread OLS algorithm (ASOLS).

The feed-forward neural network with the best generalisation ability for the forward model makes use of just two nodes in the single hidden layer. A similar approach taken by Lawrence [1997] to over-train the models as far as the number of epochs is concerned is taken here. This approach is essential in cross-validation so that the generalisation ability of a model is based on the parameterisation and not the extent of the training procedure. This neural network is markedly less complex compared to that obtained by Roodt [1998].

In the work done by Roodt [1998] the predictions for the two validation rocket motors, DB2 and C10 are fairly good. The predictions for the same two unseen data points from cross-validation produce similar results and are achieved with the simpler neural network architecture and no moving-average post-processing is required to remove any noise. The noiseless predictions show that the model is not over-parameterised. Care is taken not to train with too many epochs, yet the network built by Roodt [1998] still shows signs of over-parameterisation.

The forward RBFPLS models with ASOLS training of the inner PLS radial-basis-basis functions perform better compared to the K-means and OLS training algorithms.

A summary of the forward modelling results is shown in Table 4-7 in section 4.3. The NNPLS algorithm appears to be the most promising model. The average R^2_{cv}-value of 0.626 (0.746 for maximum R^2_{cv}) for all 146 output variables is satisfactory when considering the lack of data available. Some of the individual R^2_{cv}–values are even in excess of 0.8 in the H_2O absorption region. The IR absorbance region between the H_2O and CO_2 absorbance regions also produces relatively high R^2_{cv}–values. However, the predictability of the dominant CO_2 absorbance region is not as desired with R^2_{cv}–values as low as 0.3. These predictability trends for the absorbance regions discussed are found for all the models tested. This NNPLS model is simpler (307 parameters per output variable) compared to the feed-forward neural network proposed by Roodt [1998] (2921 parameters per output variable).

The average pseudo-dimension for the NNPLS model with 11 latent dimensions is 12.33. This leaves approximately 5 lack-of-fit degrees of freedom as a check for the model complexity. The feed-forward neural network with 146 hidden nodes has an average of 17.77 pseudo degrees of freedom. This indicates saturation and for a linear model it would represent a near spline fit to the training data.

The results from PCA and PLS modelling show that dimensional reduction is possible and does improve the modelling performances. This is also proven by the fact that the overall NNPLS model with 11 latent dimensions explains 99.8% of the X-block variance. The redundancy in the input space could be attributed to the lack of data for variables such as F, S, Si, Ti and Fe. NNPLS performs better compared to linear PLS, mainly because it better predicts the DB-class rocket motor spectra (although the unseen predictions for DB3 to DB6 remain poor). This shows that non-linearities in the model are required to compensate for both classes of rocket motor designs. The fact that NNPLS performs better than EBNNPLS shows that the dimensional reduction is best achieved by linear projections. After the fourth latent dimension in the overall NNPLS model the in- and output latent variables (scores) are best related by near linear prediction curves. From this discussion and the results from the simpler feed-forward neural network model it is evident that the nature of the forward modelling problem does not appear to be highly non-linear.

The sensitivity analysis done on the overall NNPLS model is based on making global changes to the input space. This analysis is identical to the one presented by Roodt [1998] and qualitatively outperforms the feed-forward neural network of that work. This is especially the case where polyester is used to replace HTPB/Isophoron as binder (C11-D) and the expected increase in irradiance is predicted.

The lack-of-fit F-statistic tests show that there is reason to doubt the adequacy of the model. This is not surprising when considering the limited available data. The result implies that the model is not adequate for application, but the NNPLS model appears to hold promise for development on larger, more representative data.

A summary of the reverse modelling results for all output variables pooled is shown in Table 5-2 in section 5.3. Due to the significantly smaller number of output variables used for the reverse modelling problem and the varying units of the variables, it is feasible to evaluate the optimum model complexities for each individual output variable. The summary of results for the optimum set of each individual output variable is shown in Table 5-3. The feed-forward neural network, NNPLS and RBFPLS models achieve similar results superior to those of linear PLS.

The reverse RBFPLS model with ASOLS training of the radial basis-functions for the inner PLS models produces results slightly superior to those of the other models. The optimal complexity for all the output variables pooled together is 5 latent dimensions. In the optimal set the complexities vary between 1 and 14 latent dimensions. The average number of latent dimensions and pseudo-degrees of freedom are relatively low for all the models. This proves that there is a fair amount of redundant information in the spectral data and that there are numerous lack-of-fit degrees of freedom. Over 90% X-block explained variance is achieved by the first latent dimension for all PLS models.

Even though the relatively low R_{cv}^2 –value of 0.427 for the optimum set of the RBFPLS models is a poor result, it does not reflect on their true performance. This is due to the data of output variables K, F, S, Si, Ti and Fe consisting of numerous zero-entries and the inability of the model to handle these irregularities. The R_{cv}^2 –values for C, H, O, N, Al, Cu, Pb, Cl and E_C range between 0.6 and 0.8. The R^2 –values calculated from the overall models range between 0.8 and 1.0 for C, H, O, N, Al, Cu, Pb, Cl, Si, E_C and D_T. The variables C, H, O, N, Al, Cu, Pb and Cl are predicted with reasonable precision as opposed to P_C, E_C and D_T, for which the confidence intervals are very broad.

There appears to be no bias in prediction ability towards any one of the rocket motor design classes (DB- or C-class) for the feed-forward neural network, NNPLS and RBFPLS models. This can be attributed to the non-linearities in the models. The linear PLS model tends to better predict the C-class rocket motor design parameters compared to those of the DB-class rocket motors. The NNPLS model performs better compared to the EBNNPLS model. Similarly to the forward problem the reverse problem is therefore not all that non-linear in nature.

In sections 4.2.1 and 5.2.1 it is shown how the linear PLS regression coefficients can be used to indicate the weighting of each input variable. This qualitative information from the linear model adds value to the analysis. In the case of the forward model it is evident that the strategic addition of additives can ‘mould’ the IR emission spectra in predetermined bands.

The F-statistics for testing the null hypothesis of equivalence for $MSECV_j$ and MSE_j show that measures of residual variances can be used to estimate the model residual variances, $\sigma_{e,j}$. Similar levels of accuracy can also be expected from predictions made on unseen data points as for the overall model predictions.

The loadings of the first three primary latent dimensions of the overall reverse NNPLS model provide information on the influential regions of the input spectra. The H_2O and CO_2 absorbance regions exert the most leverage on the first and third latent dimensions. The absorbance region between the H_2O and CO_2 absorbance exerts the most leverage for the second latent dimension. One of the advantages of PLS modelling is that this qualitative information is easily obtained by analysing the loading values.

The repeat measurements for the IR irradiance spectra are used to validate the accuracy of the overall reverse RBFPLS model predictions. It is found that the predictions are within the confidence limits and the RMSE bands.

The predictabilities for the reverse model are not ideal, but the results hold promise for further development with a larger data set. One or two of the insignificant variables such as F and S could be omitted, if necessary. The alternative is to ensure that the additional data contains enough information regarding these variables.

In both the forward and reverse modelling problems the dimensional reduction using PLS helps to work around the dimensional imbalance in the data and therefore obtain models with a fair amount of predictive ability. All the PLS models for both the reverse and forward modelling problems work with X-block explained variances of over 90%. This means that in the case of the reverse PLS models with 5 latent dimensions, 141 input dimensions contain redundant information. The feed-forward neural networks effectively also reduce dimensionality in the hidden layer, however redundant information is not excluded. The feed-forward neural networks thus work with 100% of the input space variance and this may explain why the neural network PLS models perform slightly better in comparison.

In the work by Roodt [1998] it is reported at the time that a single field trial costs in the vicinity of $50 000. Data acquisition is therefore extremely costly and the motivation of further acquiring data is not justified unless the possibility of building a model seems promising. The work done by Roodt [1998] shows promise and that the modelling may be possible. However, the approach developed is more one of finding a function that fits the data set and not one of developing a parametrically optimised model. The approach developed in this work shows that the development of a model capable of generalisation may be possible even if the models developed here are not recommended for application. This study may therefore be justified in providing a framework from which to motivate funding for the development of a more robust model.

It is recommended that the way forward is to collect data from field trials where more different types of rocket motor formulations are prepared and tested. This will expand the information covering the input spaces and therefore also the available degrees of freedom. Repeat measurements are essential, but only three to ten repeats are sufficient, depending on the available budget. It is further recommended to develop a design of experiments in order to make sure the input space is covered by as few experiments as possible.

More robust models can be developed for data with inherent linear input-output relationships such as linear PLS. The results from this work show that it may be possible to build two separate linear models, one for each class of rocket motor design (DB- and C-classes). This is due to the improvements in predictions made from the non-linear models being primarily attributed to the improved compensation for the DB-class rocket motors. It has further been shown that a technique such as class-based PCA can be used to effectively cluster the data into the two classes. A classifier can then be used to distinguish between the two classes and choose which of the two linear models is applicable. A data set with approximately equal and adequate numbers of data points for the different rocket motor design classes is recommended.

References

1. **Anton, H., Rorres, C. (1994)**, *Elementary Linear Algebra,* Toronto: John Wiley & Sons, Inc.

2. **Baffi, G., Martin, E.B., Morris, A.J. (1999a)**, *Non-linear Projection to Latent Structures Revisited: The Quadratic PLS Algorithm,* Computers and Chemical Engineering **23**: p. 395-411.

3. **Baffi, G., Martin, E.B., Morris, A.J. (1999b)**, *Non-linear Projection to Latent Structures Revisited (the Neural Network PLS algorithm),* Computers and Chemical Engineering **23**: p. 1293 - 1307.

4. **Baffi, G., Martin, E., Morris, J. (2002)**, *Prediction Intervals for Non-linear Projection to Latent Structures Regression Models,* Chemometrics and Intelligent Laboratory Systems **61**: p. 151-165.

5. **Bates, D. M., Watts, D.G. (1988)**, *Nonlinear Regression Analysis and its Applications,* New York: John Wiley & Sons.

6. **Beale, M., Demuth, H. (2000)**, *Neural Network Toolbox for Use with Matlab,* Boston: The Mathworks Inc.

7. **Bertran, E., Blanco, M., Maspoch, S., Ortiz, M.C., Sanchez, M.S., Sarabia, L.A. (1999)**, *Handling Intrinsic Non-linearity in Near-infrared Reflectance Spectroscopy,* Chem. Intell. Lab. Syst. **49**: p. 215 - 224.

8. **Bishop, C. M. (1995)**, *Neural Networks for Pattern Recognition,* Oxford University Press.

9. **Blanco, M., Coello, J., Iturriaga, S., Maspoch, S., Pages, J. (1999)**, *Calibration in Non-linear Near Infrared Reflectance Spectroscopy: A comparison of Several Methods,* Anal. Chim. Acta **384**: p. 207 - 214.

10. **Blanco, M., Coello, J., Iturriaga, H., Maspoch, S., Pages, J. (2000)**, *NIR Calibration in Non-linear Systems: Different PLS Approaches and Neural Networks,* Chem. Intell. Lab. Syst. **50**: p. 75 - 82.

11. **Brereton, R. G. (1990)**, *Chemometrics Applications of Mathematics and Statistics to Laboratory Systems,* Ellis Horwood.

12. **Brereton, R. G. (1992)**, *Multivariate Pattern Recognition in Chemometrics,* Amsterdam: Elsevier Science Publishers.

13. **Bulsari, A. B. (1995)**, *Neural Networks for Chemical Engineers,* Amsterdam: Elsevier.

14. **Chen, S., Cowan, C.F.N., Grant, P.M. (1991)**, *Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks,* IEEE Transactions on Neural Networks **2**(2): p. 302-309.

15. **Chryssolouris, G., Lee, M., Ramsey, A. (1996)**, *Confidence Interval Prediction for Neural Network Models,* IEEE Transactions on Neural Networks **7**(1): p. 229-232.

16. **Diamantaris, K. I., Kung, S.Y. (1996)**, *Principle Component Neural Networks Theory and Applications,* John Wiley and Sons, Inc.

17. **Donaldson, J. R., Schnabel, R.B. (1987)**, *Computational Experience With Confidence Regions and Confidence Intervals for Nonlinear Least Squares,* Technometrics **29**(1): p. 67-82.

18. **Draper, N. R., Smith,H. (1981)**, *Applied Regression Analysis,* New York: John Wiley & Sons.

19. **Fukunaga, K. (1972)**, *Introduction to Statistical Pattern Recognition,* New York: Academic Press, p. 225-254.

20. **Geisser, S. (1975)**, *The Predictive Sample Reuse Method with Applications,* J. Amer. Statist. Assoc. **70**(350): p. 320 - 327.

21. **Geladi, P., Kowalski, B.R. (1986)**, *An Example of 2-block Predictive Partial Least-Squares Regression with Simulated Data,* Analytica Chimica Acta **185**: p. 19-32.

22. **Geladi, P., Kowalski, B.R. (1986)**, *Partial Least Squares Regression: A Tutorial,* Analytica Chimica Acta **185**: p. 1-17.

23. **Haaland, D. M., Thomas, E.V. (1988)**, *Partial Least-Squares Methods for Spectral Analyses. 1. Relation to Other Quantitative Calibration Methods and the Extraction of Quantitative Information,* 60 **60**(11): p. 1193-1201.

24. **Hadjiiski, L., Geladi, P., Hopke, P. (1999)**, *A Comparison of Modelling Nonlinear Systems with Artificial Neural Networks and Partial Least Squares,* Chem. Intell. Lab. Syst. **49**: p. 91 - 103.

25. **Hagan, M. T., Mohammed, B.M. (1994)**, *Training Feedforward Networks with the Marquardt Algorithm,* IEEE Transactions on Neural Networks **5**(6): p. 989-993.

26. **Hartman, E. J., Keeler, J.D., Kowalski, J.M. (1990)**, *Layered Neural Networks with Gaussian Hidden Units as Universal Approximations,* Neural Computation **2**(2): p. 210-215.

27. **Haykin, S. (1999)**, *Neural Networks - A Comprehensive Foundation,* Prentice-Hall.

28. **Hogg, R. V., Ledolter, J. (1992)**, *Applied Statistics for Engineers and Scientists,* New York: Macmillan Publishing Company.

29. **Holcomb, T. R., Morari, M. (1992)**, *PLS/Neural Networks,* Computers chem. Engng **16**(4): p. 393-411.

30. **Johnson, R. A., Wichern, D.W. (1988)**, *Applied Multivariate Statistical Analysis,* Prentice-Hall.

31. **Johnson, R. A. (1994)**, *Miller and Freund's Probability and Statistics for Engineers,* New Jersey: Prentice Hall, Englewood Cliffs.

32. **Lawrence, S., Giles, C.L., Tsoi, A.C. (1997)**. *Lessons in Neural Network Training: Overfitting May Be Harder than Expected*, Proceedings from the Fourteenth National Conference on Artificial Intelligence, Menlo Park, California, AAAI Press, p. 540-545.

33. **Ljung, L. (1987)**, *System Identification Theory for the User,* Prentice-Hall.

34. **Lorber, A., Wangen, L.E., Kowalski, B.R. (1987)**, *A Theoretical Foundation for the PLS Algorithm,* Journal of Chemometrics **1**: p. 19-31.

35. **Lorber, A., Kowalski, B.R. (1988)**, *A Note on the Use of the Partial Least-Squares Method for Multivariate Calibration,* Applied Spectroscopy **42**(8): p. 1572-1574.

36. **Malthouse, E.C. Tamhane, A.C., Mah, R.S.H. (1997)**, *Nonlinear Partial Least Squares,* Computers chem. Engng **21**(8): p. 875-890.

37. **Marquardt, D. W. (1963)**, *An Algorithm for Least-Squares Estimation of Non-linear Parameters,* J. Soc. Indust. Appl. Math. **11**(2): p. 431-441.

38. **Martens, H., Naes, T. (1989)**, *Multivariate Calibration,* New York: John Wiley & Sons.

39. **Nguyen, D., Widrow, B. (1990)**. *Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights*, Proceedings from the International Joint Conference on Neural Networks, p. 21 - 26.

