An Expert System for the Performance Control of Rotating Machinery

William N. Pearson

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

This research programme was carried out in collaboration with Performance Improvements Ltd.

Napier University

December 2000

PAGE

NUMBERING

AS ORIGINAL

Abstract

This research presented in this thesis examines the application of feed forward neural networks to the performance control of a gas transmission compressor. It is estimated that a global saving in compressor fuel gas of 1% could save the production of 6 million tonnes of CO_2 per year.

Current compressor control philosophy pivots around prevention of surge or anti-surge control. Prevention of damage to high capital cost equipment is a key control driver but other factors such as environmental emissions restrictions require most efficient use of fuel. This requires reliable and accurate *performance* control.

A steady state compressor model was developed. Actual compressor performance characteristics were used in the model and correlations were applied to determine the adiabatic head characteristics for changed process conditions.

The techniques of neural network function approximation and pattern recognition were investigated. The use of neural networks can avoid the potential difficulties in specifying regression model coefficients. Neural networks can be readily re-trained, once a database is populated, to reflect changing characteristics of a compressor.

Research into the use of neural networks to model compressor performance characteristics is described. A program of numerical testing was devised to assess the performance of neural networks. Testing was designed to evaluate training set size, signal noise, extrapolated data, random data and use of normalised compressor coefficient data on compressor speed estimates. Data sets were generated using the steady state compressor model. The results of the numerical testing are discussed.

Established control paradigms are reviewed and the use of neural networks in control systems were identified. These were generally to be found in the areas of adaptive or model predictive control. Algorithms required to implement a novel compressor performance control scheme are described. A review of plant control hierarchies has identified how the scheme might be implemented. The performance control algorithm evaluates current process load and suggests a new compressor speed or updates the neural network model.

Compressor speed can be predicted to approximately ± 2.5% using a neural network based model predictive performance controller. Comparisons with previous work suggest potential global savings of 34 million tonnes of CO_2 emissions per year. A generic, rotating machinery performance control expert system is proposed.

Declaration

I declare that this thesis and the work described within was undertaken by me, except where otherwise referenced in the text. The research was performed at Napier University, Edinburgh between October 1997 and October 2000.

W.N. Pearson

December 2000

Acknowledgements

My thanks are extended to Napier University for the opportunity to carry out this research and in particular to Dr. Alistair Armitage in his role as second supervisor. The remaining supervisors have been, over the course of the research: Prof. A. Almaini; Dr. W. Buchanan; Prof. T. Fogarty; Prof. J. Kennedy; Dr. Mike Mannion; Dr. S. Mowbray - I acknowledge their contributions. The supplemental financial assistance received from the Industrial Sponsor, during the three years of research, is, likewise, acknowledged.

The comment and constructive criticism of my external examiner, Dr. Kamal Botros, has been invaluable in shaping my thesis, thank you sir.

I would like to thank Dr. Dickon Jackson for his constructive, critical proof reading of my thesis draft, Dr. Andrew Hunter of Trajan software for his advice on aspects of neural networks, Dr. Les Short for his advice on mathematical machinations.

Dedication

I dedicate this thesis to my family: to my partner in life Linda, for her encouragement to strive for what I might yet be; to my three sons, William, Alex and Thomas, that they might be as proud of me, as I am of them.

Contents

1 Introduction

1.1 Natural Gas Distribution in the UK

In the UK, gas is produced from offshore oil and gas fields. The gas is pipelined ashore to reception terminals where it is treated to remove entrained liquids and corrosive elements. Once treated, the gas is dispatched to a distribution grid for supply to customers, both industrial and domestic. The machinery, pipelines and vessels used to transport the gas are referred to as the National Transmission System (NTS). This NTS is operated by Transco and is shown in Figure 1.1.

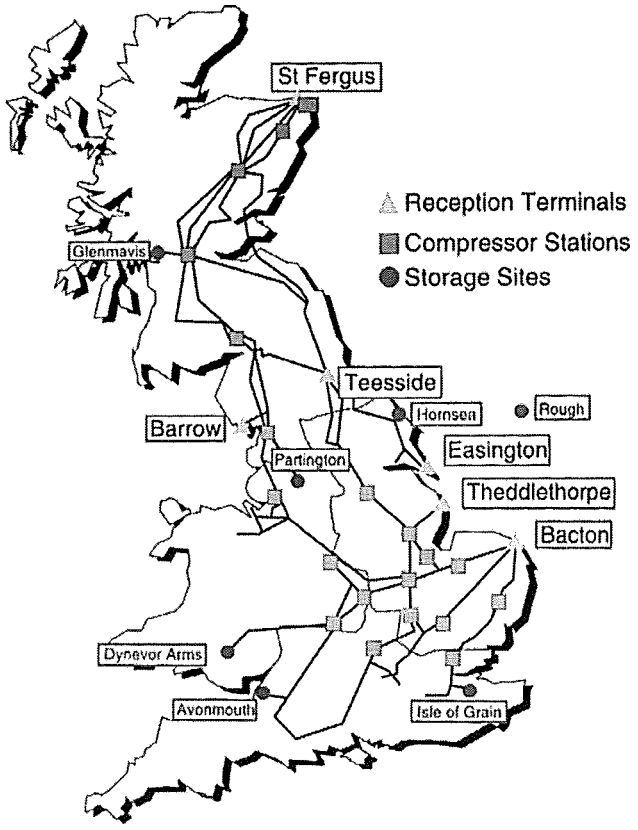

Figure 1.1: UK Gas Transmission System (courtesy of Transco)

1.2 UK Gas Transmission Statistics

Data in this section is taken from Transco [1]. The asset value of the transmission and storage network is estimated at around £11.6 billion. The NTS has approximately 270,000 km of pipe ranging from 63 mm to 1200 mm diameter. The network supplies 19.5 million industrial and domestic consumers making up 48% of the UK energy requirements. (18,000 kilometres of this pipe is used for gas transmission to 13 delivery zones). The main storage facility is the "Rough" gas field reservoir, with a capacity of around 2.8 billion cubic metres of gas at 200 bar pressure. This represents around 82% of the NTS total storage capacity. Gas is both produced from, and injected into the reservoir.

Two onshore subterranean salt cavities can hold 189 million cubic metres of gas, around 9% of the total storage capacity. Other storage facilities include more than 500 gas holders capable of storing 27.5 million cubic metres of gas, enough to supply 4 million homes for one day. There are also five Liquefied Natural Gas (LNG) sites, close to high population centres, each holding an equivalent of 25 million cubic metres of gas.

The operating pressure of the NTS at 75 bar, dropping to 25 mbar for domestic consumer supply. Gas moves around the NTS at approximately 10 m/s. Average daily demand runs at around 200 million cubic metres (based on 1996 estimates by Transco [1]), increasing by up to six fold in winter. Typical household annual gas consumption is equivalent to 46,000 kWh, a typical calorific value for gas being 39 megajoules per cubic metre (MJ/m^3).

1.3 NTS Operating Strategy

The daily production of gas from the offshore fields may be supplemented by drawing down on the stored supplies. Similarly, excess in supply can be put into storage. Supply must be closely matched to demand since only 2% of the total linepack, around 6 million cubic metres, can be drawn down to meet supply fluctuations. Hence the gas must be moved from storage facilities to

supply the NTS as demand requires. Demand varies with the weather and the time of day, and is both seasonal and demographic. The way in which demand is met is very much dependent on the availability of compression capacity to move gas to where it is required.

Compressors are critical components in the NTS to transport gas from its point of production, or storage, to the consumer. They provide the energy to move large quantities of gas around the network to meet instantaneous demand and to replenish local storage. (An LNG storage facility can be drawn down in six days but takes sixty days to refill).

The NTS has twenty compressor stations housing 53 gas turbine driven compressors. The gas turbines range in size from an RB211 delivering 25MW shaft power to an Orenda delivering 7MW shaft power. The equivalent of around 16% of the gas supplied to customers is used as compressor fuel gas, [1].

Operation of compressors in the UK gas transmission systems transport a contractual quantity (nomination) of energy or gas from a "Shipper " to its customer. The "Shippers" are the gas producers, their customers are gas consumers. The transmission system (and its compression stations) is run in accordance with the provisions set down in the Network Code [2]. At the very least a break in gas supply would be a serious inconvenience for consumers and, at worst, incur commercial penalties or loss of business.

All things being equal, a compressor would be set at some compressor operating point (COP) and operate at constant gas flow rate (nomination) and constant discharge pressure (transmission system pipeline pressure). Unfortunately, in live plant, all things are not equal and changes in suction temperature and pressure or gas molecular weight change the pressure/flow characteristics (performance curves) of the compressor, Gresh [3]. In addition draw down on the NTS linepack may cause variations in the required compressor discharge pressure. Typically, there will be more than one set of

operational conditions to allow gas processing compressor plant to operate with flexibility.

1.4 European Markets and Beyond

Eurogas represents 15 member countries supplying gas within Western Europe. Details of the Western Europe gas supply industry are taken from Eurogas [4]. In 1998 gas consumption in Western Europe rose by 4% to reach a new high of 365 billion cubic metres. 70% of this came from within Western Europe with the remaining 30% imported from Eastern Europe and North Africa. The Western European gas transmission network amounts to 167,000 kilometres of pipeline, delivering gas to 76 million customers.

In the continent of North America it is estimated that the gas transmission network pipeline length is 119 million kilometres, [5]. In the USA, the total gas consumption for 1999 was 23500 petajoules (PJ), equivalent to 603 billion cubic metres. Total gas consumption for Canada in 1999 is estimated at 2900 PJ, or around 74 billion cubic metres. The Transcanada gas transmission network has around 76.5 million kilometres of pipeline split between three services, [6]:

- British Columbia, which has 11 gas turbine driven compressors totalling 184 megawatts (MW) compression capacity.
- Canadian Mainline, which has 191 compressors totalling 2400 MW compression capacity. Of these, 134 units are driven by gas turbines.
- Alberta System, which has 112 compressors, totalling 940 MW compression capacity, of which 93 units are driven by gas turbine.

Annual global gas sales are estimated at around 79725 PJ, equivalent to 2000 billion cubic metres, [5]. Most of this gas is transported using compressors. Applying the Transco figure of 16% as typical of compression fuel gas consumption, assuming that 76% of compressors are gas turbine driven, implies that 240 billion cubic metres of gas is used as compressor fuel gas.

The number of installed gas transmission compressors across the world is not readily available. In terms of pipeline lengths, the largest transmission network is in Russia, the Transcanada network is next largest and the United States network is the third largest. Assuming compression capacity is proportional to network length, it is estimated that the Russian network will have around 600 compressors, and the United States network around 180 compressors. Using this method the total number of gas transmission compressors in use around the world is probably between 2500 and 3500 compressors.

1.5 Controlling Gas Delivery Compressors

Gas pipeline transportation compressor stations generally use centrifugal compressors operating in parallel, although some are arranged in series. Parallel configuration allows capacity throughput to be altered to meet demand, by varying the number of compressors on line. Centrifugal compressors exhibit small discharge pressure variations for large changes in compressor flow rate. This performance characteristic can accommodate swings in customer demand whilst maintaining relatively steady supply pressures. It is however, this characteristic which makes the control of parallel compressor problematic, as a slight difference in discharge pressure characteristic can result in compressor instability.

A de-facto compressor control standard has been widely adopted in industry as a means of controlling high capital cost, gas transmission compressors. The control method was described by White [7], as ameans of anti-surge control, and is generally referred to as the "flow deltap" method. Staroselsky [8] and the

Compressor Control Corporation developed the method to implement a dimensionless factor used to control compressor operating point (COP) in terms of proximity to surge point. Further refinement of this method brought about the inclusion of an adaptive control element to improve surge protection. Boyce [9] presents several variations on anti-surge control. A review of the more typical variations are also described in Chapter 2. A formal justification underpinning the use of the "flow-deltap" method, and its variations, are based on invariant co-ordinate systems described by Batson [10]. Hence an anti-surge control technique evolved to provide implicit performance control of a compressor.

The basic method of control sets a common COP in terms of compressor proximity to the surge limit. This is not necessarily the most fuel efficient COP and makes control of multiple compressor, using a common COP setpoint, difficult to achieve. A further complication is that the required COP for each compressor, in a compressor station, are inter-dependent. Optimisation of individual COPs could improve fuel efficiency, reduce emissions and free up compression capacity. Better use of compression capacity could result in fewer compressors being required during peak demand.

Botros [11] identified that compressor (surge) control technology was in a "perfection stage". Without new invention, incremental changes in technology such as different computational techniques, would bring refinement of surge control. Relatively little published material in new computational techniques, such as fuzzy logic methods, relate to pipeline applications such as pipeline compressor control, Botros [12].

Two such examples of published material are based on the "smart" control technologies of fuzzy logic. These have brought about incremental improvements in compressor control. Vachtsevanos [13] described fuzzy logic rules in a fault prediction system based on the Dempster-Shearer possibility

theory. This system could predict the propensity a compressor surge event using incomplete information. Cordiner [14] investigated the use of fuzzy logic for performance control of compressors. This work describes the use of Fuzzy Logic as a means of advising COP to improve load sharing between for individual compressors used in compression stations. It was shown that individual compressors could be more efficiently controlled to meet compressor station throughput targets.

The work suggested neural networks could be used to optimise operating point selected by the Fuzzy control system. The advantage seen in using neural networks is their ability to learn changing compressor characteristics through supervised training techniques. This was the starting point for the current research.

2 Compressors – Principles, Operation and Control

2.1 Introduction to Rotary Compressors

A large amount of literature is available which describes the action of compressors. Examples of sources of information in this section are: GPSA [15], Harman [16] and Gresh [3].

A compressor transfers energy, supplied through the work done by an external prime mover, into a compressible fluid stream. The energy transfer causes a flow of gas through the compressor from a lower to a higher pressure. Different compressor types exhibit different compression characteristics. Selection of a compressor type is dependent upon the application in which it is used. Gas transmission compressors are usually centrifugal type. The compressor is driven by a prime mover, which could be an electric motor or a gas turbine.

Associated with both the compressor and the prime mover are discrete control systems. The control system for the prime mover will regulate fuel supply to control its speed. Compressor control is based on anti-surge control, flow and/or pressure control and load sharing, where multiple compressor operate together. Load sharing and flow/pressure control are compressor performance control techniques.

2.1.1 Compressor Types

Compressors are usually of the rotating type (either axial or centrifugal), or are positive displacement types. Higher pressure ratios are achieved with the latter whereas rotating compressors generally have higher throughput capacity. In pipeline applications, rotating compressors superseded positive displacement compressors with the advent of the gas turbine as a prime mover, and its associated improvement in efficiency and costs.

Compressor characteristics can be represented as a plot of discharge pressure

against actual inlet volume flow rate. The compressor performance characteristic of discharge pressure vs. compressor inlet actual volume flow rate (AVF) for axial and centrifugal compressors, operating at constant speed, are represented in Figure 2.1.

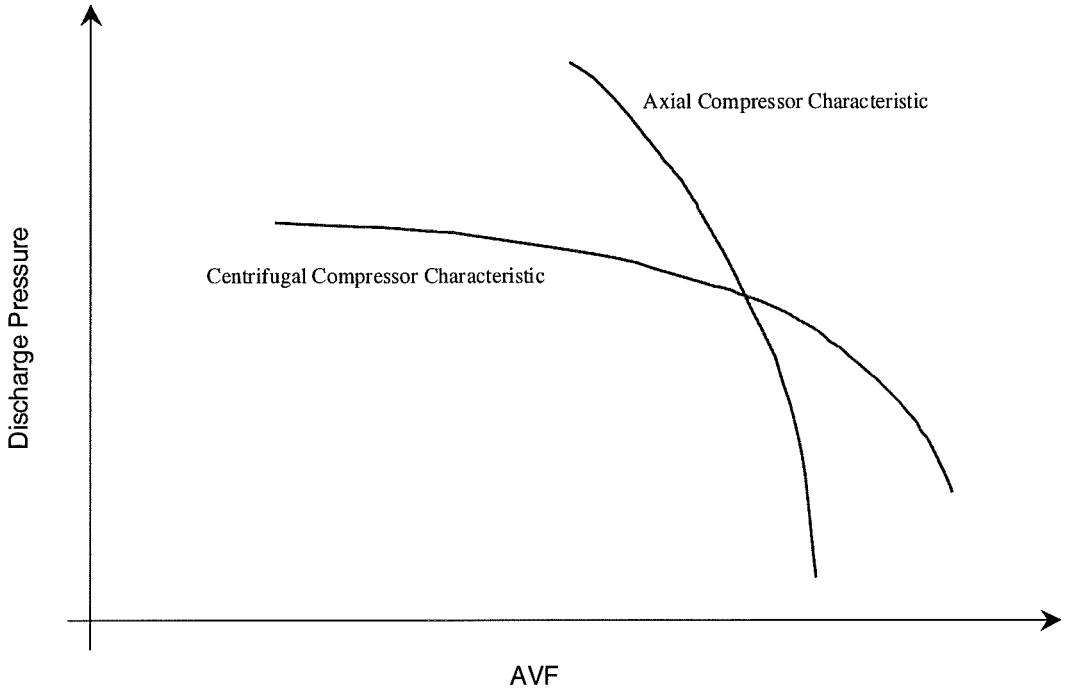

Figure 2.1 - Compressor Performance Characteristics

A comparison of the characteristics shows that the centrifugal compressor is less susceptible to changes in discharge pressure due to changes in AVF or compressor throughput. Conversely, small changes in discharge pressure can significantly affect the throughput of the centrifugal compressor. This type of characteristic is suited to gas transmission pipeline distribution networks where supply pressure transients or fluctuations are undesirable, or other near constant discharge pressure applications.

Axial compressor discharge pressure is markedly affected by relatively small changes in compressor throughput, but changes in discharge pressure

conditions do not significantly affect compressor throughput. This type of characteristic is suited to near constant volume throughput applications, where discharge pressure may fluctuate.

The type of compressor considered here is specifically the centrifugal type as these are generally used in transmission systems. Many of the describing equations and general characteristics are, however, typical of both axial and centrifugal compressors.

2.1.2 Types of Prime Mover

The type of prime mover used to drive the compressor depends on the process into which the compressor is integrated. In a process where high-grade 'waste' heat is a by-product, the compressor might be driven by a steam turbine. If the compressor is situated in a remote location, such as a Trans-continental pipeline transmission station, the prime mover could be a synchronous electric motor or a gas turbine engine. The compressor can be mechanically coupled to the engine, either directly or through a gearbox.

The prime mover can consume fuel, as in the case of a gas turbine, but can also be a turbine that extracts energy from the process itself. In this case a high-pressure gas stream expands through a power turbine coupled to a compressor. At a gas reception terminal incoming high-pressure gas may expand across a turbo-expander thus reducing the gas stream pressure whilst driving a process compressor. Alternatively, the high-pressure gas stream could be the exhaust gas from a gas turbine (gas generator). In this instance the exhaust stream expands across a turbine (gas engine) coupled to the compressor.

2.1.3 Multiple Compressor Configurations

Depending on the application, compressors can operate in series or parallel configurations. Series compressors can consist of a number of compression stages in the same casing or two different casings coupled in series. These can be

a number of compressor impellers located on the same shaft in a single compressor barrel. Pressure tapping or gas take-off points may be located on the casing between compression stages. Some applications may be met using discrete compressor units, with discrete prime movers, piped together in series.

Parallel compression is achieved through the use of a number of discrete compressors and prime movers, which have a common suction header pipe and a common discharge header pipe. Series compression generates high pressures whilst parallel compression has high throughput or transmission capacity.

Gas pipeline transmission compressor stations generally use centrifugal compressors operating in parallel. This configuration allows capacity throughput to be altered, to meet demand, by varying the number of compressors on line.

2.1.4 Compressor Performance Characteristics

Compressor performance can be represented variously as plots of discharge pressure or polytropic head against compressor inlet actual volumetric flow rate (AVF), for constant angular velocity. Supplemental performance data such as polytropic efficiency and pressure ratio can also be plotted against AVF. These groups of plots are known as compressor performance characteristics[1]. A typical Compressor Performance Characteristics, for a variable speed compressor is shown in Figure 2.2.

[1] Terms used to described compressor performance are explained in Section 2.2.

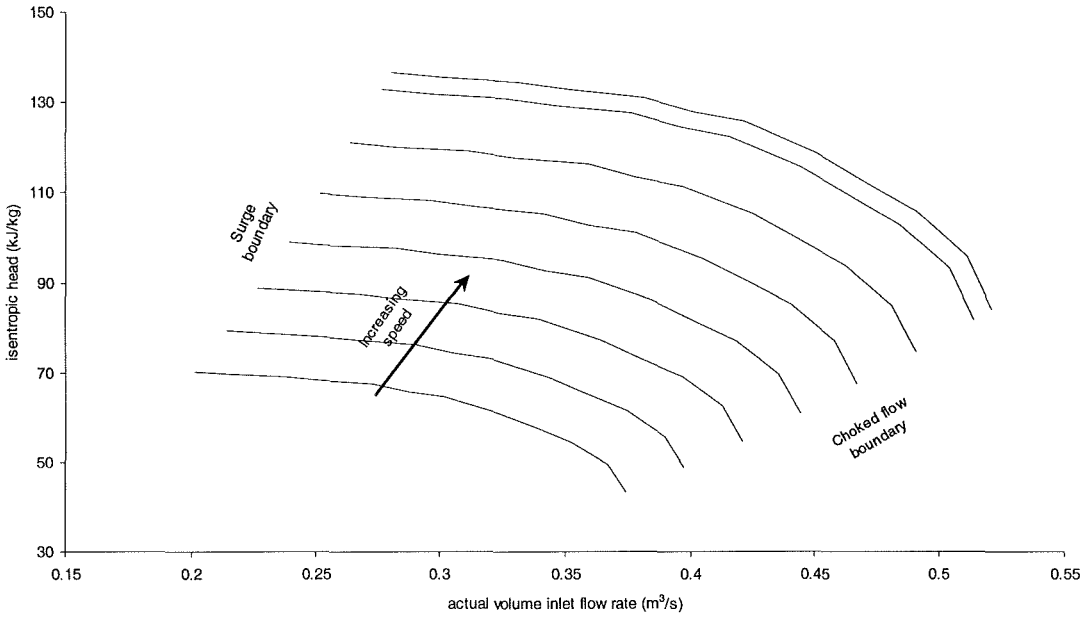

Figure 2.2 - Variable Speed Compressor Performance Characteristic

The characteristic shows Isentropic Head vs. AVF for lines of constant speed of rotation. The boundaries of operation are delineated by maximum and minimum speeds, Surge and Choked flow (Stonewall) boundaries.

2.2 Compressors - Principles of Compression

A cutaway diagram of a centrifugal compressor is shown in Figure 2.3.

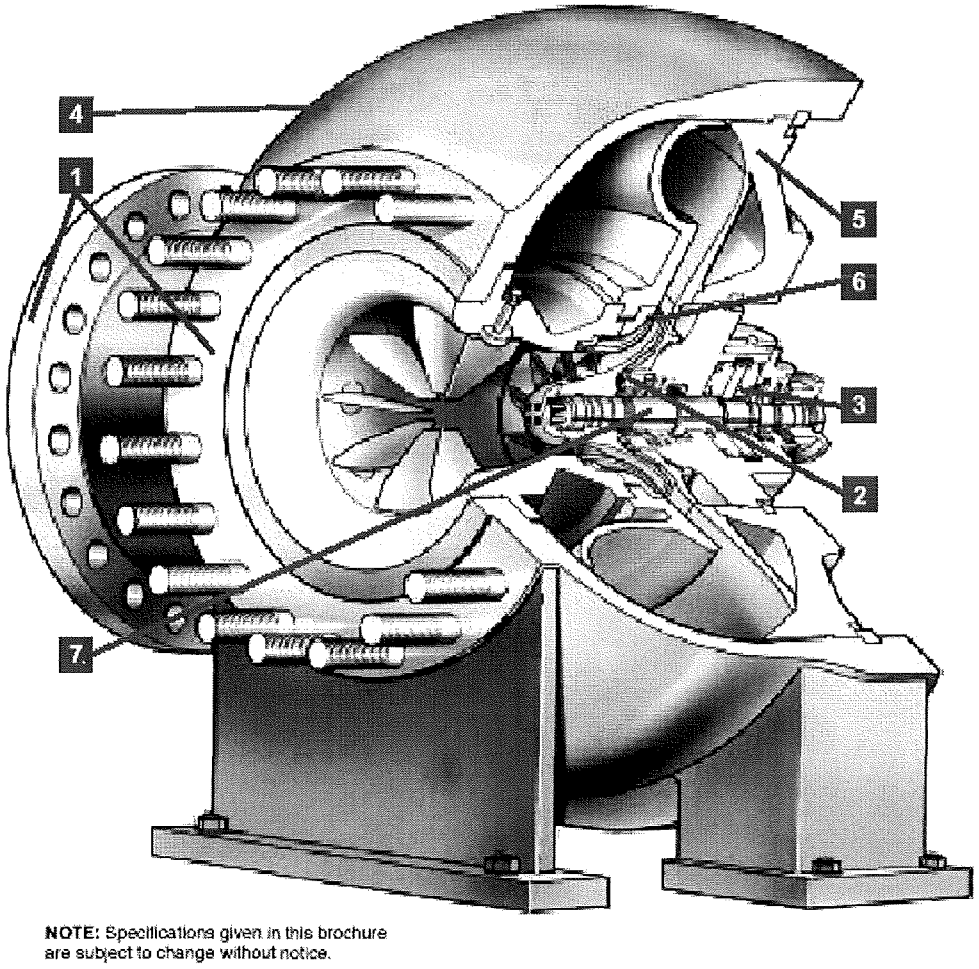

Figure 2.3 - Centrifugal Compressor Cutaway Diagram (courtesy of Cooper-Bessemer)

The following mechanical features are identified: 1. - Inlet/Discharge Flanges; 2 - Discharge Pressure fed sealing gas; 3 - High capacity tilt-pad thrust bearing; 4 – Casing, high strength alloy steel; 5 – Casing cover with segmented shear rings; 6 – Impellor/gas passages; 7 – rotor.

The prime mover causes the compressor rotor shaft and impeller to rotate. Gas is drawn in through the inlet flange and accelerated across the rotating impeller, outwards from the impeller eye, towards the casing and diffuser chamber. Kinetic energy is absorbed by the gas in the form of axial and swirl or tangential velocity components. The diffuser chamber is an involute, curved passage increasing in cross sectional area as it leads towards the outlet flange. As the gas progresses through the diffuser chamber the tangential velocity is dissipated and, consistent with Bernoulli's equation [17], the dissipated kinetic energy is converted to pressure energy. The pressure generated by a compressor is dependent on the degree of swirl velocity imparted into the gas and the path length travelled by the gas molecules in the diffuser. The path length is the distance travelled from the impellor exit to the diffuser casing. The longer the path length, the more kinetic energy is dissipated, the greater is the head loss. A diffuser may be fitted with vanes to shorten the path length of the gas and so reduce head losses, [9]

The degree of swirl velocity added is dependent on the angular acceleration induced by the rotating impeller. Angular acceleration of the gas is a function of the aerodynamic characteristics of the impeller. Guide (stator) vanes may be present on the compressor inlet. As the flowing stream impinges on the guide vanes a component of swirl velocity is imparted into the gas stream prior to acceleration across the impeller. Angle of the guide vanes adds flexibility in compressor design. Varying the angle can be used to control compressor performance.

2.2.1 Kinematic Influence Parameters

The theoretical head produced by a centrifugal compressor is determined by the velocity triangles at the inlet and exit of the impeller, shown in Fig. 2.4.

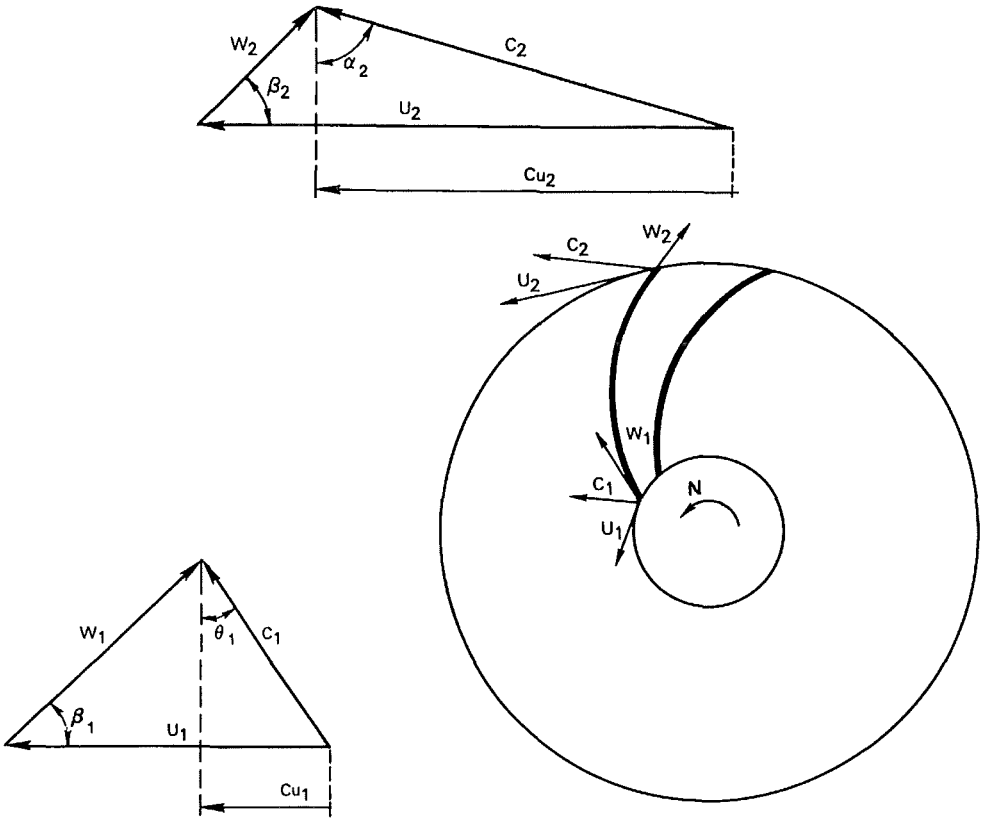

Fig. 2.4 Velocity triangles for impeller of centrifugal compressor [18]

Each triangle is composed of the absolute gas velocity, C, the rotational velocity, U corresponding to the impeller speed and the velocity, W of the gas relative to the blade passages. The rotational velocity, U is perpendicular to the impeller radius and the relative velocity, W is parallel to the blade passage; thus its direction depends on the blade angle. Total enthalpy change between the inlet and the exit of the impeller can be calculated from the "Euler" equation:

$$\Delta h_{12} = h_{o2} - h_{o1} = Cu_2.U_2 - Cu_1.U_1 \tag{2.1}$$

For most gas compressors, which are manufactured with backward swept blades, the enthalpy change decreases when the flow rate increases. This behaviour can be explained by Fig. 2.5, where the increase in flow shown by W_2^*, reduces Cu_2 to Cu_2^*. This reduces the head generated from equation (2.1).

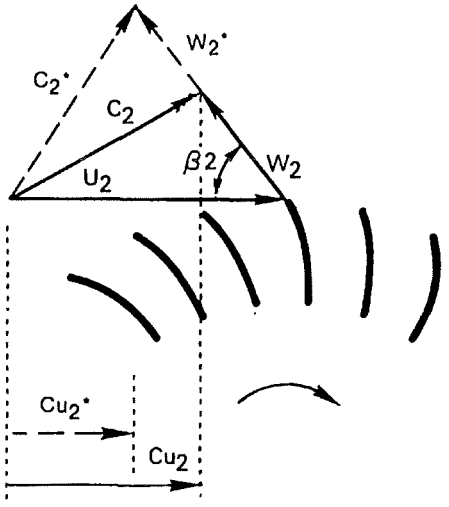

Fig. 2.5 Effect of increased flow rate for backward swept impeller [18]

2.2.2 *Internal losses*

The ideal head vs. flow characteristic for a centrifugal compressor is affected by irreversible losses that occur within the impeller blades passages and in the downstream diffuser (vaned or vaneless). The major source of these losses include:

- Frictional losses, which increase proportionally with the flow rate, squared.
- Blade loading losses where high adverse pressure gradients along the pressure and suction surfaces of the impeller blades produce flow separation.
- Incidence angle losses attributed to an incorrect angle of attack at the impeller inlet or into a vaned diffuser that creates additional boundary

layer separation losses.

- Secondary flow losses generated by the centrifugal and Coriolis forces within the rotating impeller.
- Wake mixing losses within the blade passages and behind the blades at the impeller exit.
- Separation and mixing losses within the diffuser because of incorrect expansion.
- Losses associated with shock formation at the inducer inlet when the local $M = 1$ or shock formation in the diffusing volute when the impeller exit velocities become supersonic.
- Disk friction and windage losses on the back face of the rotating impeller.

The effect of these losses on actual head vs. flow characteristics of a compressor compared to the ideal case is illustrated in Fig. 2.6.

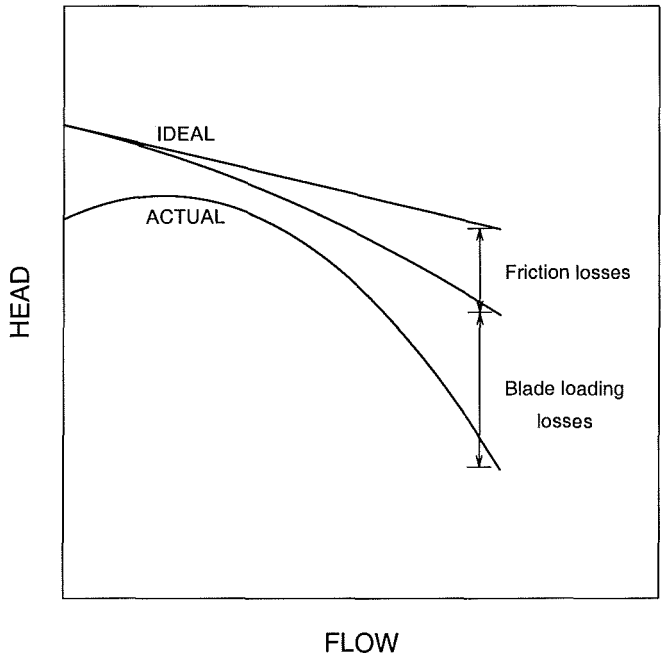

Fig. 2.6 Ideal and actual head vs. flow characteristics

2.2.3 *Compression Processes*

Figure 2.7 shows a diagrammatic representation of a compression process. Isentropic compression, represented by the path 1 – 2s, is a work cycle that does not allow heat transfer into, out of or within the compression system. This is a reversible process, implying the compressed fluid to be an ideal gas.

A polytropic compression process is an irreversible process more akin to real gas compression, shown by path 1 - 2. As the ratio of P_2 to P_1 increases, the gas properties increasingly deviate from an ideal fluid.

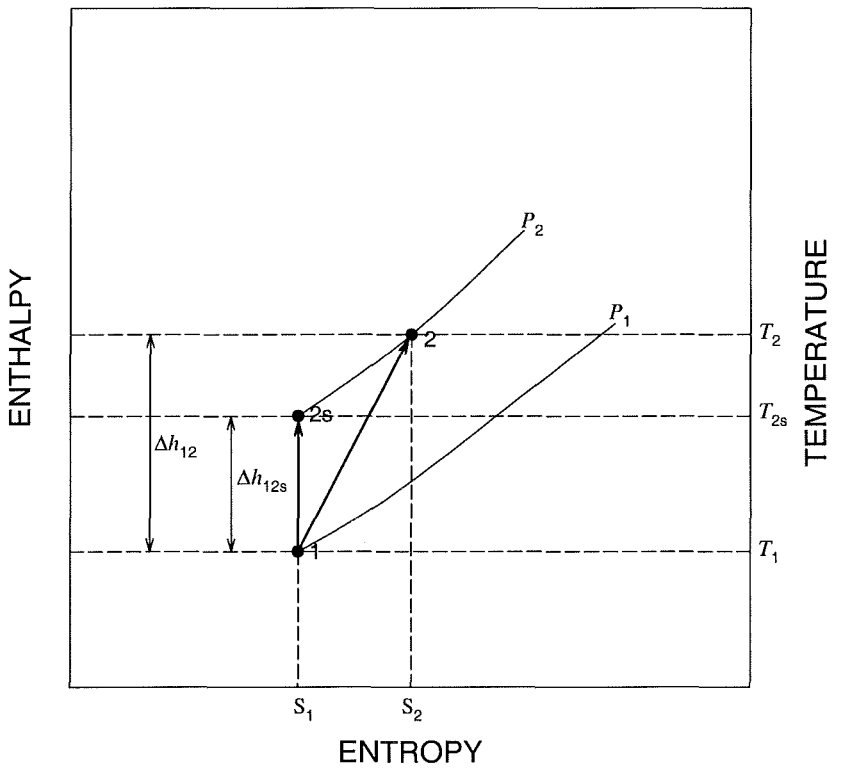

Fig. 2.7 Isentropic vs. actual compression process

An isentropic work process is defined [19] as:

$$P\upsilon^{\kappa} = \text{constant} \tag{2.2}$$

where P is the pressure in a system

υ is the specific volume of the fluid

κ is the isentropic exponent [2]

Isentropic head, H_a is the work done in compressing a volume of gas, v from pressure P_1 to P_2, as follows:

$$H_a = \int_{P_1}^{P_2} \upsilon \, dP \tag{2.3}$$

substituting for υ from $P_1\upsilon^{\kappa} = P\upsilon^{\kappa} = \text{constant}$

$$H_a = \int_{P_1}^{P_2} \left(P_1/P\right)^{1/\kappa} \upsilon_1 \, dP \tag{2.4}$$

Integrating equation 2.4 gives:

$$H_a = \frac{\kappa}{\kappa - 1} P_1 \upsilon_1 \left[\left(\frac{P_2}{P_1} \right)^{(\kappa-1)/\kappa} - 1 \right] \tag{2.5}$$

Then, substituting the real gas equation gives

---

[2] If the polytropic index were to be used the process would be a polytropic work process

$$H_a = \frac{\kappa}{\kappa - 1} Z_1 R T_1 \left[\left(\frac{P_2}{P_1} \right)^{\kappa - 1/\kappa} - 1 \right] \tag{2.6}$$

By inspection, it can be seen that Z_1, R, T_1 and κ influence compressor performance, where:

Z_1 is gas compressibility, which is a function of gas composition, pressure and temperature. Z is normally calculated from an equation of state [20].

R is the specific gas constant defined by the isochoric and isobaric heat capacities of the gas, that is, $R = C_p - C_v$, themselves properties of the gas composition. R is related to the Universal Gas Constant R_0 as follows:

$$R = \frac{R_0}{MW} \tag{2.7}$$

where R_0 = 8.314510 J mol^{-1} K^{-1} , [21] and MW is the molecular mass of the gas which varies with gas composition.

P_2/P_1 is the ratio of discharge pressure to the suction pressure, also known as the pressure ratio, P_r. Note, P_1 and P_2 may vary, keeping P_r constant. It is when P_1 or P_2 vary independently that they affect compressor performance.

T_1 is the gas suction temperature. This affects compressor performance, principally by changing the density, or specific volume, which is a property of the flowing gas [20]. It should be noted that work done on the gas, as it is compressed, causes a temperature rise. The relationship between the suction and discharge temperatures of the gas (T_1 and T_2) can be derived from:

$$\frac{T_2}{T_1} = \frac{P_2}{P_1}^{(\kappa-1)/\kappa} \tag{2.8}$$

The limiting operating temperature of the compressor materials can restrict the value of T_2. The polytropic exponent expression can be inferred from field trials [22] of the compressor from:

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1}\right)^m \tag{2.9}$$

which gives:

$$m = \frac{n-1}{n} = \frac{\ln(T_2/T_1)}{\ln(P_2/P_1)} \tag{2.10}$$

m is the polytropic index and can also be estimated from the isentropic exponent, κ, and polytropic efficiency, η_p, measured during compressor trials. n can be estimated as follows:

$$m = \frac{n-1}{n} = \frac{\kappa-1}{\kappa\ \eta_p} \tag{2.11}$$

κ is dependent on gas composition and can be estimated using a procedure, to link API Technical Handbook procedures, described by Gent [23].

