The Development and Application of Online Modelling Methods

Thomas Ian Dearing

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

University of Hull

Kingston upon Hull, United Kingdom

2008

Table of Contents

Table of Figures

Glossary of Abbreviations

API	Active Pharmaceutical Ingredient
ATR	Attenuated Total Reflection
DOE	Design of Experiments
EMSC	Extended Multiplicative Signal Correction
FDA	U. S. Food and Drug Administration
FID	Free Induced Decay
FT	Fourier Transform
GC	Gas Chromatography
GUI	Graphic User Interface
HPLC	High Performance Liquid Chromatography
IR	Infrared
LV	Latent Variable
MHz	Mega Hertz
MLR	Multiple Linear Regressions
MS	Mass Spectrometry
MSC	Multiplicative Scatter Correction
NIPALS	Nonlinear Iterative Partial Least Squares
NIR	Near Infrared
NMR	Nuclear Magnetic Resonance
OPEC	Organization of Petroleum Exporting Countries
OSC	Orthogonal Signal Correction
PAT	Process Analytical Technology
PC	Principal Components
PCA	Principal Component Analysis

PLS	Partial Least Squares
QA	Quality Assurance
QC	Quality Control
r.f.	Radio Frequency
RMSEC	Root Mean Square Error in Calibration
RMSEP	Root Mean Square Error in Prediction
RMSECV	Root Mean Square Error in Cross Validation
SIMCA	Soft Independent Modelling of Class Analogy
SMCR	Self-Modelling Curve Resolution
SNV	Standard Normal Variates
UV	Ultra Violet
UV-Vis	Ultra Violet and Visible
XS	Xylene Soluble

1 Abstract

Chemometrics and Design of Experiments (DOE) are fast becoming integral parts of process analysis and incorporated into the resulting advances in technology. To this end, two major studies were undertaken to explore the existing methods of modelling using both traditional and modern forms of process analytical technology, and to create new methods using the most current developments in the field.

The first study involved the use of chemometrics and DOE with low-resolution NMR FID spectra of a series of polymers that were collected over a period of ten months. Accompanying the NMR FID spectra were the associated laboratory reference measurements for a series of quality assurance parameters. This information was used to build an online prediction model for the Xylene Soluble (XS) content of polymer pellets. The installation of the online model was accomplished in numerous stages during which various sample selection methods, including work by Shenk and Westerhaus, were developed and evaluated. The intrinsic nature of the NMR data meant that traditional methods of sample selection could not be employed. The final model used the principal component analysis scores as a means of selecting samples for calibration. DOE was used to determine the best method of pre-processing to be applied to the data prior to partial least squares modelling. The final PLS model was evaluated and the error in prediction for the XS content was found to be 0.672%. The success of this project lead to the installation of this product online at the point of analysis in December 2006.

The second study employed chemometrics and DOE with a more traditional method of process analytical technology, the NIR spectral analysis of pharmaceutical tablets. The NIR spectra of over 250 tablets were collected over three production

campaigns from 1997 to 1999. Accompanying the NIR spectral data were the chemical and physical tablet parameters, active pharmaceutical ingredient, weight, and tablet thickness. The sample selection techniques developed as part of the polymer study were evaluated. In order to correct for variations due to specular and diffuse scattering effects, extended multiplicative scatter correction was applied to the data. As with the polymer study, DOE was used to determine the best method of data pre-processing prior to the partial least squares modelling. The best method of sample selection for this study was found to be the use of the condition number. The final prediction models for the active pharmaceutical ingredient, weight, and tablet thickness were produced. The final step for this study would be to apply this model online at the point of analysis in the same manner as the polymer study.

2 Aims and Objectives

The main aims and objective of this research were to develop a system of modelling that, when applied to any set of data, would always result in an optimised, robust model. Once produced, the models could then be used to predict various quality control parameters relating to production processes. In order to be broadly applicable, models must be able to handle and characterise data obtained from a variety of sources. The models constructed must also be expanded to cover grade-based distributions as well as normally-distributed laboratory reference data. Finally, all of the models produced must be applied online to make real-time predictions.

Progress toward these aims and objectives will proceed with the analysis of two real process analytical data sets. The first data set to be examined is the NMR spectra from the production of polypropylene, the second data set comprised of NIR spectroscopic information collected from packaged pharmaceutical tablets. Using this data a series of chemometric methods will be developed to generate robust predictive models that can be applied online to make real-time measurements.

3 Introduction

3.1 Process Analytical Technology

Process Analytical Technology (PAT) is defined by Kowalski as "the application of analytical science to the monitoring and control of industrial chemical processes."[1] The information attained through the monitoring of an industrial process using PAT can be used to control output and increase performance of the chemical process for the optimisation of the processing rate, the quality of the final product, the cost of production, and a reduction of waste.

The concept of PAT is not a new one, and it has been employed within the petrochemical industry for over fifty years. Within the last couple of decades, however, PAT experienced a renaissance and rapidly expanded into newer industrial spheres such as food science and pharmaceuticals. The necessity of this technological evolution was brought about by a combination of large-scale, large-unit cost processes and a dramatic increase in regulations from governing bodies such as the FDA. Another major factor driving PAT to the forefront of cost minimisation was OPEC's response to the 1973 Arab-Israeli war; OPEC increased the cost of crude oil, which forced the petrochemical industry to be much more conscious of costs for the first time.

As PAT has become more prominent, the shift from the traditional analytical set-up to a more localised set-up has occurred. In the traditional practice, one which could be defined as an "off-line method" of analysis, a sample is taken from a process stream and then transported to an off-site laboratory equipped with modern analytical instruments and a highly-trained professional staff. Using these resources, the analysis is performed, with a typical run time of a few hours to a full day. Due to

the retrospective nature of this kind of analysis, additional time must be incorporated into process cycles to accommodate any reworking or altering of reaction conditions that might be necessary. The advantages of this traditional method of analysis are that the analysis is performed by an expert analyst, there are flexible operation procedures such that the instrument can be used for any reaction stream or process, the instrument can be utilised for many forms of analysis, and one instrument can be used on a number of processes on various projects, reducing overall costs and overheads.

The introduction of PAT has helped to dramatically reduce the timeline of analysis and has moved the analysis framework from an off-line method to an "at-line" or "online" strategy. With PAT, a sample is taken directly from the process stream (generally via an automated process using pre-set sampling parameters), and the sample is analysed using a specific process instrument that is situated either next to or directly on the process stream. When the instrument is situated next to the process stream, this is an at-line method; an online method is situated right on the process stream. The PAT instruments used vary significantly from the traditional instruments employed in the traditional analysis laboratory. The PAT machines are far more robust and must be able to accommodate the conditional variations that are present in an industrial manufacturing setting but not typically present in a traditional laboratory. Such conditional variations may include external temperature fluctuations, changes in process flow rates, overall reaction times, and other pressures that can occur in an uncontrolled location.

An at-line process has a dedicated instrument, and, due to the immediate and on-site processing, the turnaround from sampling to results is reduced substantially in comparison to off-line methods. However, this advantage is offset by the limitations

of dedicating an entire instrument to a single process; a dedicated instrument cannot be used for any other task during downtimes, which is more costly than having one equipment item that can be used for multiple tasks, in addition to the often substantial price tag for the initial purchase of robust instrumentation.

Online methods are the pinnacle of PAT, as their full automation and dedication to a single process allow for the immediacy of results. The feedback from analysis can be almost instant, and process feedback control can be driven by a single skilled technician. Furthermore, online analysers can be used to analyse each sample within a given process stream (such as every tablet produced by a reaction), ensuring that every single sample produced meets specifications. To maximise productivity and minimise costs when using online systems, downtime should be minimised, as time when the processor is not running is time when no measurements are being taken thus no knowledge of the quality of the final product being produced; furthermore, these systems must be continually maintained to ensure all analysis performed is within certain tolerances. With the establishment of the online method, a form of sample pre-treatment may also have to be performed to facilitate the analysis of samples, which is not usually necessary for off-line methods where pre-treatment within a lab environment is straightforward.

The PAT instruments have chemometric analysis software to perform all the necessary steps to analyse samples and provide feedback. This means that a desired process parameter, such as an octane number, can be fed directly to a control centre where the fabrication process can be altered or optimised. It is important to highlight that, unlike the traditional method, the PAT approach does not require a highly-trained technical staff, as most PAT instruments are fully-automated and therefore

only require routine maintenance by non-technical staff. The in-built chemometrics software and models can be updated remotely by one highly-trained chemometrician.

The first transition from the traditional approach to the PAT approach involved the relocation of laboratory instruments next to the process streams. This shift initially lead to major advancements in areas such as process gas chromatography (GC), process high performance liquid chromatography (HPLC) and process nuclear magnetic resonance (NMR) spectroscopy.[2] More recently, the ability to make instruments more robust to environmental factors and to miniaturise what were traditionally large instruments has meant that spectroscopic methods such as process Near and mid-infrared spectroscopy,[3] process Raman spectroscopy, and process mass spectrometry (MS) have made significant advances. However, one of the major drawbacks to the PAT approach occurs in the first stage, sampling. In order to take the necessary samples without disrupting the production process or compromising the quality of the sample itself, a series of non-invasive sampling methods and tools have become increasingly important, such as process microwave spectroscopy,[4] acoustical analysis,[5, 6] and NIR probes.

Currently, the majority of PAT is implemented by retrofitting the current process equipment and blending old and new technologies to better address advances in industrial manufacturing. The future of PAT lies with industrial corporations building physical plants that have provisions for PAT included from the inception of the design process, along with the continued improvement of current PAT instrumentation. Increasing regulatory control by governing bodies such as the U. S. Food and Drug Administration (FDA)[7] means that, in the future, PAT will always play an important role in the manufacturing industry and will no longer be tied to

specific industrial processes or products. PAT will appear across the board throughout all industrial procedures.

As described above, the evolution of PAT has also re-established chemometrics as a hot field for research in both academia and industry. While there were no significant shifts in the actual chemometrics involved, the primary vehicle of change was the computer and processing technology used to perform the data analysis and modelling. Until the 1980s, chemometrics had been relegated to an area of theoretical mathematics simply because the necessary calculations required for real analysis were so time-consuming and therefore largely impractical. With the invention and popularisation of the microprocessor and the desktop computer, the calculations needed for chemometrics became easier to perform, and the field subsequently found a wider audience, including those involved in the PAT initiative. By employing chemometrics, the PAT instruments could perform real-time analysis and modelling on the data collected with only a very small time delay. This enabled real-time feedback to allow for control and optimisation, completing the PAT agenda and firmly linking the fields of PAT and chemometrics. This partnership has lead to the developments of methods and algorithms such as self-modelling curve resolution (SMCR),[8] orthogonal signal correction (OSC),[9] and extended multiplicative scatter correction (EMSC),[10-12] that are designed to correct for the variations observed in large-scale processing situations.

3.2 Near Infrared Spectroscopy

Near infrared (NIR) spectroscopy was first discovered over two hundred years ago by Herschel, but it wasn't until the early 1950s when NIR spectroscopy was first considered to be more than just and extension of the mid-IR fingerprint region.

The NIR region of an electromagnetic spectrum ranges from 12,800cm^{-1} to 4000cm^{-1}. NIR spectroscopy is concerned with the absorbance of NIR energy that occurs in this region by the molecules in a given sample. Absorption can occur by three different means: combinations, overtones, and electronic absorptions. A combination occurs when the absorption of a photon is shared between two or more vibrations. This would be observed as a single peak in the near infrared region but as two fundamental peaks in the mid-infrared region. Overtones are approximately multiples of the fundamental vibration; for example, the fundamental x will have overtones of 2x, 3x, etc., respectively called the first and second overtones. The intensity of successive overtones decreases by a factor ranging between 10 and 100. Electronic absorptions are caused by the movement of electrons from one orbit to a higher-energy orbit; these are normally observed in the UV-Vis range but can also appear in the NIR in the region from 12,800 to 9000cm^{-1}.

Combinations and overtones provide the major contributions to NIR spectra. In 1965, the chemometric technique of multiple linear regressions (MLR) was developed by Norris. MLR allowed for NIR calibrations without operator interference. This had both advantages and disadvantages, in that the user could efficiently find relevant information related to a property of interest, but this "black box" approach to calibrations could mislead the user into thinking there was a significant correlation when there was not. Still, it was not until the advent of micro-

processing and the application of advanced chemometrics in the 1980s that NIR really took off.[13]

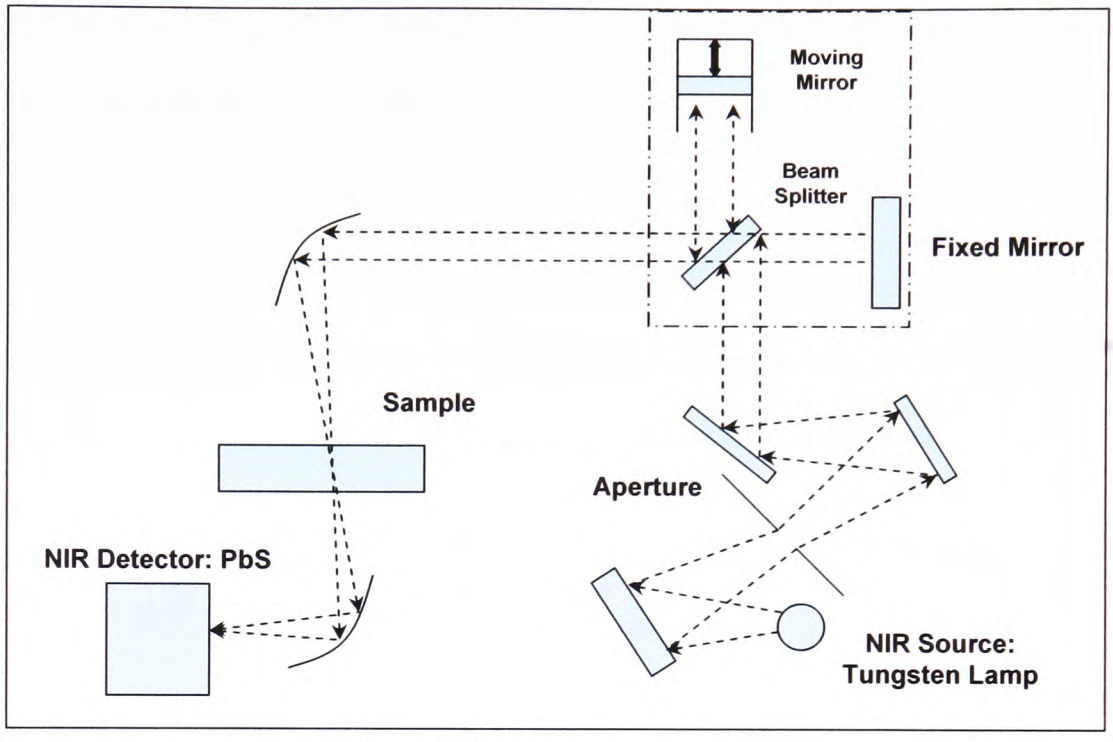

Figure 1. An FT-NIR spectrometer.
The dashed box denotes the interferometer component of the instrument.

A FT-NIR spectrometer (Figure 1) has three main components: the NIR source, the detector, and the interferometer. The interferometer is only present in an instrument that uses the Fourier transform (FT). The NIR source used is a Tungsten lamp, which emits radiation across a wide range of the electromagnetic spectrum, and the NIR region can be examined with the use of a Lead Sulphide detector.

The interferometer contains two mirrors (one moving and one stationary) and a beam splitter. Radiation from the source is split by the beam splitter and then directed to the fixed and movable mirror in equal amounts. The beam splitter can be made from SiO_2 or CaF_2 to function correctly with the NIR source and detector. The moving mirror is scanned at a constant velocity, resulting in the changing of optical path differences of the beams as a function of time. The reflected beams converge at the beam splitter, with half of the radiation returning to the source and half

continuing to the detector. The detector measures the intensity as a function of the optical path difference in both branches of the interferometer. The signal is called an interferogram. The Fourier transform is applied to the interferogram to generate the transmission spectra of the sample.

3.2.1 Fourier Transform

The Fourier transform (FT) is a mathematical operation that converts spectra from a time domain to a frequency domain and vice versa. The transform breaks down the interferogram into its sine and cosine constituents (Equation 1).

$$X = \frac{1}{2N+1}\sum_{-k}^{+k} f(k\Delta t) \quad \left(\cos\frac{2\pi kn}{2N+1} + j\sin\frac{2\pi kn}{2N+1}\right)$$

Equation 1

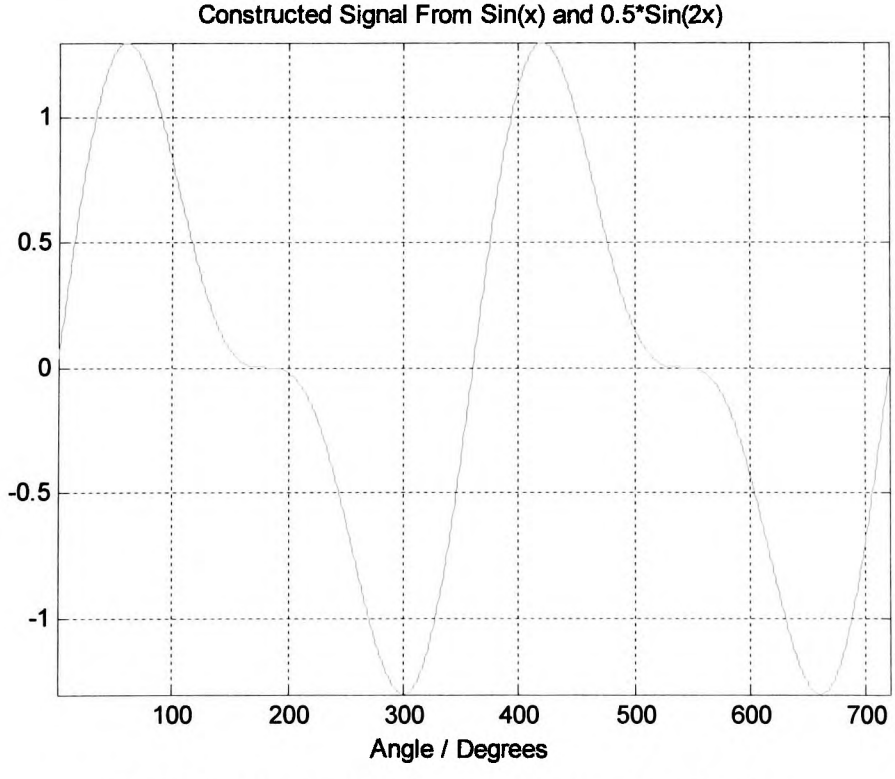

Figure 2. A signal constructed from sin(x) and 0.5*sin(2x).

Figure 2 shows a signal constructed from sin(x) + 0.5*sin(2x). When this signal is transformed using the Fourier transform it is broken down into the individual sine waves. Figure 3 shows the Fourier transform of the constructed signal, and there are two peaks. The first peak has twice the amplitude but half the frequency of the second peak, as this corresponds to the sin(x) portion of the signal. The second peak relates to the 0.5*sin(2x) segment of the signal, so is has twice the frequency but half the amplitude of the first peak.

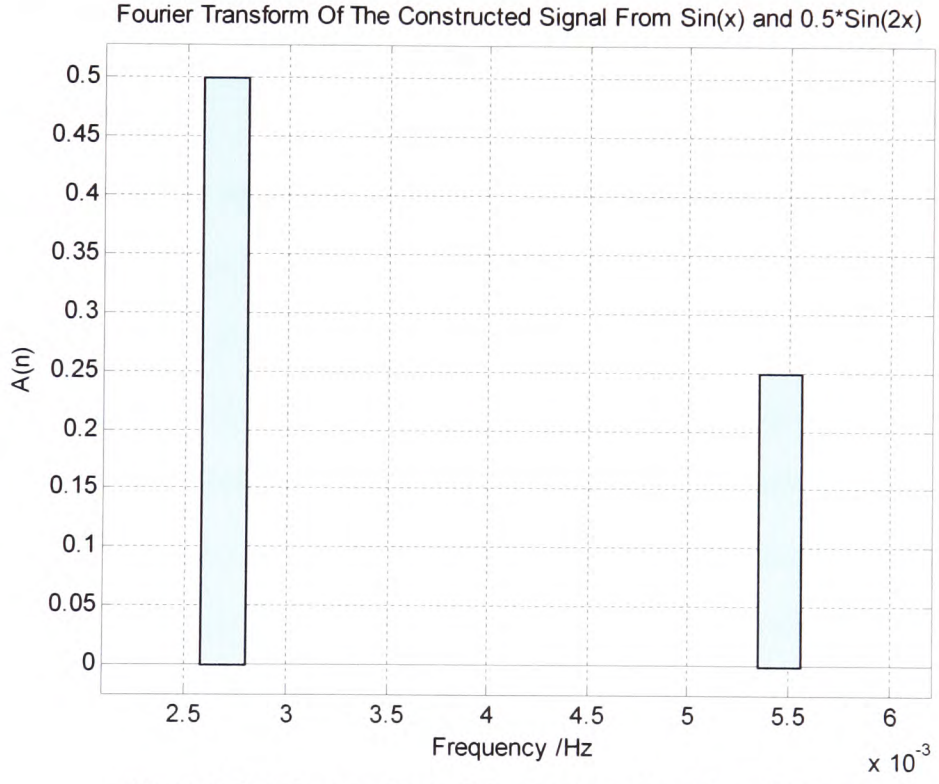

Figure 3. Fourier transform of the construct signal from Figure 2.

3.2.2 Near Infrared and Process Analytical Technology

FT-NIR has many advantages when used within industry, as it offers fast scan times, a high degree of precision, a non-invasive method of analysis, and low maintenance requirements. FT-NIR also uses a very long path-length, meaning that the analysis of bulk materials can be performed with little sample pre-treatment. These advantages make FT-NIR an ideal tool for PAT, especially due to its use of

fibre optics. Fibre optics allow the NIR instrument to be fitted with a probe, so that the probe can be directly inserted into the reaction chamber and connected to the NIR via the fibre optic cables. The cables can run up to 100 meters, allowing the NIR instrument to be somewhat remote from the process stream which makes the measurement process considerably safer.

For the NIR laboratory instrument to be converted to something suitable for PAT, it must become more robust to external factors, portable, and it must allow for remote measurements using fibre optic cables.

3.2.3 Probes

The early NIR spectrometers operated in similar manner to UV spectrometers, with the sample taken off-site to the instrument for analysis. However, by the 1980s it became apparent that fibre optics allowed light to be taken directly to the sample, thus making it possible for spectroscopy to be safe and remote. The use of probes also solved the problem of the invasive nature of sampling directly from a reaction stream. Insertion of a probe into the media meant that no physical sampling needed to be performed, thus increasing the safety and significantly reducing the errors in sampling. Without fibre optics, the implementation of sampling probes would be almost impossible.

There are several different kinds of NIR probes: transmission, transflection, reflection, and attenuated total reflection.

3.2.3.1 Transmission Probes

Transmission probes can be split into two categories: insertion and flow cell. Insertion probes are typically introduced into a sample stream where measurements are recorded.

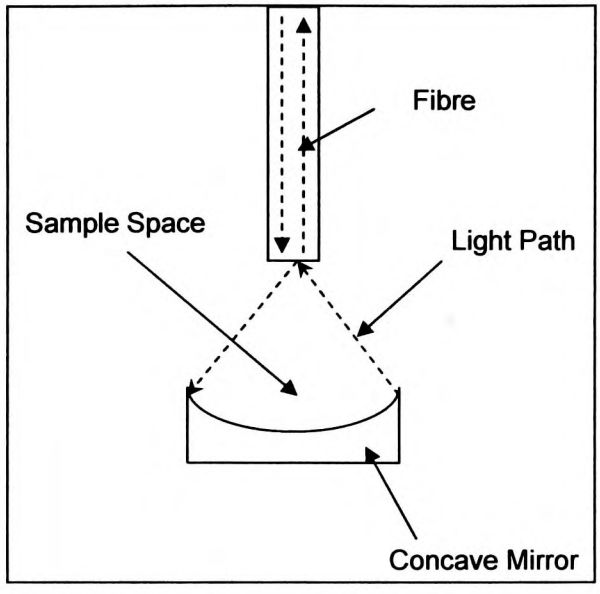

Figure 4. A simple single-fibre insertion probe.

Figure 4 shows a simple single-fibre optical insertion probe with a concave reflector. The radius of the concave reflector is equal to the distance from the fibre, thus causing total internal reflection; this directs all of the light back down the fibre. This system has major flaw in that incidental light can be returned to the detector, yielding stray light. Stray light is a major cause to the non-linear performance of the spectra. This design was quickly superseded by the design shown in Figure 5.

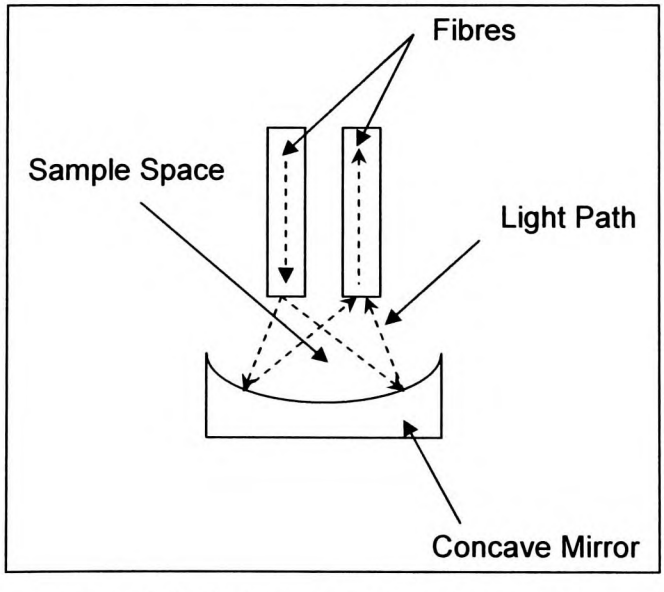

Figure 5. Revised insertion transmission probe.

Figure 5 shows a revised probe design where the returned light is separated from the transmitted light. This almost completely eliminates the problem of stray light, but this technique is only a good as the mirror employed in the instrument. A poor-quality reflector yields poor alignment and poor focus. This type of design is sensitive to the refraction variations of the sample. The variations cause a change in the divergence of the light and thus cause alignment and focus errors, which lead to baseline, offset tilt and curvature.

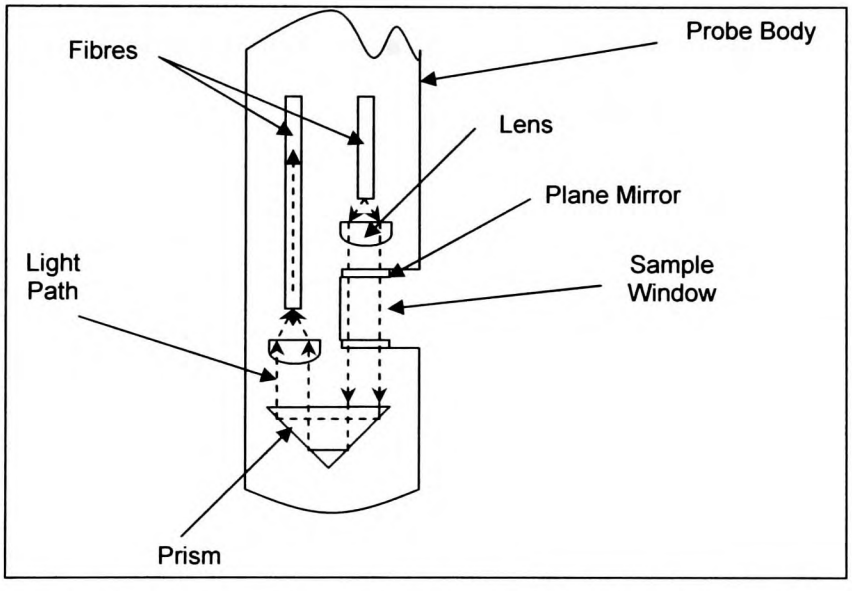

Figure 6. A common commercially-available transmission probe.

Figure 6 shows the schematic of the most common commercially-available NIR fibre optic insertion probe. This probe has an excellent design featuring good path-length definition and good optical quality. The light passes through the sample twice, thus making the path-length of the light twice the distance of the sample window. However, because this probe is encased in a body, the optical efficiency of this system is reduced to about 56% due to the Fresnel reflection losses (the reflection of a portion of light at a discrete interface between two media of different refractive indices). Unlike the designs in Figure 4 and Figure 5, the alignment and focus of the fibres in this probe can be carefully controlled. These types of insertion probes have

been used by Blanco et al. to elucidate the profiles of fermenting alcohols.[14] Blanco showed that, with the combination of NIR probes and chemometrics, an accurate profile of the fermentation process of glucose and ethanol was possible. This profile was not achievable through any other means.

3.2.3.2 Transflection Probes

Transflection probes (Figure 7) are probes that can collect spectra of the transmitted and reflected light.

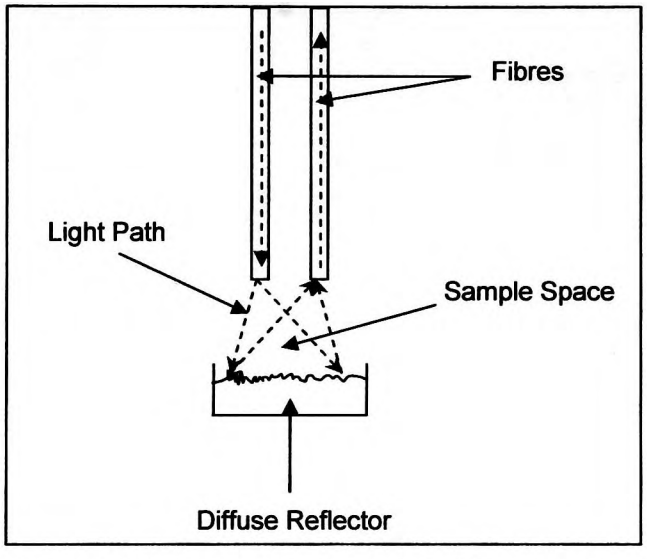

Figure 7. A typical schematic of a transflection probe.

The sample space is located between the fibres and the diffuse white reflector target. When no sample is present, the diffuse reflector scatters the light back into the receiving fibre. If the sample is a liquid the light passes through it and is then scattered by the diffuse reflector toward the receiving fibre. Solid samples scatter the light directly back to the receiving fibre. The transmission probes give higher-quality data for liquid samples, as not all of the scattered radiation is returned to the fibre. A disadvantage to the use of transmission probes is that the optical path is not well defined, and the path-length is a composite average of the individual path-lengths. This means that the average path-length is dependent upon the refractive index of the

liquid rather than being a true measurement of each individual sample, thus making quantitative studies difficult.

3.2.3.3 Reflectance Probes

Reflectance probes, unlike the previous probes, use bundles of fibres rather than single fibres in order to maximise the amount of emitted light.

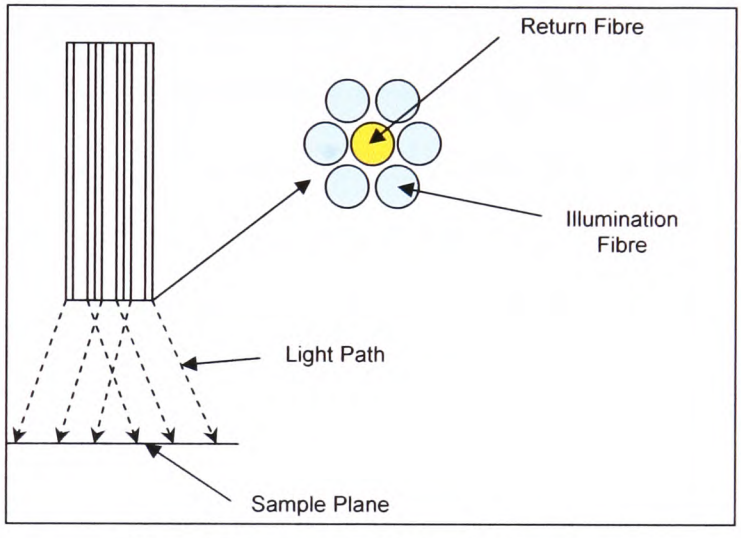

Figure 8. A 6-around-1 (6:1) reflection probe.

Figure 8 shows a close-packed 6-around-1 (or 6:1) configuration. The six outer fibres are used to illuminate the sample whilst the central fibre is used to return the light to the spectrometer. The area that is illuminated by the source fibres overlaps the area covered by the detection fibre. The percentage of overlap increases with the distance from the probe tip, although this is not a uniform effect and when the probe exceeds a distance of 2.5mm from the target the amount of reflected light is reduced dramatically.

Improvement on this design can be made by increasing the number of illumination fibres from six to nineteen. A larger return fibre is also useful, as the amount of illuminating light is increased along with the collection efficiency. A 19:1 probe still requires a small working distance between the sample and the probe. The

small target sizes and collection efficiencies of reflection probes mean that these probes may not be suitable for applications that involve grains or large granulated samples. Larger targets can be achieved by increasing the number of illumination and return fibres. Greater collection efficiency can also be attained by randomising the bundles.

On some occasions, it may be desirable for the probe to be in direct contact with the sample, and for this to be achieved without damaging the fibres a window must be in place. The disadvantage of introducing a window to the probe is that it also introduces stray light into the system via Fresnel reflections, in the same manner as was previously noted with transmission probes. None of the reflectance probes discussed can eliminate the problem of specular reflection originating at the sample; however, when using a simple 6:1 probe (for solid or sheet samples) specular reflections can be reduced by tilting the window with respect to the sample.

Reflectance probes have been successfully employed by Dunko *et al.*,[15] Garcia-Rey *et al.*,[16] and Dumitrescu *et al.*[17]

3.2.3.4 Attenuated Total Reflection Probes

ATR stands for Attenuated Total Reflection. A typical ATR probe has a truncated cone crystal, and light is shone along the length of the crystal causing total internal reflection. The light path penetrates the sample when the light is reflected off the surface in contact with the sample. The incident light then continues to reflect causing further penetrations until it reaches the end of the crystal, at which point it is collected by the detector. ATR is not used extensively in NIR applications mainly due to the small absorption coefficients for molecules in the NIR region.

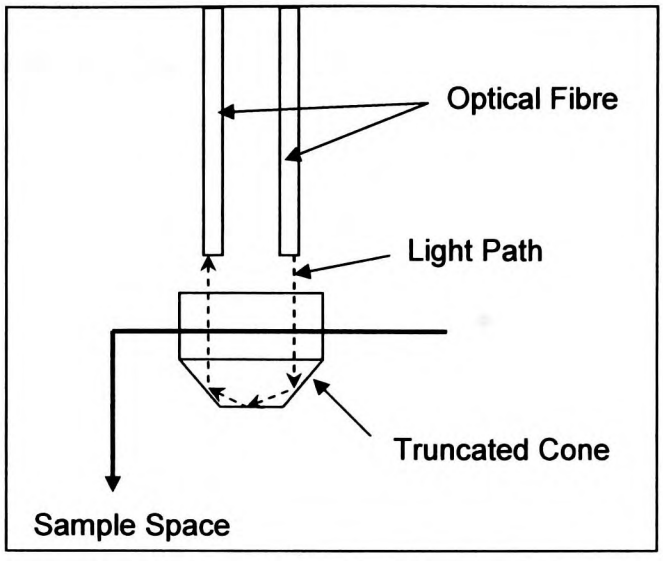

Figure 9. An ATR probe.

Figure 9 shows a typical ATR probe. This design is normally made using a truncated sapphire cone which creates three reflections of the beam. Having three reflections in a spectral region with low absorption coefficients creates a very small path-length of approximately 1μm at 3000nm. ATR probes are very sensitive to the refractive index of the sample. The refractive index of the crystal used must be significantly higher than that of the sample; if this is not the case, total internal reflection will not occur and the light will escape into the sample. Dependence on the cleanliness of the crystal, variations in refractive indices, and problems with light scatter all mean that ATR probes have limited applications within industry.

3.2.4 The Future of Near Infrared Spectroscopy

Recent regulations proposed by the FDA would require that drug manufacturers have a precise understanding of what is occurring throughout all aspects of the drug manufacturing process. These stringent regulations would require large pharmaceutical and cosmetic companies to implement more process analytical technology in order to thoroughly understand their production process. This has lead

to a greater implementation of NIR spectrometers within industry.[18, 19] Furthermore, the need for rapid and precise analysis coupled with the ability to use NIR spectrometers with fibre optics have lead companies such as BP to incorporate NIR spectroscopy as a fundamental technique in the analysis of reactions.[20-22]

3.3 Nuclear Magnetic Resonance Spectroscopy

3.3.1 Traditional Nuclear Magnetic Resonance Spectroscopy

Nuclear magnetic resonance (NMR) spectroscopy is one of the most popular techniques used within a research environment for the determination of physical, chemical, electronic, and structural information of a species. It is most widely applied in the fields of organic and inorganic chemistry for the characterisation of new compounds. A typical research instrument features magnets and a radio frequency transmitter as the primary components.

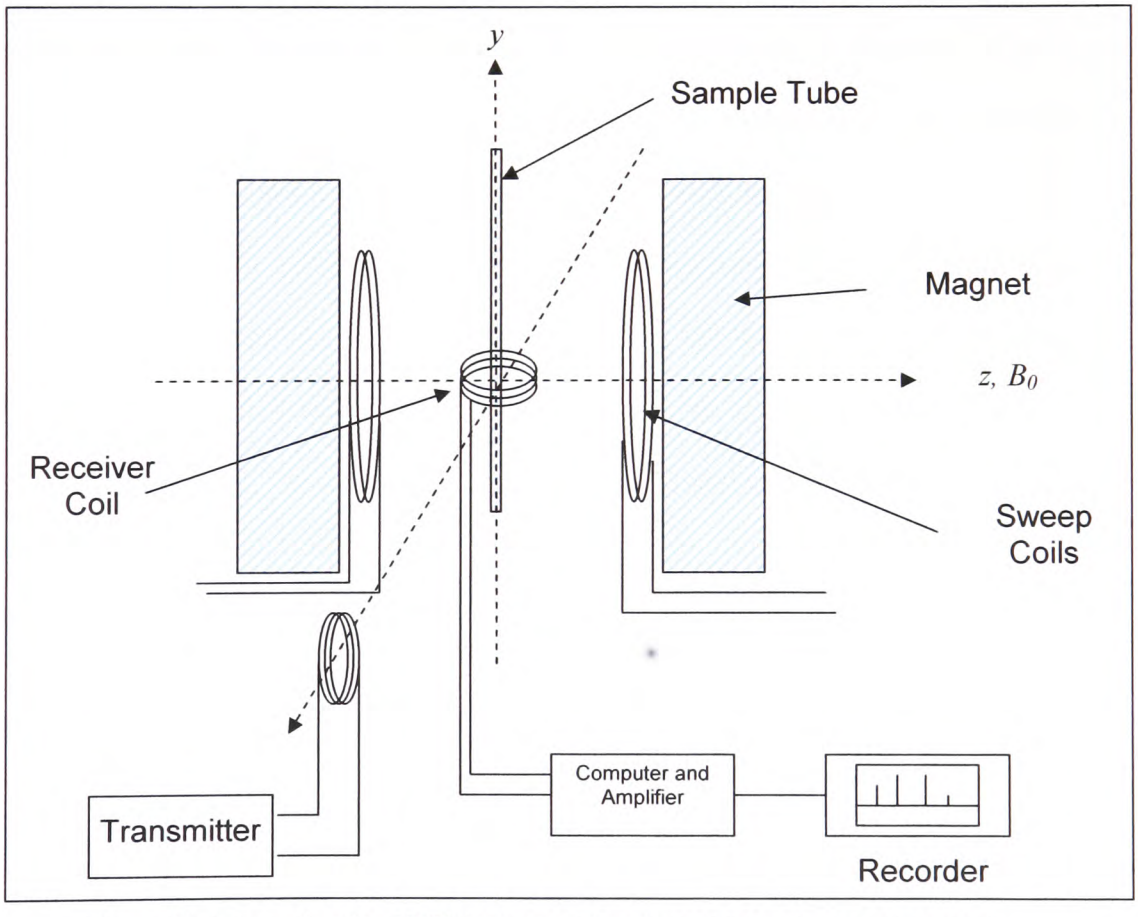

Figure 10. Schematic of an NMR instrument in a typical laboratory environment.

Most research instruments employ electromagnets, which can be tuned to the required frequency of pulsing, typically in the high-field region 200 to 750Mhz. The

radio frequency (r.f.) transmitter and receiver coils allow rotation of the net magnetic vector.