40. **Proakis, J. G., Manolakis, D.G. (1996)**, *Digital Signal Processing,* New Jersey: Prentice-Hall.

41. **Qin, S. J., McAvoy, T.J. (1992)**, *Nonlinear PLS Modeling Using Neural Networks,* Computers chem. Engng. **16**(4): p. 379-391.

42. **Riedmiller, M., Braun, H. (1993)**, *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm,* Proceeding of the IEEE International Conference on Neural Networks **1**: p. 586-591.

43. **Roodt, J. H. S. (1998)**, *The Prediction of the Emission Spectra of Solid Rocket Propellants,* Chemical Engineering, Stellenbosch, University of Stellenbosch.

44. **Rumelhart, D. E., Hinton, G.E., Williams, R.J. (1986)**, *Learning Representations by Back-propagating Errors,* Nature (London) **323**(9 October): p. 533-536.

45. **Schmitt, M. (2001)**, *Neural Networks with Local Receptive Fields and Superlinear VC Dimension,* Bochum, Ruhr University of Bochum, Germany.

46. **Sharaf, M. A., Illman, D.L., Kowalski, B.R. (1986)**, *Chemometrics,* John Wiley & Sons.

47. **Snee, R. D. (1977)**, *Validation of Regression: Methods and Examples,* Technometrics **19**(4): p. 415 - 428.

48. **Sontag, E. D. (1998)**, *VC Dimension of Neural Networks,* Department of Mathematics, State University of New Jersey.

49. **Stone, M. (1974)**, *Cross-validatory Choice and Assessment of Statistical Predictions,* J. Roy. Statist. Soc. **B**(36): p. 111 - 147.

50. **Tabachnick, B. G., Fidell, L.S. (1983)**, *Using Multivariate Statistics,* New York: Harper & Row.

51. **Urban Hjorth, J. S. (1994)**, *Computer Intensive Statistical Methods,* London: Chapman & Hall.

52. **Van der Voet, H. (1999)**, *Pseudo-degrees of Freedom for Complex Predictive Models: The Example of Partial Least Squares,* J. Chemometrics **13**: p. 195-208.

53. **Vardeman, S. B. (1994)**, *Statistics for Engineering Problem Solving,* Boston: PWS Publishing Co.

54. **Wise, B. M., Gallagher, N.B. (2000)**, *PLS Toolbox 2.1 for use with MATLAB,* Manson: Eigenvector Research, Inc.

55. **Wold, H. (1966)**, *Estimation of Principal Components and Related Models by Iterative Least Squares,* : p. .

56. **Wold, S. (1978)**, *Cross-validatory Estimation of the Number of Components in Factor and Principle Components Models,* Technometrics **20**(November): p. 397 - 405.

57. **Wold, S., Kettaneh-Wold, N., Skagerberg, B. (1989)**, *Nonlinear PLS Modeling,* Chemometrics and Intelligent Laboratory Systems **7**: p. 53-65.

58. **Zupan, J., Gasteiger, J. (1993)**, *Neural Networks for Chemists - An Introduction,* New York: VCH Publishers.

Appendix A

Processed Results of Forward Modelling

A.1 Linear PLS

Table A-1 Explained variances calculated on the standardised data with increasing latent dimension (LD) on the overall linear PLS model.

LD	X-Block (input)		Y-block (outputs)	
	% η^2	Cumulative % η^2	% η^2	Cumulative % η^2
1	46.447	46.447	38.605	38.605
2	15.144	61.592	22.433	61.038
3	7.815	69.407	4.963	66.001
4	7.501	76.908	3.038	69.039
5	6.640	83.548	2.850	71.889
6	4.017	87.566	2.890	74.779
7	5.718	93.283	0.948	75.726
8	2.644	95.927	1.484	77.211
9	1.908	97.835	1.069	78.279
10	1.530	99.365	0.434	78.713
11	0.467	99.832	0.481	79.194
12	0.079	99.911	0.171	79.365
13	0.086	99.997	0.075	79.440
14	0.002	99.999	0.561	80.001
15	0.001	100.000	0.500	80.501
16	0.000	100.000	2.916	83.417
17	0.000	100.000	1.025	84.443
18	0.000	100.000	0.000	84.443

Table A-2 Summary of results for the linear PLS model evaluated on the rescaled data.

	Cumulative % η^2											
LD	X-block (inputs)	Y-block (outputs)	Ouputs (max)	Average pdf	Average R^2_{cv}	Average $R^2_{cv(max)}$	Average R^2	Average R^2_{max}	PRESS	SSEP	$PRESS_{min}$	$SSEP_{min}$
1	48.44	29.26	39.22	2.25	0.250	0.318	0.602	0.685	775.80	568.20	30.64	23.509
2	50.76	60.86	67.48	4.12	0.431	0.515	0.763	0.831	611.80	314.59	22.65	12.580
3	61.12	67.43	75.37	6.94	0.419	0.494	0.794	0.867	777.40	261.85	28.33	9.525
4	78.90	77.20	80.18	7.58	0.426	0.495	0.814	0.880	893.60	183.30	34.94	7.664
5	86.94	81.32	86.19	10.36	0.417	0.474	0.832	0.908	1305.60	150.22	51.22	5.340
6	91.96	86.63	90.60	11.84	0.418	0.473	0.848	0.922	1708.00	107.50	67.50	3.635
7	94.50	88.05	92.02	11.62	0.435	0.494	0.855	0.928	1306.30	96.15	51.38	3.088
8	94.72	89.77	93.97	12.01	0.436	0.495	0.863	0.938	1138.60	82.29	44.47	2.334
9	95.31	90.30	94.69	12.82	0.422	0.480	0.869	0.944	1110.00	78.05	43.12	2.052
10	99.22	90.45	94.82	12.88	0.427	0.493	0.872	0.949	1072.40	76.84	41.35	2.003
11	99.69	90.52	94.98	12.42	0.461	0.541	0.876	0.957	688.50	76.28	26.10	1.943
12	99.94	90.54	94.97	12.87	0.412	0.504	0.877	0.960	839.80	76.08	32.12	1.944
13	100.0	90.66	95.07	15.01	0.213	0.276	0.878	0.960	1071.70	75.14	41.84	1.905
14	100.0	90.73	95.24	16.39	0.044	0.064	0.882	0.971	4525	74.54	173.53	1.842
15	100.0	91.28	95.84	17.75	0.101	0.077	0.885	0.974	221000	70.14	7800	1.610
16	100.0	94.70	99.51	17.95	0.082	0.067	0.901	0.991	5942000	42.63	207900	0.191
17	100.0	95.13	100.0	18.00	0.098	0.088	0.907	1.000	14266000	39.22	480800	0.000
18	100.0	95.13	100.0	18.00	0.052	0.054	0.907	1.000	75218000	39.22	2458700	0.000

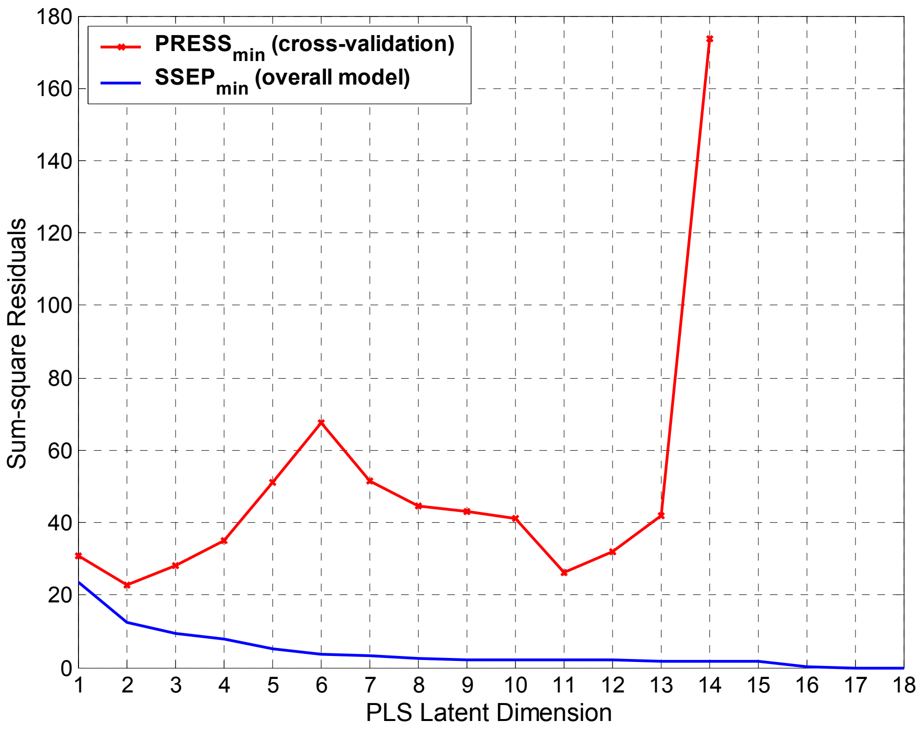

Fig. A-1 The minimum sum-squared residuals obtained from building a linear PLS model.

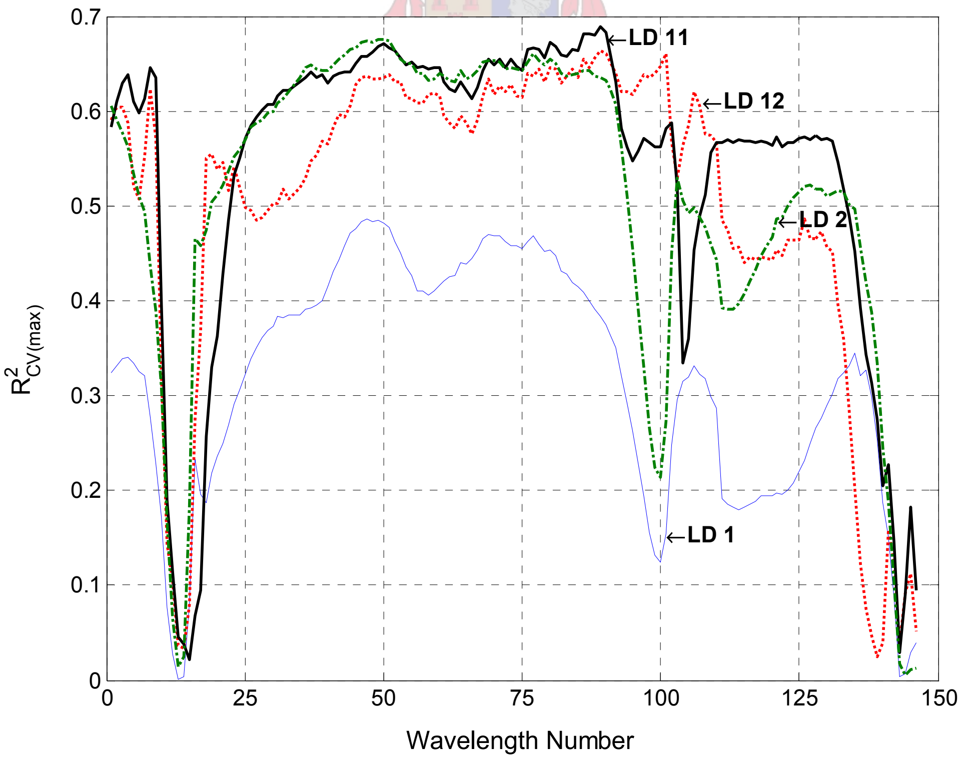

Fig. A-2 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their mean targets for linear PLS.

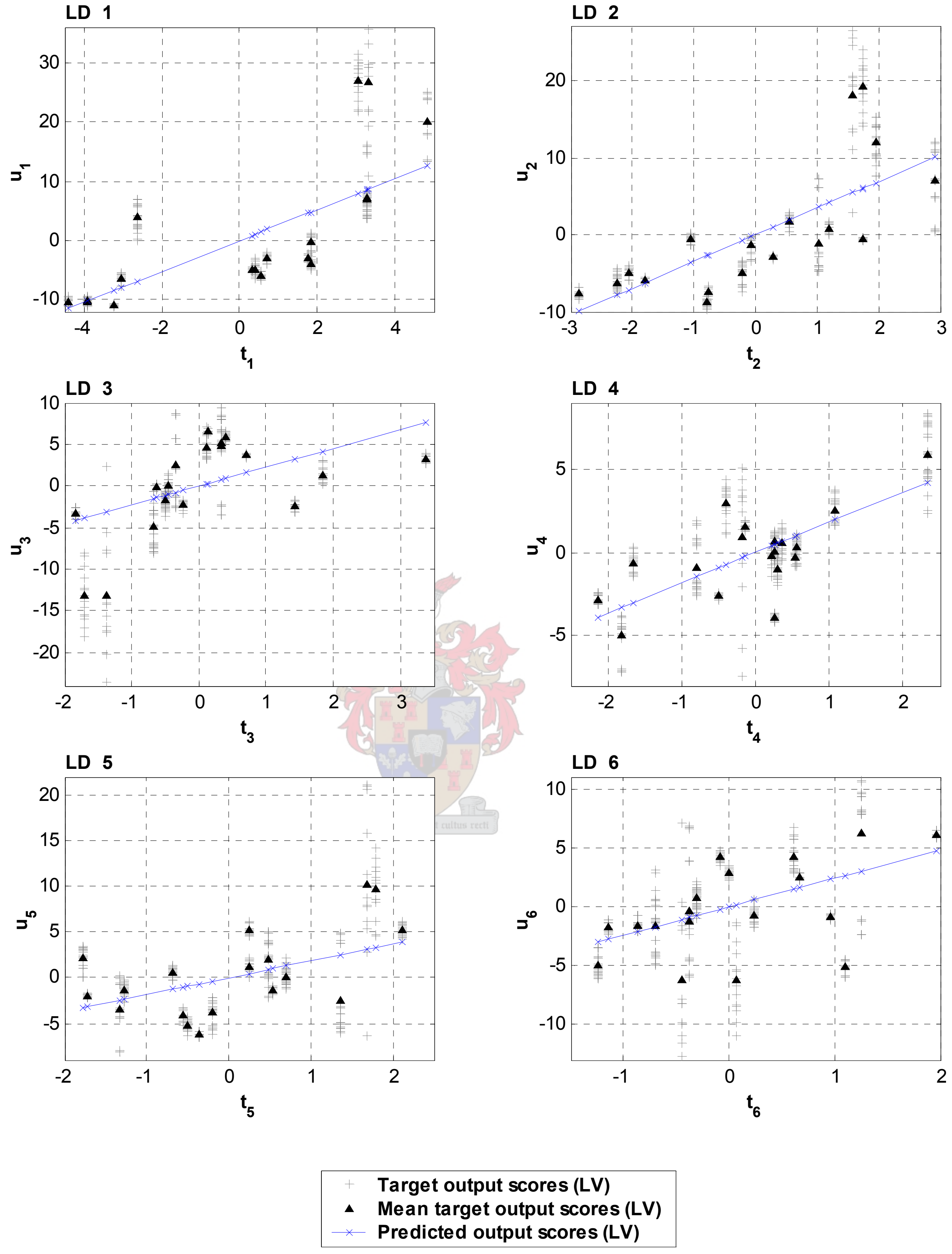

Fig. A-3 The target data and inner model relationships of the in- and output scores calculated for the first 6 latent dimensions using the overall linear PLS model.