2.2.4 *Dimensional Performance Characteristics (wheel map)*

The isentropic head, H_a, with units of [kJ/kg] is often calculated from the following simple expression derived for an ideal gas, but modified to include real gas behaviour:

$$H_a = \frac{Z_{avg} R T_1}{(k-1)/k} \left[(P_2/P_1)^{(k-1)/k} - 1 \right] \tag{2.12}$$

where P_1 is the suction pressure in [kPa], P_2 is the discharge pressure in [kPa], T_1 is the suction temperature in [K], k is the isentropic exponent, R is the gas constant [kJ/kg-K] and Z_{avg} is the average compressibility factor. The difference between an isentropic and the actual compression process is shown in Fig. 2.7. Due to the irreversible losses mentioned in section 2.2.2, the actual compression process involves an increase in entropy in order to meet the required discharge pressure. Hence, the actual discharge temperature, T_2 is somewhat higher than that corresponding to an isentropic process. The efficiency, η of the compressor is appropriately defined as the ratio between the isentropic head and the actual head:

$$\eta = \frac{\Delta h_{12s}}{\Delta h_{12}} = \frac{(P_2/P_1)^{(k-1)/k} - 1}{(T_2/T_1) - 1} \tag{2.13}$$

The calculated isentropic head and efficiency from equations (2.12) and (2.13) are only approximate for real gases [18]. A more accurate approach involves using an equation of state, such as AGA-8 [20], to estimate these quantities.
In the equation of state (EOS) approach, the static enthalpy and entropy are calculated both at inlet and outlet conditions from thermodynamics relations derived from the state equation. The actual enthalpy change, Δh_{12} is simply the

difference between the calculated inlet and outlet enthalpy. The isentropic enthalpy rise, Δh_{12s}, however, must be determined iteratively from the known outlet entropy, since the inlet and outlet entropy are equal, and the outlet pressure. The efficiency, η is then computed as the ratio of the ideal to the actual change in enthalpy across the unit.

2.2.5 Gas Property Influence Parameters

Thus compressor performance is partly determined by: suction or inlet conditions of temperature and pressure; and gas properties: molecular weight; compressibility and isentropic exponent. Gas properties are dependent on gas composition and operating conditions.

The effects of changes in suction pressure, temperature and gas composition on compressor performance can be summarised in terms of changes in gas operating density for a control volume. An increase in pressure or molecular weight or a decrease in temperature will increase gas density. Similarly a decrease in pressure or molecular weight or an increase in temperature will decrease gas density. Changes in compressor characteristics (at constant speed) due to changes in molecular mass are shown in Figure 2.8.

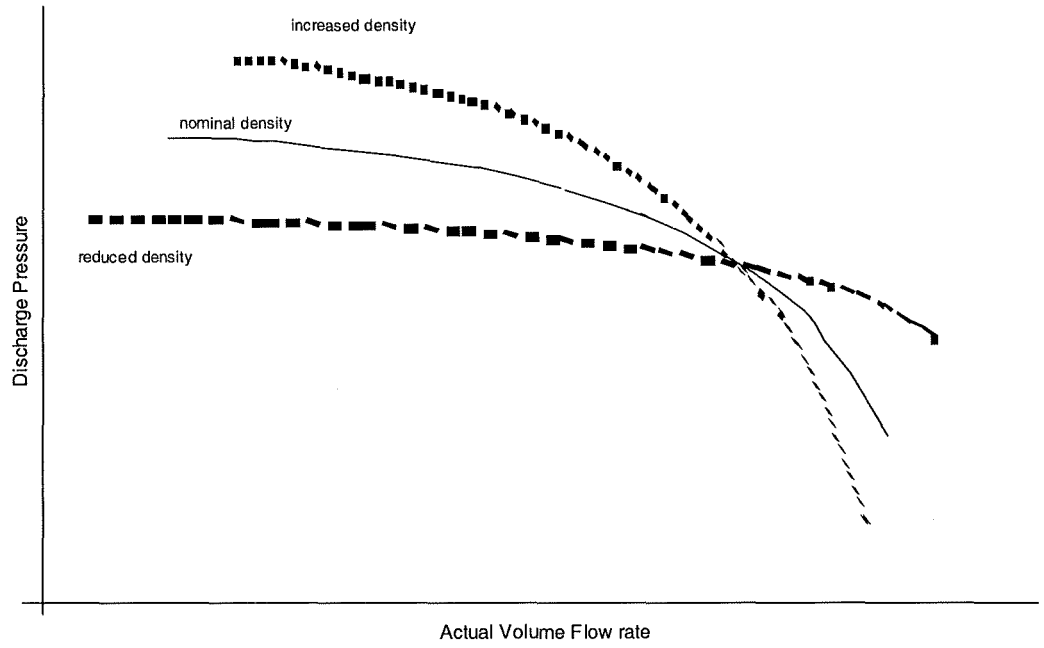

Figure 2.8 - Effects of Changes in Molecular Mass on Compressor Characteristics

2.3 Compressor –Operating Characteristics

2.3.1 Changing COP Rules

There are a number of "rules" which can be deduced from compressor head map diagrams.

1. At constant speed, reducing the flow rate through the compressor increases the compressor head.
2. Reducing the compressor head, at constant speed, increases the flow rate.
3. Increasing flow rate, with a constant compressor head, requires an increase in compressor speed.
4. Reducing the flow rate with a constant compressor head requires the compressor speed to decrease.
5. Reducing the compressor head, at constant flow rate, requires the compressor speed to be reduced.
6. Increasing the compressor head at constant flow rate requires the compressor speed to increase

These cases describe typical goals for selecting COP i.e. maintaining constant throughput or maintaining constant discharge pressure. Compressor speed can be set anywhere within the bounds of the head map, with consequent changes in both discharge pressure and throughput.

Cases 1 and 2 cannot be achieved through varying compressor speed since the compressor speed is constant. To induce changes in operating point of this type requires a change in the process external to the compressor. In the previous section it was noted that changes in gas suction density resulted in changes to the compressor head characteristics. In practice suction or discharge pressure can be adjusted using a control valve. This effectively changes the compressor characteristic and, effectively, fixes a new operating point. Throughput can be adjusted by re-cycling a proportion of the compressor. These methods of compressor control are described in Section 2.3.3 and 2.3.4 .

2.3.2 Choked Flow and Surge

The compressor operating envelope is bounded by the lower and upper operating speeds of the compressor, the surge line and the choked flow line. When the gas velocity, at the impellor inlet or diffuser passages, approaches sonic velocity, M = 1, choked flow will occur. Due to the sonic velocity of the gas, pressure waves are inhibited and increases in velocity (flow rate) cannot occur. Choked is flow is characterised by rapid decrease in discharge pressure for slight flow increases and reduced compressor efficiency. To recover form choked flow the compressor head must be increased or compressor flow rate reduced.

When operating at constant speed, if flow through the compressor is reduced a point is reached where stable operation is not possible. This point is known as the surge point or surge limit of the compression system, i.e. compressor and associated piping system. At the point of surge aerodynamic instability existing within the compressor and it is unable to produce sufficient pressure to maintain continues flow to the downstream system, [9] During surge, flow

reversals and pressure fluctuations occur and, in deep surge cycle, can cause catastrophic failure of the compressor and piping system. Surge can also come about through a reduction in flow rate caused by changes in the piping system resistance. White [7] identified that surge could be induced in a compressor when operated in parallel with another, due to differences in fuel governor characteristics.

Onset of surge is very rapid and incidence of surge might not be detected depending on the type of instrumentation associated with the control system, as identified by Staroselsky [8]. Loss of a compressor can result in significant commercial loss of revenue and market share as indicated by the Nova Chemical press release [24]. This is shown in Appendix A.

2.3.3 Compressor Capacity Control Methods

Operating practice is dependent on the type of compressor in use and the intended application of the compressor. This research is based on performance control of a compressor where throughput is critical. Other compressor applications include maintaining a constant pressure in a process vessel, which is effectively compressor suction pressure control.

A summary of capacity control techniques, as applied to centrifugal compressors, is compiled from sources [3], [9], [15], [25], [26] and shown in Table 2.1.

Drive Type	Principle Method	Supplementary method
Constant speed	Suction throttle	Guide vane angle
		Blow off or recovery turbine
Variable speed	Speed control	Suction throttle
		Guide vane angle
		Continuous Recycle

Table 2.1 – Summary of Capacity Control

Throttling techniques vary the compressor pressure ratio and so vary the flow rate through the compressor. Throttling reduces efficiency but where necessary, suction throttling is preferred over discharge throttling or, in extreme cases, "blow-off", when the compressor discharge is vented to flare or a recovery turbine. Guide vanes require a more sophisticated design of compressor and will result in additional expenditure. Recycle of compressor discharge to compressor suction can be used for capacity control but is the principle anti-surge control method. Speed control varies the speed of the prime mover by manipulating a fuel governor or variable speed electric drive.

2.3.3.1 Constant Speed Compressor Capacity Control

Suction throttle capacity control, combined with minimum flow control is shown in Figure 2.9. If the vessel pressure increases above the set point the pressure control valve will open up, allowing gas flow rate though the compressor to increase. For a constant speed compressor, as the gas flow increases the compressor discharge pressure must reduce. Since the pressure ratio is also constant the compressor suction, or vessel, pressure must reduce. Similarly if vessel pressure falls, the pressure control valve will tend to close. The consequent reduction in flow rate will cause compressor discharge pressure

to rise, thus raising the compressor suction, and vessel, pressure.

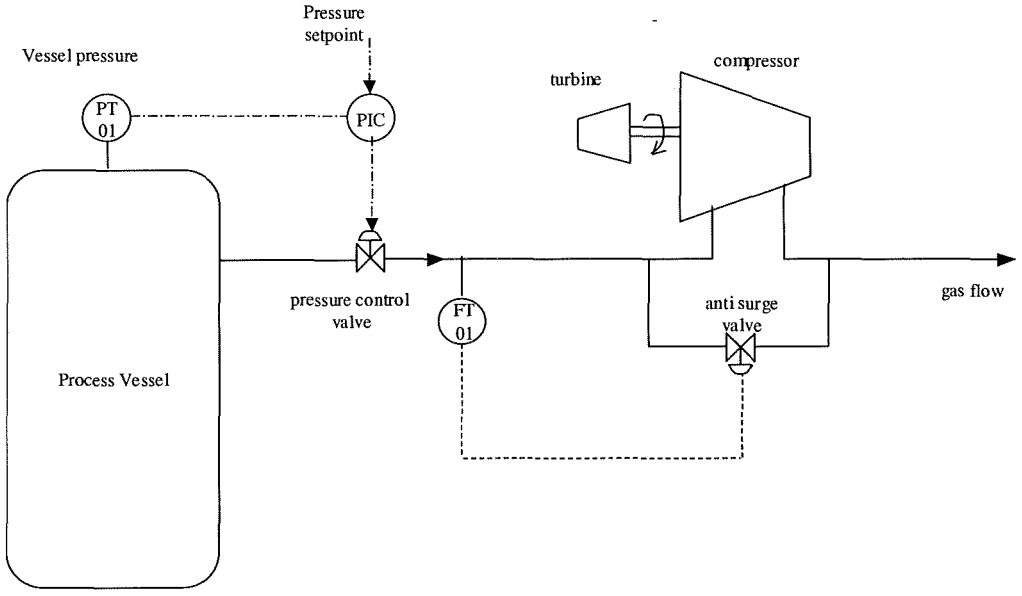

Figure 2.9: Suction Throttle Capacity Control

Minimum flow antisurge control is described in section 2.3.4.

2.3.3.2 Variable Speed Compressor Capacity Control

Figure 2.10 depicts the parameters typically monitored to operate a variable speed compressor with flow rate and discharge pressure control. This type of control may be found on a gas transmission compressor.

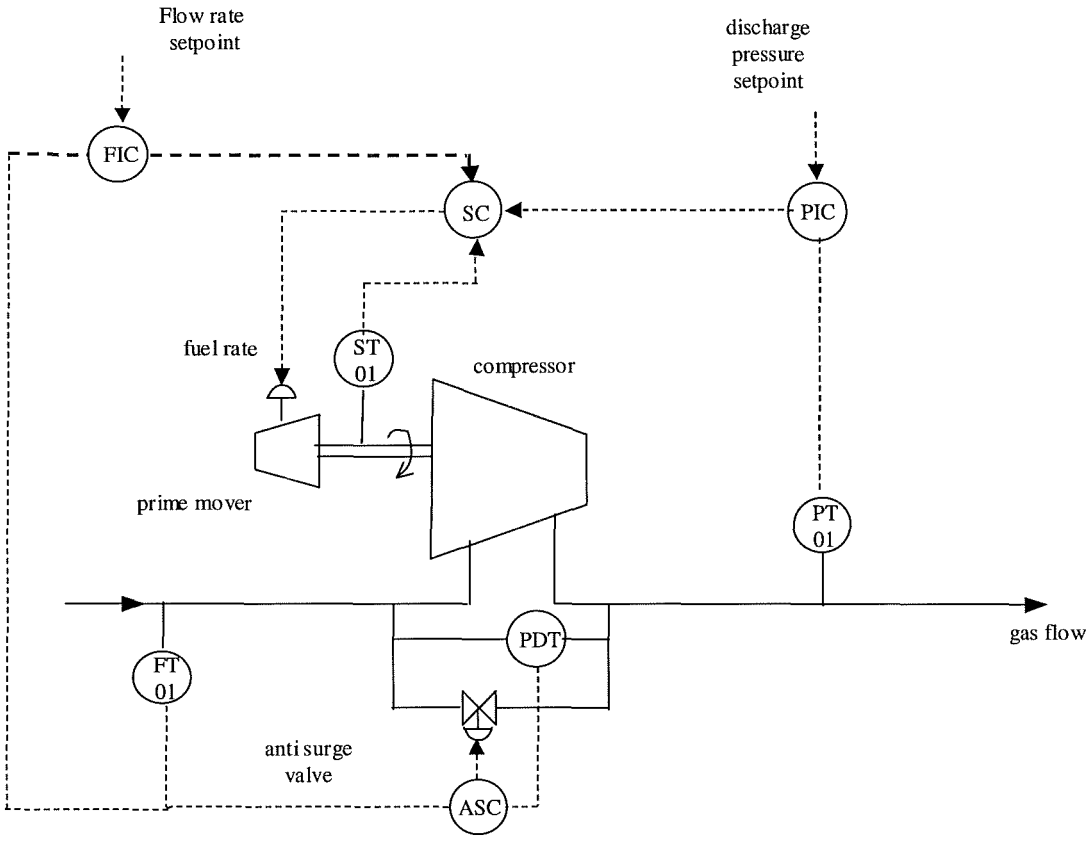

Figure 2.10, "Variable-speed compressor installation"

Two basic types of control are shown in the diagram, surge control and performance control. Performance control is implemented by changing the speed of the prime mover (speed control) to meet a required discharge pressure or flow rate. The speed controller set point is the higher error signal of the flow controller or the pressure controller. In either case, the COP will change. Changes in either controller set point will cause contention between the control loops. Iteration between the two controller set points will be required, as both controlled variables will change if either controller set point is changed. Increasing the flow rate would cause the discharge pressure to fall until the discharge pressure controller applies corrective action to raise the pressure. Similarly, increasing discharge pressure will cause flow rate to fall eventually initiating intervention by the flow controller.

Surge control is effected through a flow-deltap configuration described in section 2.3.4

2.3.3.3 Continuous Recycle Capacity Control

A very simple control scheme may be used where no external work is required to power the compressor such as in a turbo-expander. In this case the compressor is operated with the anti-surge valve partially open, continuously recycling gas, to avoid surge and to avoid over speed of the turbo-expander [27]. Such an arrangement is shown in Figure 2.11. The degree to which the anti-surge valve is open is dependent on the compressor throughput.

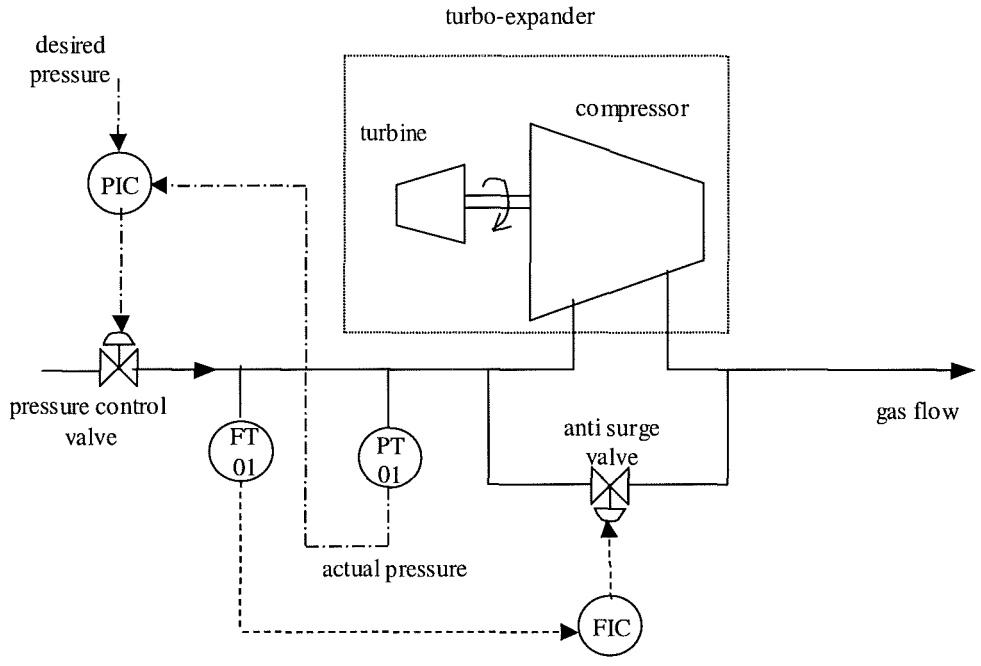

Figure 2.11 - Continuous Recycle Operation

2.3.3.4 Blow off and Recovery Turbine Capacity Control

Figure 2.12 shows blow off capacity control. When the flow controller set point is exceeded the blow off valve opens and gas is vented from the system. This method of capacity control is the least efficient of the capacity control methods described.

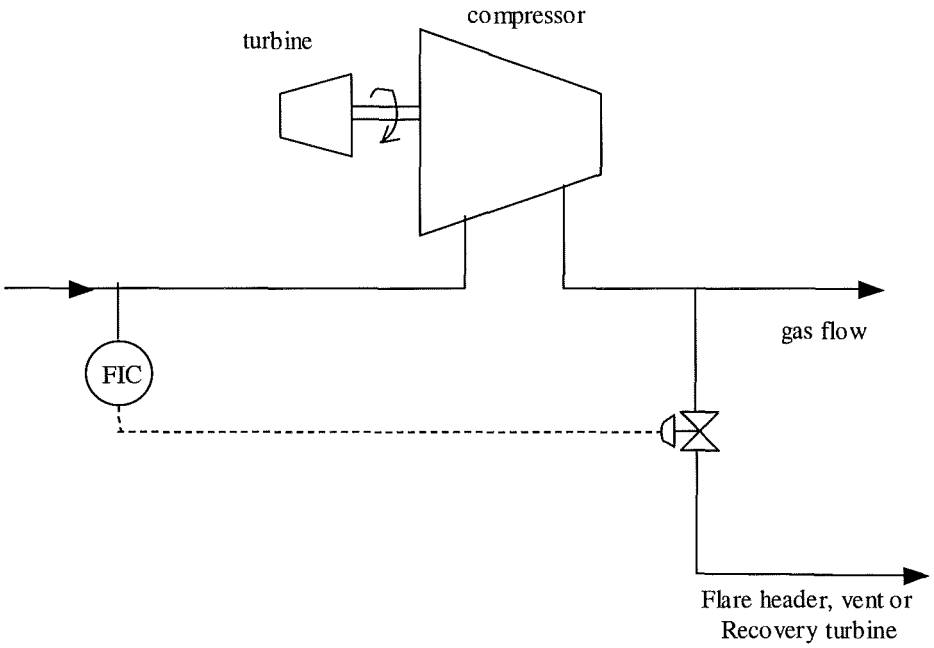

Figure 2.12 – Blow off Recycle Operation

In the case of a recovery turbine the blown off gas is expanded across a turbine to recover work from the compressed gas.

2.3.4 Compressor Anti Surge Control

In addition to capacity, or process control, protective control is required for a compressor, usually referred to as anti-surge control. Two basic methods of estimating proximity of COP to surge point are: minimum flow control and "flow-deltap". In proximity to surge, flow through the compressor will either be recycled or, less commonly, blown off. Table 2.2 lists the most widely implemented methods of surge control, compiled from sources [3], [9], [15], [25], [26].

Drive Type	Method
Constant speed	Minimum flow
	Power & head
Variable speed	Flow-deltap

Table 2.2: Anti surge control methods

2.3.4.1 Minimum Flow Anti-Surge Control

On a constant speed compressor the surge point occurs at a constant inlet volume flow rate. Minimum flow surge protection requires an anti-surge control inlet volume flow rate limit to be set, as shown in Figure 2.13. If the flow rate limit is breached anti-surge controlling action is invoked, which may be gas recycling or blowing off.

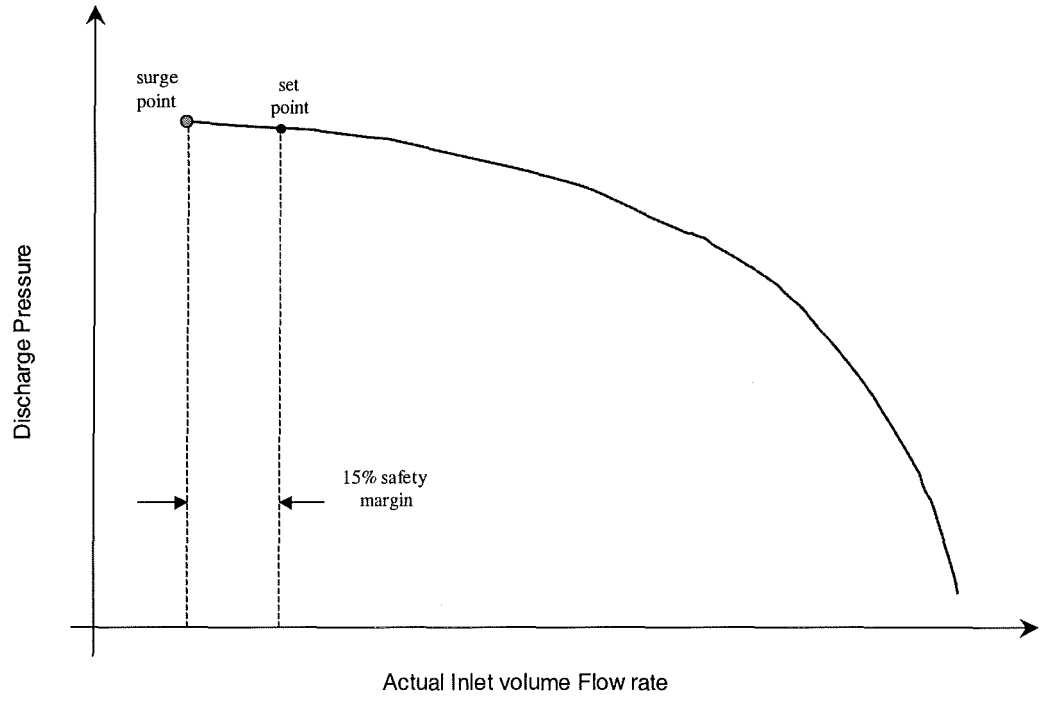

Figure 2.13: Minimum Flow Surge Safety Margin

Variations in suction pressure do not affect the surge point, however suction pressure measurement is required for actual inlet volume flow rate measurement, as changes in suction pressure would tend to reduce the safety margin. Changes in molecular mass and temperature tend to increase the safety margin therefore compensation for these parameters is not necessarily required. Large changes in these parameters can be offset in changing the minimum flow set point. A minimum flow control scheme for a constant speed compressor with suction throttling is shown in Figure 2.14.

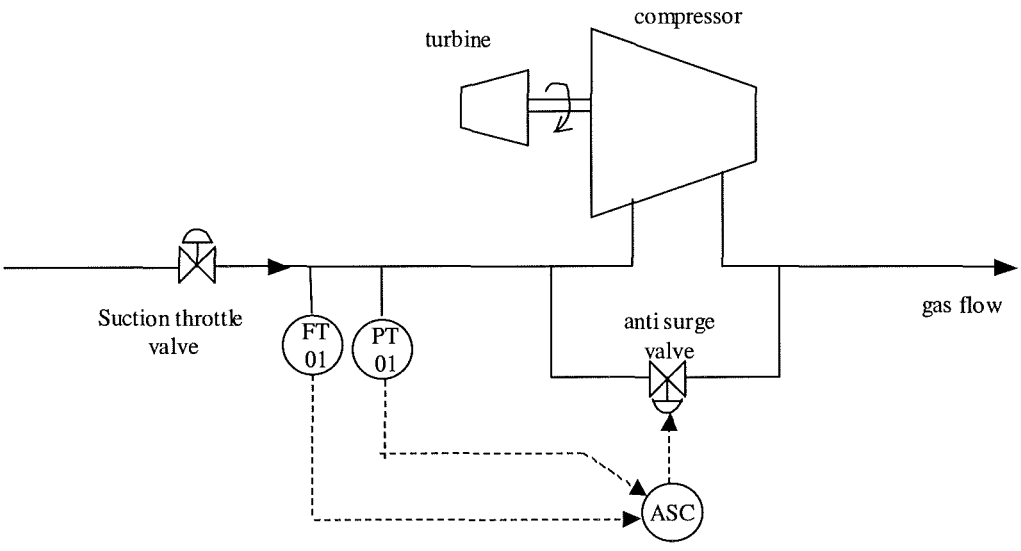

Figure 2.14: Minimum flow anti-surge control

2.3.4.2 Power & head Anti-Surge Control

An alternative form of constant speed compressor, anti-surge control is based on measurement of prime mover power and compressor discharge pressure. This is based on the relationships that power absorbed by the compressor is proportional to the weight flow of the gas and so the suction pressure, through suction density. Constant pressure ratio relates suction pressure to discharge pressure. This method does not require the expense of a flow meter but is also subject to error when changes in suction temperature and molecular mass occur.

This anti-surge control scheme is shown in figure 2.15.

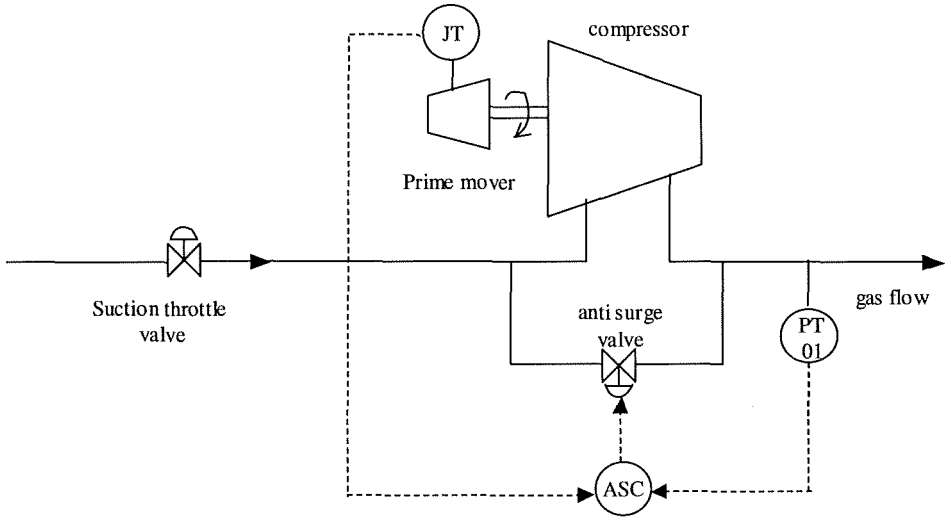

Figure 2.15: Discharge pressure & head antisurge control

2.3.4.3 "flow-deltap" Anti-Surge Control

The surge control scheme shown in figure 2.16 operates on the "flow-deltap" anti-surge control scheme and is most widely used on variable speed compressors.

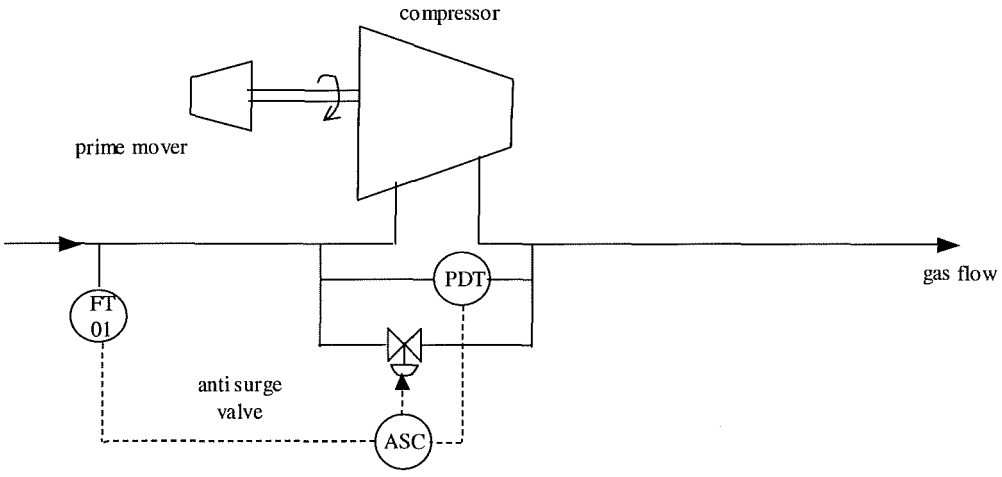

Figure 2.16: "flow-deltap" antisurge control

A flow-deltap antisurge control can also be used, but to a lesser extent, on constant speed compressors. White [7] described that the pressure rise across a compressor was proportional to the square of the inlet volume flow rate. In addition, the inlet volume flow rate at which surge occurs is proportional to compressor speed, Boyce [9]. If the inlet volume flow rate is measured using a differential pressure producing meter then an equation for the surge point would be a straight line of the form:

$$P_d - P_s = Ch \tag{2.14}$$

where C is a constant of proportionality, h is the meter differential pressure and Pd-Ps is the pressure rise across the compressor. The linear representation of a surge line on a variable speed compressor is shown in Figure 2.17.

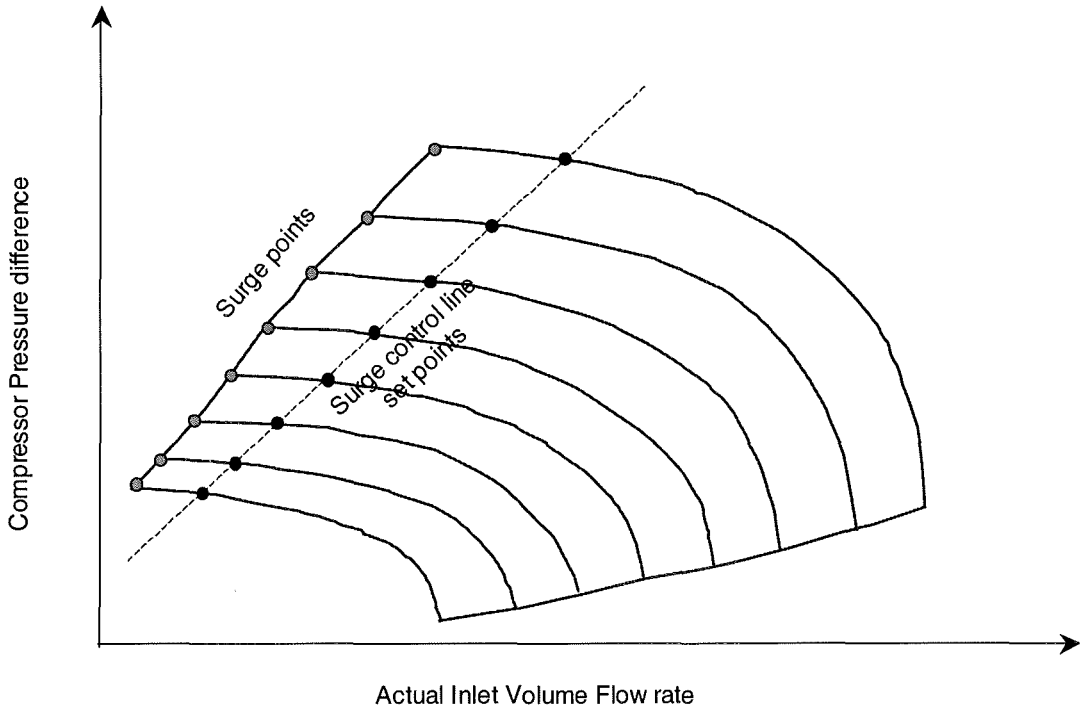

Figure 2.17: Variable Speed Compressor surge line

Also shown is a surge control line, the construction of which is described in section 2.4. Differing configurations of pressure and temperature compensation with discharge volume flow measurement can be devised. These are dependent on compressor system installation and take account of piping configuration including, for example, elements like interstage coolers [9].

The advantages of flow-deltap control are that: it can be used on variable or constant speed compressors; it is practicable to implement; it is effective in controlling action and is independent of suction pressure and compressor speed.

2.3.5 Series and Parallel Compressor Anti surge Control

Compressor can be connected in series to increase discharge pressure or connected in parallel to increase flow rate.

Series compressor can be driven for the same shaft and therefore can be protected by the same antisurge valve. Series compressors driven from different shafts, require separate anti surge valves or, more commonly, separate antisurge controllers operating a common antisurge valve. A typical antisurge arrangement., with a single recycle valve, is shown in figure 2.18. Each compressor has a dedicated antisurge controller. The surge signal variable from each of these is input to a "low signal selector" which routes the signal nearest to surge to the antisurge recycle valve actuator.

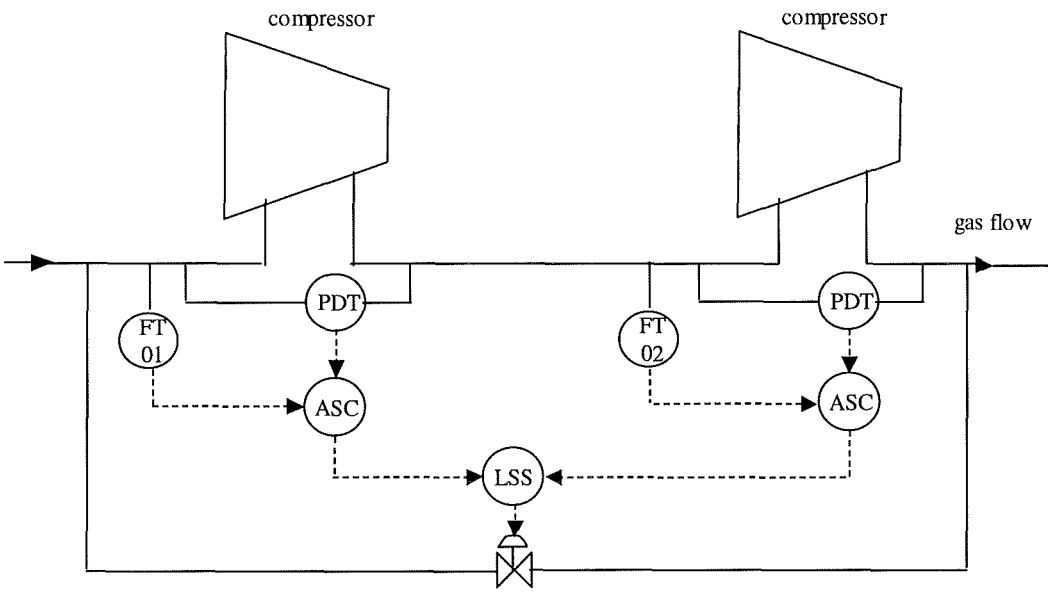

Figure 2.18: Series compressor antisurge control

Due to the "flat" nature of the centrifugal compressor discharge pressure curves, slight difference between discharge pressures might be sufficient to induce surge in one of the compressors. Changes in load would not affect each to the same extent so each compressor should have a dedicated anti-surge valve and controller, [25].

2.3.6 Parallel Compressor Load Sharing

White [7] identified the difficulty of load sharing in parallel operation compressor configurations attributable to dissimilar characteristics in the prime mover fuel governors. In addition wear over time will change compressor performance characteristics. To overcome these problems and associated propensity to surge, it is possible to balance the individual compressor suction flow rates as shown in figure 2.19:

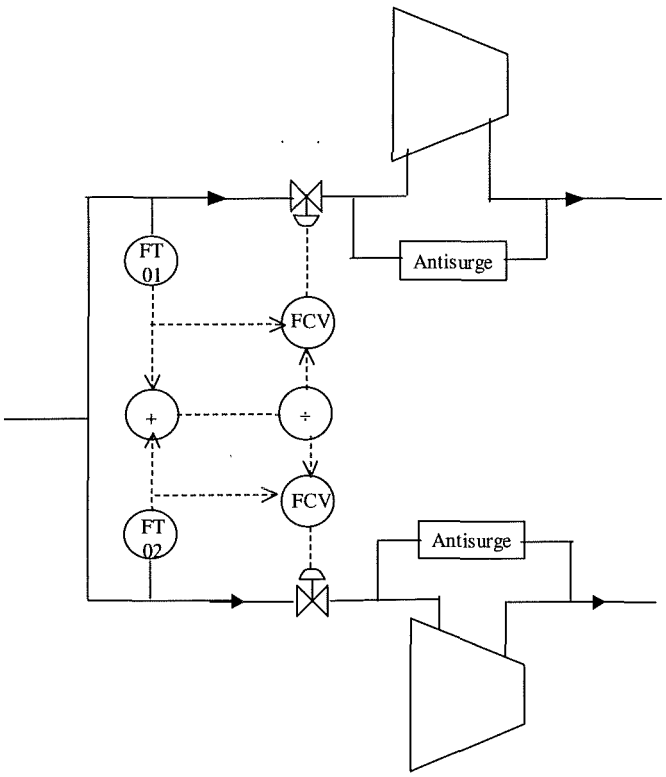

Figure 2.19: Parallel compressor load balancing

A proportional loading scheme, for compressors of different sizes, is shown in figure 2.20. This allows compressors of different sizes , but similar surge point characteristics to share load at an operating point "equidistant" from their respective surge lines.

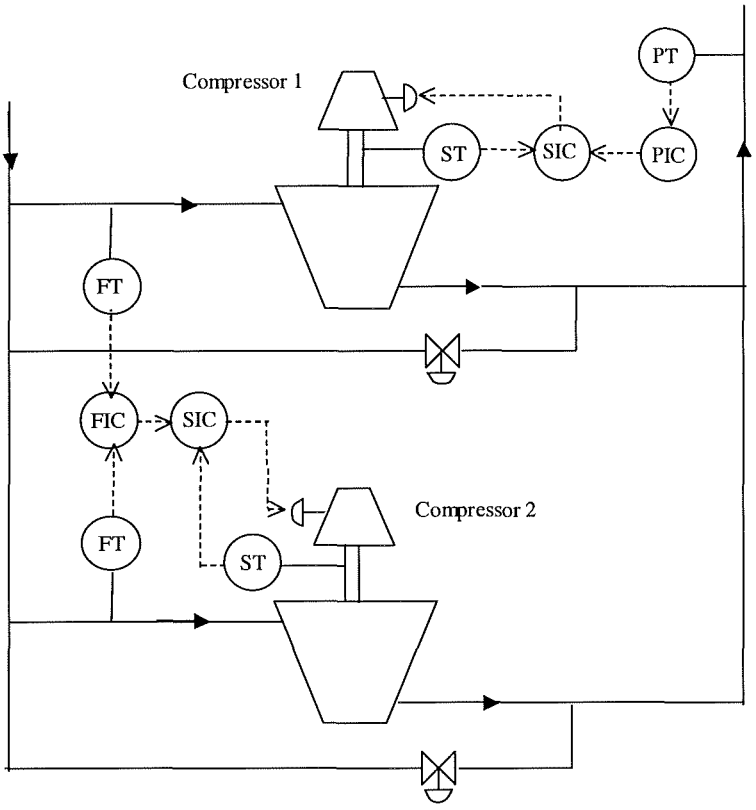

Figure 2.20: Proportional load sharing

More elaborate control systems are described, [25] which can apportion total load between compressors using suction volume flow control.

2.3.7 *Efficiency vs. Efficacy*

Capacity throttling is less efficient than changing speed to achieve the desired COP, since energy is dissipated as a pressure drop across the throttling valve. Suction throttling requires additional energy input to recover lost energy. Discharge throttling dissipates energy supplied to the gas by the compressor, which increases prime mover fuel consumption and exhaust emissions. Energy wastage increases operating fuel costs. Exhaust emissions have both a monetary and environmental cost.

Surge control is required to prevent damage to the compressor. For a fixed-speed machine this is achieved by detecting a flow rate close to the surge point flow rate and then opening the recycle valve. When this happens, the recycle valve opens the compressor discharge to the compressor suction, thereby circumventing the initiation or continuation of the forward/reverse flow cycle. In a variable-speed machine, surge detection is more sophisticated, but the controlling action is the same, that is, operation of the re-cycle valve. Continuous operation with the surge control valve open, means that gas is recycled through the compressor, which is wasteful of energy.

Operating with continuous recycle of gas may allow the compressor to operate at a thermodynamically efficient COP hence it may be described as an *efficient* operating point. A thermodynamically efficient COP means that the discharge temperature will be around the lowest possible for the achieved compression ratio. However, gas which is recycled wastes prime mover fuel so whilst the compressor efficiency may be high the *efficacy* of the compressor may be described as poor. At 100% efficacy no gas is re-cycled i.e. all of the compressed gas is absorbed by the process that the compressor feeds. The most efficacious operating point is where the discharge pressure is just sufficient to sustain the required forward flow.