NMR spectroscopy can be broken down into various stages of both physical and magnetic arrangements: a sample is initially placed into a magnetic field, B_0; a nuclear spin processes about B_0; the spin aligns itself with B_0; and results in a net magnetisation, M_0 (Figure 11a). M_0 is parallel to B_0, assuming exponential behaviour M_0 build up along B_0 at a rate of $1/T_1$ where T_1 is the spin lattice or longitudinal relax time. After this an r.f. field is applied for a matter of milliseconds (Figure 11b). Application of the r.f. rotates M_0 away from the z-axis into the xy plane. The rate at which the spin relaxes to no given orientation is given by $1/T_2$, where T_2 is spin–spin or transverse relaxation time (Figure 11c). T_1 not only defines the time taken to generate M_0 (including placement and spinning of the sample) but it also describes the time needed for the magnetisation to return to equilibrium. Both T_1 and T_2 affect the signal strength of NMR. Line width of absorption signal after a Fourier Transform is given by $1/T_2^*$ where T_2^* is T_2 in the presence of magnetic inhomogeneities. T_2 is effectively the time required for signal or free induced decay (FID) to return to 0. The FID forms the raw signal measured using NMR.[2]

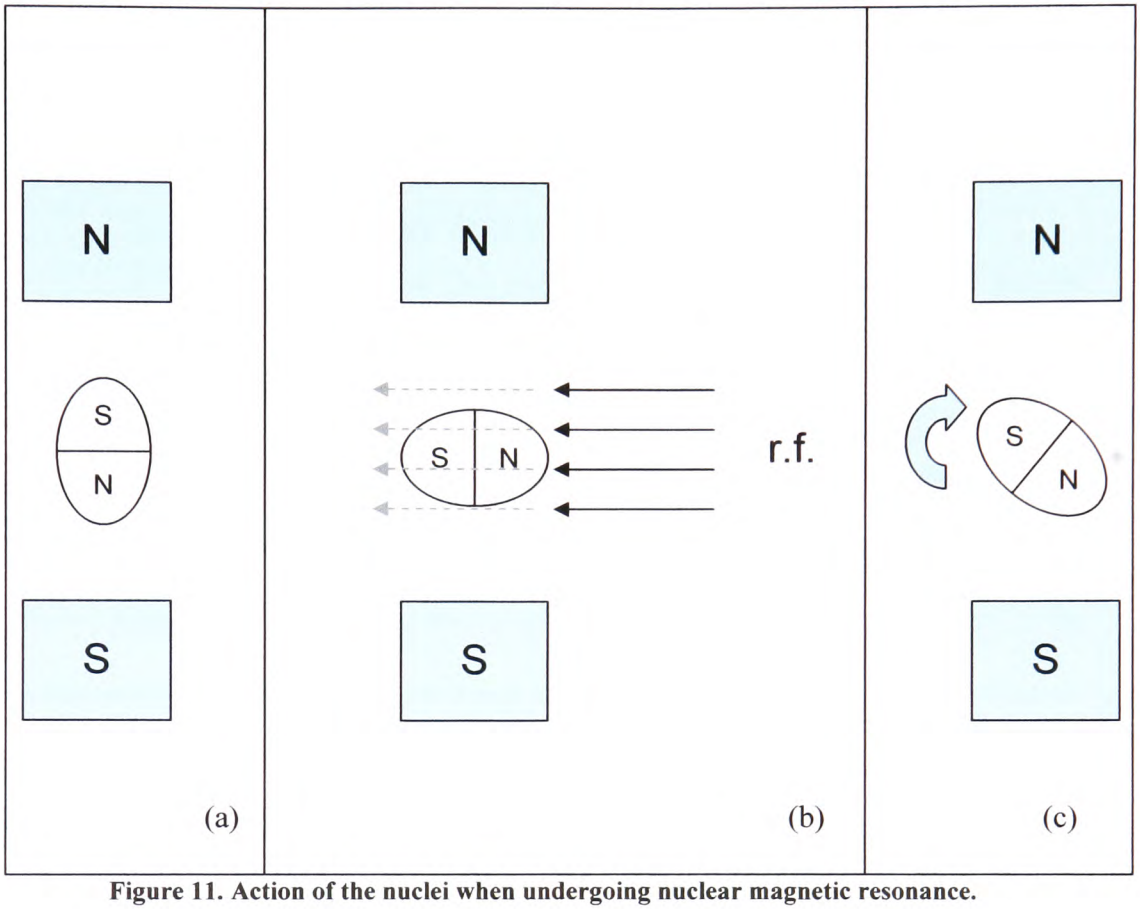

Figure 11. Action of the nuclei when undergoing nuclear magnetic resonance.

In NMR, the signal-to-noise ratio is often poor, and subsequently the signal is treated with an exponentially decaying function so that the noise at longer acquisition times is 'removed'. In most research systems, the FID undergoes a Fourier transform to produce the final NMR spectrum.

All of these effects that are present in the traditional laboratory research instrumentation rely on the sample being placed directly into the static field in an appropriate physical form. This is not the case when NMR instrumentation is used as a tool for process analysis.

3.3.2 At-line Nuclear Magnetic Resonance Analysis

Process NMR has the potential to be a very powerful technique in the world of PAT due to three factors:

1. Process NMR is non-destructive;
2. Process NMR does not require the insertion of a probe in process stream, therefore avoids issues of fouling;
3. 'Standard-less' quantitative analysis.

When process NMR was initially applied to at-line and online processes, the high-field instruments that were in place within a laboratory were simply moved to the process stream. This created problems with the calibration and maintenance of the instrumentation. In the past few years, primarily due to the production of small, dedicated low-field instruments based on permanent magnetic technologies, there has been an increase in the number of applications for at-line and online process NMR.[2] The low-field instruments have magnetic field strengths that typically range between 15 and 60MHz.

The main issue that must be addressed when using process NMR is the manner of sample insertion. Typically, NMR is used to assess the end quality of a product. Within the polymer industry, the sample being analysed tends to be in the form of a pellet. The analysing instrumentation must be able to melt the pellet so that it can be sampled as a liquid and then purge and dispose of the sample after the analysis is complete. Once it is in liquid form, the sample is fed into a sample chamber and it undergoes the same procedure as it would in research laboratory instrumentation. Unlike a laboratory instrument, a process instrument typically uses a permanent magnet as they are cheaper to use and maintain than the electromagnets used within the research setting. One of the most rapidly growing areas of process NMR is the determination of water and fats within samples, which makes it an ideal method for

the analysis of agricultural products such as dairy and corn.[2] Additionally, the intrinsic sensitivity of process NMR has lead to applications that determine the ethanol content of various alcoholic beverages.

Process NMR is still in a relative infancy compared with more-established methods of PAT, and there are still many avenues available for further investigation. In the future, process NMR has the potential to become a standard method of PAT analysis, with great opportunities for continued development and application-based research.

3.4 Chemometrics

In the past twenty five years, chemometrics has enjoyed tremendous success in fields related to calibration of spectrometers and spectroscopy based measurements. Chemometrics can be defined as the application of mathematical and statistical methods to chemical measurements.[23] Chemometrics offers many advantages when applied to calibration methods:

1. It provides speed in obtaining real-time information from data;
2. It allows high-quality information to be extracted from less-resolved data;
3. It provides clear information resolution and discrimination power when applied to second-, third-, and possibly higher-order data;
4. It provides diagnostics for the integrity and probability that the information it derives is accurate;
5. It promises to improve and reduce the number of measurements required;
6. It improves the knowledge and understanding of existing processes;
7. It techniques cost very little to apply, and can reduce the time and cost of a process.[24]

Workman *et al.* have produced a series of reviews in which they discuss many different applications of spectroscopy and chemometrics.[25-27] They summarise the reviews by stating that, without chemometrics, none of the resulting calibrations would have been possible.[24]

3.4.1 Multivariate Methods

3.4.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is a method of producing multivariate models from large and complicated data sets. This method is an upgrade from the traditional univariate models, as the multivariate method allows for the maximum amount of information to be retained within the model. PCA is performed by the

decomposition of the data matrix, **D**, into the sample scores, **U**, and the variable loadings, **V**, in accordance with

$$D = UV^T + E$$

Equation 2

where **E** is a matrix of residual errors. For the matrix **D** of size *m* x *n*, were *m* is the number of samples and *n* is the number of variables, the sample scores matrix has the size *n* x *k* and the variable loadings matrix has a size of *m* x *k*. Here *k* is the number of product vectors that can fully express **D**. These values of *k* are called the principal components (PCs). For every source of independent variation within **D** there is an associated PC. The largest source of variation is the first PC, and the second largest source of variation is the second PC, and this continues until all of the sources of variation within **D** are explained. Each PC also relates to each column of the scores matrix and each row of the variable loadings.

3.4.1.2 Non-Iterative Partial Least Squares

The decomposition of the data matrix is performed using the Non-Iterative Partial Least Squares (NIPALS) algorithm. NIPALS is the standard method for computing the principal components and the associated scores and loadings. Brereton produced an excellent text that explains the NIPALS algorithm.[28] NIPALS extracts each PC in turn, making it ideal for large data sets (such as found with spectroscopic data) that can contain over 2000 variables per sample. The sequential generation of components means that the algorithm can be halted when the desired number of PCs has been derived, saving both time and effort due to the generation of undesirable components.

NIPALS proceeds by first selecting a column from an appropriately scaled data matrix, **X**. The selected column forms the basis of an initial estimate of the scores vector, $\mathbf{U_i}$. Using $\mathbf{U_i}$ and **X**, the variable loadings are generated (Equation 3).

$$V_{un-norm} = \frac{U_i{}'.X}{\sum U^2}$$

Equation 3

These loadings are then normalised and used with **X** to calculate a new set of scores, $\mathbf{U_i}$*, (Equation 4, Equation 5).

$$V = \frac{V_{un-norm}}{\sqrt{\sum V_{un-norm}^2}}$$

Equation 4

$$U_i^* = X.V'$$

Equation 5

The two scores vectors are compared, and if the sum of the squared value of $\mathbf{U_i}$–$\mathbf{U_i}$* is large or exceeds a predetermined threshold (Equation 6), $\mathbf{U_i}$* becomes $\mathbf{U_i}$. This process of calculating loadings and new scores is repeated until the difference between $\mathbf{U_i}$ and $\mathbf{U_i}$* is small or below the predetermined threshold. At this point, the PC is determined and $\mathbf{U_i}$ becomes $\mathbf{U_k}$, the column in the scores matrix of the k^{th} PC.

$$\sum \left(U_i - U_i^*\right)^2$$

Equation 6

Following this, the information relating to the scores and loadings of the PC must be removed from **X**, to allow removal of the next PC. This is accomplished by multiplying the scores and loading together and then subtracting this product from the data matrix **X**, to form the residual data matrix, $\mathbf{X}_{res}$.

$$X_{res} = X - U.V$$

Equation 7

This residual matrix, $\mathbf{X}_{res}$, is then recycled to the beginning of the iteration procedure whereby another column is extracted. This cycle continues until all of the desired components have been removed.

3.4.1.3 Sample Scores

The sample scores can yield information about the intra-sample relationships, and this can be observed by plotting the columns of the scores matrix against one another (such as plotting the scores relating to the first PC against those of the second PC). In this case, the two largest sources of variation are plotted together, and the resulting plot can show clusters or groupings of data that suggest that the samples are related to one another by the sources of variation from the first and second PCs.

3.4.1.4 Variable Loadings

The loadings illustrate the weight or importance of each variable within the original data matrix, e.g. wavelengths, when calculating the PCs. From the loadings, it is possible to determine the variables that contribute most significantly to the sample scores, and to possibly deduce the variable responsible for the clustering, among any other observed relationships.

3.4.1.5 Eigenvalues

After completion of PCA, the size of every extracted component can be determined, and this is referred to as the eigenvalue. The first components extracted, which are the most significant components, have the largest eigenvalues. The eigenvalue is calculated from the sum of squares of the principal component scores.

$$\lambda_k = \sum_{t=1}^{T} U_{tk}^2$$

Equation 8

where λ_k is the eigenvalue associated with the k^{th} principal component (Equation 8).

3.4.1.6 Modelling Using Principal Component Analysis

One of the first methods applied to modelling using PCA was Soft Independent Modelling of Class Analogy (SIMCA). Soft modelling refers to a situation in which different classes of information overlap, essentially allowing a sample to belong to more that one class. For example, a chemical compound could contain both carbonyl and alkene functionality, and it could therefore fit into the class of alkenes or the class of carbonyls. SIMCA begins with PCA, but only the most significant principal components are retained. Independent modelling of each class (i.e. carbonyls and alkenes) is performed by calculating the orthogonal distances of each sample from a plane. New samples can be projected into the model, and the classification of new samples is performed by determining to which class or classes the sample belongs.

When using PCA to model a system of data, the number of PCs to be included in the model must be determined. There are many methods for this, and in theory the number of PCs to be included in the model is equal to the number of chemical

constituents in the analytical system; for example, the data generated from the UV spectra of differing mixtures of four metallic compounds should have a PCA model that includes four PCs. This, however, is a simplified example, and more complicated methods of determining the number of PCs to be included must be employed. These methods include use of an F-Test to determine the statistically significant PCs to be included in the model or use of a DOE approach to calculate the optimum number of PCs based upon the quality of the final model.

The use of PCA as modelling method has become less frequent since the advent and wide-scale adoption of Partial Least Squares (PLS), primarily due to the fact that PLS allows the user to produce a model that can correlate spectral information with quantitative values, such as concentration.

3.4.2 Partial Least Squares

Partial Least Squares is another method of data reduction, but unlike PCA, PLS uses both the multivariate spectra, **X**, and the corresponding concentrations of other reference information, **y**, in the decomposition to produce the PLS scores and loadings.

As with many other chemometric methods, PLS evolved from the field of economics, and it was in the late 1960s that PLS was explored for non-economic purposes by H. Wold. The use of PLS for chemical applications was pioneered by groups led by S. Wold and H. Martens during the 1970s. The 1980s saw some of the first publications of articles highlighting the use of PLS in what has become a traditional use of chemometrics,[29-39] and it was this decade that essentially marked the renaissance of PLS as a tool for chemometric analysis as opposed to its previous use as a method of economic analysis. From the 1990s to the present, the use of PLS has almost become a standard approach and many variations of the original PLS

have been produced that range in use from non-linear applications[40-44] to multiple simultaneous predictions.[44, 45]

3.4.2.1 The Partial Least Squares Algorithm

The PLS method begins by finding the first PLS direction. This begins as with PCA (see section 3.4.1.2). During PLS the spectral data may be scaled, and this same scaling must be applied to the concentration information. The algorithm begins by calculating the loading weights vector, **h**

$$h = X^{'}.y$$

Equation 9

The spectral scores are then determined using the loading weight vector and the spectral data (Equation 10).

$$U = \frac{X.h}{\sqrt{\sum h^2}}$$

Equation 10

Following this, the spectral loadings, **V**, are calculated using the newly-defined scores and the spectral data (Equation 11).

$$V = \frac{U^{'}.X}{\sum U^2}$$

Equation 11

The loadings associated with the concentrations, **Q**, are determined (Equation 12).

$$Q = \frac{y'.U}{\sum U^2}$$

Equation 12

The product of the scores and the loading vector is subtracted from the spectra, (Equation 13), and the product of the scores and the regression coefficient is added to the initial estimate of concentrations to form the new concentration estimate (Equation 14).

$$X_{res} = X - U.V$$

Equation 13

$$y_{new} = y_i + U.Q$$

Equation 14

The residual concentration is determined by subtracting the new concentration estimate from the true concentration. The true concentration values are those generated after the actual concentration data has been scaled.

$$y_{res} = y_{true} - y_{new}$$

Equation 15

The second PLS component is found by replacing the original **X** and **y** data sets with the residual data. The process is continued until the desired number of components is extracted.[28]

The PLS method can be extended to handle several concentration terms simultaneously, which is called PLS2. This method is very similar to PLS but instead of maximising the covariance between one concentration and the linear functions of the spectra, the covariance of two linear functions (one for the concentrations and one for the spectra) is maximised. This can be advantageous for calibration purposes, but for prediction, the use of PLS to predict each concentration individually produces better results.

As measurement science and PAT continue to evolve, the methods and processes to perform analyses also evolve techniques such as Neural Networks and Ridge Regression will gain further employment. But throughout these evolutions, PLS will likely remain the standard method for analysis of the data recorded due to its simplicity and precision.

3.4.3 Model Calibrations

Measurements made in any system are essentially abstract until they are compared to other measurements from within the same system. For example, the area underneath a single peak of a GC trace means nothing on its own, and trying to relate it to a concentration or to any other quantitative factor is nearly impossible. However, if a series of GC measurements is performed from samples containing known concentrations, the resulting peak areas can be related to the corresponding concentrations. This allows for calibration, and using this comparatively with the previous abstract value, a prediction of the concentration can be made based upon the peak area, which previously had no comparative value.

This is the main aim of producing a calibration model; once constructed; a calibration model can make predictions of otherwise unknown samples.[46-48] To this end, as much of the relevant variation within the model must be extracted and

incorporated into the model, with methods such as PCA and PLS being ideal for modelling (see section 3.4.1).

The first stage of producing a calibration model is the selection of the samples that will comprise the calibration model. There are many different methods for this sample selection, such as the use of correlation between spectra for selection and the use of the PCA scores and a Euclidean method.

3.4.3.1 Selecting Samples Using Signal Correlation

This method of selecting samples uses the correlation between spectra to compose a calibration set. The most highly correlated and therefore similar spectra are chosen, and this method is ideal for selecting samples for calibration sets based upon the prediction of an unknown spectrum. The correlation between all of the calibration spectra and the unknown spectrum would be calculated and the calibration spectra that are most highly correlated with the unknown spectra are used to make the calibration model and a prediction of the unknown sample using PLS. The downside of this method is that it is not suited for larger data sets that have regions of clustering, due to factors such as differing grades of material. The signal correlation method of sample selection was successfully employed by Shenk and Westerhaus *et al.*[49-51] Their study used a correlation constraint in the selection of samples to build a calibration model from a data set of over 6500 samples. As an unknown sample was determined, the LOCAL algorithm was employed and a calibration set was defined. The results from this study showed that this method of sample selection was very successful.

3.4.3.2 Selecting Samples Using the Euclidean Distance and Principal Component Analysis

The Euclidean method of selection uses PCA scores, from which the calibration set is defined. The data from the calibration set is run through the PCA algorithm to produce the sample scores. The unknown sample is then projected into these scores, and then the Euclidean distance between the unknown sample and the entire data set is determined. The samples that have the smallest Euclidean distance are closest to the unknown sample within the scores plot, and are selected for the production of a calibration set. These selected samples are then used in conjunction with PLS to make predictions of the unknown sample. This method solves the problem of clustering due to differing grade which is encountered by the correlation method, and still retains the ability to model systems of a more traditional nature.

3.4.3.3 Selecting Samples Using the Condition Number and a Squared Covariance Matrix

The use of the condition number as a method of sample selection is akin to methods of optimality insofar as the system relies upon the minimisation of the condition number of the data matrix to determine the samples for selection. The condition number of a matrix is defined as the ratio of the first eigenvalue and the last eigenvalue. This ratio is the true condition number; however, this ratio always results in very large conditions, especially when dealing with spectral data. To reduce this and make optimisation simpler, the ratio can be altered to be the ratio of the first eigenvalue to the last significant eigenvalue.

The last significant eigenvalue can be determined using an F-Test. This dramatically reduces the magnitude of the condition number, as well as the time that

is taken for the computation to reach optimisation. Additionally, the removal of the smaller eigenvalues removes potential noise from the modelling system.

The condition number expresses the amount of variation found in each principal component. A good model, with equal variance captured for each principal component removed, will have a condition number that is very close to one. However, this can be misleading, as a data set that is entirely comprised of noise will also have equal variance captured by each principal component, and will therefore also have a condition number of one. This problem is partially addressed by only using the most significant eigenvalues that relate to the most significant principal components, and it can be further solved by using a squared covariance matrix, which is performed using scaled data (Equation 16).

$$\left(X'YY'X\right)$$

Equation 16

The squared covariance matrix can be used to remove variables from a model that may contain larger amounts of noise. This reduces the potential for 'noisy' principal components and allows the condition number of the matrix to be a true representation of the data.

3.4.4 Assessing the Model Quality

The assessment of the quality of a calibration can be determined using the root mean square error in calibration (RMSEC). The calibration samples are run predicted by the model resulting in a set of actual values, $\mathbf{y}$, and predicted values, $\hat{\mathbf{y}}$. These are used with the number of calibration samples, $\mathbf{N}$ (Equation 17).

$$RMSEC = \sqrt{\frac{\sum(\hat{y} - y)^2}{N - 1}}$$

Equation 17

This gives an indication of the lack of fit the model has to the data, which can be indicative of the quality of the final predictions. It can also be an indication of the quality of the samples selected to build the model. However, the RMSEC can be a misleading tool; for example, as each principal component is extracted and included in the model the RMSEC will decrease. The RMSEC will continue to decrease as more components are included in the model, and this can cause an over-fitting of the model.

3.4.4.1 Validation of a Model

Once a model has been constructed it must be evaluated and validated to assess its quality. The main aim of a model is to make predictions, therefore using predictions to assess the quality of a model would be the most appropriate method. There are different methods for doing this; they are separated by the amount of samples remaining after calibration. If there is a sufficient number of samples remaining after calibration a separate independent validation set of data are constructed. This is applied to the model; the subsequent predictions can be used to determine the root mean square error in prediction or RMSEP (Equation 18).

$$RMSEP = \sqrt{\frac{\sum(\hat{y} - y)^2}{N - 2}}$$

Equation 18

As with the RMSEP the actual values determined by the reference method, **y**, and the values predicted by the model, $\hat{y}$, are used with the number samples, **N**, to determine the RMSEP. The RMSEP is one of the most important methods for establishing the quality of the model, unlike the RMSEC, when using PCA as more components are included in the model the RMSEP will decrease until a point, then it will begin to rise sharply. This is over-fitting a model, and to avoid this both calibration and validation errors must be monitored. The relationship between calibration and validation is shown in Figure 12.

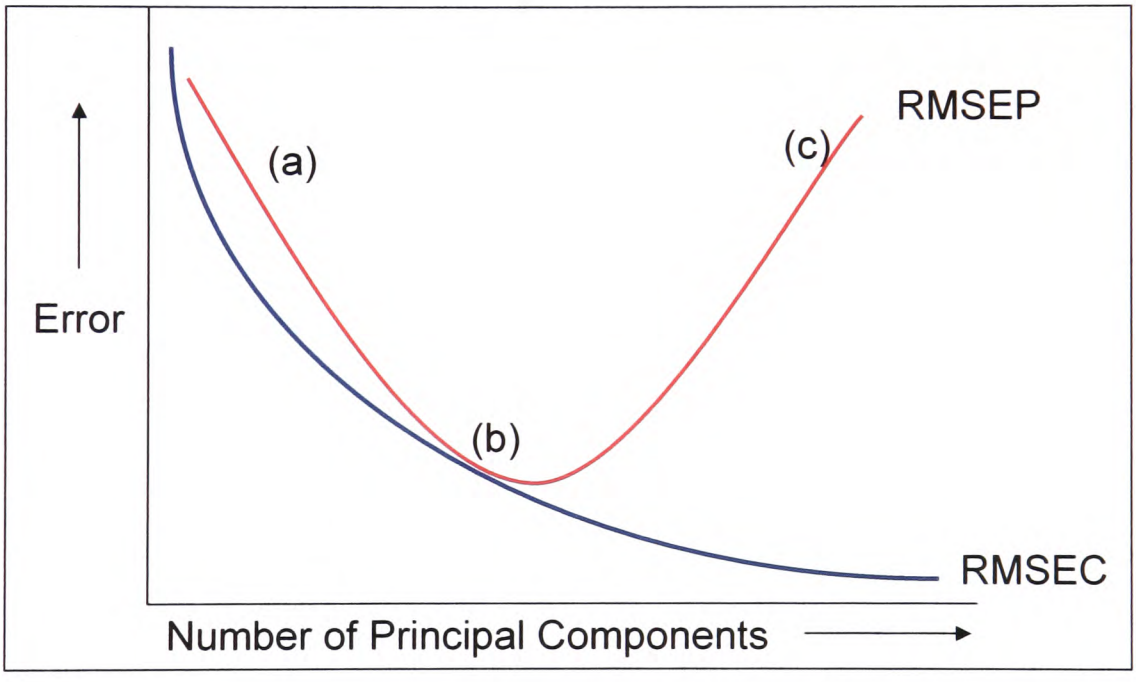

Figure 12. The relationship between the calibration (RMSEC) and prediction (RMSEP) errors.

At point (a) the model is improving until it reaches point (b). At this point the model is at its optimum performance, the ratio between RMSEC and RMSEP is at a minimum. As more components are added to the model, the RMSEC continues to fall, however, the RMSEP increases to point (c), at this time the ratio between calibration and prediction error is significantly larger than at point (b), at this position the model is over fitted.

In situations where the samples remaining after calibration are limited a system of cross validation can be employed. Cross validation is the process of removing a number of samples from the calibration set, and re-calculating the calibration error. This is redefined as the root mean square error in cross validation (RMSECV) which is calculated in the same manner as the RMSEC (Equation 17). Cross validation is only applicable when the amount of data in insufficient for production of a validation set.

3.4.4.2 Normality Plots

Normality plots are a method of determining whether the data being analysed is normally distributed. This a graphical method that plots the data value against the scaled probability of normality. If the resulting plot is one of a straight line the data can be deemed to be normally distributed. However if the data is not straight, the graphical approach can be used to look for sections of the data that are straight and thus normally distributed.

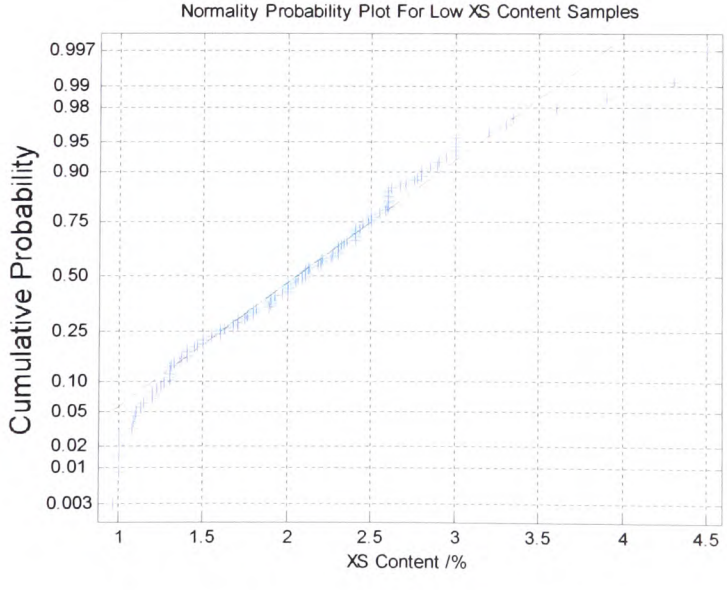

Figure 13. A typical example of a normality plot.
The data appears to be linear and therefore normally distributed.

3.5 Methods for Data Pre-Treatment

3.5.1 Scaling

Scaling methods are quick and simple ways of correcting spectra to remove baseline or magnitude effects associated with intra-variable variation.

3.5.1.1 Mean Centring

Mean centring is the process of calculating the mean spectrum, followed by subtraction of the mean from each spectrum within the data set (Equation 19).

$$x_{corr} = x - \mu$$

Equation 19

This has the effect of translating the spectra such that they are now centred on the origin. Mean centring is performed prior to any form of data reduction; mean centred scores are distributed around the origin in a similar manner to the spectra. Predictions made by models built using mean centred data are mean centred, and thus must have the mean spectrum added to them to convert them back to the appropriate data. In data sets with small intra-sample variation the effect of mean centring is negligible. However, in situation were there is a much greater amount of intra-sample variation application of mean centring results in a more significant effect upon both the scores and loadings. Mean centring has become a standard from of pre-treatment to the point were most methods will include correction by mean centring as an automatic practise, although chromatographic data is not suitable to correction using mean centring.

3.5.1.2 Auto-Scaling

Auto-scaling is a form of variance scaling that is performed down each column. Auto-scaling is a two step procedure; first mean centring is performed subtracting the mean spectrum from each sample. Following this each point on the column is divided by the standard deviation of the column (Equation 20).

$$x_{corr} = \frac{x - \mu}{\sigma_{col}}$$

Equation 20

As with mean centring auto-scaling is performed prior to any form of data reduction or modelling, predictions made by models using auto-scaled data are themselves auto-scaled. To recover the actual values multiplication by the column standard deviation and addition of the mean spectrum must be performed.

Auto-scaling is of great importance used with data that has large variations in error of signal to noise ratio when moving across from one variable to another. Use of auto-scaling reduces the skewing effects brought about by the large variable to variable magnitude effects, essentially giving each variable equal significance. However, if the data does not have large variation in the error or the signal to noise ratio use of auto-scaling can give artificial importance to noisy areas of the spectra by scaling every to unit variance. Auto-scaling has not found itself in the same company as mean centring amongst the automatic pre-treatment methods, due to the ability for it to give noise the same significance as an analytical signal.

3.5.1.3 Standard Normal Variate Transform

Standard normal variate transform (SNV) is another form of variance scaling. Like auto-scaling, the first stage of SNV is mean centring, followed by division by the standard deviation. Unlike auto-scaling, SNV uses the standard deviation of the row, scaling all rows to the same unit length (Equation 21).

$$x_{corr} = \frac{x - \mu}{\sigma_{row}}$$

Equation 21

SNV has found a niche role within applications that correct spectra for light-scattering effects due to differing path-lengths recorded when analysing diffuse powders.[52-54] More recently, SNV has been superseded by techniques such as multiplicative scatter correction (MSC) and extended multiplicative scatter correction (EMSC).

3.6 Orthogonal Signal Correction

Orthogonal signal correction (OSC) is a method that was initially developed to correct for light scatter effects but can also be used to correct more general types of interference. OSC accomplishes the correction by removing the effects and artefacts that have zero correlation with the reference value. The goal being to leave only the spectral information that directly relates to the concentration. OSC is primarily used in conjunction with NIR spectroscopy since there are regions within the NIR spectra that contain information that have little or no effect on the predictions made by a model.

OSC was first proposed by Wold *et al.* in 1998.[9] Wold showed that using OSC treated data lead to the production of models with lower RMSEP values than scatter corrected and raw unprocessed models. This meant that the OSC models predicted new samples better than the scatter correction and raw models. Further more the OSC filtered gave much simpler calibration models when compared against the raw models. Wold's results showed that OSC did indeed remove data that was not correlated with the spectral data thus making calibrations and predictions simpler and more accurate. Wold also showed that OSC was as effective with single reference values as it was with multiple references values, so correcting for more than one compounds concentration at once.

3.6.1 The Orthogonal Signal Correction Process

Using the algorithm proposed by Fearn,[55] OSC first proceeds by creating a matrix, **M**, that contains the majority of variation in the spectral data, **X**, that is not associated with the concentration data, **y**.

$$M = 1 - X'y\left(y'XX'y\right)^{-1} y'X$$

Equation 22

The next step is to multiply **M** by **X** to form **Z**, such that **ZZ'** is symmetrical.

$$Z = XM$$

Equation 23

Following this PCA is used to determine the first principal component of **ZZ'** and subsequent first eigenvalue, λ, along with the associated loading vector **V**. From these the loading weight vector, **w**, is calculated (Equation 24).

$$w = \frac{MX'V}{\sqrt{\lambda}}$$

Equation 24

Using **w,** a new scores vector is determined (Equation 25). This new scores vector is then orthogonalised to the concentrations, **y** (Equation 26).

$$U = Xw$$

Equation 25

$$U_{osc} = U - y\left(y'y\right)^{-1} y'U$$

Equation 26

The OSC scores, $\mathbf{U}_{OSC}$, are then used with the spectral data to calculate the OSC loadings, $\mathbf{V}_{OSC}$ (Equation 27).

$$V_{osc} = \frac{X^{'}U_{osc}}{U_{osc}^{'}U}$$

Equation 27

Using $\mathbf{U}_{OSC}$ and $\mathbf{V}_{OSC}$ the OSC component is determined; OSC component is then subtracted from **X** to yield the residual spectral matrix $\mathbf{X}_{res}$.

$$OSC_{COMP} = U_{osc}V_{osc}^{'}$$

Equation 28

$$X_{res} = X - OSC_{COMP}$$

Equation 29

To remove further OSC components substitute $\mathbf{X}_{res}$ with **X** at the start of the process. This process works as an 'anti-NIPALS' method, were NIPALS removes the components of greatest of greatest correlation between samples and spectra, OSC removes the components of least correlation.

Since Wold's initial publication of OSC and the subsequent follow up by Sjoblom,[56] over 200 papers have been published citing the work of Wold *et al.* In 2000, a paper was published by Fearn *et al.* that highlighted some problems with Wold's algorithm and suggested improvements.[55] Fearn stated that the current method by Wold resulted in models that could be achieved by simply taking one more PLS component when building the model. However, the improvements Fearn suggested did not result in major advancements leading Fearn to surmise that Wold's method was not the best but it is the best available. The OSC algorithm has been used for a variety of different applications ranging from the analysis of port wine[57] and the classification of coffee beans correcting for calibration transfer.[58]

3.7 Derivatisation

The use of derivatives was first proposed by Savitsky and Golay in 1964,[59] when they used an *n*th-order derivative and a polynomial to correct the analytical signal. This can have the effect of removing different baseline effects depending upon the order of the derivative. A first-order derivative can be used to correct an additive baseline. The first derivative spectrum is generated based on the gradient of each point in the analytical spectrum. The steepest point of a positively-inclined curve in the original spectrum results in a maxima in the first derivative spectrum; conversely, the point with the steepest negative inclination in the original spectrum results in a minima in the first derivative spectrum. The absolute top of each maxima and bottom of each minima are points where the curves have no gradient and are flat. In the first derivative spectrum these areas translate to points of the spectra that cross through zero on the x-axis. This procedure is illustrated in Figure 14 and Figure 15.

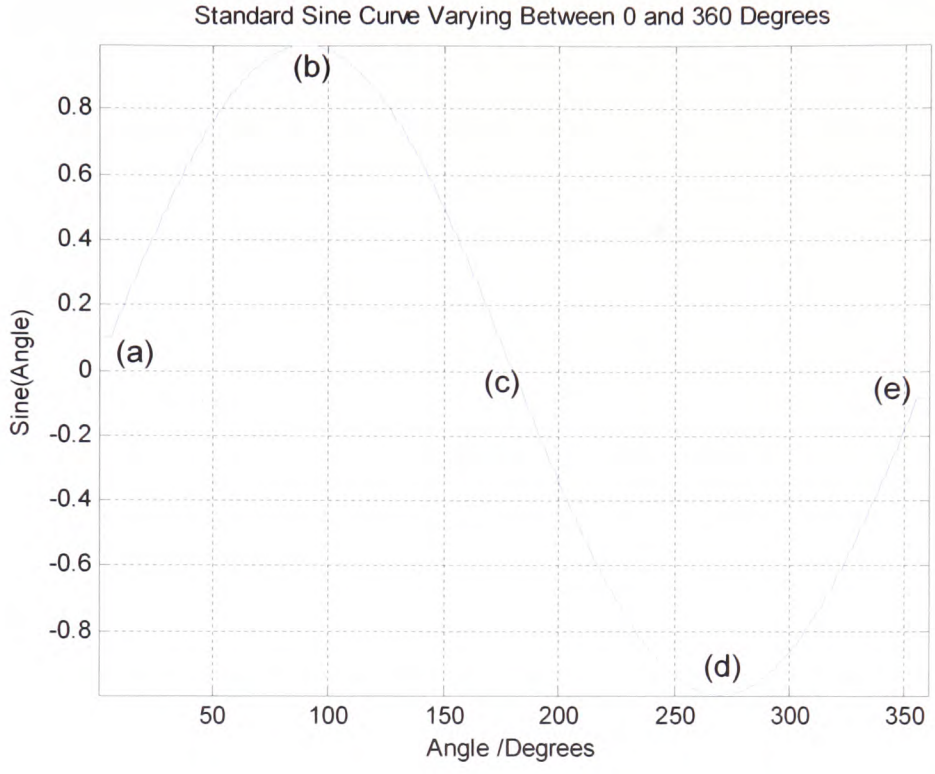

Figure 14. Standard sine curve. (a), (c), and (e) show the areas with steepest gradients, while (b) and (d) show the maxima and minima of the curve.

Figure 14 shows the original spectrum of the sine wave after it has completed one cycle. Points (a) and (e) represent the areas where there is the steepest positive gradient when the angle passed through is 0° and 360°; these points also correspond to points (a) and (e) in Figure 15, although in the second figure these points have become maximums after the application of a first-order derivative. Point (c) in Figure 14 highlights the gradient at the steepest negative inclination, 180°; this again translates to point (c) in the first derivative spectrum (Figure 15) where the steepest negative gradient has become a minimum.

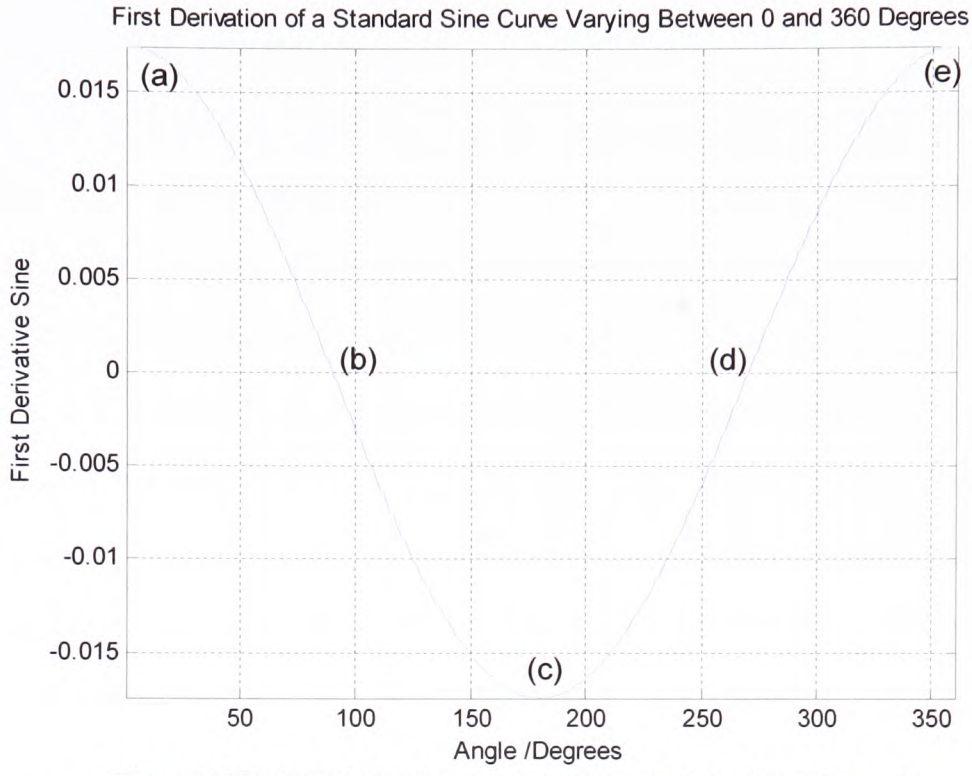

Figure 15. The first derivative spectrum of the sine wave from Figure 14. (a), (c), and (e) show the maxima and minima, while (b) and (d) cross the x-axis at zero.

In Figure 14, points (b) and (d) can be seen to be a maxima and a minima; since these points have no gradient they translate to zero points in the first derivative spectrum, shown by points (b) and (d) in Figure 15. However, Savitsky and Golay also showed that unlike the sine wave an analytical signal is not a continuous mathematical curve, and each point recorded is a discrete measurement usually taken at evenly spaced intervals, i.e. wavelengths. This creates an interesting situation when the maxima or minima falls between two points of discrete measurement. Savitsky and Golay proposed the use of a window that encapsulated a set number of points, to which a polynomial curve is fitted. Using the curve, which is continuous, derivations can be performed on the areas between points. The use of a window does require a form of optimisation to be included, as a window that is too small gives artificial significance, after derivation, to noise within the original spectra, while a window that is too large results in reduced maximums and minimums. The size of

the window must be optimised to find the right balance of the reduction of the artefacts and the increased noise.

The publication by Savitsky and Golay rapidly became one of the most widely-cited papers in the journal *Analytical Chemistry* even though the original paper contained a few typographical errors (subsequently corrected in a paper by Steiner *et al.).*[60] Over the past three decades, Savitsky-Golay smoothing and derivation has become a standard form of pre-treatment for the removal of redundant variations from spectral data, largely due to the fact that it can be applied to many different fields such as spectroscopy, biochemistry, physics, and other scientific disciplines. An added advantage is that the initial work performed by Savitsky and Golay was prior to the invention of the microprocessor; in today's modern computer age, very little effort is required to perform the calculations.

3.7.1 Multiplicative Scatter Correction

Multiplicative scatter correction (MSC) was first developed and reported by Martens and Naes[61]. It was employed as a method of correction for varying baseline effects and the variation in path-length brought about a particle-size distribution of the NIR spectra of powdered samples. The NIR signal is reflected by powdered surfaces in two ways: diffuse reflectance and specular reflectance. Diffuse reflectance occurs when the NIR signal penetrates the sample and is reflected back to the detector. Specular reflectance occurs when the NIR signal does not penetrate the sample. MSC attempts to correct for variations in both forms of reflectance by constructing an individual linear regression model for each spectrum recorded that accounts for the variations when combined with a reference spectrum. The reference spectrum is usually determined by finding the mean of the calibration spectra. The MSC procedure was superseded by Extended Multiplicative Signal Correction.

3.7.2 Extended Multiplicative Signal Correction

The extended multiplicative signal correction (EMSC) method of pre-processing allows a separation of physical light-scattering effects from chemical light absorbance effects in spectra from powders or turbid solutions. EMSC was originally designed for use with diffuse reflectance or transmittance spectroscopy, where uncontrolled variations in light scattering is often a complicating factor that can make multivariate calibration difficult. EMSC can be used to correct for multiplicative effects such as path-length variation and light-scattering effects, additive chemical effects such as analyte absorbance and interferents, as well as additive physical effects such as temperature shifts and baseline variations. The ability to correct for all three effects makes EMSC a powerful technique; however, it relies upon the assumption that each sample has a significantly different spectrum and is therefore linearly independent.[62, 63]

Martens *et al.* first reported on EMSC in 1998, but since then it has had limited application. This is mainly because EMSC was first published around the time the OSC correction was reported. EMSC has one significant disadvantage when compared with OSC, in that EMSC can only correct for one sample's concentration at a time, which is problematic in analytical systems where more than one sample is being analysed. This is not a problem with OSC. However, in 2005 Saiz-Abajo published a paper evaluating EMSC, and they reported that the use of EMSC with prior knowledge of the system produces robust models with good predictive performance.[62] They also reported that EMSC was an "interesting" method for correcting temperature deviations.[62] The ability to correct for temperature effects could be very useful since the NIR spectra are susceptible to changes in temperature.[64]

3.8 Design of Experiments

Montgomery defines design of experiments (DOE) as "a scientific approach to planning experiments such that the results will yield the most appropriate information."[65] Design of experiments is a concept that has been around for approximately seventy years and was first introduced by Fisher. Fisher was responsible for the basic guidelines for an experimental design and its implementation. Fisher implemented his designs in an agricultural context, whereas in 1951 Box and Wilson saw an application for designs within an industrial context and introduced the concept of response surfaces.[65, 66]

The late 1970s saw the inception of what was at that point a controversial chapter in DOE with the publication of work by Tagutchi. Tagutchi's studies expanded interest in the use of experimental design; however, most of the underlying science proposed by Tagutchi had not been published or reviewed by his peers. By the late 1980s, his concepts had been investigated and they were found to have been well-founded, but there were significant problems with the experimental designs and the data analysis. In the end, Tagutchi's work was not all in vain because he did encourage industries to seriously consider the employment of DOE and he increased the level of awareness and training of scientists and engineers in this field.[65]

DOE allows an investigator to produce optimisations in a multivariate manner. While traditional methods involve varying a single variable at a time, DOE employs experimental designs that allow the variation of multiple variables, thus giving the investigator information about the interactions between the variables being optimised. An interaction is the failure of a factor to produce the same effect in responses at different levels of another factor. This is a major advantage of DOE when compared to the traditional "one-at-a-time" methods. Another is the ability of

the design to restrict the experimentation so that only procedures that are statistically significant or have major interactions are performed. This is actually beneficial in two ways: when compared to the traditional method, DOE saves time (due to the performance of fewer experiments) and yields higher-quality results (as the final model does not include superfluous information, making it more pertinent and robust). These many advantages emphasise that DOE is the best method for performing process experimentation.