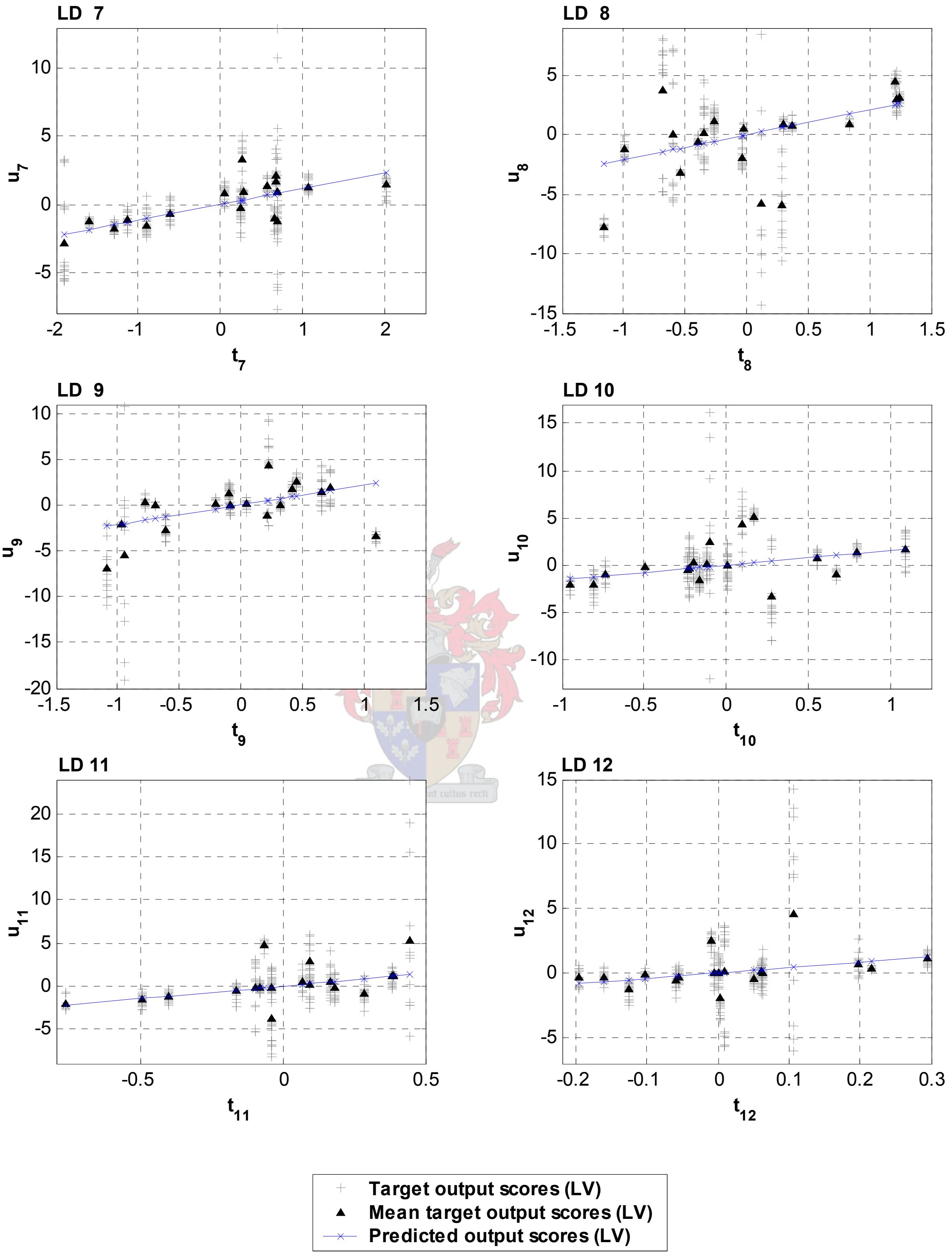

Fig. A-4 The target data and inner model relationships of the in- and output scores calculated for latent dimensions 7 to 12 using the overall linear PLS model.

A.2 Feed-forward Neural Network

Table A-3 Leave-one-out cross-validation was repeated 10 times for each single hidden layer neural network.

No. of hidden nodes	PRESS Run 1	PRESS Run 2*	PRESS Run 3	PRESS Run 4	PRESS Run 5	PRESS Run 6	PRESS Run 7	PRESS Run 8	PRESS Run 9	PRESS Run 10	PRESS Average
1	763.20	810.88	809.34	870.08	1166.07	834.18	781.16	1058.75	856.40	809.34	875.94
2	807.15	613.23	707.81	1183.03	702.96	792.92	921.09	798.16	863.76	707.81	809.79
10	1150.93	1186.23	1096.97	793.80	940.96	728.12	965.34	1049.27	1009.72	1096.97	1001.83
15	843.86	1227.08	773.98	932.91	945.65	1443.76	907.94	1055.31	768.22	773.98	967.27
20	1192.43	1377.30	1053.77	1227.03	1281.44	1304.66	1039.94	1340.81	1591.87	1053.77	1246.30
30	1155.99	1499.16	2075.66	1257.91	1490.20	874.94	1408.77	1635.16	1910.54	2075.66	1538.40
40	1200.13	1495.66	1269.90	1648.34	1200.90	1682.60	1445.11	1704.67	1351.66	1269.90	1426.89
50	1807.56	1911.12	1703.01	1449.64	1609.50	2292.21	1704.64	1454.09	1512.31	1703.01	1714.71
100	2759.38	2874.75	2246.80	2168.46	2313.36	2737.85	2290.72	2301.24	2665.48	2246.80	2460.49
146	3427.61	4414.43	3892.68	3560.66	3446.77	3586.11	3846.26	3301.36	2985.87	3892.68	3635.44

*** The best run chosen for further evaluation**

Table A-4 The overall model training was repeated 5 times for each single hidden layer neural network.

No. of hidden nodes	SSEP Run 1*	SSEP Run 2	SSEP Run 3	SSEP Run 4	SSEP Run 5	SSEP Average
1	161.99	161.67	162.02	162.05	161.74	161.89
2	52.74	53.80	81.44	93.00	56.84	67.56
10	41.50	41.99	42.41	41.33	40.62	41.57
15	42.15	40.82	40.77	41.09	41.76	41.32
20	41.35	40.80	40.76	40.60	41.23	40.95
30	40.51	40.84	41.34	41.61	40.75	41.01
40	41.81	40.86	40.24	40.66	40.47	40.81
50	40.36	40.07	40.23	41.38	40.40	40.49
100	40.59	39.95	39.65	39.76	40.89	40.17
146	39.56	39.69	39.66	39.51	39.97	39.68

*** The best run chosen for further evaluation**

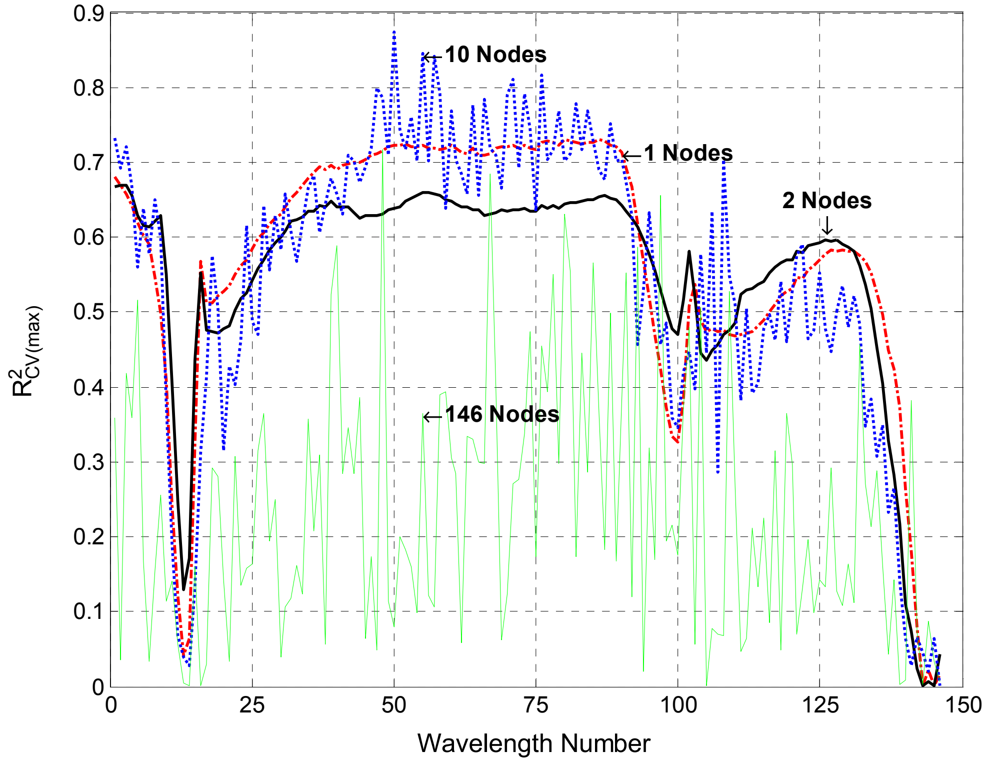

Fig. A-5 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their mean targets for the neural network models.

Table A-5 Summary of results using feed-forward networks. Results are evaluated on the rescaled data from the best of the repeated runs.

No. of hidden nodes	Cumulative % η^2		Average pdf	Average R^2_{cv}	Average $R^2_{cv(max)}$	Average R^2	Average R^2_{max}	PRESS	SSEP	$PRESS_{min}$	$SSEP_{min}$
	Y-block (outputs)	Ouputs (max)									
1	79.8	88.9	10.47	0.446	0.572	0.750	0.865	810.9	162.0	26.95	4.311
2	93.4	98.3	12.91	0.417	0.548	0.803	0.907	613.2	52.7	22.24	0.655
10	94.8	99.7	16.46	0.425	0.549	0.839	0.987	1186.2	41.5	42.08	0.125
15	94.7	99.6	16.09	0.376	0.502	0.836	0.979	1227.1	42.2	42.46	0.174
20	94.8	99.7	16.54	0.368	0.495	0.839	0.987	1377.3	41.3	50.18	0.126
30	95.0	99.8	17.16	0.384	0.478	0.842	0.993	1499.2	40.5	54.39	0.068
40	94.8	99.6	16.81	0.304	0.397	0.839	0.989	1495.7	41.8	58.66	0.160
50	95.0	99.8	17.09	0.280	0.369	0.841	0.993	1911.1	40.4	68.16	0.066
100	94.9	99.8	17.39	0.251	0.319	0.842	0.996	2874.7	40.6	111.22	0.085
146	95.1	100.0	17.77	0.175	0.234	0.844	0.999	4414.4	39.6	175.40	0.019

A.3 Neural Network PLS

Table A-6 Original cross-validation and overall model-building results for single runs using EBNNPLS (see discussion in section 4.1.2).

LD	PRESS	SSEP
1	521.05	131.20
2	494.51	50.97
3	406.18	46.74
4	386.93	46.19
5	362.92	43.94
6	324.66	43.04
7	325.67	41.89
8	307.36	41.88
9	303.46	40.34
10	276.22	40.24
11	226.96	40.15
12	218.29	40.10
13	209.15	40.05
14	212.36	40.00
15	199.22	39.85
16	190.02	39.65
17	165.18	39.58
18	152.29	39.55

Table A-7 The overall model training was repeated 5 times for each LD using the NNPLS algorithm.

LD	SSEP Run 1	SSEP Run 2	SSEP Run 3*	SSEP Run 4	SSEP Run 5	SSEP Average
1	212.7	259.5	195.9	244.2	261.6	234.8
2	107.5	199.6	88.3	133.1	188.9	143.5
3	68.8	75.1	58.0	71.3	93.1	73.3
4	63.4	65.7	53.8	67.3	59.0	61.8
5	55.5	60.0	48.5	56.2	54.9	55.0
6	53.7	58.1	47.9	56.0	52.7	53.7
7	52.7	50.5	46.6	51.7	51.4	50.6
8	51.6	48.6	46.5	46.8	49.2	48.6
9	49.9	48.5	46.6	46.6	48.7	48.1
10	48.5	46.9	45.6	46.5	48.0	47.1
11	48.4	46.3	45.4	45.5	47.9	46.7
12	47.7	45.9	43.9	45.5	47.1	46.0
13	46.9	45.3	43.9	43.4	46.1	45.1
14	43.7	43.6	41.7	42.9	42.2	42.8
15	43.0	41.9	41.0	42.1	41.8	41.9
16	41.2	41.4	39.9	40.0	41.6	40.8
17	41.1	41.3	39.6	39.9	41.2	40.6
18	40.5	39.7	39.6	39.8	41.1	40.1

*** The best run chosen for further evaluation**

Table A-8 Leave-one-out cross-validation was repeated 10 times for each LD using NNPLS.

LD	PRESS Run 1	PRESS Run 2	PRESS Run 3	PRESS Run 4	PRESS Run 5	PRESS Run 6	PRESS Run 7	PRESS Run 8	PRESS Run 9	PRESS Run 10*	PRESS Average
1	553.0	587.1	553.3	553.3	687.8	540.9	595.1	631.9	555.6	553.3	581.1
2	621.7	789.6	576.7	576.7	626.5	542.3	607.7	587.3	554.7	576.7	606.0
3	567.0	733.8	470.4	470.4	579.3	462.7	517.7	574.7	449.3	470.4	529.6
4	541.6	653.8	441.4	441.4	564.0	475.4	502.2	544.1	349.6	441.4	495.5
5	472.7	623.1	360.5	360.5	480.7	386.1	431.3	500.7	321.8	360.5	429.8
6	469.2	658.8	357.1	357.1	469.8	381.0	424.5	492.8	324.1	357.1	429.2
7	471.0	690.0	350.6	350.6	477.7	384.4	413.5	534.7	324.6	350.6	434.8
8	493.4	703.8	341.5	341.5	489.8	396.0	388.8	529.5	331.0	341.5	435.7
9	495.8	683.6	296.2	296.2	483.0	409.8	390.3	495.2	333.7	296.2	418.0
10	487.9	692.6	295.5	295.5	480.3	405.7	391.5	494.2	321.2	295.5	416.0
11	477.4	493.6	258.1	258.1	421.6	402.7	344.7	339.6	315.6	258.1	357.0
12	426.7	488.9	289.9	289.9	437.0	418.6	355.8	361.2	324.1	289.9	368.2
13	399.0	1610.3	320.0	320.0	808.1	449.6	1563.4	364.0	315.1	320.0	646.9
14	400.9	1513.8	313.7	313.7	744.4	536.0	1577.4	368.5	346.7	313.7	642.9
15	440.5	1505.8	319.3	319.3	762.7	527.1	1660.0	376.8	342.1	319.3	657.3
16	451.0	1548.3	294.0	294.0	742.5	475.7	1780.8	369.4	349.4	294.0	659.9
17	519.0	1573.4	315.7	315.7	773.4	1008.5	1781.6	476.1	348.2	315.7	742.7
18	1769.2	1552.0	2286.9	2286.9	793.2	985.2	1814.0	875.0	353.6	2286.9	1500.3

*** The best run chosen for further evaluation**

Table A-9 The number of hidden nodes for each latent dimension inner model obtained for each modified cross-validation repeat run using NNPLS.

LD	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Rounded Average
1	4	3	6	6	3	12	4	4	15	6	6
2	4	1	4	4	4	7	3	2	6	4	4
3	2	2	4	4	3	4	2	1	5	4	3
4	9	1	4	4	2	2	2	1	6	4	4
5	2	1	1	1	1	3	6	1	3	1	2
6	13	4	1	1	2	1	1	3	1	1	3
7	1	1	3	3	12	3	7	1	10	3	4
8	2	1	2	2	4	1	1	1	4	2	2
9	4	3	2	2	2	5	1	1	7	2	3
10	1	1	1	1	1	4	1	1	10	1	2
11	1	2	1	1	6	1	7	1	11	1	3
12	4	10	1	1	1	1	1	5	1	1	3
13	5	2	12	12	2	10	7	7	16	12	9
14	2	7	9	9	9	1	5	1	7	9	6
15	8	13	12	12	15	18	9	11	8	12	12
16	1	8	10	10	8	9	1	9	13	10	8
17	9	1	10	10	17	4	1	6	19	10	9
18	3	6	1	1	14	8	8	1	21	1	6

Table A-10 Summary of results using NNPLS. Results are evaluated on the rescaled data from the best of the repeated runs.