2.4 Controller Implementation of "flow-deltap" strategy

Current compressor control technology has the flow-deltap control scheme as its basis. The "flow-deltap" characteristic can be depicted graphically as shown in figure 2.21 where the parameters of compressor pressure rise, discharge pressure less suction pressure (Pd-Ps) and suction orifice flow meter differential pressure, dp. The origin on the graph is actual inlet volume flow rate = 0 and Pd = Ps i.e. Pd-Ps = 0.

The surge point is fixed whilst the operating point moves along the compressor speed or performance characteristic curve. As the operating point approaches the surge point the gradient of the lines become the same i.e. their ratio approaches unity.

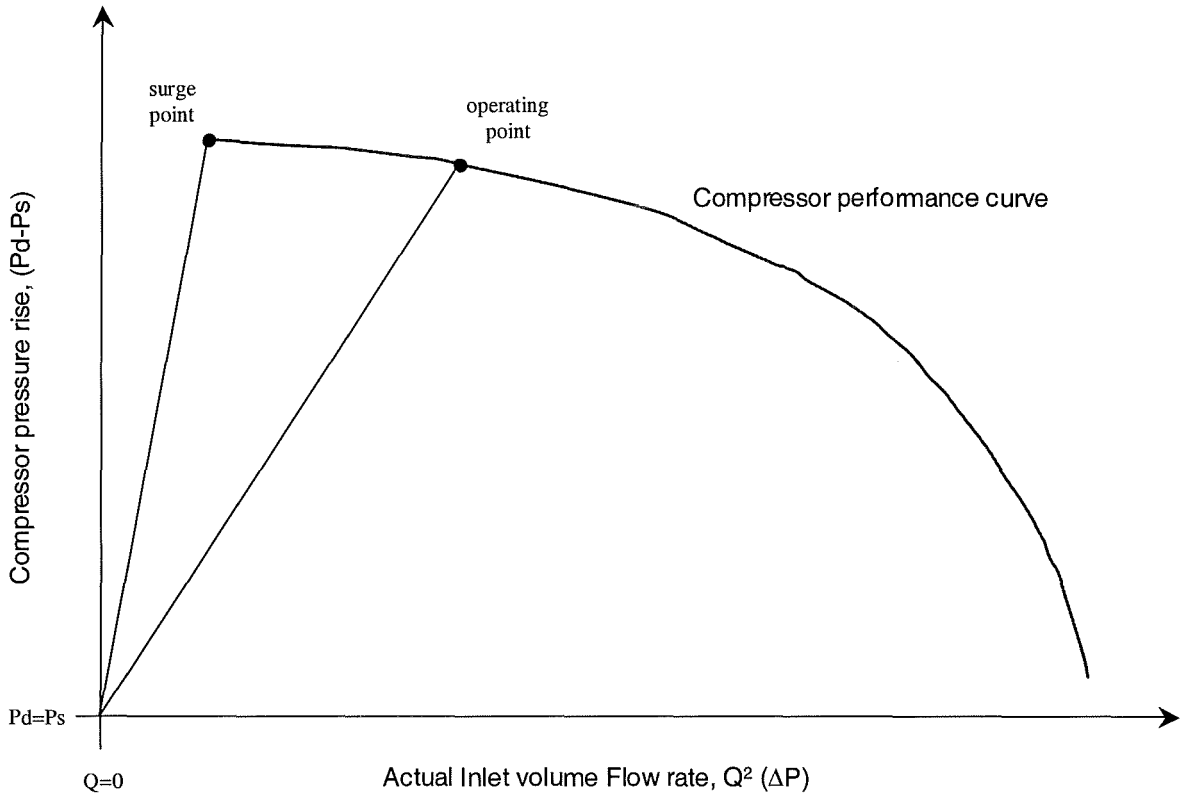

Figure 2.21: Compressor "flow-deltap" diagram

White [7] described how the compressor pressure rise or pressure ratio is proportional to the head of the compressor hence the gradient of the operating line at any single point is expressed as:

$$\frac{H}{Q^2} = cons\tan t \tag{2.15}$$

Staroselsky [8] developed a similar relationship for polytropic head using actual compressor instrumentation inputs:

$$\frac{H_p}{Q^2} \alpha \frac{P_s \dfrac{R_c^m - 1}{m}}{dp_o} \tag{2.16}$$

where H_p is the polytropic head, Q is the actual inlet volume flow rate, P_s is the suction pressure, R_c is the compressor pressure ratio, m is the polytropic index and dp_o is the differential pressure across the suction flow rate meter, proportional to the flow rate. Staroselsky further defined that a surge criterion, S could be described as [28] :

$$S = \frac{\left[\dfrac{P_s \dfrac{R_c^m - 1}{m}}{dp_o}\right]_{operating\ \ po\ \text{int}}}{\left[\dfrac{P_s \dfrac{R_c^m - 1}{m}}{dp_o}\right]_{surge\ \ po\ \text{int}}} \tag{2.17}$$

When S = 1 the compressor operating point is coincident with the surge point i.e. the compressor will surge. When S < 1 the compressor is working to the right of right of the surge point. By specifying an S value it is possible to set the operating point of a compressor on a performance characteristic, be it a constant speed curve or universal speed curve (described in section 3), relative to the point of surge. In this respect setting the operating point, (performance control), of the compressor is implicit in surge control i.e. it is relative to the proximity of the surge point.

The S value of compressor point, for fixed compressor speed and guide vane angle, can be calculated as:

$$S_{op} = K_1 \left[\frac{P_s \dfrac{R_c^m - 1}{m}}{dp_o} \right]_{operating} \tag{2.18}$$

where K_1 is the inverse gradient of the surge point line. K_1 can be defined as a function of compressor speed, f(N), and guide vane angle, f(α). A margin of safety, as a function of actual inlet volume flow rate $K_2 f(dp_o)$ as shown in Figure 2.22, can be added to create a surge limit line. S is redefined as:

$$S_{op} = f(N) f(\alpha) \frac{P_s \dfrac{R_c^m - 1}{m}}{dp_o} + f(dp_o) \tag{2.19}$$

Now, when S = 1 the operating point lies on a displaced surge limit line at which point corrective controlling action can be take to prevent surge.

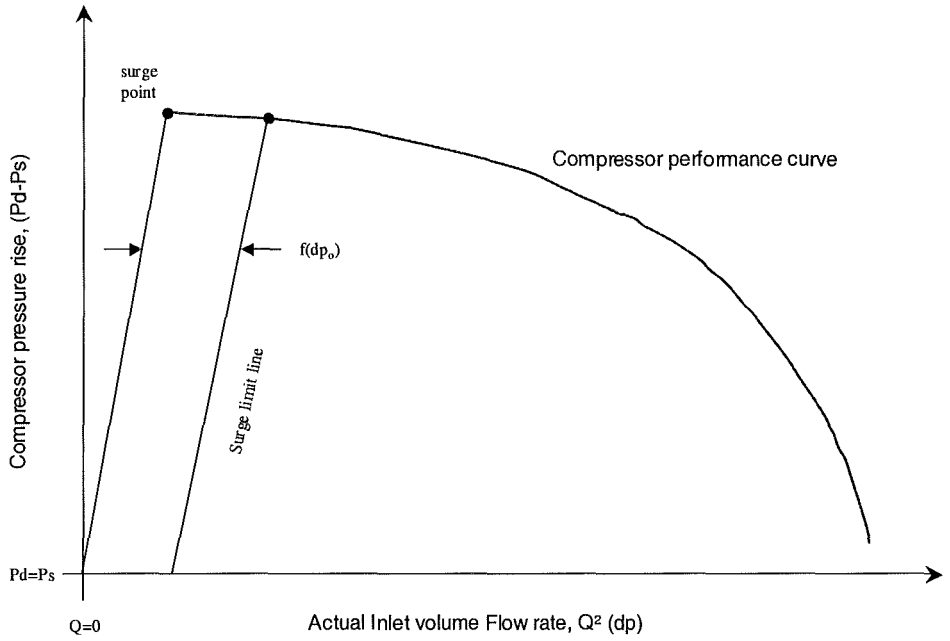

Figure 2.22: Surge Limit line

With the advent of digital controllers, more complex surge line contours, associated with higher compression ratios, could be represented. In these cases the compressor surge line characteristic is not parabolic. The linear approximation described by White [7] may, therefore, not hold. Surge line representation by linear segmentation is shown in Figure 2.23.

The "flow-deltap", implemented in proximity to surge point parameters, approach to compressor control was summarised as being current practice by deSa and Maalouf [29].

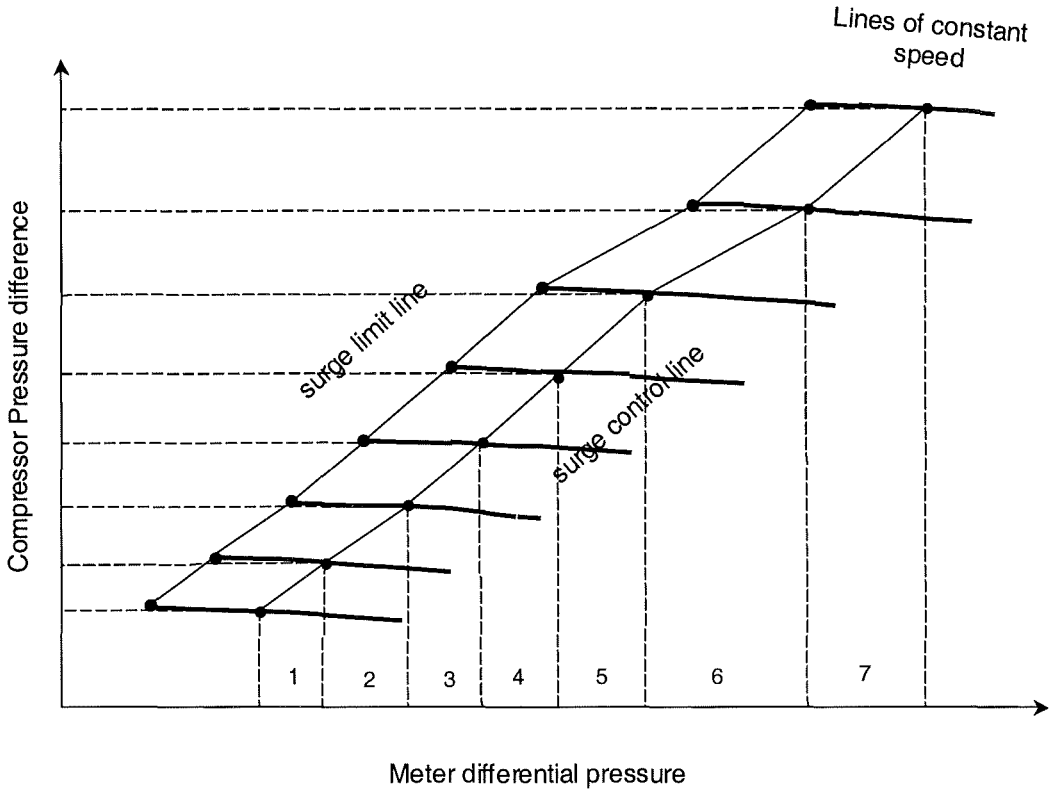

Figure 2.23 - Surge Control Line Characterisation

A control scheme to implement S, as the controlled variable, is shown in Figure 2.24. The surge controller calculates the S factor, based on field inputs, and passes it to a performance controller as the current operating point. (The field inputs could be any of those defined by Batson [10] representing a set of invariant co-ordinates). The desired operating point is the set point as advised by an optimisation system or the operator. Simple PID control is used to generate a set point for the prime mover fuel governor.

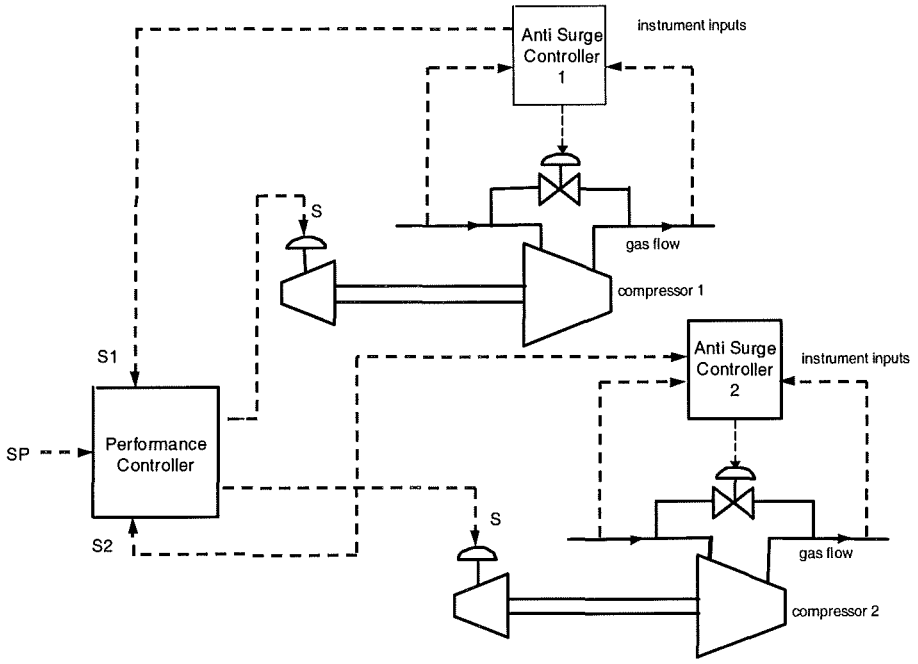

Figure 2.24: S-factor control

Whilst this control technique avoids compressor surge and its consequences, it does not address control in terms of fuel efficiency. This is generally left to an advisory optimisation routine or the control room operator.

Control of multiple compressors, as found in a gas transmission station for example, is somewhat more complex using these conventional control techniques. This is particularly the case where compressors of dissimilar performance characteristics are run in parallel. Conventional performance control selects the highest S value operating point, i.e. the compressor most likely to surge. This value is then scaled for each of the prime mover ranges in use and then downloaded as the fuel governor set point. The compressors run at the same COP on their respective performance characteristics. This is termed "equidistant load sharing" - equidistant in that the distance between the COP line and the Surge control line is the same for each compressor.

Performance control is still centred around surge prevention of the compressor closest to surge and does not reflect fuel efficiency considerations.

2.5 Conditional Stability

Analysis of the gradient ratio control line is described with reference to Figure 2.25. Assume the COP is set on the nominal suction pressure curve. The compressor is in a dynamic equilibrium where discharge pressure and flow rate are balanced against the discharge pressure constraint, shown by the straight line. The controlled variable set point, the ratio of gradients, is shown at point 1.

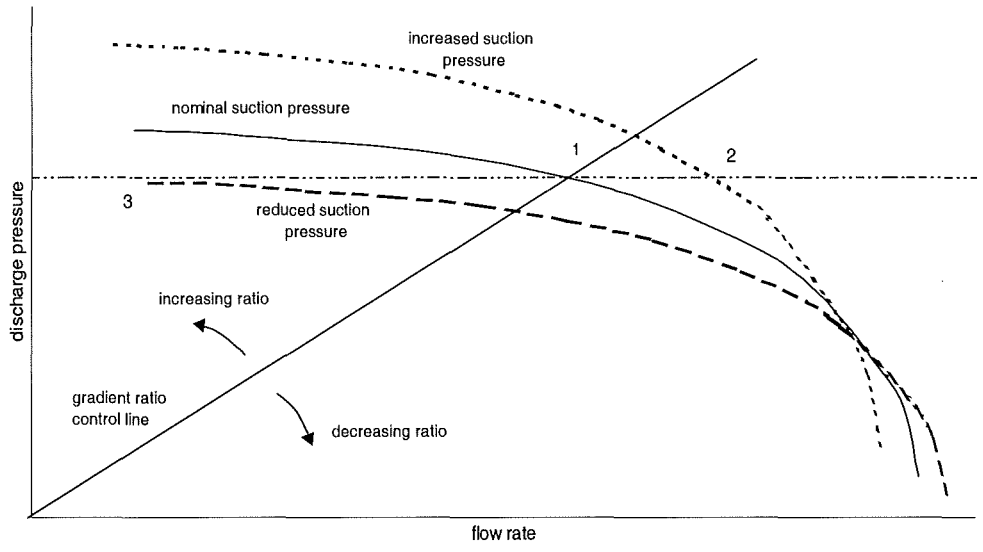

Figure 2.25 – Conditional Stability

If an increase in suction pressure occurs the compressor will try to reach a new dynamic equilibrium at point 2. Without a change of set point, the control system attempts to bring the controlled variable back to point 1, but the compressor now moves on the increased suction pressure characteristic. The equilibrium of the compressor will oscillate along the increased suction pressure curve, between points 1 and 2. This *conditional stability* will result in fuel wastage and increased exhaust emissions as the discharge pressure and the flow rate are always greater than required, relative to the original COP, point 1.

If a decrease in suction pressure occurs, the compressor characteristic follows the reduced suction pressure curve. The compressor will attempt to reach a new dynamic equilibrium COP, towards point 3. As before the control system will try to maintain the ratio gradient towards point 1. The compressor will oscillate between points 1 and 3 and eventually surge as the discharge pressure is backed up by the pressure constraint. The compressor system is now unstable. The dynamics of the instability are determined by the inertia of the rotating components.

Unless the set point is changed to accommodate changes in suction conditions and gas properties, a control system based on this philosophy is only unconditionally stable at one point. This point is where the discharge pressure and flow rate are balanced against the discharge pressure constraint with suction conditions and gas properties fixed.

2.6 Compressor Control Trends

Botros & Henderson [11] identified two main (anti-surge) control trends. These are:

- Designed-In. This is the integration of surge prevention mechanisms *designed in* to the compressor design. These include: variations in impeller blade angle and blade thickness; varying the diffuser shape; adjusting the inlet guide vane geometry; addition of casing slots; and so on.
- Operational devices. This is the use of operational devices such as: aerodynamic techniques involving the discharge gas re-injection with directional nozzles (Gysling and Greitzer, *et. al.* [30]); various anti-perturbation pressure oscillations schemes described by Jungkowski *et. al.* [31].

They identified that surge control technology is in a 'perfection' stage and that control advances will come about through incremental improvements in existing technology and improvements in digital computer methods.

In the context of this research, application of fuzzy logic to conventional surge control was described by Cordiner [14]. Fuzzy logic was used to weight the significance of the anti-surge control factor. At 'LOW' values of S, away from the surge point, optimisation of COP was the most prevalent control factor. At 'HIGH' values of S, anti-surge considerations were the prevalent control factor and control was transferred to the anti-surge controllers. Under this control strategy each compressor was operated at a unique S value hence load sharing was "non-equidistant". Performance control could be considered from a fuel efficiency perspective. The difficulty then would be in selecting an optimised COP. Cordiner [14] further identified that "fine tuning" or optimisation of the fuzzy controlled variable could be attempted using neural networks. This control scheme represents an incremental improvement in anti-surge control technology methodology.

2.7 Discussion

This chapter has built up a broad overview of the principles of, and the operational implications, for compression. Compressor control development has been researched and described. Compressor control pivots around prevention of surge or anti-surge control. Performance control, from a point of view of compressor efficacy or fuel efficiency, is a secondary consideration. Prevention of damage to high capital cost equipment and costly downtime of process critical plant is a key driver. Other factors such as environmental emissions restrictions and pressure on operating costs require most efficient use of fuel and extended plant availability. These require reliable and accurate *performance* control.

Effective performance control needs to hold the compressor at some operating point where capacity throughput and/or pressure constraints are met. This is likely to be achieved through the use of a steady state control scheme where averaging the performance control can meet longer term control objectives against plant transients. Performance control and anti-surge control

are complimentary: effective, longer-term performance control should reduce the propensity of the compressor to surge and effective anti-surge transient control improves the efficacy of the compressor.

Henderson & Pearson [32] described a steady-state control scheme for a micro-hydrogenerator set. This method used the force balance equilibrium characteristics of the generator set and electrical load. The speed of the generating set was adjusted by applying or removing electrical load, based on defined characteristics. Transient control was implemented by other systems but improvements to transient response were suggested by the use of under or over correction of steady state loading, see Henderson & Pearson [33]. The recycle valve in an anti-surge control system can be thought of as load control of the compressor where opening the re-cycle valve reduces the "load" on the compressor. There are similarities too between the derivative control action described for the generator set [33] and a system described for a quick opening solenoid compressor recycle valve [26], improving surge control response.

In a compressor system the characteristics are less well defined and affected by plant external to the system including process load. The compressor, however exhibits the same tendency to settle at a natural equilibrium operating point. If the compressor characteristics are well known then it would be a simple matter to set the compressor speed based on load constraints of capacity throughput and pressures. Compressor optimisation schemes are available which recommend COP based on process conditions or plant load. These are in effect, Model Predictive Control, MPC, schemes where compressor characteristics are predicted using mathematical modeling techniques. Such systems often require expert intervention and can be un-reliable.

The work described by Cordiner [14] retains, as its basis, conventional anti-surge control with fuzzy categorisation of the controlled variable S. This has allowed independent S values to be used for each compressor. His suggestion regarding the use of neural networks for fine tuning COP was the starting point for this research.

2.8 Conclusions

Having reviewed principles, operation and control of compressors it is clear that there is a growing need for reliable performance control as environmental issues attract greater commercial penalties such as pollution tax and restriction on production due to pollution quotas. Given the similarities between compressor and micro-hydrogenerator operating equilibrium points there is merit in considering the performance control of compressors as an exercise in operating at natural equilibrium points determined by the process load. This requires knowledge of the compressor performance at various points in its operational envelope. This is very similar to MPC however there is an essential difference. This research describes a technique to use neural networks to approximate the compressor characteristics instead of using conventional modeling techniques.

As described in chapter 5 supervised training techniques will allow a neural network to be trained with actual compressor data from operational plant. The neural network can be re-trained (or replaced) to reflect changes in compressor characteristics through time. The decision to update compressor characteristics can be left to the experience of Operations Staff and will not require intervention by modeling experts or mathematicians. Hence the neural network will embody the "knowledge" of Operations Staff, one element of a knowledge based or expert system, as described in chapters 6 and 7. This knowledge will represent goal-oriented operation of the compressor i.e. longer-term goal oriented, surge free operation.

This aspect of the research is not believed to have been previously examined in the context of compressor control and hence gives opportunities to be researched. In this respect, the research described in this thesis constitutes a novel contribution to compressor control and, in a broader sense, to the control of rotating equipment.

Whilst performance control is the primary concern here, there is the potential for development of a diagnostic aid to compressor performance. Several neural

networks might be trained throughout the compressor life either by time or by event using actual operating data. Compressor performance can be tracked and deterioration monitored to construct a Reliability Centered Maintenance, RCM, programme tailored to each compressor *in situ*.

3 Developing a Compressor Model

3.1 Introduction

Recent modelling of compressor stations under various operating conditions has been presented, most notably Botros et al, [34-38]. A review of earlier compressor modelling is to be found in Boroomand [39]. These references describe various numerical solutions to dynamic simulation of compressor stations comprising compressor, piping elements and associated control system. The principle numerical solution techniques in use are Method of Characteristics and Finite Difference. Each of these simulations contains a compressor element originally described by Greitzer [40-41] for aero-engine axial compressors. Greitzer's work was extended to centrifugal compressors by Hansen [42]

A linearly distributed parameter compressor system model was described by Badmus [43-44] which claimed improvement over the lumped parameter model, first order empirical compressor lag described by Greitzer [40-41]. This model was again, developed for research into aero-engine control.

This chapter discusses the basis of estimating steady state compressor operating point based on the application of Dimensional Analysis and Fan Laws. One such approach is described by Godse [45]. The principles described are used to generate data sets for researching neural network based compressor performance control.

3.2 Dimensional Analysis

Dimensional analysis is an invaluable tool towards understanding the general behavior of a turbomachine by reducing the number of pertinent physical variables for the unit into a smaller number of dimensionless groups. It has several important applications for it's process compressors:

- Allows direct comparison between a family of wheels from the same or different manufacturers based on the non-dimensional performance characteristics. This improves ability to select the most suitable type of wheel over the estimated head and flow range.
- Monitor changes in the non-dimensional performance characteristics for deterioration in the unit operation. This could be used for predictive maintenance by developing a fault signature table in order diagnose a specific problem area within the unit.
- Extrapolating the non-dimensional performance curves to estimate the choke limit (maximum flow rate) for compressor.

The main benefit of utilizing non-dimensional performance variables, as opposed to the conventional head, flow, efficiency and unit speed, is that only two non-dimensional curves are needed instead of numerous speed and efficiency lines that are included on the wheel map. This allows normalized operating points to be plotted and directly compared with each baseline performance curve. Conversely, an operating point may lie in the middle of the wheel map, yet it would be impossible to determine whether the unit was functioning properly without specifying the unit speed for the data. Further, it is a straightforward procedure to compare the non-dimensional performance characteristics of two or more units.

Where compressor speed is specified, non-dimensional performance characteristics can be used to estimate, or model, compressor performance characteristics for various operating conditions. Dimensionless parameters are identified by reducing the number of dimensional variables, as described in the next section.

3.2.1 Buckingham pi theorem

There are several methods of reducing dimensional variables into a smaller number of dimensionless groups. The most commonly used approach was proposed by Buckingham in 1914, and is appropriately named the "Buckingham" pi theorem [46]. The pi theorem, and all other dimensional analysis techniques, are based on the following assumptions:

i.) The principle of dimensional homogeneity that states if an equation correctly characterizes the relationship between dimensional variables for a given physical process, it must be dimensionally homogeneous; i.e., each of its additive terms will have the same dimensions.

ii.) All of the relevant dimensional variables for the problem must be included in the analysis.

The first part of the pi theorem then states what reduction in variables is possible:

- If it is known that a physical process is governed by a dimensionally homogeneous relation involving n dimensional variables, it can be reduced to a relation between only k dimensionless variables or Π's. This reduction, $j = n - k$, equals the maximum number of variables which do not form a pi among themselves and is always less than or equal to the number of fundamental dimensions involved in the dimensional variables [47].

The second part of the pi theorem explains how to find these dimensionless groups:

- After determining the reduction amount, select j variables that do not form a Π group among themselves. Each desired pi group will be a power product of these j variables plus one additional variable that is assigned any convenient non-zero exponent. Every pi group found in this manner is independent [47].

A simple example of estimating the total force, F_T acting on a bluff body moving through a fluid is included to clarify these concepts. This force depends on the diameter of the body, D along with the velocity of the body, V and the fluid density, ρ and viscosity, μ. This relationship may be expressed as:

$$F_T = f(D, V, \rho, \mu) \tag{3.1}$$

These five variables ($n = 5$) contain the following dimensions, as shown in Table 3.1.

F_T	D	V	ρ	μ
$[MLT^{-2}]$	$[L]$	$[LT^{-1}]$	$[ML^{-3}]$	$[ML^{-1}T^{-1}]$

Table 3.1: Force Dimension Table

The first part of the Buckingham pi theorem states that the reduction j is less than or equal to the number of fundamental dimensions M, L, T contained in the variables. Thus, j is less than or equal to three ($j \leq 3$) and $k = n - j \geq 2$ Π's. Initially we assume that $j = 3$ and look for j variables which do not form a pi product. We choose D, V and ρ which cannot form a dimensionless group because only ρ contains mass and only V has units of time. Thus, the reduction j must equal three and the pi theorem guarantees for this problem that there will be exactly two independent dimensionless groups.

In the second part of the pi theorem, we combine D, V and ρ with one additional variable, in sequence, to find the two dimensionless pi groups. For Π_1, we select the force, F_T raised to the first power:

$$\Pi_1 = D^a V^b \rho^c F_T = (L)^a (LT^{-1})^b (ML^{-3})^c (MLT^{-2}) = M^0 L^0 T^0 \tag{3.2}$$

Equating the exponents for each fundamental dimension:

Length, L	$a + b - 3c - 1 = 0$
Mass, M	$c + 1 = 0$
Time, T	$b + 2 = 0$

Solving explicitly for a = -2, b = -2 and c = -1 yields the following expression for Π_1:

$$\Pi_1 = \frac{F_T}{\rho V^2 D^2} \tag{3.3}$$

and using μ as the additional variable instead of F_T, we obtain Π_2:

$$\Pi_2 = \frac{\rho VD}{\mu} = \mathrm{Re} \tag{3.4}$$

where Re is the Reynolds number. The pi theorem guarantees that the functional relationship between the two dimensionless groups can be expressed as:

$$\frac{F_T}{\rho V^2 D^2} = F\left(\frac{\rho VD}{\mu}\right) = F(\mathrm{Re}) \tag{3.5}$$

The main benefit of using the functional relationship shown above, equation (3.5), instead of the original expression, equation (3.1), is that we do not have to vary D, V, ρ or μ separately but only alter the dimensionless group, $\rho VD/\mu$. We can do this simply by changing the fluid velocity in, for example, a wind tunnel or water channel, without having to construct several different bluff bodies to alter D or use many different fluids to vary the fluid density and viscosity. Another important benefit of dimensional analysis is that it improves our thinking and planning for experiments, computations or theory. For example, it suggests dimensionless ways of writing an equation before we waste computer time trying to find solutions. It may also suggest which variables can be discarded, or groups them off to the side, where a few simple tests will show them to be unimportant. Finally, dimensional analysis often gives a great deal of insight into the form of the physical relationship we are trying to study.

Although dimensional analysis is based on a strong physical and mathematical background, considerable skill and experience is needed to apply it effectively. The pi theorem only tells us how many dimensionless groups are needed and gives general rules on how to construct them. It does not tell us which dependent variables should be initially selected to generate the groups. As a result, many different Π's can be found for the same problem and therefore the value of the results obtained by dimensional analysis ultimately depends on the experience and insight of the person applying it.

3.2.2 Derivation of Π's for compressor performance

The performance of a centrifugal compressor is often expressed in terms of the isentropic enthalpy increase across the unit, Δh_{12_s} the isentropic efficiency, η

and the hydraulic power requirement, P. The functional relationship for each of these performance variables can be written in terms of the unit operating parameters, physical properties and composition of the gas:

$$\Delta h_{12s} = f_1(D, N, Q_1, \rho_{o1}, \mu_{o1}, a_{o1}, y) \tag{3.6}$$

$$\eta = f_2(D, N, Q_1, \rho_{o1}, \mu_{o1}, a_{o1}, y) \tag{3.7}$$

$$P = f_3(D, N, Q_1, \rho_{o1}, \mu_{o1}, a_{o1}, y) \tag{3.8}$$

where D is the impeller diameter, N is the unit rotational speed, and Q_1 is the actual inlet flow rate, ρ_{o1} is the inlet gas density, μ_{o1} is the inlet gas viscosity and a_{o1} is the speed of sound all based on inlet stagnation conditions, and finally y is the mole fraction vector representing the gas composition. Applying the Buckingham pi theorem, we reduce the eight independent variables, containing the primary dimensions of mass, length, and time, down to five dimensionless groups. Although many different pi groups are possible depending on the selected combination of independent variables for the analysis, the following non-dimensional groups are the most commonly used by the compressor industry.

The dimensionless isentropic or "adiabatic" head coefficient, ψ:

$$\psi = \frac{\Delta h_{12s}}{N^2 D^2} = F_1\left(\frac{Q_1}{ND^3}, \frac{ND}{a_{o1}}, \frac{\rho_{o1} ND^2}{\mu_{o1}}, y\right) = F_1(\phi, \mathrm{M}, \mathrm{Re}, y) \tag{3.9}$$

Similarly, the functional dependence of isentropic efficiency, η may be written

as:

$$\eta = F_2\left(\frac{Q_1}{ND^3}, \frac{ND}{a_{o1}}, \frac{\rho_{o1}ND^2}{\mu_{o1}}, y\right) = F_2(\phi, \mathrm{M}, \mathrm{Re}, y) \tag{3.10}$$

and the power coefficient, Ω as:

$$\Omega = \frac{P}{\rho_{o1}N^3D^5} = F_3\left(\frac{Q_1}{ND^3}, \frac{ND}{a_{o1}}, \frac{\rho_{o1}ND^2}{\mu_{o1}}, y\right) = F_3(\phi, \mathrm{M}, \mathrm{Re}, y) \tag{3.11}$$

where $\phi = Q_1/ND^3$ is referred to as the flow coefficient, $\rho_{o1}ND^2/\mu_{o1}$ is a form of Reynolds number, Re for the unit based on the tangential velocity at the impeller tip and ND/a_{o1} is the blade tip Mach number, M based on the impeller tip velocity. A unique connection exists between the Ω and the remaining groups, (ψ, ϕ, η) because the required hydraulic power must equal $\rho_{o1}Q_1\Delta h_{12s}/\eta$. Thus, Ω can be computed simply from the following expression:

$$\Omega = \frac{\phi\psi}{\eta} \tag{3.12}$$

The other dimensionless relationships, i.e., F_1 and F_2, must be determined either experimentally or from a detailed theoretical approach (e.g., computational fluid dynamics).

3.2.3 Non-dimensional performance map

In this section, we have modified the dimensionless performance groups introduced in section 3.2.2 to incorporate units for dimensional variables. The inlet flow coefficient, ϕ is obtained by dividing the actual inlet flow rate, Q_1 by

the hypothetical flow from passing the impeller tip velocity, U_2 through the projected frontal area based on the impeller tip diameter, D:

$$\phi = \frac{Q_1}{U_2 A_2} = \frac{240 Q_1}{\pi^2 N D^3} \tag{3.13}$$

with units for Q_1 [m^3/s], N [rpm], and D [m]. The head coefficient, ψ is the isentropic head, Δh_{12s} normalized by the hypothetical dynamic head corresponding to U_2:

$$\psi = \frac{\Delta h_{12s}}{\frac{1}{2} U_2^2} = \frac{7.2 \times 10^6 \, \Delta h_{12s}}{\pi^2 N^2 D^2} \tag{3.14}$$

with units for Δh_{12s} [kJ/kg]. The compressor efficiency, η is already dimensionless and the power coefficient, Ω is calculated by substituting the known values of ϕ, ψ, and η into equation (3.12).

The non-dimensional parameters translate onto a single curve, even though the data may be taken at different compressor speeds. This simplified relationship is utilized to determine the Cheesman coefficients for the head and efficiency equations. These coefficients are subsequently used to generate the dimensional performance characteristics in section 2.2.4. To elucidate this relationship, assume we can approximate the function, $\psi = F_1(\phi)$, by a second-order polynomial:

$$\psi = F_1(\phi) = a_o + a_1 \phi + a_2 \phi^2 \tag{3.15}$$

Substituting equation (3.13) for ϕ and equation (3.14) for ψ, keeping the constant terms in the polynomial coefficients, the isentropic head, Δh_{12s} can be written in terms of the inlet flow rate, Q_1 and compressor speed, N as follows:

$$\frac{\Delta h_{12s}}{N^2D^2} = a_o + a_1\left(\frac{Q_1}{ND^3}\right) + a_2\left(\frac{Q_1}{ND^3}\right)^2$$

$$\therefore \Delta h_{12s} = N^2D^2\left\{a_o + a_1\left(\frac{Q_1}{ND^3}\right) + a_2\left(\frac{Q_1}{ND^3}\right)^2\right\} \tag{3.16}$$

Similar algebraic manipulation provides the relationship for efficiency, η:

$$\eta = b_o + b_1\left(\frac{Q_1}{ND^3}\right) + b_2\left(\frac{Q_1}{ND^3}\right)^2 \tag{3.17}$$

where a_o, a_1, a_2, b_o, b_1 and b_2 are the Cheesman coefficients which are determined by a least-squares fits of performance test data or digitized points from the manufacturer's wheel map.

3.3 Affinity Laws

For the same machine, D is constant so equations (3.14) and (3.15) can be expressed in the form:

$$Q \alpha N \tag{3.18}$$

$$H \alpha N^2 \tag{3.19}$$

A third equation, for power, can be added:

$$P \alpha N^3 \tag{3.20}$$

The constants of proportionality which are not shown are the flow, head and power coefficients, ϕ, ψ, Ω. These, common, forms of the non-dimensional equations can be used to estimate compressor parameters away from test data operating points, for example:

$$\frac{Q_1}{Q_2} = \frac{N_1}{N_2} \tag{3.21}$$

$$\frac{H_1}{H_2} = \frac{N_1^2}{N_2^2} \tag{3.22}$$

The effects of changing speed on flow rate or head can be estimated. Alternatively a speed can be estimated at which a desired flow rate or head can be achieved.

3.4 Steady State Compressor Modeling

The steady state model was implemented using the non-dimensional parameters described in the previous section, techniques similar to those described by Godse [45] and by implementing the thermodynamic equations described in section 2.2.

The model was to be capable of estimating compressor speed for any flow rate within the range of the performance characteristic, generating any isentropic head value for any set of compressor suction parameters within a pre-defined operating envelope.

The standard tests for a neural network would require six inputs. These were selected at random within the predefined operating range or calculated from compressor characteristics or thermodynamic relationships. :

- Suction pressure

- Suction temperature
- Molecular mass
- Differential pressure (to represent actual volume flow rate)
- Discharge pressure
- Discharge temperature

For extended testing the molecular mass was replaced by gas composition of either ten or twenty-one components.

An iterative procedure was followed to calculate discharge pressure and consequently discharge temperature. The following steps describe the procedure.

1. Select suction temperature, pressure and gas composition.
2. Calculate gas compressibility, in accordance with AGA8, at suction conditions using gas composition
3. Calculate isentropic exponent at suction conditions using molecular mass from a regression equation.
4. Select random dp, calculate flow rate
5. Select speed, calculate flow coefficient
6. Calculate isentropic head coefficient from quadratic regression equation
7. Calculate isentropic head from the isentropic head coefficient and compressor speed
8. Calculate discharge pressure and temperature.
9. Calculate compressibility of discharge gas then the average of suction and discharge compressibility factors.
10. Repeat steps 8 and 9 until successive value of discharge pressure agree to within a convergence criterion.

3.4.1 Steady State Model Basic Data

The steady state model is an actual compressor performance characteristic at one, base, set of suction conditions. The performance characteristic covers eight operating speeds generating isentropic head of between 43 – 136 kJ/kg at flow rates of between 725 – 1876 acm/h. The suction conditions are 62 bar pressure , 33.3 °C temperature and 20.1 kg/kg mol molecular mass. The performance characteristics data points are shown in Appendix B, Table B1. A graph of the performance characteristic is shown in Figure 3.1.

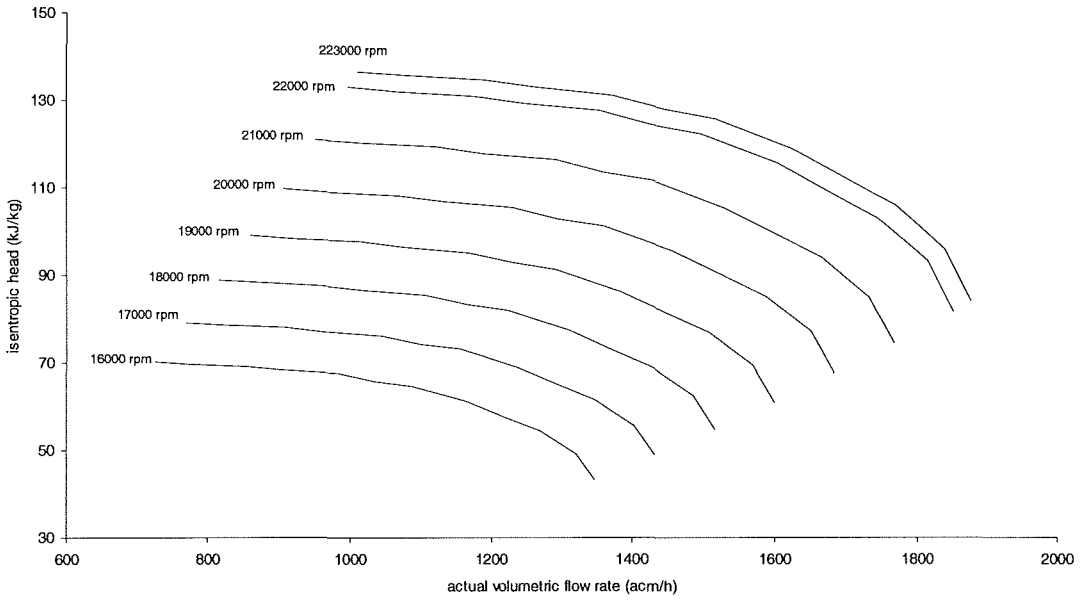

Figure 3.1 – Base Head Map

Isentropic head is plotted against actual volume flow rate for eight lines of constant speed.

The data was converted to a universal speed curve representing isentropic head coefficient, ψ, as a function of flow coefficient, ϕ, by the 4^{th} order polynomial equation:

$$\Psi = 6.13537 \times 10^{-7} - 0.057219519\phi + 2726.218977\phi^2 - 1980.23 \times 10^9 \phi^4 \qquad (3.23)$$

The higher order regression equation yielded slightly better standard error in ψ over a quadratic regression equation of the form shown in equation (3.15).