Montgomery set out a series of guidelines that must be employed to produce a successful design of experiments:

1. Recognition of the Problem;
2. Choice of Factors and Levels;
3. Selection of Response Variable;
4. Choice of Experimental Design;
5. Perform Experiments;
6. Analysis of Data;
7. Conclusions and Recommendations.[65]

3.8.1 Recognition of the Problem

Recognising the problem is probably the simplest part of the procedure, yet one of the most important. Defining the problem is a critical step, as subsequent decisions in the design will hinge upon this definition. Design of experiments is commonly used for optimisations, process yields,[67-69] reaction times, and conditions.[70, 71] The systematic approach of DOE makes it ideal for optimisation. In this phase of the procedure it is also important to consider the number of experiments that can be feasibly executed; for example, the definition of a problem that requires many expensive experiments could rule the design out as being financially unrealistic.

3.8.2 Choice of Factors and Levels

A factor is the variable being changed through the design of experiments, examples of which are temperature, mixing times, or reagent concentrations. The levels are the values by which the factors will be tested, i.e. the differing concentrations of samples. The range of factors is the spread or difference between the highest and lowest levels.[65]

This section of the design procedure must be completed using prior knowledge of the system. An investigator has to know which factors are important and will impact the final optimisation. These factors must be orthogonal so that all factors can be varied at once. Levels must be reset using knowledge of the system; for example, in an enzymatic system an investigator must know at what temperature an enzyme is denatured, and set the levels accordingly. Levels set beyond the threshold will result in the destruction of the enzyme. This is a very important step of the design process, as the selection of the wrong factors or inappropriate levels will result in a poor design.

As a further note, the stage in which design of experiments is employed can determine the nature of the input data and thus the available factors. If DOE is intended to optimise a process, the factors could range from the typical reaction parameters outlined in the previous two paragraphs. However, if DOE is employed after experimentation, the input data can change to the spectra collected, the number of PCs within the model, or the PCA scores and loadings. This change in factors also changes the levels employed.

3.8.3 Response Variables

This stage requires selection of response variables that will be used to determine the quality of the experiment performed, such as prediction error, material yield, or peak resolutions. Regardless of which variable is selected it must provide the most suitable information to assess the efficiency of the design. This relates back to definition of the problem as an accurate definition should make selection of the response variable simple. However, in some situations there can be more than one form of response, and thus selection of the response that will be the most accessible and yield the most information is paramount.

These first three stages of design will always be initiated prior to the start of any experimentation.

3.8.4 Choice of Experimental Design

The experimental design defines and outlines the experiments to be performed as part of the DOE process. There are all different sorts possible of designs, including full and partial factorials and optimal designs.

3.8.4.1 Factorials

3.8.4.1.1 Full Factorials

Factorial designs allow the examination of two or more factors. A factorial design relies on experiments being performed at every combination of factors and levels. This is a very systematic process that thoroughly maps a data space, and it will produce an optimal solution as long as the correct factors and levels were selected. The downside of full factorial designs is that a large number of experiments must be performed. The number of experiments is determined by Equation 30.

$$n^k$$

Equation 30

In Equation 30, **n** is the number of levels in the design and ***k*** is the number of factors being investigated. So in a design with five factors at two levels, 2^5 or 32 experiments must be performed.[65]

To alleviate the excessive amount of experimentation in this method, the use of partial or fractional factorials was proposed.

3.8.4.1.2 Partial Factorials

In the previous design of five factors at two levels, there are 15 experiments that involve the individual factors and two component interactions, with the remaining 17 experiments contained three, four, or five component interactions. An investigator employing a partial factorial would only investigate the one and two component interactions, as the higher-order interactions would yield little additional information. In this case, the number of experiments performed would be reduced by over half, from 32 to 15, saving both money and time. Other advantages of the partial factorial method is the ability to project results into a larger design, and to use the partial factorial design as a subset of a larger set of designs; this makes it ideal for screening experiments.[72, 73] However, partial factorials do have some disadvantages. By removing higher-order interactions the data spaces are not mapped as thoroughly as occurs with full factorial method. This is not an issue when employing partial factorials for the screening process. A partial factorial would be used to determine the important factors in an optimisation; then, using this information, a full factorial would be implemented that focuses on the areas highlighted by the partial factorial.[65]

3.8.5 Optimal Designs

Optimal designs determine points for experimentation based on the maximisation or minimisation of a specific design criterion. Optimal designs have two main applications, calibration and sampling. They can be employed either before or after experimentation, i.e. prior to experimentation to determine the best experiments to perform, or after the use of a full factorial design experiment to determine which information to include in a model. Optimal designs can significantly reduce the number of experiments performed; however they can be less systematic than factorial designs. Optimal designs only test sample points that have significant interactions, with the significance of the interactions determined by the design criterion, e.g. a D-optimal design will study the interactions that are at the extremities of a system. For the previous five factors/two levels design, a D-optimal approach would require sixteen experiments to be performed. There are many different types of optimal designs including D-optimal and A-optimal; the key variation between each optimal design is the design criterion.[65, 74]

3.8.5.1 D-Optimal Designs

D-optimal designs are possibly the most popular designs used in scientific research, ranging from chemistry to psychology. They were introduced by Kiefer in 1959,[74] but gained greater clout with the adoption of computer-generated designs executing fast computation of design criteria. The D-optimal algorithm has the effect of selecting the sample points that surround the edge of the data space. The samples within the data space do not add new information to the model when compared to the D-optimal points. The ability to select samples from the edge of the data space makes D-optimal designs ideal for producing designs based on irregularly-shaped

data spaces.[65, 72, 75] A design is determined to be D-optimal if it minimizes the determinant of the assessment data (Equation 31).

$$\left|\left(X'X\right)^{-1}\right|$$

Equation 31

The modelling procedure begins by removing a sample and re-calculating the D-optimal criterion. If the D-optimal criterion has improved, the sample remains excluded from **X** and the next sample is then removed for a re-calculation. If removal of the sample causes the D-optimal criterion to worsen, the sample is replaced and the iterative sequence again moves on to the next sample. This procedure continues until all samples have been removed and tested.

Models produced using D-optimal criteria are thought to be less robust when one or more the variables within the model contain more variation than the other variables.[74] In these situations, the use other forms of optimal designs (such as A-optimal or E-optimal) would yield better results.

3.8.5.2 A-Optimal Designs

The A-optimal design uses the variation from within the regression coefficients. The criterion used in this design is shown in Equation 32.

$$\sum diag\left(X'X\right)^{-1}$$

Equation 32

This is the sum of the diagonal (or trace, tr) of the inverted square matrix **X'X**.[65, 74] As with the D-optimal approach, the A-optimality criterion can be applied

in an iterative series that tests the samples individually and removes or returns them to the model depending upon the nature of the optimality criterion.

3.8.5.3 E-Optimal Designs

As with the previous optimal designs, E-optimal design relies on the optimisation of a specific criterion, and in this case a design is said to be E-optimal when the minimum value of the largest eigenvalue from the inverted matrix of (**X'X**) is determined:

$$\lambda_{\max}(X'X)^{-1}$$

Equation 33

E-optimal designs can be employed with the use of subsets (Figure 16).

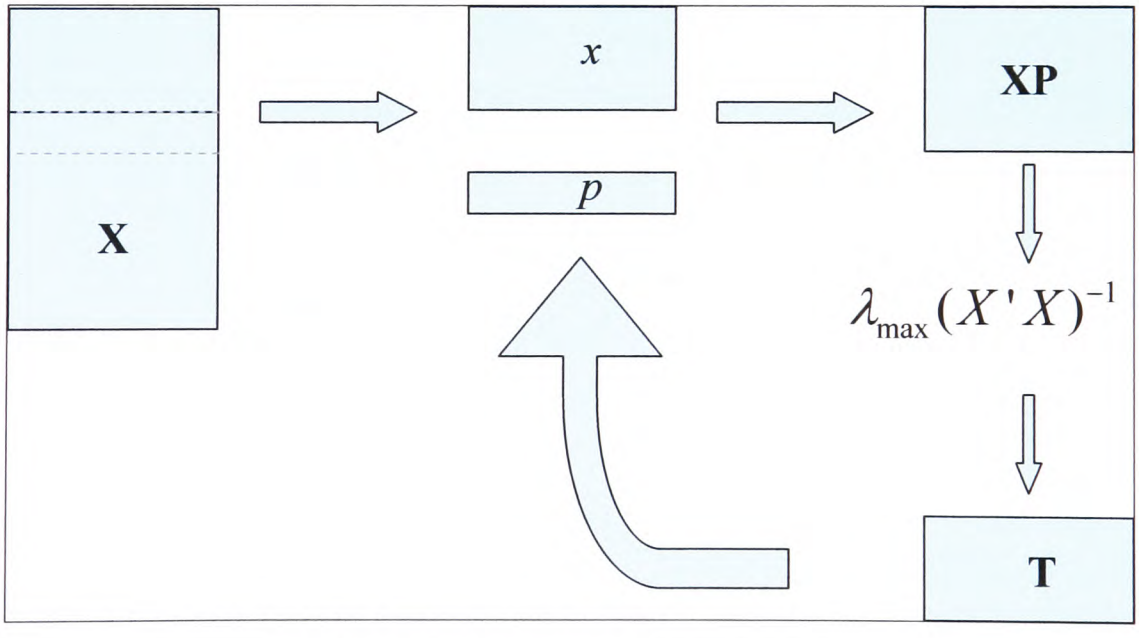

Figure 16. E-optimal procedure using subset analysis.

The E-optimal procedure begins by randomising a data matrix, **X**. From **X,** a number of test samples of size *m* are selected to form a subset, *x*. From **X**, another subset of samples is selected and forms the optimisation set *p*. These two subsets are

then added together to form **XP**. **XP** undergoes the E-optimality iterative sequence of sample removal and PCA of the remaining matrix to determine the eigenvalues. If the optimal criterion improves, the sample is excluded from **XP** and the next sample is tested; this is the same as in the D-optimal approach. But, in this case, the iterative sequence loops around until m number of samples remain within **XP**. The new matrix, **T**, then replaces x and the new optimisation subset, p, is removed from **X.**

This application of E-optimal modelling makes it ideal for maintaining a calibration model by restricting the samples within a model to a fixed number whilst ensuring that the resulting data set is optimal and contains as much pertinent information to the model as possible. The new subset, p, are samples that could be potentially added to the calibration model if they would improve the quality of the predictions made.

4 Experimental

4.1 Materials and Methods

4.1.1 Equipment

The modelling, data processing, and programming were performed on a Dell Dimension GX60 PC with a 2.00GHz Dual Core Processor and 2.00GB of RAM. The designed optimisations of the pre-processing were preformed using a PC with a 2.80GHz Celeron processor and 1.00GB of RAM.

4.1.2 Data Processing and Software Development

All data received was transferred into MatLab 7.1, published by Mathworks, Inc. (Natick, Massachusetts, USA), for analysis and treatment. The in-house software was written with MatLab Editor 7.5, also from Mathworks, Inc. Routines from the PLS Toolbox 3.5, published by Eigenvector Technologies (Manson, Washington, USA), were used in the construction of the PLS models.

4.1.3 Adaptive Sample Selection Algorithms

4.1.3.1 The Euclidean Distance Algorithm

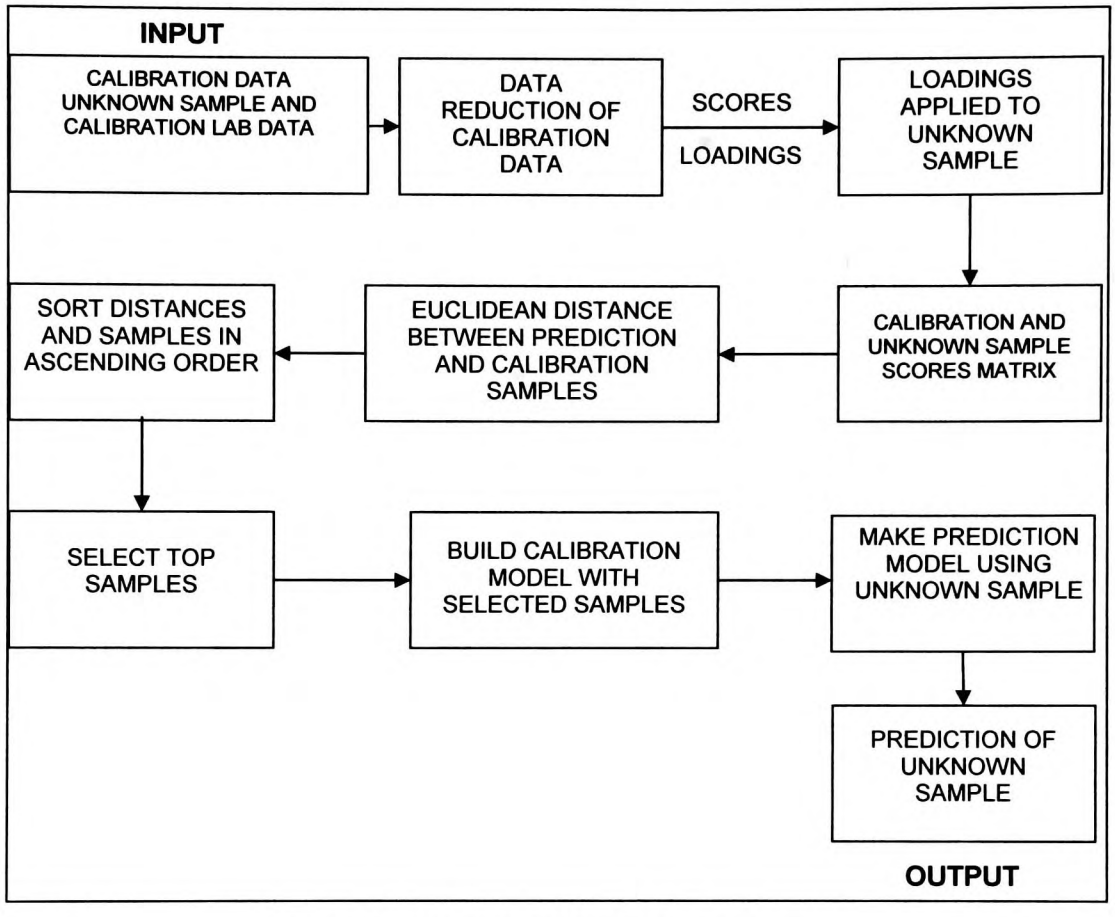

Figure 17. Adaptive sample selection algorithm.
The samples were selected for calibration using PCA scores and Euclidean distance.

The aim of this programme was to perform sample selection based on the Euclidean distance in the scores space of the input data (***td_adapt2.m,*** section 10.1.1.1). The calibration data undergo data reduction to produce scores and loadings. The loadings are then applied to the prediction data to produce the prediction scores. The Euclidean distance between the prediction scores and the scores of all the points of the calibration scores were then calculated. The distances were then ranked in ascending order, with the calibration samples with the smallest Euclidean distance selected; at the same time, the respective lab values were selected

and used to produce a calibration model. The lab values are used as the reference information, y, for the PLS calibration model. The calibration model was then used to predict the unknown sample. The output from this programme was the lab value of the unknown value.

4.1.3.2 The Shenk and Westerhaus Algorithm

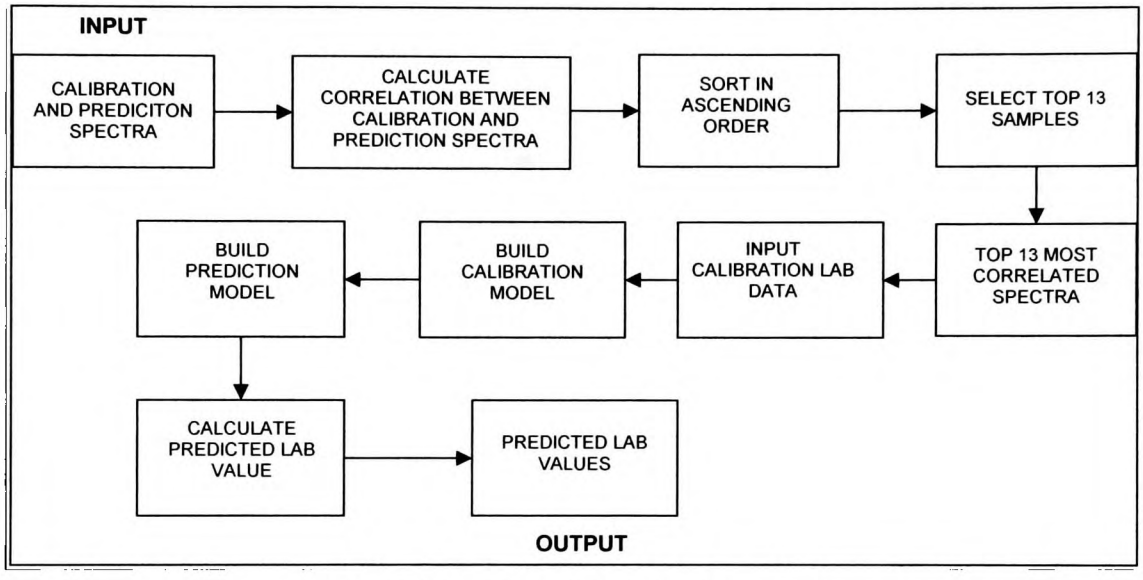

Figure 18. Sample selection algorithm.
The samples were selected based on the correlation of calibration to prediction data.

Figure 18 shows the programme used to select calibration samples based on the correlation between the prediction spectrum and the calibration spectrum.[49-51] The figure shows the process employed in the programme ***td1.m*** (Section 10.1.1.2). The inputs for the correlation selection method were the calibration spectrum and the prediction spectrum. The correlation between the prediction spectrum and all of the calibration spectra were determined. These were ranked in ascending order, with the top-correlated samples being selected. At this point, the lab calibration data was inserted into the system, and this information along with the most-correlated spectra was used to build a calibration model from which a prediction model was produced. The prediction model was used to produce a lab value for the prediction spectra. The

output for this programme was the predicted lab values based on the prediction spectra.

4.1.3.3 The Condition Number Selection Algorithm

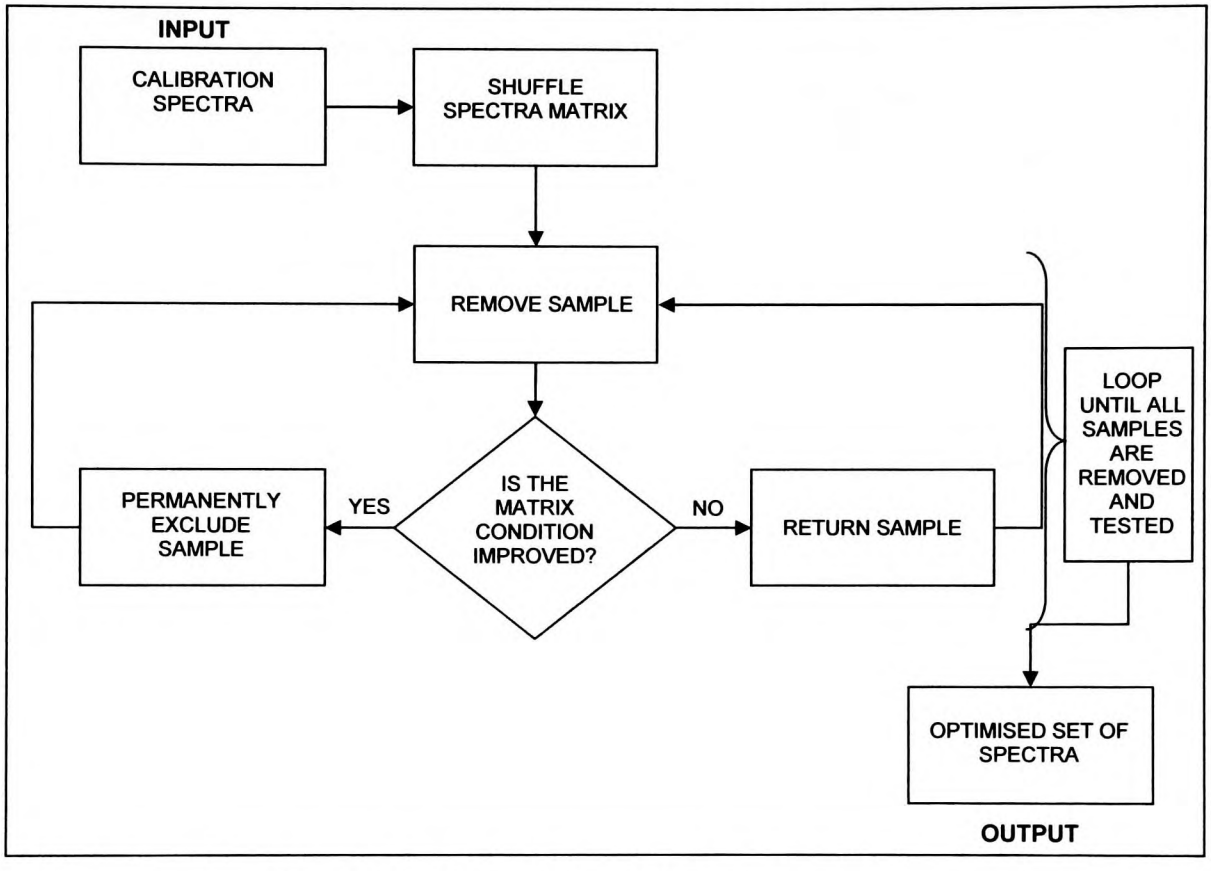

Figure 19. Condition number selection algorithm.
The samples were selected for calibration based on their improvement of the matrix condition.

The condition number algorithm (Figure 19) was used to produce a data set that has been optimised such that the final matrix has the lowest condition number. Input into the programme was the calibration spectra. The spectra were shuffled so that the starting point was randomised. From here a sample was removed and the condition of the matrix was calculated. If the removal of the spectrum improved the condition of the remaining spectral matrix the sample was permanently excluded from the calibration set. However, if the condition of the matrix worsened the spectra was returned to the calibration set. After determining exclusion or inclusion, the next

sample was removed and the condition was calculated again. This sequence of spectra removal, testing of condition, and inclusion or exclusion continued until all the spectra were tested.

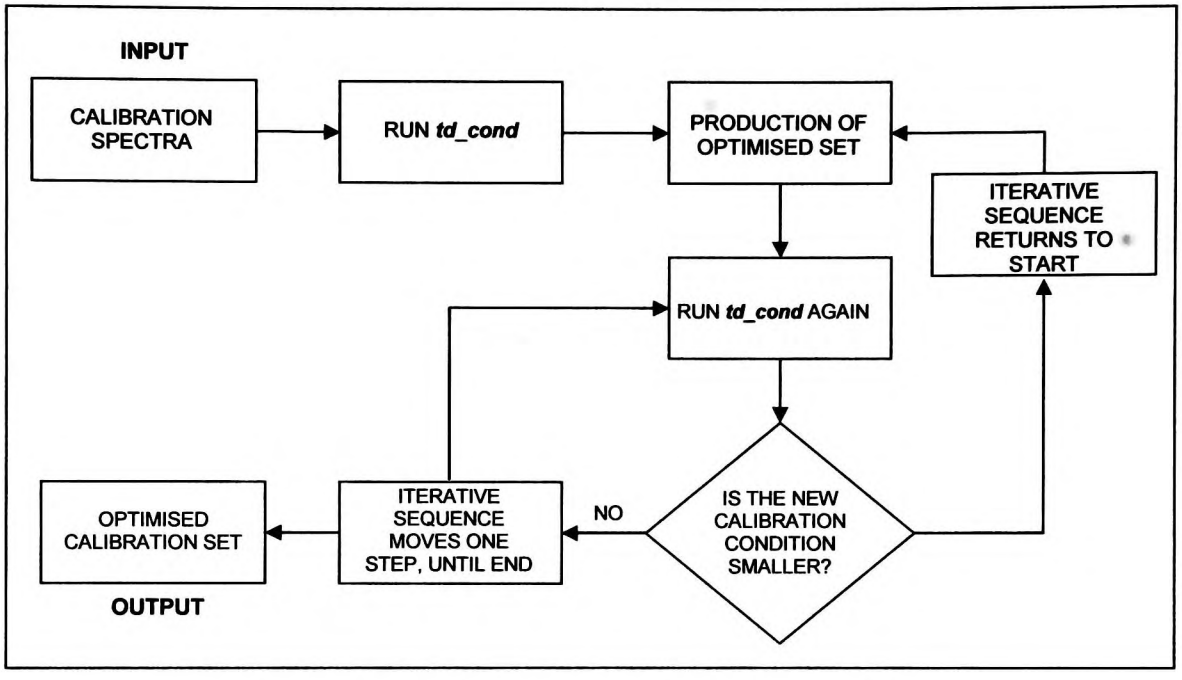

Figure 20. Iterative condition number sample selection algorithm.

The function in Figure 20 was used to produce the optimal sample set by using an iterative procedure to ensure that the calibration set selected has the lowest condition number. The function began with the input of the calibration spectra and the number of iterations to perform. The iterative procedure began with the production of another calibration set using the function in Figure 19. The two calibration sets were compared, and if the second calibration set had a higher condition number than the first set produced, another iteration was processed; however, if the second calibration set had a lower condition number, it replaced the original calibration set and the iteration count was set back to one. The loop continued until there were a successive number of calibration sets produced with a higher condition number than the original. The number of iterations was defined as an input term.

4.2 Polymer Study

A series of reference low-resolution Nuclear Magnetic Resonance Free Induced Decay (NMR FID) spectra were collected over a period of ten months starting in June 2006. The spectra were taken using the MM2720 Industrial Magnetic Resonance Solution (Progression, Inc.; North Andover, Massachusetts, USA). Reference spectra were recorded every ten minutes; however, the matching reference measurement, Xylene Soluble content (XS), was scheduled every eight hours. Any incomplete data was excluded from data processing along with the relevant reference spectra. This meant that the primary data set contained 233 reference NMR FID spectra, with 233 XS measurements within the reference data set.

4.2.1 Initial Examination

The initial examination begins by investigating the distribution of the laboratory measured values of the XS content, XS, using normality plots and histograms. Following this PCA is performed to produce the scores relating to the NMR FIDs.

Subsequently the FID and laboratory values were split in accordance with their XS content. Samples with an XS content greater than 6% form FID_H and XS_H. The samples with XS content less than 6% were used to produce the data sets FID_L and XS_L. Using normality plots and histograms the distributions of XS_H and XS_L were tested. The variables created as part of this examination are outlined in Table 1.

Table 1. The variables created as part of the initial examination.

Variable Name	Description
FID	The entire NMR FID data set.
XS	The laboratory measured value for XS percentage with in the polymer
FID_H	The NMR FIDs that pertain to samples that have a lab determined XS value **greater than 6%**
FID_L	The NMR FIDs that pertain to samples that have a lab determined XS value **lesso than 6%**
XS_H	The samples with a laboratory XS percentage **greater than 6%**
XS_L	The samples with a laboratory XS percentage **less than 6%**

4.2.2 Current Model

This modelling procedure mimicked the current model used online to make predictions of the XS content of the polymers analysed. It was important to recreate this model as a comparison for the further models produced. This method began with FID and XS being randomly split in the ratio of 4:1 to form the calibration and prediction subsets FID_{CAL}, XS_{CAL}, FID_{PRED} and XS_{PRED}. A full factorial design was employed to determine the best method of pre-processing of FID_{CAL} and XS_{CAL}. The numbers of factors and levels used in the design are outlined in Table 2.

Table 2. The factors and levels used in the design for optimisation of the pre-processing of calibration models.

Factor	Level
Regression Method	1. PLS
	2. PCR
Scaling	1. No Scaling
	2. Mean Centring
	3. Auto-scaling
	4. Standard Normal Variates
Orthogonal Signal Correction	1. No OSC
	2. OSC Component
	3. OSC Components
	4. OSC Components
Savitsky-Golay Derivatisation and Smoothing	1. No Smoothing or Derivatisation
	2. 1^{st} Derivative
	3. 2^{nd} Derivative
	4. 1^{st} Order Polynomial
	5. 2^{nd} Order Polynomial
Latent Variables	1. One
	2. Two
	3. Three
	4. Four
	5. Five
	6. Six
	7. Seven
	8. Eight

Using the results from the design, the pre-processed FID_{CAL} and XS_{CAL} were used to build a calibration a PLS calibration model, containing four latent variables. From this, the RMSEC was determined. Using this model and the appropriately pre-processed FID_{PRED}, predictions of the XS content of the samples whose spectra are contained within FID_{PRED} were made. The predicted XS content was then compared to the laboratory determined values within XS_{PRED} to determine the RMSEP.

4.2.3 Local Models

The initial examination demonstrated the potential of using local models due to the data's distribution and graded nature. The aim of this part of the study was to produce models that could benefit from the multi-modal, grade-based distribution of the FID information. To this end, the variables with an XS content on either side of 6% (Table 1) were split into calibration and prediction subsets (Table 3).

Table 3. Variables created as part of the local modelling procedure. The data is split in accordance with the XS content.

Variable Name	Description
FID_{H_CAL}	The NMR FIDs that pertain to **calibration** samples that have a lab determined XS value **greater than 6%**
FID_{H_PRED}	The NMR FIDs that pertain to **prediction** samples that have a lab determined XS value **greater than 6%**
FID_{L_CAL}	The NMR FIDs that pertain to **calibration** samples that have a lab determined XS value **less than 6%**
FID_{L_PRED}	The NMR FIDs that pertain to **prediction** samples that have a lab determined XS value **less than 6%**
XS_{H_CAL}	The **calibration** samples with a laboratory XS percentage **greater than 6%**
XS_{H_PRED}	The **prediction** samples with a laboratory XS percentage **greater than 6%**
XS_{L_CAL}	The **calibration** samples with a laboratory XS percentage **less than 6%**
XS_{L_PRED}	The **prediction** samples with a laboratory XS percentage **less than 6%**

Pre-processing was determined using a full factorial design using the same factors and levels as in the previous design (Table 2).

The pre-processed variables were then used to build PLS calibration and prediction models. The RMSEC for each calibration models were determined. Predictions were made for the respective prediction set and these were compared to the measured values to produce the model RMSEP.

4.2.4 Adaptive Selection Models

Construction of the local model highlighted several key advantages to building models that used the multi-modal nature of the data to make better predictions. To this end, the development of sample selection routines that selected the appropriate samples for calibration based the FID information were investigated. This aimed to combine the advantages of using local models for better predictions and the global models for their ease of classification.

Three forms of sample selection were investigated with regard to the production of adaptive models; the data used in all three models is shown in Table 4.

Table 4. The variables used as part of the adaptive sampling experiments.

Variable Name	Description
FID_{CAL}	The NMR FID **calibration** set.
FID_{PRED}	The NMR FID **prediction** set.
XS_{CAL}	The **calibration** laboratory measured XS percentages
XS_{PRED}	The **prediction** laboratory measured XS percentages

4.2.4.1 Sample Selection Using the Euclidean Distance

The calibration data was first auto-scaled, and following this the validation spectra were scaled individually in accordance with the calibration. This formed the input for the Euclidean distance algorithm (Figure 17). Output from the Euclidean distance algorithm was the value for the XS content as predicted by the PLS model. Using XS_{PRED} with the values output from the Euclidean distance algorithm the RMSEP for this model was determined.

Within the Euclidean distance algorithm is a series of pre-processing methods as part of the modelling stage. The method of pre-processing was determined using a full factorial design, and the factors and levels used are outlined in Table 2. A second full factorial design was employed to determine the optimum number of samples and latent variables to be used when generating the PLS model, and the factors and levels are shown in Table 5.

Table 5. The factors and levels used to optimise the number of samples and latent variables to be included in the PLS models.

Factor	High Level	Low Level
Number of Samples	25	5
Latent Variables	6	2

4.2.4.2 Sample Selection Using Spectral Correlation

As a means of comparison, the approach of Shenk and Westerhaus using correlation between prediction and calibration spectra[49-51] was also applied to the calibration and validation data. Using this method, sample selection is performed by calculating the correlation between the calibration sample and validation sample, so

that the calibration samples that are most highly correlated with the validation sample are selected to build the calibration model.

With the variables described in Table 4, the Shenk and Westerhaus programme (Figure 18) was initiated using the calibration and prediction spectra as the inputs. Output from this programme were the values of the XS content as determined from the PLS models and the samples selected for calibration. The RMSEP for these models was calculated by using XS_{PRED} and the values output from the correlation selection algorithm. Again, the best scheme of pre-processing used with the PLS models was found using a full factorial design, the levels and factors of which are shown in Table 2 (p. 80). As with the previous method, the number of samples and latent variables to be included in the PLS models was defined using a full factorial design, and the factors and levels of the design are given in Table 5.

4.2.4.3 Selection Using the Condition of the Matrix

As with previous methods the calibration spectra are outlined in Table 4, and these were input into the programme from Figure 19. The condition selection method produced a calibration set of spectra with the smallest condition number, and this was determined to be the optimum using the condition optimisation method (Figure 20). The use of the condition selection method and the condition optimisation method produced a finalised optimal set of spectra with the lowest condition number. This optimal set was then used to predict the XS content of the prediction spectra, FID_{PRED}, using PLS. The pre-processing involved in the modelling stage was determined using DOE and a full factorial design (Table 5). The RMSEP was then calculated using the values predicted by the PLS model and XS_{PRED}.

4.2.4.4 Random Selection

As a control and comparison a series of models were built using a random selection of samples. Selecting samples at random provides a control method by which proof that the methods by which samples are selected are important. The first model used the optimal methods of pre-processing and modelling parameters (number of latent variables and samples included in the model) as determined for the model using the Euclidean distance as selection criteria. Using these parameters and pre-processing PLS calibration and prediction models were constructed; the values of the XS content predicted by the PLS model were used with XS_{PRED} to determine the RMSEP of the model.

The second model used random sample selection with the pre-processing and modelling parameters determined to be optimal for the model using correlation as the selection criteria. Using the pre-processed calibration spectra, PLS calibration and prediction models were produced. The output from these models was used along with XS_{PRED} to determine the model RMSEP.

4.2.5 Implementation of the Online User Interface

The final stage of the NMR study involved the production of a Graphic User Interface (GUI) that could be employed online at the point of analysis. The development of the GUI involved many iterations and refinements. Feedback from the plant engineers was used to refine and alter the GUI so that it became fit for purpose. Also as part of the implementation the XS reference measurements were performed to determine the time frame and reliability of the reference measurements.

The final iteration of the GUI was implemented on the process NMR based on the polypropylene reactor PP5 at the Borealis facility in Schwechat, Austria. The

GUI was deployed on the instrument making predictions continuously of the XS content of the polymer pellets being produced. The prediction errors for the online GUI were recorded and compared to that of the online model.

4.3 Pharmaceutical Tablet Study

This body of work involved the analysis of NIR spectra of a series of tablets. The spectra were collected a period of three years across four different processing campaigns. The NIR spectra were recorded at the final stage of packaging, the tablets analysed are removed from the production line to record the laboratory reference information, tablet thickness, tablet weight, and active pharmaceutical ingredient. The thickness and weight were recorded using standard methods, and the active pharmaceutical ingredient (API) was determined using high performance liquid chromatography.

The experiments performed as part of this study were split into three sections each relating to a particular property of the tablet being examined: the API, the tablet weight, and the tablet thickness.

The variables used as part of this study are shown in Table 6.

Table 6. Variables used in the examination of pharmaceutical data.

Variable Name	Description
SPT	NIR absorbance spectra from the tablets.
API	The API content of the tablets as assessed by HPLC.
THK	The thickness of the tablets.
WGT	The weight of the tablets.

4.3.1 Modelling the Active Pharmaceutical Ingredient

The initial examination of this data began by splitting of the data into calibration and prediction sets (Table 7). Then normality plots and histograms were produced to assess the normality of the API distribution.

Table 7. Variables created for the modelling of the API content.

Variable Name	Description
SPT_{CAL}	NIR absorbance **calibration** spectra from the tablets.
SPT_{PRED}	NIR absorbance **prediction** spectra from the tablets.
API_{CAL}	The **calibration** set of the API content of the tablets as assessed by HPLC.
API_{PRED}	The **prediction** set of the API content of the tablets as assessed by HPLC.

Following this a procedure of variable selection was employed using the cross correlation matrix. After variable removal, sample selection was performed using SPT_{CAL} to generate a calibration set. Three methods of sample selection were investigated: selection using the Euclidean distance (see section 3.4.3.2), the correlation between calibration and prediction spectra (see section 3.4.3.1), and the condition number (see section 3.4.3.3). The Euclidean distance and correlation sample selection algorithms (Figure 17, Figure 18), and the condition optimisation method (Figure 20), were employed to generate the calibration sets for investigation. The best method of sample selection was determined to produce a calibration set, using this set and the respective samples from API_{CAL} underwent EMSC to produce a corrected set of spectra. Then the corrected calibration spectra and the respective samples of API_{CAL} were used in a full factorial design to determine the best method of pre-processing to be applied to the data prior to building a PLS model. Using the results from the design, the corrected calibration spectra were pre-processed with API_{CAL} and used to build a PLS calibration model. SPT_{PRED} and API_{PRED} were appropriately scaled and used to build a PLS prediction model to calculate the API content of the tablets associated with the spectra in SPT_{PRED}. The predicted API values were compared to the values in API_{PRED} to produce the model's RMSEP.

4.3.2 Modelling the Tablet Weight

The procedure employed with this step was the same as that for the tablet API (see section 4.3.1). The variables created and used are shown in Table 8.

Table 8. Variables created as part of the modelling of the tablet weight.

Variable Name	Description
SPT_{CAL}	NIR absorbance **calibration** spectra from the tablets.
SPT_{PRED}	NIR absorbance **prediction** spectra from the tablets.
WGT_{CAL}	The **calibration** set of weights of the tablets.
WGT_{PRED}	The **prediction** set of the weights of the tablets.

As with the API study, this procedure began by investigating the nature of the tablet weight. This was performed by producing normality plots and histograms of WGT (Table 6). Taking these results into account, the variable selection scheme was applied using the cross correlation matrix (see section 3.4.4.1) to decide which variables should be retained. As with the API modelling, variable selection was followed by sample selection; again, as with the API modelling, three methods of selecting samples (based on the Euclidean distance, spectral correlation, and condition of the matrix) were used. From this a calibration set of spectra was defined, SPT_{CAL}, and used in conjunction with a full factorial design to determine the optimal method of pre-processing the spectra. Then the processed calibration spectra and associated tablet weights were used to build a PLS calibration model. The prediction spectra SPT_{PRED} were scaled in accordance with the calibration pre-processing and used in conjunction with the calibration model to produce a PLS prediction model. This yielded predictions of the tablet weights that were compared to the weights in WGT_{PRED} to calculate the RMSEP.

4.3.3 Modelling the Tablet Thickness

As with the procedures outlined in sections 4.3.1 and 4.3.2, variable selection using the cross correlation matrix and sample selection using the Euclidean distance-based algorithm (Figure 17), the correlation based selection algorithm (Figure 18), and the condition number based algorithm (Figure 20) were used to generate a set of samples for calibration (Table 9) from which the PLS calibration model was produced. The best method of pre-processing involved in calibration was determined using a full factorial design, and the factors and levels are displayed in Table 5.

Table 9. Variables used as part of the modelling of the tablet thickness.

Variable Name	Description
SPT_{CAL}	NIR absorbance **calibration** spectra from the tablets.
SPT_{PRED}	NIR absorbance **prediction** spectra from the tablets.
THK_{CAL}	The **calibration** set of the thickness of the tablets.
THK_{PRED}	The **prediction** set of the thickness of the tablets.

Using this, the prediction information was scaled and used to produce a PLS prediction model. The values generated by the PLS model were compared to the values contained in THK_{PRED} to determine the model RMSEP.

5 Results and Discussion

5.1 Polymer Study

5.1.1 Initial Study

Initial assessment began by examining the FID spectra of the polypropylene powder (Figure 21). Each spectrum was recorded over a period of 2000 seconds. The decay curves appear to contain a large degree of variation, but the average intra-sample correlation was calculated to be 97%. This meant that all the spectra were highly correlated and that methods of data reduction were required in order to break the correlation and build the prediction models.

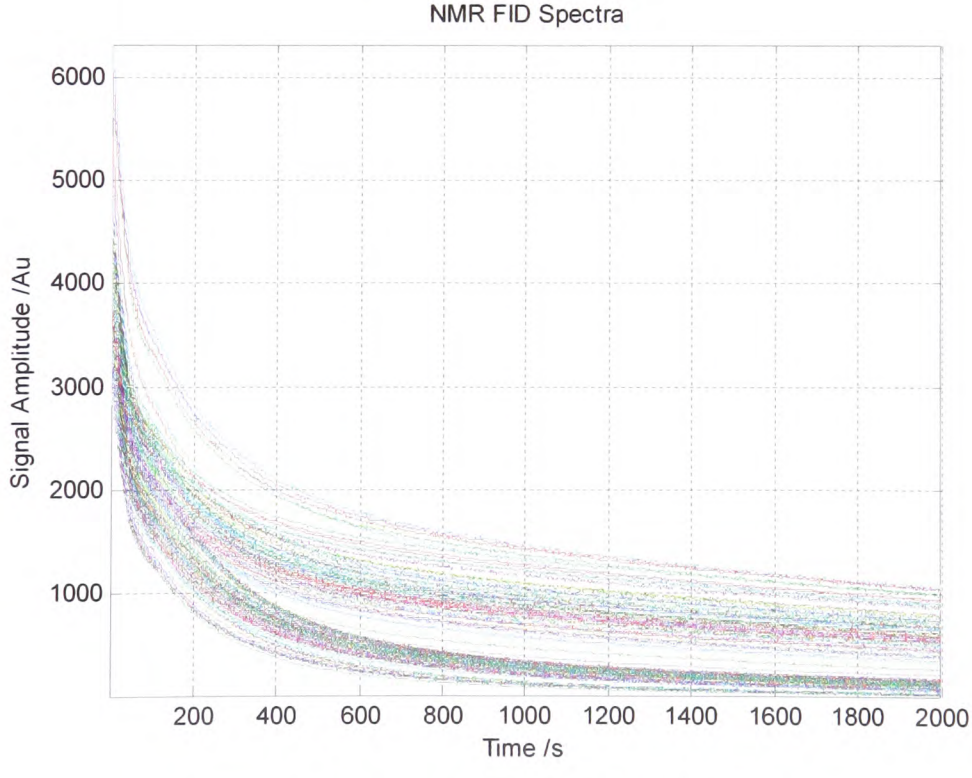

Figure 21. The collected NMR FID spectra of polypropylene.