	Cumulative % η^2											
LD	X-block (inputs)	Y-block (outputs)	Ouputs (max)	Average pdf	Average R^2_{cv}	Average $R^2_{cv(max)}$	Average R^2	Average R^2_{max}	PRESS	SSEP	$PRESS_{min}$	$SSEP_{min}$
1	48.44	75.59	83.31	8.59	0.448	0.557	0.737	0.840	553.3	195.9	21.07	6.454
2	60.57	89.00	92.04	9.83	0.440	0.560	0.771	0.861	576.7	88.3	22.18	3.078
3	80.00	92.77	97.60	11.10	0.534	0.669	0.799	0.908	470.4	58.0	13.30	0.929
4	86.04	93.30	98.01	11.48	0.553	0.689	0.809	0.922	441.4	53.8	12.02	0.771
5	90.45	93.96	98.58	11.68	0.580	0.711	0.815	0.931	360.5	48.5	9.56	0.548
6	94.16	94.04	98.67	11.68	0.583	0.710	0.816	0.934	357.1	47.9	9.43	0.516
7	96.78	94.19	98.89	11.97	0.597	0.725	0.820	0.944	350.6	46.6	9.23	0.430
8	97.62	94.20	98.90	11.96	0.614	0.742	0.821	0.946	341.5	46.5	8.75	0.427
9	98.15	94.19	98.87	11.89	0.636	0.758	0.822	0.947	296.2	46.6	7.76	0.436
10	99.64	94.32	98.95	12.05	0.633	0.755	0.824	0.951	295.5	45.6	7.71	0.408
11	99.76	94.34	98.98	12.33	0.626	0.746	0.825	0.954	258.1	45.4	6.64	0.395
12	99.94	94.53	99.25	13.02	0.600	0.727	0.828	0.960	289.9	43.9	7.62	0.291
13	100.00	94.53	99.24	13.96	0.557	0.694	0.831	0.963	320.0	43.9	8.55	0.295
14	100.00	94.81	99.67	15.89	0.568	0.700	0.839	0.985	313.7	41.7	8.53	0.129
15	100.00	94.90	99.76	16.16	0.561	0.691	0.840	0.988	319.3	41.0	8.91	0.091
16	100.00	95.02	99.90	16.46	0.578	0.700	0.842	0.993	294.0	39.9	8.32	0.037
17	100.00	95.07	99.95	16.84	0.583	0.693	0.843	0.997	315.7	39.6	9.07	0.018
18	100.00	95.07	99.96	17.49	0.347	0.407	0.843	0.998	2286.9	39.6	97.82	0.016

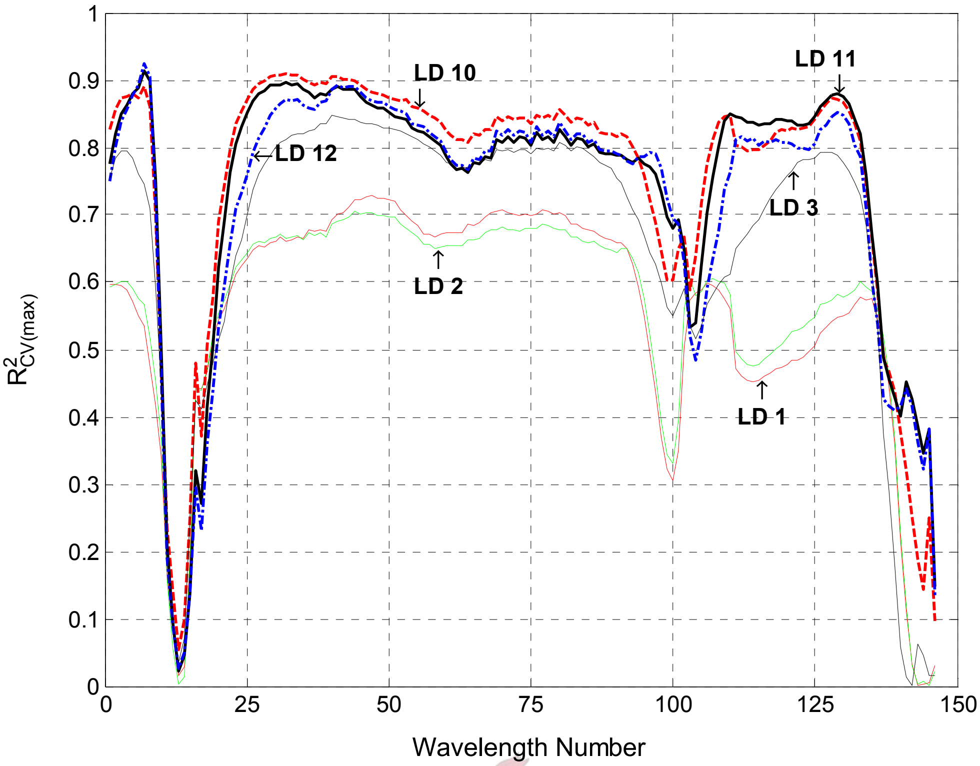

Fig. A-6 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their mean targets for the NNPLS models.

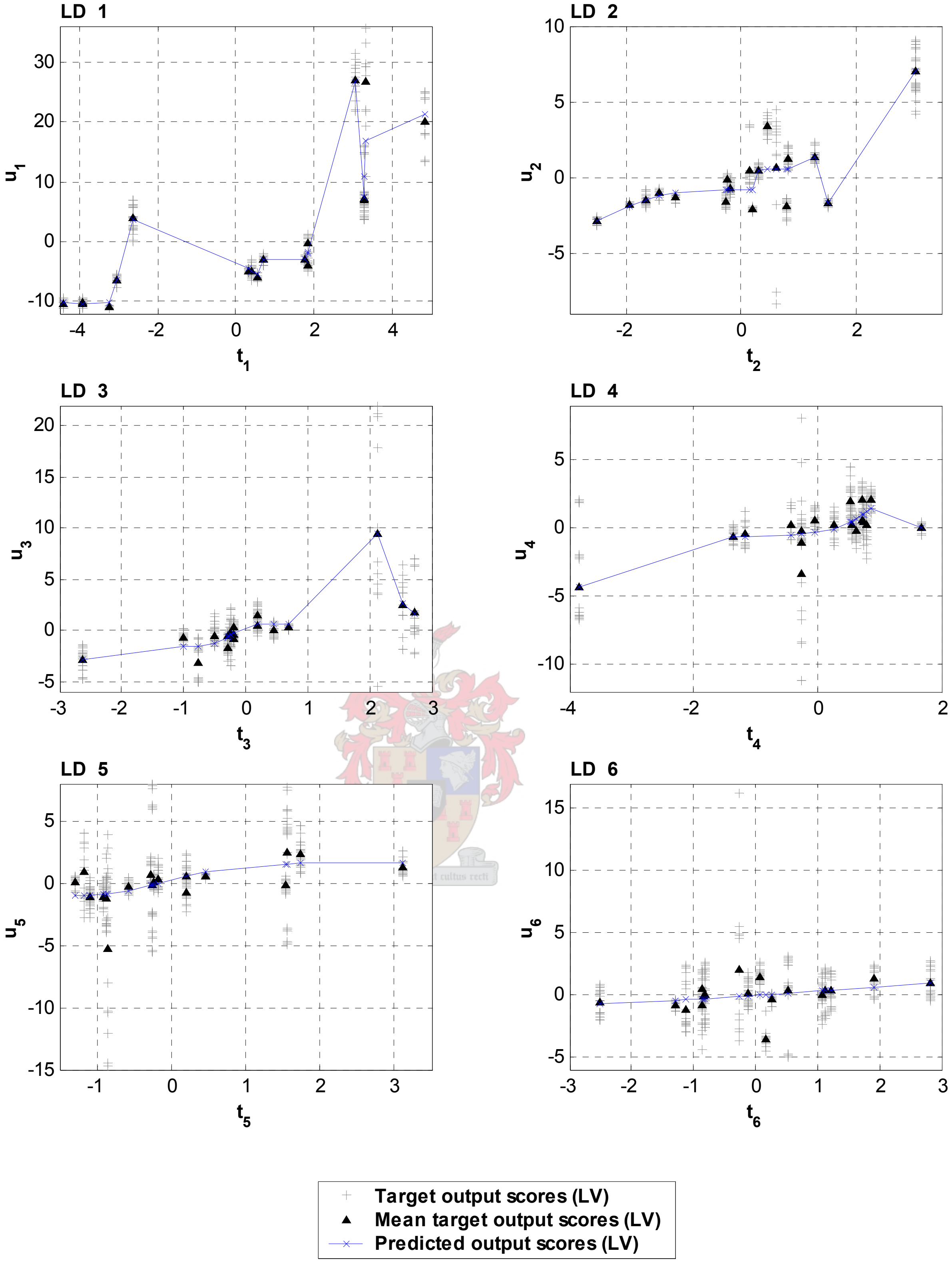

Fig. A-7 The target data and inner model relationships of the in- and output scores calculated for the first 6 latent dimensions of the overall NNPLS model.

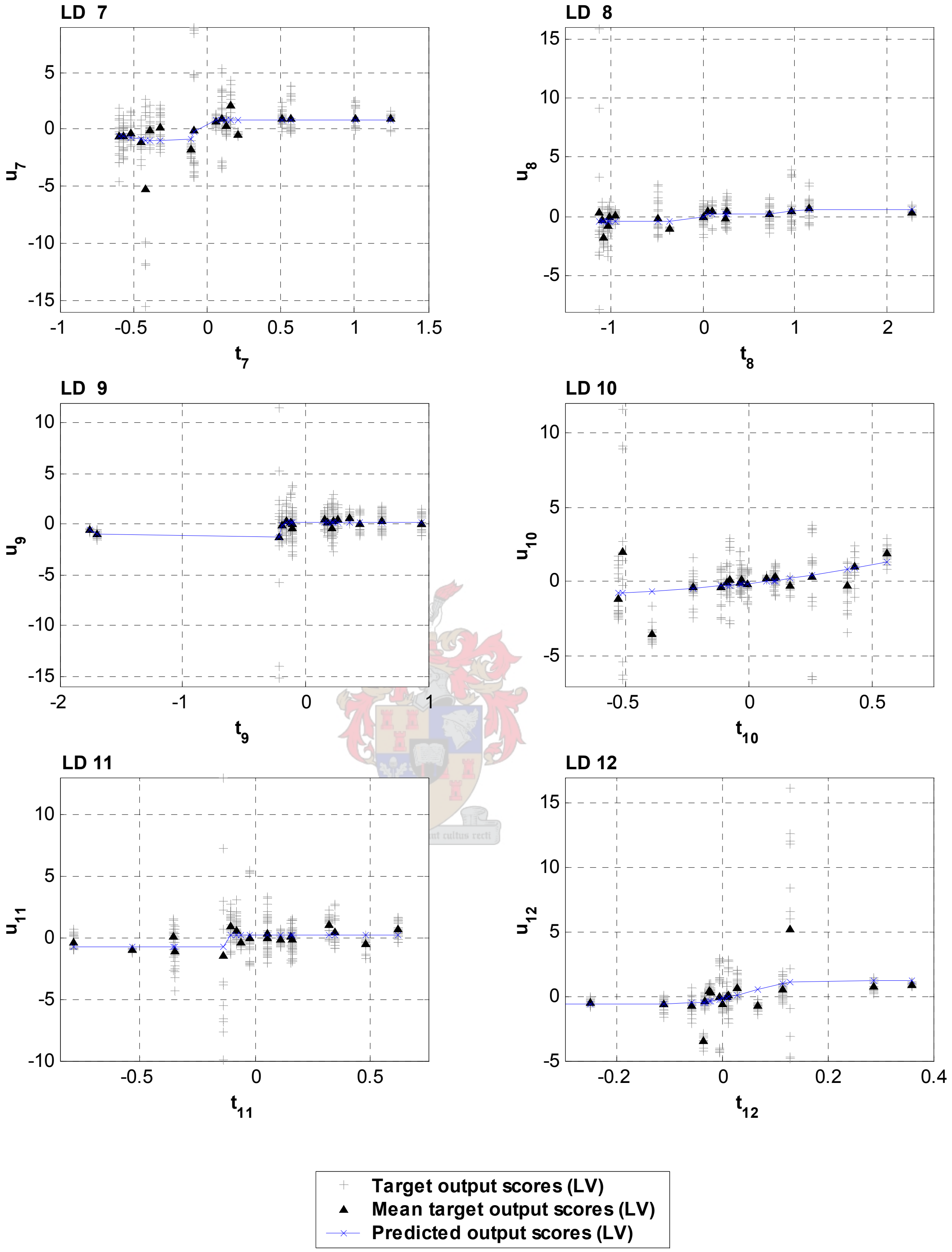

Fig. A-8 The target data and inner model relationships of the in- and output scores calculated for latent dimensions 7 to 12 of the overall NNPLS model.

A.4 Radial-basis Function PLS

Table A-11 The sum-square residuals for the RBFPLS models using different initialisation methods for the K-means method of training.

	Maximum spread initialisation			**PCA binary split initialisation**		
LD	**PRESS**	**SSEP**	**nodes per LD**	**PRESS**	**SSEP**	**nodes per LD**
1	834.03	563.33	2	834.03	563.33	2
2	810.98	344.20	1	844.23	243.77	5
3	723.99	207.06	1	837.94	246.49	1
4	755.61	203.25	1	841.34	240.24	1
5	817.71	120.47	1	771.28	236.39	2
6	876.67	95.87	1	632.97	230.88	4
7	761.91	89.22	3	575.94	191.08	2
8	668.55	78.16	1	579.51	143.58	4
9	620.08	74.02	2	605.31	143.16	1
10	601.27	72.53	1	571.27	132.22	2
11	563.76	71.68	1	586.47	129.00	2
12	566.05	68.32	1	581.82	126.83	2
13	572.44	66.94	1	573.01	125.97	1
14	574.49	64.05	1	549.75	119.56	2
15	577.21	57.15	2	570.83	114.77	1
16	646.36	50.31	1	580.72	111.72	1
17	653.41	50.17	2	566.98	111.19	1
18	623.22	49.32	2	596.95	110.87	1

Table A-12 The sum-square residuals for the RBFPLS models using differently calculated spread values for the OLS method of training.

	80 nearest neighbours			**120 nearest neighbours**		
LD	**PRESS**	**SSEP**	**nodes per LD**	**PRESS**	**SSEP**	**nodes per LD**
1	628.94	377.31	3	707.69	344.48	5
2	663.78	149.16	5	872.09	176.71	1
3	602.76	103.63	3	742.25	135.19	3
4	621.92	82.78	1	749.83	130.23	3
5	588.77	77.30	5	696.29	63.51	7
6	573.13	73.82	1	644.22	57.36	3
7	561.26	59.86	4	632.81	57.03	2
8	553.02	55.94	8	649.85	53.05	2
9	553.03	55.26	2	621.26	52.83	6
10	547.40	55.55	4	616.01	51.01	2
11	552.71	51.33	3	596.47	49.88	5
12	551.53	47.95	10	590.98	47.57	1
13	550.66	47.02	10	589.73	47.61	9
14	546.17	46.47	15	586.42	46.17	1
15	544.28	46.28	10	574.70	45.14	1
16	546.30	45.55	6	577.56	43.86	3
17	546.55	42.01	4	576.54	42.70	4
18	546.23	41.37	1	574.12	42.27	1

Table A-13 The SSE-values recorded during the inner model training of the overall RBFPLS model when using the ASOLS algorithm (see section 2.12.2).

```
Extracting DIM 1...
Iteration   1, sse= 10269.9, time(s)= 0.481
Iteration   2, sse= 10511.5, time(s)= 0.481
Iteration   3, sse= 7319.05, time(s)= 0.481
Iteration   4, sse= 7319.05, time(s)= 0.54
Iteration   1, sse= 10269.9, time(s)= 0.56
Iteration   2, sse= 10511.5, time(s)= 0.56
Iteration   3, sse= 7319.05, time(s)= 0.561
Iteration   4, sse= 7319.05, time(s)= 0.561
Extracting DIM 2...
Iteration   1, sse= 1756.78, time(s)= 0.19
Iteration   2, sse= 1756.78, time(s)= 0.19
Iteration   1, sse= 1756.78, time(s)= 0.191
Iteration   2, sse= 1756.78, time(s)= 0.19

Extracting DIM 4...
Iteration   1, sse= 2297.68, time(s)= 0.651
Iteration   2, sse= 2279.38, time(s)= 0.651
Iteration   3, sse= 2236.66, time(s)= 0.651
Iteration   4, sse= 2239.05, time(s)= 0.641
Iteration   5, sse= 2239.05, time(s)= 0.651
Iteration   1, sse= 2297.68, time(s)= 0.641
Iteration   2, sse= 2279.38, time(s)= 0.651
Iteration   3, sse= 2236.66, time(s)= 0.661
Iteration   4, sse= 2239.05, time(s)= 0.661
Iteration   5, sse= 2239.05, time(s)= 0.661

Extracting DIM 6...
Iteration   1, sse= 845.944, time(s)= 1.011
Iteration   2, sse= 827.692, time(s)= 1.022
Iteration   3, sse= 790.241, time(s)= 1.011
Iteration   4, sse= 783.577, time(s)= 1.022
Iteration   5, sse= 783.577, time(s)= 1.021
Iteration   6, sse= 783.577, time(s)= 1.012
Iteration   1, sse= 845.944, time(s)= 1.021
Iteration   2, sse= 827.692, time(s)= 1.022
Iteration   3, sse= 790.241, time(s)= 1.012
Iteration   4, sse= 783.577, time(s)= 1.011
Iteration   5, sse= 783.577, time(s)= 1.022
Iteration   6, sse= 783.577, time(s)= 1.021
```

Table A-14 Leave-one-out cross-validation for 5 runs with 80, 120, 160, 110 and 130 nearest neighbours respectively to initialise the ASOLS training algorithm in each inner model.