3.4.2 Selecting Suction Temperature and Suction Pressure Values

Where "base" suction temperature and pressure values were specified, slight random variation was introduced to eliminate the effect described in section 5.3.3 with respect to network architecture. The nominal values could vary between: $P_{suc} = 6200 \pm 0.005$ kPa and $T_{suc} = 33.3 \pm 0.00005$ °C.

Where specified, random values of suction pressure and suction temperature were selected from within the pre-defined ranges as shown in Table 3.2

Parameter	Maximum	Minimum
Ps	7000 kPa	5000 kPa
Ts	40 °C	25 °C

Table 3.2: Suction Parameter ranges

3.4.3 Selecting Molecular Mass and Gas Composition

Gas composition was required to calculate thermodynamic properties for calculating discharge pressure from isentropic head, hence molecular mass was calculated from gas composition. Three gas compositions were chosen to represent nominal molecular masses of 18, 20 and 22 kg/kg-mole as shown in Table 3.3.

Component	Nominal Mol. Mass. 18 kg/kg-mole	Nominal Mol. Mass. 20 kg/kg-mole	Nominal Mol. Mass. 22 kg/kg-mole
Methane	93	80.71	70
Ethane	2	11.4	20
Propane	0.8	3	4
i-Butane	0.21	0.4	0.81
n-Butane	0.21	0.6	0.81
i-Pentane	0.15	0.2	0.42
n-Pentane	0.15	0.2	0.42
n-Hexane	0.1	0.1	0.15
Nitrogen	0.39	0.39	0.39
Carbon Dioxide	3	3	3

Table 3.3: Gas Composition Corresponding to Nominal Molecular Mass

Where the "base" value of molecular mass was specified a slight random variation was introduced to eliminate the effect described in section 5.3.3 with respect to network architecture. The nominal value could vary between: MW = 20.086 ± 0.000005 kg/kg-mole.

Gas molecular mass was derived from a gas composition to facilitate gas compressibility computation using the AGA 8 equation of state [20]. For this purpose the following gas components, which could typically be identified by an online gas chromatograph, were chosen at random from the ranges shown in Table 3.4:

Component	Maximum Mol %	Minimum Mol %
Methane	93	70
Ethane	20	2
Propane	3	0.8
i-Butane	0.27	0.21
n-Butane	0.27	0.21
i-Pentane	0.42	0.15
n-Pentane	0.42	0.15
n-Hexane	0.15	0.1
Nitrogen	= 0.39	
Carbon Dioxide	= 3	

Table 3.4: "Short" Gas Analysis Component

Additional components: n-Heptane; n-Octane; n-Nonane; n-Decane; hydrogen; water vapour; hydrogen sulphide; carbon monoxide; argon; helium; oxygen; as identified in the AGA8 equation of state, were taken to be 0 mol %. The composition was selected for use when the sum of all components was within 100 ± 1 mol % and the molecular mass was within 20 ± 2 kg/kg-mol. The components were then re-normalised to total 100 mole % prior to input to the neural network.

3.4.4 Selecting a Differential Pressure Input and Inlet Flow Rate

The compressor actual inlet volume flow range was mapped onto a differential pressure range of 22.5 – 150 mbar corresponding to a mass flow range of 41665 kg/hr to 107845 kg/hr at "base" suction conditions. A differential pressure was randomly selected using:

$$dp = 127.5 \times rnd + 22.5 \tag{3.24}$$

where rnd is a random number between 0 and 1. This value of dp was then converted to actual inlet volume flow rate at the prevailing suction conditions using:

$$qv = \frac{dp}{150} \frac{107845}{\rho_{suc}} \tag{3.25}$$

where ρ_{suc} is the density of the selected gas composition at selected suction pressure and temperature, calculated using AGA8.

3.4.5 Calculating Discharge Pressure and Temperature

Compressor test speed was randomly selected using:

$$N = 6300 \times rnd + 16000 \tag{3.26}$$

Using the actual inlet volume flow rate, qv, the flow coefficient, ϕ, was calculated using:

$$\phi = qv/3600/N \tag{3.27}$$

Compressor isentropic head coefficient was calculated using equation (3.23), hence isentropic head, H, could be estimated using:

$$H = \Psi \times N^2 \tag{3.28}$$

Isentropic exponent was calculated from a regression equation using suction conditions as follows:

$$\kappa = 1.74166259 + 0.000136889 P_S - 0.003673681 T_S - 0.015897153 MW \qquad (3.29)$$

The regression equation data is based on isentropic exponent data for pure methane calculated using [23].

At this stage discharge pressure can then be calculated using equation 2.6 and discharge temperature can be calculated using equation 2.8. Once calculated compressibility at discharge conditions can be calculated. Iteration around the following loop $P_d \rightarrow T_d \rightarrow Z_d \rightarrow Z_{ave} \rightarrow P_d$ until P_d converges to within 0.02 %, yields Pd and Td.

3.4.6 Simulating Instrument Noise

These instrument noise levels were selected to represent typical instrument uncertainties as follows: ± 0.25% of span for pressure transmitters; ± 0.5% of span for temperature transmitters; ± 0.1% span for differential pressure transmitters; ± 0.25% of reading for molecular mass and ± 0.25% of reading for speed transmitters.

Randomly distributed instrument uncertainty levels were generated with a mean of 0% and standard deviation corresponding to 95% or 2σ confidence levels corresponding to the uncertainty ranges stated in the previous paragraph. The uncertainty levels were then applied to the instrument range or spot reading, as appropriate, to create a noise component. The noise component was then added to the nominal input value to give the "noisy" input. The instrument ranges shown in Table 3.5 were used to calculate noise components.

Signal	Instrument range	Noise component
Suction pressure	5000 – 7000 kPa	% of 2000 kPa
Suction temperature	0 – 200 °C	% of 200°C
Differential pressure	0 –150 mbar	% of 150 mbar
Molecular mass	18 - 22 kg/kg-mole	% of reading
Discharge pressure	7500 – 20000 kPa	% of 12500 kPa
Discharge temperature	0 – 200 °C	% of 200°C
Compressor speed	16000 – 22300	% of reading

Table 3.5: Instrument Noise / Uncertainty Component

3.4.7 Generating Extrapolated Data

The individual data-points in the limited training set are shown against the complete performance characteristics in Figure 3.2.

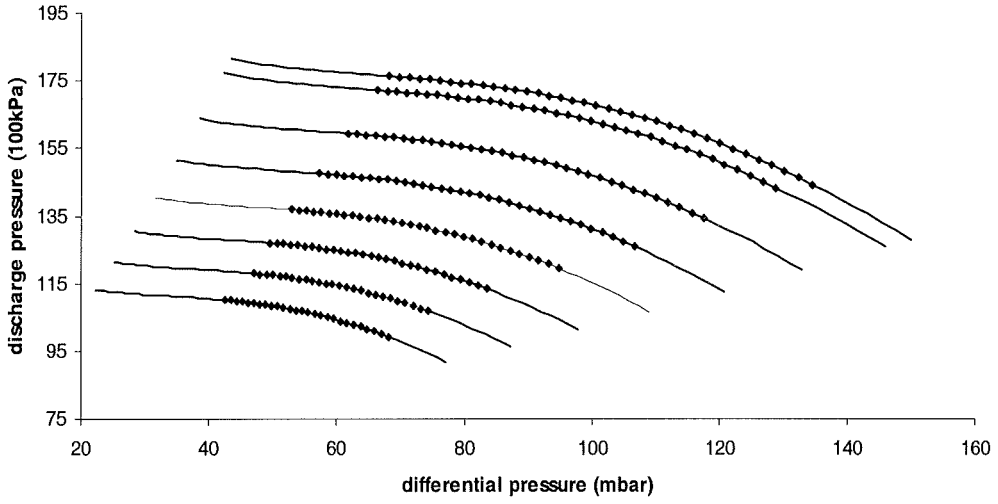

Figure 3.2: Extrapolated Data Set

These points were generated by selecting arbitrary limits on flow coefficient for each speed as shown in Table 3.6.

Speed (rpm)	Flow coefficient, ϕ, range
22300	$0.0000157704 < \phi < 0.0000221376$
22000	$0.0000157704 < \phi < 0.0000219607$
21000	$0.0000159743 < \phi < 0.0000219607$
20000	$0.0000161242 < \phi < 0.0000219607$
19000	$0.000016301 < \phi < 0.0000217839$
18000	$0.0000166548 < \phi < 0.000021607$
17000	$0.0000171854 < \phi < 0.000021607$
16000	$0.0000173622 < \phi < 0.0000201921$

Table 3.6: Flow coefficient limits for extrapolated data

The restricted training data points were then generated as described by the procedure in section 3.4.5, testing differential pressure and speed to ensure the limits imposed on ϕ were met.

3.4.8 Normalised Compressor Performance Data

Some limited testing was undertaken training neural networks using flow and isentropic head coefficients, $\phi$and ψ, respectively. This data set was taken directly from the basic performance characteristics data normalised against compressor speed as described by equations (3.13) and (3.14).

3.5 Discussion

The methods described, in this chapter, can be applied across the range of operating data bounded by the flow coefficient, ϕ. Isentropic head coefficient as a function of ϕ, $\psi(\phi)$, yields a continuous representation of isentropic, or adiabatic, head. Coupled with the thermodynamic relationships for gas compressibility and isentropic exponent, compressor characteristics can be calculated for any combination of gas properties and compressor operating conditions.

3.6 Conclusions

The steady sate model uses industry standard techniques. Rotodynamic and thermodynamic relationships are well defined in many reference texts. The flexibility of the head and flow coefficient techniques are a very powerful modelling tool for researching compressor performance control strategies.

4 Neural Networks - Function Approximation and Data Clustering

4.1 Introduction

This chapter begins with a review of artificial neuron models. Two artificial neuron types are identified as most significant to the research. For these types, neural network architectures are examined and training of these networks is described. A vector analogy of training is used to illustrate the function of the neural networks. The review is developed to summarise the types of neural networks commonly in use and their likely application.

4.2 Overview of Artificial Neuron Models

Gurney [48] summarises the most common of neuron types. These are as shown in Table 4.1:

Neuron Type	Activation function	Application
Threshold Logic Unit (TLU),semi-linear nodes	$w \bullet x$	Classification, function interpolation
Radial basis function (RBF)	$\|w - x\|$	Classification, function interpolation
Sigma-pi units	$\sum_k w_k \prod_{i \in I_k} x_i$	Biological, presynaptic inhibition
Digital node	$S_\mu : \mu = x$	Boolean function networks, RAM applications

Table 4.1 - Overview of Neuron Types

The TLU neuron model allows output representation of its input pattern as a binary state, (classification), or as an analogue between the states, (function interpolation). The activation level of the neuron is given by the scalar product of the input vector, **x**, and the weight vector, **w**. In the case of the semi-linear nodes, neurons with analogue output, the output is proportional to the scalar product of the vectors.

A Radial Basis Function, (RBF), neuron represents its input pattern as the Euclidian distance between the input pattern vector, **x**, and an exemplar vector, **w** (the centre of the neuron), in vector space. Classification comes about through association of the input vector with a specific RBF neuron. Representing the input pattern in terms of a distance from a known point is equivalent to function approximation. In vector space modelled by n RBF neurons the input vector pattern is represented as an interpolation between the n - neurons.

The sigma-pi unit is a model of an *axo-axonic* synapse i.e. it is a specific biological model in which multiple axon terminals impinge upon each other at the dendritic synapse. The activation of one axon can inhibit the excitatory effect of another axon on the synapse potential. In biological terms, this action models *pre-synaptic inhibition*, believed to be of significance in motion detection. Picton [49] describes how a sigma-pi network can be trained, using Hebbian learning techniques, to determine Rademacher-Walsh transform coefficients for an n-input logic circuit. This implies that a single sigma-pi neuron with 2^n weights could be trained to represent a logic function. The sigma-pi unit belongs to a category of higher order networks where the network function is determined as a weighted sum of products of inputs.

Digital nodes are used in Boolean neural networks. These have associative memory applications where the neural network function is a translation between an input pattern (RAM address), and the Boolean pattern stored in the address location.

Further examination of the characteristics of the principle neurons, a TLU neuron and its semi-linear variants, and an RBF neuron, is described in the

following sections. No further consideration is given to the sigma-pi or digital node neurons as these have been developed for specialised biological or computer memory fields of application, not approached in this research.

4.2.1 Threshold Logic Unit

The artificial logic unit, TLU, proposed by McCulloch & Pitts 1943 [50], is the basis of artificial neurons generally found in most types of neural networks. The TLU mimics the biological neuron in certain key aspects.

TLUs consist of an array of inputs passing through individual weighting blocks arriving at a common summation block. The summation block outputs are then passed to a threshold "switch", which changes the output state of the artificial neuron. These components are shown in Figure 4.1.

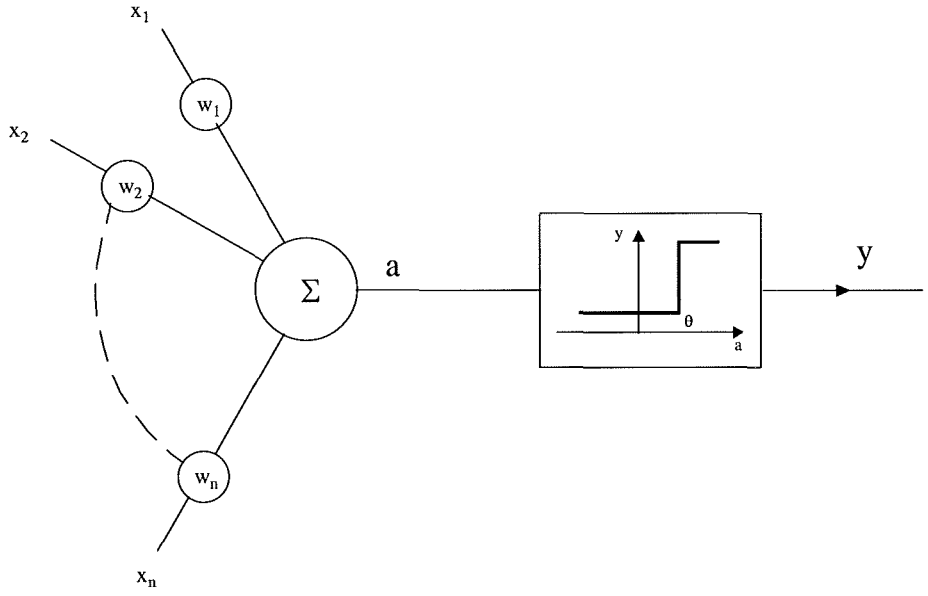

Figure 4-1: Schematic Diagram of TLU

The input array, $x_1 \ldots x_n$, is analogous to synapses, where:

- Weighting block simulates post-synaptic potential, depicted as w in Figure 4.1.
- A summator, Σ, accumulates neuron activation level.
- A threshold function controls the *firing* level of the neuron, simulating the action of the axon-hillock membrane potential.

If the neuron activation level exceeds the threshold the output from the neuron is one. If the activation level does not reach the threshold the output is zero. Hence the two state output associated with the threshold level is reflected in the name of the neuron, *Threshold Logic Unit*, (TLU). The activation level of the neuron, a, is written as:

$$a = \sum_{i=1}^{n} w_i x_i \tag{4.1}$$

where a is the activation level of the Neuron,

x is an input to the Neuron.

w is the weighting given to an input.

The output is determined by a threshold level, θ, such that:

$$y = \begin{cases} 1 \; if \; a \geq \theta \\ 0 \; if \; a < \theta \end{cases} \tag{4. 2}$$

Hence the TLU can distinguish between two classes of input combinations, indicating a "1" or a "0" class. The TLU is a non-linear device, the output does not follow the input and may only assume a Boolean state. It is time-independent and is assumed to react instantaneously to input changes.

4.2.1.1 Geometric Interpretation of TLU Classification

Gurney [48] describes classification using a simple, 2 input TLU. A vector analogy with a simple model identifies the properties of a higher order, n-dimensional classification function. For a two input TLU the neuron activation level is:

$$a = w_1 x_1 + w_2 x_2 \tag{4.3}$$

re-arranging equation 4.3 with a = θ gives:

$$x_2 = -\left(\frac{w_1}{w_2}\right) x_1 + \left(\frac{\theta}{w_2}\right) \tag{4.4}$$

This is in the form of the equation for a straight line where $\frac{\theta}{w_2}$ is the intercept on the x_2 axis. Figure 4.2 shows separation of a vector space implementing an AND function. For a TLU to classify data the data elements must be *linearly separable.*

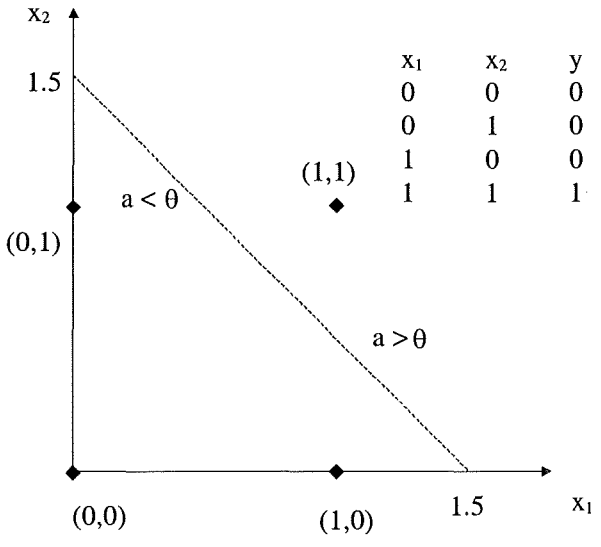

Figure 4.2 - Linear Separation of Vector Space

θ represents a *decision-plane,* where two possible output classifications separate the vector space of x_1 and x_2 into two regions, $a > \theta$ and $a < \theta$. The vector analogy considers the weighted input, x_w , which can be represented as a vector projection of the input vector, **x** on the weighting vector **w,** as shown in Figure 4.3.

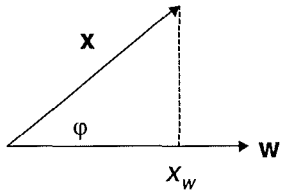

Figure 4.3 - Input and weight vectors

x_w is given by the expression:

$$x_w = \frac{\|\mathbf{x}\|\|\mathbf{w}\| cos\,\varphi}{\|\mathbf{w}\|} \tag{4.5}$$

This can be written in terms of the vector *dot,* or inner, product as follows:

$$x_w = \frac{\mathbf{x} \cdot \mathbf{w}}{\|\mathbf{w}\|} \tag{4.6}$$

For a two dimensional vector, the inner product is given by:

$$\mathbf{x} \cdot \mathbf{w} = x_1 w_1 + x_2 w_2 \tag{4.7}$$

where x_1, x_2 and w_1, w_2 are the Cartesian space co-ordinates for vectors **x** and **w**, respectively.

This compares directly with the activation level expression for the neuron in equation 4.3. Setting the activation level to the threshold gives:

$$\boldsymbol{x} \cdot w = \theta \tag{4.8}$$

and

$$x_w = \frac{\theta}{\|w\|} \tag{4.9}$$

For two, *n*-dimensional vectors, **x** and **w**, activation level is written as:

$$\boldsymbol{x} \cdot \boldsymbol{w} = \sum_{i=1}^{n} x_i w_i \tag{4.10}$$

The output condition of the TLU is determined by the threshold, θ. If $\boldsymbol{x} \bullet \boldsymbol{w} \geq \theta$, the activation threshold has been reached and the artificial Neuron output is '1', similarly, if $\boldsymbol{x} \bullet \boldsymbol{w} < \theta$, the threshold has not been reached and the output is '0'. The analogy demonstrates the extension of the TLU function to n-dimensional vector space. A 3-input TLU acquires a *decision-surface*. Greater than three inputs, n-dimensions, is a *decision hyper-plane*.

Where a pattern space consists of non-linearly separable classes, a number of decision planes may be fitted to a pattern space effectively to compartmentalise the pattern space into classes. This requires more than one *layer* of Neurons, as discussed in section 4.3. Use of several layers of neurons gave rise to the "connectionist" paradigm where successive layers of neurons are interconnected to perform more complex classification tasks. As the pattern space becomes more complex two problems are exacerbated:

- Inspection of the pattern space becomes increasingly difficult hence *a priori* classification knowledge becomes less available
- assigning weight values becomes increasingly complex due to the increasingly complex network structure.

The linearly separable limitation of a single layer perceptron (TLU), and perceived difficulties in assigning weight values within highly connected, multi-layered perceptrons (TLUs) led to conjecture that cast doubt over the computational capabilities of neural networks until the mid 1980s, [48].

It was the formulation of an error back-propagation training algorithm that overcame the problems, weight assignment and *a priori* knowledge of data sets, with multi-layered perceptrons (MLP).

4.2.2 Semi-linear nodes

Semi-linear modes are distinguished from the TLU by their output function. Whilst the TLU has a step function, the semi-linear nodes have a continuous, differentiable output function. Two most common functions in use allow representation of uni-polar or bi-polar output functions. These are from the family of logistic functions, the sigmoid and hyperbolic tangent functions respectively.

The linear range of the functions, prior to any biasing and scaling, is not much greater than –1 to 1 so the input to the TLU is normalised against the maximum and minimum of the input range. The normalised input is usually between 0 to 1. Nominal or categorical data are usually input as "1 of n" .

Both functions exhibit sigmoidal non-linearity, however the tanh function can have a negative output that can be thought of as a negative, inhibitory output as opposed to the positive excitatory output. With a sigmoidal output function the TLU can provide an analogue classification, or representation, of input combinations. Caudill & Butler [51] describe the development of *semi-linear* neuron models e.g. adaline, padaline, although the roots of these neurons are to be found in the basic TLU.

4.2.2.1 Logistic Sigmoid Function

The output from a TLU can be modified by changing the output step function of the TLU to a *softer* function, such as a logistic sigmoid function:

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-(a-\theta)/\rho}} \tag{4.11}$$

Figure 4.4 shows a plot of the sigmoid functions. Each exhibits a transition that gives an approximately linear function over a limited range of inputs. This transition describes the output of a *semi-linear* artificial neuron. The parameter ρ is used to change the slope of transition. Large values of ρ lengthen the transition region, implying large changes in activation level are required to change neuron output i.e. low sensitivity. Small values make the *linear* transition shorter, implying high sensitivity to changes in input.

θ, is an arbitrary threshold level analogous to the neuron activation level and can be used to bias neuron output.

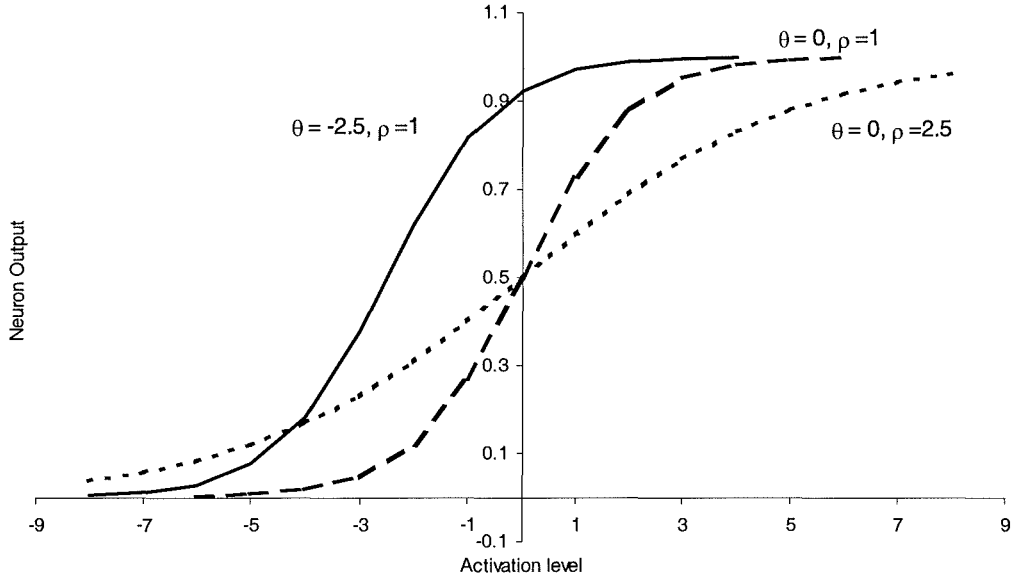

Figure 4.4 - Logistic Functions

4.2.2.2 Hyperbolic Tangent Function

The logistic function assigns a non-symmetrical function to the neuron output. A hyperbolic function can be used for a bi-polar output as shown in Figure 4.5. Haykin [52] describes the hyperbolic tangent (tanh) function as a biased, re-scaled logistic function that has an equation of the form:

$$y = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.12}$$

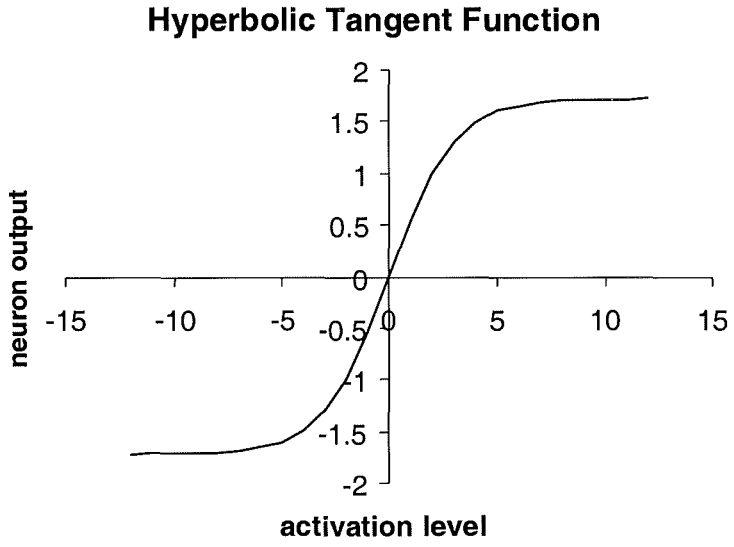

Figure 4.5: Logistic Functions

4.2.2.3 Perceptron - Basic Algorithmic Training

Pre-processing of TLU inputs was identified by Rosenblatt in 1962 [53], and the resulting development of the TLU neuron model was called a *perceptron*. Rosenblatt's application of pre-processing was in connection with image recognition and the pre-processing, or "association", units identified elements in a grid prior to classification of the image. The difference between the TLU and perceptron is:

- The perceptron has pre-processing or scaling of inputs
- The perceptron has an analogue or semi-linear output function in contrast to the TLU logical output

A perceptron performs a classification function through the activation of a threshold level, which is triggered through a combination of input weighting and threshold setting. Thus, for a given input vector, the threshold and input weighting can be adjusted to cause the artificial neuron output to mimic a classification pattern. The perceptron is trained to give a response, $\mathbf{t}_k$, with an input $\mathbf{x}_k$, for every k, using a training set $\mathbf{t}$. The required response, $\mathbf{t}_k$ is referred to as the training target. This type of training is categorised as Supervised Learning.

Adjusting the weight vector is achieved through the addition of a correcting vector represented as a function of the input vector, as illustrated in Figure 4.6.

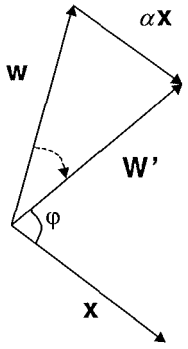

Figure 4.6 - Input and weight vectors

The new weight vector, $\mathbf{w}'$ is given by:

$$\mathbf{w}' = \mathbf{w} + \alpha\mathbf{x} \tag{4.13}$$

If an output, y, of unity is obtained for a training target, t, of 0 then the weighting strength must be reduced, that is, $\mathbf{x}_w$ is reduced to below the threshold level, θ. This implies that $\cos\varphi$ is reduced, that is, the input vector, $\mathbf{x}$, is

to be rotated away from the weight vector, **w**. Hence, the angle between x and w tends towards 90°, that is $\alpha\mathbf{x}$ is negative. The converse case applies, t=1, y=0 and $\mathbf{x}_w$ must be increased to above θ, therefore $\alpha\mathbf{x}$ is positive.

The general training rule can be written in terms of the training target and the untrained output:

$$\mathbf{w}' = \mathbf{w} + \alpha(t - y)\mathbf{x} \tag{4.14}$$

or, in terms of the change in weight:

$$\Delta\mathbf{w} = \alpha(t - y)\mathbf{x} \tag{4.15}$$

or, in terms of the vector components:

$$\Delta w_i = \alpha(t - y)x_i \tag{4.16}$$

The parameter α is known as the *learning rate*. A larger learning rate will speed up training over a smaller learning rate, but may lead to training *instability*. A large learning rate may "overstep" the training target causing oscillation of the learning algorithm about the target. Adjustment of the weight vector, w, as described is known as the *Perceptron training rule*, as it was on this variant of TLU that this type of training was first described. The application of vector weight adjustments can be described with:

```
repeat
    for each training vector pair (x,t)
    evaluate the output y when x is input to the TLU
        if y≠t then
        form a new weight vector w'
        else
        do nothing
```

```
        end if
    end for
until y=t for all vectors
```

4.2.2.4 ADALINE and Error Based Learning

ADAptive LINear (ADALINE) elements are based on a TLU with a modified bipolar output, that is –1 to +1, as opposed to 0 and +1. This type of Neuron was trained using the delta-rule described, in 1960, by Widrow and Hoff [54].

The delta-rule determines the minimum of a function, in this case, the *error* between the actual output from the artificial Neuron and the training target. The minimum of the error function is determined using the *Gradient Descent* method. This method relies on the value of the derivative of a function approaching zero as the function approaches a maximum or a minimum. For a function $y = y(x)$, it can be shown that a small change in y, δy, for a small change in x, is approximated by:

$$\Delta y \approx \frac{dy}{dx} \times \Delta x \tag{4.17}$$

If y(x) is differentiable and dy/dx can be calculated, then Δx can be expressed:

$$\Delta x = -\alpha \frac{dy}{dx} \tag{4.18}$$

For α sufficiently small to ensure $\Delta y = \delta y$, and substituting equation 4.18 into equation 4.17 gives:

$$\delta y \approx -\alpha \left(\frac{dy}{dx} \right)^2 \tag{4.19}$$

Equation 4.19 can be solved by iteration until $\delta y = 0$, at which point the value of x, x_0, defines the minimum of the function, $y = y(x)$.

Similarly, for a function of more than one variable, partial derivatives can be used and the expression for Δx can be written as:

$$\Delta x_i = -\alpha \frac{\partial y}{\partial x_i} \tag{4.20}$$

The training error of a neuron can be described as a function of the training targets and actual outputs determined by the weighting vector. The minimum of the error function, and so the optimal weight vector, can be determined using the gradient descent method. An expression for the optimal weight vector components is written as:

$$\Delta \mathrm{w_i} = -\alpha \frac{\partial \mathrm{E}}{\partial \mathrm{w_i}} \tag{4.21}$$

The error function, E, is defined as the mean of individual errors, e^p, for p training patterns:

$$E = \frac{1}{N} \sum_{p=1}^{N} e^p \tag{4.22}$$

where N is the total number of patterns in the training set.
To avoid cancelling of a positive and negative error, e^p, is defined as:

$$e^p = \frac{1}{2}\left(t^p - y^p\right)^2 \tag{4.23}$$

Depending on the threshold function, for example, Boolean or semi-linear, discontinuity may prevent convergence of the gradient descent. To allow convergence for the boolean function, the output y can be replaced by the activation level, a. Thus:

$$e^p = \frac{1}{2}\left(t^p - a^p\right)^2 \tag{4.24}$$

The true minimum gradient, or error, $\delta E/\delta w_i$, can be approximated by the error for one training pattern, p:

$$\frac{\delta e^p}{\delta w_i} = -\left(t^p - a^p\right)x_i^p \tag{4.25}$$

where t is the p^{th} training pattern
a is the p^{th} activation level
x_i^p is the p^{th} input to the i^{th} neuron

In terms of the optimal weight vector, the true gradient approximation is:

$$\Delta w_i = \alpha\left(t^p - a^p\right)x_i^p \tag{4.26}$$

For a semi-linear unit, an additional term is included for the rate of change of activation level, $d\sigma(a)/da = \sigma'(a)$. This term represents the sensitivity of the output with respect to activity level. For a semi-linear unit:

$$\Delta w_i = \alpha\sigma'(a)\left(t^p - y^p\right)x_i^p \tag{4.27}$$

The delta-rule in this form is applicable to a single neuron only. For a single

layer net, where there are several neurons, the *i*th weight and *j*th neuron, influences the training error as:

$$\Delta w_{ji} = \alpha\sigma'(a_j)(t_j^p - y_j^p)x_{ji}^p \tag{4.28}$$

4.2.3 Non-Linearly Separable Classification

Hinde et al [55] describes a method of extending vector space whereby an array of linear neurons, PADALINES, can perform non-linear classification. Identifying the additional vector combinations is similar to specifying coefficients in polynomial regression models. This is a "variation on the theme" of the semi-linear nodes and has the same drawbacks of a prior knowledge for the vector space and weight assignment in complex vector space.

4.2.3.1 PADALINE

A Polynomial ADaptive LInear NEuron (PADALINE) allows for the implementation of a non-linear decision plane. As shown in Figure 4.7, a vector space $\mathbf{F}_0$, $\mathbf{F}_1$ and $\mathbf{F}_2$ might be separable, as a circle with centre (C_1, C_2) and radius R in the $\mathbf{F}_1\mathbf{F}_2$ plane.

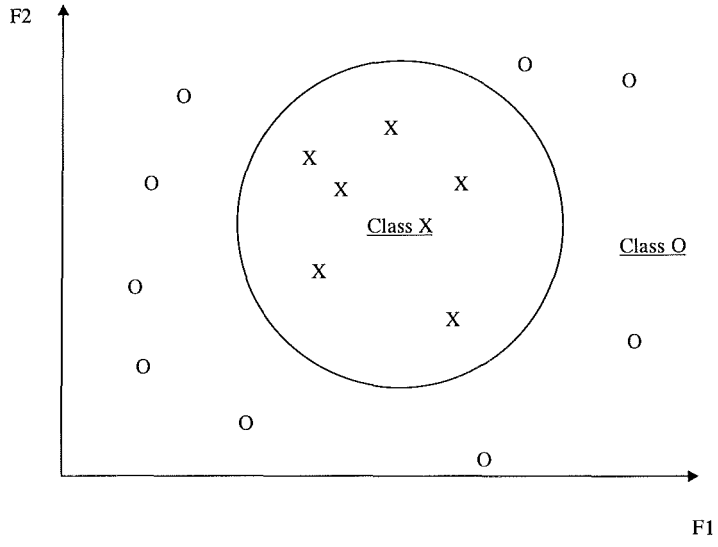

Figure 4.7 – Non-linearly Separable Classes

The equation of the circle is:

$$(F_1 - C_1)^2 + (F_2 - C_2)^2 = R^2 \tag{4.29}$$

The patterns can be grouped as follows:

$$X \text{ if } R^2 + 2C_1F_1 + 2C_2F_2 - F_1^2 - F_2^2 - C_1^2 - C_2^2 > 0 \tag{4.30}$$

$$O \text{ if } R^2 + 2C_1F_1 + 2C_2F_2 - F_1^2 - F_2^2 - C_1^2 - C_2^2 < 0 \tag{4.31}$$

If F_1^2 is substituted by vector $\mathbf{F}_3$, and F_2^2 by vector $\mathbf{F}_4$, then the coefficients of each vector, or neural network weights, can be written as follows:

$$w_0 = R^2 - C_1^2 - C_2^2 \tag{4.32}$$

$$w_1 = 2C_1 \tag{4.33}$$

$$w_2 = 2C_2 \tag{4.34}$$

$$w_3 = -1 \tag{4.35}$$

$$w_4 = -1 \tag{4.36}$$

Substituting with $\mathbf{F}_3$ and $\mathbf{F}_4$ effectively extends the vector space but requires accurate *a priori* knowledge of the vectors space.

4.2.3.2 Radial Basis Function (RBF) Neurons

Whilst the TLU based neurons use *hyper-planes* to partition vector space the RBF neuron use *hyper-spheres.* Associated with each hyper-sphere is an RBF neuron with an assigned centre of unit radius. An input vector pattern is compared with the RBF centre vector. The Euclidian distance is calculated as the deviation of the input from the centre. When the input vector lies on the surface of the hyper-sphere the output from the unit is 1. If the input vector is exactly on the RBF centre then its output is zero. The surface of the RBF is determined as a

Gaussian function. The activation surface of an RBF neuron is shown in Figure 4.8.

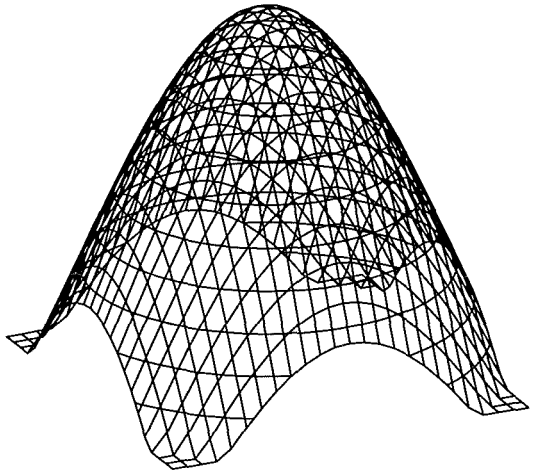

Figure 4.8 – Gaussian Activation Function

Haykin [52] gives the output of a normalised RBF neuron as:

$$G\left(\|x-t\|^2\right)=exp\left(-\frac{1}{d^2}\|x-t\|^2\right) \tag{4.37}$$

where x is the input vector

t is the RBF centre vector

d is the deviation of the RBF

This can be likened to the standard form of a Gaussian distribution:

$$y=exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{4.38}$$

where x is the input data

σ is the standard deviation of the input data

Comparing equations 4.37 and 4.38 the deviation, d, of the RBF affects the gradient of the slope of the Gaussian distribution. The $\|x-t\|$ term can be thought of as the activation level of the neuron as shown, for an n component space vector, in equation 4.39:

$$a = \sqrt{\sum_{i=1}^{n}(x_i - t_i)^2} \tag{4.39}$$

The function of the threshold in the RBF is different to that of the threshold in the semi-linear nodes. The parameter, d, is chosen to provide a smoothness factor when interpolating between the RBF nodes in a network. The threshold is also dependent on the number of RBF units representing a vector space. If there is a large distance between RBF centres then the slopes of the RBF functions must be sufficiently long to allow the outputs of the RBF to "mesh". If the slope is not sufficiently long to mesh RBF output surfaces, "holes" will appear in the vector space representation. The effects of changing the RBF threshold are shown in Figure 4.9.

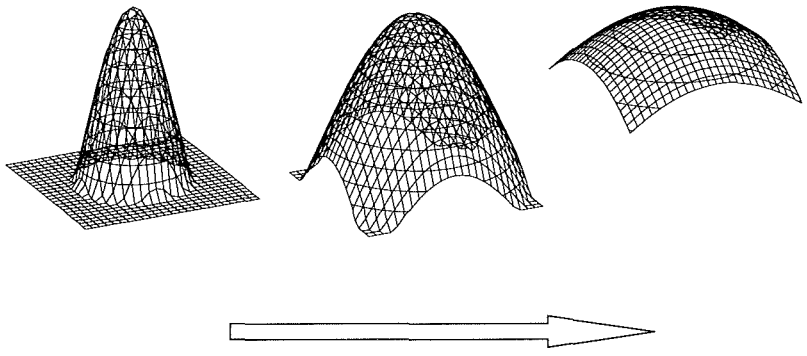

Figure 4.9 – Changing RBF Thresholds

A vector space can be "filled" with RBF units, each representing a data cluster in the vector space. Figure 4.10 depicts how two RBF neurons, of fixed centre, might be made to mesh in vector space. Meshing, may increase interpolation error but fitting additional more RBFs neurons could overcome this.

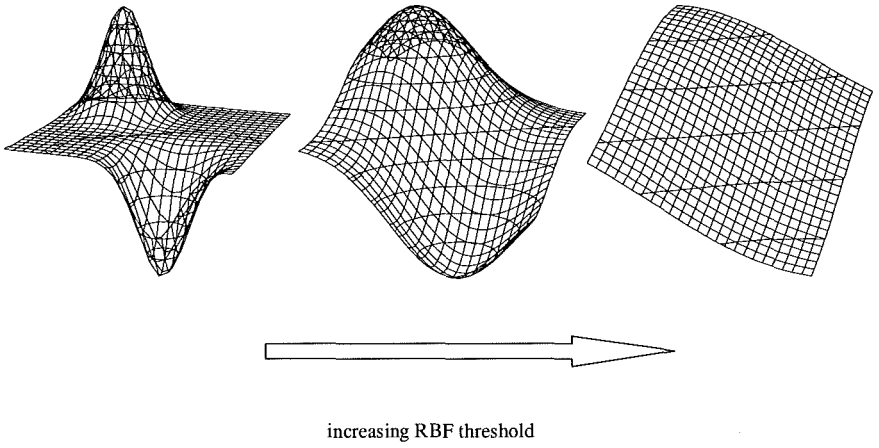

Figure 4.10 – RBF Meshing by Changing Thresholds

In an RBF neural network a function is mapped interpolating between RBF nodes. For this reason it is likely that extrapolating beyond boundary RBF nodes will lead to errors.