Data reduction was performed using PCA of the auto-scaled NMR FIDs (Figure 22).

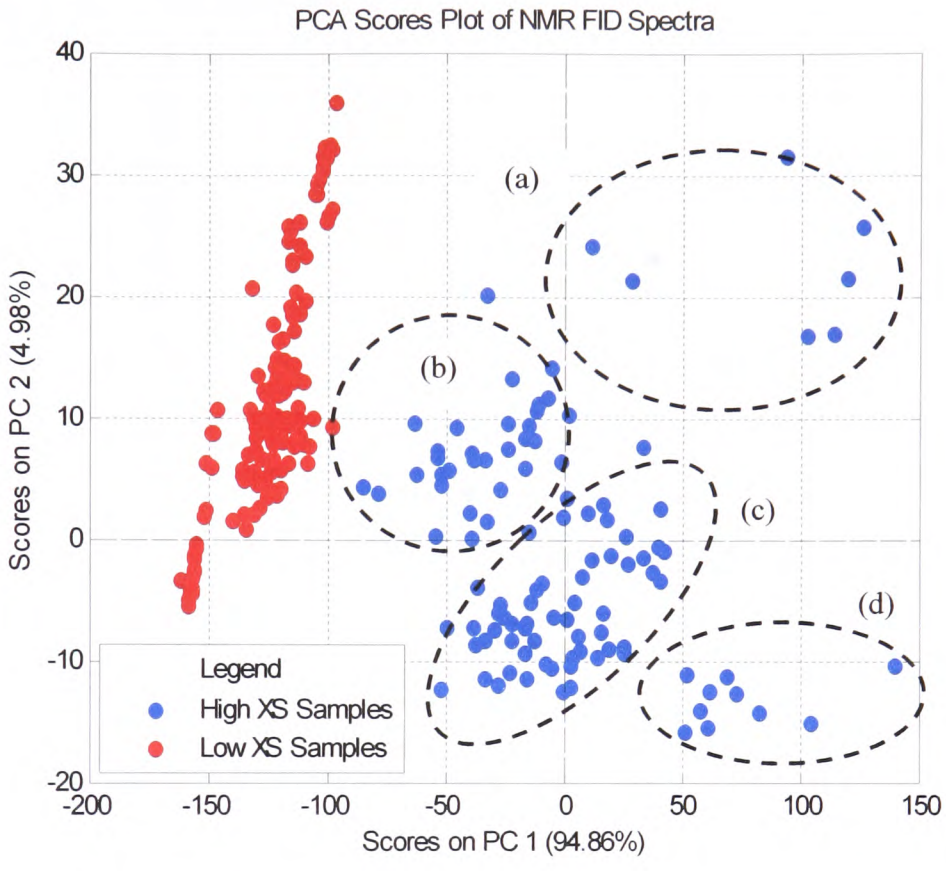

Figure 22. PCA scores of auto-scaled NMR FIDs showing the splitting of samples with low XS content (red) and high XS content (blue).

Figure 22 clearly shows two distinct clusters. The samples with a low Xylene Soluble (XS) content (marked in red) formed a tight cluster, with most of the variation contained on PC2. The samples with a high XS content (marked in blue) showed a higher degree of variation, and could be further arbitrarily split up into four separate clusters (labelled (a), (b), (c) and (d). The XS content of the polymers varies for different polymer grades, so from Figure 22 one could conclude that there were two main grades in production. However, it can be seen that this was not the case when compared to the histogram that denotes the distribution of the complete set of reference measurements regarding the XS content (Figure 23).

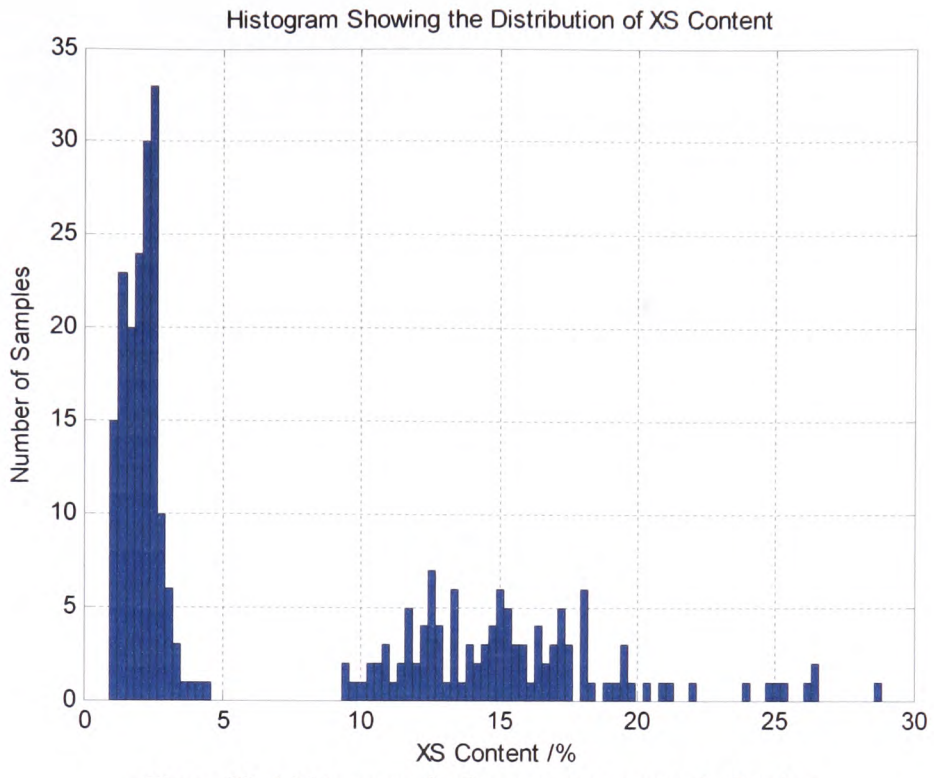

Figure 23. A histogram of reference measurements showing the distributions of samples with low and high XS content.

It is clear that while there was one large grade at low XS that was normally distributed, at the higher XS content levels there were more grades, and there was little evidence of normality in the distribution. To further investigate the distributions of the reference material normality plots were produced, both for the data set as a whole, XS_{lab}, and for the individual sets split according to XS content. Samples with an XS content lower than 6% formed XS_L, and samples with an XS content greater than 6% formed XS_H.

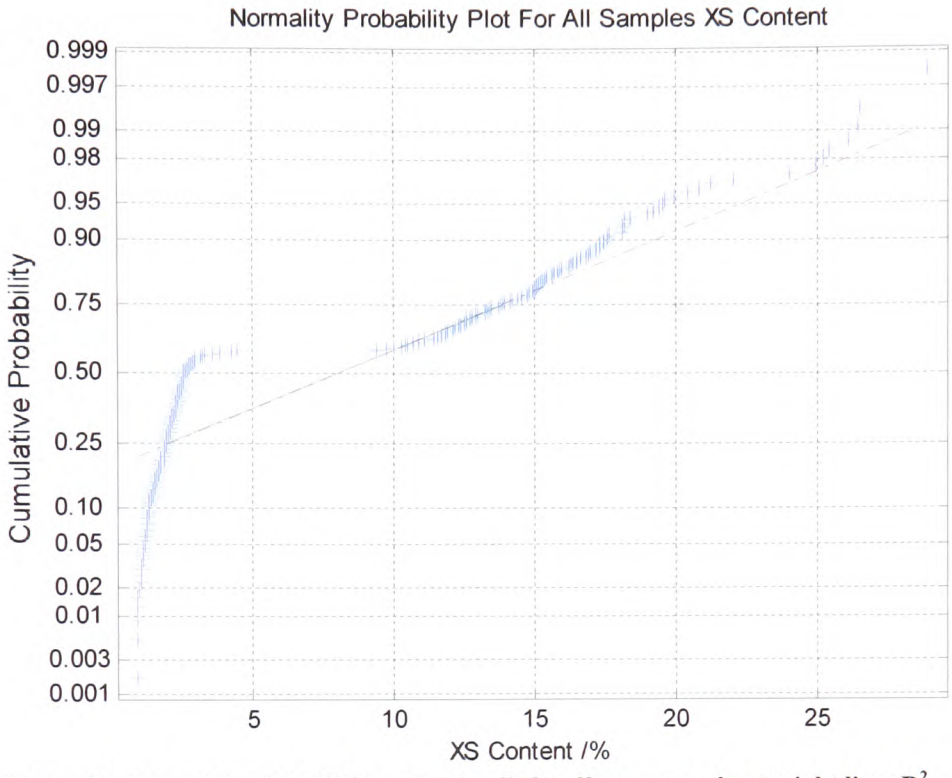

Figure 24. Normality plot of POL_Y showing little adherence to the straight line; $R^2 = 0.831$.

Normality plots are a graphical method for determining whether a system is normally distributed. A plot consists of a scaled axis and a straight line. If the data conforms to the straight line, the null hypothesis that the data is distributed normally cannot be disregarded. However, if the data does not conform to the straight line, the null hypothesis of normality must be rejected. The normality plot for XS_{lab} (Figure 24) shows that at the lower end of the XS content range (XS_L) the samples did not conform to the straight line, suggesting that the null hypothesis was false and should be rejected. The samples at the higher end of the XS content range (XS_H) adhered to the line and this suggests that the data at the higher end was normally distributed. When taken as a whole the R^2 was determined to be 0.831. The evidence from Figure 24 contradicts that from the histogram in Figure 23, which indicated that XS_L should be normally distributed, rather than XS_H.

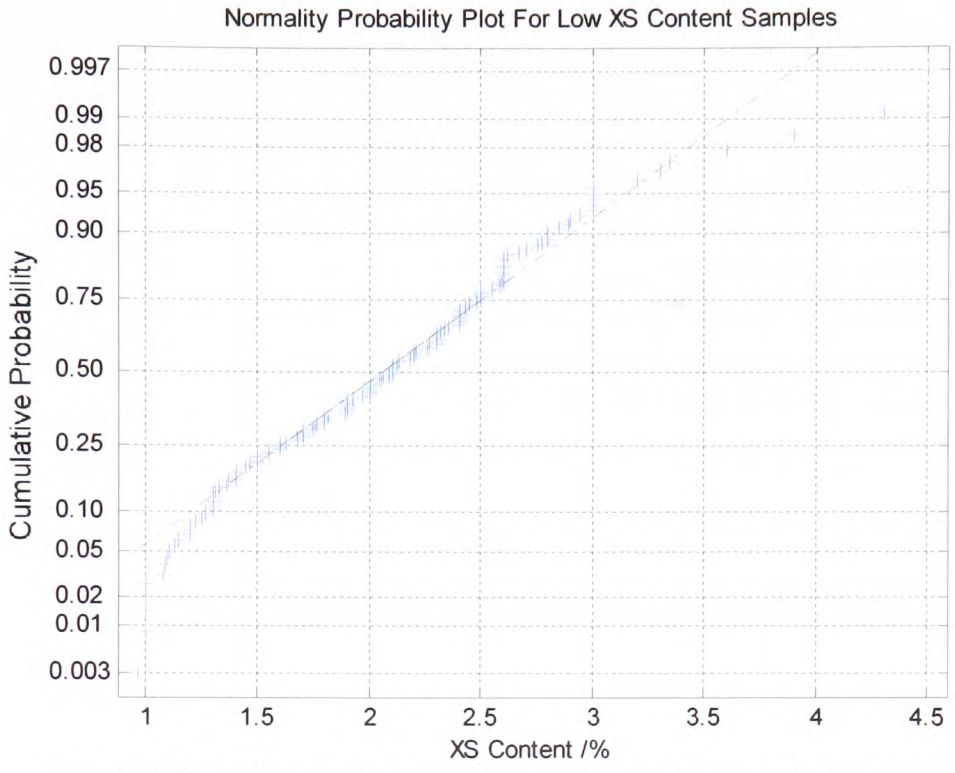

Figure 25. Normality plot for XS_L showing greater correlation to the straight line, suggesting a normal distribution; $R^2 = 0.925$.

Figure 25 shows the normality plot for XS_L. The points on this plot approximately conformed to the straight line with an R^2 of 0.925, supporting the hypothesis that XS_L was distributed normally. Figure 25 agrees with the information from Figure 23, meaning that the normality plot shown in Figure 24 could have been skewed by the magnitude of XS_H. This would give XS_L the appearance of non-conformity to the normal distribution.

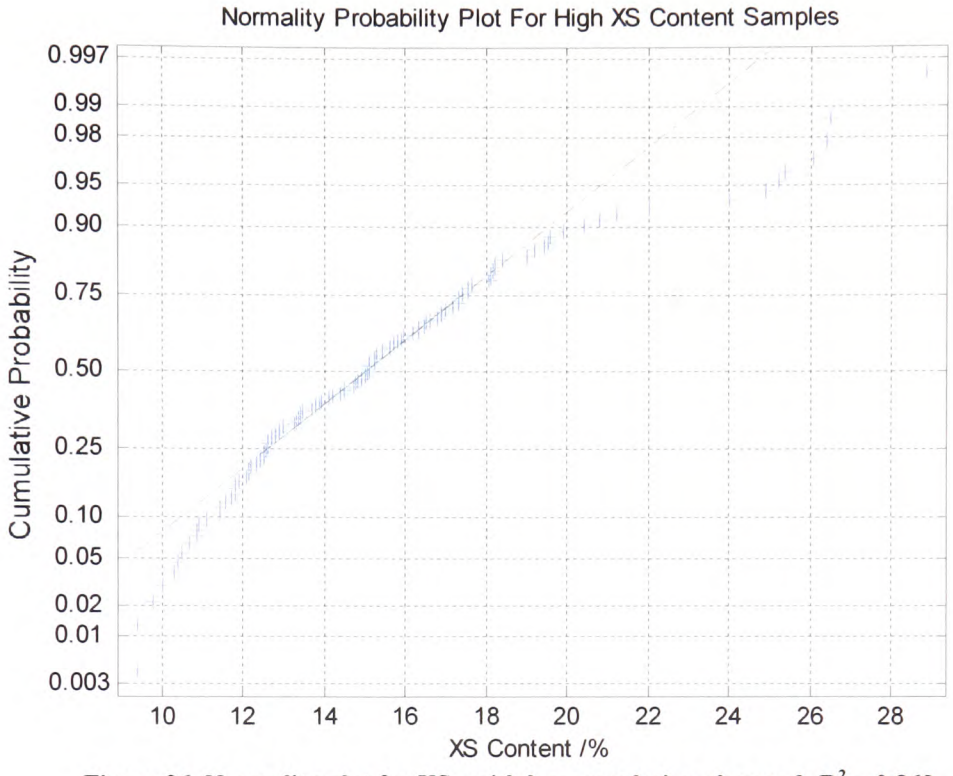

Figure 26. Normality plot for XS_H with less correlation observed; $R^2 = 0.869$.

The normality plot for XS_H (Figure 26) again supports the hypothesis that the samples with XS content greater than 6% were not distributed normally, as only a few of the data points corresponded to the straight line. The R^2 was determined to be 0.869.

Using standard methods of model building (such as PLS) to produce models in order to predict samples with a high XS content would be difficult due to the non-normal distribution. The reason for the lack of normal distribution is that the polymers with a high XS content are made in much smaller quantity than those with a low XS content, and with a fixed scheduled sampling procedure in place there are many fewer reference samples collected and analysed. A proactive sampling procedure, in which samples are taken and measured when there is new information content, i.e. when the samples fall outside the ranges currently encountered, or to fill holes in the reference material distribution, would be suitable in this instance, as it

would require fewer samples being sent for analysis and would maximise the value of each reference measurement.

5.1.2 Interpretation of the Pre-processing Designs

Design of experiments was employed extensively to determine the appropriate method of pre-processing to be applied to the data prior to the predictive modelling. The means of displaying the results from the design are shown in Figure 27. The main diagonal exhibits the results from each individual pre-processing factor. The rows and columns show the interactions between each processing method.

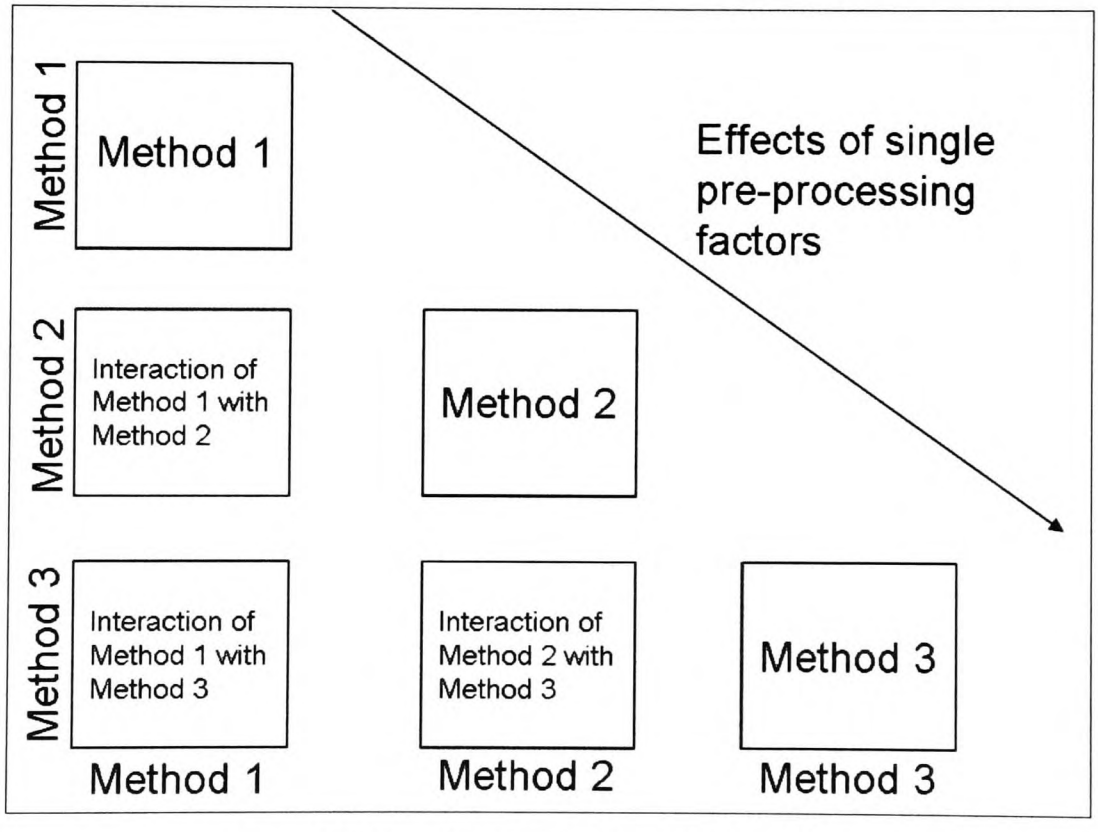

Figure 27. Schematic of a typical pre-processing design.

Further discussion regarding the interpretation of these designs can be found in the papers by Flaten and Walmsley.[76, 77]

5.1.3 Current Model

The first step of this modelling procedure was to reproduce the model currently being used online. This generated a baseline to which subsequent models could be compared.

This initial PLS model was made in three stages. The first stage was the determination of the best method for pre-processing the FID spectra. This was achieved through a full factorial design of experiments, using the final model prediction error as the response function (Figure 28).

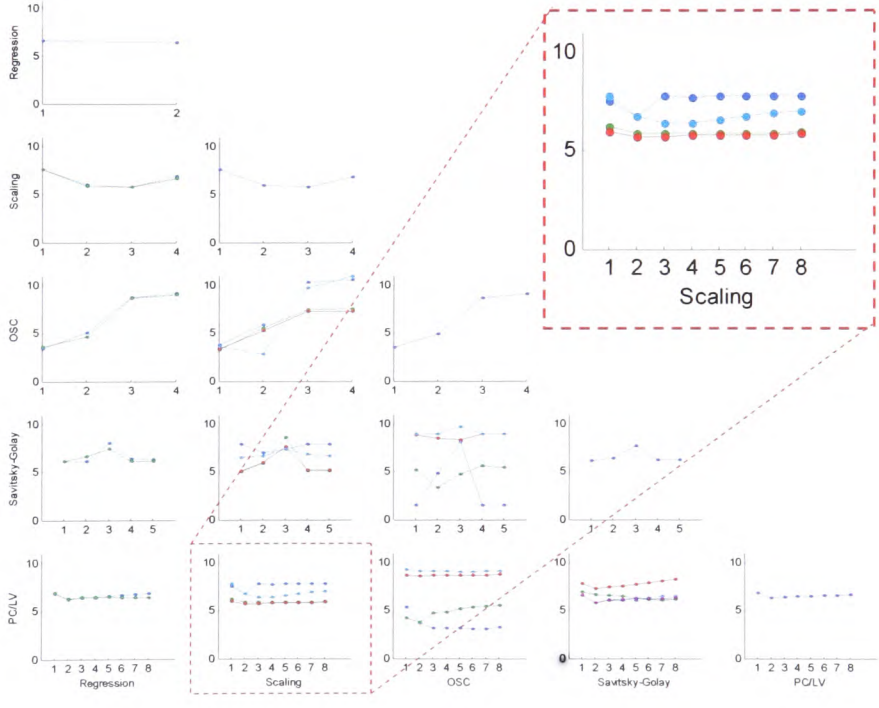

Figure 28. DOE model for the pre-processing of FID and XS_{lab}. Inset: the best method selected, mean centring with four LVs.

The inset highlights the most important result from this design. It shows that the best model will result when the FIDs are mean centred and that the final PLS model should contain four latent variables.

The second stage applied the optimal pre-processing method in order to build the calibration model. The samples selected to build the calibration model were chosen randomly. The data was split using an approximate ratio of 3:1 between the calibration and validation subsets. This led to FID_{cal} and XS_{cal}, the calibration data set that constituted the FIDs and the XS reference measurements containing a total of 233 samples. The remaining 78 samples were used to form the validation data set FID_{pred} and XS_{pred}. The resulting PLS calibration model (Figure 29) contained four latent variables (describing 99.6% of the total variance in the data) and the RMSEC was found to be 1.75%.

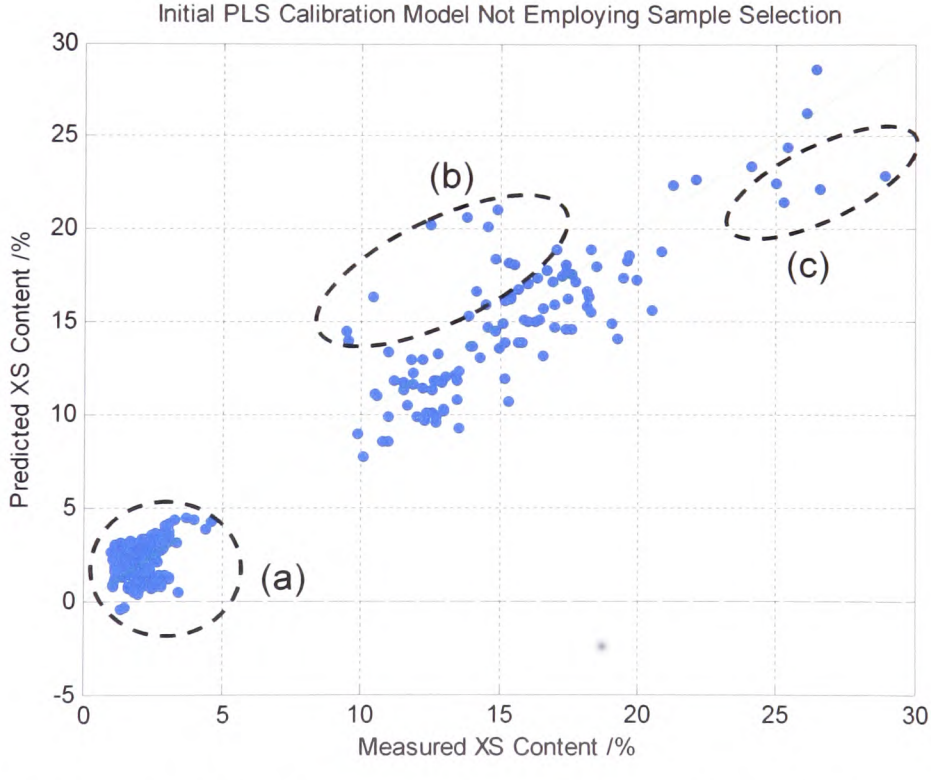

Figure 29. Reproduction of the PLS calibration currently used online; RMSEC = 1.75%.

The plot of the values predicted by the model versus the actual measured values of the data is shown in Figure 29. This figure further highlights the results from the initial study that the samples with a lower XS content produced a tight cluster of samples (a), with only small residual errors and minor amount of deviation from the

incorporated errors in both the NMR and the reference measurements. The RMSEP of 2.15% was slightly higher than that of the calibration error and it was considered fit for purpose. The bias of this model was determined to be 0.0787%.

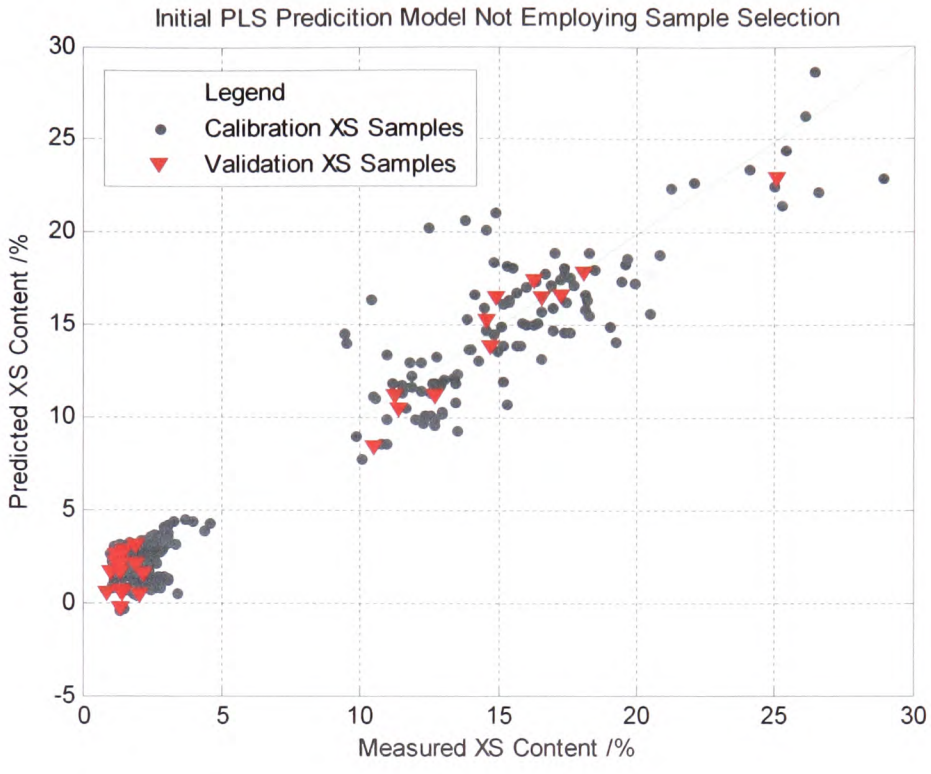

Figure 30. Reproduction of PLS prediction model currently used online; RMSEP = 2.15%.

Figure 30 shows that the validation samples with a lower XS content were accurately predicted, as these were the samples best described by the calibration data. The validation samples with a higher XS content were not predicted as

line of predicted values versus actual values. An examination of the samples with a higher XS content showed a greater amount of deviation and higher residual errors, especially the samples within (b) and (c).

The third stage of this procedure was the prediction of the XS content for the validation data, $FID_{\_pred}$ and $FID_{\_pred}$ (Figure 30) using the model parameters determined in the previous stage. The validation data was true validation data, as it was taken over the same period of time as the calibration samples, and as such, it

accurately, and this had a large effect on the final RMSEP. This indicates that some local type of modelling or sample selection would lead to more accurate models.

5.1.4 Local Partial Least Squares Models

To build the local models the calibration and validation data were split based on XS content. Samples with an XS content lower than 6% formed FID_{L_CAL}, XS_{L_CAL}, FID_{L_PRED}, and XS_{L_PRED}, data sets for calibration and validation, respectively. The samples with an XS content higher than 6% formed the calibration and validation data sets FID_{H_CAL}, XS_{H_CAL}, FID_{H_PRED}, and XS_{H_PRED}.

5.1.4.1 High Content Model

The modelling occurred in three stages, the first of which use a design of experiments to determine the optimal pre-processing method The best predictive model (inset, Figure 31) was found to be the mean centring of data prior to building a model with three latent variables. Figure 31 also shows that other methods of pre-processing the data (such as OSC and Savitsky-Golay derivation and smoothing) were unsuccessful in producing a better predictive model, due to the lack of a baseline and high correlation between each NMR FID.

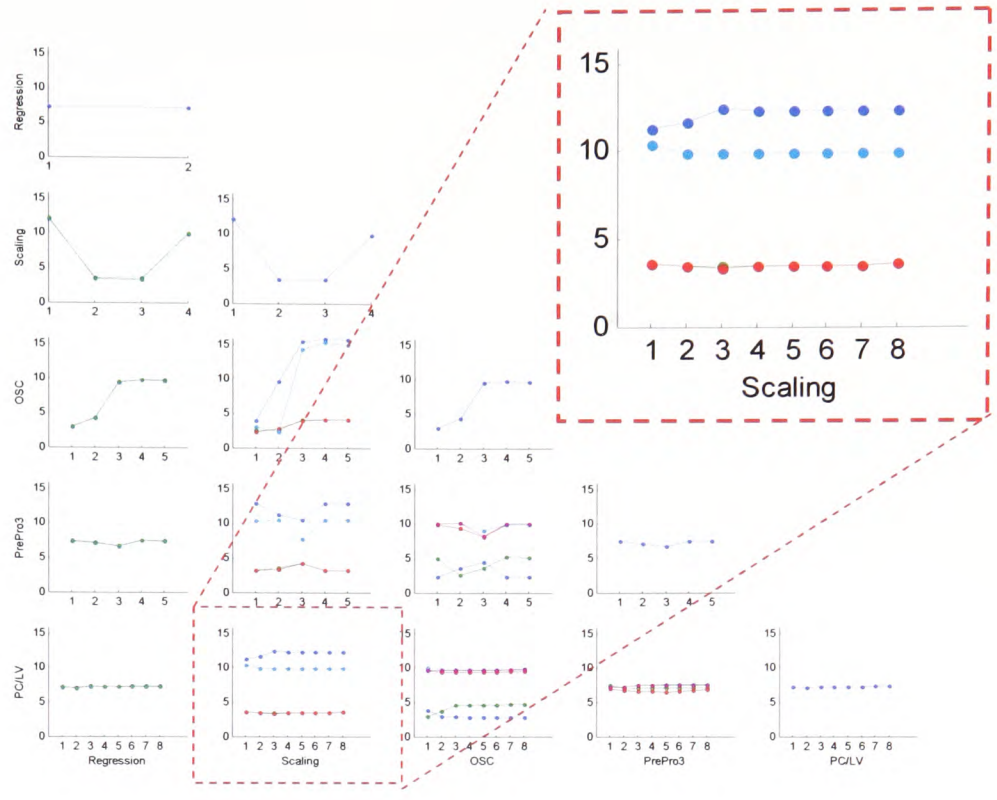

Figure 31. DOE results for the pre-processing of samples with high XS content. Inset: the best method selected, mean centring with four LVs.

The second stage was the application of the selected pre-processing method and creation of a calibration model using FID_{H_CAL} and XS_{H_CAL} (Figure 32). The RMSEC for the model was found to be 1.82%. Using this calibration model, a PLS validation model was constructed for the data contained in FID_{H_PRED} and XS_{H_PRED} (Figure 33). The RMSEP was calculated to be 2.12%. The relatively high prediction error can be attributed to the distribution of the data; some samples in the validation set appeared only once and were thus difficult to predict.

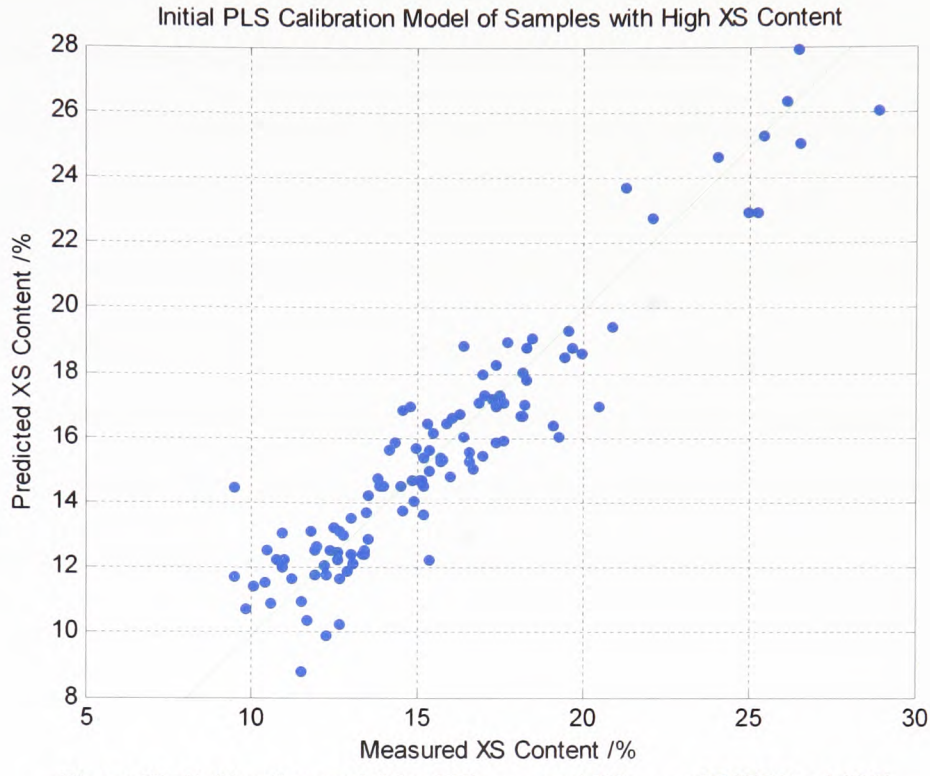

Figure 32. Calibration model for FID_{H_CAL} and XS_{H_CAL}; RMSEC = 1.82%.

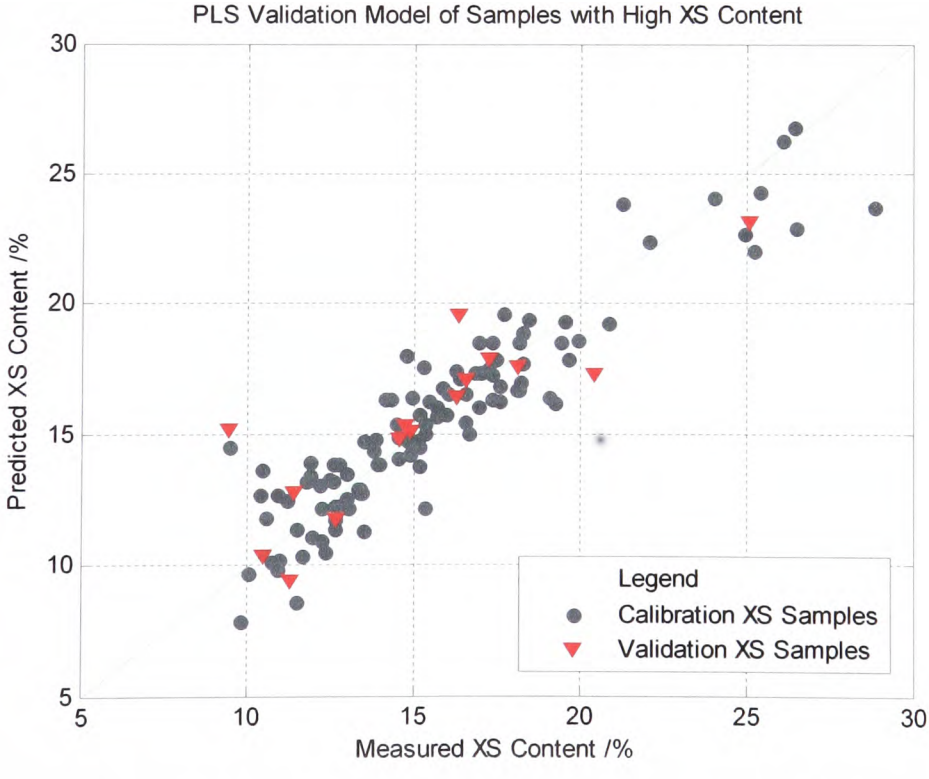

Figure 33. Validation model for FID_{H_PRED} and XS_{H_PRED}; RMSEP = 2.12%.

5.1.4.2 Low XS Content Model

The same modelling procedure was applied to the data with an XS content lower that 6%, FID_{L_CAL} and XS_{L_CAL}. DOE was employed and the best pre-processing method was found to be mean centring the data with three latent variables within the calibration model (Figure 34). OSC and Savitsky-Golay Derivatisation, the nearly-standard methods of processing spectroscopic data, were also tested, and the results showed that use of either technique would result in a model with a higher RMSEP than the optimal. This was attributed to the high degree of correlation between samples and the relatively low amount of noise within the system.

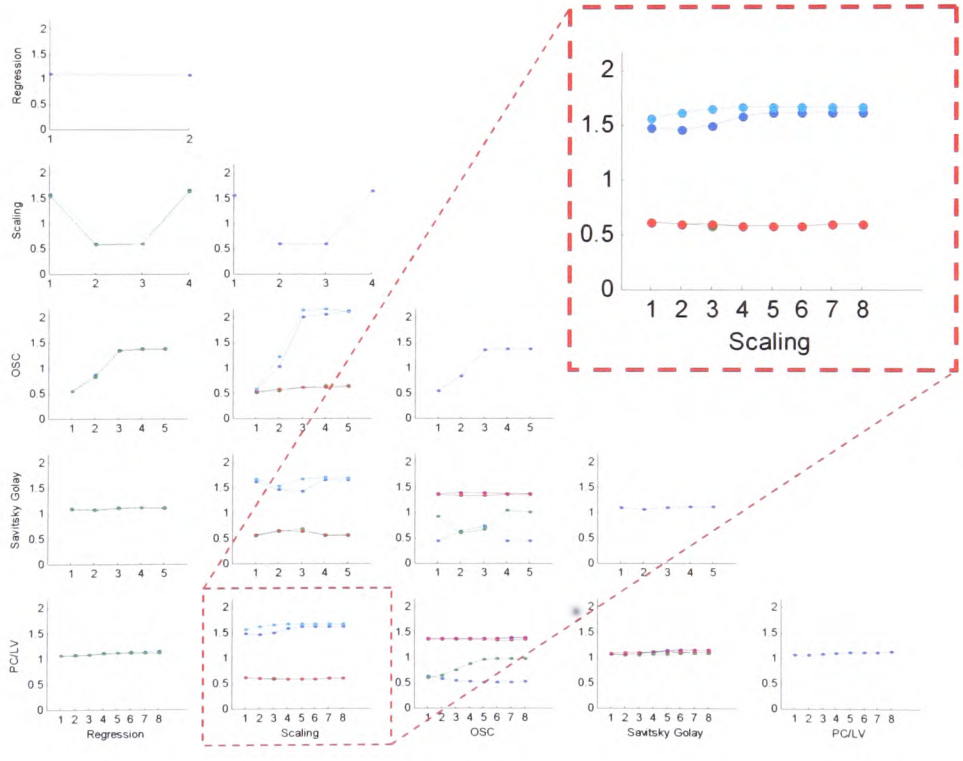

Figure 34. DOE model for the pre-processing of FID_{L_CAL} and XS_{L_CAL}. Inset: the best method selected, mean centring with three LVs.

As before, the calibration model was produced using the optimal methods of pre-processing (Figure 35), and the RMSEC was determined to be 0.182%. This was

an order of magnitude less than the calibration error determined for the data with a higher XS content. This can be ascribed to the normal distribution of the data when compared to samples in the high XS content model. It was also partly due to the method of calculation of the RMSEC, as making calibrations with smaller numbers introduces a magnitude bias effect.

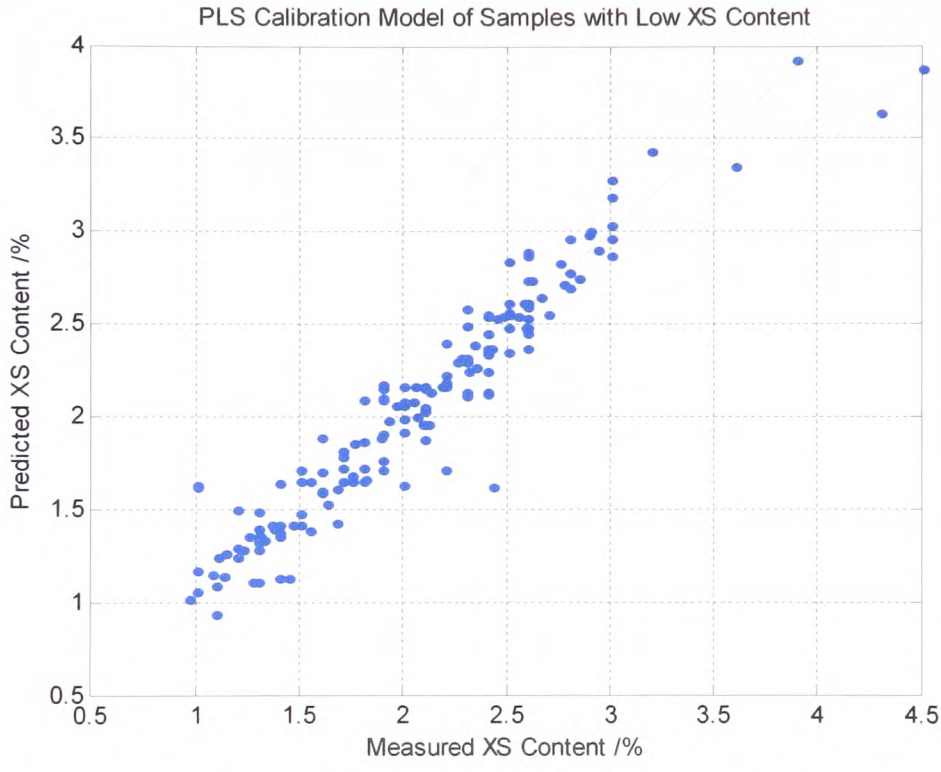

Figure 35. PLS calibration model for FID_{L_CAL} and XS_{L_CAL}; RMSEC = 0.182%.