LD	PRESS Run 1	PRESS Run 2*	PRESS Run 3	PRESS Run 4	PRESS Run 5	Average
1	709.87	716.95	725.42	612.46	738.78	700.70
2	737.69	691.65	749.60	586.57	813.12	715.73
3	610.37	646.31	703.86	566.79	804.55	666.38
4	566.27	442.60	641.43	577.87	797.56	605.15
5	552.77	442.25	653.97	537.66	775.59	592.45
6	529.40	381.20	583.88	528.09	749.65	554.44
7	518.39	369.13	584.57	521.85	744.83	547.75
8	557.37	381.97	582.13	516.69	747.68	557.17
9	570.18	377.55	573.76	506.25	759.74	557.50
10	574.22	367.56	573.66	506.09	712.81	546.87
11	572.40	368.75	563.28	491.75	692.10	537.66
12	576.13	369.74	564.70	498.98	694.57	540.82
13	570.41	368.64	555.38	495.04	687.39	535.37
14	572.42	376.28	547.23	502.88	686.96	537.15
15	568.85	376.37	548.35	496.64	679.99	534.04
16	573.22	377.30	430.32	432.19	634.03	489.41
17	568.53	372.19	427.94	428.89	632.13	485.94
18	567.61	371.08	427.74	422.73	630.25	483.88

*** The best run chosen for further evaluation**

Table A-15 The overall model training results for each of the 5 runs using the RBFPLS algorithm with ASOLS training of the inner models.

LD	SSEP Run 1	SSEP Run 2*	SSEP Run 3	SSEP Run 4	SSEP Run 5	SSEP Average
1	426.19	250.85	476.51	175.55	392.00	344.22
2	141.77	149.19	195.84	75.56	171.77	146.82
3	80.74	132.91	181.89	68.88	125.35	117.96
4	56.34	129.30	68.66	61.39	108.19	84.78
5	50.56	84.54	65.37	54.85	94.57	69.98
6	46.97	77.22	52.36	53.68	64.79	59.00
7	47.47	53.82	51.45	55.53	55.28	52.71
8	46.57	54.04	51.37	54.11	55.00	52.22
9	45.05	53.46	50.12	53.28	54.30	51.24
10	44.90	53.06	49.96	47.15	49.86	48.99
11	44.78	50.71	49.73	46.10	49.13	48.09
12	44.33	46.92	46.93	47.52	48.87	46.91
13	44.16	47.41	46.66	47.47	48.39	46.82
14	42.20	44.10	46.02	44.86	47.72	44.98
15	41.89	43.45	45.58	44.16	44.13	43.84
16	41.47	41.35	45.02	43.92	44.21	43.19
17	40.88	41.06	43.38	43.50	41.93	42.15
18	40.53	40.77	42.39	41.44	40.97	41.22

*** The best run chosen for further evaluation**

Table A-16 The number of hidden nodes for each inner model (LD) obtained for each modified cross-validation run using RBFPLS with ASOLS.

LD	Run 1	Run 2	Run 3	Run 4	Run 5	Rounded Average
1	3	5	1	8	3	4
2	3	1	4	3	7	4
3	6	2	1	2	8	4
4	7	6	5	3	1	4
5	3	3	5	1	2	3
6	6	10	10	2	2	6
7	3	6	2	2	4	3
8	6	4	5	1	1	3
9	8	1	2	1	1	3
10	1	2	1	1	4	2
11	1	2	4	4	1	2
12	1	4	4	1	5	3
13	1	7	1	9	4	4
14	1	1	1	1	5	2
15	6	4	6	6	6	6
16	6	6	2	3	3	4
17	2	1	7	1	2	3
18	6	5	3	3	4	4

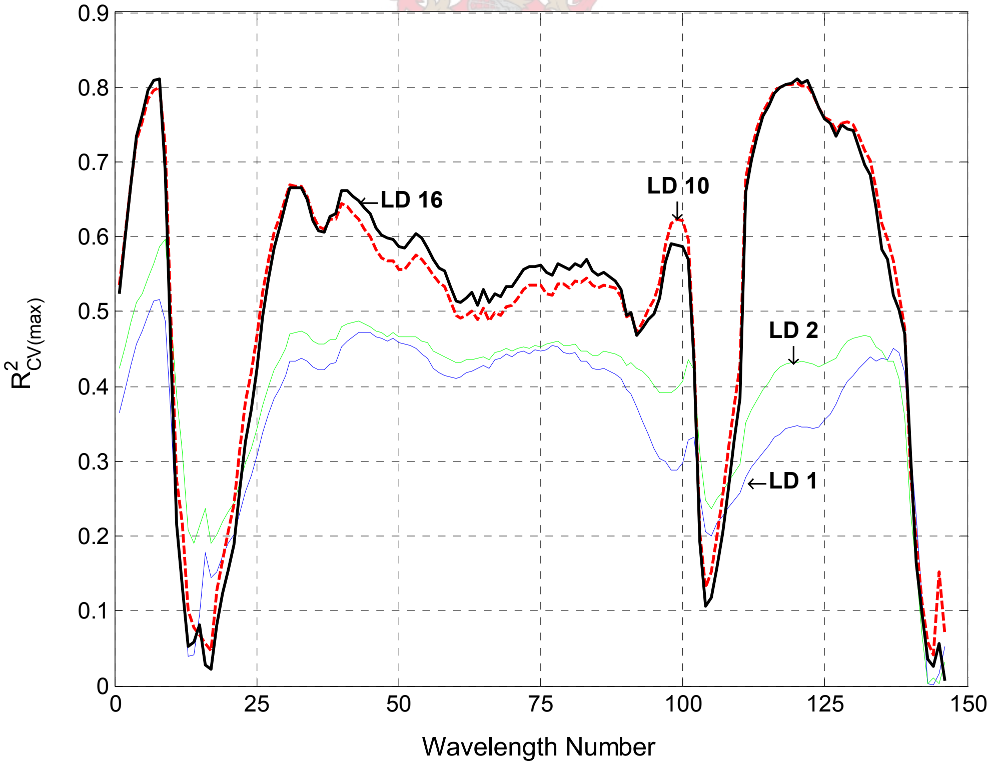

Fig. A-9 The squared correlations between the 146 predicted cross-validation irradiance absorbance values and their mean targets for the RBFPLS models using ASOLS (run 2).

Table A-17 Summary of results using RBFPLS with ASOLS. Results are evaluated on the rescaled data from the best of the repeated runs.

LD	Cumulative % η^2 X-block (inputs)	Cumulative % η^2 Y-block (outputs)	Cumulative % η^2 Ouputs (max)	Average pdf	Average R^2_{cv}	Average $R^2_{cv(max)}$	Average R^2	Average R^2_{max}	PRESS	SSEP	$PRESS_{min}$	$SSEP_{min}$
1	48.44	68.75	75.84	7.69	0.248	0.360	0.675	0.769	716.95	250.85	27.246	9.343
2	57.75	81.41	82.49	8.00	0.279	0.400	0.705	0.784	691.65	149.19	24.991	6.771
3	77.20	83.44	85.93	9.72	0.284	0.419	0.752	0.844	646.31	132.91	22.734	5.442
4	79.13	83.89	86.31	10.00	0.355	0.477	0.772	0.874	442.60	129.30	16.590	5.296
5	80.83	89.47	94.14	11.04	0.383	0.504	0.790	0.901	442.25	84.54	16.477	2.265
6	91.85	90.38	94.99	11.43	0.414	0.535	0.800	0.919	381.20	77.22	13.101	1.937
7	93.79	93.29	98.00	12.46	0.415	0.525	0.811	0.934	369.13	53.82	12.418	0.773
8	98.06	93.27	98.00	13.09	0.412	0.520	0.816	0.942	381.97	54.04	12.890	0.773
9	98.42	93.34	98.05	13.14	0.413	0.522	0.817	0.944	377.55	53.46	12.731	0.755
10	98.93	93.39	98.09	13.21	0.417	0.527	0.818	0.946	367.56	53.06	12.453	0.737
11	99.60	93.68	98.33	13.35	0.421	0.530	0.820	0.948	368.75	50.71	12.409	0.644
12	99.91	94.15	98.82	13.79	0.424	0.530	0.825	0.954	369.74	46.92	12.505	0.456
13	100.00	94.09	98.61	14.18	0.424	0.531	0.828	0.962	368.64	47.41	12.459	0.537
14	100.00	94.51	99.33	14.91	0.420	0.524	0.833	0.977	376.28	44.10	12.722	0.257
15	100.00	94.59	99.42	15.21	0.419	0.522	0.835	0.980	376.37	43.45	12.798	0.224
16	100.00	94.85	99.72	15.64	0.419	0.522	0.837	0.984	377.30	41.35	12.823	0.108
17	100.00	94.88	99.76	16.13	0.421	0.523	0.840	0.992	372.19	41.06	12.628	0.091
18	100.00	94.92	99.79	16.32	0.421	0.524	0.841	0.994	371.08	40.77	12.581	0.080

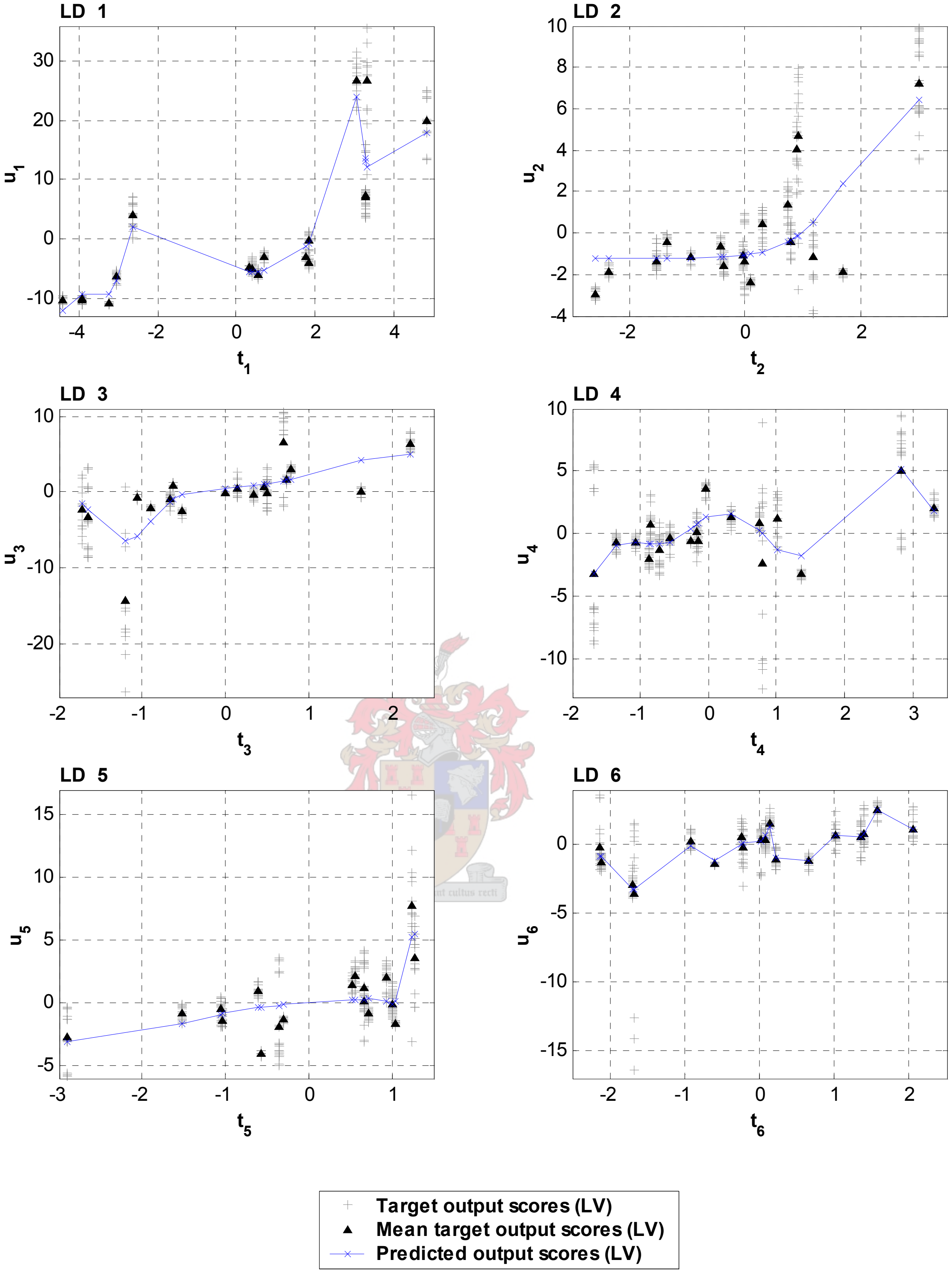

Fig. A-10 The target data and inner model relationships of the in- and output scores calculated for the first 6 latent dimensions of the overall RBFPLS model.

A.5 Sensitivity Analysis

Table A-18 The perturbations made to the input variables (indicated in bold italic font) to test the qualitative behaviour of the overall NNPLS model.

	C10			DB1		C9			C1	
C	1.115	1.115	1.115	1.869	1.869	0.894	0.894	0.894	0.997	0.997
H	4.399	4.399	4.399	2.658	2.658	4.203	4.203	4.203	3.908	3.908
O	2.769	2.769	2.769	3.362	3.362	2.855	2.855	2.855	2.407	2.407
N	0.695	0.695	0.695	1.335	1.335	0.708	0.708	0.708	0.604	0.604
Al	0.148	0.148	0.148	0	0	0.148	0.148	0.148	0.593	***0***
K	0	0	***0.05***	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
Cu	0	0	0	0.007	0.007	0	0	0	0	0
Pb	0	0	0	0.008	0.008	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0
Cl	0.681	0.681	0.681	0	0	0.702	0.702	0.702	0.596	0.596
Si	0	0	0	0	0	0	0	0	0	0
Ti	0	0	0	0.008	0.008	0	0	0	0	0
Fe	0	0	0	0	0	0.009	0.009	0.009	0	0
T_c	2.81	2.81	2.81	2.92	2.92	2.98	2.98	***3.15***	3.27	3.27
P_c	1.29	***3***	1.29	4.9	4.9	1.06	1.06	1.06	2.7	2.7
E_c	4.5	4.5	4.5	6.2	***3***	1.7	1.7	1.7	6.2	6.2
D_T	15	15	15	13	13	25	***15***	25	13	13

Table A-19 The global modifications made to the rocket propellant, C11 to investigate the effect on the spectral irradiance.