4.3 Overview of Network Architecture classification

Summarised from Gurney [48], Table 4.2 shows network architecture types.

Network Architecture	Tasks
Principally feed forward	• Classification • Function interpolation
Principally Re-current	• Associative memory
Competitive	• Cluster template formation • Identifying topological relationships

Table 4.2- Summary of Neural Network Architecture Types

Feed forward networks have "signal" paths that do not loop back on themselves, i.e. there is no feedback signal within the network. The back propagation of error, where this method of training is used, occurs in an algorithm external to the neural network. This type of network requires supervised training. Supervised training involves training a neural network with the characteristics to be represented. Training data is presented to the neural network. An error is derived by comparing the corresponding output of the network with the training data value, giving rise to an error value. The weights of the neural network are manipulated such that this error is minimised. Iterative training steps are carried out until the mean error for the training set is minimised. Supervised training enables a network to "learn" the underlying relationships in training sets. If the training data presented represents a function, then the network is a *function approximator*. Similarly if the training set represents a decision plane then the network is a *classifier*.

Re-current networks have feed back paths within the structure of the neural network. There is a signal path between the output, or intermediate layers,

neurons and input neurons. Inclusion of feedback paths simplify network architecture and a recursive network can be thought of as an equivalent circuit of a corresponding feed forward network. The network reacts to changes in input training patterns and changes in internal state due to feedback. This is similar to the way in which an asynchronous logic circuit reacts to changes in input and internal state. Neurons, in the network, change state until they reach a steady state, for the presented input pattern, is reached. This type of network also undergoes supervised training. The main application of this network is in identifying patterns in noisy data and associating that pattern with a vector such as a memory address. This is known as associative memory recall.

Competitive networks do not undergo supervised training in that no training data is presented to the network. Rather, the network "organises" its neurons into clusters in pattern space to encode the data set it is presented with. Clusters might then be assigned "classes" egg in alphanumeric pattern recognition, where a class is an alphanumeric character. Therefore if a pattern contains a particular set of clusters it could be assigned an alphanumeric character. In supervised training each neuron has its weights adapted to reflect the training input patterns. In competitive learning only the neuron that best represents the input pattern has its weights adjusted. Hence the notion of competition is that the winning neuron receives modification in a "winner–take-all" scenario. There are variations on this whereby positive or negative increments to all weights might be implemented according to the degree to which each neuron represents the input pattern. This notion of positive or negative reinforcement of weights is analogous to Hebbian learning, [51].

4.3.1 Feed forward Neural Networks

Architecture, typical of a feed forward neural network, is shown in Figure 4.11, Typical Feed Forward Neural Network. There is no standard method of depicting the various types of neurons discussed in previous sections. The convention adopted in this thesis follows that used in the Trajan Neural

Network package [56].

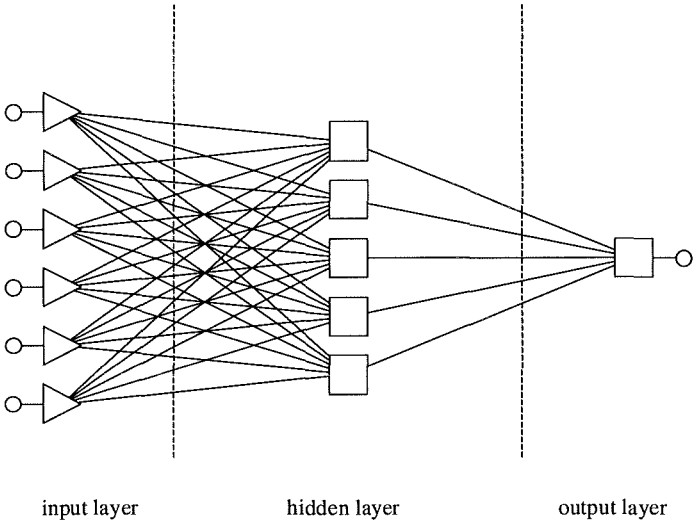

Figure 4.11 – Typical Feed forward Neural Network

An input is shown as a triangular node, a neuron with a linear activation level, (equation (4.1)), is shown as a square node and a neuron with a radial activation level, (equation (4.39)) is shown as a circle. Thus the diagram shows a six input, fully connected feed forward neural network. The network has three layers, an input layer with six inputs, a hidden layer with five neurons and an output layer with a single output neuron. In other conventions this may be referred to as a two-layer network, the input layer not counting as a neuron layer since it performs simple input scaling. The activation function of hidden layer linear neurons is logistic (equation (4.11)) or hyperbolic tangent (tanh, equation (4.12)), whilst the activation function of RBF neurons are Gaussian (equation (4.37)). Linear units in the output layer have linear activation functions, i.e. their output is the same as the activation level or a unity identity.

For each connecting line there is an implied weighting unit not shown explicitly in the diagrams. As seen with PADALINEs, non-linearly separable classification can be achieved by extending the vector space. In terms of Neural

Networks, this is equivalent to adding an intermediate or Hidden Layer between the Input and Output Layers. The neural network shown in Figure 4.11 is also referred to as a multi-layer perceptron (MLP).

An RBF neural network is shown in Figure 4.12. This shows a three layer, six input RBF network. The hidden layer units have radial activation levels, (Euclidian distance), and Gaussian activation function. The output neuron is a simple linear activation function.

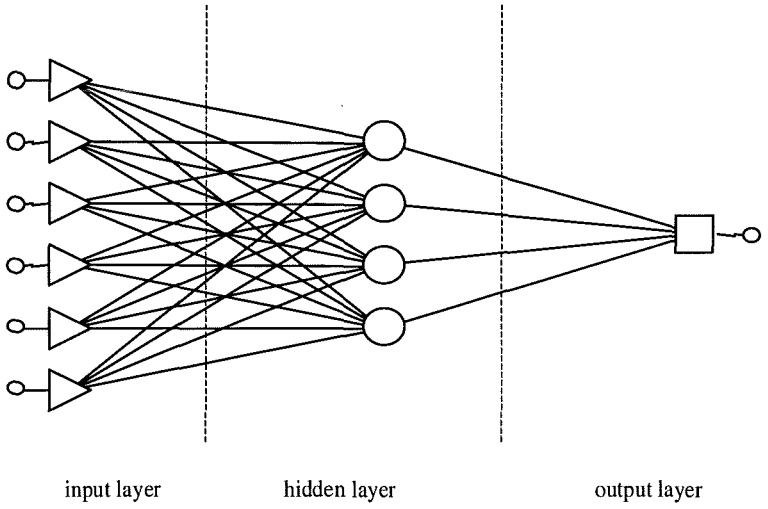

Figure 4.12 – Typical RBF Network Architecture

4.3.1.1 Network Architecture

The number of hidden layers decides the architecture of the network and the number of hidden units required to solve a problem, be it function approximation or classification. The number of hidden layers is, in part, determined by the activation function of the neurons. Networks with neurons that have a step activation function may require two hidden layers. Neurons with logistic activation functions require only a single layer but may need a large number of hidden units. Some simple non-linear problems can be approximated using a simple linear network.

Hornik [57] describes a three layer MLP with a sigmoid output neuron as a universal approximator that can be trained to approximate any input and

output mapping. Park and Sandberg [58] prove that a single hidden layer RBF network is capable of universal approximation. The key determinants in network architecture are:

- the number of training cases available
- the amount of "noise" in the data
- function or classification complexity
- training algorithm used

Generally, more complex problems require higher numbers of hidden units, however it is possible to over fit or under fit the training data where the number of weights in the network is significantly different for the number of training cases. Trajan [56] advises that network design is matter of heuristic knowledge based on trial and error, but includes an "Intelligent Problem Solver" (IPS) for network design.

Gurney[48] describes two methods of network design involving: clustering of the vector space and assigning neurons to each cluster; and a method that adds or subtracts neurons to a network during training.

Once constructed, neural networks can be pruned to remove inputs that have little effect on network performance. This is similar to a sensitivity analysis whereby inputs are disabled and the effect of their loss of contribution to the network can be assessed in terms of error introduced. There are no hard and fast rules for network design. In this research initial attempts were made to construct a neural network, the results of this work are described in section 5.2.2. Following this phase, the Trajan package was used to investigate neural network architectures in the context of compressor control.

4.3.1.2 Training and Generalisation

It was the application of an error back propagation training algorithm, first described by Werbos [59], by Rumelhart and McLelland [60] that overcame the

linear separability and *a priori* data knowledge limitations associated with the TLU .

Back-propagation training of networks involves two-step procedure. During the training phase, the network is presented with training data. The inputs ripple forward through the network to give an output. The difference between the actual output and the training target, i.e. error, is propagated back through the network from the output layer to the hidden layer weights. The weights on each of the layers are then adjusted to minimise the error as it passes back through the net to the input layer. The change in weight expression, for the *k*th hidden Neuron becomes:

$$\Delta w_{ki} = \alpha\sigma'(a_k)\delta^{k} x_{ki}^{p} \tag{4.40}$$

This is sometimes known as the generalised delta-rule. Back Propagation can be slow to train as many epochs may be required before the training set error is reduced to within the error criteria, if it is ever met at all. One cycle of presenting the complete training set to the network and calculating weight adjustments, for each input vector, is referred to as an epoch. The Back Propagation algorithm can become trapped by local minima, network training is trapped in a local solution and the network is unable to generalise over the whole input domain. This problem can be minimised using a variety of techniques, including adding a momentum term in the weight adjustment equation. The momentum term is a product of a momentum constant and the weight adjustment from the previous training step. As the error function converges on a minimum the influence of the momentum term reduces. This is indicative that a true minimum is being approached.

Function optimisation schemes are also used for training such as: Quasi-Newton, Levenberg-Marquardt and conjugate gradient descent. These schemes are described in Haykin [52] and Bishop [61]. Each scheme has its strengths and weaknesses.

Training using any algorithm follows the same general theme. The training data set is fed into the network until the error criteria for the forward pass of the network is met. The error criteria usually refers to the RMS error of the training set input vectors.

Once trained, it is not possible to prove that the network has achieved an optimal solution over the input domain. This would require all possible solutions to be compared with the neural network output. A separate validation set of data, usually derived from the same pool as the training data can be used to check network performance. However, by withholding data from the network, the data model on which the network is trained is incomplete.

The network can become over trained i.e. it is trained to recognise or approximate the training set but cannot generalise, or extend its performance to recognise or approximate outputs, from input data out with its training set. The validation set error is therefore used to control the extent of training the network receives. If the validation error falls in line with the training error then training continues. If the training error continues to fall whilst the validation error does not training should be stopped. This circumstance is indicative of over training i.e. the network is being trained to approximate the training set pattern but not other patterns, from the same vector space, which are in the validation set. Error comparison between training sets and validation sets is also referred to as cross validation, see Cheng and Titterington [62].

4.3.2 Self Organising Neural Networks

Self-organising neural networks learn features in pattern space. This type of network is sometimes referred to as a Self Organising Map or SOM. Unlike feed forward networks they do not undergo supervised training. Usually there is no training target for the network to emulate. The network is presented with a training set and competitive learning dynamics are used to assign a neuron to a feature or data cluster. Once a network has learned the pattern space, an input vector causes the most representative neuron (or exemplar vector) to have the

highest activation level. Classes may then be assigned to each neuron to associate inputs with the assigned class.

More sophisticated self-organising networks may include classification of clusters. Techniques such as "k-nearest neighbours", Bishop [61], allows excitation, to a lesser extent, of neurons adjacent to the closest match exemplar. In this way a "topology" of the feature space can be constructed. A Kohonen self-organising map [62], uses inhibition as well as excitation, of neurons to improve network classification. This is implemented using Learning Vector Quantisation (LVQ) (also known as Adaptive Vector Quantisation, AVQ) techniques.

4.3.2.1 Network Topology

A SOM consists of an input layer and an output layer. The network inputs are scaled and normalised in the input layer. The output neurons are RBF type neuron. Each input is connected to each output neuron, as shown in Figure 4.13.

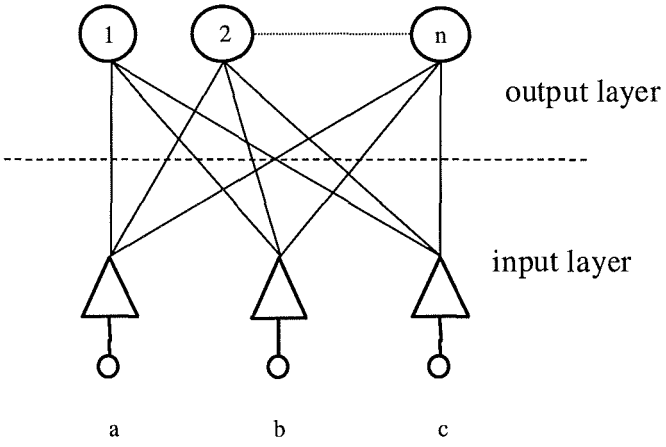

Figure 4.13 – Self-Organising Network Layers

Figure 4.13 shows three inputs a, b and c each connected to every one of the n output neurons. The input connections to one output neuron are sometimes

referred to as the fan-in vector. To envisage the feature space topology the output neurons may be arranged in a matrix format. Figure 4.14 shows an array of output neurons, A1 through E5, each with the fan in vector **F**. The figure depicts that the presented fan-in vector **F**, principally excites neuron C3 as denoted by the darker shading. Secondary levels of excitation are depicted in the adjacent, or neighbouring, neurons by the lighter shading. In this way features in a vector space can be visualised.

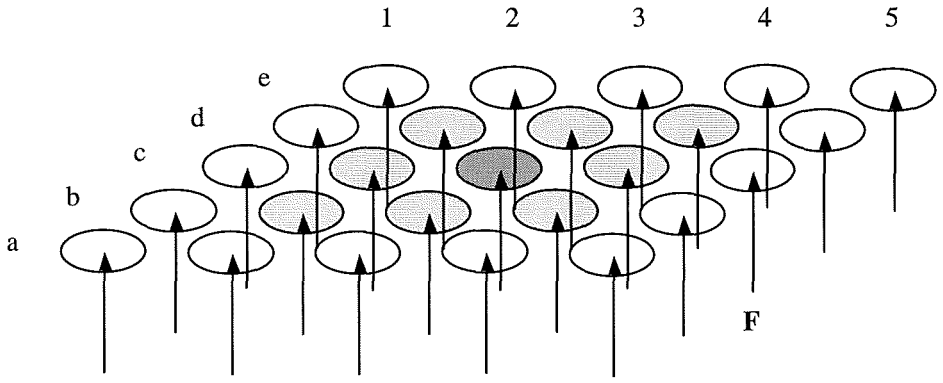

Figure 4.14 – SOM Output Neuron Array

As with feed forward networks, there are no hard and fast rules to determine the number of neurons required to represent feature space. As the complexity of the feature space increases so the numbers of output neurons required to map feature space increases. The locations of the RBF neurons in vector space are selected at random or at feature locations, if these are approximately known.

4.3.2.2 Competitive Learning

Data is presented to the SOM and the Euclidian distance, d, of the vector **F** to each of the RBF neuron centres is calculated using equation (4.41).

$$d_n = \|F - X_n\| \tag{4.41}$$

where X_n is the n^{th} output neuron exemplar vector

F is the input fan-in vector

The output neuron, for which d_n is a minimum, is the "winner". Only the winning neuron has its weights adjusted. The weights of the RBF neuron is the locating vector of the neuron in feature space, thus the neuron moves towards that input vector. This is sometimes known as "winner takes all" learning. The change in neuron weights, Δw, is written as:

$$\Delta w = \begin{cases} \alpha(F - X_n) : "winner" \\ 0 \qquad\quad : "loser" \end{cases} \tag{4.42}$$

where α is a learning rate

$(F\text{-}X_n)$ is the scalar distance for each vector component

LVQ learning involves inhibition as well as excitation of neurons. Inhibition is met by moving losing neurons away from the input vector. This is represented in equation (4.43).

$$\Delta w = \begin{cases} \alpha(F - X_n) & : "winner" \\ -\alpha(F - X_n) & : "loser" \end{cases} \tag{4.43}$$

Each data point is presented to the network and each winning output neuron has its weights adjusted until a pre-set number of epochs have been completed. One epoch is completed when the complete data set has been presented to the network. The learning rate is reduced linearly with the number of epochs completed, reducing the "upset" to the network as learning progresses.

4. 4 Data Clustering Techniques

The application of RBF neurons requires initial placement of the neuron centres in vector space. Location of the centres is stored in the weight values which are adjusted during training. Adjusting the weights through training moves the RBF centre in the vector space.

Initial placement of RBF centres can be chosen at random or by using data clustering techniques. Two "stand alone" data clustering techniques are described below. These techniques could also be used to examine vector space independently of applying neural networks.

4.4.1 Hard Clustering, K-means

The K-means clustering technique, described by Bishop [61], is used to assign data, in vector space, to a cluster of similar data points. The number of clusters, K, in the vector space is chosen arbitrarily. The algorithm assigns data points to each of the K clusters as *disjoint* subsets of the vector space. The clustering function, J, is written as:

$$J = \sum_{j=1}^{K} \sum_{n \in Sj} \left\| x^n - \mu_j \right\|^2 \tag{4.44}$$

where K is the number of clusters

S_j is the jth data subset

x^n is the nth point in the jth data subset

μ_j is the mean of the x^n points in the jth data subset

The mean of the jth data subset, μ_j, is given by:

$$\mu_j = \frac{1}{N_j} \sum_{n \in Sj} x^n \tag{4.45}$$

where N is the number of points in the in the jth subset

Data points are assigned to a cluster and the mean of the cluster is calculated. J is then calculated for each point. A data point is assigned to the nearest cluster and the cluster mean recalculated. This procedure continues until no data points are re-assigned clusters. In assigning data points to disjoint subsets or clusters the data space continuum, or homogeneity, is lost. This may result in "holes" in the vector space if new data points are added without re-clustering taking place.

4.4.2 Fuzzy clustering, C-means

Fuzzy C-means (FCM) is described by Bothe [64]. It is described as an extension to the "crisp" clustering of the K-means technique. In FCM each cluster is a fuzzy set and each data point has a fuzzy set membership of each cluster. It is possible to preserve the homogeneity of the vector space by classifying data points with a degree of membership of all the clusters identified in the vector space.

Two conditions apply to fuzzy set membership, for any point x_k the sum of its fuzzy membership of N clusters is unity:

$$\sum_{j=1}^{N} \mu_{jk} = 1 \quad : \forall k = 1 \, to \, K \tag{4.46}$$

the total fuzzy membership of all points, K, in N clusters does not exceed the number of points:

$$\sum_{k=1}^{K} 0 \leq \mu_{jk} \leq K \quad : \forall j = 1 \, to \, N \tag{4.47}$$

The centre of the j^{th} fuzzy cluster, c_j is given by:

$$\mathbf{c}_j = \frac{\sum_k (\mu_{jk})^q \mathbf{x}_k}{\sum_k (\mu_{jk})^q} \tag{4.48}$$

where μ_{jk} is the fuzzy set membership of the k^{th} point in the j^{th} cluster

x_k is the k^{th} point in the vector space

q is a "contrast" parameter given a value, ~1.5

The fuzzy cluster membership of the k^{th} point in the j^{th} cluster is given by:

$$\mu_{jk} = \frac{1}{\sum_{i=1}^{N} \left(\frac{\|x_k - c_j\|}{\|x_k - c_i\|} \right)^{1/q-1}} \tag{4.49}$$

Pedrycz [65], proposed *conditional* fuzzy clustering where an objective function is introduced to preserve the homogeneity of clustering in vector space. This method allows the intersection of clusters to be defined as separate, distinct clusters.

4. 5 Network Metrics

Bishop [61] suggests that cross validation alone may cause over training of the network to the validation set and that a third set of data, test data, should be used for an independent comparison of the performance of networks. The output data from the network can be treated in the same way as regression data to establish a "quality" of fit. There are a number of standard statistical metrics which can be applied to assess the performance of a neural network. The most obvious of these is a correlation coefficient. This and other metrics are described in Trajan [56].

4.6 Discussion

There are strong similarities between statistical techniques and neural networks as described in [63]. Two of these areas are function interpolation and pattern recognition, or data clustering. Joseph et al [66], concluded that neural networks performed better than regression models where high non-linearities were present, but that both performed poorly. It was further reported that data partitioning improved both regression model and neural network performance. These techniques would be of use in representing the compressor performance characteristic and identifying operating clusters in the suction conditions vector space. An advantage of using neural techniques over statistical techniques is that no particular expertise would be required to retrain a neural network to reflect changing characteristics of a compressor with time.

Current trends in neural network research include:

- the use of probabilistic techniques to create graphical models and to reduce dimensionality in large data sets, Bishop [67].

- probabilistic combination of weight components to encompass many neural networks into one representative network, so called "product of experts, Hinton [68]

- use of a "spiking" neuron model to investigate the propagation of signals through a neural network when presented with unknown stimuli leading on to rapid learning of neural networks, Gerstner [69]

4.7 Conclusions

It is anticipated that a compressor performance characteristic may be represented using a neural network as a function approximator. The use of neural networks will preclude the need to identify regression coefficients and to continuously update their values, as compression conditions change. Function approximation may be met with either a multi-layer perceptron or a radial basis function, feed forward neural network. Neural network performance in learning compressor head maps is researched in Chapter 5.

5 Neural Networks and Compressor Performance Characteristics

5.1 Introduction

This chapter describes research into the use of neural networks for the modelling of compressor performance characteristics. A program of testing is devised to assess the performance of neural networks. The tests range from the ideal case of modelling an exact compressor head map to modelling a random set of compressor operating points. Two different sets of compressor performance characteristics were used to train neural networks. The first set used for initial testing was atypical, non-fan law data. The compressor performance characteristics used for empirical data closely corresponded to the Fan Laws. During numerical testing, data subsets were generated using the methods described in Chapter 3.0. No derived data was generated from the atypical performance characteristics.

A brief summary of the evaluation of two neural network packages is undertaken and an account of the development of a neural network, using non-specialised software, is included. The key features required of neural network software, for research purposes, was flexibility in usage, accessibility to the various network parameters and returning consistent, *reproducible* results. It was found that the greater the flexibility required, the more complicated the software became to use.

The first commercial package, NCS NeuFrame [70], evaluated used an object-oriented type Graphical User Interface, GUI, for network construction using standard objects. Since the software was an evaluation copy there was a restriction on the data set size. Following this, a neural network was programmed using spreadsheets and macros. Whilst this offered vast flexibility, changing the network architecture was time consuming and each network

required a dedicated spreadsheet and specific macro using specific spreadsheet cell references. The second commercial package evaluated was a fully featured, evaluation copy of Trajan [56]. This package featured feed forward, supervised training networks and self-organising maps. Several different training algorithms could be selected. In addition it included a proprietary Intelligent Problem Solver (IPS), which designed and tested several tens or hundreds of networks automatically.

The outcomes of the structured testing results are discussed and demonstrate that a neural network can accurately represent compressor performance characteristics. A method to embody the neural network head map representation in a control scheme is introduced.

5.2 Neural Network Design

5.2.1 NCS NeuFrame

Initial investigations into the use neural networks began with an evaluation copy of a commercial neural network package, NCS NeuFrame [70]. A screen shot of the NCS package is shown in Figure 5.1.

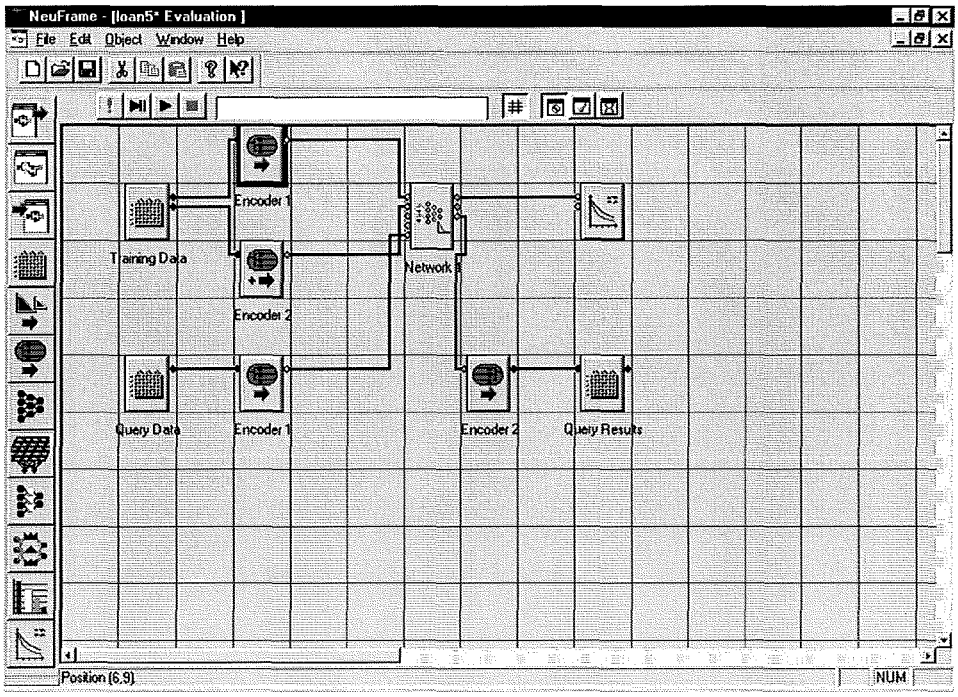

Figure 5.1: NCS NeuFrame Workspace Screen Shot

The screen shot shows the NCS workspace with an implementation of a back propagation neural network. Neural network component objects are shown in the vertical, left hand toolbar. These include data sheets, scalers, encoders, back propagation neural networks, Kohonen neural networks, RBF neural networks, NeuFuzzy neural networks and graphing objects.

The neural network shown consists of two data sets, Training and Query. The inputs to the neural network object are encoded either as categorical or numerical data. Several variables can be encoded in one encoder object. The back propagation neural network object can be trained using the Training data sheet. Output from the neural network is then de-scaled and written to the Output data sheet. Training error is passed from the neural network object to the graphing object, for display. Once trained, output from the data in the Query Data sheet inputs can read from the Query Results data sheet. Implementation

of custom neural networks could be done through drag n drop of the network objects onto the workspace, then graphically linking connections with specific block I/O ports, as shown in Figure 5.1.

The evaluation version of the software was only able to take thirty data points, less than the number required for compressor head map training. Whilst enabling visualisation of neural network concepts, it was conclude that access to the network parameters was limited by the object approach. The NCS package was not used with compressor data.

5.2.2 *A Feed Forward Neural Network using a Spreadsheet and Macros*

It was decided to build a network using a spreadsheet and macro type approach. These tools were readily available and some work had already been done on spreadsheets examining the calculations used in the neural networks. A commercial, Excel97 neural network add-in was available [71] but creating a custom neural network would allow full realisation of the equations used and, at least, one training algorithm. A neural network was designed and implemented using a common spreadsheet package, Excel97.

The feed forward phase was implemented in the spreadsheet automatic re-calculation feature. Back propagation of the error was implemented using a Visual Basic (VB) "macro". Training was activated and continued until a fixed number of epochs had passed. A screen shot of the spreadsheet is shown in Figure 5.2.

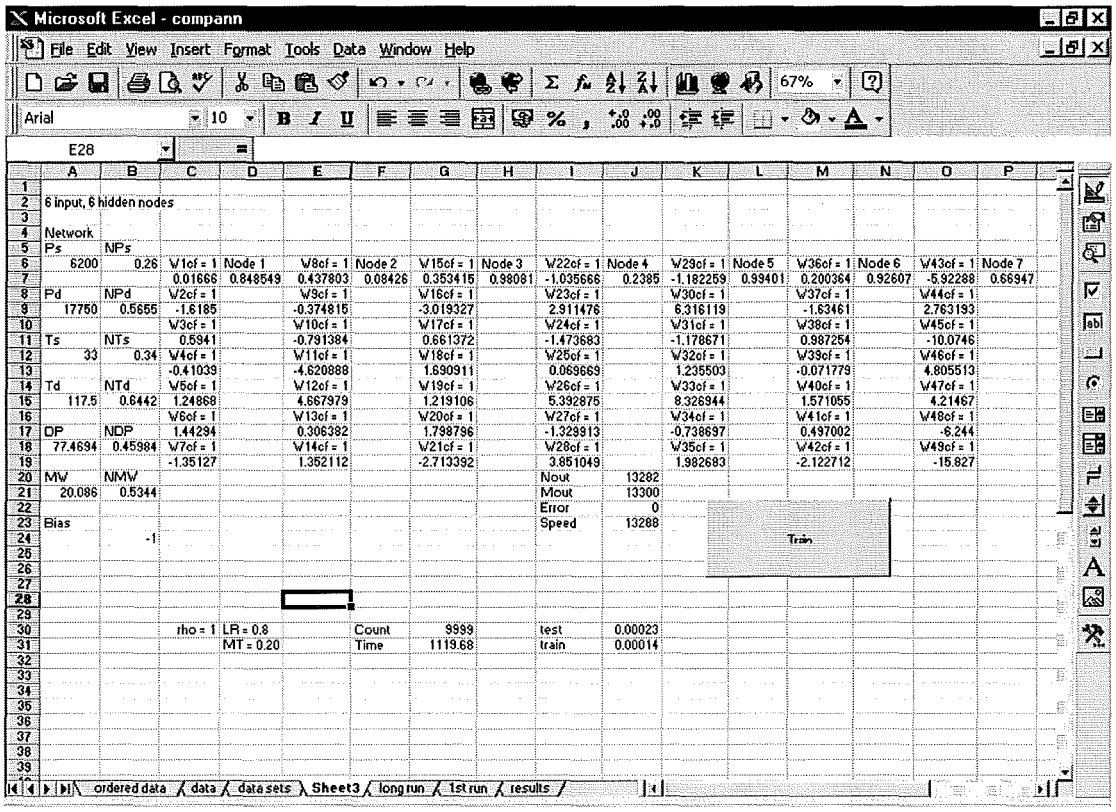

Figure 5.2: Excel97 Implementation of a Neural Network

The screen shot shows a six input, six hidden node, (node 7 is the output node), feedforward neural network. Data from the basic head map array was written to the input cells, down column A, by the VB macro. As these values were written, the automatic re-calculation feature allowed the changes to "ripple" through the spreadsheet to the output cell, P7, on the far right. (The re-calculation process was completed prior to execution of the next line of macro-code. This was verified by inserting a time delay in the macro and observing propagation of values through the network). The input values were normalised in column B. Each of the normalised inputs were weighted with the appropriate weight in columns C, E, G, I, K, M, O using equation 4.1. The sum of the weighted values (activation) was passed to the hidden layer neuron cell, D, F, H,

J, L, N. Each hidden layer neuron had a logistic sigmoid activation function, as described in equation 4.11. Finally, the weighted sum the outputs from the hidden nodes was passed to the output node, column P, using equation 4.1. The output node was also assigned a logistic sigmoid function described by equation 4.11.

The training method was Gradient Descent using the Generalised Delta Rule. The macro read the normalised output from the network and compared it with the normalised target value to generate an error. The error was summated for the training set as shown in equation 4.22. After one complete pass of the training set (or one epoch) the summated error was "back propagated" through the network to adjust the weights. The change in network weights was calculated using equation 4.28. The results of training using this neural network are presented in section 5.6.1.1. The Excel97 model as described was relatively unsophisticated in that only one training algorithm, Generalised Delta Rule, was implemented. It was satisfactory in allowing access to the various neural network parameters visualising the effects of changes to them.

Changing the network architecture could involve major manipulation of the spreadsheet and macro, depending on the number of hidden nodes designed in. This was a significant drawback in that complex training algorithms would have to be edited and the possibility of error introduced by each edit. For this reason the spreadsheet neural network approach was not used for testing.

5.2.3 Trajan Neural Network Package

A third commercial neural network was examined, Trajan 4.0 [56], received as a fully featured, evaluation copy. The Trajan package allowed full access to network parameters and also featured a number of training algorithms such as Back Propagation, Conjugate Gradient Descent, Quasi Newton, Levenberg-Marquardt amongst others. Two additional features were a Windows API

programming interface and an Intelligent Problem Solver (IPS) for neural network design. The IPS proved to be a valuable tool, able to achieve good optimisation of the number of nodes required in a network, and of node placement in vector space. Whilst these properties cannot be absolutely verified the network metrics indicate accurate modelling of compressor characteristics, as will be seen. This could indicate near optimal performance in network design. A screen shot of the standard Trajan screen layout, using IPS is shown in Figure 5.3.

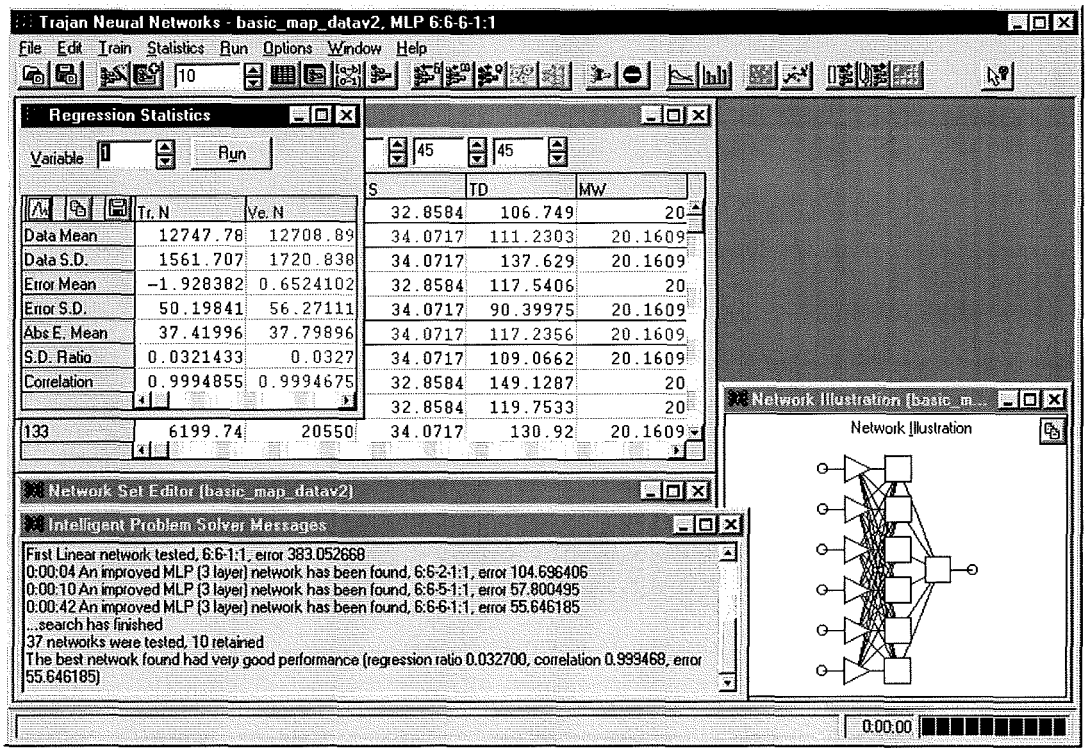

Figure 5.3: Trajan Screen Shot

Trajan was used as the main research tool in representing compressor characteristics with neural networks. In particular the IPS produced a diverse range of network architectures with varying performance.

In the initial research phase of network design and testing, one data set and a single division of that data set was tested. The reason for this was to attempt to identify any error inherent in the design and training process, since the training data was the same in all networks.

For the structured testing phase, the basic procedure with Trajan was to create and save a data set in Comma Separated Value (CSV) file format. The data set could then be opened for training and with the IPS invoked from the appropriate menu. The IPS would investigate a number of network designs and present the best performing, in terms of error standard deviation. The network set was then saved with the data file in Binary Data Format (BDF). The BDF file was saved as a unique binary file, hence the parameters of each and every network saved could be reinstated. This method preserved the data set split between Training, Test and Validation data. Training was only initiated with the CSV file.

Length of training varied from fifteen minutes for a medium search of the basic data sets, to more than twelve hours for a thorough search of MLPs on random data sets.

5.3 Initial Research

The initial research phase involved a qualitative assessment of how neural networks might represent or encode compressor characteristics. In this respect the inputs to the neural networks would be:

- suction temperature and pressure, T_s and P_s respectively
- discharge temperature and pressure, T_d and P_d respectively
- differential pressure, dp, to represent actual inlet volume flow rate
- molecular mass, MW

Compressor speed, N, would be the target. Typically, neural network

architecture would appear as that shown in Figure 5.4, consisting of an input layer, a hidden node layer and a single output node.

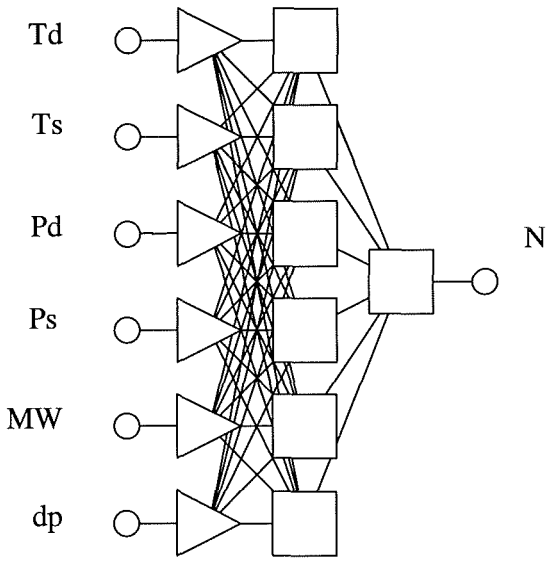

Figure 5.4: Typical Test Neural Network Architecture

A neural network of this architecture can be demonstrated to be capable of closely representing a compressor performance characteristic as shown, by points of estimated discharge pressure plotted against lines of constant speed, in Figure 5.5.

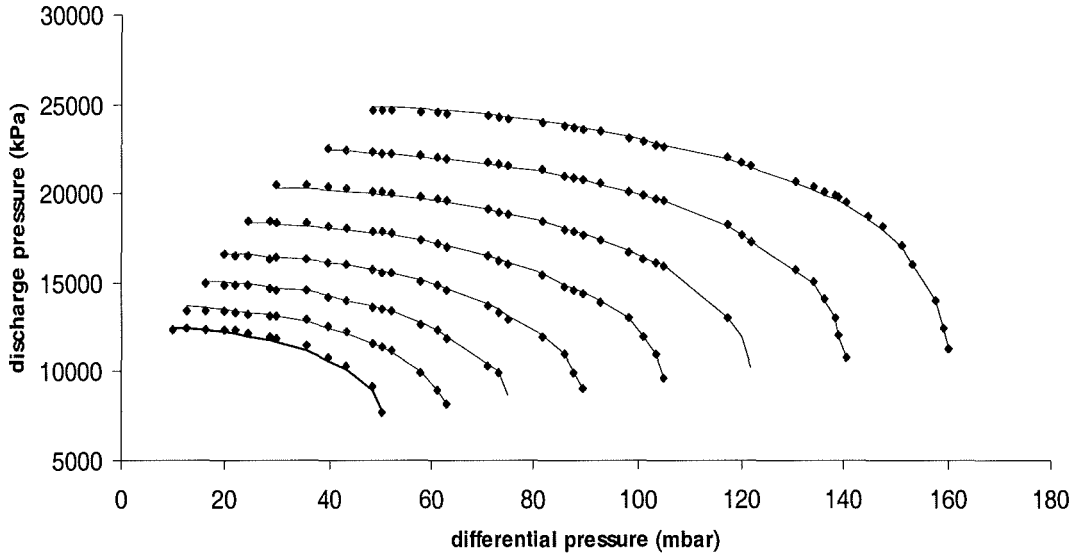

Figure 5.5: Compressor Performance Estimation by Neural Network

The results of the ad-hoc testing are described in this section whilst the results of a formal testing programme are reported in section 5.6.

Initial research involved repeated network design for the basic data set using the IPS in Trajan. This phase also gave some indication regarding the performance of Trajan in terms of repeatability given the same split of data set over and over.

A one hundred point data set, the number of points taken from the basic head map, was trained using the Trajan IPS with standard constraints. An "Optimal" search, for optimal, diverse networks with input sub-sets enabled was first run. This was followed with an "Extensive" search using the same BDF files. The results are summarised in Appendix C, Tables C1 – C3.

5.3.1 Optimal Architecture Search

The "Optimal" search produced Linear, MLP and RBF networks in the following proportions: 43%, 50% and 7%.