Following this, FID_{L_PRED} and XS_{L_PRED} were put into the calibration model to produce a prediction model (Figure 36). The RMSEP was calculated to be 0.549%. This was approximately four times the RMSEC. The relatively large ratio between the calibration and prediction errors was partially due to samples (a), (b), and (c), as these three samples were the only samples to be poorly predicted by the model. However, further investigation showed that these samples were not outliers and should be included in the model. The exclusion of these samples lead to a new RMSEP of 0.379%.

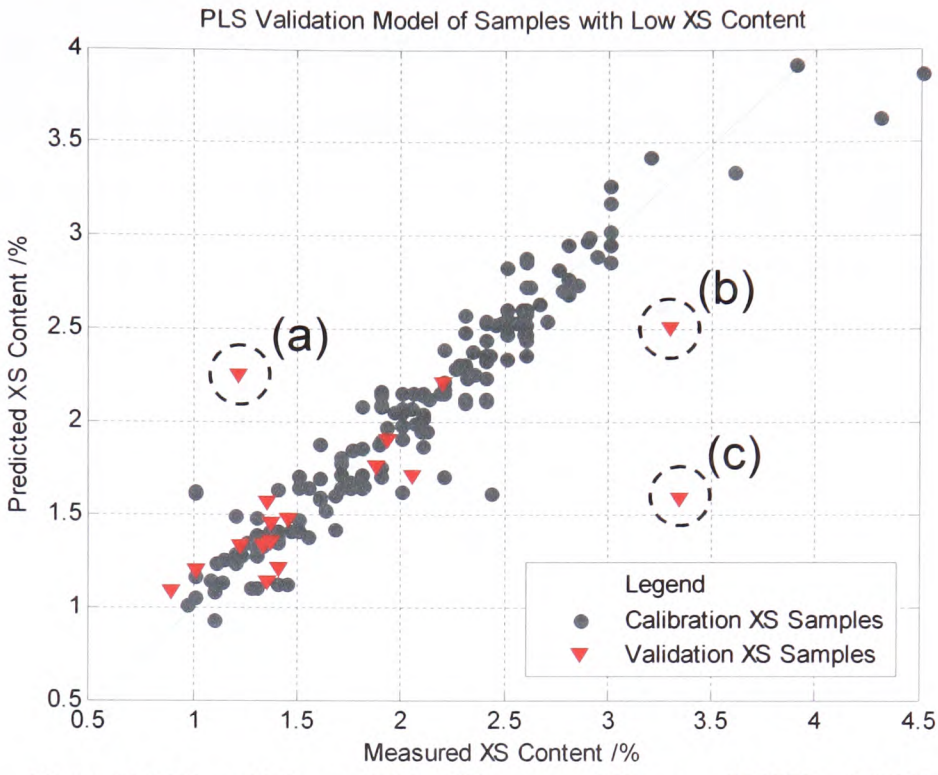

Figure 36. PLS prediction model for FID_{L_PRED} and XS_{L_PRED}; RMSEP = 0.549%. (a), (b), and (c) are samples that were poorly predicted by the model.

This difference in RMSEPs was due to the nature of the sample distributions with each subset. From the initial examination, it was shown that the data was not distributed normally. Examination of the lower XS content sample subset indicated that this was normally distributed about the mean, whereas the higher XS content sample subset was not. The distribution of the samples was attributed to the non-continuous nature of producing different batches of polymer grades with varying percentages of XS content. In some cases it may not be possible to make valid predictions due to insufficient information contained locally about the samples.

Although localised modelling resulted in two models that made better predictions than the initial PLS model, it would be very difficult to use such a method online. Firstly, a local model would be required for every grade of polymer produced, which currently stands at over forty. The introduction of any new grade would require the construction of a new model, or else predictions would be

unreliable until the information was sufficiently updated information within the initial model. Secondly, each model would require the individual optimisation of pre-processing methods and modelling parameters as well as the removal of outliers. Thirdly, before any predictive models could be constructed the correct local calibration model must be chosen for the sample to be predicted from. This classification is not a trivial procedure. The need to add another layer of complexity to the model would introduce an additional area for potential error. The method of classification would also require optimisation on a broader scale, such that all the samples could be classified into the model. Fourthly, while the initial examination of the data suggested a bimodal distribution, the manufacturers informed us that there were, in fact, over 40 grades within the initial PLS model and therefore over 40 modes present. This indicated that the grades must overlap significantly such that when PLS is performed the 40 modes appear to only number two. Lastly, the localised modelling method would struggle to deal with inliers, transition points used to monitor the production cycle and the samples which fall between two grades and hence between two models. All of these factors show that applying localised modelling online would be an inappropriate method for the treatment of this data.

5.1.5 Sample Selection Models

5.1.5.1 Optimal Solution

The first sample selection procedure applied to the data used the condition number of the matrix to choose an optimal set of calibration samples. This used sample selection to build a calibration set that best described the entire data set in one model. The use of the condition number produced the calibration spectra and reference matrices FID_{COND_CAL} and XS_{COND_CAL}. The remaining samples formed the

validation sets FID_{COND_PRED} and XS_{COND_PRED}. The calibration samples selected (Figure 37) show that the samples were taken from areas of both lower and higher XS content.

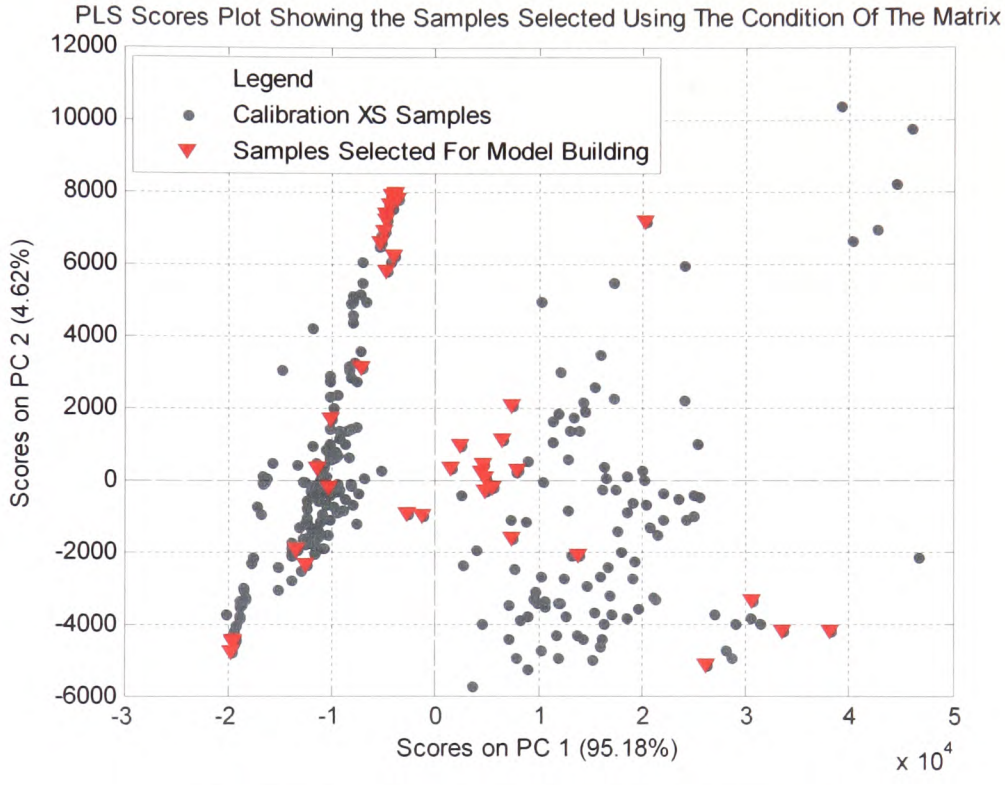

Figure 37. PLS scores plot. The samples selected using the condition number come from areas of low and high XS content.

A full factorial design was employed to determine the best method of pre-processing, which was found to be mean centring with the removal of two OSC components. The results from this design are shown in Figure 38.

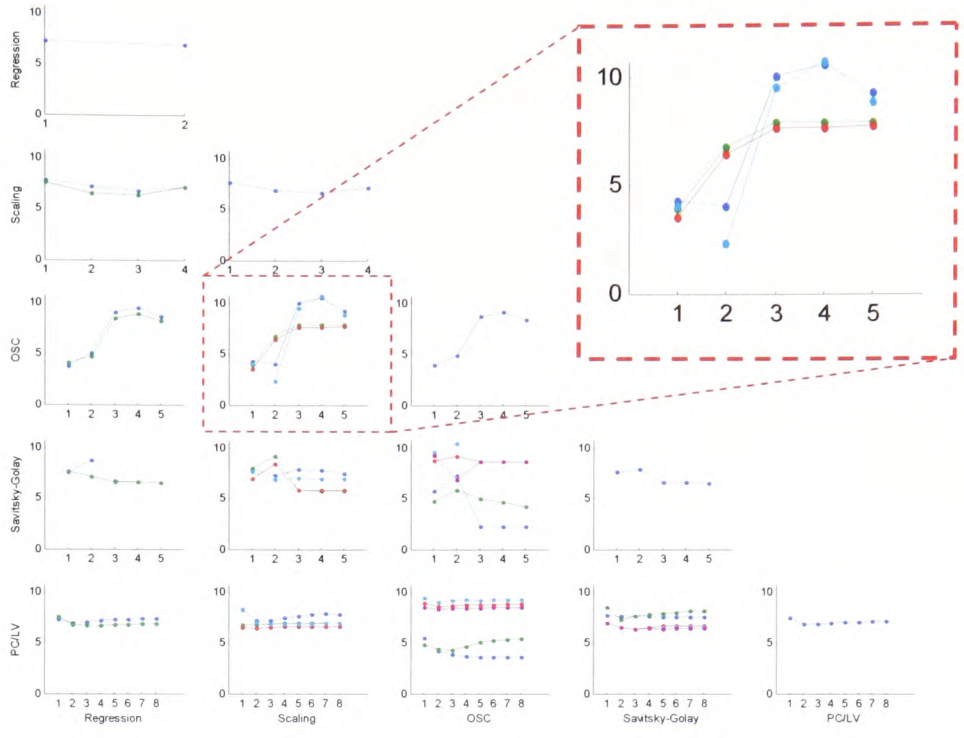

Figure 38. DOE model for pre-processing using the condition number for sample selection. Inset: the best method selected, mean centring with two OSC components.

The resulting calibration model (Figure 39) has a RMSEC of 0.588%. Using FID_{COND_PRED} and XS_{COND_PRED} the prediction model was generated (Figure 40), and it has a RMSEP of 1.76%.

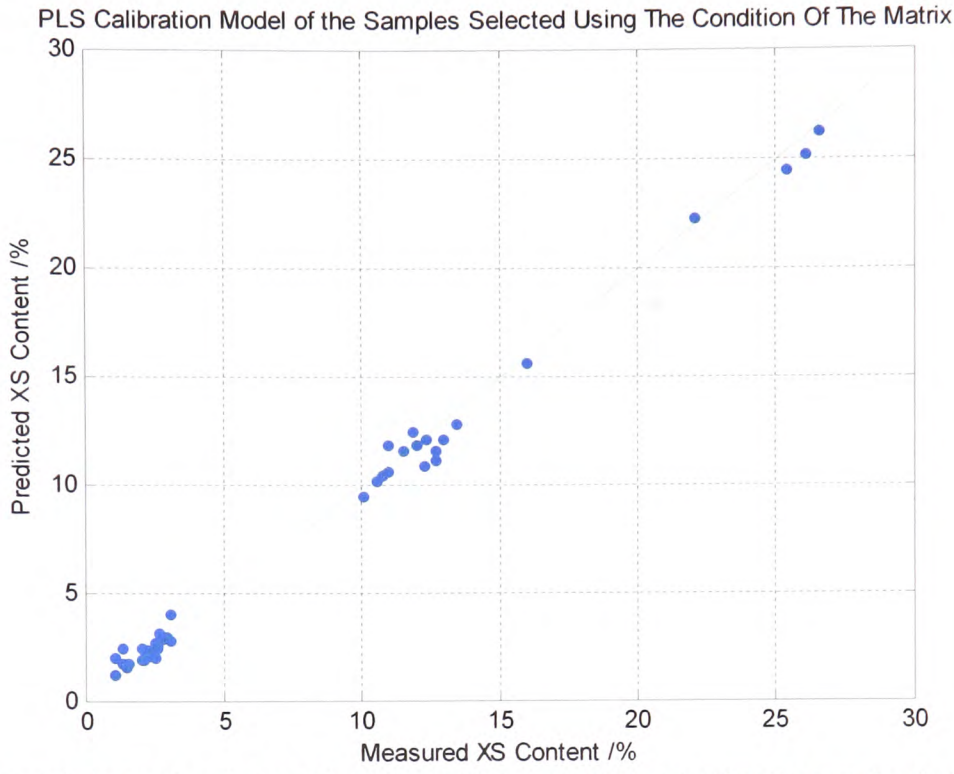

Figure 39. PLS calibration model produced using the condition number for sample selection; RMSEP = 0.588%.

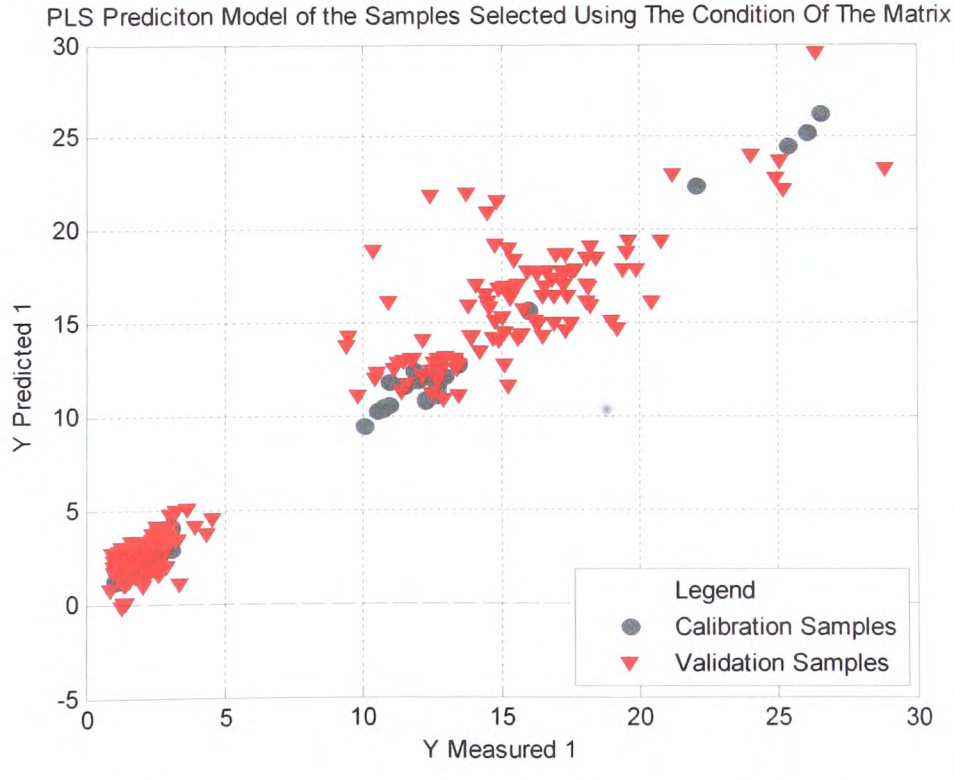

Figure 40. PLS prediction model using the condition number for sample selection, RMSEP = 1.76%

The results from this analysis show that the data could not be treated as a whole with one calibration set, due to the distributions and varying modes.

5.1.5.2 Adaptive Selection

Adaptive sample selection is defined as a method of selecting samples that are most pertinent to the sample being predicted. Unlike the use of the condition number, adaptive sample selection allows for the production of a calibration model for each validation sample. Adaptive sample selection determines the criteria by which the most pertinent samples can be selected for modelling, with the goal of maximising the strengths of both local and global modelling systems. Several different sample selection methods with differing selection criteria were explored, such as the distance in the scores and the correlation and the distribution amongst the scores space. A model with samples selected at random was built as a control model for comparison.

5.1.5.3 Shenk and Westerhaus

The Shenk and Westerhaus criteria use the correlation between the calibration and validation spectra to choose samples. The optimum number of samples for calibration was found to be 15, with the optimum number of latent variable determined to be six. These values were assessed using a design (Figure 41).

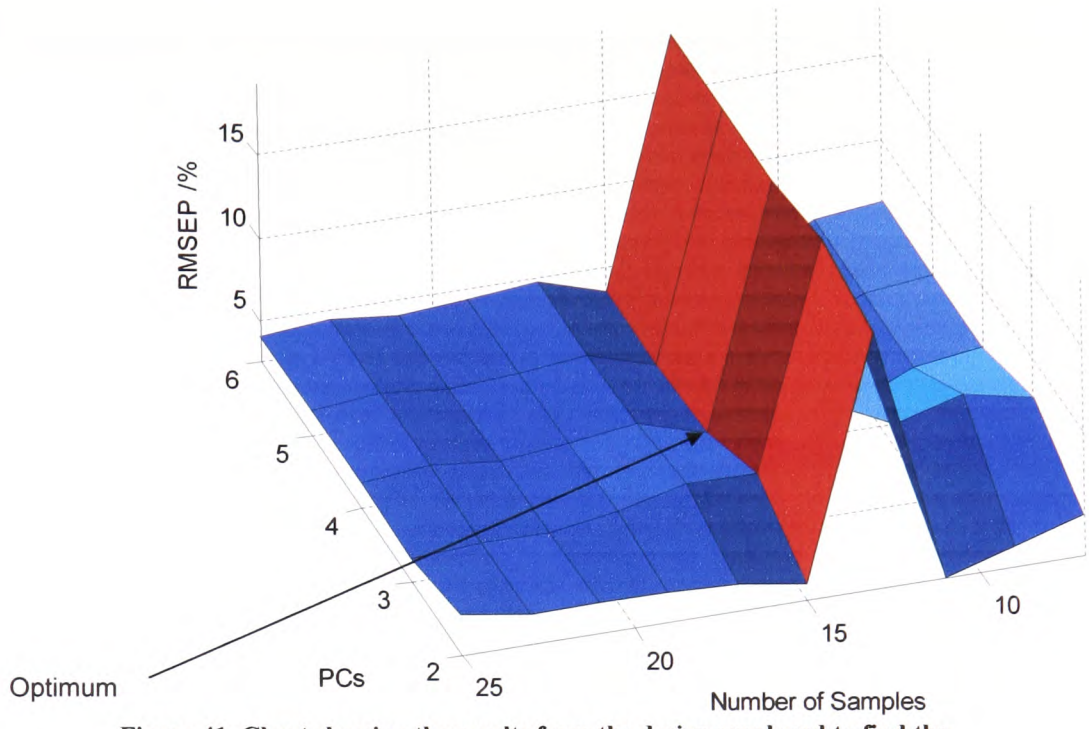

Figure 41. Chart showing the results from the design employed to find the optimum number of samples and latent variables to build a calibration model.

The average variance captured by the six latent variables was 92.6% of the spectral variation. An example of the samples selected using correlation as the selection criteria is shown in Figure 42. The ridge formed on the addition of 13 samples into the model could be due to the addition of a sample that is at the extremes of the model. The ridge declines as more samples similar to the extreme sample have been added and thus normalises and reduces the error and leverage of the first extreme sample.

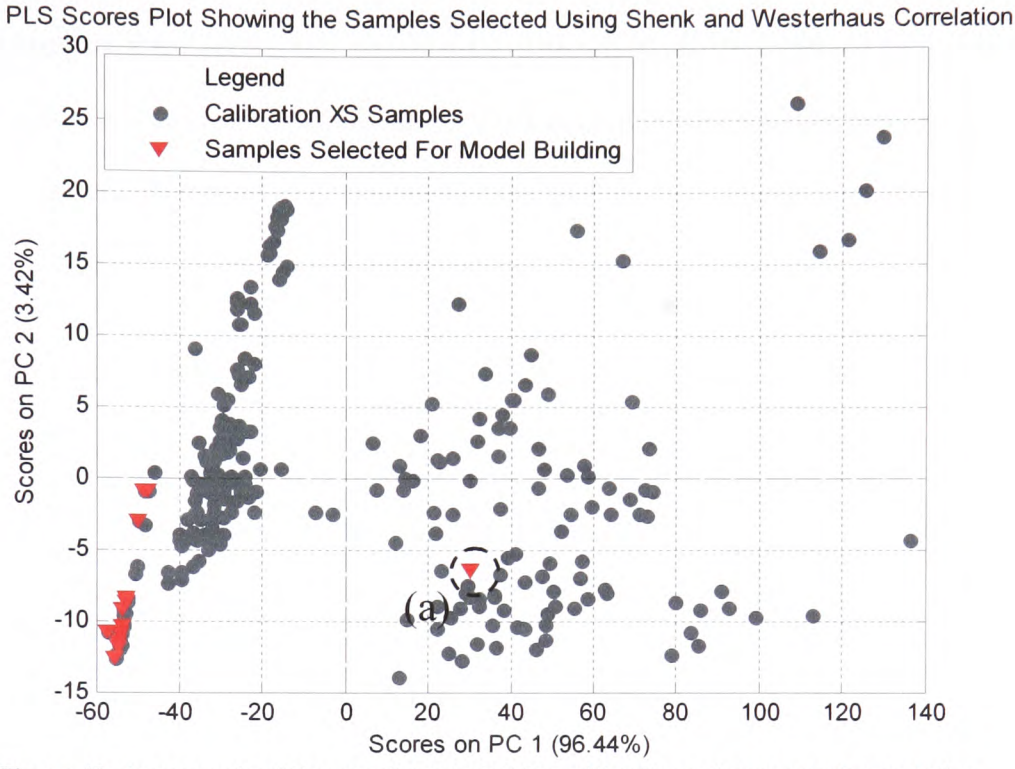

Figure 42. An example of the samples selected for calibration using spectral correlation. The validation sample is labelled (a).

Unlike the samples selected using the Euclidean distance, those selected using correlation seemed to bear no relevance to the validation sample (a). The average XS content was 2.09%, while the XS content of the validation sample was 14.9%.

The final optimised model had an average RMSEC of 5.66×10^{-3}%, which was significantly lower than any models produced thus far. The RMSEP was determined to be 3.58%, which was higher than the RMSEP of both the initial and Euclidean-based sample selection models. The fact that the RMSEP was approximately 1600 times greater than the RMSEC strongly suggests that this method of sample selection over-fits the data, resulting in models that are highly calibrated but predict poorly.

The Shenk and Westerhaus approach gave an RMSEP value that was six times the RMSEP of the Euclidean-based sample selection approach. This was due to the nature of the FIDs; all of the spectra were highly correlated and therefore the spectra selected were not from the appropriate grade of polymer within the calibration set.

Additionally, the continuous nature of chemical processing operations means that a new sample is recorded every ten minutes. These samples are collected continually and are time-series correlated. By using a selection method that employs correlation as the selection criteria, only the most recent samples (i.e. the last fifteen samples) would be selected to build calibration models.

5.1.5.4 Euclidean Distance-Based Selection

Euclidean distance-based sample selection chooses the most pertinent samples to build a calibration model from the distance between the calibration and validation samples. Essentially, the calibration samples with the smallest Euclidean distance from the sample to be predicted are chosen to build the model. Using this method, a model is created for each new validation sample. In this case, the number of samples and latent variables to build the calibration model was optimised using DOE and was found to be 18 samples with five latent variables to be included in the calibration model (Figure 43). PLS models were built for each validation sample in $FID_{\_pred.}$

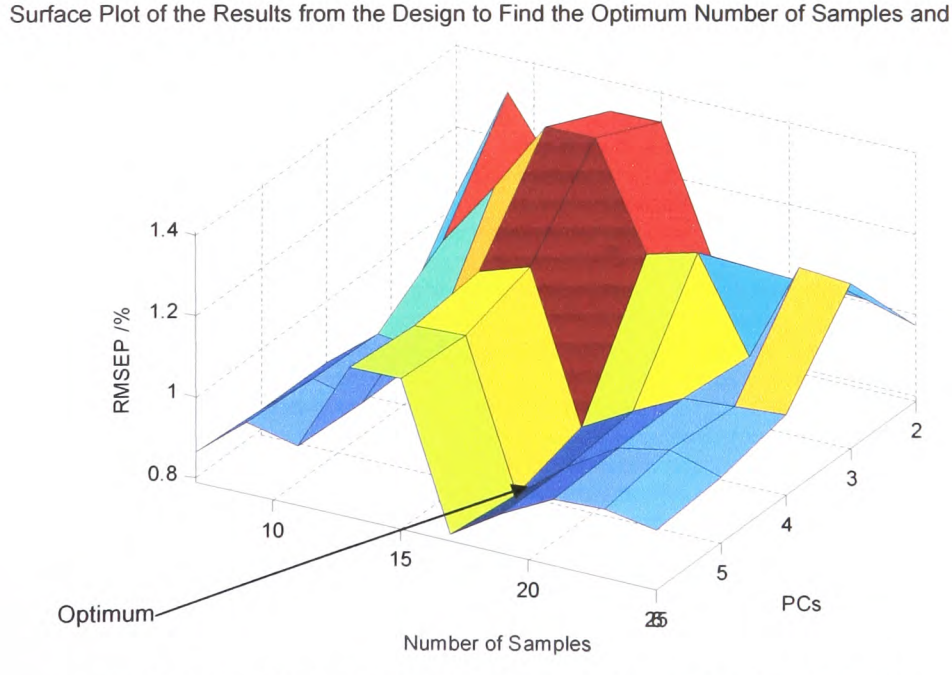

Figure 43. Chart showing the results from the design employed to find the optimum number of samples and latent variables to build a calibration model.

The average RMSEC was determined to be 0.348%. A typical example of the samples selected using the Euclidean distance is shown in Figure 44. The RMSEP was found to be 0.672%.

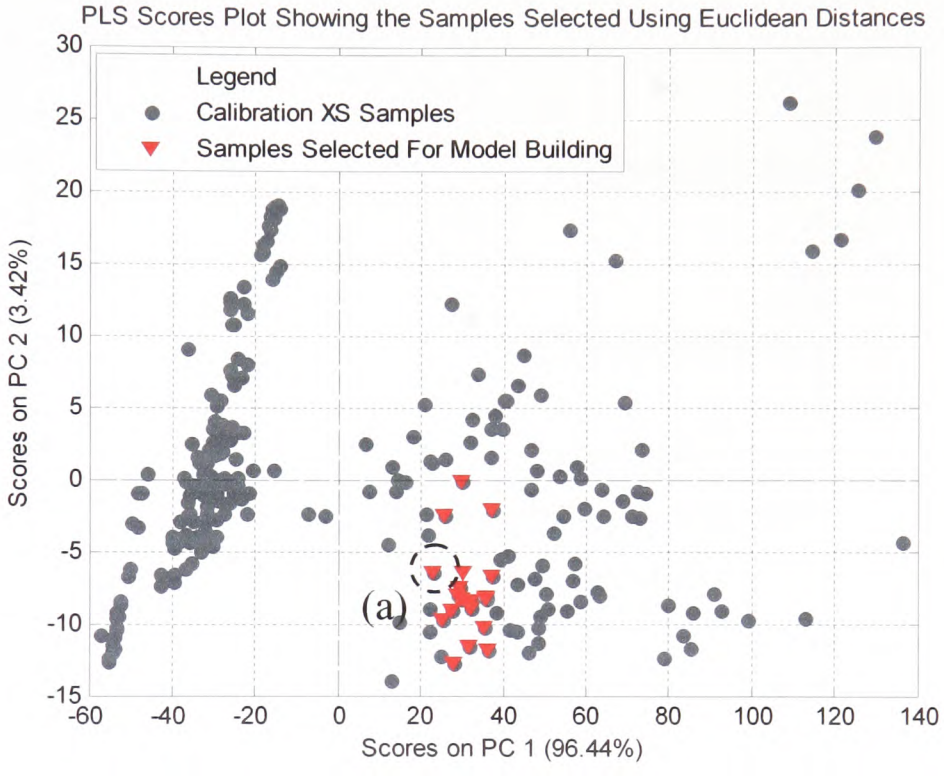

Figure 44. Example of samples selected to build a calibration model based on the Euclidean distance. The validation sample is labelled (a).

The average XS content of the samples selected was 15.6%, while the validation sample was found to have an XS content of 14.9%

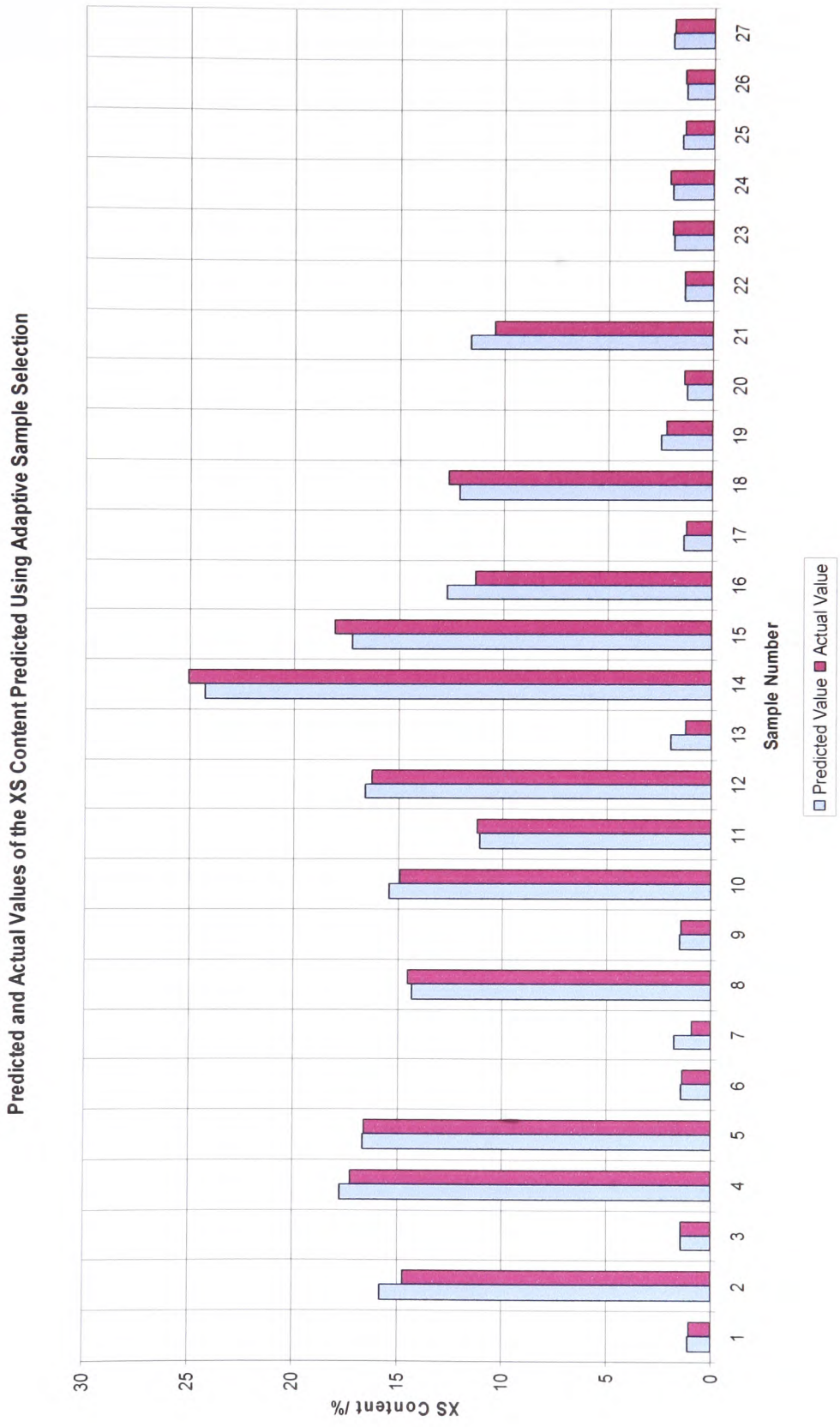

Figure 45. Predicted and measured values for the samples predicted using Euclidean distance sample selection.

The advantage of using Euclidean-based sample selection is that the samples closest to the validation samples are used, which allows the algorithm to pick samples from the same or very similar grades for each new sample. This negates the effects of variant XS content and the subsequent need for a classification procedure.

The prediction error of this modelling procedure was higher than that of the local model for the samples with an XS content lower than 6%, but this modelling methodology does not feature any of the complicating factors that the local models require. This method of selection criteria also predicted the samples with a lower XS content as well as the samples with a higher XS content (Figure 45) and the measured values differed by the same amount regardless of XS content. This further emphasised that locally modelling this data would be the wrong approach. The prediction error using the Euclidean distance to select samples was also less than that of the initial model, and by selecting only the pertinent samples the skewing and leverage of irrelevant samples was removed.

5.1.5.5 Implementation of the Online Model

The final stage of this study was the implementation of the off-line model at the point of analysis online on the NMR instrument. The process stream schematic in

Figure 46 shows that the NMR is located after the polymer has been formed into pellets. Every eleven minutes pellets are diverted from the process stream into the NMR chamber. Inside the chamber, the pellets are heated to form a liquid, and an NMR FID is then collected for the liquid sample. This sample is then purged from the instrument, and the sampling process begins again. Three times a day a laboratory sample is taken from the same sampling point as the online NMR. This sample is used to calculate a reference XS content measurement, and this measurement is then matched to the most recent NMR FID collected.

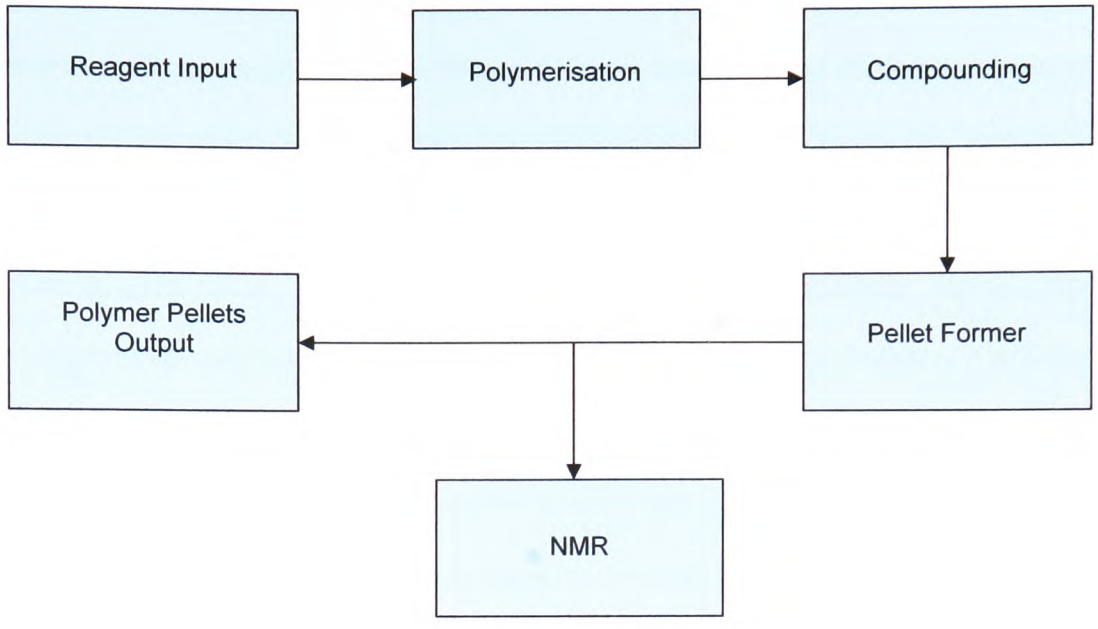

Figure 46. The process stream of the polymer production cycle. The NMR is situated after the formation of sampling pellets.

The implementation of the online procedure began by following the sampling procedures already in place at the plant, primarily the laboratory analysis and assessment of the time-frame by which measurements were made and could be used. The elapsed time from the collection of a sample of pellets to the recording of an XS measurement could range from eight hours to two days. The large discrepancy in analysis time depends on when the sample is taken, as a sample collected on a Friday afternoon might not have the XS value determined until Monday afternoon, simply due to the constraints of the average workweek.

The next step of the installation was the development of a graphic user interface (GUI) that could be used by laboratory staff, plant engineers, and process managers to track the predictions and performance of the modelling method. The success of the Euclidean distance off-line sampling procedure lead to its selection as the method for online sampling. The first GUI designed is shown in Figure 47, and this model contained sections displaying the newly collected spectra (a) and calibration

data (b), which gave the user a visual means of outlier detection by observing the spectrum. Sections (c) and (d) in Figure 47 show the Euclidean distances calculated as part of the sample selection stage from the scores shown in section (e). Section (f) is the control panel, and within this the model output the predicted XS value along with an alert status. The alert status allowed the user to determine the relevant validity of the prediction being made by displaying one of three indicators. A green alert status meant that the prediction made was reliable, while a yellow alert status meant that the prediction was within bounds but was less reliable and that a reference measurement should be ordered. A red alert status meant that the spectra collected and prediction generated could not be trusted, and at this point the process engineer must address the problem. By using the alert status and sections (a), (b), and (f), the GUI could be used for feedback control as well as process monitoring.

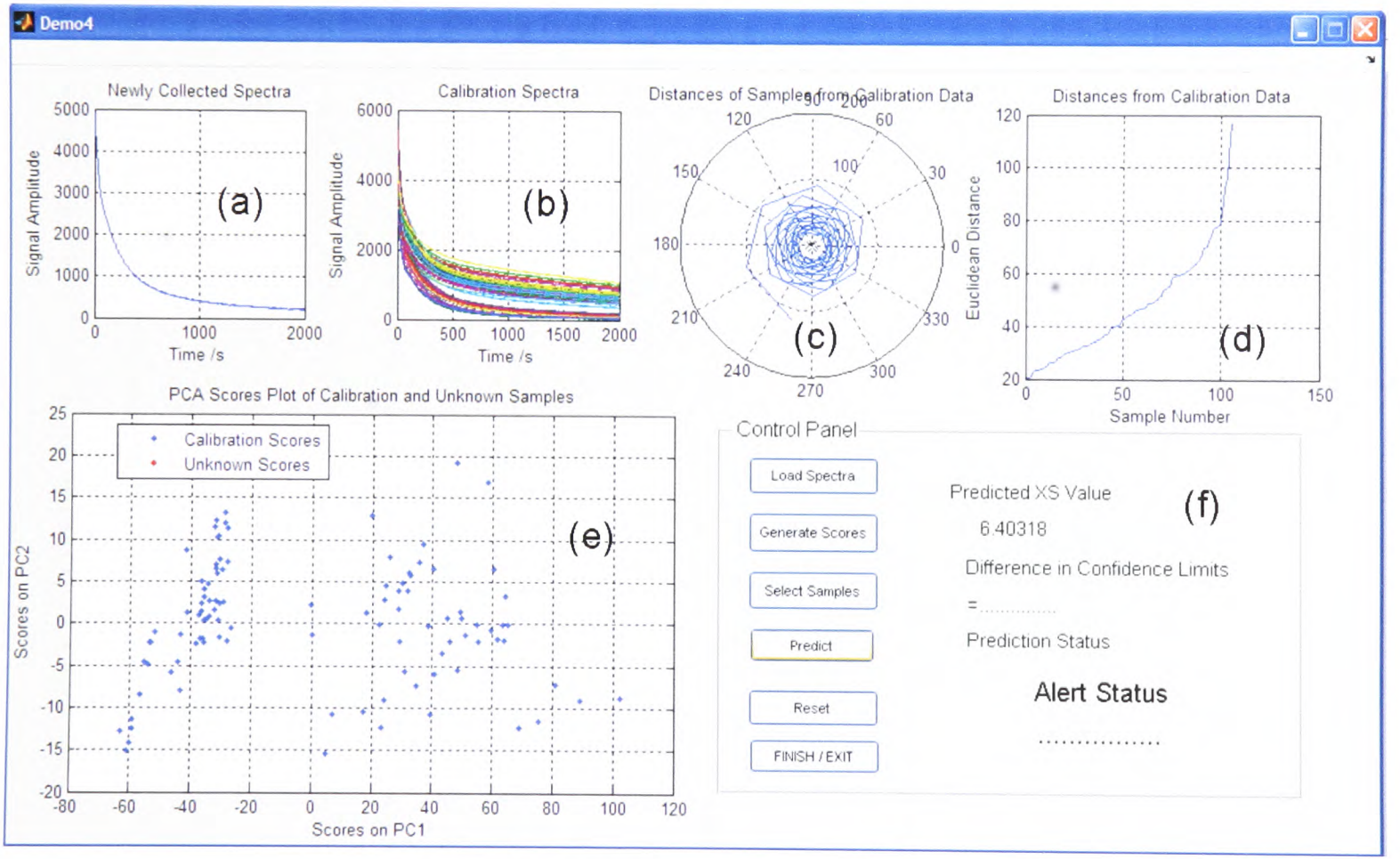

Figure 47. The first iteration of the online GUI.

After receiving feedback from the plant engineers regarding the initial user interface, a second GUI was produced (Figure 48). This still contained the sections pertaining to the new spectra (a), the calibration spectra (b), and the plot of Euclidean distances, but the section with the radar plot of the distances was removed. The scores plot (d) was modified to show the new sample within the scores of the calibration data (e), allowing a user to see the grade or cluster in which the sample was located. This new section allowed the user to track the production process, so that when there was an alteration to the grade being produced one could see the movement of the new sample within the scores towards the new grade; this was an important part of the feedback control procedure. Another improvement made for the second version of the GUI was to completely automate the control panel (f). This required only that the user load the spectra for prediction, and the predicted value would be calculated automatically. However, the automation of the procedure required an overhaul of the GUI so that the user would only be able to start or stop the process, as needed.

Added to the second version was the inclusion of results from the global model being used, and the comparison of the two values allowed for a rough form of visual validation. Also included was the calculation of the relative errors in each prediction, as well as the overall confidence in the data if the new sample spectrum was to be included within the overall calibration data. This was the first form of model maintenance employed. If the confidence within the data improved due to the inclusion of the new sample, the sample should be added to the calibration set with a corresponding laboratory reference value.