	C11	C11-A	C11-B	C11-C	C11-D
AP	81%	77%	83%	81%	81%
Al	4%	8%	2%	4%	4%
DOA	2%	2%	2%	2%	2%
Isophoron	1.32%	1.32%	1.32%	1.32%	-
Polyester	-	-	-	-	13%
CTPB	-	-	-	12%	-
HTPB	12%	12%	12%	-	-
T_C (K)	2860	2980	2820	2870	3050
P_C (MPa)	1	1.23	1.23	1.23	1.23

Appendix B

Processed Results of Reverse Modelling

B.1 Linear PLS

Table B-1 Summary of results obtained from the reverse linear PLS model evaluated on the rescaled data.

	Cumulative % η^2							
LD	X-block (inputs)	Y-block (outputs)	Average pdf	Parameters Per Output Variable Cumulated	Average R^2_{cv}	Average R^2	PRESS	SSEP
1	92.44	13.57	1.87	148	0.173	0.217	466.1	362.83
2	95.25	26.85	3.75	296	0.183	0.351	475.6	307.08
3	98.63	32.61	4.91	444	0.172	0.390	551.1	282.89
4	99.60	43.81	7.45	592	0.144	0.442	664.3	235.89
5	99.85	60.05	8.66	740	0.168	0.558	632.3	167.72
6	99.91	67.20	10.30	888	0.221	0.662	622.3	137.69
7	99.93	69.50	11.16	1036	0.214	0.708	630.9	128.03
8	99.95	73.51	12.79	1184	0.192	0.771	849.6	111.18
9	99.96	75.05	13.29	1332	0.203	0.806	964.3	104.75
10	99.97	77.21	13.88	1480	0.239	0.845	1867.7	95.69
11	99.98	84.17	14.81	1628	0.196	0.867	1893.4	66.46
12	99.99	88.47	15.35	1776	0.205	0.912	1971.9	48.42
13	100.0	89.24	15.66	1924	0.213	0.939	2411	45.19
14	100.0	91.59	16.18	2072	0.215	0.955	3053.6	35.29
15	100.0	94.53	16.57	2220	0.217	0.971	2914.7	22.95
16	100.0	95.74	17.18	2368	0.212	0.982	5408.9	17.88
17	100.0	100.00	18.00	2516	0.266	1.000	3.59E+21	0.00
18	100.0	100.00	18.00	2664	0.265	1.000	1.02E+22	0.00
19	100.0	100.00	18.00	2812	0.264	1.000	1.31E+22	0.00
20	100.0	100.00	18.00	2960	0.270	1.000	1.29E+22	0.00

Table B-2 The summary of results for each output variable evaluated at its optimum model complexity using linear PLS.

	$PRESS_j$	$SSEP_j$	RMSECV	RMSE	LD	Parameters Per Output Variable	pdf	R^2_{cv}	R^2
C	4.058	2.132	0.475	0.404	2	296	4.95	0.163	0.502
H	9.265	7.714	0.717	0.685	1	148	1.58	0.001	0.071
O	0.856	0.045	0.218	0.104	13	1924	13.88	0.737	0.979
N	1.097	0.599	0.247	0.212	2	296	4.69	0.228	0.538
Al	0.246	0.014	0.117	0.057	12	1776	13.67	0.769	0.986
K	0.0142	0.006	0.028	0.022	7	1036	6.57	0.301	0.666
F	0.0043	0.003	0.016	0.015	1	148	1.92	0.037	0.053
Cu	0.0003	0.000	0.004	0.003	2	296	3.75	0.114	0.392
Pb	0.0002	0.000	0.003	0.003	1	148	1.66	0.099	0.223
S	0.0003	0.000	0.004	0.004	1	148	1.46	0.826	0.000
Cl	1.891	1.566	0.324	0.309	1	148	1.62	0.033	0.147
Si	0.0021	0.002	0.011	0.010	1	148	1.35	0.000	0.067
Ti	0.0002	0.000	0.003	0.002	7	1036	8.13	0.181	0.691
Fe	0.0008	0.000	0.007	0.004	4	592	10.28	0.057	0.781
T_C	1.1827	0.805	0.256	0.233	2	296	3.15	0.456	0.604
P_C	57.35	46.600	1.785	1.695	1	148	1.77	0.008	0.062
E_C	137.29	33.384	2.762	1.939	6	888	9.12	0.390	0.764
D_T	238.46	186.118	3.640	3.421	1	148	2.10	0.002	0.103
Σ	451.7	279.0		***Ave***	3.61	534.44	5.09	0.244	0.424

Table B-3 Explained variances calculated on the standardised data with increasing latent dimension (LD) on the overall linear PLS model.

LD	X-Block (input)		Y-block (outputs)	
	% η^2	Cumulative % η^2	% η^2	Cumulative % η^2
1	88.56	88.56	21.68	21.68
2	2.38	90.94	13.45	35.13
3	3.91	94.85	3.82	38.95
4	2.15	97.00	5.29	44.24
5	1.46	98.46	11.55	55.79
6	1.07	99.53	10.40	66.19
7	0.130	99.66	4.60	70.79
8	0.072	99.73	6.30	77.09
9	0.118	99.85	3.45	80.55
10	0.070	99.92	3.90	84.45
11	0.031	99.95	2.28	86.73
12	0.012	99.96	4.45	91.18
13	0.011	99.97	2.70	93.88
14	0.013	99.99	1.66	95.54
15	0.008	99.99	1.58	97.13
16	0.005	100.00	1.06	98.19
17	0.002	100.00	1.81	100.00
18	0.000	100.00	0.00	100.00
19	0.000	100.00	0.00	100.00
20	0.000	100.00	0.00	100.00

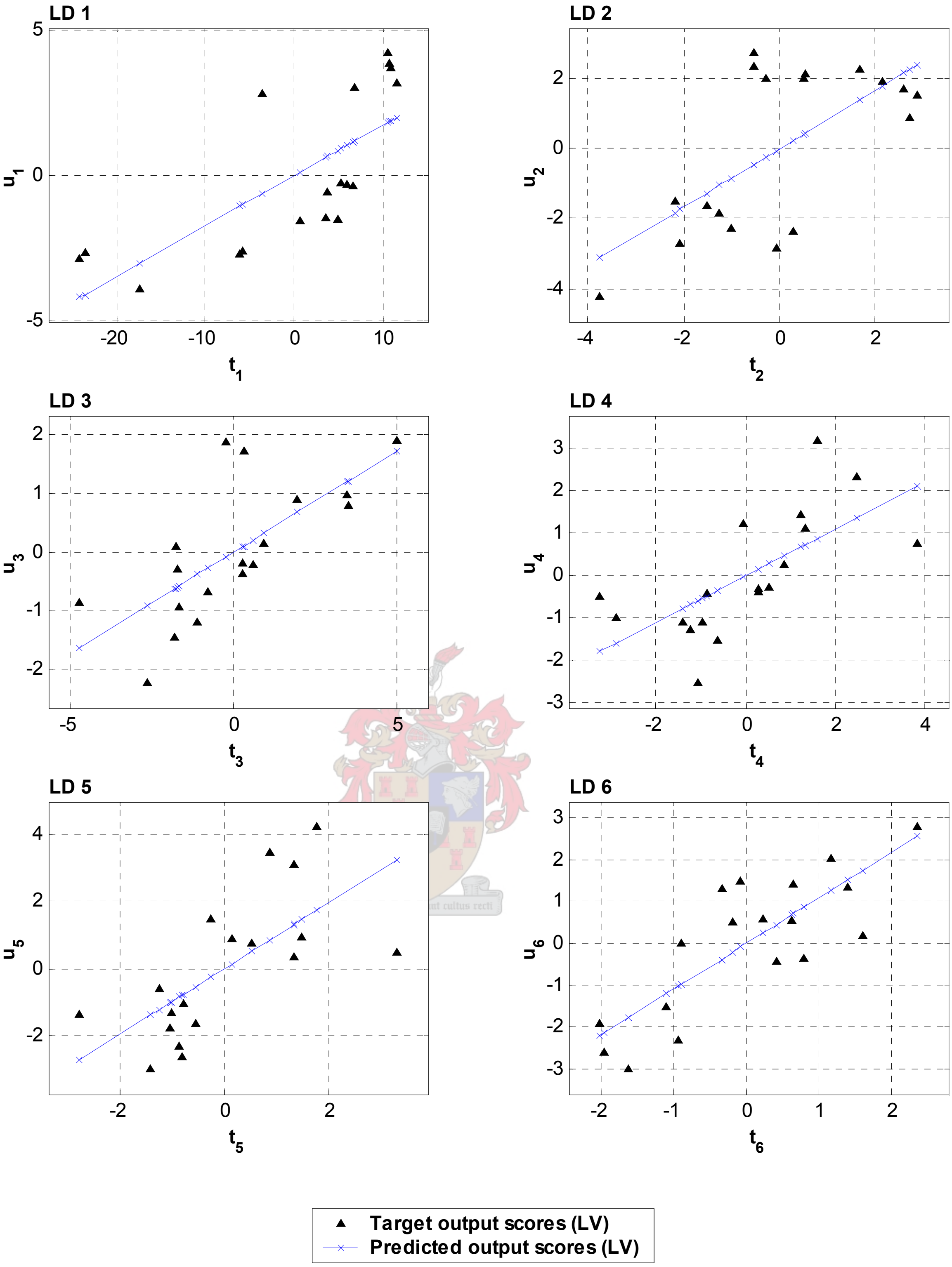

Fig. B-1 The target data and predicted inner model relationships of the in- and output scores calculated for the first 6 latent dimensions using the overall linear PLS model.

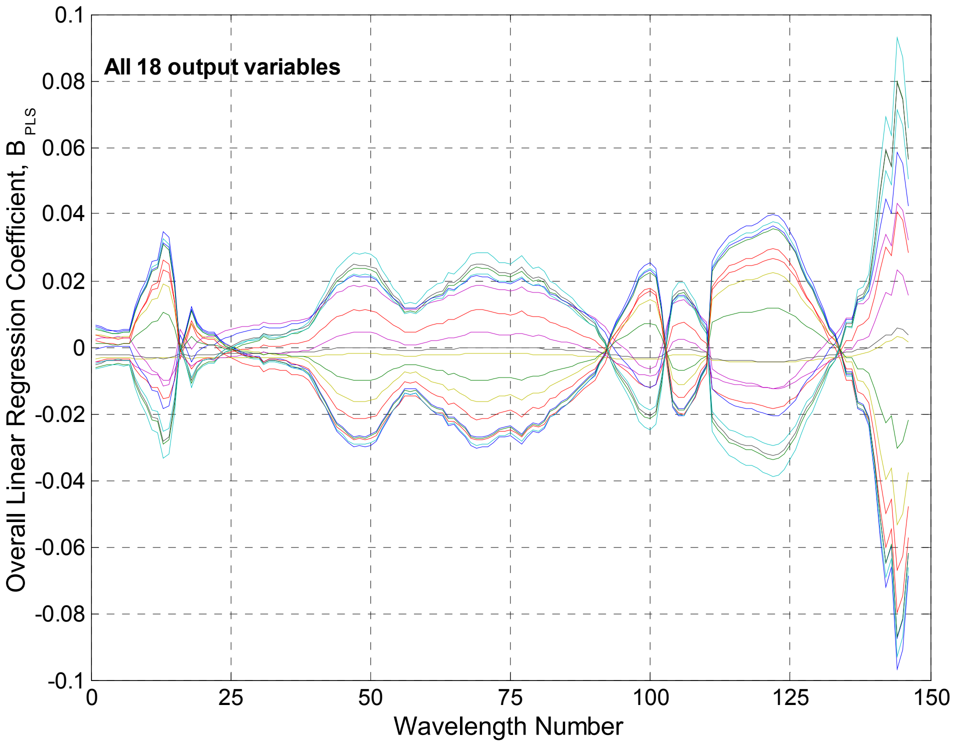

Fig. B-2 A plot of the linear regression coefficients of the linear PLS model.

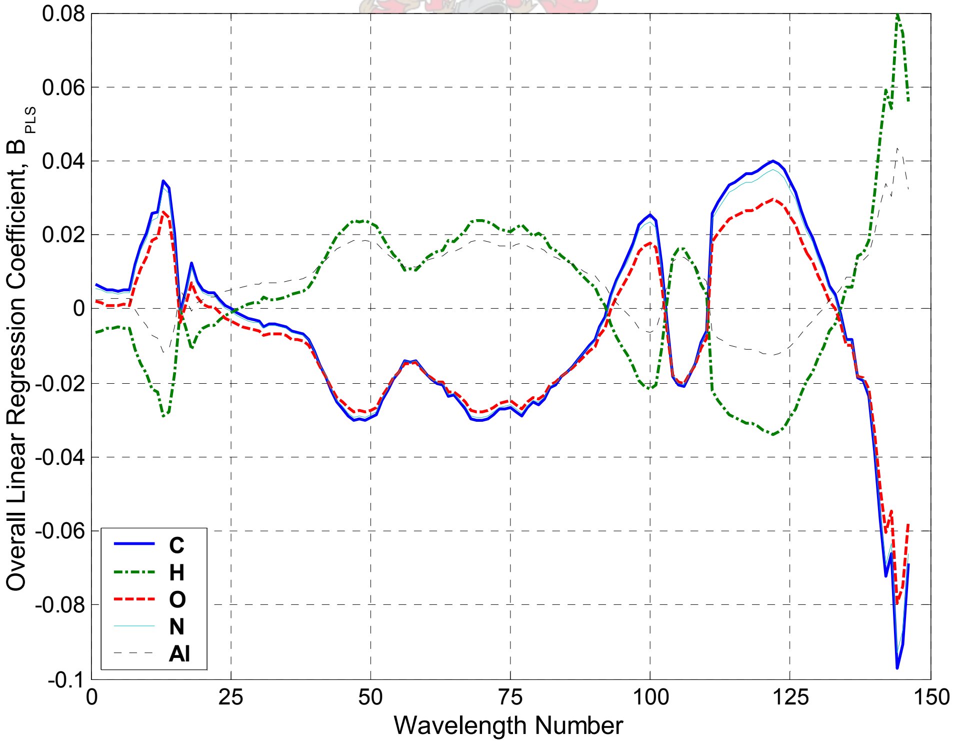

Fig. B-3 A plot of the linear regression coefficients of the reverse linear PLS model for the selected output variables.

B.2 Feed-forward Neural Network

Table B-4 Leave-one-out cross-validation was repeated 10 times for each single hidden layer neural network.

No. of hidden nodes	PRESS Run 1*	PRESS Run 2	PRESS Run 3	PRESS Run 4	PRESS Run 5	PRESS Run 6	PRESS Run 7	PRESS Run 8	PRESS Run 9	PRESS Run 10	PRESS Average
1	363.30	351.40	360.40	350.70	351.40	422.40	408.40	333.30	351.80	334.40	362.75
2	329.10	446.80	386.20	376.20	360.90	329.40	412.50	361.60	384.50	402.50	378.97
4	809.2	1207.30	1260.40	481.90	814.60	1164.70	857.50	1005.90	671.70	1528.20	980.14
6	1585.7	2376.40	2958.30	2797.60	1234.60	1268.20	759.40	1200.70	1329.10	1075.60	1658.56
8	1433.3	4923.10	762.00	1447.40	1444.00	2193.40	3435.10	1022.10	1411.90	1847.30	1991.96
10	720.60	1204.20	927.60	1508.70	1355.10	1478.70	1647.40	1429.00	993.30	1809.50	1307.41
15	1238.60	2215.50	1110.00	1613.00	1007.20	1625.90	1795.20	1574.00	1206.80	1828.40	1521.46
20	1863.30	1361.80	1062.90	1453.10	1493.10	1928.80	1501.30	1135.00	1316.50	1182.00	1429.78
30	1497.70	2670.50	1642.00	1783.10	1509.90	3012.70	1688.50	1260.10	1903.40	1668.40	1863.63
40	1590.30	1736.70	2550.60	1437.10	1459.00	1612.50	1479.50	2087.70	2033.70	1486.90	1747.40

Table B-5 The overall model training was repeated 5 times for each single hidden layer neural network.