The linear networks generally used a subset of inputs consisting of Ts, Td and dp, achieving a linear solution to an inherently non-linear process. This is believed to have come about through the use of constant Ps, Ts, and MW values in use. In addition T_d was calculated using a fixed value of specific heat ratio of 1.3, effectively linearising the relationship between T_s and T_d as P_d/P_s. The fixed value of specific heat ratio was used as a starting point for the initial research. Intuitively, the compressor characteristics can be thought of as an "onion skin" type function so the marginally higher numbers of MLP networks was thought to be in keeping with that notion. The levels of error were one order of magnitude higher in the linear networks that suggests that they reflected best performance only in the absence of a good MLP network. Only one RBF function had better performance than MLP or linear networks.

5.3.2 Extensive Architecture Search

The "Extensive" search resulted in 93% of all neural networks having an MLP architecture. The average number of hidden nodes was six. Input subsets comprised of P_d, T_d and dp. The average of the error standard deviations for the twenty-nine MLP networks was 24 rpm. This suggested that an MLP would be, at best, capable of estimating compressor speed to around 50 rpm at 95% confidence level using a subset of inputs comprising P_d, T_d and dp with fixed suction conditions. An average number of 6 hidden nodes was noted in the MLP architecture neural networks.

5.3.3 RBF and MLP Neural Networks

With fixed suction conditions and the constraint of all six inputs to be used all thirty networks designs were the same RBF network. This network exhibited an error standard deviation of 64 rpm. It was noted that the input scaling associated with the fixed suction parameters was set as constant by Trajan. When training the Excel97 neural network the fixed suction conditions had been expressed as a fraction of a notional input range. The data set was modified such that the fixed conditions would appear to be the same fraction of input range as the Excel97 neural network input. This was achieved by changing two values of the training data for each of P_s, T_s and MW to represent the minimum and maximum of the required input range. For the Initial Research, the values were changed as shown in Table 5.1. Values used during Numerical Testing are as advised in sections 3.4.2 and 3.4.3.

Parameter	Nominal	Max.	Min.	Range Fraction
Ps	6200	6200.458789	6199.838804	0.26
Ts	33.3	33.302198	33.298867	0.34
MW	20.086	20.086935	20.084927	0.53

Table 5.1: Modified suction values for range scaling

The modified data set was then used with Trajan to train networks. In all cases the modified data set delivered MLP networks. This was taken as confirmation that the underlying relationships in compressor head map would probably be best represented by an MLP network.

On average 9 hidden nodes were present in the MLP networks and an average error standard deviation of 24 rpm was recorded. The probability density function of the distribution of the error standard deviations is shown in

Figure 5.6, plotted against error standard deviation. The distribution of the error standard deviations follows a normal distribution.

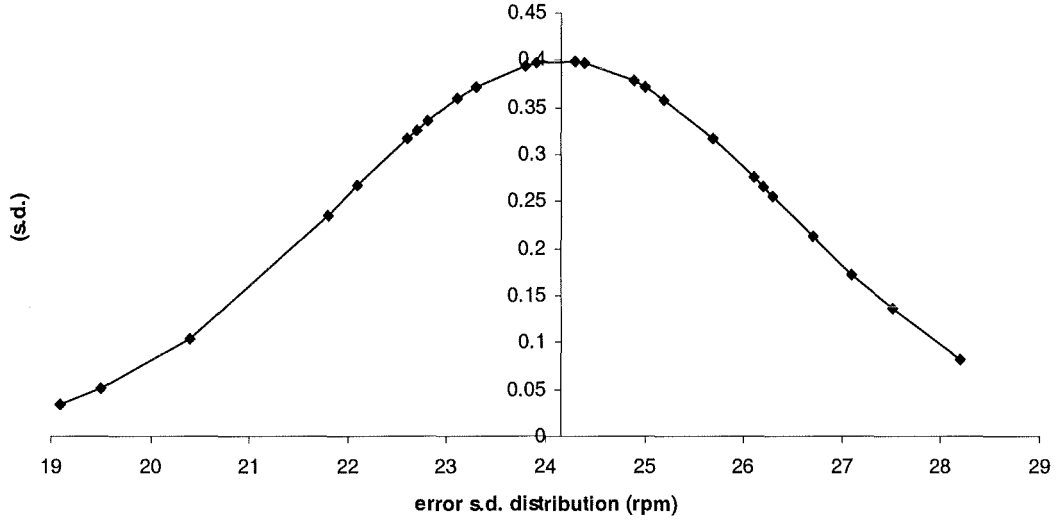

Figure 5.6: MLP error standard deviation distribution

Allowing changes of data set split, with all inputs forced resulted in a network average hidden node count of 9, with an average error standard deviation of 31 rpm being recorded. The performance data used for the initial research was subsequently found not to conform to the Fan Laws hence no quantitative testing was done using derived data for different suction conditions.

Following initial investigations a structured test program was devised to generate realistic compressor operating data.

5.4 Numerical Testing

Performance characteristics used for numerical testing closely conformed to the Fan Laws, hence derived data sets representing varying suction conditions were generated for empirical testing. Changes in compressor performance characteristics, due variations in suction conditions, were calculated using the flow and head coefficient techniques described in Chapter 3.

Four data sets generated to assess neural network training performance for:

(a) variations in training set size;
(b) inclusion of instrumentation noise in the input/output signals;
(c) training on random operating points;
(d) training on random operating points with extended input sets i.e. gas composition in place of molecular mass;
(e) use of input subsets used for training;
(f) extrapolation over the full performance characteristic range from a restricted range training set.

The sets are detailed below.

1. *"Clean" Set.* Nominally fixed values of molecular weight, suction temperature and pressure to represent the conditions at which the default compressor performance characteristics were created. Neural network inputs and outputs (simulated field signals) are taken to be free from noise. The compressor performance characteristics are exactly defined. Variation in neural network performance can only be due to data set split between training/test/verification sets, training set size and training algorithms used. This is a "best case" as far as neural network training is concerned hence these networks are taken to represent

control cases. The neural network architecture for these data sets would typically look like figure 5.4

2. *"Noisy" set.* Data used as for the *"Clean" set* but with the addition of random instrument noise equivalent to typical instrument uncertainty levels. Differences in results between this set and the same data set size from the "clean" set would be indicative of the effects of neural network training performance with "noisy" inputs. The "noisy" set was also used for examining neural network performance using input sub sets. The neural network architecture of the input sub sets would typically look like that shown in Figure 5.7

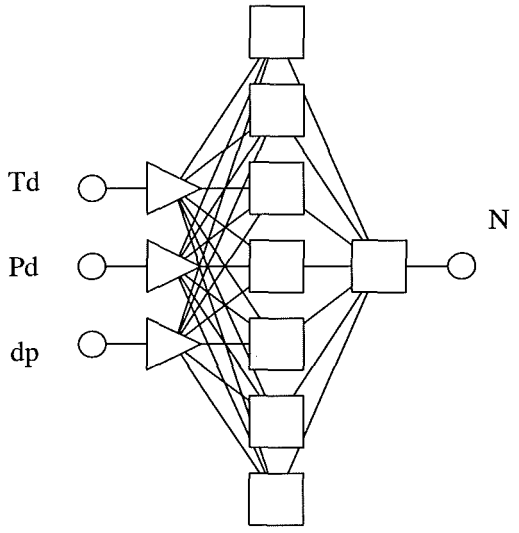

Figure 5.7: Input Subset Neural Network Typical Architecture

3. *"Random" sets.* The *"random" set* data was created to assess the performance of neural networks trained on a number of random points across the operating envelope of the compressor in terms of its performance characteristics at varying suction conditions

and gas composition. This is probably the most onerous compressor performance control scenario where operating point can vary, largely dependent on the process plant upstream of the compressor.

4. *"Extrapolated" set.* This fourth uses a limited training set taken from approximately the middle third of the flow coefficient range for each speed. Training with the limited data would allow assessment of the neural network to extrapolate across the entire operating range of the compressor. This may include assessment of how well the neural network may estimate compressor speed for previously un-encountered conditions.

5. *"Normalised" set.* The fifth set comprises the *"clean" set* normalised against compressor speed i.e. this is a non-dimensional data set comprising flow and isentropic head coefficients.

5.5 Generating Test Data

All data sets, described in Section 5.4, were generated from the basic compressor performance. Set 1(a) comprised of 96 original compressor performance data points. Sets 1(b) and 1(c) comprised of the original points with interpolated points added to give 184 and 496 point respectively. Set 2 comprised of Set 1(c) with the addition of simulated noise present on the neural network inputs. Set 2 was used to test input subsets for training purposes.

Set 3(a) comprised 500 points randomly selected within the operating envelope of the compressor. Sets 3(b), (c) and (d) comprised of 1000 randomly selected points containing gas property inputs represented by: only molecular mass as in previous data sets; full AGA 8, 21 input component representation; "short", 10 input component analysis typical of OGC analysis.

Set 4 was comprised of 281 data points extracted from set 1(c) to represent a restricted training set. Set 5 comprised of the normalised flow and head coefficients.

The methods and procedures used to generate the data sets are described in section 3.4

5.6 Results of Testing

The data sets described in section 5.5.1 though 5.5.4 were run through Trajan Intelligent Problem Solver (IPS), thirty one times. Thirty-one runs were chosen to bring the 95 % confidence limits down to two standard deviations. Each run produced a number of neural network architectures. Whilst the same data point set was used for each run, the split between training, test and verification sub-sets varied between runs. The "best" performance network was taken for each run and the thirty-one best performing networks presented as the "results" for the training of each data set. It was noted that networks trained with the smaller data sets were prone to over training hence the % incidence of over-training for each data set is presented.

The results were treated as thirty-one samples of a population of all possible neural networks for the given data set. The structure of the results is shown in Figure 5.8.

$$\begin{array}{cccccc} S_1 & N_{1,1} & N_{2,1} & \cdot & N_{j,1} \\ S_2 & N_{1,2} & N_{2,2} & \cdot & N_{j,2} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ S_n & N_{1,n} & N_{2,n} & \cdot & N_{j,n} \end{array}$$

Figure 5.8: Structure of Results

S denotes the training target for each data point, n = 96, 184, 496, 500 or 1000 depending on the data set for which results are being calculated. N is the j^{th} neural network estimate of speed, $0 < j < 32$, for the n^{th} data point. The following statistical parameters were calculated to represent neural network performance. Standard statistical techniques have been employed, drawn from statistics reference texts, Griffiths et al [72] and Lapin [73].

The absolute peak positive and negative error of each data set calculated as the maximum and minimum values of:

$$\varepsilon_{S_n} = \left(S_n - N_{j,n}\right) \tag{5.1}$$

The relative peak positive and negative error calculated as a percentage of the training target:

$$E_{S_n} = \frac{100\left(S_n - N_{j,n}\right)}{S_n} \tag{5.2}$$

Peak positive and negative average speed error calculated as an average of the thirty-one networks calculated for the n^{th} data point:

$$\overline{S}_n = \frac{\sum_{j=1}^{31}\left(S_n - N_{j,n}\right)}{31} \tag{5.3}$$

The 95% confidence limits for the positive and negative maximum and minimum values of $\overline{S}_n$ as:

$$sd_{n95} = \frac{2.042}{\sqrt{31}}\sqrt{\frac{31\sum \varepsilon_{S_n}^2 - \left(\sum \varepsilon_{S_n}\right)^2}{31(31-1)}} \tag{5.4}$$

The average absolute speed error for each network:

$$\varepsilon_{S_j} = \frac{\sum_{i=1}^{n}\left(S_i - N_{j,i}\right)}{n} \tag{5.5}$$

The standard deviation of each network's average absolute error:

$$sd_j = \sqrt{\frac{n\sum \varepsilon_{S_j}^2 - \left(\sum \varepsilon_{S_j}\right)^2}{n(n-1)}} \tag{5.6}$$

The average of the neural networks average absolute error:

$$sd_{bar} = \frac{\sum_{j=1}^{31} sd_j}{31} \tag{5.7}$$

The Pearson correlation coefficient for the j^{th} neural network calculated as:

$$r = \frac{n\left(\sum_{i=1}^{n} S_i N_{j,i}\right) - \sum_{i=1}^{n} S_i \sum_{i=1}^{n} N_{j,i}}{\sqrt{\left[n\sum_{i=1}^{n} S_i^2 - \left(\sum_{i=1}^{31} S_i\right)^2\right]\left[n\sum_{i=1}^{n} N_{j,i}^2 - \left(\sum_{i=1}^{n} N_{j,i}\right)^2\right]}} \tag{5.8}$$

The average of the 31 networks Pearson correlation coefficients calculated as:

$$\bar{r} = \frac{\sum_{i=1}^{31} r_i}{31} \tag{5.9}$$

The following parameters are reported to define and compare neural network performance: peak positive and negative values of E_{Sn} ; peak positive $S_{n95} = S_n + sd_{n95}$; peak negative $S_{n95} = S_n - sd_{n95}$; sd_{bar} ; $\bar{r}$. The incidence of neural network over training is expressed as the number of networks exhibiting over training with the total number of networks trained to achieve 31 networks for analysis.

5.6.1 Data set 1 - "Clean" Data Set Results

Three data set sizes were used as follows: 96 data-points representing the actual default compressor performance points - 12 actual inlet volume / isentropic head data pairs for each of eight constant speed lines, (set 1(a)); the original set supplemented with interpolated points, 184 points in total, (set1(b)) and the original set supplemented with interpolated points, 496 in total, (set1(c)). For set 1(b) linear interpolation was applied to the flow coefficient such that original data points were bisected. For set 1(c), the flow coefficient range was split into 61 linear intervals, giving 62 points per constant speed line. A summary of results is presented for the clean data sets.

5.6.1.1 "Clean" Data Set 1a

number of data points = 96

speed point peak errors, E_{Sn}:	+3.2% / -4.5%
peak average speed errors, S_{n95}:	+182 / -191 rpm
standard deviation of average error, sd_{bar}:	99 rpm
networks average correlation, r:	0.999011
over trained vs. total networks:	43/74

5.6.1.2 "Clean" Data Set 1b

number of data points = 184

speed point peak errors, E_{Sn}:	+1.5% / -1.9%
peak average speed errors, S_{n95}:	+170 / -149 rpm

standard deviation of average error, sd_{bar}:	72 rpm
networks average correlation, r:	0.999502
over trained vs. total networks:	11/42

5.6.1.3 "Clean" Data Set 1c

number of data points = 496	
speed point peak errors, E_{Sn}:	+1.3% / -1.1%
peak average speed errors, S_{n95}:	+125 / -97 rpm
standard deviation of average error, sd_{bar}:	41 rpm
networks average correlation, r:	0.999828
over trained vs. total networks:	6/37

5.6.2 Data set 2 - "Noisy" Data Set Results

5.6.2.1 "Noisy" Data Set 2 – full inputs

number of data points = 496	
speed point peak errors, E_{Sn}:	+1.4% / -1.5%
peak average speed errors, S_{n95}:	+160 / -161 rpm
standard deviation of average error, sd_{bar}:	58 rpm
networks average correlation, r:	0.999658
over trained vs. total networks:	2/33

5.6.2.2 "Noisy" Data Set 2 – input subsets

number of data points = 496	
speed point peak errors, E_{Sn}:	+1.1% / -1.4%
peak average speed errors, S_{n95}:	+158 / -165 rpm
standard deviation of average error, sd_{bar}:	59 rpm
networks average correlation, r:	0.999621
over trained vs. total networks:	1/32

Table 5.2, input subset combinations:

Input combinations	No. of networks
d.p., Pd, Td	23
d.p., Pd, Td, MW	4
d.p., Pd, Td, Ps	2
d.p., Pd, Td, Ts	1
d.p., Pd, Td, Ps, MW	1

Table 5.2: Input Subset Combinations

5.6.3 Data set 3 - "Random" Data Set Result

Four random data sets were generated. Each of the four sets included neural network inputs of suction pressure and temperature; discharge pressure and temperature; differential pressure and a training target of compressor speed. Set 3(a) and 3(b) contained 500 and 1000 data points respectively, and an additional input of molecular mass to represent gas characteristics. Set 3(c)contained 1000 data points and an additional 21 inputs, one for each of the gas components identified in AGA8, to represent gas characteristics. Set 3(d) contained 1000 data points and an additional 10 inputs, one for each gas component typically identified by an on-line gas chromatograph, to represent gas characteristics.

5.6.3.1 "Random" Data Set 3a

number of data points = 500

speed point peak errors, E_{Sn}:	+2.5% / -1.9%
peak average speed errors, S_{n95}:	+161 / -170 rpm
standard deviation of average error, sd_{bar}:	49 rpm
networks average correlation, r:	0.999663
over trained vs. total networks:	4/35

5.6.3.2 "Random" Data Set 3b

number of data points = 1000

speed point peak errors, E_{Sn}:	+2.0% / -2.4%
peak average speed errors, S_{n95}:	+194 / -218 rpm
standard deviation of average error, sd_{bar}:	40 rpm
networks average correlation, r:	0.999769
over trained vs. total networks:	1/32

5.6.3.3 "Random" Data Set 3c

number of data points = 1000

speed point peak errors, E_{Sn}:	+2.0% / -2.5%
peak average speed errors, S_{n95}:	+169 / -273 rpm
standard deviation of average error, sd_{bar}:	51 rpm
networks average correlation, r:	0.999655
over trained vs. total networks:	6/37

5.6.3.4 "Random" Data Set 3d

number of data points = 1000

speed point peak errors, E_{Sn}:	+1.8% / -2.6%
peak average speed errors, S_{n95}:	+173 / -228 rpm
standard deviation of average error, sd_{bar}:	47 rpm
networks average correlation, r:	0.999696
over trained vs. total networks:	11/42

5.6.4 Data set 4 - "Extrapolated" Data Set Results

5.6.4.1 "Extrapolated" Data Set 4a

number of data points = 496

speed point peak errors, E_{Sn}:	+1.6% / -4.8%
peak average speed errors, S_{n95}:	+137 / -629 rpm

standard deviation of average error, sd_{bar}:	176 rpm
networks average correlation, r:	0.995927
over trained vs. total networks:	2/33

5.6.5 Data set 5 - "Normalised" Data Set Results

Training was undertaken using only normalised flow and isentropic head coefficients. Only five networks were trained and the average training error exceeded 2000 rpm in each case, as shown in 5.6.5.1. Further training was undertaken with the normalised data set but the inputs were augmented with suction pressure, suction temperature, discharge temperature and molecular mass. Input subsets were enabled, similar to data set 2(b) and the results are reported in 5.6.5.2. Further training was undertaken with discharge pressure replacing head coefficient and results are reported in 5.6.5.3.

5.6.5.1 "Normalised" Data Set 5 – normalised inputs

Since only five networks were trained peak, mean and standard deviation of the network errors are tabulated. The data set contained the 96 points generated for data set 1c. The errors for normalised training data only are summarised in Table 5.3.

Network no.	Peak errors %	Mean error %	Error s.d. Rpm
1	+28 / -12	4.0	2187
2	+21 / -18	2.2	2197
3	+15 / -23	1.4	2189
4	+22 / -16	0.8	2195
5	+25 / -13	2.9	2185

Table 5.3: normalised input training errors

5.6.5.2 "Normalised" Data Set 5 – augmented inputs

number of data points = 96

speed point peak errors, E_{Sn}: +4.6% / -2.3%

peak average speed errors, S_{n95}: +426 / -324 rpm

standard deviation of average error, sd_{bar}: 122 rpm

networks average correlation, r: 0.998414

over trained vs. total networks: 9/40

Input combinations are summarised in table 5.4:

No. of Inputs	Input combinations	No. of networks	Average s.d. Rpm
6	ϕ, ψ, Td, Ts, Ps, MW	1	152
5	ϕ, ψ, Td, Ps, MW	1	138
5	ϕ, ψ, Td, Ps, Ts	3	128
4	ϕ, ψ, Td, Ps	3	113
4	ϕ, ψ, Td, Ts	1	95
4	ϕ, ψ, Td, MW	2	113
4	ψ, Td, MW, Ts	1	122
3	ϕ, ψ, Td	8	109
3	ψ, Td, Ps	1	142
3	ψ, Td, MW	1	137
2	ψ, Td	9	132

Table 5.4: normalised data – augmented inputs subsets

5.6.5.3 "Normalised" Data Set 5 – flow coefficient with augmented inputs

number of data points = 96

speed point peak errors, E_{Sn}: +3.3% / -2.6%

peak average speed errors, S_{n95}: +164 / -262 rpm

standard deviation of average error, sd_{bar}: 100 rpm

networks average correlation, r: 0.998987

over trained vs. total networks: 17/48

Input combinations are summarised in table 5.5:

No. of Inputs	Input combinations	No. of networks	Average s.d. Rpm
6	ϕ, Td, Ts, Ps, Pd MW	1	96
5	ϕ, Td, Ps, Pd, MW	1	87
5	ϕ, Ts, Td, Ps, Pd	4	77
4	ϕ, Td, Pd, Ps	2	110
4	ϕ, Pd, Td, Ts	3	94
4	ϕ, ψ, Td, MW	1	91
3	ϕ, Pd, MW	1	139
3	ϕ, Pd, Td	17	102
3	ϕ, Td	1	118

Table 5.5: normalised data – flow coefficient with augmented inputs subsets

5.6.5.4 "Normalised" Data Set 5 – flow coefficient, discharge pressure, discharge temperature input combination

number of data points = 96

speed point peak errors, E_{Sn}:	+2.8% / -1.9%
peak average speed errors, S_{n95}:	+166 / -198 rpm
standard deviation of average error, sd_{bar}:	75 rpm
networks average correlation, r:	0.999432
over trained vs. total networks:	16/47

5.7 Discussion

5.7.1 Developed Software vs. Commercial Packages

A spreadsheet neural network was developed to investigate the mechanisms associated with the feed forward neural network and error back propagation training algorithm. In addition it was hoped that this would be a low cost, flexible format comparable with commercial packages. At its current level of development the neural network does not match the performance of proprietary software. Changing the spreadsheet neural network architecture would require significant revision for every change, hence the flexibility requirement is not satisfied in its current form. The complimentary copy of the Trajan software was adopted as the principle research tool.

5.7.2 Basic Performance Data, fixed training set

It is clear from the initial research into network designs that compressor characteristics can be modelled using an MLP or an RBF type neural network. By keeping the data set split, (between training, verification and test sets), constant throughout it has been possible to identify a "best possible" error standard deviation of around 24 rpm. This has been achieved training an MLP

with "clean" data where speed values are known without error. The errors recorded are assumed to be a function of the procedures involved in designing and training a neural network. The "residual" error is probably due to differing degrees of error plane convergence for a fixed number of training epochs or allowable training tolerance. The differing degrees of convergence are probably attributable to the different sets of starting weights (these set the starting point on the error plane), which are randomly selected. The procedure used has identified a peculiarity with the neural network package which occurs if data is not "set" in a range and which can result in a less accurate network architecture.

The average number of hidden nodes increased from 6 to 9 when the network architecture was forced to six inputs from selected input subsets. This implies that a more complex network architecture is required to deal with higher dimension space through the addition of inputs. These observations are supported by background reading identified in Chapter 4.

5.7.3 Basic Performance Data, variable training set

When changing data set splits, with forced inputs remaining, was introduced the average hidden node count stayed at 9 but the average error standard deviation increased to 31 rpm. Using Root-Sum-Square (RSS) random uncertainty combination suggests that the uncertainty component introduced by changing data set split from one network to the next is around 20 rpm, almost double the fixed set error of 24 rpm. Increased error standard deviation is consistent with increased variance due to changes between training, verification and test subsets. Training set composition can have a significant effect on network performance even when the subsets are drawn from the same data "pool".

5.7.4 Variation in training set size – accuracy and over-training

In all cases, the training : verification : testing set split was 50% : 25% : 25%, the default associated with Trajan. This means that for a 200 point data set, 100 points would be used for training, 50 points for verification and 50 points for neural network testing. Cross verification is used to assess neural network performance in terms of whether or not over training has occurred. Over training is characterised by a divergence of training set error and verification set error and can occur when training set size is small. Complex neural network architecture may also result in over-training.

To record these phenomena incidence of over-training was recorded for each of the data sets as shown in section 5.6. A summary of incidence of over training in terms of training set size is shown in Table 5.6.

The table suggests non-linear decrease in incidence of over-training as the training set size approximately doubles. For the same basic data set, the addition of simulated noise further reduces incidence of over-training. For a random data set, equivalent to continuously varying inputs, the results suggest that an over-training incidence of 3 in 100 will be experienced.

Data set	Training set size (no. of points)	Peak error (%)	Incidence of Over-training (%)
Basic 1a	48	+3.2 / -4.5	58
Basic 1b	92	+1.5 / -1.9	26
Basic 1c	248	+1.3 / -1.1	16
Noisy 2 a	248	+1.4 / -1.5	6
Noisy 2b	248	+1.1 / -1.4	3
Random 3a	250	+2.5 / -1.9	11
Random 3b	500	+2.0 / -2.4	3
Random 3c	500	+2.0 / -2.5	16
Random 3d	500	+1.8 / -2.6	26
Extrap. 4	248	+1.6 / -4.8	6
Normalised 5 (aug. inputs)	48	+ 4.6 / -2.3	23
Normalised 5 (ϕ with aug. inputs)	48	+3.3 / -2.6	35
Normalised 5 (ϕ, Pd, Td)	48	+2.8 / -1.9	35

Table 5.6: Incidence of Over training vs. Training set size

Training set size does not appear to significantly influence accuracy of the neural network predictions above 250 points. As the number of inputs increase e.g. between sets 3(b) and 3(d) the accuracy performance is similar but the incidence of over-training increases.

5.7.5 *Inclusion of simulated input noise*

In a field implementation the input signals to the neural network will be measured process variables. The measurement will include random noise largely defined by the instrumentation uncertainty levels.

Comparing sets 1(c) and 2 (full input networks), typical instrument noise levels appear to marginally degrade accuracy and variance levels but greatly reduce incidence of over training, by approximately two thirds.

5.7.6 *Training on random operating points*

Training with random data (set 3(a), within the operating envelope) degraded peak error performance from around ± 1.5% to around ± 2.5% but with improved variance performance, from 58 rpm to 49 rpm. Doubling training set size to 1000 data points (set 3(b)) further improved variance performance to 40 rpm and reduced incidence of over training by around two thirds half.

Data set 3(c) and 3(d) used an extended gas properties inputs. Increasing the number of inputs increased the incidence of over training and degraded variance performance. It should be noted that this "degraded performance" can still estimate compressor speed to around ± 2.5%.

5.7.7 *Training with Input sub sets*

Set 2 was used to investigate the use of input subsets for neural network training. Overall training performance was not significantly different from the full input training performance. There is a marked bias, 23 networks out of 31 trained, towards an input subset of differential pressure, discharge pressure and discharge temperature. Discharge temperature is a function of pressure ratio and isentropic exponent as shown in equation 2.8. Isentropic exponent, used this research, is calculated as a regression function of molecular mass, suction pressure and suction temperature. The successful reduction of six inputs may

indicate that discharge temperature is sufficient to represent the gas property related inputs of molecular mass, suction pressure and suction temperature. It may be that discharge temperature could be used to represent live gas properties based on composition as opposed to molecular mass.

A further example of input subsets is described in section 5.7.9 where normalised data is augmented with discharge temperature.

5.7.8 Extrapolation from a restricted training set

The constant speed at which the peak –ve and +ve errors occur are shown in Figure 5.9 and Figure 5.10 respectively. The constant speed line is shown as the solid line with the training points. The neural network compressor speed estimate is shown by the broken line.

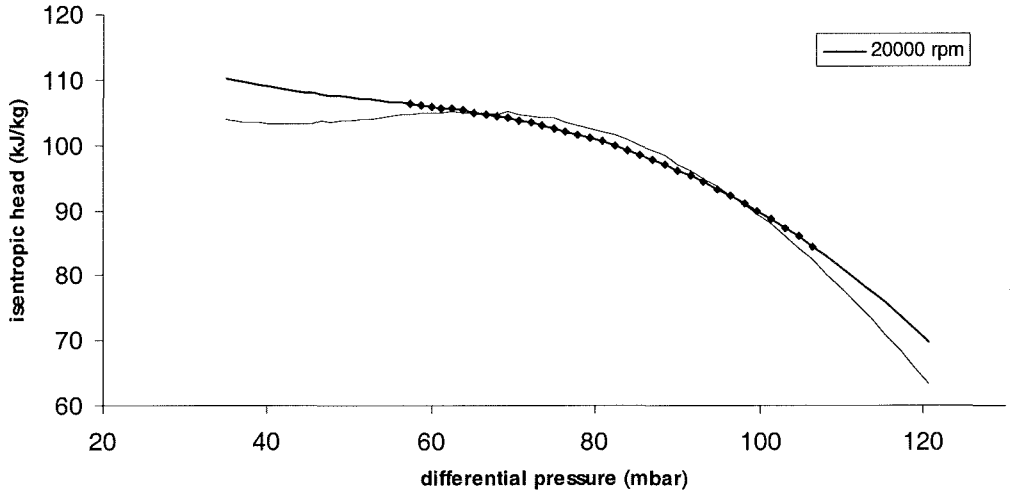

Figure 5.9: Peak negative extrapolation error

Peak negative error is –4.8% occurring at a training speed of 20000 rpm.

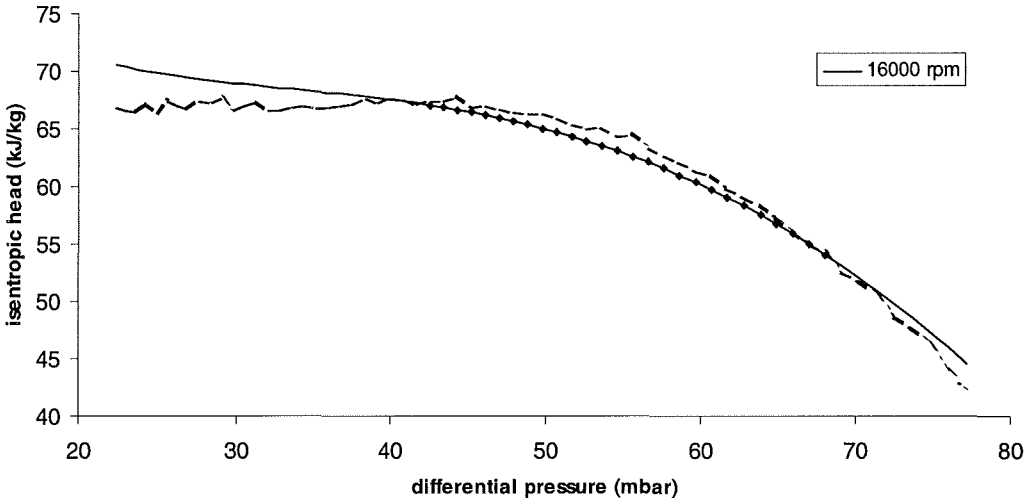

Figure 5.10: Peak positive extrapolation error

Peak positive error, of +1.5%, occurs at a training speed of 16000 rpm. Extrapolation is possible but could result in an asymmetrical, negative biased speed error of up to approximately 5%.

5.7.9 Training with normalised performance data

Training with only isentropic head and flow coefficients resulted in large errors in compressor speed estimates. Errors generated were one order of magnitude greater than neural networks when trained with absolute engineering units. The plot of the isentropic head vs. flow coefficient, the "universal speed" curve, is a multi-valued function in terms of compressor speed. It is possible to have several values of compressor speed at constant flow or isentropic head coefficient value. The universal speed curve and associated neural network estimated speeds are plotted for one neural network trained on normalised data is shown in Figure 5.11.

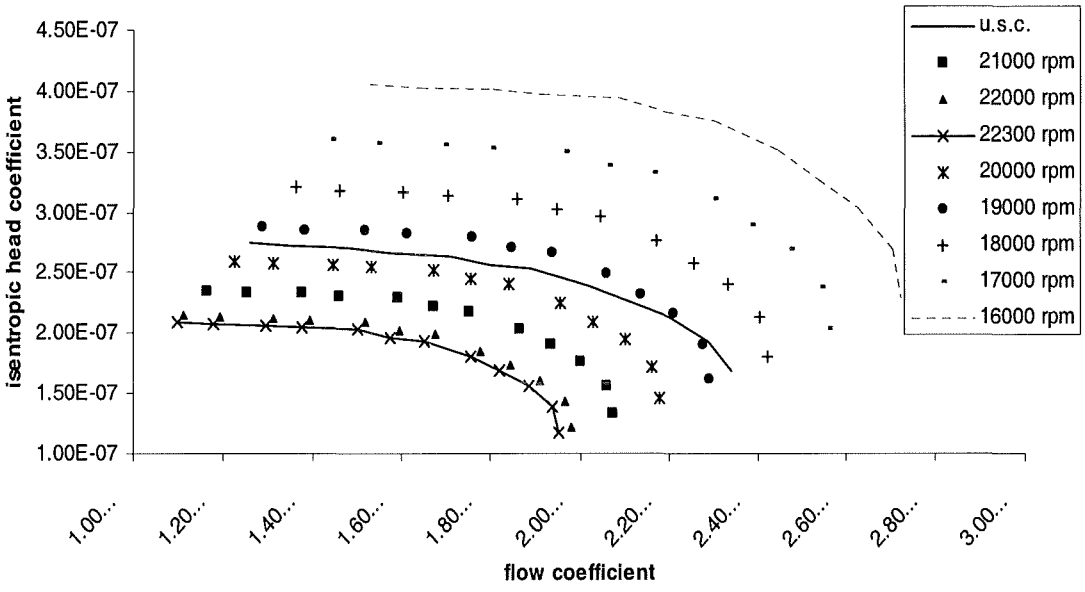

Figure 5.11: Normalised data neural network performance

The neural network creates a separate curve for each compressor speed where one, universal curve is expected. The neural network can represent the shape of the curve but cannot locate it in vector space. Plots of compressor speed, estimated from normalised data augmented with discharge temperature are shown in Figure 5.12 and Figure 5.13.

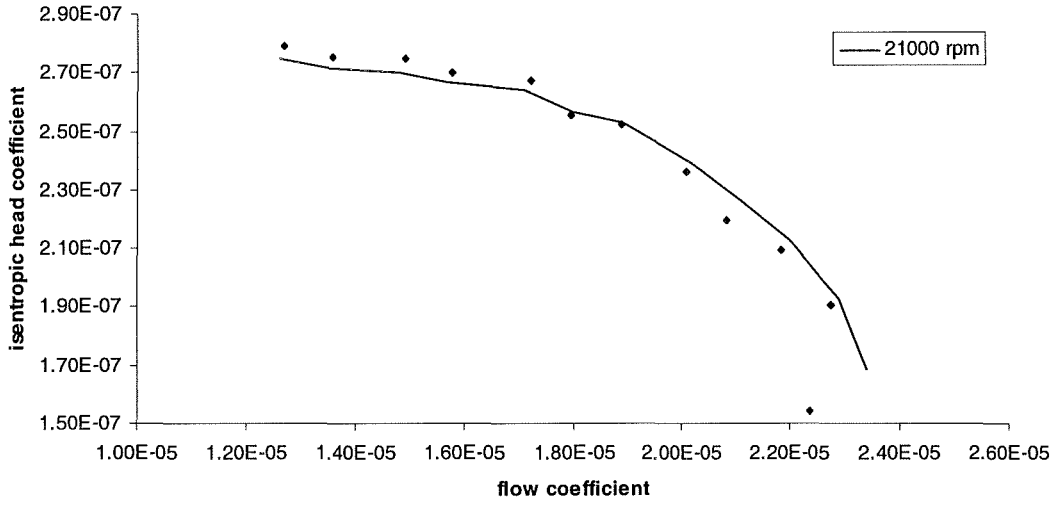

Figure 5.12: Td Augmented Input, peak –ve error

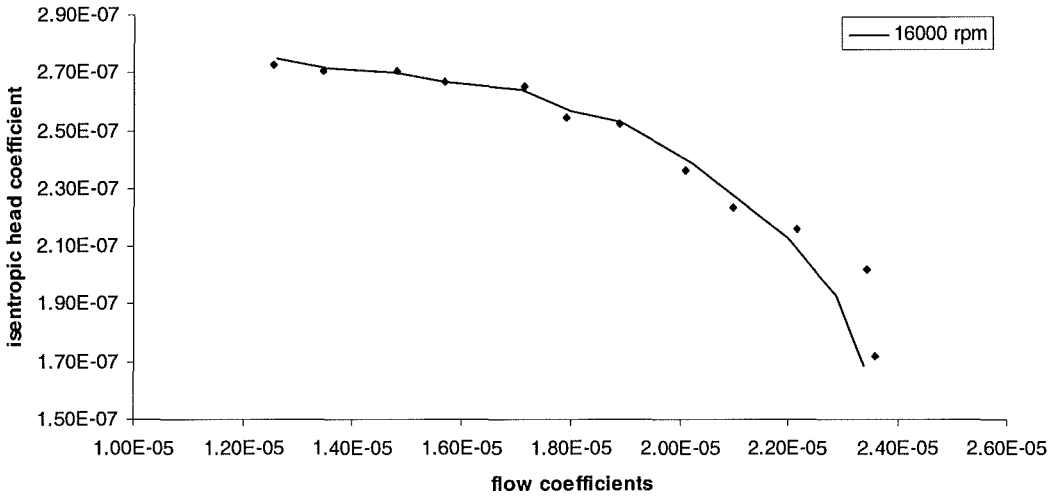

Figure 5.13: Td Augmented Input, peak +ve error

Normalised inputs of isentropic head and flow coefficient can be used to represent the universal speed curve when an augmenting input such as discharge temperature is added.

Further testing was undertaken with flow coefficient as the only normalised input. Compressor speed estimates were marginally improved, but frequency of over training increased by around 50%. More than 50% of the networks trained produced an input subset of flow coefficient, discharge pressure and discharge temperature. Training with the combination of flow coefficient, discharge pressure and discharge temperature was undertaken. Compressor speed estimates were within ± 3% with an average error standard deviation of 75 r.p.m. Levels of over-training were consistent with reduced training set size at around 35%.

5.8 Conclusion

The results of the testing are clearly indicative that neural networks can "learn" compressor head characteristics. The most rigorous test case, comprising a random data set of operating points, suggest that compressor speed can be estimated to within ± 2.5% of the speed required to meet a given flow rate for a given discharge pressure constraint.

The "best network" design returned the lowest training error in each of the thirty-one tests conducted for each data set. Performance is related to training algorithm employed as opposed to network architecture. Neural network architecture is overwhelmingly biased toward multilayer perceptron, feed forward networks. This was found in the Initial Research phase and during the numerical testing.

- Results of testing suggest that a data set size of 500 points can train a neural network to estimate compressor speed to ± 2.5%, table 5.5.
- The use of gas components as inputs in place of molecular mass fall within this error band, but increase incidence of over-training, possibly related to more complex network architectures.

- Extrapolation from the middle third of the flow coefficient range is possible but the error band increases to ± 5% and the error distribution is biased toward underestimating compressor speed, figures 5.9and 5.10. This type of error would tend to move the compressor toward surge for a fixed discharge pressure constraint.
- The use of normalised flow and head coefficient data for training neural networks is possible but the speed curves are reversed as shown in figure 5.11. An augmenting input is required to locate the universal performance curve in absolute vector space, sections 5.6.5.2 and 5.7.9. Use of head and flow coefficients augmented with discharge temperature, as inputs, accounted for 26% of neural network "best network" design. Use of head coefficient and discharge temperature accounted for 29%, Table 5.4. Error levels are similar to those reported for the actual number of training points, 48 in number. Augmented input performance is shown in figures 5.12 and 5.13.
- It may be possible to reduce the instrumentation associated with compressor control to three: flow rate measurement; discharge pressure and discharge temperature. Training using flow coefficient, discharge pressure and discharge temperature, section 5.6.5.4, indicate compressor speed can be estimated to within ± 3% using these measurements alone.

The research, testing and results described in this chapter are novel and form the basis of a novel performance control scheme for compressors. The learning capability implies that changing performance characteristics can be accommodated without specific expert intervention. The performance control set point can be expressed in absolute engineering units without the need for translation to and from non-dimensional parameter domains.

A novel performance control system, based on absolute operating point, is described in chapter 7.

6 Control Systems Paradigms

6.1 Introduction

The work described in the previous chapters is drawn together to form a proposal for a performance control algorithm. The algorithm might be implemented in a micro-controller or a higher-level Supervisory, Control and Data Acquisition (SCADA) system. In developing this approach, the role of neuro-control and conventional control schemes are outlined. Review of conventional control schemes is included to introduce the various "terms" used in describing control systems, both conventional and intelligent.

6.2 Conventional Control Paradigms

6.2.1 On/Off Control Action

On/off control is the most basic form of control. This type of control is generally used where the energy input/output into the system is small in comparison with the system inertia [74]. An example of a system of this type is a domestic central heating system. The thermal inertia of the house is large in comparison to the heat input from the boiler. The thermostat responds to a falling set point crossing by switching the heating on and a rising set point crossing by switching the heating off. The control action is determined by the sense, falling or rising, of the difference between the thermostat setting and the actual room temperature. The control action does not respond to the magnitude of the difference, or error, between the two.

6.2.2 Three Term (Mode) Control Action

Controlling action is determined by the magnitude of the difference between the set point and the measured variable, also known as the error. Three types of

controlling action and the resulting combination of these actions are described.