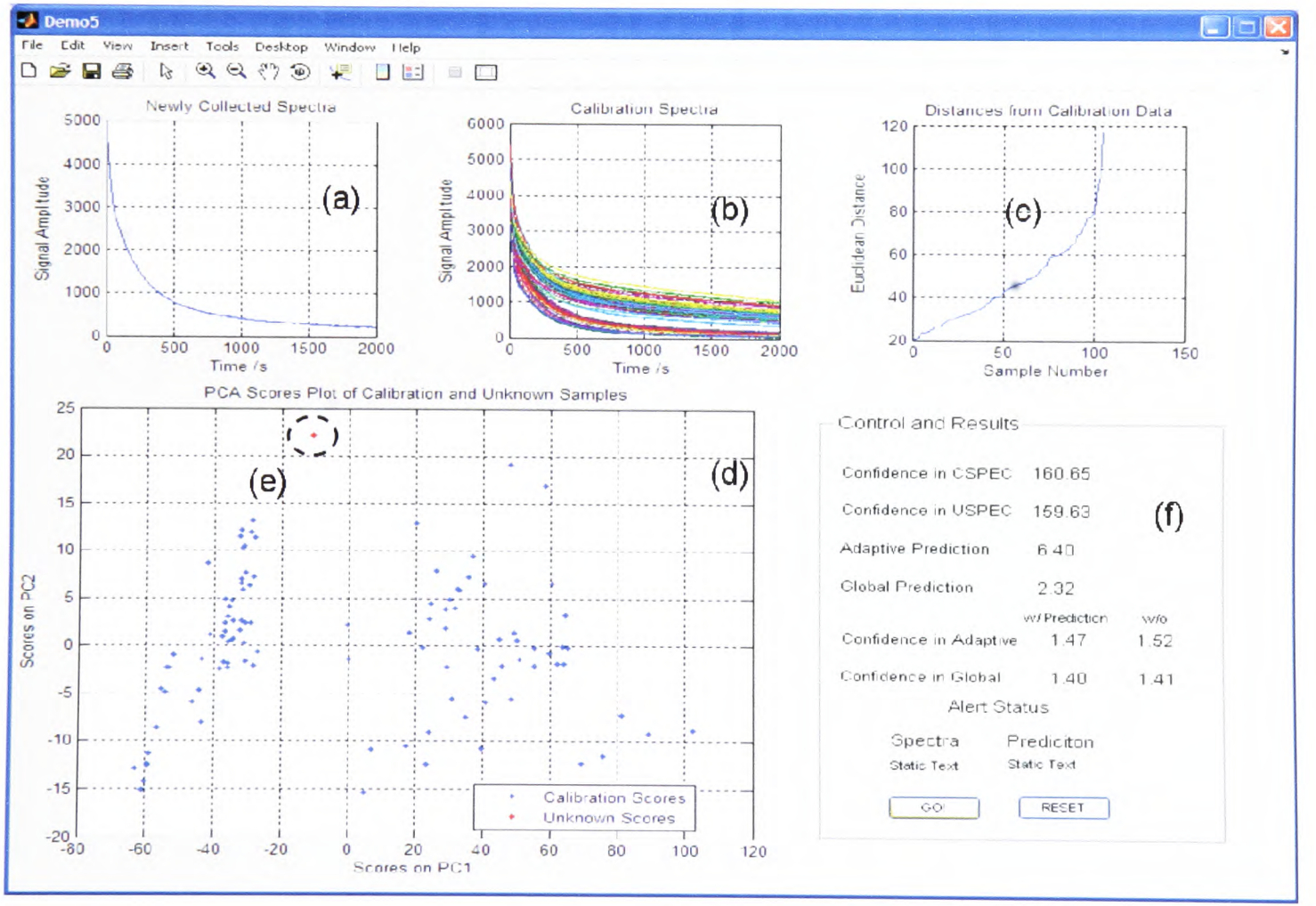

Figure 48. The second iteration of the online GUI. This included minor improvements to automated processing and error calculation.

Feedback regarding the second GUI lead to the development of the third (and current) user interface. Once started, this interface is fully-automated, importing the new FID from the capture software, selecting samples, making predictions, and determining an alert status for the spectra and predicted XS content. The use of two alert status procedures allows a user to separate bad predictions made with good spectra from bad predictions made with bad spectra. This version still included the plots of the new spectrum and the calibration data, but section about Euclidean distances was removed, as it was found to be superfluous. The scores plot is included, as it was determined to be a good method of process control. The control panel went through another evolution, and this version allows the model to update by adding samples to the calibration model. The major improvements to the interface involved the functions behind the interface. This GUI reads in data directly from the capture interface and process the data accordingly. It also has an error catch term that stops the GUI from making a prediction if no new spectrum is recorded, and there is a status box which was developed to let the user know what the model is currently doing, giving the user an idea of the processes occurring in the background.

The third GUI (Figure 49) was installed in December 2006 on a Progression MM2720 NMR located at the Borealis Polymers facility in Schwechat, Austria. Online validation of the model is currently underway, but the preliminary results show that the RMSEP is 1.23%. The difference in prediction errors between the models validated in the laboratory and the model validated online is due to the limited information contained within the calibration set used for the online model. This makes a strong case for automated model maintenance to control and update the information within the calibration set.

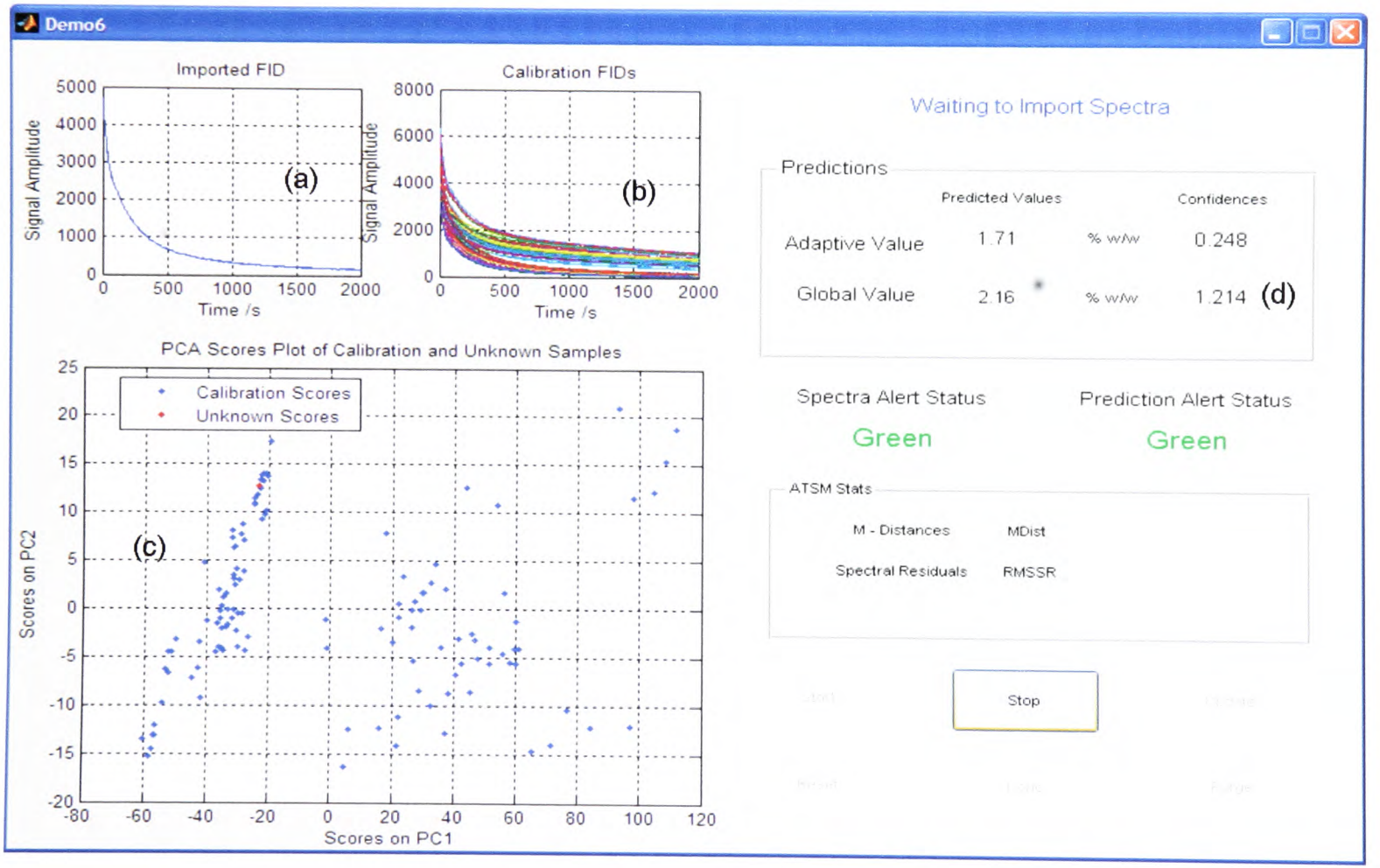

Figure 49. GUI using the Euclidean distance sample selection method. Deployed online in December 2006.

5.1.5.6 Random Sample Selection

The main reason for producing calibration models with random sample selection is to confirm that the sample selection criteria employed in a model is actually the determining factor in its ability to predict. Using randomly selected samples, two models were produced, one using the parameters optimised for the Euclidean distance-based model and the other produced using the parameters optimised for the Shenk and Westerhaus approach. The RMSEP for the Euclidean-based model was 9.84%, while the RMSEP for the Shenk and Westerhaus based model parameters was 11.4%. An example of the randomly selected samples is shown in Figure 50.

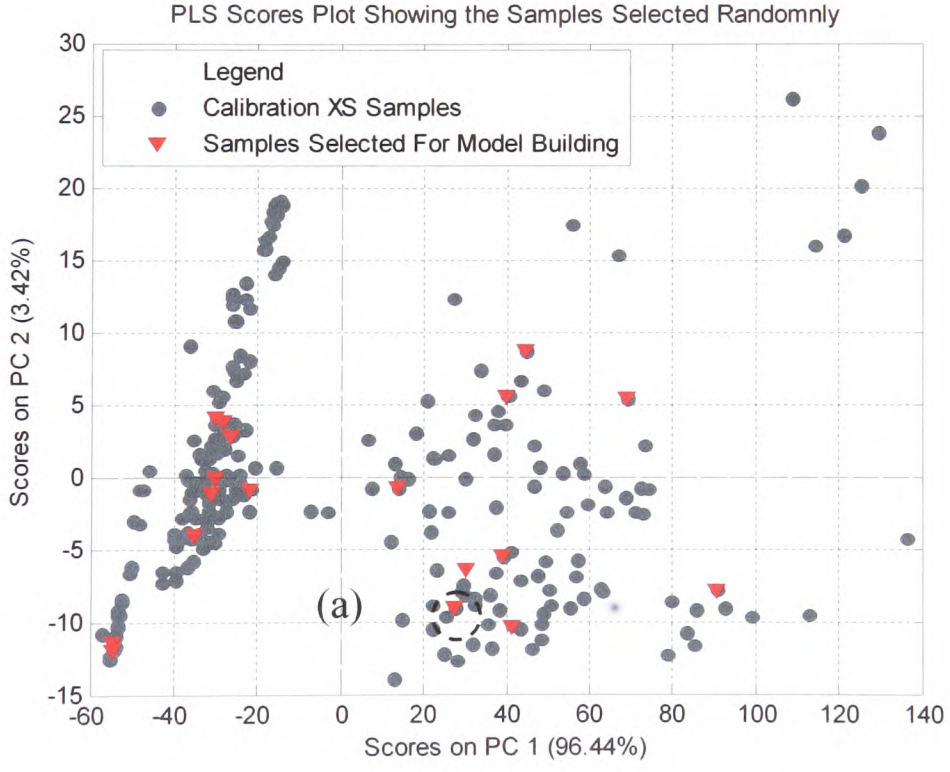

Figure 50. An example using randomly selected samples to build a calibration model for the validation sample (a).

The significantly higher RMSEPs occurring when selecting samples at random demonstrates that the employed selection criteria are essential in the final determination of the prediction ability for the model.

5.1.6 Summary

Table 10 and Figure 51 show the summary of the RMSEC and RMSEP determined from each model. The correlation-based model had a much smaller RMSEC than any of the other models, although it has already been shown that this model is highly over-fit, a theory supported by the ratio of the RMSEP and RMSEC. Of note are the errors for the local models, which were lower than the errors of the current PLS model; however, the complexities that arise due to the need to perform classification along with the inability of this system to handle samples between grades and inliers meant that this method of modelling was not implemented.

Table 10. Summary of the calibration and prediction errors of the differing sample selection methods investigated.

Model Type	RMSEC/%	RMSEP/%
Current PLS Model	1.75	2.15
Local - Low XS Content	0.182	0.379
Local - High XS Content	1.82	2.12
Condition-Based Selection	0.588	1.764
Euclidean-Based Selection	0.348	0.672
Correlation-Based Selection	0.00566	3.58
Random – Euclidean Model		9.84
Random – Correlation Model		11.4

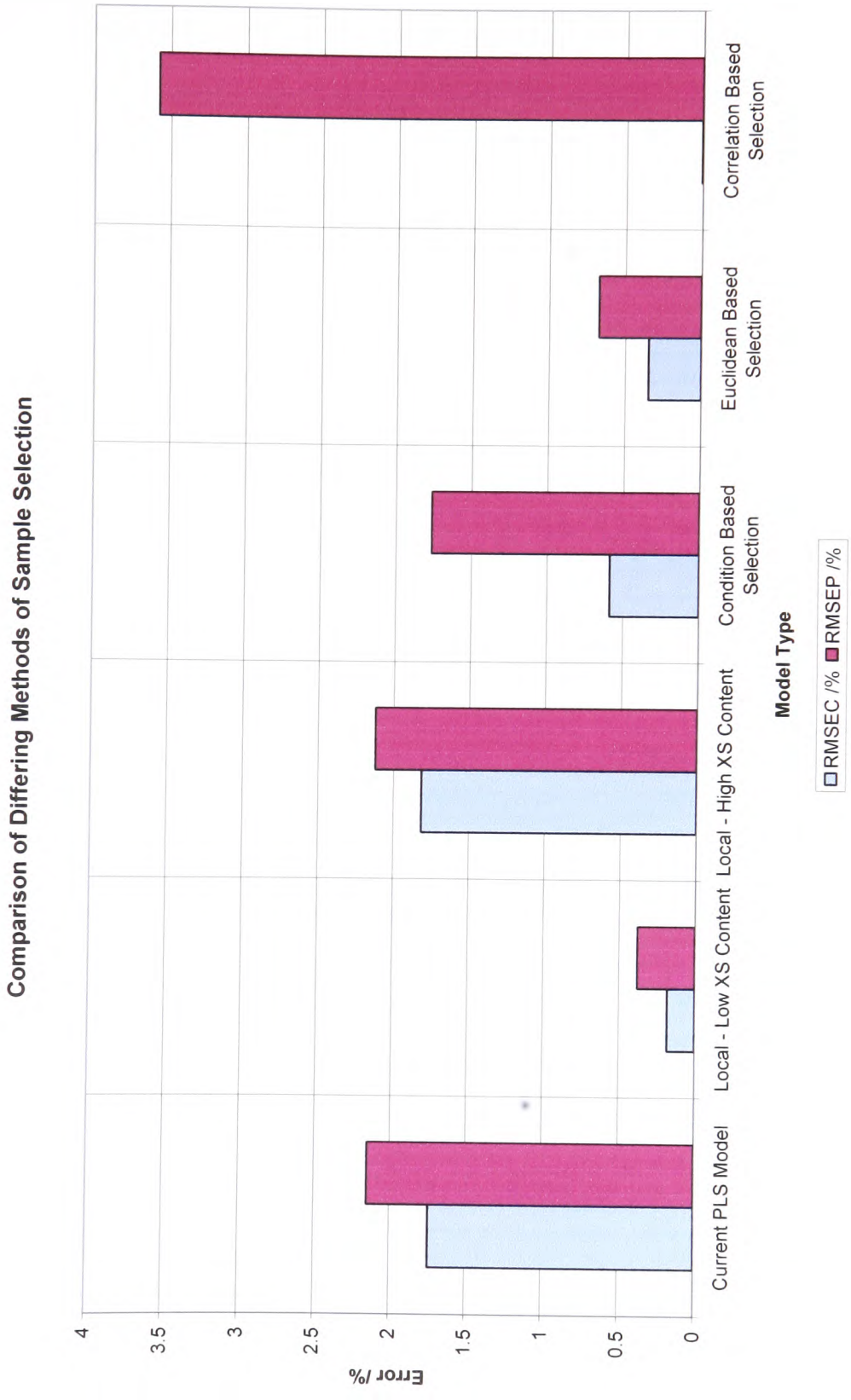

Figure 51. Chart showing the comparison of errors in calibration and validation of the different modelling types.

The success of the Euclidean distance-based model lead to its installation online and it is currently under going a rigorous validation procedure. Upon successful validation, the Euclidean distance-based model and an accompanying automated system of model maintenance will replace the current PLS model being used. The preliminary validation error for the model installed online was determined to be 1.23%.

5.2 Pharmaceutical Tablet Study

The aim of this work was to produce calibration and prediction models for the quality assurance (QA) parameters, active pharmaceutical ingredient (API), and individual tablet weight and thickness of a series of pharmaceutical tablets produced by Pfizer Pharmaceuticals, UK. Using a model to predict these parameters can help the manufacturer save both time and money. To determine the QA parameters of a series of tablets in the traditional laboratory setting is very time-consuming and destructive. By using NIR in conjunction with modelling, each tablet contained within the blister packs can have a prediction of the QA parameters performed efficiently and without the need for in-lab analysis or destruction of the tablets.

The current method of analysis takes the NIR spectra of each tablet after it has been produced (Figure 52). After collection of the spectra a random selection of tablets was taken from the process stream to the laboratory where the thickness and weight are measured and the API content is determined using HPLC.

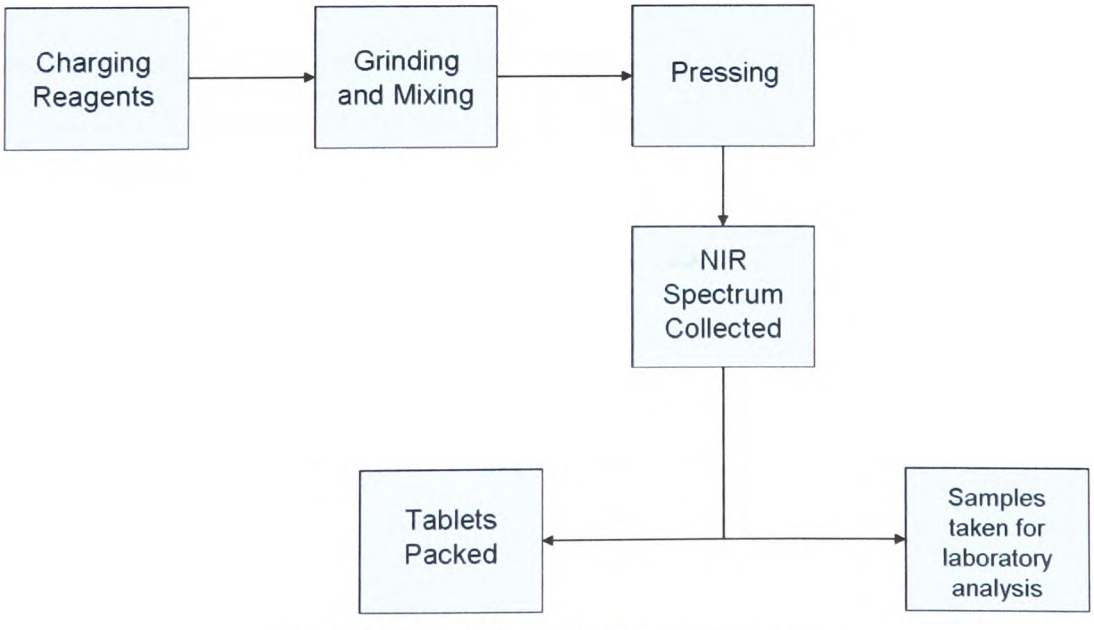

Figure 52. Schematic of a NIR sampling scheme.

Use of the NIR data has one major flaw in that there is an intrinsic variation within the spectra that does not occur within QA parameters (such as tablet thickness and density). This variation is due to the effect of light scattering caused by the differing forms of reflectance, and it is heavily affected by tablet thickness and density. In order to produce a model that can be applied online, any variation due to this light-scattering effect must be accounted for and corrected. A process must be implemented to systematically remove the variation and build a robust online model.

5.2.1 Initial Examination

For the purpose of clarity, the initial examination and modelling will be demonstrated using the reference information for the API, making predictions of the API content.

The initial examination of the data began by inspecting the spectral and reference information. The NIR absorbance spectra, SPT (Figure 53), were reduced to their PCA (Figure 54). The data was split into three groups, (a), (b), and (c). Each group relates to a specific tablet production campaign undertaken in 1997 (a), 1998 (b), or 1999 (c).

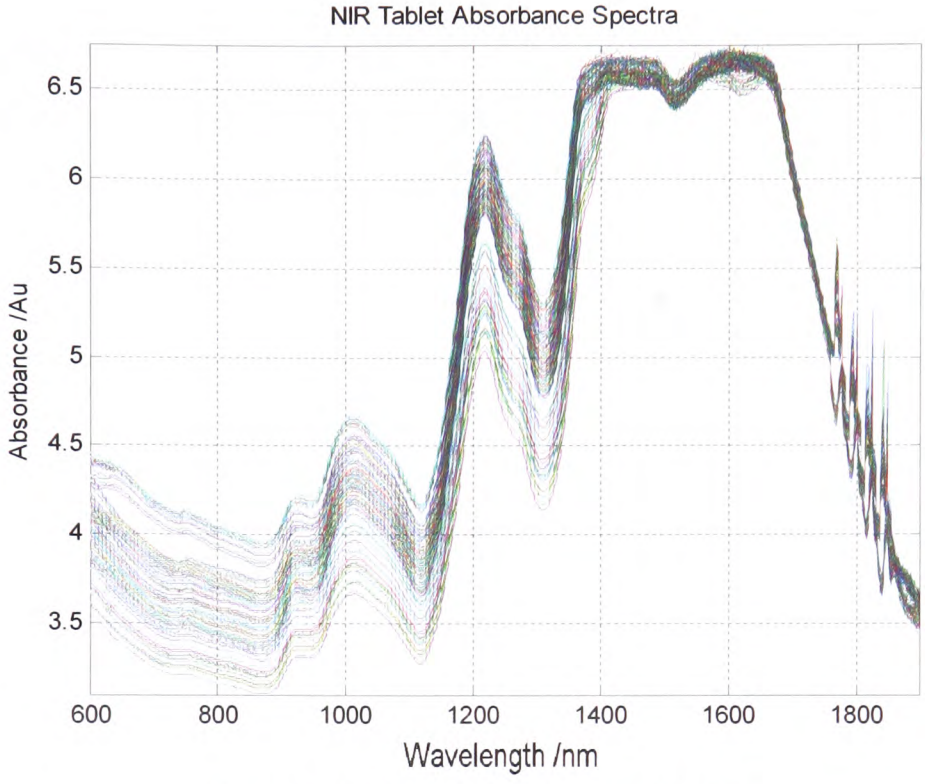

Figure 53. NIR tablet absorbance spectra (SPT).

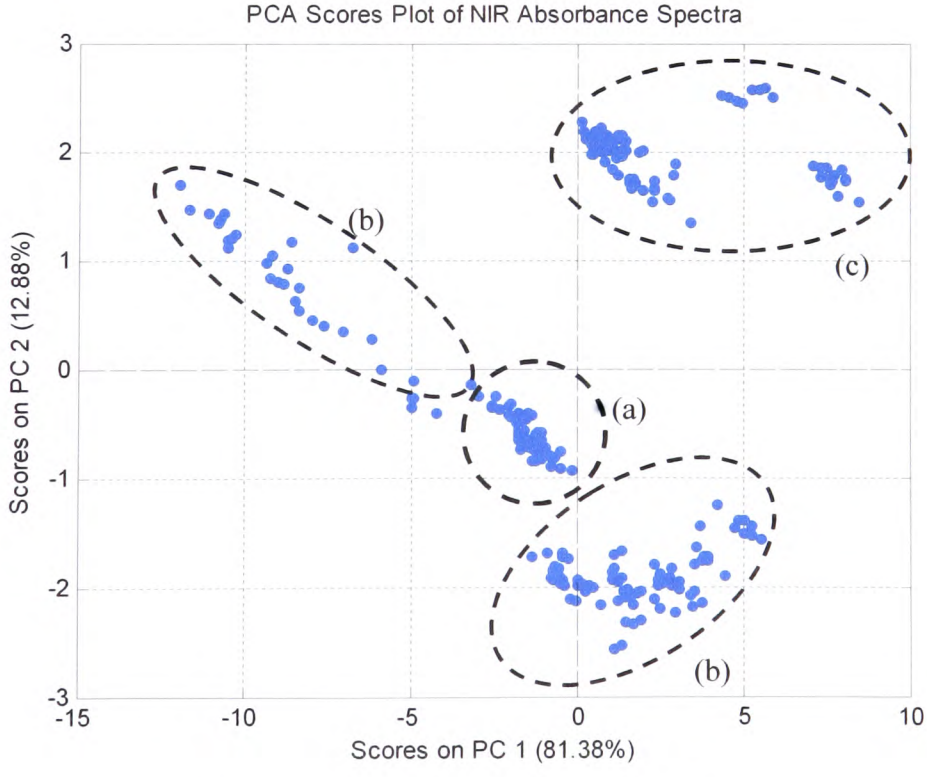

Figure 54. PCA scores plot of SPT showing the tablets produced in 1997 (a), 1998 (b), and 1999 (c).

The distribution of the data was investigated through the use of histograms and normality plots, examples of which are shown in Figure 55 and Figure 56. The histogram in Figure 55 shows the typical shape displayed in a normal distribution. The adherence of the scored points to the straight line in Figure 56 also confirms the normality of the data.

5.2.1.1 Tablet Active Pharmaceutical Ingredient

The relative standard deviation of SPT was calculated to be 2.87%. By comparison, the relative standard deviation for the API was found to be 1.56%, suggesting that there was variation within the spectra that could not be attributed to the variation in the API. As noted previously, one of the major drawbacks of using NIR to measure tablets is the occurrence of diffuse reflectance and light-scattering effects. Thus any models built must include pre-treatment methods (such as EMSC and OSC) that can account for the additional variation or remove problematic wavelengths that contain variation not due to the analytical signal.

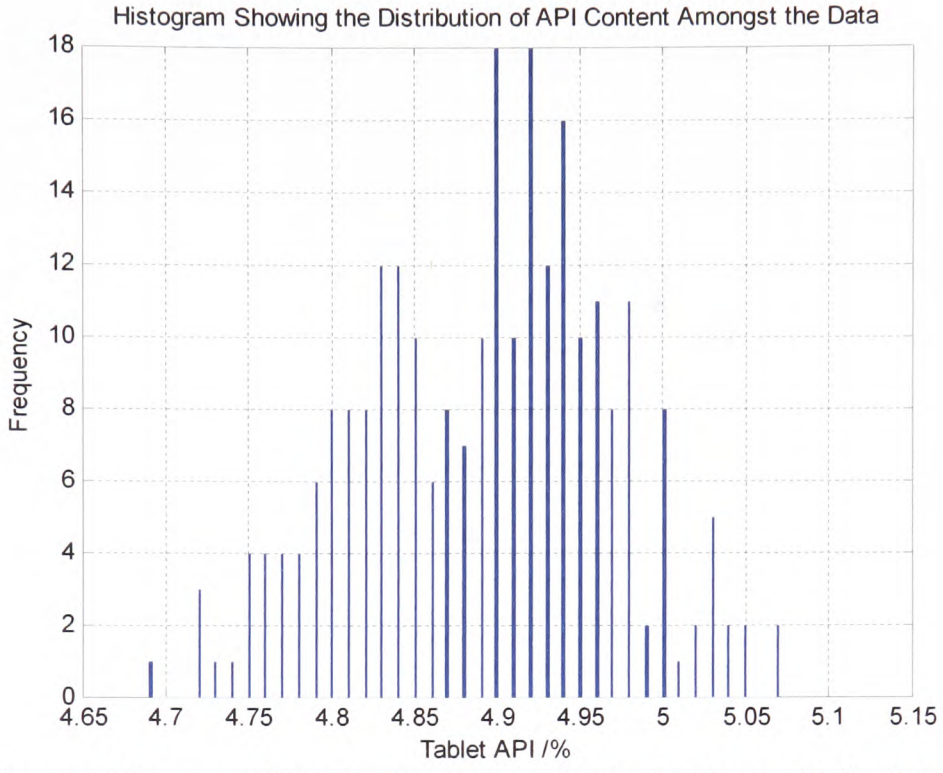

Figure 55. Histogram of API. The shape indicates that the data is normally distributed.

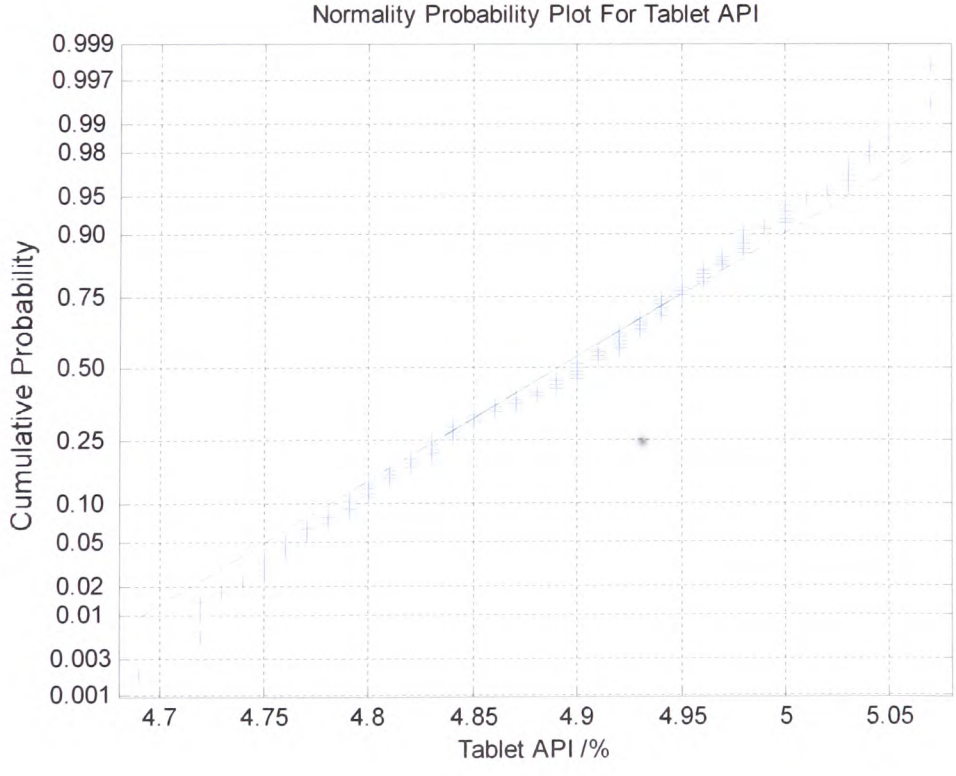

Figure 56. Normality plot for API. Conformation to the straight line confirms normality.

5.2.1.2 Selecting Variables

Figure 57 shows the product from the diagonal of the cross correlation matrix from SPT and API. After wavelength 1400nm the covariance exhibited appears to be random and has no correlation with the variation associated with the API. Subsequently the wavelengths after 1400nm were removed (Figure 58).

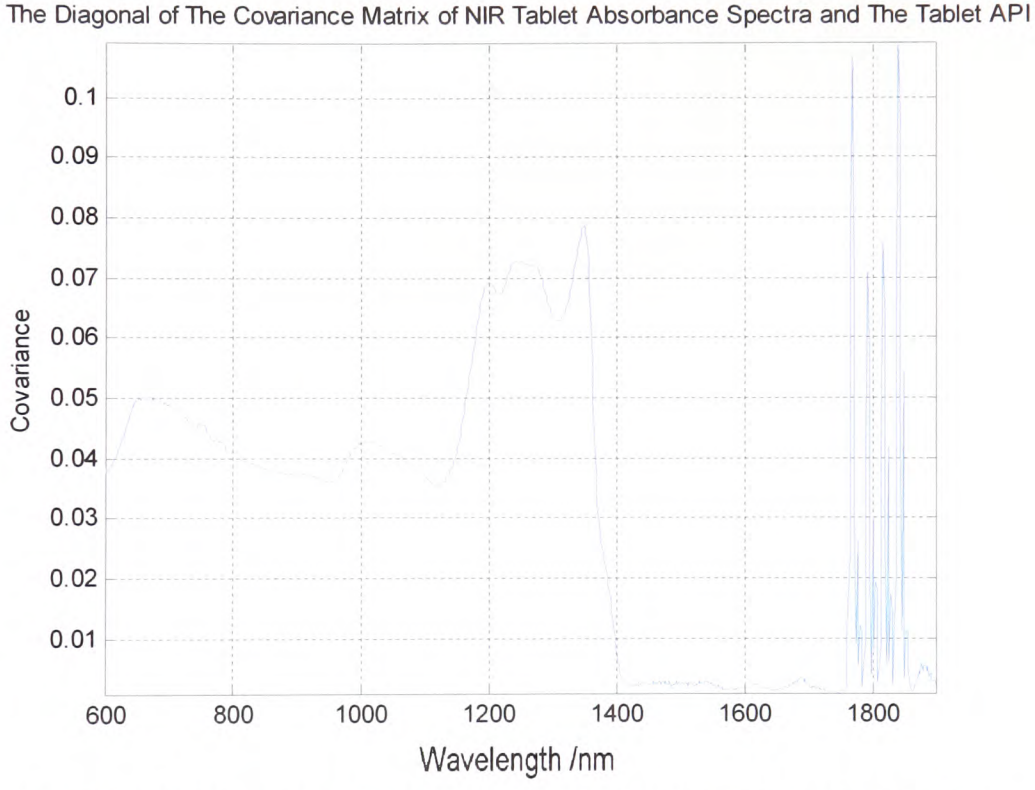

Figure 57. The diagonal response from the cross correlation matrix of SPT and API. The wavelengths after 1400nm do not contribute to the variation associated with API.

Following variable selection, the spectra underwent EMSC (Figure 59). The use of EMSC accounted for the variation of the spectra due to light-scattering effects leaving spectra that no longer exhibit illicit variation. After taking these steps, the relative standard deviation of the spectra was re-calculated to be 0.604%, which was a reduction in the variation within the spectra of approximately 79%.

All of the stages of variable selection and EMSC correction are all performed prior to model building, and will from here on be referred to as hidden layers.

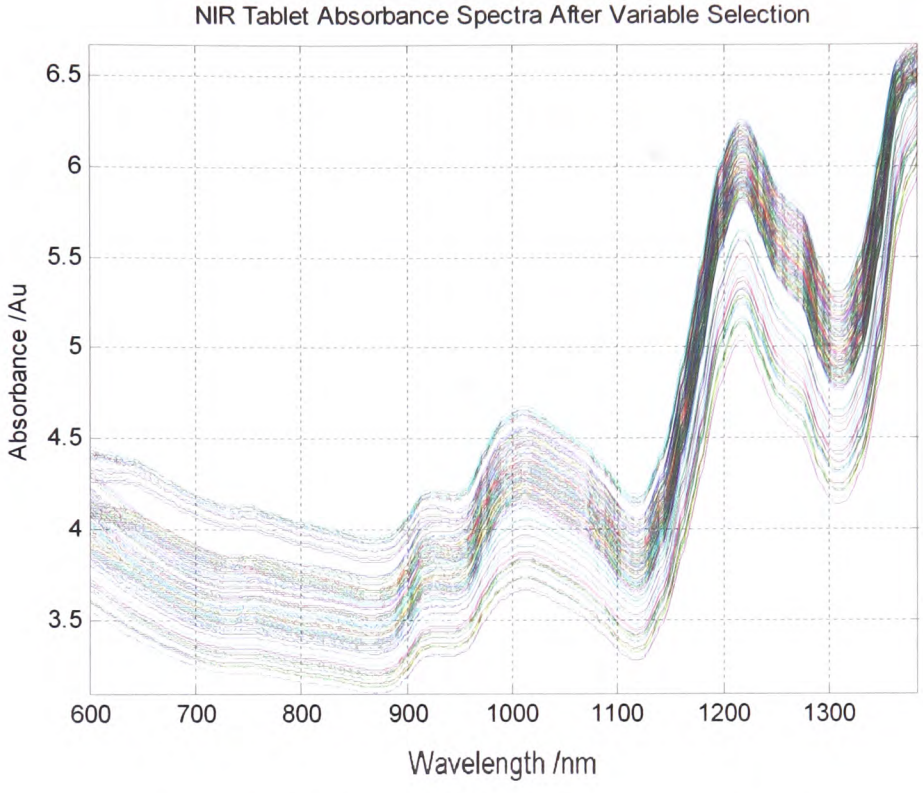

Figure 58. SPT after variable selection has been performed.

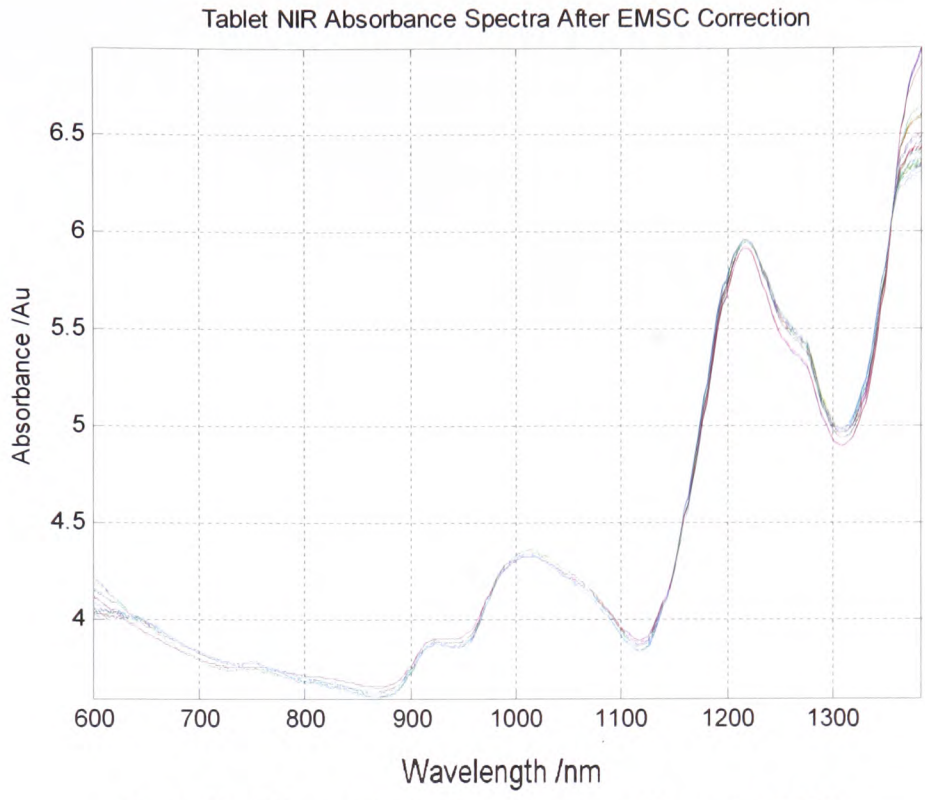

Figure 59. SPT corrected for light-scattering effects using EMSC.

5.2.1.3 Selecting Samples and Pre-processing

Following the application of the hidden layers of calibration, samples were selected using various sample selection methods with criteria such as correlation, condition number, and the Euclidean distance in the scores space.

5.2.1.3.1 Selection Using the Condition Number of the Matrix

Using the condition number to select samples has the effect of building a calibration set that has retained as much variation as possible. Samples selected using the condition number (Figure 60) came from most of the main clusters (excluding (a)), and these samples were taken from a production run from 1998. The samples from (a) did not contribute significant variation to the model and could be removed.

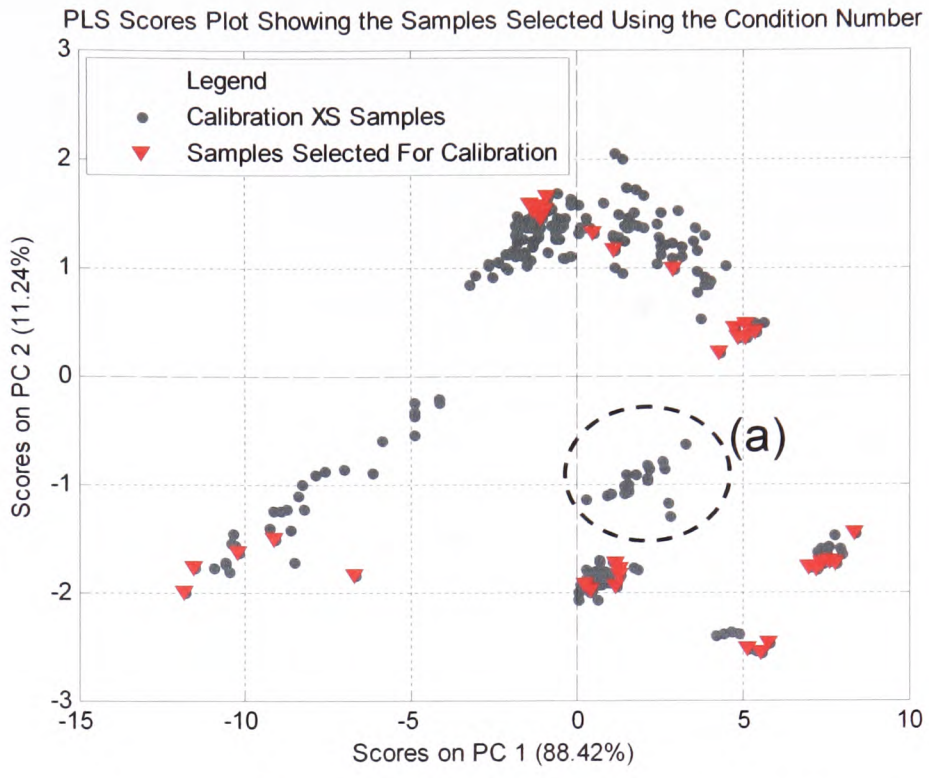

Figure 60. PLS scores plot showing the samples selected using the condition number.

Following the sample selection design of experiments was employed to determine the best method of pre-processing. The best method was found to be mean centring of the data with the incorporation of three latent variables (Figure 61).