No. of hidden nodes	SSEP Run 1	SSEP Run 2	SSEP Run 3*	SSEP Run 4	SSEP Run 5	SSEP Average
1	249.76	249.86	249.82	249.69	249.90	249.81
2	246.65	247.23	245.61	249.07	245.80	246.87
4	150.27	143.59	117.63	158.47	101.05	134.20
6	48.98	108.78	63.26	53.72	87.31	72.41
8	34.95	38.05	32.43	32.56	35.92	34.78
10	8.10	6.85	5.21	12.05	15.91	9.62
15	0.34	0.12	0.64	0.27	0.17	0.31
20	0.02	0.01	0.05	0.00	0.00	0.02
30	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00

*** The best run chosen for further evaluation**

Table B-6 Summary of results obtained from the reverse feed-forward neural network model evaluated on the rescaled data.

No. of hidden nodes	Y-block Cumulative % η^2	Average pdf	Parameters Per Output Variable	Parameters for All Output Variables	Average R^2_{cv}	Average R^2	PRESS	SSEP
1	40.49	5.40	149	183	0.224	0.501	363.3	249.82
2	41.49	7.07	297	348	0.351	0.615	329.1	245.61
4	71.98	12.49	593	678	0.265	0.786	809.2	117.63
6	84.93	14.49	889	1008	0.206	0.876	1585.7	63.26
8	92.27	15.96	1185	1338	0.164	0.961	1433.3	32.43
10	98.76	17.02	1481	1668	0.161	0.992	720.6	5.21
15	99.85	17.72	2221	2493	0.211	1.000	1238.6	0.64
20	100.0	17.94	2961	3318	0.166	1.000	1863.3	0.05
30	100.0	18.00	4441	4968	0.246	1.000	1497.7	0.00
40	100.0	18.00	5921	6618	0.217	1.000	1590.3	0.00

Table B-7 The summary of results for each output variable evaluated at its optimum model complexity using feed-forward neural networks.

	PRESS_j	SSEP_j	RMSECV	RMSE	No. of Hidden Nodes	Parameters Per Output Variable	pdf	R^2_{cv}	R^2
C	2.605	0.773	0.380	0.281	2	297	8.19	0.467	0.819
H	5.192	0.353	0.537	0.274	2	297	13.30	0.539	0.958
O	0.836	0.054	0.216	0.109	6	889	13.41	0.716	0.975
N	0.506	0.152	0.168	0.124	2	297	8.14	0.641	0.883
Al	0.582	0.030	0.180	0.086	6	889	13.92	0.694	0.970
K	0.012	0.011	0.025	0.025	1	149	0.24	0.440	0.345
F	0.004	0.003	0.016	0.014	1	149	2.99	0.005	0.172
Cu	0.000	0.000	0.003	0.002	2	297	4.26	0.576	0.738
Pb	0.000	0.000	0.003	0.002	2	297	12.06	0.514	0.933
S	0.000	0.000	0.004	0.004	1	149	2.00	0.065	0.069
Cl	0.984	0.065	0.234	0.119	2	297	13.37	0.571	0.965
Si	0.002	0.000	0.011	0.007	2	297	10.59	0.219	0.817
Ti	0.000	0.000	0.003	0.002	1	149	4.36	0.225	0.528
Fe	0.001	0.001	0.006	0.006	1	149	1.82	0.033	0.075
T_C	1.360	1.003	0.275	0.255	2	297	2.54	0.344	0.507
P_C	39.94	29.06	1.490	1.376	1	149	2.65	0.232	0.415
E_C	93.72	69.82	2.282	2.120	2	297	2.46	0.374	0.507
D_T	182.11	144.37	3.181	3.001	2	297	1.97	0.168	0.305
Σ	327.8	245.7		***Ave***	2.11	313	6.57	0.379	0.610

B.3 Neural Network PLS

Table B-8 Leave-one-out cross-validation was repeated 10 times for each NNPLS latent dimension.

LD	PRESS Run 1	PRESS Run 2	PRESS Run 3	PRESS Run 4	PRESS Run 5	PRESS Run 6	PRESS Run 7	PRESS Run 8*	PRESS Run 9	PRESS Run 10	PRESS Average
1	347.7	347.7	347.7	347.7	347.7	347.7	373.5	347.7	347.7	347.7	350.3
2	327.7	338.5	330.6	338.8	339.2	349.2	374.1	346.5	334.4	334.6	341.4
3	290.8	319.0	287.3	278.5	330.5	304.5	361.1	273.8	295.6	299.3	304.0
4	311.5	346.1	323.7	320.1	375.8	317.6	386.9	310.5	319.8	339.3	335.1
5	637.3	611.7	865.0	395.4	483.0	816.0	779.9	583.2	437.8	404.0	601.3
6	703.1	667.7	906.9	446.5	555.8	844.4	799.7	651.7	482.9	447.6	650.6
7	759.3	658.0	1107.1	769.9	643.9	1103.5	962.9	1583.0	572.9	1332.8	949.3
8	764.0	685.3	1010.1	719.2	784.7	980.2	1031.3	1644.5	535.4	1424.8	958.0
9	978.2	748.1	1133.1	808.4	781.6	1078.8	1087.2	1760.6	1058.7	1715.7	1115.0
10	1091.6	810.5	1262.8	861.5	758.2	1211.5	1269.8	1829.4	1115.5	1713.9	1192.5
11	1115.5	767.9	1183.2	791.3	1284.7	1099.1	1198.3	1847.6	1068.3	1639.5	1199.5
12	1104.7	781.4	1217.7	915.1	1329.8	1102.8	1217.5	1832.4	1036.3	1685.3	1222.3
13	1059.3	717.2	1121.4	842.7	1360.0	1010.9	1108.7	1591.5	1088.3	1573.7	1147.4
14	1055.7	691.0	1072.2	767.9	1344.1	1004.3	1041.0	1581.6	1086.1	1480.3	1112.4
15	1026.5	725.2	1305.5	741.5	1353.5	967.7	1009.0	1869.4	1127.9	1543.8	1167.0
16	986.7	658.4	1212.0	687.9	1320.3	953.0	660.1	1872.9	1059.3	1517.0	1092.8
17	4004.6	2673.4	1250.0	1984.6	1751.8	1948.3	760.7	1984.9	1395.9	4908.7	2266.3
18	3942.5	2630.3	1693.0	2055.6	1754.3	1933.3	714.8	1987.0	1408.1	6545.6	2466.5
19	5064.2	2625.8	1684.0	2972.3	4245.1	2157.3	1710.5	2036.3	1387.0	6797.7	3068.0
20	5194.2	2956.0	1698.1	3015.6	2493.7	4016.6	2274.2	2299.9	4683.5	6877.7	3551.0

*** The best run chosen for further evaluation**

Table B-9 The number of hidden nodes for each latent dimension inner model obtained for each modified cross-validation repeat run using NNPLS.

LD	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8*	Run 9	Run 10	Rounded Average
1	1	1	1	1	1	1	2	1	1	1	1
2	5	3	5	4	5	1	1	3	4	3	3
3	3	3	3	2	1	2	3	2	3	3	3
4	1	1	1	1	1	1	1	1	1	1	1
5	4	5	1	2	11	1	1	12	6	2	5
6	2	3	2	2	1	1	3	1	2	2	2
7	2	1	2	1	2	2	1	4	1	2	2
8	2	3	1	1	1	2	4	4	2	1	2
9	1	2	1	1	10	1	8	2	13	1	4
10	1	1	1	1	2	1	2	1	1	2	1
11	4	1	1	1	5	1	1	5	3	1	2
12	2	2	4	1	1	2	4	6	1	4	3
13	8	1	1	11	11	1	1	7	1	1	4
14	8	12	1	1	10	8	9	12	1	3	7
15	4	7	3	1	2	1	1	1	1	4	3
16	10	3	1	1	11	2	1	1	1	6	4
17	1	9	14	1	8	8	7	5	1	5	6
18	7	9	5	13	8	1	9	10	12	1	8
19	6	10	9	9	5	3	4	1	12	6	7
20	14	6	11	11	7	4	4	11	5	6	8

Table B-10 The overall model training was repeated 5 times for each NNPLS latent dimension.

LD	SSEP Run 1	SSEP Run 2	SSEP Run 3*	SSEP Run 4	SSEP Run 5	SSEP Average
1	265.8	265.8	265.8	265.8	265.8	265.8
2	239.2	245.7	239.3	253.2	252.8	246.0
3	239.2	176.3	171.2	175.6	186.5	189.8
4	201.2	173.4	168.1	170.7	180.7	178.8
5	122.9	152.5	150.0	150.6	149.1	145.0
6	108.2	114.4	123.1	127.1	122.7	119.1
7	107.9	115.7	118.4	119.2	118.8	116.0
8	82.6	92.1	99.7	94.1	89.8	91.7
9	61.7	78.6	63.5	66.8	72.3	68.6
10	40.2	58.6	45.8	46.0	53.3	48.8
11	31.3	26.3	21.5	19.2	23.6	24.4
12	27.0	22.2	20.1	17.2	21.5	21.6
13	23.2	21.0	16.1	14.1	17.1	18.3
14	23.3	13.0	14.3	12.2	14.4	15.4
15	19.1	10.8	11.8	10.2	12.5	12.9
16	15.7	8.7	10.2	8.0	10.5	10.6
17	14.8	6.9	6.4	6.6	7.0	8.3
18	11.1	6.7	4.5	5.6	6.2	6.8
19	8.8	5.4	2.9	4.5	5.6	5.4
20	4.6	3.4	2.1	3.6	5.4	3.8

*** The best run chosen for further evaluation**

Table B-11 Summary of results obtained from the reverse NNPLS model evaluated on the rescaled data.

	Cumulative % η^2								
LD	X-block (inputs)	Y-block (outputs)	Average pdf	Number of Inner Model Paramters	Parameters Per Output Variable	Average R^2_{cv}	Average R^2	PRESS	SSEP
1	97.82	36.68	2.35	4	151	0.166	0.287	347.7	265.79
2	99.21	43.00	4.89	14	308	0.227	0.521	346.5	239.27
3	99.84	59.22	6.19	21	462	0.322	0.615	273.8	171.18
4	99.93	59.94	7.34	25	613	0.322	0.659	310.5	168.15
5	99.98	64.28	8.66	62	797	0.316	0.690	583.2	149.97
6	99.99	70.68	9.75	66	948	0.298	0.722	651.7	123.08
7	100.00	71.81	11.44	79	1108	0.229	0.756	1583.0	118.35
8	100.00	76.24	12.11	92	1268	0.236	0.794	1644.5	99.75
9	100.00	84.87	13.12	99	1422	0.220	0.837	1760.6	63.49
10	100.00	89.10	13.74	103	1573	0.221	0.878	1829.4	45.78
11	100.00	94.87	14.38	119	1736	0.207	0.894	1847.6	21.55
12	100.00	95.22	14.78	138	1902	0.201	0.920	1832.4	20.08
13	100.00	96.18	15.50	160	2071	0.176	0.956	1591.5	16.05
14	100.00	96.59	16.07	197	2255	0.161	0.969	1581.6	14.29
15	100.00	97.19	16.53	201	2406	0.133	0.979	1869.4	11.79
16	100.00	97.57	16.85	205	2557	0.136	0.988	1872.9	10.20
17	100.00	98.47	17.15	221	2720	0.114	0.991	1984.9	6.41
18	100.00	98.92	17.26	252	2898	0.110	0.993	1987.0	4.53
19	100.00	99.32	17.35	256	3049	0.111	0.995	2036.3	2.86
20	100.00	99.49	17.43	290	3230	0.104	0.996	2299.9	2.14

Table B-12 The summary of results for each output variable evaluated at its optimum model complexity using NNPLS.

	$PRESS_j$	$SSEP_j$	RMSECV	RMSE	LD	Parameters Per Output Variable	pdf	R^2_{cv}	R^2
C	2.18	0.522	0.348	0.244	5	797	9.20	0.609	0.879
H	5.06	0.548	0.530	0.304	4	613	12.08	0.452	0.937
O	0.776	0.160	0.208	0.140	3	462	9.82	0.646	0.937
N	0.514	0.0757	0.169	0.105	5	797	11.09	0.698	0.942
Al	0.450	0.135	0.158	0.117	3	462	8.16	0.565	0.902
K	0.015	0.0107	0.029	0.027	1	151	2.81	0.144	0.389
F	0.004	0.0031	0.015	0.014	2	308	2.17	0.000	0.151
Cu	0.0002	0.0001	0.003	0.002	3	462	7.25	0.399	0.785
Pb	0.0001	0.0000	0.002	0.001	6	948	11.04	0.671	0.936
S	0.0002	0.0001	0.004	0.003	9	1422	8.21	0.078	0.692
Cl	0.843	0.0411	0.216	0.102	7	1108	14.03	0.706	0.978
Si	0.002	0.0011	0.011	0.009	4	613	4.92	0.097	0.444
Ti	0.0002	0.0001	0.003	0.002	4	613	7.07	0.028	0.600
Fe	0.0008	0.0006	0.007	0.006	1	151	2.33	0.002	0.096
T_C	1.35	0.940	0.274	0.250	1	151	3.00	0.379	0.546
P_C	47.82	28.00	1.630	1.426	4	613	4.23	0.100	0.438
E_C	58.49	31.85	1.803	1.549	3	462	4.72	0.599	0.778
D_T	154.39	105.99	2.929	2.666	3	462	3.09	0.272	0.494
Σ	271.90	168.28		***Ave***	3.78	589	6.95	0.358	0.662

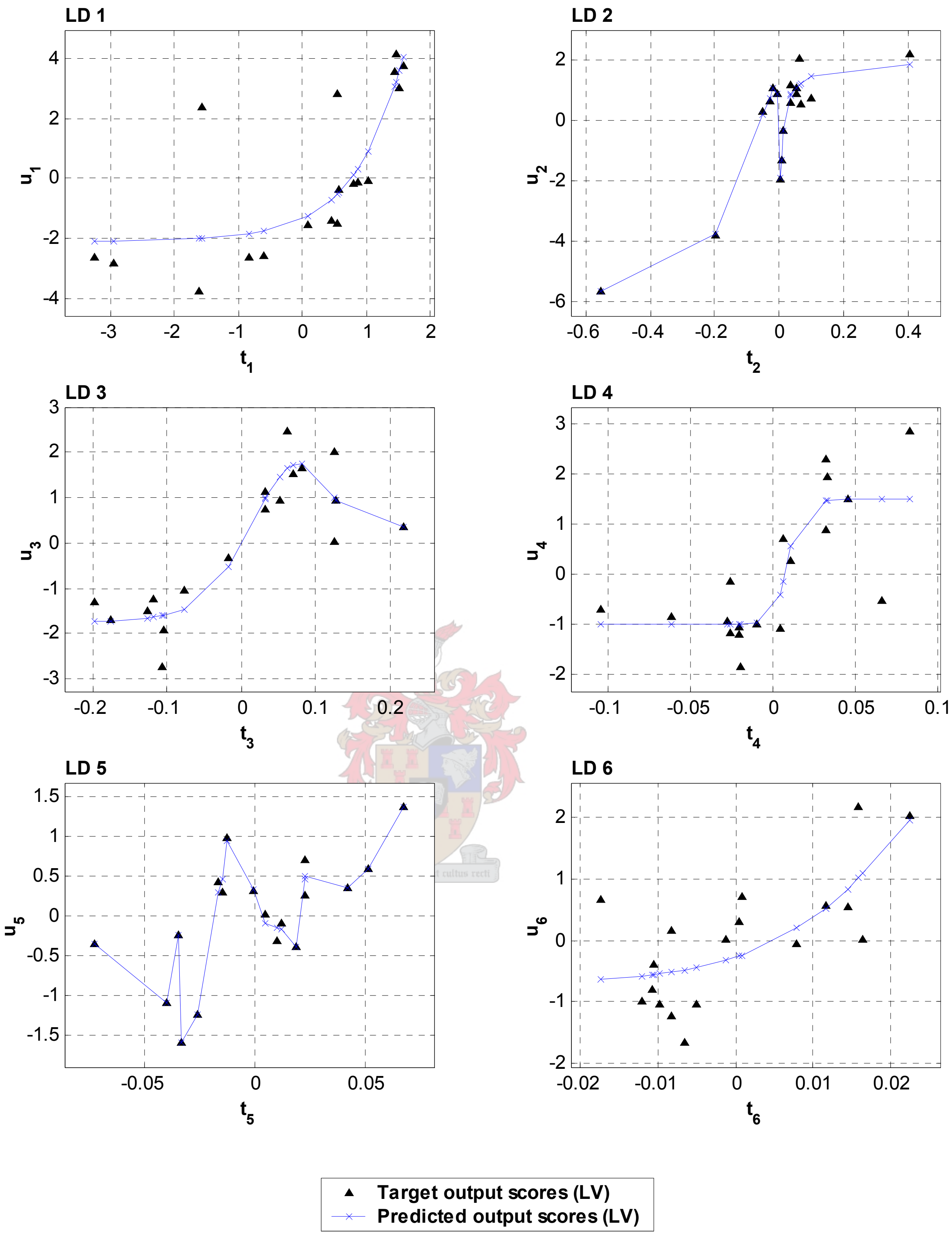

Fig. B-4 The target data and inner model relationships of the in- and output scores calculated for the first 6 latent dimensions using NNPLS.