6.2.2.1 Proportional Control Action

The basic controlling action is *proportional* action where the response of the control system is proportional to the magnitude of the error. This is expressed as [75]:

$$m = \frac{100}{P} e + b \tag{6. 1}$$

where m is the controlling action

e is the difference between set point (desired) value and the measured (actual) value of the controlled variable

P is the proportional band

b is the bias

The proportional band is defined as the change in controlled variable output for the change in measured input variable. A 20% proportional band implies a full-scale change (100%) in output span for a 20% change in input span. The steady state gain of the system is the inverse of the proportional band fraction. For the example described the steady state gain is 5. The steady state gain of the system is constant for linear systems but may vary for non-linear system control.

When the error is zero the output of the controller is the bias term, b. Whilst a system is balanced when the error is zero, there can be some difference between b and the live value of the controlled variable, or process load. This difference is termed proportional offset. Low steady state gain can result in unacceptable offset whilst high steady sate gain, to reduce offset, can result in system cycling or instability. The bias term can be adjusted to set the controller output to match the process load. Manual adjustment of the bias is termed

manual reset. Reset action can be automated as a function of error using an integral control action term.

6.2.2.2 Integral Control (Reset) Action

Integral control action is generally used in conjunction with proportional action. The proportional-integral, PI, control action is described by:

$$m = \frac{100}{P}\left(e + \frac{1}{I}\int e\,dt\right) \qquad (6.2)$$

where I is the integral (reset) time

The addition of integral action slows the response of the proportional controller. Smaller values of integral time speed up the response of a PI control action whilst longer integral times slow the PI action.

Integral or reset windup can occur where the error cannot be eliminated over a time period. This may occur in a batch process where the error exists between batches. The integral action attempts to reduce the error driving the controller output to maximum. Once the error is removed, when the batch process starts up, the controller error output stays at maximum for a time after the start even when the error has reversed sense. Different *types* of controller may inhibit integral action when a preset control action level is reached.

6.2.2.3 Derivative Control Action

The slowing effect of integral action can be compensated for by the addition of derivative action. Derivative action can also speedup the response of proportional action. This alone cannot be used to maintain a set point, hence it is added to proportional action or proportional-integral action. Proportional-derivative, PD, control action is described by:

$$m = \frac{100}{P}\left(e + D\frac{de}{dt}\right) \tag{6. 3}$$

where D is the derivative time.

The faster the rate of change of error is, the larger the response of the controlling action. The higher the derivative time, the greater the controlling action response will be.

6.2.2.4 Proportional, Integral & Derivative (PID) Control Action

Combined proportional, integral and derivative action is described by:

$$m = \frac{100}{P}\left(e + \frac{1}{I}\int e\,dt + D\frac{de}{dt}\right) \tag{6. 4}$$

PID controllers require tuning for a specific process to balance the controlling action components. Figure 6.1 shows the control action response to a unit step process upset.

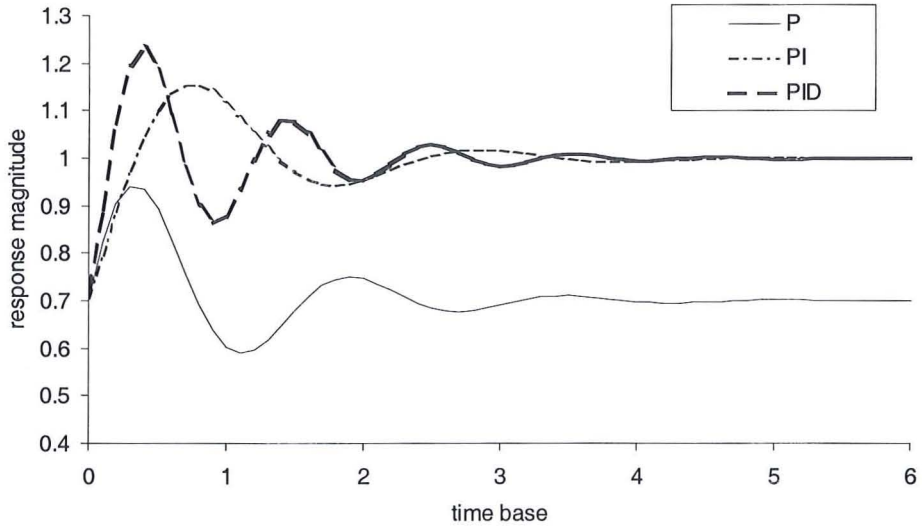

Figure 6.1: Controlling Actions

The proportional response, P, displays the offset that is corrected with the addition of integral action, PI. The speed of response is improved with the addition of derivative action, PID.

6.2.2.5 Types of Basic Control

Most common types of controller are : Batch; Auto Select; Ratio and Cascade.

Batch Control

Batch control is used in discontinuous processes where product batches can be run at intervals. Between batches the control system can be open loop i.e. batch tanks are empty so no feedback signal completes the control loop. In this situation, integral wind up may occur where integral action is in use. When the measurement is within 0-100% the controller applies standard P or PI control action. If the measurement goes out with these limits, i.e. the system is open loop and the standard control mode cannot restore measurement to the set point, the batch controller invokes a "batch switch" mode. In this mode, the integral action is inhibited to prevent controller windup.

Autoselect control

Autoselect control, also known as Selective control, monitors several variables in a process and manipulates one variable to keep all measured variables within safe limits. Each measured variable has a dedicated controller whose output goes to a selector. The selector compares the measured variable control signals and selects the lowest (or highest) to assume control of the controlled variable. A typical Autoselect control system may be found on a pump station. Measured variables will typically be pump suction pressure, motor load and discharge pressure. The control variable will be a Flow Control Valve (FCV) on the pump discharge. Each of the measured variable controller outputs are monitored. If the pump suction pressure is lowest, control of the FCV will pass to the pump suction pressure controller to prevent cavitation in the pump. If the motor load rises the controller signal will assume control of the

FCV to prevent the motor overheating. If the discharge pressure rises above safe pipeline pressure limits, the discharge pressure controller will assume control of the FCV. The controller outputs are compared to their own respective set points rather than to the value of measured variable.

Ratio Control

Ratio control may be used in a blending process to blend to mix fluids in constant proportions or to control fuel/air mixture in a combustion process. In a combustion process the measured variable would be the fuel flow rate, set by a governor or throttle, and the controlled variable would be the airflow rate. The ratio control would regulate the airflow to ensure stoichiometric combustion as the fuel flow rate varies.

Cascade Control

Cascade control controls a secondary or slave variable using a set point generated by a primary or master variable. An example of this is where the temperature of a heated product is controlled by varying the fuel rate to a heater. The primary variable is product temperature. The output of the temperature controller is input to a secondary controller as its set point. The secondary controller is a fuel flow rate controller whose output controls a fuel-regulating valve. Thus the fuel flow is indirectly controlled by the product temperature. This type of control is useful where fuel supply pressure is unsteady or where the primary variable responds slowly but with high amplitude to variation in the secondary variable (fuel supply).

6.2.3 Adaptive Control

Advanced control techniques usually specify some objective function as the control system objective, rather than regulation of a system about a set point. Control to a set point is still a requirement of an advanced system and this may well be implemented using a PID controller for the controlled variable.

An adaptive control system automatically updates its parameters, such as gain, to compensate for changes in the properties of the process. The control system "adapts" to the process. Specifying the value of controlled variable, or set point, is not sufficient in itself. An "objective function" of the controlled variable is also specified. The objective function determines the form of adaption required. Adaptive systems may be categorised as *dynamic* adaptive, or *steady state* adaptive. In a dynamic adaptive system the objective function is designed to control a dynamic parameter of the system, such as damping. Similarly, in a steady state adaptive system the objective function is designed to control a steady state parameter such as steady state gain.

A further distinction is made in terms of the adaption mechanism [75]. If the process to be controlled is sufficiently well defined, the adaption can be *programmed* in terms of the process variables. That is, the control system parameters can be estimated from the programme or model.

Where the control system parameters are changed through changes in the measured value of the objective function, then adaption is effected through a feedback loop. This is referred to as *self-adaption*, or as a *self- adaptive* system.

6.2.3.1 Dynamic Adaptive Control

Dynamic loop gain, the net effect of integral and derivative time dependent controller gain, is generally the objective function of the controlled variable in a dynamic adaptive system. This parameter is most effective in maintaining the stability of a control loop. Changes in steady state gain are usually classified as system non-linearity. Non-linearity, from this source, can be compensated through gain scheduling, Hagglund [76]. Variable process gain is compensated for by introducing non-linearity in the controller. This may be done through inclusion of a non-linear characteristic of a control valve in the control loop.

Shinskey [75], describes an example where setting the dynamic gain of a control loop (objective function) is dependent on a variable out with the control

loop. The adaption is programmed from knowledge of the process and the dynamic loop gain is set as a function of the measured variable, out with the control loop. Figure 6.2 shows a heat exchanger control loop that exhibits capacity and dead time.

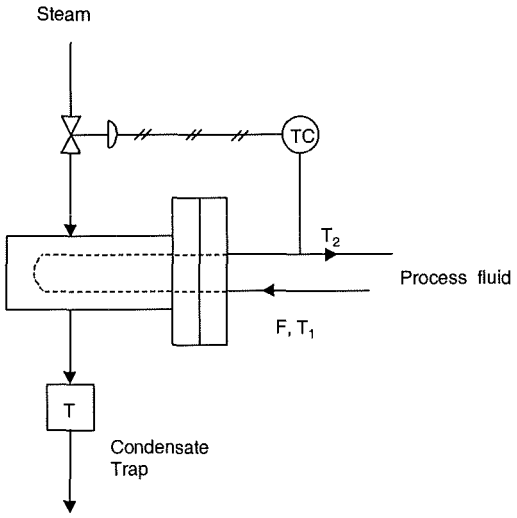

Figure 6.2: Heat Exchanger loop

The dynamic loop gain changes with the inverse of process fluid flow rate. Hence, a programmed adaptive control algorithm can be written as:

$$m = \frac{100\,f}{P}\left(e + \frac{f}{I}\int e\,dt + \frac{D}{f}\frac{de}{dt}\right) \qquad (6.5)$$

where f is the fractional flow of F

With *a priori* knowledge of the process an operator could adjust the proportional band, integral time and derivative time to achieve the desired control effect. However this can be implemented in an adaptive controller with an additional input of the measured variable, f. Hence the dynamic gain is automatically adapted to changes in flow rate , f, to improve control loop stability. Potential error and consequent loop instability can be avoided through

automation in the controller.

6.2.3.2 Dynamic, Self Adaptive Control

A self-adaptive controller is attributed with the capability of identifying when adaption is required and knowing the adapted value of controller parameters to use. These attributes are necessary where no *a priori* knowledge of the process is available or where the objective function is too complex to programme in a loop controller.

The knowledge of loop gain and damping is termed *identification* in control terms. System identification can only be deduced during perturbations from the process steady state. A perturbation can be induced by the controller periodically changing a controller parameter to upset the system, or by a naturally occurring process disturbance. Identifying system parameters from a naturally occurring disturbance requires the disturbance to occur before corrective control action can be implemented. The self-adaptive controller is reactive to process disturbances and does not predict of disturbances.

Self-adaptive controllers could relieve operators of the need to adjust controller parameters. The performance of critical control loops could be made independent of the skill level of the operator. Given the inherently unstable nature of system identification, for parameter adaption, self-adaptive control is used only where no satisfactory alternative is available.

6.2.3.3 Steady State Adaptive Control

Steady state adaptive control is concerned with finding a constant value of steady state gain required for process equilibrium. This implies that for a given process, steady state gain is variable and that a single value satisfies the process equilibrium.

This can be illustrated using fuel combustion control as an example [75]. In combustion, ratio control is used to control the proportions of fuel to air. For a

given set of conditions only one fuel/air ratio will ensure efficient combustion. The controlled variable is fuel efficiency, the manipulated variable is fuel/air ratio. Too much air results in a lean fuel mixture, too little air results in a rich fuel mixture. Either mixture results in inefficient, or sub-optimal, combustion. In the case of fuel combustion the objective function can be programmed into a controller by differentiating the efficiency parabolic equation and setting the derivative to zero. The action of an adaptive controller in seeking out the single value of steady state gain required is termed *optimisation*. Optimisation is a special case of steady state adaption.

No feedback is required in the control loop since the operating point is mapped by a combustion efficiency equation. Combustion control of this type is categorised as *feed forward* control.

6.2.4 Feed forward control

Feed forward control allows a control system to adjust to process disturbances before a measurable set point deviation would initiate controlling action. A feedback controller can only respond as its parameters allow and these are largely governed by the dynamics of the process it tries to control. A feed forward controller reacts to process disturbances immediately, on the basis of a computed control action. The control action is computed from key measured process variables for a desired operating point. A manipulated variable is used to bring the process to the stable, predicted operating point for a given set of process conditions.

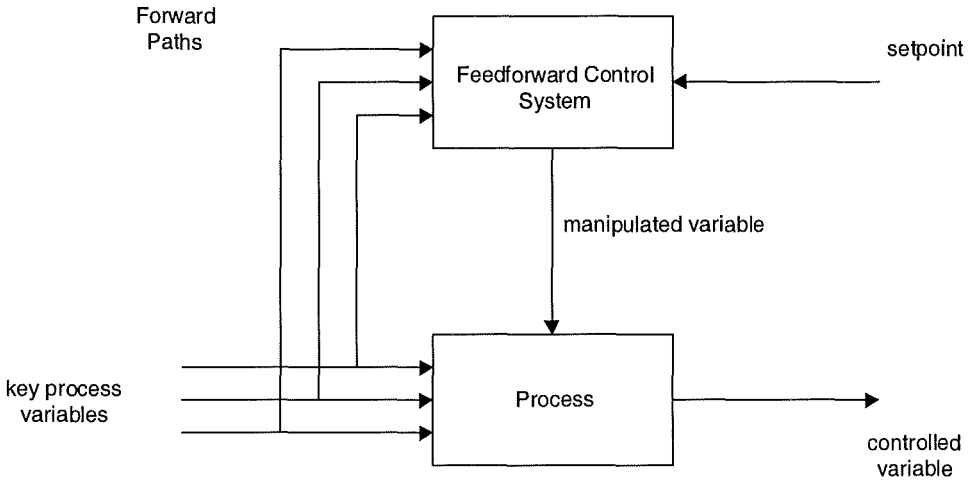

Figure 6.3: Feed forward Control Schematic

Figure 6.3 shows a schematic of a feed forward control scheme. No measurement of the controlled variable is taken to derive a set point error. The value of manipulated variable is estimated, from the process *model*, to give the desired process operating point (set point) for the prevailing, key process variables. Ideally the model will combine steady state characteristics with dynamic characteristics to cover transients between operating points. Many process models are made up of systems of linear differential equations. Model based control is also known as Model Predictive control (MPC). MPC based control systems are likely to require periodic "maintenance" by expert modellers, plant operators and instrument technicians.

Feed forward control is theoretically capable of perfect control, however causes of error or other limitations include:

- Accuracy of measurement of the key process variables. Inaccuracy may come about through poor specification of instruments, mis-calibration of instruments or instrument failure.

- Errors in the computing components. Software bugs, inadequate number resolution.
- Failure of the model to represent the characteristics of the process. Process characteristics changing with time, novel operating conditions.
- omission of key parameters from input to the model. Incomplete specification of the process key parameters.

6.3 Intelligent Control

6.3.1 Overview

A framework for intelligent control is suggested by Astrom & McAvoy in [77]. This is shown in Figure 6.4.

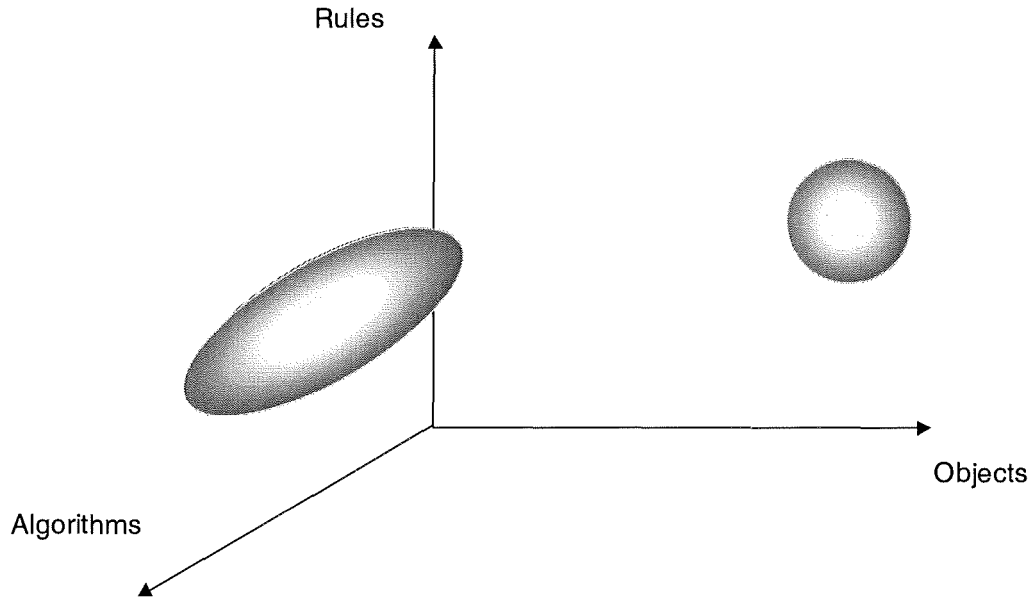

Figure 6.4: Framework for Intelligent Control

An intelligent control system made up from an expert system can only be implemented using "If-Then" type rules. As the complexity of the system increases the number of rules increases to the point where the system may become unwieldy and the software un-maintainable [78]. Identifying a number of control objects, each of which have a discrete set of rules, improves the "maintainability" of the system. The rules are confined to a discrete object and can be updated, added to or taken from in a controlled context. An expert system of this type would belong in the "rules-object" plane of the diagram.

Adding a neural network (an algorithmic function) to the expert system would move the system into the 3 dimensional control space. Neural networks generally require objects and rules that facilitate analysis and interpretation of the neural network output, such as in the case of character recognition.

An intelligent controller requires an expanded algorithmic axis to accommodate linear, non-linear, adaptive or model based control algorithms. The control space is then 6 dimensional. A control system design dilemma arises, is it better to have an algorithmic or heuristic control strategy?

An algorithmic controller may be a PID controller in control space and a heuristic controller may be a fuzzy controller in the rule - object plane, both regions depicted in Figure 6.4.

Digital controllers can be programmed with PID algorithms, fuzzy rules sets and computational implementations of neural networks. Modern controllers have these available as software library functions. Recent trends in intelligent controller have included a basic fuzzy controller for guidance of an unmanned boat [79]. A learning, fuzzy controller has been reported [80] which allows control adaption through changes to fuzzy membership function and rule modification. Techniques for intelligent controller design are reported in [81] and [82].

6.3.2 *Knowledge Based Systems*

Knowledge based systems use the entire control space shown in Figure 6.4. Microprocessor based instruments, such as Programmable Logic Controllers or Digital Controllers, contain the software to implement algorithmic control and heuristic control made up from logical functions. A knowledge based controller architecture is shown in Figure 6.5 [78]. The system shows a simple controller and process feedback loop. Co-ordination of each of the components is carried out by the knowledge-based system. Identification algorithms can test the process for parameter identification. Supervision algorithms monitor the controller performance and initiate loop tuning if required. The controller has the capability to implement the most appropriate control algorithm for the process based on supervision and identification activities. The operator has access to the controller through the knowledge-based system.

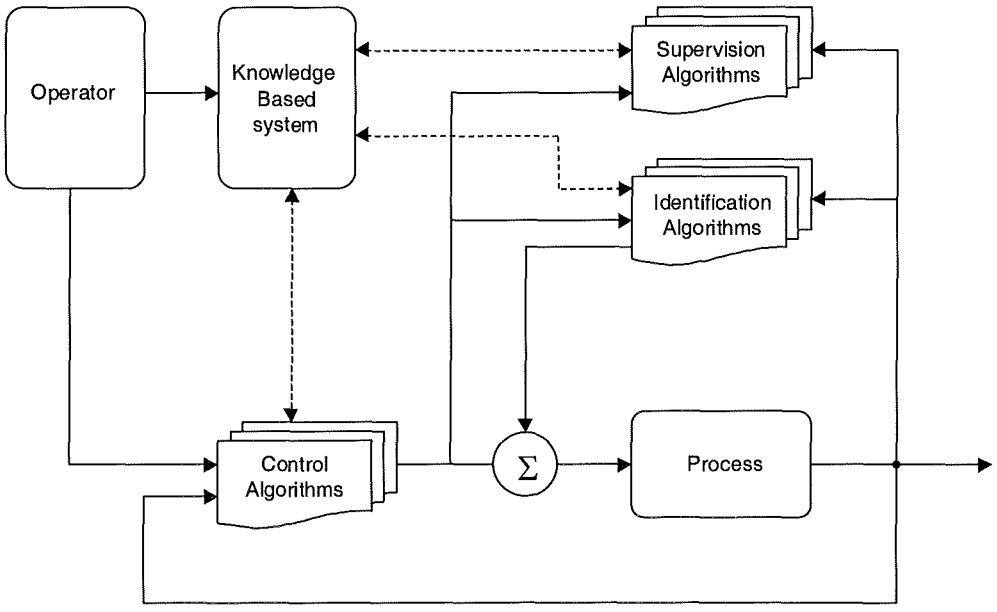

Figure 6.5: A Knowledge Based Control System Architecture [78]

A knowledge based system will, typically, have the components identified in Figure 6.6 [83]. The "Intelligent Program" provides clear separation between the knowledge that the sytem uses and the program that uses the knowledge for problem solving. The "Knowledge Base" contains the knowledge gathered by the knowledge engineer from various sources. Knowledge information is interpreted by the "Inference Engine" to derive additonal data and conclusions.

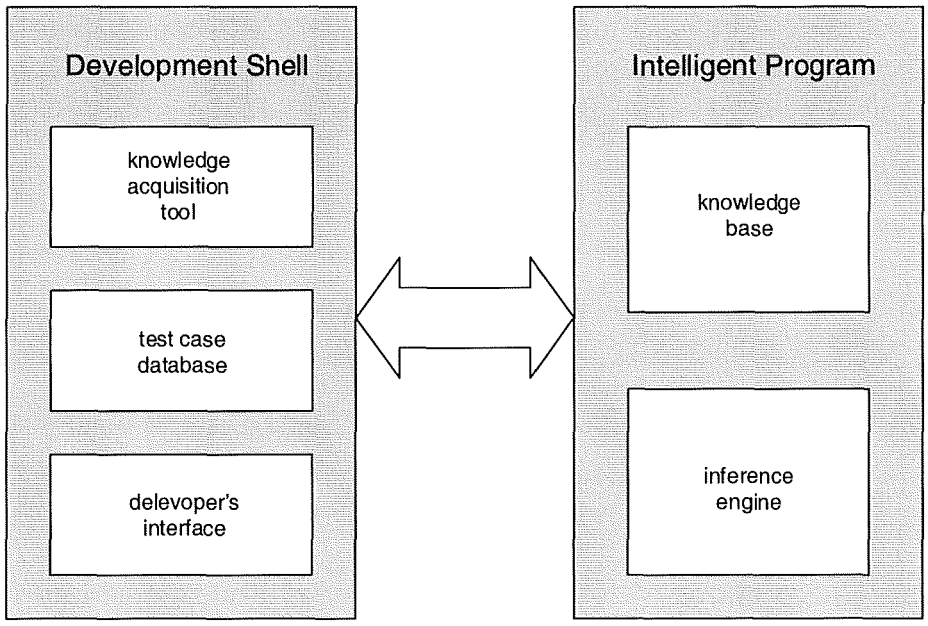

Figure 6.6: Knowledge Based System Components [83]

A development shell allows the knowledge engineer to structure, de-bug and modify expert knowledge. The "Knowledge Acquistion Tool" allows structuring of the knowledge and may be a relational database. The "Test Case Database" is a collection of problems which have been successfully executed on the knowledge system. This can be used to confirm changes to the knowledge base have not compromised its performance. The "Developer's Interface" is the same as the end user's interface but with additional development features.

Knowledge Based Systems are distinguished from conventional algorithmic and search based systems by three fundamental concepts. These are the separation of knowledge from how it is used; the use of highly specific domain knowledge and the heuristic, rather than algorithmic, nature of the knowledge.

A hierarchical knowledge based controller, capable of generating its own Knowledge Base, is described in [84]. A knowledged based system for gas turbine maintenance diagnosis is described by Travé-Massuyès & Milne [85].

6.3.3 Neuro control

Neurocontrol came to the fore in the late 1980s through improved training of neural networks using error backpropogation techniques. Early reviews included Antsaklis [86] with more recent reviews by Armitage [87]. Werbos [88], identifies neurocontrol as a subset of traditional control theory and neural network research, as shown in Figure 6.7.

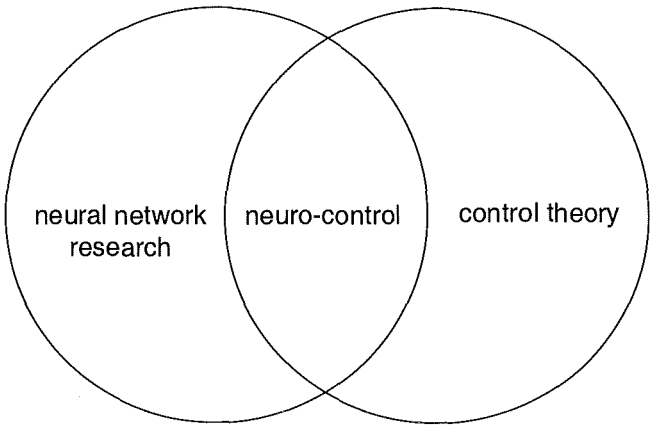

Figure 6.7: Neuro-control subset

Three basic types of neurocontrol are identified: cloning, tracking and optimisation. Agarwal [89] suggests a systematic structure to neurocontrol classification. For the purposes of this research the simpler three-category taxonomy is sufficient to explain the categories of neurocontrol.

6.3.3.1 Cloning Control

Neural cloning systems mimic the relationship, or mapping, between system inputs and outputs exhibited by human experts or automatic controllers. A distinction is drawn, [88], between a fuzzy 'clone' control system implementing what experts *say*, through its rule base, and a neurocontroller implementing what experts *do*. Cloning control is the basis of an expert control system.

A "truck backer upper" is described b Nguyen & Widrow [90] that learns and improves loci of backing up a truck from any point in a confined vector space. Bhat et al [91] describes a chemical process model using neural networks that learns non-linear process characteristics from process inputs and outputs. Process control using reinforcement learning of a neural network is described by Hoskins & Himmelblau [92]. The method of reinforcement does not use an objective function, instead using a *critic* to deem a given control action "acceptable" or "unacceptable". This method mimics human learning by trial and error. Operator skill is embedded into neural models, at various levels of control hierarchy, of a steel plant, Bloch et al [93].

6.3.3.2 Tracking control

Conventional adaptive control, described in sections 6.2.3 and 6.2.4, is concerned with control systems that maintain a set point or *track* a reference model (MPC). Neural adaptive control offers generalised non-linearity and the ability to learn adaption parameters, as opposed to programmed (or self adaptive) non-linearity and parameter identification.

An early example of "backpropagation through time" is described by Bhat and McAvoy [94]. The subject is dynamic control of a continuously stirred tank reactor. In this system the inputs are a moving time window of pH and corresponding instantaneous values of a reactant. On each training step the time window is moved forward one time step. The outputs are time increments of mixture pH. Hence a neural network is trained to predict future values of

reactor pH. This network is used as a model to control the reactant flow rate to achieve a desired mixture pH. The future value of pH is optimised by using a value of reactant that minimises a pH error squared function for the first predicted time step.

Chen [95] describes non-linear, self-tuning adaption control for a Single Input/Single Output (SISO) plant simulation. Control action is generated by an MLP neural network. Online back propagation training, of the controller neural network, is turned on and off as required by an error function. The plant output is tracked by the controller after 10^5 time-steps. Willis et al [96] describes a Multiple Input/Multiple Output (MIMO) neural network predictive control scheme for a distillation column. A study of neural network based non-linear dynamic controllers, with respect to stability, has been presented by Hernandez & Arkin [97]. It is concluded that a non-linear process can be stabilised by a neural network model based controller model. Park et al [98] describes a design of optimal neurocontroller that can control the motion of a cart to keep an inverted pendulum balanced at an angle to the vertical. Fabri & Kadirkamanathan [99] presents a method of activating Gaussian RBF nodes in the vicinity of controlled variable operating point. The network architecture is thus dynamic. This allows use of an "economic" network in terms of size for subsets of state space. Horn [100] describes a method of feedback linearisation using neural process models with significant improvements over conventional controllers.

6.3.3.3 Optimisation Control

The learning ability of neural networks is suited to optimisation of an objective or utility function over time. the technique of "back propagation through time" [101] is suited to optimisation type control. Neurocontrol techniques are also used to enhance conventional optimisation techniques such as calculus of variations, model predictive control and approximate dynamic

programming.

A description of an optimising controller has been described by Joubert et al [102]. The controller is implemented on a SCADA computer system and uses two neural networks. One network is trained continuously on-line and the other is used as a plant model (active network) for predicting controlled variable set point. If the prediction causes > 2% error in the desired operating point the training network replaces the active network. The controller optimises for maximum revenue, in terms of product spectrum, or maximise principle product yield from available input streams. El-Sayed [103] proposed a similar system of on line training for steam power electricity generation. Kleymenov et al [104] describes temperature compensation for humidity measurement device calibration using a neural network trained using genetic algorithms.

6.3.4 Expert Systems

An expert system is programmed with mathematical and heuristic knowledge from an expert in a particular field. The type of knowledge represented is experts "know how" [105]. An expert system does not attempt to reason, as a knowledge-based system may, but is limited to the domain of the particular expertise encapsulated in the system. The system may adapt within these constraints, [78]. An expert system for pipeline scheduling is described by Seskin, [106]. Early process control expert system architecture is described in Moore et al, [107]. An expert system for multiphase flow rate measurement has been developed by Toral, [108].

6.4 Discussion

This chapter has reviewed principle control paradigms at a level that allows identification of control scheme from a characterisation of each. A search of control engineering sources has identified past and recent trends in neurocontrol. The search has by no means been exhaustive such is the scope of

control engineering. Knowledge based and Expert systems have been included as neural networks are often found in this type of control system.

Very little was found regarding the control of simple rotating machinery. This is believed to be due to the lack of advanced control application where conventional control is suffice vis. section 6.2.2.5 and the "Autoselect" control of a pump station. A similar point is made in [106] where attitudes toward control, in manufacturing industries, tend to favour the traditional approach, which does "well enough". "Autoselect" control for example, whilst considering equipment protection, does not consider optimising prime mover energy consumption or associated exhaust emissions.

6.5 Conclusions

There are established control paradigms for the use of neural networks in control systems. These are generally in the area of adaptive or model predictive control. A more obscure application is in the use of neural networks for feedback linearisation for example. Neural modelling is increasingly used in complex measurement devices demonstrating the accuracy possible when good training data is available, [108].

It has been demonstrated, in Chapter 5, that neural networks can accurately represent compressor characteristics. Using a neural network in a model based controller forms the basis of the compressor performance controller described in the next chapter.

7 Compressor Performance Control Scheme

7.1 Review of Existing Compressor Control Actions

As described in Chapter 2, the "state of the art" compressor control, considered as a whole, fits into the programmed, adaptive control category. The surge line characteristic is programmed into a controller. Manipulation of the controlled variable is based on several measured process parameters. The objective function is to prevent the compressor from entering the surge region of operation.

On closer scrutiny there are several types of controlling action, described in Chapter 6, in the overall control function. The controlling element, usually a progressive action anti-surge valve, is controlled using a PID control algorithm. A rapid opening valve may be fitted in parallel with the progressive action anti-surge valve. Where this is the case the rapid action valve is operated with an on-off or open-close action. When load sharing the worst case, or "closest to surge", controlled variable is selected as the primary controlling variable, a feature of auto-select control.

7.2 Novel Compressor Performance Control

7.2.1 Re-statement of Principle Research Objective

In gas distribution systems, control of compressor throughput is critical in ensuring that contracted gas quantities, known as nominations, are delivered. Failure to meet contractual obligations in delivered gas quantities can result in commercial loss and penalties. Optimising compressor operating point, to ensure required throughput will result in contract fulfilment, can be operator intensive or require expensive optimising, modelling packages.

The principle objective of performance control, for this research, is to use neural networks to optimise compressor operating point (COP). In doing this it is necessary for individual compressors to operate at different points relative to their surge line and to deliver a desired through put for any given set of process condition.

This can be achieved through adjustment of compressor speed, in absolute vector space, thus setting COP. Chapter 5 has shown that neural networks can model compressor characteristics for changes in conditions of suction pressure, suction temperature and gas composition (molecular mass). The model can be "interrogated" to give a required compressor speed (controlled variable) for a desired flow rate given a discharge pressure constraint. Chapter 6 describes adaptive systems based on reference models and feed forward control.

7.2.2 Steady State Adaptive Control

Steady state gain can have many values for prevailing process parameters suction & discharge. The steady state gain of the system is compressor flow rate for the given speed i.e. acmh/rpm[1]. For prevailing suction conditions, molecular weight and discharge pressure constraint only one compressor speed will satisfy the flow rate requirement. The proposed control scheme selects compressor speed or optimises steady state gain to achieve the objective function i.e. compressor flow rate.

Inputs to the systems are shown in Figure 7.1. The set point is the required flow rate. The following parameters are used to train the neural network:

- measured flow rate
- measured discharge pressure
- measured suction pressure

---

[1] (actual cubic metres per hour) per (revolutions per minute)

- measured discharge temperature
- measured suction temperature
- molecular mass

Measured flow rate is also used to generate a flow rate discrepancy event. The neural network model outputs a recommended compressor speed to achieve the required flow rate for the given process load. The compressor speed can be translated into a speed set point for the prime mover or left as an advisory setting for operator acceptance and downloading

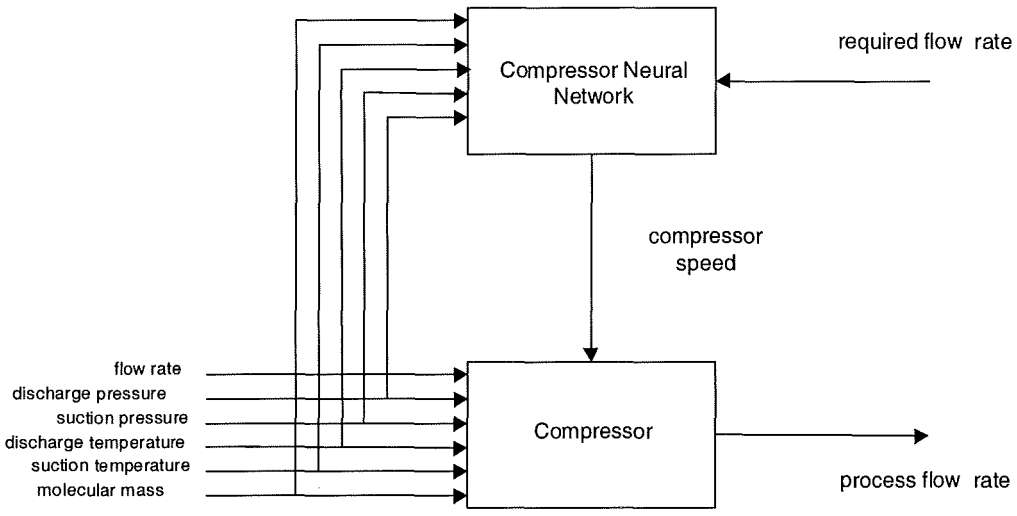

Figure 7.1: Compressor Performance Control Schematic

7.2.3 *Self optimisation*

Addition of a feedback loop for the flow rate will facilitate error generation or flow rate discrepancy. When the flow rate error exceeds a desired level an interrupt can initiate correcting action or optimisation. Self optimisation is achieved through periodic events, such as cumulative flow checks, as well as event driven adjustment of speed. The feedback loop is shown in Figure 7.2.

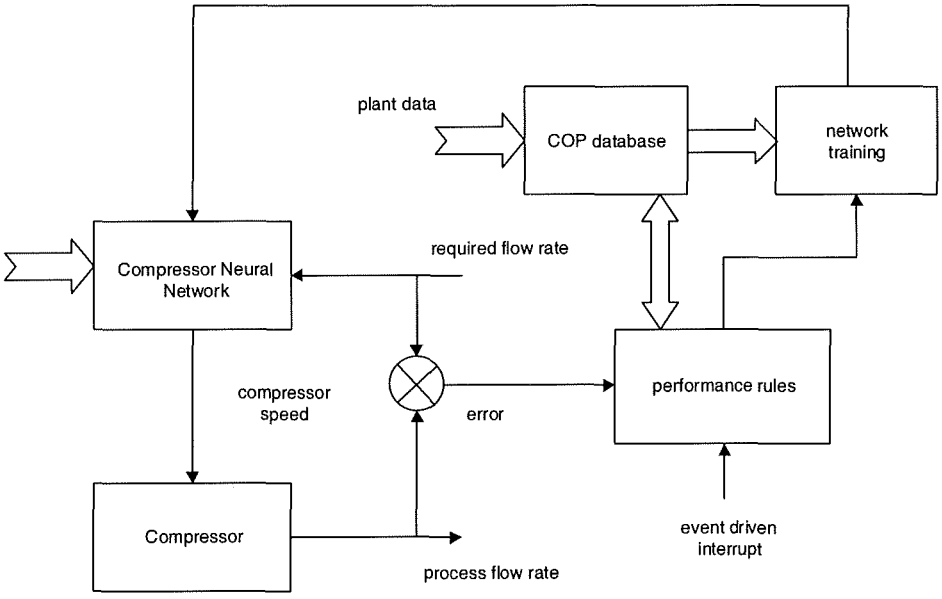

Figure 7.2: Feedback loop for Self optimisation

7.2.3.1 Performance Rules

The performance rules block is shown in Figure 7.3 and applies to individual compressors. Initiation of the performance related tasks will be by routine interrupt or by exception interrupt.

The performance rules will be part of a control hierarchy. The significance of the performance rules [14] will be low compared to plant safety considerations

and prime mover protection. If the advised speed impinges on either of these constraints the prime mover speed set point should be ignored. This will impact on the delivery performance of the compressor and performance control package.

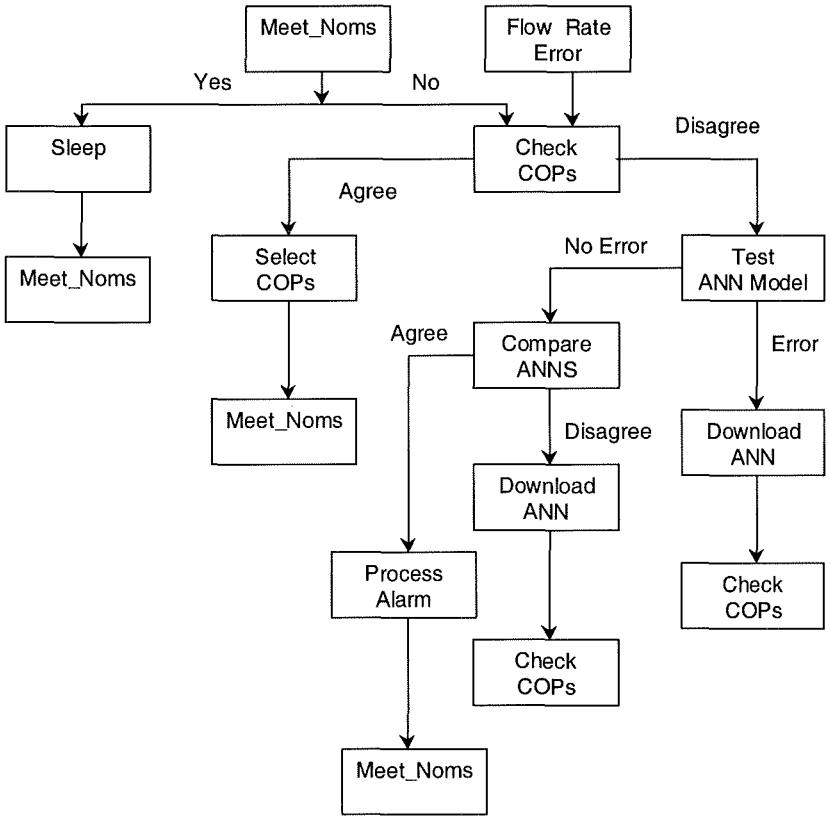

Figure 7.3: Performance Related Tasks

The routine interrupt is "Meet_Noms" which checks that the compressor throughput is as required to meet its apportionment of nominations, (in the case of a gas transmission compressor). A difference between the desired flow rate (set point) and the actual flow rate will raise an error interrupt, "Flow_Rate_Error".