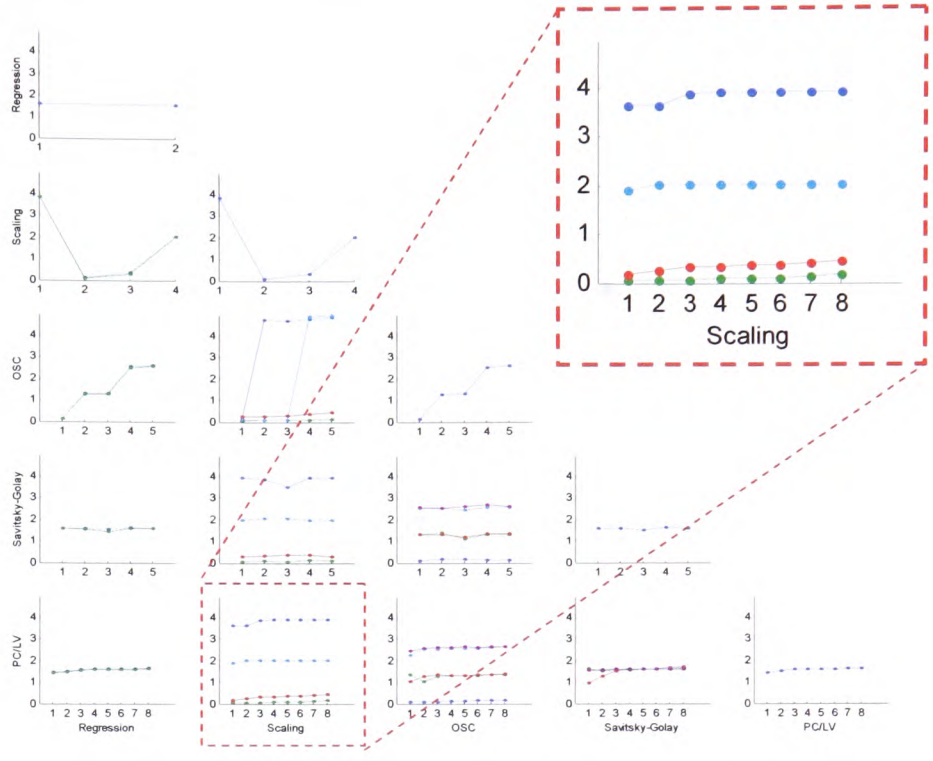

Figure 61. DOE to determine the best method of pre-processing. Inset: the best method selected, mean centring with three LVs.

Following the pre-processing, PLS calibration, and validation modelling (Figure 62, Figure 63), the RMSEC was found to be 0.00598% and the RMSEP was found to be 0.00624%. The ratio between the errors in calibration and prediction was approximately 1.05, suggesting that, despite the very small calibration error, there was no over-fitting of the model. Figure 62 shows that very few the samples fell upon the line of best fit through the predicted and actual values, but the calculated residuals showed that the deviations were small, and hence gave a small calibration and subsequent validation error.

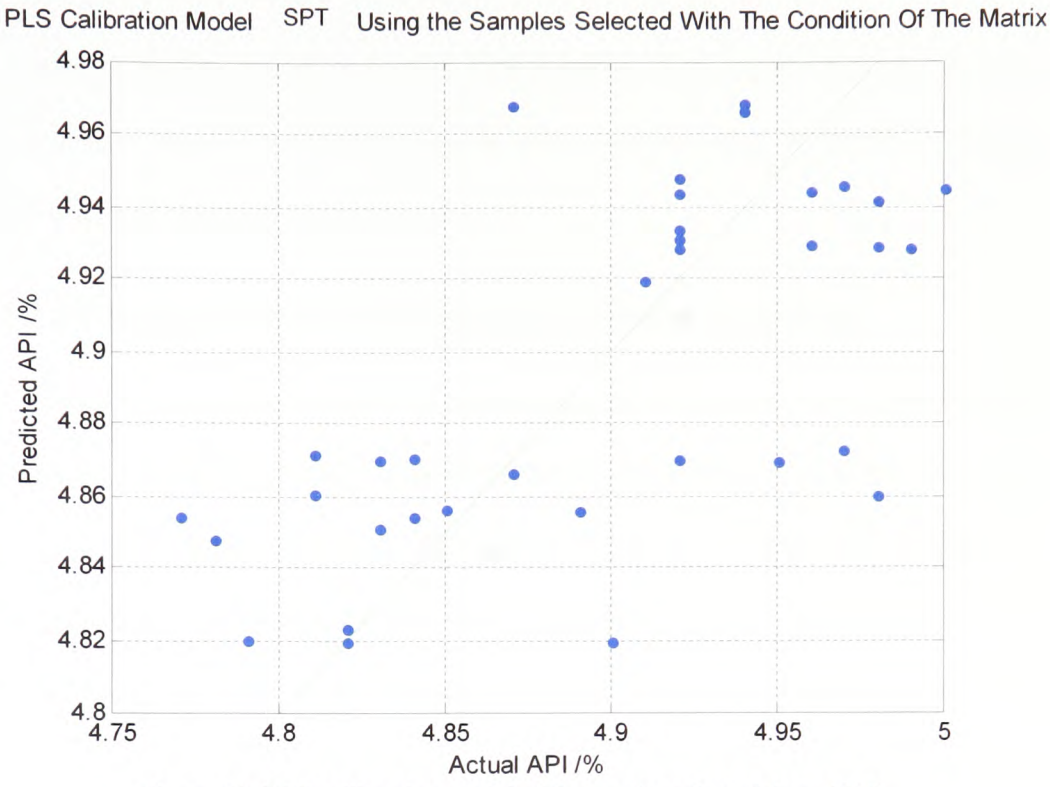

Figure 62. PLS calibration model with samples selected from SPT using the condition number of the matrix; RMSEC = 0.00598%.

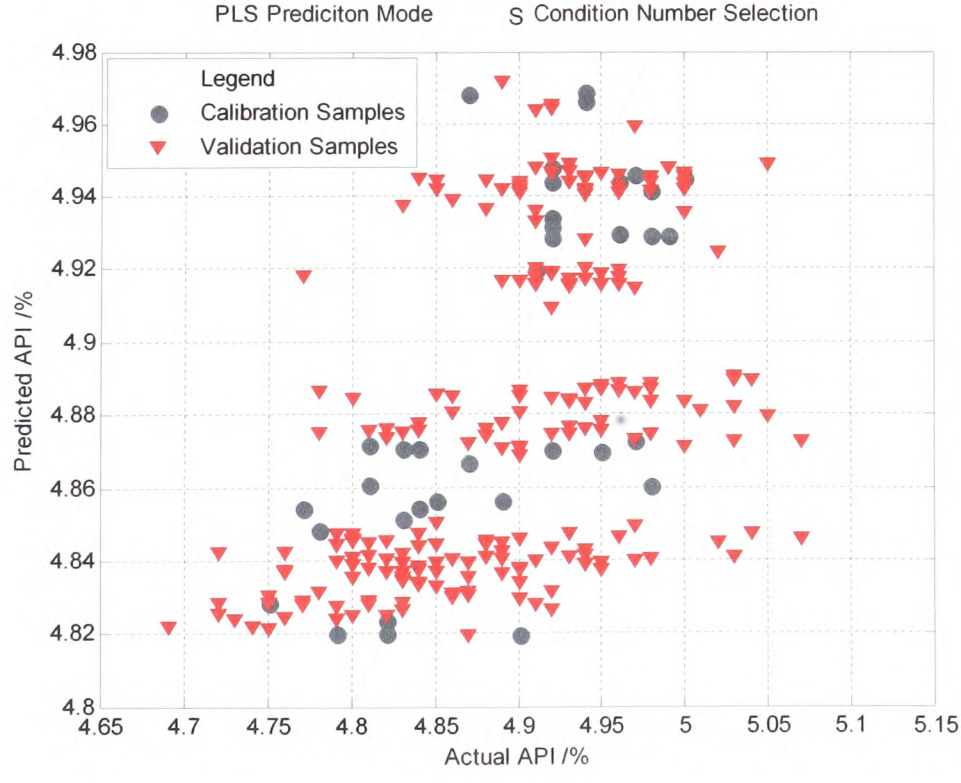

Figure 63. PLS prediction model of the sample remaining after the use of the condition number; RMSEP = 0.00624%.

5.2.1.4 Adaptive Sample Selection

Using adaptive models with SPT will result in models that struggle to predict the reference data. This flaw is due to the reference data, as SPT is normally distributed and has a very small range.

5.2.1.4.1 Sample Selection Using the Correlation of Spectra

Figure 64 shows an example of sample selection using the correlation between calibration and validation spectra as the selection criteria in a manner similar to that employed in the previous section. However, production of subsequent selections for each validation model showed that these same samples were selected every time. This was due to the very small range of variation within the reference data, as this method was developed for reference material with a significant variation of many grades, which would thus be highly correlated. The very small range of variation of the API rendered the data ill-suited to the use of this method of sample selection.

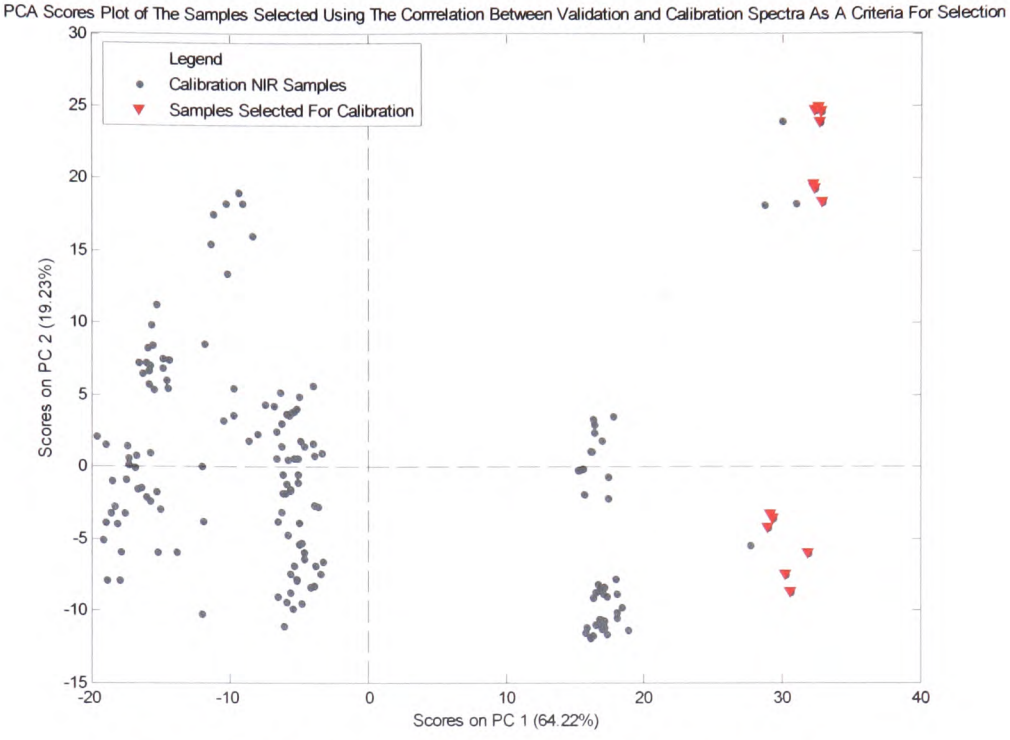

Figure 64. PCA scores plot showing the samples selected using the correlation coefficient as the criteria for selection.

The unsuitability of this sample selection method was further established with the production of the PLS calibration and validation models. Prior to modelling, DOE ascertained that the best method of pre-processing was auto-scaling with the incorporation of three latent variables. Although the same samples were selected each time, the samples selected were representative of the data set as whole and produced models with RMSEC of 0.987% and RMSEP determined to 1.20%. The yield of higher error rates confirmed the unsuitability of this method.

5.2.1.4.2 Sample Selection Using the Euclidean Distance

Figure 65 shows the calibration samples selected from the entire calibration set using the Euclidean distance in the scores space as the selection criteria.

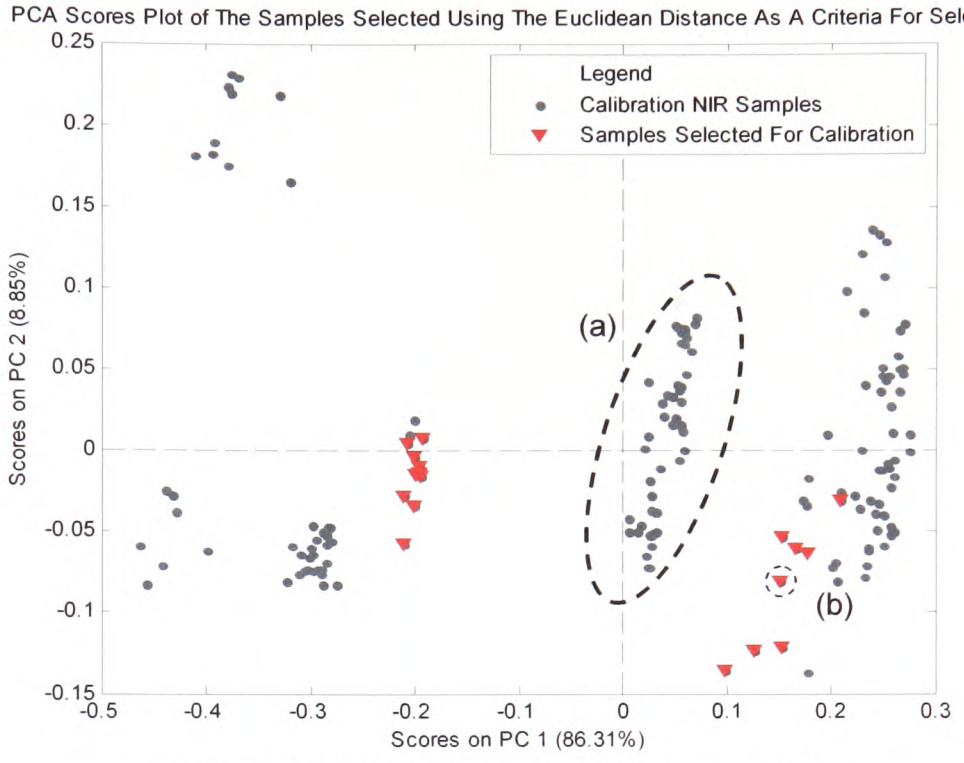

Figure 65. PCA scores plot showing the samples selected for calibration using the Euclidean distance.

In this figure, the distance between the validation samples (b) and sample cluster (a) appeared to be small, yet no samples were selected from this region (a). The reason for this discrepancy lies with the method of visual representation. Because the Euclidean distance method of selection is calculated in three dimensions, the two-dimensional sample plot from Figure 65 needed to be enhanced to accommodate the additional dimension. The new plot (Figure 66) shows the samples selected using Euclidean distance in three dimensions, and it is now observable that the samples in (a) were considerably farther away from the validation sample. This explained why no samples were selected for calibration from this region.

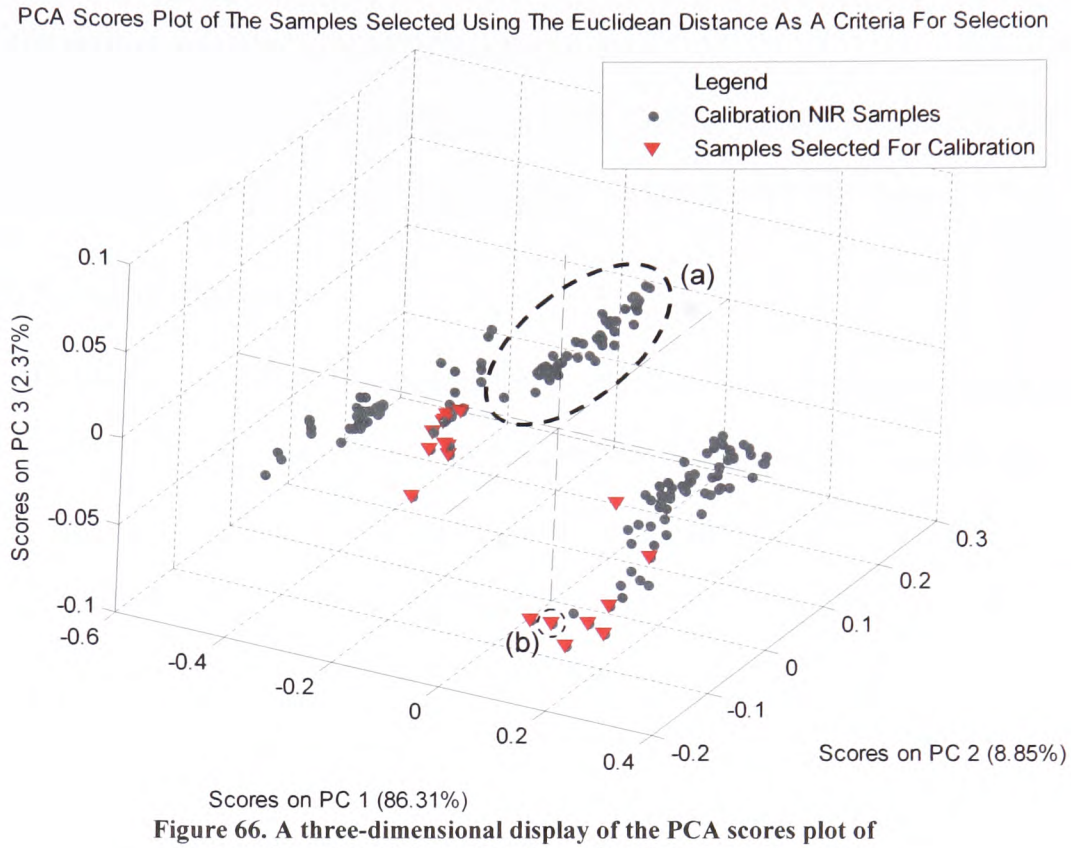

Figure 66. A three-dimensional display of the PCA scores plot of the samples selected for calibration using Euclidean distance.

Similar to the correlation-based sample selection, the Euclidean distance-based method also selected the same calibration samples each time. This was due to the relatively small amount of variation in the reference measurements, and after EMSC correction the variation in the spectra was reduced dramatically. These adaptive modelling systems are designed for in widely-varying systems that require robust multi-modal modelling. As this data did not have these characteristics, the adaptive sampling methods were found to be unsuitable.

The calibration models had a RMSEC of 0.596%, and the subsequent validation models had a RMSEP of 39.7%. The high ratio between calibration and validation error suggested that the model was over-fit; the number of latent variables was then reduced to two which led to RMSEC and RMSEP values of 0.753% and 24.5%, respectively. The fact that ratio between the errors remained very high meant that the

model was not actually over-fit, but was simply poor at making predictions using the Euclidean distance as the criteria for sample selection.

The final assessment of this stage of the modelling determined that the best sample selection method used the condition number as the selection criteria.

5.2.2 Tablet Weight

Following the process undertaken for the API, the same procedure was then used to build models for the parameter pertaining to the weight of each tablet assessed. While EMSC was previously used to correct for light scattering and correlate the reference information to the API, in this situation EMSC was used to remove variation in light scattering and variation due to the API concentration. The remaining variation within the data pertained to the weight of the tablet.

Initial investigation of the tablet weights suggested that the data was not distributed, as evidenced by the two distributive peaks (a) and (b) in the histogram (Figure 67). The non-linearity of the data in the normality plot confirmed this (Figure 68). This reference information more closely resembled POL_Y in its distribution; however, the range was still very small when compared that of POL_Y.

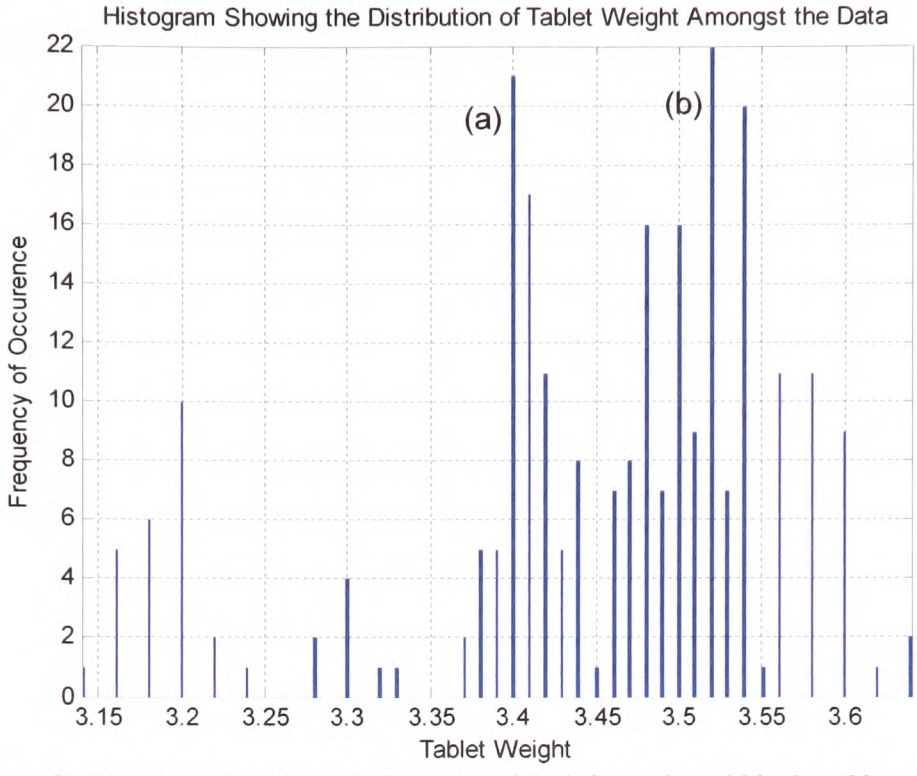

Figure 67. Histogram showing the distribution of the information within the tablet weights.

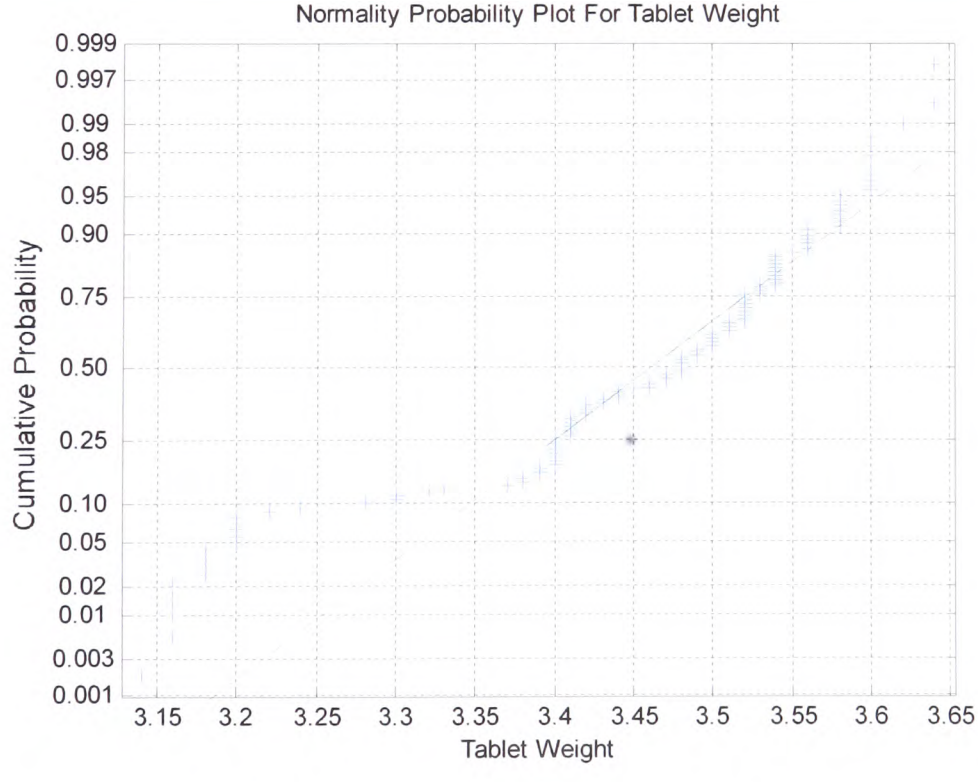

Figure 68. Normality plot for the tablet weight; $R^2 = 0.894$.

5.2.2.1 Modelling

The hidden layers of variable and sample selection were performed as with the API method, and the spectra were cropped at 1400nm. Samples were selected for calibration using the condition number. The data then underwent EMSC correction to account for the variation observed that was not correlated with the variation observed in the tablet weights. The EMSC-corrected data was then used to produce calibration and prediction models in the same manner that was employed with the API models. The calibration model was determined to have a RMSEC of 0.987% and a RMSEP of 1.120% for the prediction model.

5.2.3 Tablet Thickness

5.2.3.1 Initial Study

From the start, making predictions about tablet thickness seemed the least important of the three parameters. However, within industry, of the three laboratory measurements examined, the process of measuring the thickness is the most destructive. Because of this, the use of NIR spectroscopy to predict tablet thickness would save both time and money. As shown in the histogram (Figure 69) and normality plot (Figure 70) from the initial study of the thickness data, the data was normally distributed. This was indicated by the shape of the histogram and the data's adherence to the straight line of the normality plot, with an R^2 of 0.990.

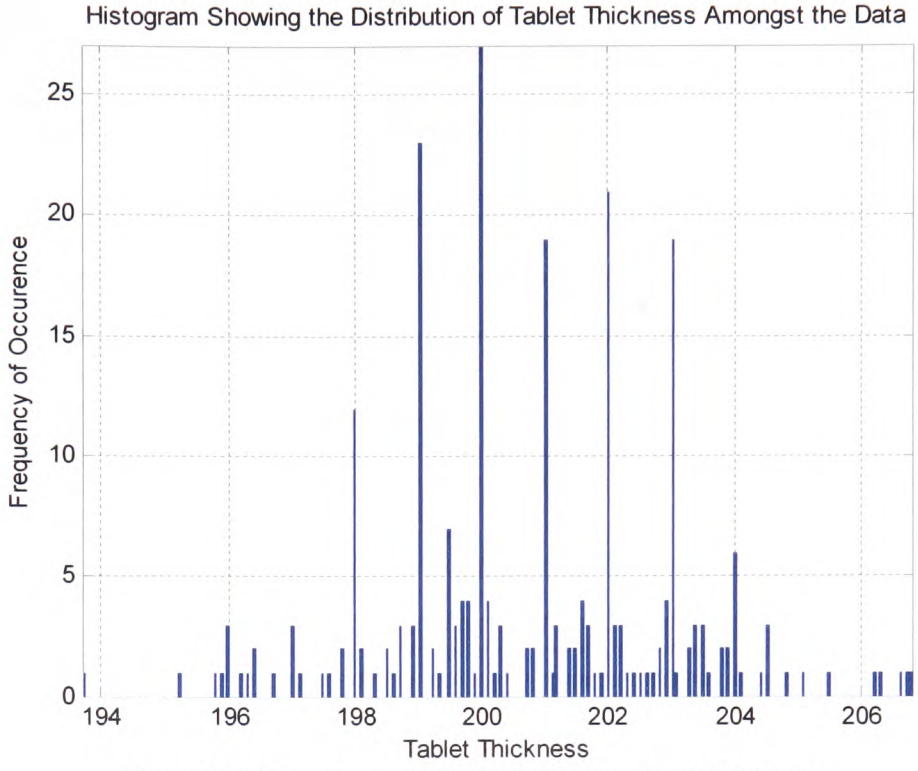

Figure 69. Histogram showing the frequency of tablet thickness. The shape suggests a normal distribution.

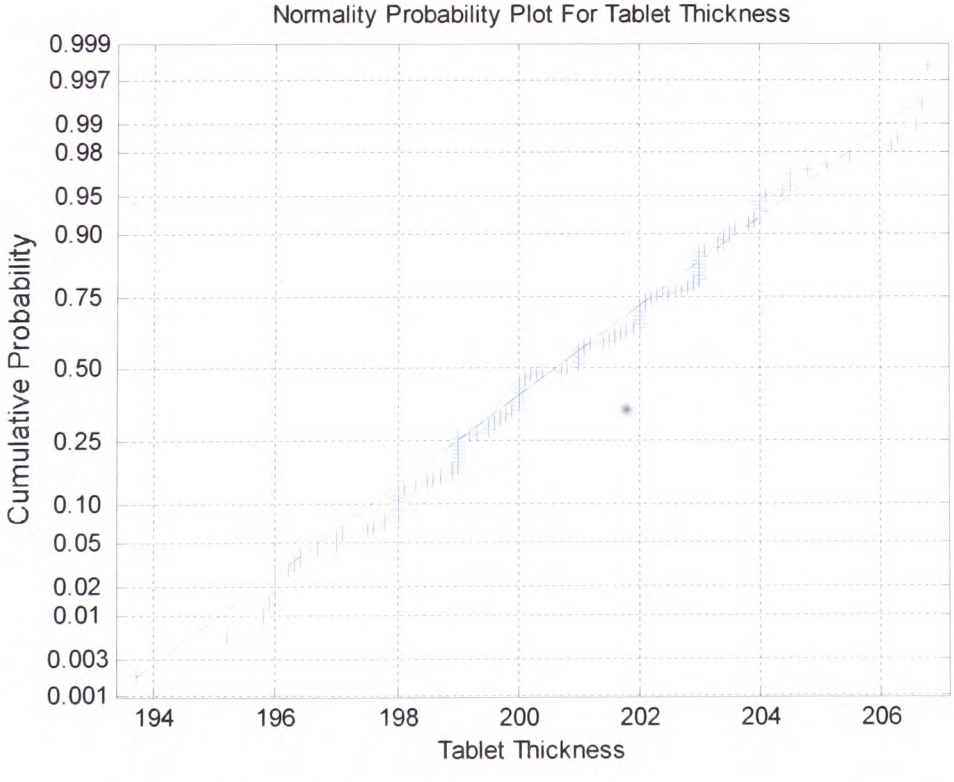

Figure 70. Normality plot of the tablet thickness measurements; $R^2 = 0.990$.

5.2.3.2 Modelling

The prediction of tablet thickness was performed using the same scheme as for API and tablet weight. The processing within hidden layers reduced the spectral variables by removing wavelengths from 1402nm to 1900nm, and samples were selected for calibration using the condition number as the selection criteria.

Again, as with the previous models, the best method for spectral pre-processing was determined using design of experiments; this was determined to be mean centring. The subsequent calibration model had a RMSEC of 1.70%. From the calibration model a prediction model was produced with a RMSEP of 2.46%.

The resulting RMSEP showed that the tablet thickness could be predicted successfully using EMSC correction and hidden layers, proving that this method could replace the destructive methods currently used to measure the tablet thickness.

5.2.4 Blank Models

For comparison and confirmation, control models were built for each of the three prediction parameters. These did not use EMSC or hidden layers. Samples were selected randomly for calibration, while the number of samples and PCs used remained the same as those used in the previous models. Results for the blank models are shown in Table 11. Figure 71 shows the comparison between each model and its respective blank model, including hidden layers and EMSC correction.

Table 11. The calibration and prediction errors of the blank models.

Parameter	RMSEC/%	RMSEP/%
API	0.0993	0.134
Weight	2.29	2.61
Thickness	2.17	2.98

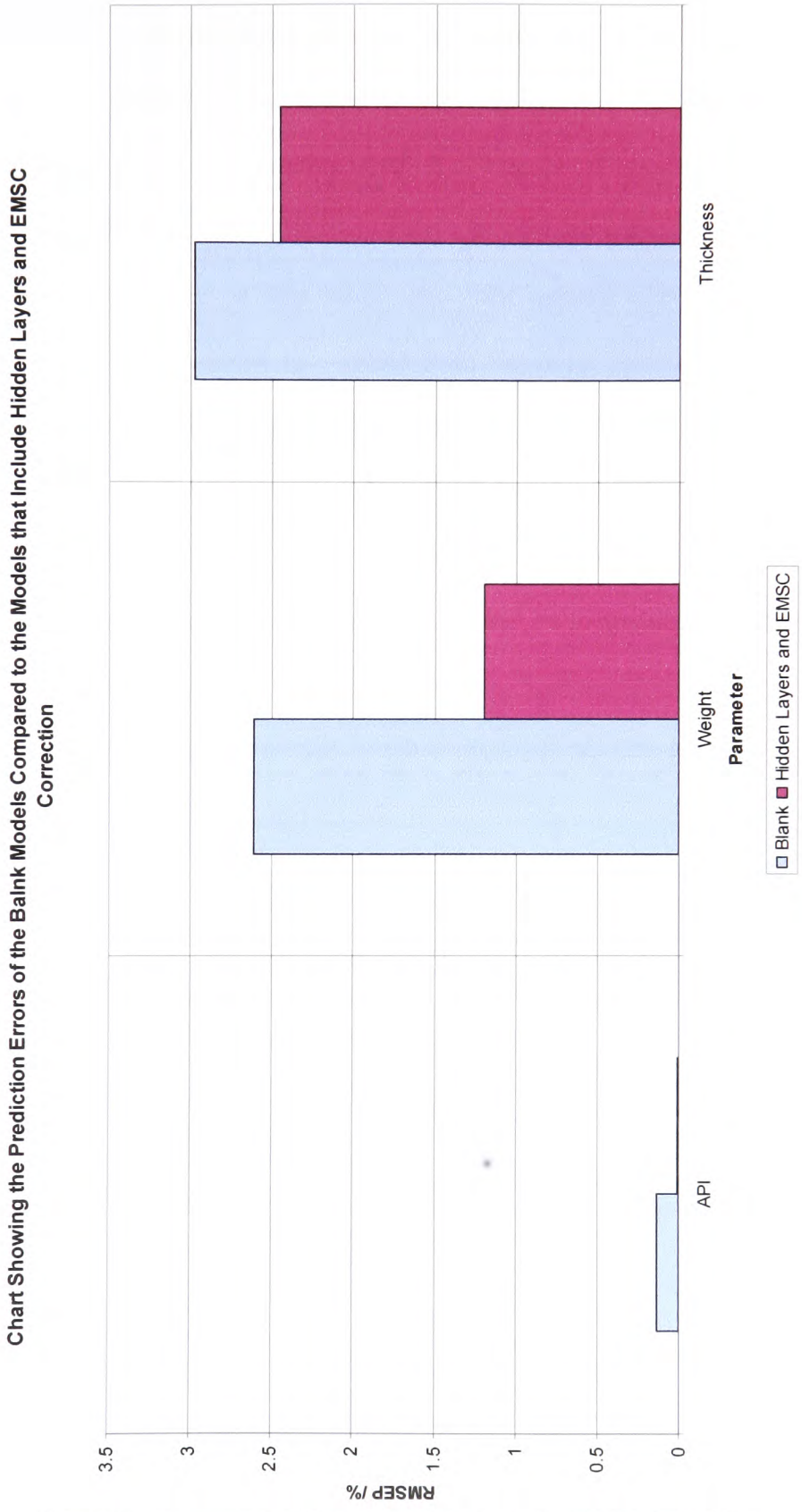

Figure 71. A comparison of the prediction errors for the blank models and the respective models with hidden layers.

The results of this work highlight two key points. The first is that the use of sample selection methods and appropriate pre-processing can be used to produce models that can robustly predict a wide variety of parameters. The second is that the determination of the most appropriate method of sample selection is essential to the success of a model. The method chosen must use a criterion that is suitable for the data under examination. In this study, the NIR spectral data were normally distributed, and samples could be selected from the entire information data space. This is in contrast to the data examined in the polymer section, which were not normally distributed. In that case, a method of sample selection that selected samples from specific regions of the information scores space was required.

The logical next step for the tablet study would be to create a means to apply this modelling scheme online. To accomplish this, aspects of model maintenance must be employed in a similar manner to those employed with the polymer model.

6 Conclusions

6.1 Polymer Study

The main aim of this study was the production of an online robust modelling method that could be used to predict the Xylene Soluble content of polymer pellets.

NMR FID spectra of a series of polymers were collected over a period of ten months. Using the PCA scores, the data could be portioned into two categories based on the XS content. Using the FIDs and the information regarding the XS content, a series of calibration models were produced.

The first part of this study focused on the reproduction of the model being used at that time. This model was a global model that used all of the samples, and the RMSEP for the prediction of XS content with this model was 2.15%. This model gave a baseline performance to which the performance of subsequent models could be compared. After the PCA the data appeared to be bimodal and this lead to the development and production of two local models for samples with high and low XS content. The prediction errors for the local models for high and low samples were 2.12% and 0.379%, respectively. Although the prediction errors of these models were better than those of the model then employed online, this system was rejected due to its need for an additional stage of classification before predictions could be made. However, this part of the study did show that each mode of the data (in this case, each grade of polymer being produced) must be dealt with separately to produce good predictions.

Sample selection methods that combined the strengths of the global model with the strengths of the local models were then developed and employed, and samples

were selected from each grade within a global set of data. The sample selection routines selected samples for calibration based on the sample to be predicted. Of the methods investigated, sample selection based on the PCA scores and the Euclidean distances resulted in the best prediction models with a RMSEP of 0.672%. Although the prediction error for this model was greater than that of the local system, the use of this selection method required no form of classification prior to making predictions. Furthermore, the use of this form of sample selection allowed for tracking of in-lying samples that move between grades and monitoring as the process cycle moves from one grade to another.

The final stage of this study was the development and deployment of a user interface at the point of analysis that incorporated the model using Euclidean distance-based sample selection. The resultant GUI was installed in December 2006.

The next step in this study would be to produce an automated method of model maintenance that would ensure that only the most pertinent samples were retained in the model. Automated model maintenance would allow the model to determine the times when a laboratory measurement should be taken. The current method of model maintenance requires a scheduled analysis of samples collected three times a day. Each sample is then added to the model, regardless of whether any this sample contributes any additional information to the model. Design of experiments could be used employing an E-optimal criterion to ensure that any updates to the calibration data set involve only the most informative samples. Automating this process would also decrease laboratory analysis costs. If the model can determine the accuracy of its predictions then no manual reference measurements are needed, making the sampling procedure a proactive initiative.

The adaptive sampling algorithms could also be expanded to be applied to any process that works within a production cycle and requires grade-specific predictions, such as the prediction of aromatic and olefin content of petroleum and diesel. Any system that contains sampling clusters due to the reference measurements would be an ideal arena for the application of these procedures for sample selection.

6.2 Tablet Study

The main aim of the tablet study was to produce a robust model that could account for variations in the analytical signal that were not caused by variations within the tablet. This study also involved a traditional method of PAT which made it ideal for evaluating the modelling methods developed in the polymer study.

The NIR spectra of over 250 tablets were collected over three production campaigns from 1997 to 1999. Accompanying the NIR spectral data were the chemical and physical tablet parameters for the active pharmaceutical ingredient, tablet weight, and tablet thickness.

This study began with the PCA of the data which showed the data distributed in accordance with the three production campaigns. Unlike the polymer study, the variation observed was not due to differing grades of tablets being produced, but instead due to diffuse and specular reflectance of the NIR radiation from the surface of the tablet. The reflectance variation was addressed using EMSC and variable selection. Three models (one for each of the tablet parameters) were produced, and the best predictive models were constructed with samples selected using the condition number. The prediction errors for these models are in Table 12.

Table 12. Calibration and prediction errors for the tablet study.

Parameter	RMSEC/%	RMSEP/%
API	0.00598	0.00624
Thickness	0.9874	1.20
Weight	1.70	2.46

The ability to make the predictions of the tablet API, weight, and thickness from one spectrum would save time and money in a process environment, improving upon the sampling procedure so that fewer samples need be destroyed for the purpose of analysis. Control and maintenance of an automated model could also convert sampling from a scheduled practice to a proactive one, using the model to determine when tablets should sampled in order to improve predictions and robustness. Additionally, automated NIR spectroscopy in PAT provides a practical means to analyse every table from the production line, and the ability to control the information in the calibration set using an E-optimal approach (as in the polymer study) would ensure that the calibration data set only includes relevant samples.

The next logical step for this study would be the development of a user interface and then installation online at the point of analysis, as accomplished in the polymer study.

6.3 Summary

From the results of these two experimental studies it has been demonstrated that the use a regimented and designed procedure to determine criteria for sample selection, correction methods, and data pre-treatment procedures will result in the creation of robust, accurate models. The polymer study evolved the use of sample selection algorithms based upon the actual sample being predicted, and these models successfully predicted polymer samples, but performed poorly in the prediction of the NIR data. This discrepancy was attributed to intrinsic differences in the data being analysed, which emphasised the fact that there is no one standard approach to data analysis. The successful use of chemometrics and design of experiments to determine the best method for modelling in both studies indicates that this combination of methods should perhaps be established as the standard approach.

The next step for both studies is to employ design of experiments to maintain the calibration models, ensuring that they do not grow exponentially and maximising the amount of relevant information retained. Further work within the model will allow the modelling system to determine if and when reference measurements are needed and if a decline in the quality of predictions requires a laboratory measurement. This would replace the traditional manual sampling procedure so that samples are taken and laboratory references measurements are recorded only when the model deems it necessary.

This work shows that by delving deeper into modelling strategies and employing appropriate sample and variable selections with the analytical application of design of experiments result in better models capable of make better predictions. Advancements in PAT must be accompanied by complementary advances in chemometrics to ensure that both remain at the forefront of analytical science.

7 Self Reflection and Appraisal

The section details the aspects of my personal development over the past three years that lead to the successful completion of this research.

When I started my PhD in 2004, I had a basic understanding of the principles and applications of chemometrics. I also had a working knowledge of Excel and limited experience with MatLab. Three years later, through immersion in challenging and enjoyable research, I was pushed to develop new skills and advance beyond my expectations. One of the most important skills I gained was the ability to produce algorithms and programmes with MatLab. During the last three years I wrote and developed a large number of programmes, the most important of which being the user interface that is currently employed by Borealis to predict various parameters of the polymers they produce. This project forced a significant shift of my internal paradigm as I evolved from simply being a user, a button pusher, and embraced a new philosophy when developing programmes – that of an artist. This development took a lot of hard work and patience, and it reminded me of the first steps in learning a foreign language; but the results bore a programme that is now in use online at a major manufacturing facility.

This accomplishment required both an understanding of programming itself and an understanding of the people who would use the programme. I gained the necessary insight into the people who would use (and ultimately benefiting from) the software in development when I spent a month at the Borealis plant in Schwechat, Austria. My time there was spent writing code and working in the laboratory where I performed the analysis methods used to generate reference information for the models. This gave me a deep appreciation for the work involved; previously had I craved and demanded data, but upon returning from Austria I realised that the

reference measurements were only a small part of a bigger picture. The time in Austria also allowed me the opportunity to interact with and learn from the people who would be the primary users of my software, and their feedback lead to the implementation of a traffic light system to indicate the quality of the model predictions. To this I would look to implement an on-demand sampling procedure, so that when an inaccurate value was predicted by the model a system would be initiated to collect a sample, record a spectrum, and call for a laboratory reference measurement. The newly-recorded sample would then be added to the model using a maintenance algorithm with an E-optimal approach. This method could also be used to identify and remove samples that no longer add sufficient information within the framework of the model. The model maintenance algorithm would control the size of the model and prevent it from expanding, thus keeping the system information-rich, as opposed the data-rich, information-poor state that trap models of ever increasing sizes.