B.4 Radial-basis function PLS

Table B-13 The sum-square residuals for the RBFPLS reverse models using different initialisation methods for the K-means method of training.

	Maximum spread initialisation			PCA binary split initialisation		
LD	PRESS	SSEP	nodes per LD	PRESS	SSEP	nodes per LD
1	361.2	281.3	3	379.7	279.5	4
2	380.3	271.7	2	404.2	267.6	3
3	341.4	200.3	4	324.1	192.3	2
4	346.6	194.3	3	370.0	182.7	4
5	344.2	181.6	1	378.2	173.8	1
6	364.0	160.3	1	400.2	175.1	1
7	378.5	158.4	1	437.3	171.7	2
8	443.5	114.0	2	434.0	172.3	1
9	450.1	94.8	2	426.4	172.4	1
10	480.8	84.7	2	447.2	166.2	2
11	437.7	34.9	1	480.9	157.3	1
12	458.0	28.3	2	485.8	157.4	1
13	499.1	26.3	2	488.7	157.6	1
14	500.3	22.6	2	489.9	157.8	1
15	495.2	21.8	1	488.2	157.9	1
16	473.5	16.7	2	496.2	156.2	1
17	484.2	15.3	2	492.8	155.6	1
18	494.3	13.1	2	485.7	156.3	1

Table B-14 The sum-square residuals for the reverse RBFPLS models using different spread values for the OLS method of training the inner models.

	4 nearest neighbours			3 nearest neighbours		
LD	PRESS	SSEP	nodes per LD	PRESS	SSEP	nodes per LD
1	346.90	280.38	1	360.39	316.08	1
2	350.47	258.67	9	347.01	294.35	4
3	322.47	224.47	4	311.21	262.14	4
4	356.58	219.62	6	337.18	254.90	4
5	330.14	148.52	3	309.81	193.83	7
6	371.04	146.00	1	351.96	187.92	1
7	398.42	108.48	7	362.74	143.34	3
8	379.85	99.02	3	374.91	131.72	7
9	369.39	65.70	1	375.64	63.54	11
10	379.37	63.22	1	385.96	68.40	1
11	360.06	53.90	1	421.57	62.30	1
12	370.82	49.80	3	433.46	49.23	1
13	365.59	43.05	1	434.02	43.20	1
14	355.89	42.11	1	434.53	42.29	1
15	351.38	40.33	6	434.41	40.30	1
16	349.90	35.48	1	425.77	35.81	1
17	344.50	35.26	1	426.61	31.91	6
18	353.79	27.37	1	421.43	24.28	6

Table B-15 Leave-one-out cross-validation for 3 runs with 2,3 and 4 initialisation nearest neighbours respectively (for ASOLS inner model training).

LD	PRESS Run 1	PRESS Run 2	PRESS Run 3*	Average
1	397.98	422.17	365.72	395.29
2	427.68	427.84	349.40	401.64
3	376.69	335.86	299.09	337.21
4	374.02	349.21	298.99	340.74
5	330.57	323.71	276.07	310.12
6	366.02	390.94	320.22	359.06
7	390.58	393.95	309.61	364.72
8	412.44	379.01	309.88	367.11
9	407.16	371.57	303.27	360.67
10	418.73	376.99	294.77	363.50
11	404.91	350.86	284.89	346.89
12	424.62	358.61	282.19	355.14
13	418.36	363.37	337.68	373.14
14	412.75	369.24	329.78	370.59
15	394.26	350.75	369.15	371.39
16	376.61	341.69	365.55	361.28
17	378.72	348.79	371.26	366.26
18	329.80	351.77	372.35	351.31
19	319.94	363.17	374.02	352.38
20	318.96	364.79	371.93	351.89

*** The best run chosen for further evaluation**

Table B-16 The number of hidden nodes for each inner model (LD) obtained from each modified cross-validation run using RBFPLS with ASOLS.

LD	Run 1	Run 2	Run 3	Rounded Average
1	3	2	3	3
2	1	1	10	4
3	2	8	3	4
4	4	1	1	2
5	6	1	5	4
6	3	2	2	2
7	11	2	5	6
8	1	4	1	2
9	4	1	2	2
10	3	6	7	5
11	1	11	2	5
12	2	2	2	2
13	13	8	1	7
14	3	1	3	2
15	6	1	3	3
16	7	1	2	3
17	13	6	2	7
18	4	9	1	5
19	4	1	3	3
20	8	5	7	7

Table B-17 The overall reverse model training results for each of the 3 runs using the RBFPLS algorithm with ASOLS training of the inner models.

LD	SSEP Run 1	SSEP Run 2	SSEP Run 3*	SSEP Average
1	281.97	274.45	262.11	272.84
2	274.04	267.22	240.94	260.73
3	217.36	226.02	182.76	208.71
4	213.24	223.50	178.41	205.05
5	155.25	209.84	178.18	181.09
6	122.19	180.60	101.12	134.64
7	114.12	181.39	99.34	131.62
8	97.28	120.16	78.42	98.62
9	78.80	92.47	62.65	77.97
10	57.01	77.22	47.22	60.48
11	34.98	47.91	50.61	44.50
12	32.41	41.05	46.56	40.01
13	24.51	39.30	36.17	33.33
14	23.53	35.74	34.25	31.17
15	18.44	30.75	26.40	25.20
16	16.24	27.21	19.76	21.07
17	14.55	24.46	19.62	19.55
18	9.81	14.68	18.62	14.37
19	6.73	11.02	16.66	11.47
20	5.02	4.97	15.28	8.42

*** The best run chosen for further evaluation**

Table B-18 Summary of results obtained from the reverse RBFPLS model with ASOLS inner model training evaluated on the rescaled data.

	Cumulative % η^2								
LD	X-block (inputs)	Y-block (outputs)	Average pdf	Number of inner model paramters	Parameters Per Output Variable	Average R^2_{cv}	Average R^2	PRESS	SSEP
1	97.82	37.56	2.97	10	157	0.136	0.312	365.72	262.11
2	99.21	42.60	5.01	41	335	0.219	0.518	349.40	240.94
3	99.84	56.46	6.07	51	492	0.320	0.607	299.09	182.76
4	99.93	57.50	7.22	55	643	0.300	0.646	298.99	178.41
5	99.98	57.55	6.82	71	806	0.383	0.669	276.07	178.18
6	99.99	75.91	7.69	78	960	0.355	0.701	320.22	101.12
7	100.00	76.33	8.16	94	1123	0.350	0.738	309.61	99.34
8	100.00	81.32	9.25	98	1274	0.363	0.780	309.88	78.42
9	100.00	85.08	10.67	105	1428	0.359	0.825	303.27	62.65
10	100.00	88.75	11.50	127	1597	0.368	0.873	294.77	47.22
11	100.00	87.94	11.78	134	1751	0.368	0.891	284.89	50.61
12	100.00	88.91	12.06	141	1905	0.360	0.905	282.19	46.56
13	100.0	91.38	12.68	145	2056	0.338	0.917	337.68	36.17
14	100.0	91.84	13.45	155	2213	0.341	0.940	329.78	34.25
15	100.0	93.71	14.24	165	2370	0.329	0.955	369.15	26.40
16	100.0	95.29	14.48	172	2524	0.333	0.962	365.55	19.76
17	100.0	95.33	14.72	179	2678	0.339	0.969	371.26	19.62
18	100.0	95.56	14.94	183	2829	0.349	0.972	372.35	18.62
19	100.0	96.03	15.34	193	2986	0.355	0.977	374.02	16.66
20	100.0	96.36	15.51	215	3155	0.357	0.980	371.93	15.28

Table B-19 The summary of results for each output variable evaluated at its optimum model complexity using RBFPLS.

	PRESS$_j$	SSEP$_j$	RMSECV	RMSE	LD	Parameters Per Output Variable	pdf	R$^2_{cv}$	R^2
C	1.747	0.41	0.312	0.217	5	806	9.26	0.597	0.904
H	2.114	0.75	0.343	0.264	5	806	7.31	0.748	0.910
O	0.894	0.13	0.223	0.139	5	806	11.03	0.598	0.941
N	0.310	0.03	0.131	0.076	12	1905	12.03	0.781	0.974
Al	0.424	0.08	0.153	0.100	5	806	10.41	0.617	0.937
K	0.0137	0.01	0.028	0.026	1	157	2.30	0.200	0.402
F	0.0040	0.00	0.015	0.014	1	157	2.09	0.000	0.151
Cu	0.0001	0.00	0.002	0.002	6	960	0.79	0.679	0.706
Pb	0.0001	0.00	0.003	0.002	5	806	10.92	0.516	0.923
S	0.0003	0.00	0.004	0.004	2	335	1.92	0.398	0.069
Cl	0.3810	0.01	0.146	0.062	14	2213	14.78	0.795	0.993
Si	0.0012	0.00	0.008	0.005	14	2213	11.21	0.447	0.917
Ti	0.0002	0.00	0.003	0.003	2	335	6.19	0.004	0.486
Fe	0.0008	0.00	0.007	0.006	1	157	2.35	0.004	0.099
T_C	1.59	0.83	0.297	0.253	5	806	4.99	0.269	0.594
P_C	45.31	32.84	1.587	1.464	2	335	2.67	0.089	0.390
E_C	51.57	8.23	1.693	1.070	11	1751	10.81	0.646	0.942
D_T	150.56	30.58	2.892	1.942	12	1905	9.89	0.300	0.888
Σ	254.91	73.92		***Ave***	6.00	959	7.27	0.427	0.679

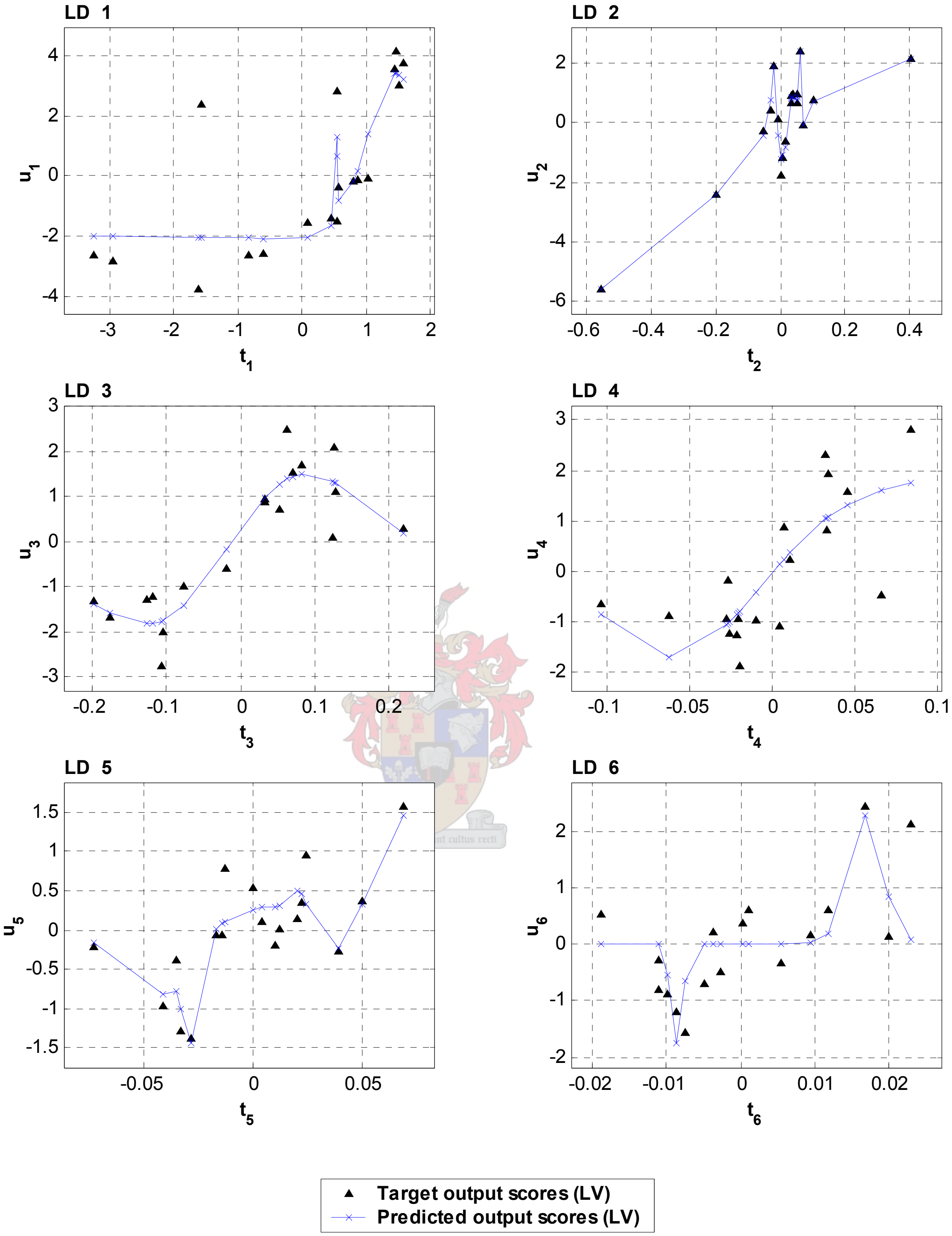

Fig. B-5 The target data and inner model relationships of the in- and output scores calculated for the first 6 latent dimensions using RBFPLS.

B.5 Sensitivity Analysis

Table B-20 The global modifications made to rocket motor C11 re-modelled by the reverse overall NNPLS model with 3 latent dimensions to investigate the effect of the forward modelled spectral irradiance.

	C11	C11-A	C11-B	C11-C	C11-D
C	1.047	0.806	1.103	0.994	1.266
H	4.382	4.249	3.997	4.220	3.847
O	2.796	2.416	2.772	2.856	3.142
N	0.701	0.523	0.716	0.729	0.965
Al	0.148	0.497	0.233	0.108	-0.062
K	0	0.00	0.01	0.00	0.00
F	0	0.00	0.00	0.00	0.00
Cu	0	0.00	0.00	0.00	0.00
Pb	0	0.00	0.00	0.00	0.00
S	0	0.00	0.00	0.00	0.00
Cl	0.689	0.747	0.585	0.678	0.492
Si	0	0.00	0.01	0.01	0.01
Ti	0	0.00	0.00	0.00	0.00
Fe	0	0.01	0.01	0.01	0.00
T_c	2.86	3.24	2.95	2.93	2.83
P_c	1.00	2.47	3.01	2.68	3.22
E_c	4.50	4.88	6.31	4.91	5.04
D_T	15.00	15.09	13.81	15.15	14.98