In the past three years I have found that to people outside of the field chemometrics appears to be some unintelligible form of black magic. To this end, I have made an effort to communicate my work to other scientists through posters and talks at conferences. I also place great importance on the demonstration of chemometrics to undergraduate students, as the demonstration classes have provided me with a means to increase awareness of and enthusiasm for the field, and hopefully inspire some potential future chemometricians. Demonstrating in chemometrics has required me to reconsider the field as it is seen by the uninitiated in order to be able to communicate the fundamental theories and principles of chemometrics to students who likely have no prior experience with this form of data

analysis. Breaking down fundamental concepts such as PCA has served to ensure that I myself have a thorough understanding of the theories and practices of my field.

In addition to my time in Austria, I also had the opportunity to spend three months in Seattle working for the Center for Process Analytical Chemistry (CPAC) at the University of Washington. My time there was spent working on a project based on calibration transfer between gas chromatography instruments in different parts of the world. This work again added another string to my bow as I experienced working within a new group, one with different ideas and expectations; additionally, I had to adjust as I worked with an entirely different form of data. As a whole, I feel that my experiences over the past three years have allowed me to develop skills that will be indispensable throughout the entirety of my career.

8 References

[1] B. Kowalski and F. McLennan, *Process Analytical Chemistry*, Blackie Academic & Professional, 1995, p. 378.

[2] A. Nordon, C. A. McGill and D. Littlejohn, *Analyst* 2001, *126*, 260-272.

[3] W. F. McClure, *Journal of near Infrared Spectroscopy* 2003, *11*, 487-518.

[4] R. Wellock and A. Walmsley, *Spectroscopy Europe* 2006, *23*, 4.

[5] A. Nordon, Y. Carella, A. Gachagan, D. Littlejohn and G. Hayward, *Analyst* 2006, *131*, 323-330.

[6] A. Nordon, R. J. H. Waddell, L. J. Bellamy, A. Gachagan, D. McNab, D. Littlejohn and G. Hayward, *Analyst* 2004, *129*, 463-467.

[7] U. D. o. H. a. H. Services and FDA in *Guidance for Industry. PAT - A Framework for Innovative Pharmaceutical Development, Manufacturing and Quality Assurance*, *Vol.* 2004.

[8] A. de Juan and R. Tauler, *Critical Reviews in Analytical Chemistry* 2006, *36*, 163-176.

[9] S. Wold, H. Antti, F. Lindgren and J. Ohman, *Chemometrics and Intelligent Laboratory Systems* 1998, *44*, 175-185.

[10] A. Kohler, C. Kirschner, A. Oust and H. Martens, *Applied Spectroscopy* 2005, *59*, 707-716.

[11] D. K. Pedersen, H. Martens, J. P. Nielsen and S. B. Engelsen, *Applied Spectroscopy* 2002, *56*, 1206-1214.

[12] S. N. Thennadil, H. Martens and A. Kohler, *Applied Spectroscopy* 2006, *60*, 315-321.

[13] W. F. McClure, *Journal of near Infrared Spectroscopy* 2003, *11*, 487-518.

[14] M. Blanco, A. C. Peinado and J. Mas, *Analytica Chimica Acta* 2005, *544*, 199-205.

[15] A. Dunko and A. Dovletoglou, *Journal of Pharmaceutical and Biomedical Analysis* 2002, *28*, 145-154.

[16] R. M. Garcia-Rey, J. Garcia-Olmo, E. De Pedro, R. Quiles-Zafra and M. D. Luque de Castro, *Meat Science* 2005, *70*, 357-363.

[17] O. R. Dumitrescu, D. C. Baker, G. M. Foster and K. E. Evans, *Polymer Testing* 2005, *24*, 367-375.

[18] J. Kraunsoe in *The Role of PAT in Developing New Products, Vol.* Astra Zeneca, 2005.

[19] J. Rantanen, H. Wikstrom, R. Turner and L. S. Taylor, *Analytical Chemistry* 2005, *77*, 556-563.

[20] A. Thomson in *Making Light Work Online, Vol.* BP - Hull, 2005.

[21] W.-D. Hergerth in *Industrial Polymerisation Monitoring, Vol.* 2005.

[22] S. Macho and M. S. Larrechi, *Trac-Trends in Analytical Chemistry* 2002, *21*, 799-806.

[23] R. Bro, J. J. Workman, P. R. Mobley and B. R. Kowalski, *Applied Spectroscopy Reviews* 1997, *32*, 237-261.

[24] J. Workman, Jerry, *Chemometrics and Intelligent Laboratory Systems* 2002, *60*, 13-23.

[25] J. Workman and J. Brown, *Spectroscopy* 1996, *11*, 48-51.

[26] J. J. Workman, *Applied Spectroscopy Reviews* 1996, *31*, 251-320.

[27] J. J. Workman, P. R. Mobley, B. R. Kowalski and R. Bro, *Applied Spectroscopy Reviews* 1996, *31*, 73-124.

[28] R. Brereton, *Chemometrics*, Wiley, 2003, p.

[29] A. H. Aastveit and H. Martens, *Biometrics* 1986, *42*, 829-844.

[30] K. H. Esbensen and H. Martens, *Chemometrics and Intelligent Laboratory Systems* 1987, *2*, 221-232.

[31] H. Martens, L. Izquierdo, M. Thomassen and M. Martens, *Analytica Chimica Acta* 1986, *191*, 133-148.

[32] M. L. Bisani, D. Faraone, S. Clementi, K. H. Esbensen and S. Wold, *Analytica Chimica Acta* 1983, *150*, 129-143.

[33] D. Johnels, U. Edlund, E. Johansson and S. Wold, *Journal of Magnetic Resonance* 1983, *55*, 316-321.

[34] W. Lindberg, J. Ohman, S. Wold and H. Martens, *Analytica Chimica Acta* 1985, *174*, 41-51.

[35] W. Lindberg, J. Ohman, S. Wold and H. Martens, *Analytica Chimica Acta* 1985, *171*, 1-11.

[36] W. Lindberg, J. A. Persson and S. Wold, *Analytical Chemistry* 1983, *55*, 643-648.

[37] M. Martens, H. Martens and S. Wold, *Journal of the Science of Food and Agriculture* 1983, *34*, 715-724.

[38] M. Sjostrom, S. Wold, W. Lindberg, J. A. Persson and H. Martens, *Analytica Chimica Acta* 1983, *150*, 61-70.

[39] S. Wold, A. Ruhe, H. Wold and W. J. Dunn, *Siam Journal on Scientific and Statistical Computing* 1984, *5*, 735-743.

[40] P. J. Gemperline, *Chemometrics and Intelligent Laboratory Systems* 1992, *15*, 115-126.

[41] S. J. Qin and T. J. McAvoy, *Computers & Chemical Engineering* 1992, *16*, 379-391.

[42] V. M. Taavitsainen and P. Korhonen, *Chemometrics and Intelligent Laboratory Systems* 1992, *14*, 185-194.

[43] S. Wold, *Chemometrics and Intelligent Laboratory Systems* 1992, *14*, 71-84.

[44] I. N. Wakeling and H. J. H. Macfie, *Journal of Chemometrics* 1992, *6*, 189-198.

[45] S. Dejong, *Chemometrics and Intelligent Laboratory Systems* 1993, *18*, 251-263.

[46] T. Naes, T. Isaksson, T. Fearn and T. Davies, *Multivariate Calibration and Classification*, NIR Publications, 2002, p. 344.

[47] R. Kellner, *Analytical Chemistry*, 1998, p. 916.

[48] P. Gemperline, *Practical Guide to Chemometrics*, CRC, 2006, p. 541.

[49] P. Berzaghi, J. S. Shenk and M. O. Westerhaus, *Journal of Near Infrared Spectroscopy* 2000, *8*, 1-9.

[50] D. Perez-Marin, A. Garrido-Varo and J. E. Guerrero, *Applied Spectroscopy* 2005, *59*, 69-77.

[51] J. S. Shenk, M. O. Westerhaus and P. Berzaghi, *Journal of Near Infrared Spectroscopy* 1997, *5*, 223-232.

[52] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond and N. Jent, *Journal of Pharmaceutical and Biomedical Analysis* 2007, *44*, 683-700.

[53] Y. Roggo, L. Duponchel, C. Ruckebusch and J. P. Huvenne, *Journal of Molecular Structure* 2003, *654*, 253-262.

[54] I. S. Helland, T. Naes and T. Isaksson, *Chemometrics and Intelligent Laboratory Systems* 1995, *29*, 233-241.

[55] T. Fearn, *Chemometrics and Intelligent Laboratory Systems* 2000, *50*, 47-52.

[56] J. Sjoblom, O. Svensson, M. Josefson, H. Kullberg and S. Wold, *Chemometrics and Intelligent Laboratory Systems* 1998, *44*, 229-244.

[57] A. Rudnitskaya, I. Delgadillo, A. Legin, S. M. Rocha, A. M. Costa and T. Simoes, *Chemometrics and Intelligent Laboratory Systems* 2007, *88*, 125-131.

[58] A. J. Myles, T. A. Zimmerman and S. D. Brown, *Applied Spectroscopy* 2006, *60*, 1198-1203.

[59] A. Savitsky and M. Golay, *Analytical Chemistry* 1964, *36*, 1627 - 1639.

[60] J. Steiner, J. Deltour and Y. Termonia, *Analytical Chemistry* 1972, *44*, 1906 - 1909.

[61] T. Naes, C. Irgens and H. Martens, *Applied Statistics-Journal of the Royal Statistical Society Series C* 1986, *35*, 195-206.

[62] M. J. Saiz-Abajo, B. H. Mevik, V. H. Segtnan and T. Naes, *Analytica Chimica Acta* 2005, *533*, 147-159.

[63] H. Martens, J. P. Nielsen and S. B. Engelsen, *Analytical Chemistry* 2003, *75*, 394-404.

[64] F. Wulfert, W. T. Kok and A. K. Smilde, *Analytical Chemistry* 1998, *70*, 1761-1767.

[65] D. C. Montgomery, *Design and Analysis of Experiments*, Wiley, New York, 1997, p. 684.

[66] Box , Wilson, *Royal Statistics Soeciety* 1951, *13*, 1-45.

[67] G. Vicente, M. Martinez and J. Aracil, *Bioresource Technology* 2007, *98*, 1724-1733.

[68] R. Bahloul, A. Mkaddem, P. Dal Santo and A. Potiron, *International Journal of Mechanical Sciences* 2006, *48*, 991-1003.

[69] G. Began, M. Goto, A. Kodama and T. Hirose, *Food Research International* 2000, *33*, 341-345.

[70] G. Vicente, M. Martinez and J. Aracil, *Bioresource Technology* 2007, *98*, 1754-1761.

[71] S. Dreer, R. Krismer and P. Wilhartitz, *Surface & Coatings Technology* 1999, *114*, 29-38.

[72] P. M. Andersson, T. Lundstedt and L. Abramo, *Journal of Chemometrics* 1996, *10*, 379-384.

[73] Y. L. Loukas, *Analytica Chimica Acta* 1998, *361*, 241-251.

[74] M. Berger and W. K. Wong, *Applied Optimal Designs*, Wiley, 2005, p. 285.

[75] P. F. deAguiar, B. Bourguignon, M. S. Khots, D. L. Massart and R. PhanThanLuu, *Chemometrics and Intelligent Laboratory Systems* 1995, *30*, 199-210.

[76] G. R. Flaten and A. D. Walmsley, *Chemometrics and Intelligent Laboratory Systems* 2004, *73*, 55-66.

[77] G. R. Flaten and A. D. Walmsley, *Analyst* 2003, *128*, 935-943.

9 Acknowledgements

This work was funded by the University of Hull with the assistance of Borealis Polymers. For this, I would like to thank Dr. Richard Garner and Peter Mayo.

I would like to thank my supervisor, Dr. A. D. Walmsley, for his patience, assistance, and guidance throughout my research and during the writing process.

I would also like to thank my fellow colleagues past and present for their time, advice and discussions. I would especially like to thank Dr. Samantha Barry and Dr. Ruth Wellock.

My thanks also go out to Kim Quigley, Lou Evans, and Sal Peyman – without their kind words, support, and motivation this thesis would probably never have been completed.

My final and most important thanks go to my parents, Peter and Melanie. If not for their unwavering support throughout my university career none of this would have been accomplished. I owe you a great debt.

I could not have completed this work without you all. Thank you.

10 Appendix

10.1 MatLab Programmes

10.1.1 Sample Selection Routines

10.1.1.1 Euclidean Distance Routine

```
    function [Value,dist,s,I] =
td_adapt2(Calibration,Validation,TConc)
    % Adaptive sampling using the euclidean distance in the PCA
scores space

    % I/O: [Value,dist,s] = td_adapt2(Calibration,Validation,TConc)



    T = Calibration;
    [T_x,mc,stds] = auto(T);
    %[U,S,V] = svd(T_x)
    options = [];
    % creates the options inputs for the PCA programme.
      options.name           = 'options';
      options.display        = 'off';
      options.plots          = 'none';
      options.outputversion = 3;
      options.preprocessing = {[] []};
      options.algorithm      = 'sim';
      options.blockdetails   = 'standard';
    model = pca(T_x,3,options);
    U = model.loads{1};
    V = model.loads{2};
    U_d = U;
    V_d = V;
    [m,n] = size(Validation);
    Value = [];
    dist = [];
     for i=1:m
        v = Validation(i,:);
        v_x = scale(v,mc,stds);
        nx = v_x*V_d;
        x_o = nx(:,1);
        y_o = nx(:,2);
        z_o = nx(:,3);
        dx = U_d(:,1)-x_o;
        dy = U_d(:,2)-y_o;
        dz = U_d(:,3)-z_o;
        dx2 = dx.^2;
        dy2 = dy.^2;
        dz2 = dz.^2;
        D2 = [dx2+dy2+dz2];
        D = sqrt(D2);
```

```
        [y,I] = sort(D);
        dist = [dist y];
        s = I(1:18,:);
        X = T(s,:);
        Y = TConc(s,:);
        [aX,mX,stX] = auto(X);
        [aY,mY,stY] = auto(Y);
        [a_v] = scale(v,mX,stX);
        options = [];
        options.name          = 'options';
        options.display       = 'off';     %Displays output to the
command window
        options.plots         = 'none';  %Governs plots to make
        options.outputversion = 3;         %2,3 Tells what to output
(3=ModelStruct)
        options.preprocessing = {[] []};  %See preprocess
        options.algorithm     = 'sim';    %SIMPLS algorithm
        options.blockdetails  = 'standard';  %level of details
        model = pls(aX,aY,5,options);
        p_model = pls(a_v,model,options);
        pred = p_model.pred{2};
        pred = [(pred*stY)+mY];
        Value = [Value;pred];
    end
```

10.1.1.2 Shenk and Westerhaus Routine

```
    function [sel,ssel] = td1(tspec,vspec);
    tic
    [m,n] = size(tspec);
    cof = [];
    for i = 1:m
        a = tspec(i,:);
        b = vspec;
        t = corrcoef([a;b]');
        s = t(2,1);
        cof  = [cof;s];
    end
    [D,I] = sort(cof);
    [p,q] = find(cof>0.99);
    sel = I(1:13);
    ssel = tspec(sel,:);
    toc


    function [v,model,pmodel,RMSEC] =
td_adapt_shenk(tspec,vspec,tconc)

    [sel] = td1(tspec,vspec);
    ts = tspec(sel,:);
    tc = tconc(sel,:);
    [atx,mx,stds] = auto(ts);
    [atc,mc,stdc] = auto(tc);
    [ay] = scale(vspec,mx,stds);
    comp = 3;
    options = [];
      options.name          = 'options'
```

```
    options.display       = 'off';     %Displays output to the
command window
    options.plots         = 'none';  %Governs plots to make
    options.outputversion = 3;         %2,3 Tells what to output
(3=ModelStruct)
    options.preprocessing = {[] []};  %See preprocess
    options.algorithm     = 'sim';    %SIMPLS algorithm
    options.blockdetails  = 'standard';  %level of details
  model = pls(atx,atc,comp,options);
  pmodel = pls(ay,model,options);
  RMSEC= model.detail.rmsec(1,comp);
  % [mcx,mx] = mncn(tspec);
  % [mcc,mc] = mncn(tconc);
  % [ay] = scale(a,mx);
  % model = pls(mcx,mcc,5);
  % pmodel = pls(ay,model);
  p = pmodel.pred{2};
  v = [(p*stdc)+mc];
  %v = [p+mc];
```

10.1.1.3 Condition Number Routine

```
  function [pcs] =td_f(X)
  load FCrit;
  S = svd(X);
  tot_eig = sum(S);
  for i = 2:19;
      a = [sum(S(1:i,:))/tot_eig]*100;
      n = i+1;
      if n>19
          disp('Final PC');
          break
      else
      end
      b = [sum(S(1:n,:))/tot_eig]*100;
      F = [b.^2]/[a.^2]
      CValue = FCrit([i-1],i)
      if F<CValue
          pcs = i;
          break
      else
      end
  end
  pcs;


  function [sel,cnumfor,cnumback,index] = td_cond2(matrix);

  %try out1
  [p,q] = size(matrix);
  [X,index] = shuffle(matrix,[1:p]');
  [mcx,mc] = mncn(X);
  [U,S,V] = svds(mcx,15);
  s = diag(S);
  [o,w] = size(s);
  [pcs] = td_f(mcx);
```

```
final_pc = pcs
a = 1;
cnumfor = [];
[m,n] = size(X);
while m>7
    a;
    X;
    m_x = mncn(X);
    [m,n] = size(m_x);
    m
    s1 = svds(m_x,final_pc);
    c1 = s1(1,:)/s1(final_pc,:);
    e = delsamps(X,a);
    index_e = delsamps(index,a);
    m_e = mncn(e);
    s2 = svds(m_e,final_pc);
    c2 = s2(1,:)/s2(final_pc,:);
    if c2<c1
        X = e;
        index = index_e;
        disp('Sample Removed!')
        a = a;
        cnumfor = [cnumfor;c2];
    else
        X = X;
        a = a+1;
        if a >= m
            break
        else
        end
    end
end
X;
index;
[f,g] = size(X);
cnumback = [];
z = f;
while z>0
    z
    S3 = svd(X);
    S3a = S3(1:final_pc,:);
    c3  = [(S3a(1,:))/(S3a(final_pc,:))];
    j = delsamps(X,z);
    index_j = delsamps(index,z);
    S4 = svd(j);
    S4a = S4(1:final_pc,:);
    c4 = [(S4a(1,:))/(S4a(final_pc,:))];
    if c4<c3
        X = j;
        z = z;
        index = index_j;
        cnumback = [cnumback;c4];
        disp('Sample Removed!')
    else
        X = X;
        z = z-1;
    end
end
sel = X;
```

10.1.2 Fourier Transform Routine

```
function [D] = td_fftk(kdt,N,n)
d = [];
f = [];
for k=-N:N
    a = cos((2*pi*k*n)/((2*N)+1));
    d = [d;a];
    b = (sin((2*pi*k*n)/((2*N)+1)))*sqrt(-1);
    f = [f;b];
end
f = f.*kdt;
d = d.*kdt;
h = [d+f];
j = sum(h);
D = (1/((2*N)+1)*j);


function [X,D,v] = td_fftn(time_interval,measurment_time,input);
% Calculates the discrete fourier transform of a signal. This is
not a FAST
% FOURIER TRANSFORM so care must be taken runing large sample
sets.
%
% I/O [X,v] = td_fftn(time_interval,measurment_time,input)
%
% Tom Dearing University of Hull v.2 15/2/2006

tic
kdt = input;
[p,r] = size(kdt);
N = (p-1)/2;
tm = measurment_time;
dt = time_interval;
vmin = 1/tm;
vmax = 1/(2*dt);
nmax = vmax/vmin;
v = 0:vmin:vmax;
D = [];
kdt = mncn(kdt);
tom = waitbar(0,'Please Wait....');
for n = 0:nmax;
    tn = n/nmax;
    waitbar(tn)
    q = td_fftk(kdt,N,n);
    D = [D;q];
end
close(tom)
X = D;
WIDTH = 0.025;
figure,bar(v,abs(X),WIDTH);
xlabel('Frequency /Hz');
ylabel('A(n)');
toc
```

10.1.3 Graphic User Interfaces

10.1.3.1 First Iteration – Demo4

```
function
% DEMO4 M-file for Demo4.fig
%      DEMO4, by itself, creates a new DEMO4 or raises the
existing
%      singleton*.
%
%      H = DEMO4 returns the handle to a new DEMO4 or the handle
to
%      the existing singleton*.
%
%      DEMO4('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in DEMO4.M with the given input
arguments.
%
%      DEMO4('Property','Value',...) creates a new DEMO4 or
raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before Demo4_OpeningFunction gets
called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to Demo4_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Demo4

% Last Modified by GUIDE v2.5 21-Nov-2006 15:46:33

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct( 'gui_Name',       mfilename, ...
                    'gui_Singleton',  gui_Singleton, ...
                    'gui_OpeningFcn', @Demo4_OpeningFcn, ...
                    'gui_OutputFcn',  @Demo4_OutputFcn, ...
                    'gui_LayoutFcn',  [] , ...
                    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
    % End initialization code - DO NOT EDIT


    % --- Executes just before Demo4 is made visible.
    function Demo4_OpeningFcn(hObject, eventdata, handles, varargin)
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin   command line arguments to Demo4 (see VARARGIN)

    % Choose default command line output for Demo4
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);

    % UIWAIT makes Demo4 wait for user response (see UIRESUME)
    % uiwait(handles.figure1);


    % --- Outputs from this function are returned to the command
line.
    function varargout = Demo4_OutputFcn(hObject, eventdata,
handles)
    % varargout  cell array for returning output args (see
VARARGOUT);
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Get default command line output from handles structure
    varargout{1} = handles.output;


    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
    uspec = uiimport;
    uspec = uspec.data;
    [m,n] = size(uspec);
    if m>1
        uspec = uspec(:,2)';
    else
    uspec = uspec;
    end
    load demodata;
    save demodata1 cspec vspec2 uspec cconc vconc2
    axes(handles.axes1);
    cla;
    plot(uspec'),axis     ,grid,title('Newly Collected
Spectra'),xlabel('Time /s'),ylabel('Signal Amplitude');
    axes(handles.axes2);
    cla;
    plot(cspec'),axis auto,grid,title('Calibration
Spectra'),xlabel('Time /s'),ylabel('Signal Amplitude');
    load output
    spectra.FID = [spectra.FID;uspec];
```

```
    o = clock;
    spectra.name =
[spectra.name;sprintf('Fname.%02.0f/%02.0f/%02.0f.%02.0f:%02.0f:%02.
0f',o(3),o(2),o(1),o(4),o(5),o(6))];
    save output spectra
    % hObject    handle to pushbutton1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)



    % --- Executes on button press in pushbutton2.
    function pushbutton2_Callback(hObject, eventdata, handles)
    load demodata1;
    [AX,mc,stds] = auto(cspec);
    [BX] = scale(uspec,mc,stds);
    options = [];
    % creates the options inputs for the PCA programme.
      options.name          = 'options';
      options.display       = 'off';
      options.plots         = 'none';
      options.outputversion = 3;
      options.preprocessing = {[] []};
      options.algorithm     = 'sim';
      options.blockdetails  = 'standard';
    model = pca(AX,3,options);
    U = model.loads{1};
    V = model.loads{2};
    v = BX(1,:);
    Uv = v*V;
    axes(handles.axes3);
    plot(U(:,1),U(:,2),'.');
    plot(Uv(1,1),Uv(1,2),'r.')
    xlabel('Scores on PC1'),ylabel('Scores on PC2'),title('PCA
Scores Plot of Calibration and Unknown Samples');
    axis auto,legend('Calibration Scores','Unknown
Scores','Location','Best');
    % hObject    handle to pushbutton2 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)



    % --- Executes on button press in pushbutton3.
    function pushbutton3_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton3 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    load demodata1
    [Value,dist,s] = td_adapt2(cspec,uspec,cconc);
    axes(handles.axes4)
    polar(dist),axis auto;
    title('Distances of Samples from Calibration Data');
    axes(handles.axes5)
```

```
    plot(dist),title('Distances from Calibration
Data'),xlabel('Sample Number'),ylabel('Euclidean Distance');
    grid,axis auto


    % --------------------------------------------------------------------
------
    function Untitled_1_Callback(hObject, eventdata, handles)
    % hObject    handle to Untitled_1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)




    % --- Executes on button press in pushbutton4.
    function pushbutton4_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton4 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    axes(handles.axes1)
    cla
    axes(handles.axes2)
    cla
    axes(handles.axes3)
    cla
    axes(handles.axes4)
    cla
    axes(handles.axes5)
    cla
    set(handles.text4,'String','=.........');
    set(handles.text6,'String','=.........');
    set(handles.text10,'String','=.........');
    set(handles.text10,'ForegroundColor','k')



    % --- Executes on button press in pushbutton5.
    function pushbutton5_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton5 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    load demodata1
    [Value,dist,s] = td_adapt2(cspec,uspec,cconc);
    load output
    spectra.predictions = [spectra.predictions;Value];
    set(handles.text4,'String',Value);
    [conf,conf_p,conf_d] = p_conf2(cspec,uspec)
    %[conf,cop,conf_p,conf_d] = pconf(cspec,uspec);
    c = conf_d(1,1);
    spectra.confidence = [spectra.confidence;conf_d(1,1)];
    save output spectra
    set(handles.text6,'String',c);
    if c>0
        load output
        set(handles.text10,'String','Green');
```

```
        set(handles.text10,'ForegroundColor','g')
        spectra.alert_status = [spectra.alert_status;'g'];
        save output spectra
    elseif c<0&c>-1
        load output
        set(handles.text10,'String','Yellow');
        set(handles.text10,'ForegroundColor','y');
        spectra.alert_status = [spectra.alert_status;'y'];
        save output spectra
    else
        load output
        set(handles.text10,'String','Red   ');
        set(handles.text10,'ForegroundColor','r');
        spectra.alert_status = [spectra.alert_status;'r'];
        save output spectra
    end



    %set(handles.text9,'String',time);


    % --- Executes on button press in pushbutton6.
    function pushbutton6_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton6 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)




    % --- Executes on button press in pushbutton8.
    function pushbutton8_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton8 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
   close(Demo4)
```

10.1.3.2 Second Iteration Demo5

```
    function varargout = Demo5(varargin)
    % DEMO5 M-file for Demo5.fig
    %      DEMO5, by itself, creates a new DEMO5 or raises the
existing
    %      singleton*.
    %
    %      H = DEMO5 returns the handle to a new DEMO5 or the handle
to
    %      the existing singleton*.
    %
    %      DEMO5('CALLBACK',hObject,eventData,handles,...) calls the
local
    %      function named CALLBACK in DEMO5.M with the given input
arguments.
```

```
    %
    %      DEMO5('Property','Value',...) creates a new DEMO5 or
raises the
    %      existing singleton*.  Starting from the left, property
value pairs are
    %      applied to the GUI before Demo5_OpeningFunction gets
called.  An
    %      unrecognized property name or invalid value makes
property application
    %      stop.  All inputs are passed to Demo5_OpeningFcn via
varargin.
    %
    %      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
    %      instance to run (singleton)".
    %
    % See also: GUIDE, GUIDATA, GUIHANDLES

    % Edit the above text to modify the response to help Demo5

    % Last Modified by GUIDE v2.5 29-Nov-2006 16:34:50

    % Begin initialization code - DO NOT EDIT
    gui_Singleton = 1;
    gui_State = struct('gui_Name',       mfilename, ...
                       'gui_Singleton',  gui_Singleton, ...
                       'gui_OpeningFcn', @Demo5_OpeningFcn, ...
                       'gui_OutputFcn',  @Demo5_OutputFcn, ...
                       'gui_LayoutFcn',  [] , ...
                       'gui_Callback'    []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT


    % --- Executes just before Demo5 is made visible.
    function Demo5_OpeningFcn(hObject, eventdata, handles, varargin)
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin   command line arguments to Demo5 (see VARARGIN)

    % Choose default command line output for Demo5
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);

    % UIWAIT makes Demo5 wait for user response (see UIRESUME)
```

```
    % uiwait(handles.figure1);


    % --- Outputs from this function are returned to the command
line.
    function varargout = Demo5_OutputFcn(hObject, eventdata,
handles)
    % varargout  cell array for returning output args (see
VARARGOUT);
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Get default command line output from handles structure
    varargout{1} = handles.output;


    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
      'C:\Program Files\MATLAB71\work\Model'
    uspec = uiimport;
    uspec = uspec.data;
    [m,n] = size(uspec);
    if m>1
        uspec = uspec(:,2)';
    else
    uspec = uspec;
    end
    load demodata;
    save demodata1 cspec vspec2 uspec cconc vconc2;
    axes(handles.axes1);
    plot(uspec'),axis auto,grid,title('Newly Collected
Spectra'),xlabel('Time /s'),ylabel('Signal Amplitude');
    axes(handles.axes2);
    plot(cspec'),axis auto,grid,title('Calibration
Spectra'),xlabel('Time /s'),ylabel('Signal Amplitude');
    [AX,mc,stds] = auto(cspec);
    [BX] = scale(uspec,mc,stds);
    load options
    model = pca(AX,3,options);
    U = model.loads{1};
    V = model.loads{2};
    v = BX(1,:);
    Uv = v*V;
    axes(handles.axes4);
    plot(U(:,1),U(:,2),'.');
    plot(Uv(1,1),Uv(1,2),'r.');
    xlabel('Scores on PC1'),ylabel('Scores on PC2'),title('PCA
Scores Plot of Calibration and Unknown Samples');
    axis auto,legend('Calibration Scores','Unknown
Scores','Location','Best');
    [Value,dist,s] = td_adapt2(cspec,uspec,cconc);
    axes(handles.axes3)
```

```
    plot(dist),title('Distances from Calibration
Data'),xlabel('Sample Number'),ylabel('Euclidean
Distance'),grid,axis auto;
    [s4,su4] = td_confs(cspec,uspec);
    [preds] = td_pls(cspec,cconc,uspec);
    [c_conf,glob_conf,confs_o,confsg_o] =
td_pconf4(Value,preds,cspec,cconc,uspec);
    set(handles.text7,'String',sprintf('%-6.2f',s4));
    set(handles.text8,'String',sprintf('%-6.2f',su4));
    set(handles.text9,'String',sprintf('%-6.2f',Value));
    set(handles.text10,'String',sprintf('%-6.2f',preds));
    set(handles.text11,'String',sprintf('%-6.2f',c_conf));
    set(handles.text12,'String',sprintf('%-6.2f',glob_conf));
    set(handles.text20,'String',sprintf('%-6.2f',confs_o));
    set(handles.text21,'String',sprintf('%-6.2f',confsg_o));
    % --- Executes on button press in pushbutton2.
    function pushbutton2_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton2 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    axes(handles.axes1)
    cla
    axes(handles.axes2)
    cla
    axes(handles.axes3)
    cla
    axes(handles.axes4)
    cla
    set(handles.text7,'String','.......');
    set(handles.text8,'String','.......');
    set(handles.text9,'String','.......');
    set(handles.text10,'String','.......');
    set(handles.text11,'String','.......');
    set(handles.text12,'String','.......');
    set(handles.text20,'String','.......');
    set(handles.text21,'String','.......');
```

10.1.3.3 Installed GUI

```
    function varargout = Demo6(varargin)
    % DEMO6 M-file for Demo6.fig
    %      DEMO6, by itself, creates a new DEMO6 or raises the
existing
    %      singleton*.
    %
    %      H = DEMO6 returns the handle to a new DEMO6 or the handle
to
    %      the existing singleton*.
    %
    %      DEMO6('CALLBACK',hObject,eventData,handles,...) calls the
local
    %      function named CALLBACK in DEMO6.M with the given input
arguments.
    %
```

```
    %      DEMO6('Property','Value',...) creates a new DEMO6 or
raises the
    %      existing singleton*.  Starting from the left, property
value pairs are
    %      applied to the GUI before Demo6_OpeningFunction gets
called.  An
    %      unrecognized property name or invalid value makes
property application
    %      stop.  All inputs are passed to Demo6_OpeningFcn via
varargin.
    %
    %      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
    %      instance to run (singleton)".
    %
    % See also: GUIDE, GUIDATA, GUIHANDLES

    % Edit the above text to modify the response to help Demo6

    % Last Modified by GUIDE v2.5 11-Dec-2006 15:11:09

    % Begin initialization code - DO NOT EDIT
    gui_Singleton = 1;
    gui_State = struct('gui_Name',       mfilename, ...
                       'gui_Singleton',  gui_Singleton, ...
                       'gui_OpeningFcn', @Demo6_OpeningFcn, ...
                       'gui_OutputFcn',  @Demo6_OutputFcn, ...
                       'gui_LayoutFcn',  [] , ...
                       'gui_Callback',   []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT


    % --- Executes just before Demo6 is made visible.
    function Demo6_OpeningFcn(hObject, eventdata, handles, varargin)
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin   command line arguments to Demo6 (see VARARGIN)

    % Choose default command line output for Demo6
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);

    % UIWAIT makes Demo6 wait for user response (see UIRESUME)
    % uiwait(handles.figure1);
```

```
    % --- Outputs from this function are returned to the command
line.
    function varargout = Demo6_OutputFcn(hObject, eventdata,
handles)
    % varargout  cell array for returning output args (see
VARARGOUT);
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Get default command line output from handles structure



    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    %start button!
    cd('c:\Program Files\MATLAB71\work\Model');
    set(handles.pushbutton1,'enable','off');
    set(handles.pushbutton4,'enable','off');
    set(handles.pushbutton6,'enable','off');
    load loop_control
    while loop_control>0
        set(handles.text5,'String','Waiting to Import Spectra');
        set(handles.text5,'ForegroundColor','b');
        pause(60)
        %timing loop, on the plant machine this is set for 11.5
minutes
        set(handles.text5,'String','Importing');
        % set to import raw FID file
        cd('c:\Program Files\Aztec\Rawdata_Res');
        n = importdata('rawdata.raw');
        new_spec = n.data(:,2)';
        cd('c:\Program Files\MATLAB71\work\Model');
        load last_ispec
        M = (sum((last_ispec - new_spec).^2));
        load loop_control
        if M>0
            last_ispec = new_spec;
            save last_ispec last_ispec;
            last_spec = last_spec +1;
            save loop_control loop_control last_spec;
            load output_m
            d = now;
            dt_import = [dt_import;d];
            load out_spec
            spec_num = [spec_num;last_spec]; %variable for output
            %save loop_control loop_control last_size last_spec;
            uspec = new_spec(:,1:2000);
            clear new_spec;
            load cdata
            %save imported spectra for later updating and maintenane
            ispec = [ispec;uspec];
```

```
            save out_spec ispec
            %status to plotting FIDs
            axes(handles.axes1);
            cla
            axes(handles.axes2);
            cla
            axes(handles.axes3);
            cla
            set(handles.text5,'String','Plotting FIDs');
            axes(handles.axes1);
            plot(uspec),title('Imported FID'),grid,xlabel('Time
/s'),ylabel('Signal Amplitude');
            axes(handles.axes2);
            plot(cspec'),title('Calibration FIDs'),grid,xlabel('Time
/s'),ylabel('Signal Amplitude');
            [s4,su4] = td_confs(cspec,uspec);
            if (su4/s4)<1
                set(handles.text8,'String','Green');
                set(handles.text8,'ForegroundColor','g');
                a = 1;
                spec_alert = [spec_alert;a];
            else
               set(handles.text8,'String','Red');
               set(handles.text8,'ForegroundColor','r');
               a = -1;
               spec_alert = [spec_alert;a];
            end
            %calculating scores
            set(handles.text5,'String','Calculating Scores Space');
            [AX,mc,stds] = auto(cspec);
            [BX] = scale(uspec,mc,stds);
            model = pca(AX,3,options);
            U = model.loads{1};
            V = model.loads{2};
            v = BX(1,:);
            Uv = v*V;
            %plotting scores
            set(handles.text5,'String','Plotting Scores');
            axes(handles.axes3);
            plot(U(:,1),U(:,2),'.');
            plot(Uv(1,1),Uv(1,2),'r.')
            xlabel('Scores on PC1'),ylabel('Scores on
PC2'),title('PCA Scores Plot of Calibration and Unknown
Samples'),axis auto,legend('Calibration Scores','Unknown
Scores','Location','Best');
            %calcuating adaptive samples
            set(handles.text5,'String','Building Adaptive Model');
            [Value] = td_adapt2(cspec,uspec,cconc); %variable for
output
            [m_preds] = [m_preds;Value];
            [preds] = td_pls(cspec,cconc,uspec);
            set(handles.text1,'String',sprintf('%-6.2f',Value));
            set(handles.text2,'String',sprintf('%-6.2f',preds));
            % calculating model confidences
            set(handles.text5,'String','Calculating Confidences');
            [c_conf,glob_conf,confs_o] =
td_pconf4(Value,preds,cspec,cconc,uspec);
            m_confs = [m_confs;c_conf];
            set(handles.text3,'String',sprintf('%-6.3f',c_conf));
%variable for output
            set(handles.text4,'String',sprintf('%-6.3f',glob_conf));
```

```
            if (c_conf/confs_o)<1
                set(handles.text9,'String','Green');
                set(handles.text9,'ForegroundColor','g'); %variable
for output
                b = 1;
                conf_alert = [conf_alert;b];
            elseif (c_conf/confs_o)>1&&(c_conf/confs_o)<2
                set(handles.text9,'String','Yellow')
                set(handles.text9,'ForegroundColor','y');%variable
for output
                b = 0;
                conf_alert = [conf_alert;b];
            else
                set(handles.text9,'String','Red');
                set(handles.text9,'ForegroundColor','r');%variable
for output
                b = -1;
                conf alert = [conf alert;b];
            end
            % calculate ATSM stats %variable for output
            %compose output form of csv
            save output_m spec_num spec_alert conf_alert dt_import
m_confs m_preds
            save loop_control loop_control last_spec;
            v1 = (spec_num);
            v2 = (m_preds);
            v3 = (m_confs);
            T = [v1 dt_import v2 v3 spec_alert conf_alert];
            save edata.asc T -ascii -double
            %displaying saving data.

        else
            disp(['New Spectra Not Found - Skipped:
',datestr(now)]);
            set(handles.text5,'String','No New Spectra');
            set(handles.text5,'ForegroundColor','r');
            loop_control;
        end
        loop_control;
    end
    set(handles.text5,'String','Stopped');
    set(handles.pushbutton5,'enable','on');
    set(handles.pushbutton4,'enable','on');

    % --- Executes on button press in pushbutton2.
    function pushbutton2_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton2 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    set(handles.pushbutton2,'enable','off');
    set(handles.pushbutton3,'enable','on');
    load loop_control
    loop_control = -1;
    save loop_control loop_control last_spec;

    % --- Executes on button press in pushbutton3.
    function pushbutton3_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton3 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
```

```
    % handles    structure with handles and user data (see GUIDATA)
    set(handles.pushbutton1,'enable','on');
    set(handles.pushbutton2,'enable','on');
    set(handles.pushbutton3,'enable','off');
    set(handles.pushbutton5,'enable','off');
    set(handles.pushbutton6,'enable','off');

    load loop_control
    loop_control = 1;
    save loop_control loop_control last_spec;
    set(handles.text5,'String','System Reset');


    % --- Executes on button press in pushbutton4.
    function pushbutton4_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton4 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    close(Demo6);


    % --- Executes on button press in pushbutton5.
    function pushbutton5_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton5 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    cd('c:\Program Files\MATLAB71\work\Model');
    set(handles.pushbutton5,'enable','off');
    load cdata;
    load out_spec;
    load out_up;
    %load loop_control;
    load output_m;
    UPdata = csvread('update.csv');
    s_num = UPdata(:,1);
    a = [spec_num(1,1)-1];
    n_num = [s_num-a];
    ncspec = ispec(n_num,:);
    ncconc = UPdata(:,9);
    spec_addn = [spec_addn;n_num];
    conc_add = [conc_add;ncconc];
    new_spec = [new_spec;ncspec];
    save td_added spec_addn conc_add new_spec
    % at this point for future work we must look at an automatic
purge of
    % samples that may become surplus to requirement based upon
addition of
    % these samples
    cspec = [cspec;ncspec];
    cconc = [cconc;ncconc];
    save cdata cspec cconc options
    set(handles.text5,'String','Update Completed');
    set(handles.pushbutton6,'enable','on');
```

```
    % --- Executes on button press in pushbutton6.
    function pushbutton6_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton6 (see GCBO)
    % eventdata  reserved - to be defined in a future version of
MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % purge button
    set(handles.pushbutton6,'enable','off');
    load output_m
    spec_num = [];
    dt_import = [];
    m_confs = [];
    m_preds = [];
    spec_alert = [];
    conf_alert = [];
    save output_m spec_num spec_alert conf_alert dt_import m_confs
m_preds
    ispec = [];
    save out_spec ispec
```

10.1.3.4 Confidence Algorithm

```
    function [c_conf,glob_conf,confs_o,confsg_o] =
td_pconf4(pred_a,pred_g,cspec,cconc,uspec)


    [AX,mc,stds] = auto(cspec);
    [BX] = scale(uspec,mc,stds);
    options = [];
    % creates the options inputs for the PCA programme.
      options.name           = 'options';
      options.display        = 'off';
      options.plots          = 'none';
      options.outputversion = 3;
      options.preprocessing = {[] []};
      options.algorithm      = 'sim';
      options.blockdetails   = 'standard';
   model = pca(AX,3,options);
   U = model.loads{1};
   V = model.loads{2};
   v = BX(1,:);
   Uv = v*V;
   xo = Uv(:,1);
   yo = Uv(:,2);
   zo = Uv(:,3);
   dx = U(:,1) - xo;
   dy = U(:,2) - yo;
   dz = U(:,3) - zo;
   dx2 = dx.^2;
   dy2 = dy.^2;
   dz2 = dz.^2;
   ed2 = dx2+dy2+dz2;
   ed = sqrt(ed2);
   [sel,I] = sort(ed);
   Isel = I(1:18,:);
   Iconc = cconc(Isel,:);
```

```
Ipconc = [Iconc;pred_a];
stca = std(Ipconc);
[m,n] = size(Ipconc);
sqm = sqrt(m);
a = (stca/sqm);
c_conf = 1.96*a;
IGconc = [cconc;pred_g];
stcg = std(IGconc);
[p,q] = size(IGconc);
sqp = sqrt(p);
b = (stcg/sqp);
glob_conf = 1.96*b;
sd_conc = std(Iconc);
sqrm = sqrt(m-1);
c = [sd_conc/sqrm];
confs_o = 1.96*c;
stdcconc = std(cconc);
sqrp = sqrt(p-1);
d = (stdcconc/sqrp);
confsg_o = 1.96*d;
```