7.2.3.1.1 "Meet_Noms" Algorithm

"Meet_Noms" is the default algorithm and is the only algorithm likely to be run when the compression process is in a steady state. A check on the cumulative and remaining period projected throughput, actual cumulative *Act_Cum*, is carried out and compared with the required cumulative, *Req_Cum*, satisfy nomination apportionment. The projected nomination apportionment is calculated from:

$$Proj\_Nom = Act\_Cum + Mez\_Inst \times Per\_Out \qquad (7.1)$$

where *Proj_Nom* is the projected nomination apportionment at period end
Act_Cum is the current period flow total
Mez_Inst is the current compressor flow rate
Per_Out is the remaining time in the nomination period

If *Act_Cum* and *Req_Cum* meet *Cum_Satisfactory* criterion then "Meet_Nom" goes to sleep for an *Interval* before being run again. If the *Cum_Satisfactory* criterion is not met then it is deemed that the nomination apportionment will not be met and corrective action is initiated through the "Check_COPs" algorithm.

Nomination apportionment, *Req_Cum*, is decided for each compressor at the beginning of a nominations period. The plant operator taking consideration of any plant constraints or scheduled downtime and equipment availability must plan the apportionment. The *Cum_Satisfactory* criterion and *Interval* period is determined in consultation with the plant operator with due regard to spare compression capacity and plant steadiness.

"Meet_Noms" is intended to have an "integral" type control action, reducing any nominations offset averaged over the nomination period.

7.2.3.1.2"Flow_Rate_Err" Algorithm

"Flow_Rate_Err" is run when an excessive discrepancy flag, *Exc_Inst*, is generated between the measured instantaneous compressor throughput, *Mez_Inst*, and the desired instantaneous compressor throughput, *Dez_Inst*. Corrective action is initiated through "Check_COPS". *Exc_Inst* is set at a level that reflects allowable variation in the plant steady state and could be determined as a rolling average. When *Exc_Inst* is set a *Bad_Data* flag is set in the COP database.

7.2.3.1.3"Check_COPs" Algorithm

This algorithm is initiated from either "Meet_Noms" or "Flow_Rate_Err". If *Exc_Inst* is set, further error testing occurs through the "Test_ANN_Model" algorithm. If *Exc_Inst* is not set then the "Change_COP" algorithm is run.

7.2.3.1.4"Select_COP" Algorithm

At this point the compressor will not meet its nomination apportionment and there is no significant variation between the measured flow rate, *Mez_Inst*, and the last flow rate estimate required to meet nominations, *Dez_Inst*. A new *Dez_Inst* is calculated to meet nomination apportionment over the remainder of the nomination period. This is calculated as:

$$Dez\_Inst = \frac{Req\_Cum - Act\_Cum}{Per\_Out} \qquad (7.2)$$

where *Dez_Inst* is the required flow rate

Req_Cum is the nomination apportionment

Act_Cum is the current period cumulative flow total

Per_Out is the remaining time in the nomination period

The new value of *Dez_Inst* is input to the ANN model along with the process inputs and a new value of compressor speed is downloaded a change in set point to the compressor prime mover speed governor. After a period of time,

Wait designed to allow the compressor to reach steady state at its changed COP, "Meet_Noms" is run. Following the COP change "Meet_Noms" should follow the default path and sleep for an *Interval.*

Wait should be selected as a minimum of five time constants based on the rotating set inertia [109]. Following successful COP change a *Good_Data* flag is set in the COP database.

7.2.3.1.5"Test_ANN_Model" Algorithm

Entering this algorithm, the circumstances are that nominations will not be met and that there is a discrepancy between *Mez_Inst* and *Dez_Inst.* This could be due to failure of field equipment or failure / corruption of the ANN model. At this point the *Bad_Data* flag is set in the COP database.

A test set of data, which is a subset of the training data used to train the network used in the model, is run and the residuals of the compressor speed estimates are compared with the training set residuals. If the two are significantly different then the active network is considered to have been corrupted and the "Download_ANN" algorithm is run.

If there is no significant difference between the residuals of the two datasets the algorithm "Compare_ANNs" is run.

7.2.3.1.6"Download_ANN" Algorithm

Running this algorithm downloads the most recently trained ANN to become the active reference model ANN. Following download, the algorithm "Check_COPs" is immediately run.

7.2.3.1.7"Compare_ANNs" Algorithm

The test set data is run on the active ANN and most recently trained ANN. If comparison of the residuals reveals a significant difference then the "Download_ANN" algorithm is run. The difference will have come about through a change in the relationship between the input vectors and the compressor speed. This action will be indicative of a change in compressor

characteristics and is expected to be an infrequent course of action. "Check_COPs" is run immediately following download.

If there is no significant difference between the residuals then the ANN model performance is considered to have been proven and the discrepancy in flow rates is due to a process/instrument fault. The "Process_Alarm" algorithm is run.

7.2.3.1.8"Process_Alarm" Algorithm

"Process_Alarm" will at least involve some sort of fault annunciation, following which program flow reverts to the "Meet_Noms"/"Sleep" cycle. If the process fault is not cleared the program flow will result in "Process_Alarm" on the next "Meet_Noms" cycle or "Flow_Rate_Error" cycle.

7.2.3.2 Network Training

Training the networks should serve a threefold purpose:

- to have prepared an alternative compressor ANN model
- to accommodate changes in compressor characteristics
- to represent as wide an operating envelope as the data will allow

Joubert [102], describes the application of neural network control to an unsteady process:

1. New data is accepted only if 80% of the inputs have changed in the preceding five minutes
2. The input values to the network are calculated from 20 minute moving averages to approximate steady state conditions.
3. Training can only take place when 60 time based data sets are available
4. Half of the data sets deemed to be "old" are discarded to reflect changing plant dynamics

Point 1 is designed to identify inputs that have failed and are "frozen" at their last "good" value. This is a characteristic of some SCADA software in use, but validation of training data values is required before they are committed to the COP database. Point 2 is unlikely to be required in a quasi-steady state process such as compression since the input values should be slow changing under normal operating conditions. Point 3 identifies a minimum amount of data required for training. This was noted in chapter 5 where smaller training sets led to increased residual estimate error. Point 4 identifies the need to have the neural network trained on data representative of the process.

As a starting point, assuming a populated COP database, the following training strategy would be implemented:

- Subject to scan times, new data points are created each minute, if any of the inputs have changed in the preceding five minutes. This will create 1440 data points per day per compressor in the COP, sufficient to train a day specific neural network
- Since the process is quasi-steady state no averaging is required.
- A "standby" network is trained on day data at the end of a nomination period. A "full" network is trained on a rolling, thirty (operating) day data set at the nomination period end, providing no changes to compressor hardware have been made which will change the characteristics.
- Active networks would only be updated as described under the Performance Rules.

At start-up, either at commissioning, or following compressor wheel change out or seal replacement the model will start with manufacturers or plant trials head maps. As each day of operation is completed, the COP is updated,

gradually building up a training set to the full, thirty day set. At this point the 31st day data set is discarded. The operator can override selection of the "full" network to use the "day" network, although this represents the previous chronological day of compressor performance.

7.2.3.3 COP Database

A limited database management function is required to structure COP data. The management function will include compilation of COP data point from the various data sources. Neural network weight data will also be held in the database. The COP will have an archive facility that will allow characterisation of compressor performance through its lifetime.

7.2.3.3.1 Input Data Point Fields

Input data point fields will include appending *Surge* and *Good_Data* or *Bad_Data* flags to the data point. Data points with *Surge* or *Bad_Data* flags set are not used for training purposes. A data point fields structure is shown in Figure 7.4.

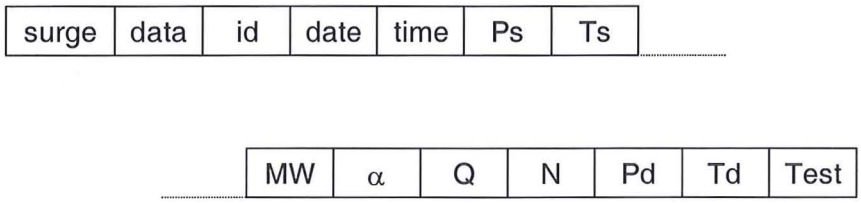

Figure 7.4: COP Data Point Format

Each data point has:

- surge field, set if the surge valve control signal was active
- data flag field, set if the data is classified as *Bad*
- id field, a compressor identifier
- date field
- time field

- Ps field, suction pressure input value
- Ts field, suction temperature input value
- MW field, gas molecular mass value
- field, guide vane angle value
- Q field, compressor flow rate value
- N field, compressor speed value
- Pd field, discharge pressure value (discharge pressure constraint)
- Td field, discharge temperature value
- Test field, set if used as ANN test data

7.2.3.3.2Neural Network Weight Data

The neural network neuron weight and input scaling values are stored by the day. These will include scale and zero values for six or seven inputs, (suction and discharge pressure, suction and discharge temperature, flow rate, molecular mass and guide vane angle). Indications are that data for between three and up to thirty hidden nodes and one output node will be stored. All active network data will be kept on the database for one year or until the compressor is serviced at which point the data is archived.

7.3 Compressor Performance Controller Integration

7.3.1 Discrete Controller

It would be possible to program a discrete controller with a neural network architecture and to accept the download of weight values from a supervisory computer. A potential drawback with this is that the network architecture could probably not be changed without operator intervention to re-program the controller.

The performance control scheme is designed to assist plant operators in managing throughput requirements on the compressor plant, with minimum

intervention. It is likely that a plant of this level would have a supervisory computer control system. Since the function of the performance controller is to schedule/plan compressor throughput the most likely "home" for the performance controller is in a hierarchical control system. Given the level of computing power required for neural network training and database memory capacity it is unlikely that the controller could be implemented in a stand alone discrete controller.

7.3.2 Hierarchical Control Systems

A hierarchical control system will typically consist of 5 levels. The five levels are further divided into high, intermediate and low level tasks. These are summarised in Figure 7.5. Each of these levels are interdependent, failure of a low level data acquisition task can result in incomplete information available for decision making at higher levels, see Pearson [110].

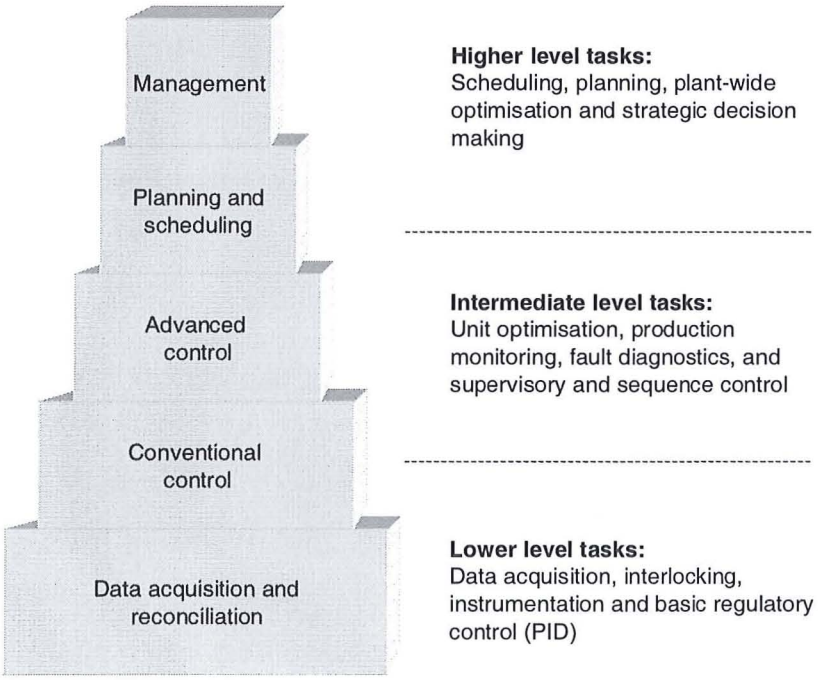

Figure 7.5: Generic Plant Hierarchy

Irwin [111] outlines the development of controllers from micro controllers through to distributed control systems (DCS) and supervisory control and data acquisition (SCADA). DCS and SCADA computers are generally found in intermediate to higher level tasks category. Low level tasks and data acquisition are carried out by programmable logic controllers (PLC), with data being communicated upwards to DCS and/or SCADA.

Relatively sophisticated functions, previously associated with higher level computers, such as DCS systems, are now built in to lower level controllers such as PLCs. These functions include PID algorithms or fuzzy logic functions. PLC programming was previously confined to a ladder logic approach but recent standard, IEC 1131-3 Programmable Controllers, now define the use of higher level languages for PLC programming [112]. Typical SCADA architecture [113], as shown in Figure 7.6, reflects the three level task hierarchy.

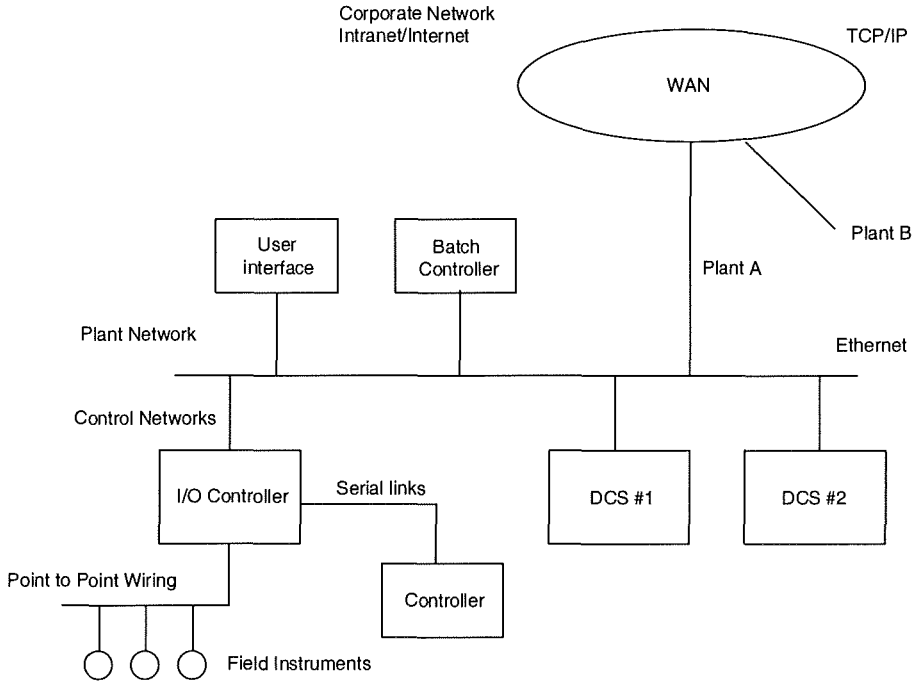

Figure 7.6: Typical SCADA Architecture

The system shown in Figure 7.6 shows discrete wiring from individual field instruments, such as transmitters of valve positioners, which are collated in an input/output (I/O) controller. Data is passed from the control network, usually proprietary to the SCADA manufacturer, on to the Plant Network. Higher level control is implemented on the Plant Network using information from other DCS loops. Access to the Plant ethernet provides management information for strategic decision making. Very latest, and future, architectures are likely to be based on Fieldbus technology, [114], using high speed deterministic networks shown in Figure 7.7.

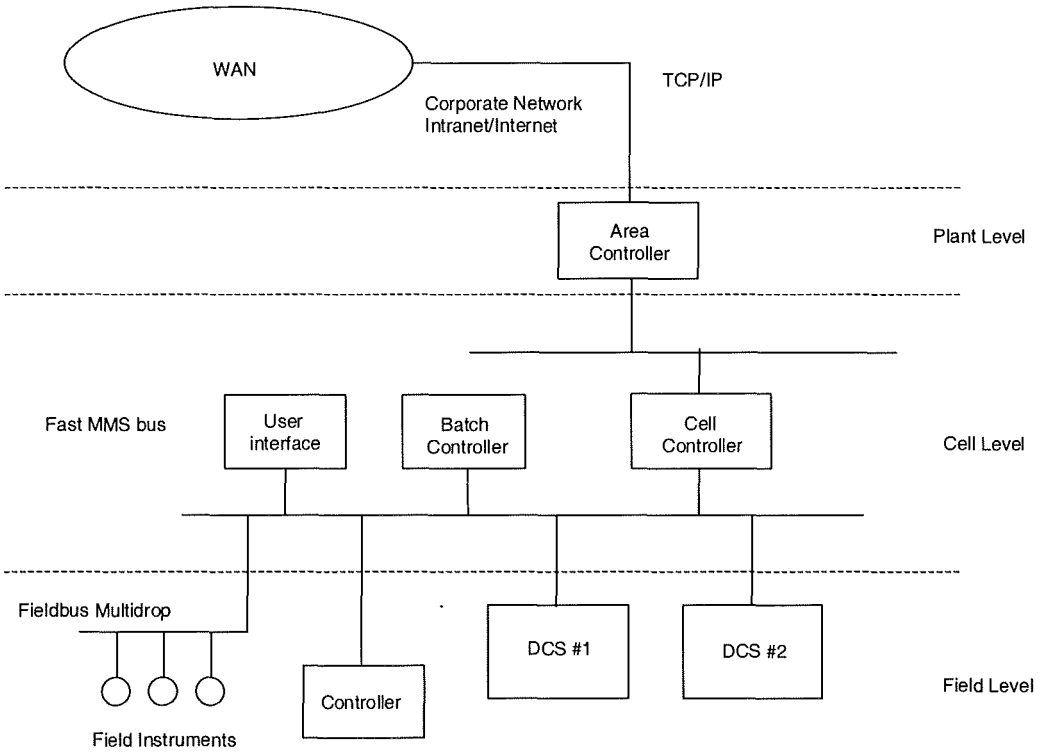

Figure 7.7: Fieldbus Based SCADA Architecture

Fieldbus technology uses SMART instruments communicating on a fast Multiple Master Slave bus. Network cycle time is typically less than 100 ms at

Field and Cell Levels, and less than 1 second at Plant Level [114].

7.3.3 Performance Controller Implementation

The performance controller package could be implemented on a PC type computer connected at plant network level. Data would then be collected from a real time server using a Real Time – Application Programme Interface (RT-API), or from a process database using a "standard" tool such as Microsoft ODBC, Bodington [115]. In more modern systems data could be embedded using OLE object technology. The performance controller is event driven, generating events such as flow rate discrepancies or projected nominations shortfalls. Yazdi [116] has described event driven supervision and control in a formal context.

7.4 Discussion

The performance rule logic in the controller is such that minimal intervention should be required. An advantage of using a neural network model is that no update of constants associated with compressor equation based models is needed. As the compressor characteristics change with time, no re-modelling is required for altered characteristics as neural networks can be retrained to reflect new performance characteristics.

Initial set up of the controller will need site-specific steady state flow tolerances and wait periods to be "tuned". The database of operating points is automatically maintained. No surge point can be used for training therefore the likelihood of a surge incidence should be reduced. Uncertainty in neural network speed estimate, see chapter 5, requires a surge system to be in place. Speed of response in a hierarchical system is slow therefore performance control is supplemental and transparent to surge control system. ANN model speed estimate should be more accurate than operator estimate. Better speed estimates combined with continuous monitoring should facilitate smoother operation of

the compressor plant. This, in turn, should extend maintenance intervals for the compressor.

Despite an increased level of automation, the plant operator must schedule compressor usage. Once the compressor throughput is determined, the neural network model will estimate compressor speed and recommend a set point for the fuel governor. The set point could be downloaded direct, given confidence in the system.

Implementation in a discrete controller is possible, but would require further research. Improved response times in the discrete controller would allow implementation of a surge prediction scheme using time step propagation techniques. The controller would then embody surge control and performance control characteristics.

7.5 Conclusions

This chapter has defined the algorithms required to implement a novel compressor performance control scheme. A review of plant control hierarchies has identified the scheme would be likely to be implemented at plant network level. System response time at this level is not sufficiently short to allow surge control.

Coupled with uncertainty in speed estimates, the performance system is complementary to the surge control system. The proposed system could be developed for a discrete controller if constraints on neural network architecture are acceptable, in terms of compressor speed accuracy. As a discrete controller, with shorter system response time scales, a surge control algorithm, based on neural network time step prediction could be researched.

8 Conclusions

8.1 Summary

Natural Gas transmission requires the operation of a large and high cost infrastructure of transmission and distribution pipelines, as summarised in Chapter 1. Central to the movement of gas, from production source to consumer, are gas transmission compressors. It is estimated that approximately 1400 million tonnes of gas is sold each year with an annual growth of 4% [5]. Compressor fuel gas is estimated to be 16% of this figure or 224 million tonnes [1]. This corresponds to an annual CO_2 production of 616 million tonnes. A saving in compressor fuel gas of 1% could save the production 6 million tonnes of CO_2 per year.

Current compressor control philosophy pivots around prevention of surge or anti-surge control [8]. Prevention of damage to high capital cost equipment is a key control driver but other factors such as environmental emissions restrictions and pressure on operating costs require most efficient use of fuel and extended plant availability. These require reliable and accurate *performance* control. Previous work on performance control had been described [14] and suggested a starting point for this research. The principles and operation of compressors are described in Chapter 2.

To investigate performance control a steady state compressor model was developed based on flow and adiabatic head coefficients. This method is broadly similar to that described in [45], and is regarded as state of the art, industry standard technique. Actual compressor head data was used in the model and correlations were applied to change the characteristics for changed process conditions. The basis of the model and its development is described in Chapter 3.

The techniques of neural network function approximation and pattern

recognition were investigated in Chapter 4. Strong similarities between neural network function approximation and statistical regression techniques were noted. Using neural networks avoid the potential difficulties in specifying regression model coefficients. Neural networks can be readily re-trained, once a database is populated, to reflect changing characteristics of the input/output relationship thus precluding costly re-modelling by an expert.

Research into how well neural networks might be used to model compressor head map characteristics was described, in Chapter 5. A limited description of two neural network packages was undertaken and an account of the development of a neural network, using standard "office" software, is included. A program of testing was devised to assess the performance of neural networks. The tests range from the ideal case of modelling an exact compressor head map to modelling a random set of compressor operating points. Data sets were generated using the steady state compressor model. The results are discussed and a method to embody the neural network compressor performance characteristics representation in a control scheme is introduced.

Chapter 6 reviewed established control paradigms and the use of neural networks in control systems were identified. These were generally to be found in the areas of adaptive or model predictive control. The advantage of using neural networks for model predictive control (MPC) is that they can be trained periodically, to reflect changing conditions in plant. This is applicable to compressors and the performance deterioration observed due to bearing wear, degradation of aerodynamic properties and degradation of impellor/casing seals. The decision to update the neural network model can be left to plant operations staff. They are in the best position to observe changes in plant and to decide when action needs to be taken to preserve, or improve, plant efficiencies.

Algorithms required to implement a novel compressor performance control scheme were described in Chapter 7. A review of plant control hierarchies has

identified how the scheme might be implemented. A performance control algorithm would be implemented on a PC connected at plant network level. The performance controller depends on model predictive control, based on a neural network model of current compressor characteristics. Performance control objectives are to monitor on going compressor throughput against a delivery period target. If it is predicted that the target will not be met an event is generated. The algorithm evaluates current process load and suggests a new compressor speed or updates the neural network model to reflect current compressor characteristics. A throughput variation event can also initiate performance controller corrective action. Thus, the algorithm would be invoked periodically or through event generation.

8.2 Conclusions

8.2.1 The justification for a dedicated Performance Controller

It is clear that there is an increasing requirement to improve compressor performance in terms of fuel consumption and gas delivery, without repeated operator intervention. Improved performance can partially abate increasing pressures to meet increasingly stringent environmental emissions restrictions. It is likely that, as Climate Change mechanisms are brought into place, direct financial penalties will result as a consequence of failure to curb emissions. Improved fuel efficiency can result in reductions in fuel gas costs, improving margins in increasingly competitive markets. Indications are that significant economies of scale are to be made with a performance controller that can achieve even a modest reduction in fuel gas consumption. There is a definite requirement for a flexible performance controller that can be integrated within both new, and existing compressor control installations.

8.2.2 Research into Neural Networks and Compressor Characteristics

The results of the testing are clearly indicative that neural networks can "learn" compressor head characteristics. The most rigorous test case, comprising a random data set of operating points, suggest that compressor speed may be estimated to within ± 2.5% of the speed required to meet a given flow rate for a given discharge pressure constraint, see section 5.6. Since the neural network trains on actual operating data no assumptions regarding invariant parameters are required. In a similar vein, the method can be applied to any combination of inputs associated with invariant parameter based compressor control e.g. suction throttling or reduced parameter control.

The compressor head map characteristic is clearly shown to be best represented by MLP type feed forward networks. Results for RBF networks indicate a comparable average error level but exhibit larger instantaneous errors in speed. These are believed to be associated with speed extrapolation outside the RBF trained domain, see sections 5.6.7.2 and 5.7.5.

The research into the use of neural networks to represent compressor characteristics, as described in this thesis, is believed to be novel.

8.2.3 Operational advantages of Neural Network based MPC

The use of supervised training techniques will allow a neural network to be trained with actual compressor data from live plant. The neural network can be re-trained (or replaced) to reflect changes in compressor characteristics with time. The decision to update compressor characteristics can be left to the experience of Operations Staff and will not require intervention by modeling experts or mathematicians. Hence the neural network will embody the "knowledge" of Operations Staff, one element of a knowledge based or expert system. This knowledge will facilitate goal-oriented operation of the compressor i.e. longer-term goal oriented, surge free operation. Automated

performance control can free up compression capacity and avoid human errors induced by operator intervention.

The compressor performance control system, based on "learning" neural networks, is believed to be novel. The control system can be adapted to represent changed compressor characteristics through re-training the neural networks model(s). No mathematical or modeling expert intervention is required. This flexibility surpasses existing state of the art performance control systems.

8.2.4 Implementation and Implications of Performance Control

The neural network head map model can be incorporated in a model predictive control scheme to estimate compressor speed. Using critical process parameters as inputs, the speed estimate can be used for controlling compressor throughput with compressor speed as the controlled variable. The controlled variable set point can be specified in absolute units in vector space. This aspect of the research is believed to be previously untried in the context of compressor control and hence constitutes original research.

The use of an absolute variable, compressor speed, as opposed to a relative variable, ratio of COP line to surge line gradient, is believed to be novel in the field of compressor control. In these respects, the research described in this thesis constitutes a novel contribution to compressor control.

Aside from the potential emissions and cost savings there are control benefits from the performance controller as described in section 7.2. Implicit in the performance control is improved anti-surge control since the neural network will not embody surge data – the neural network model will not learn how to surge. A conditional stability has been described in section 2.4.4 that, identifies that there is only one unconditionally stable point of operation for fixed process conditions. This point is where the compressor is in dynamic equilibrium with

the process when the throughput and discharge pressure are balanced. It is this compressor speed for this point which is identified by the neural network model. In effect the performance controller will attempt to drive the compressor into a dynamic balance with the process load.

8.2.5 Generic Expert System

The idea of dynamic balance and equilibrium operating point was described in section 2.6, relating to control issues with a micro-hydrogenerator. Given the similarities between compressor and micro-hydrogenerator operating equilibrium points there is merit in considering the performance control of compressors as an exercise in operating at natural equilibrium points determined by the process "load". Most rotating machinery equipment is designed to deliver a throughput or a head for a given set of conditions and constraints. As described the performance controller could be applied to any rotating equipment system. Identification of the critical process variables and description of an adequate rule set would be required for each system. As such the performance controller is generic model of an expert system for the performance control of rotating machinery.

8.2.6 Comparisons with conventional MPC performance controllers

Cordiner [14], reported simulated fuel gas savings, of 5% between conventional load sharing (equidistant) and the fuzzy logic based (non-equidistant) load-sharing controllers. It was further stated that the fuzzy optimiser would predict compressor speed to approximately 5%[1].

---

[1] Verbal statement by S. Cordiner.

Applying those figures to the neural network performance controller would indicate that incremental fuel gas savings of approximately 8% are possible on the basis that the compressor speed is estimated to within 2.5%.

On the global consumption estimates [1] this equates to a saving of 34 million tonnes of CO2 per year and an energy saving of 205PJ (1PJ = 10^{15}J).

8.3 Suggestions for Further Work

8.3.1 Reduced Field Inputs to Measure Compressor Performance

The research into normalised flow and head coefficients suggests that three, or even two, parameters are required to train a neural network to represent compressor performance characteristics. Further research could be undertaken to confirm the validity of these findings and to extend them into a new control technique using fewer field inputs. Coupled with research into surge control, based on neural network technology, this could revolutionise, and greatly simplify, compressor control philosophy.

8.3.2 Surge Prediction

The proposed system could be developed for a discrete controller if constraints on neural network architecture are acceptable, in terms of compressor speed accuracy. As a discrete controller, with shorter system response time scales, a surge control algorithm, based on neural network time step prediction could be researched. With reference to Greitzers dynamic model equations, see section 3.1, two the compressor parameters, namely mass flow rate and compressor pressure rise can be represented by measured values. These are the measured flow meter differential pressure and suction and discharge pressures. The rates of change of these parameters represent dynamic compressor behaviour and could be used as inputs to train a neural network to

predict compressor surge. It is recognised that a patent [117] for surge prediction, based on rates of approach to surge, exists, but it is believed that this is not implemented with neural networks. The surge prediction scheme could be similar to a continuously stirred tank reactor (CSTR) control scheme described by Bhat [91]. This scheme features a neural network reagent predictor that has a time step look ahead capability. Suggested inputs and outputs could be as shown in Figure 8.1.

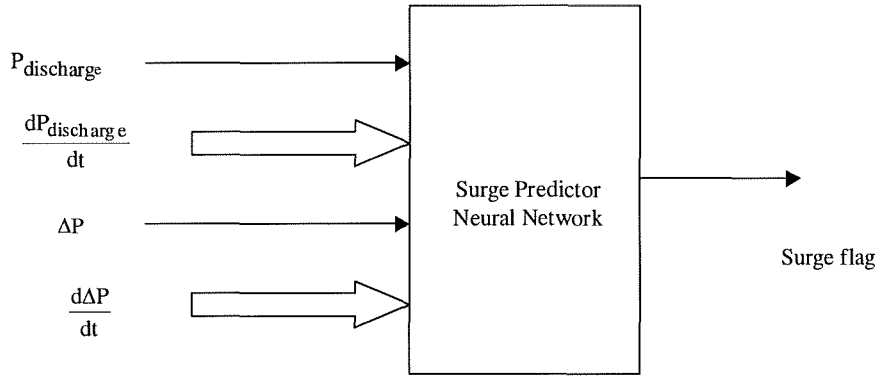

Figure 8.1 – Surge Prediction Scheme

The absolute discharge pressure and flow meter differential provide "location" information in the compressor performance map. The differential terms provide information regarding the next likely value of parameter and so head map location. Large values of $\frac{d\Delta P}{dt}$ accompanied by small values of $\frac{dP}{dt}$ are indicative of surge line proximity. The training target could be the compressor surge condition in n-time steps. The output could be a digital state of 'surge' or 'no surge'. As an alternative the output could be an analogue of "propensity to surge", decided by a fuzzy rule base.

8.3.3 Fuel consumption optimiser

In the proposed performance control scheme, the operator on the basis of a merit order would plan the required flow rate for each compressor. The merit order can be represented as a ranking, or digit. This information could be represented in a scheduling neural network such that the operator need only enter the compressor station throughput. The scheduler neural network would then output individual compressor flow rates and the performance controller for each would track monitor throughput. A schematic of the scheduler neural network inputs and outputs is shown in Figure 8.2

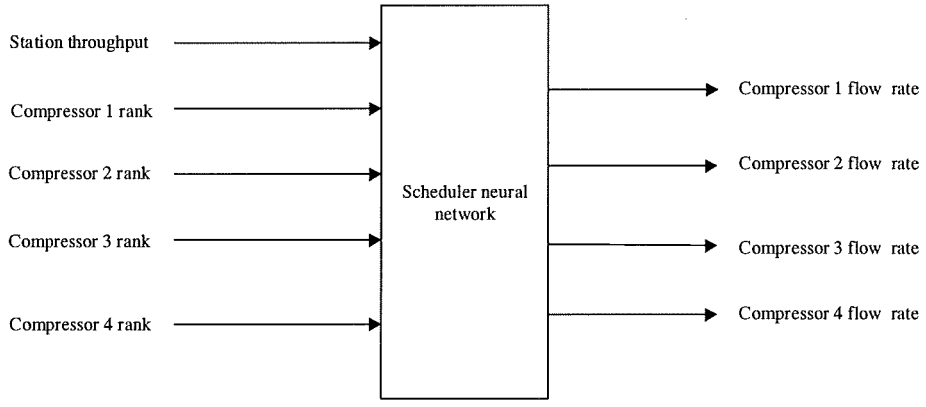

Figure 8.2 – Compressor Scheduler Neural Network

8.3.4 Reliability Centred Maintenance (RCM) Aid

Whilst performance control is the primary concern here, there is potential to develop a diagnostic aid to assess compressor performance. Several neural networks might be trained throughout the compressor life either by time or by event using actual operating data. Compressor performance can be tracked and deterioration monitored to construct a Reliability Centred Maintenance, RCM,

programme tailored to each compressor in situ. This could lead to reduced compressor downtime and reduced maintenance costs. An indication of drift in compressor performance could be determined from the frequency at which the compressor performance map neural network was being updated.

Appendix A

Nova Chemical's Press Release

NOVA Chemicals' News

Thursday, October 14, 1999

Temporary shutdown of NOVA Chemicals' Corunna, Ontario olefins facility:

Estimated three-week outage for repairs to charge gas compressor

FOR IMMEDIATE RELEASE

SARNIA, Canada – NOVA Chemicals Corporation (NOVA Chemicals) announced today that it is shutting down its Corunna, Ontario olefins facility for approximately three weeks to repair the charge gas compressor. The outage is required due to higher-than-normal vibrations occurring in the compressor. The repairs are necessary to ensure the compressor operates safely and reliably in the winter months and beyond.

NOVA Chemicals has the necessary replacement parts on site and the outage and repairs will begin immediately. The financial impact of the outage – including lost sales, costs of the repairs and increased logistics costs – is currently estimated at U.S. $10 million (after tax). The company expects that the financial impact will be recorded in the fourth quarter of this year.

NOVA Chemicals has advised its ethylene, propylene and chemical and energy products customers that the company will be allocating supply of these products during the outage.

NOVA Chemicals is working to minimize the impact of this shut down on polyethylene customers by drawing down inventories and coordinating customer needs. It is expected that NOVA Chemicals' two polyethylene plant sites in the Sarnia region (Moore and St. Clair) will undergo temporary outages over the next three weeks as ethylene inventories decline.

Ethylene is also used as a feedstock at NOVA Chemicals' styrene plant at Sarnia. The styrene plant will operate at reduced capacities. The company is working with its styrene monomer and polystyrene customers to minimize the impact of the outage.

The Corunna facility (in the Sarnia, Ontario region) produces ethylene, propylene, and chemical and energy products, with the following annual production capacities:

- 1.6 billion pounds of ethylene;
- 750 to 875 million pounds of propylene; and
- 3,250 to 3,750 million pounds of chemical and energy products.

NOVA Chemicals Corporation produces styrenics and olefins/polyolefins at 18 facilities in the U.S., Canada, France and the United Kingdom. NOVA Chemicals Corporation also has two equity investments: Methanex Corporation (27% owned), the world's largest and lowest cost producer of methanol; and Dynegy Inc. (25% owned), a major North American energy services provider. NOVA Chemicals Corporation shares trade on the New York, Toronto, Montréal and Alberta exchanges under the trading symbol NCX.

Visit NOVA Chemicals on the Internet at www.novachem.com

Appendix B
Table B1 – "Base Head Map"

21000rpm		22000rpm		22300rpm		20000rpm		19000rpm		18000rpm		17000rpm		16000rpm	
Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg	Qa-m3/s	Ha-kJ/kg
0.264325	121.0643	0.276912	132.8688	0.280689	136.5172	0.251739	109.8089	0.239152	99.10255	0.226565	88.94522	0.213978	79.33694	0.201391	70.27771
0.283206	120.1698	0.296692	131.887	0.300738	135.5084	0.26972	108.9975	0.256234	98.37026	0.242748	88.28799	0.229262	78.75071	0.215776	69.75841
0.311526	119.2752	0.326361	130.9052	0.330811	134.4997	0.296692	108.1861	0.281857	97.63798	0.267023	87.63076	0.252188	78.16447	0.237354	69.23912
0.330407	117.7843	0.346141	129.2689	0.350861	132.8184	0.314673	106.8338	0.29894	96.4175	0.283206	86.53537	0.267472	77.18742	0.251739	68.37363
0.358727	116.2933	0.37581	127.6326	0.380934	131.1372	0.341645	105.4815	0.324563	95.19703	0.307481	85.43999	0.290398	76.21036	0.273316	67.50814
0.377608	113.3711	0.395589	124.4254	0.400984	127.842	0.359627	102.8309	0.341645	92.8049	0.323664	83.29304	0.305683	74.29533	0.287701	65.81178
0.396488	111.5223	0.415369	122.3964	0.421033	125.7572	0.377608	101.154	0.358727	91.29151	0.339847	81.93476	0.320967	73.08378	0.302086	64.73858
0.424809	105.2604	0.445038	115.5238	0.451107	118.696	0.40458	95.47425	0.384351	86.16551	0.364122	77.33414	0.343893	68.98015	0.323664	61.10352
0.443689	99.59479	0.464817	109.3058	0.471156	112.3072	0.422561	90.33541	0.401433	81.52771	0.380305	73.17168	0.359177	65.26734	0.338049	57.81466
0.46257	93.92922	0.484597	103.0879	0.491205	105.9185	0.440542	85.19657	0.418515	76.88991	0.396488	69.00922	0.374461	61.55452	0.352434	54.52581
0.48145	84.98358	0.504376	93.26996	0.511254	95.83103	0.458524	77.08261	0.435598	69.56706	0.412671	62.43692	0.389745	55.69219	0.366819	49.33287
0.49089	74.547	0.514266	81.81576	0.521279	84.06231	0.467514	67.61633	0.444139	61.02373	0.420763	54.76922	0.397387	48.8528	0.374012	43.27445

Volumetric flow rate, Q_a (m^3/s) is shown in the left most column for each compressor speed (rpm) shown in the top row. All other values are compressor isentropic head, H_a (kJ/kg) arranged in corresponding columns of speed. The row of highest flow rates is the choked flow boundary. The row of lowest flow rates is the surge boundary.

Appendix C

Initial Network Research

	"Optimal" Search, input sub sets				"Extensive" Search, input subsets			
Net.	Type	Inputs	Hidden	Error	Type	Inputs	Hidden	Error
1	L	Ts, Td, dp	-	395	MLP	Pd, Td, dp	3	25
2	L	Ts, Td, dp	-	395	MLP	Td, dp	6	26
3	MLP	Pd, dp	6	52	MLP	Pd, Td, dp	7	32
4	L	Ts, Td, dp	-	395	MLP	Pd, Td, dp	6	36
5	MLP	Pd, dp	5	59	MLP	Pd, Td, dp	4	28
6	MLP	Pd, Td, dp	4	44	MLP	Pd, Td, dp	5	29
7	MLP	Pd, dp	6	59	MLP	Pd, Td, dp	9	30
8	L	Ts, Td, dp	-	395	MLP	Pd, Td, dp	5	38
9	L	Ts, Td, dp	-	395	MLP	Td, dp	6	31
10	MLP	Pd, Td, dp	6	51	MLP	Pd, Td, dp	5	28
11	MLP	Pd, Td, dp	6	33	MLP	Pd, Td, dp	11	29
12	L	Ts, Td, dp	-	395	MLP	Pd, Td, dp	9	37
13	MLP	Td, dp	2	351	MLP	Td, dp	6	25
14	MLP	Td, dp	4	52	MLP	Pd, Td, dp	5	24
15	MLP	Pd, dp	2	342	MLP	Pd, Td, dp	5	37
16	L	Ts, Td, dp	-	395	MLP	Td, dp	9	31
17	L	Ts, Td, dp	-	395	MLP	Pd, Td, dp	6	35
18	MLP	Td, dp	5	361	RBF	Pd, Td, dp	7	43
19	L	Pd, Td, dp	-	395	MLP	Td, dp	6	44
20	MLP	Pd, Td, dp	3	49	MLP	Pd, Td, dp	7	40
21	MLP	Pd, pd	4	60	MLP	Pd, Td, dp	7	40
22	L	Pd, Ts, Td, dp	-	484	MLP	Pd, Td, dp	5	38
23	L	Ts, Td, dp	-	433	MLP	Pd, Td, dp	4	34
24	MLP	Pd, Td, dp	7	48	MLP	Td, dp	7	37
25	RBF	Pd, dp	4	242	MLP	Td, dp	9	40
26	L	Pd, Ts, Td, dp	-	397	MLP	Pd, Td, dp	6	37
27	MLP	Pd, Td, dp	7	37	MLP	Pd, dp	5	34
28	L	Pd, Ts, Td, dp	-	475	MLP	Td, dp	9	38
29	RBF	Pd, Td, dp	2	1309	MLP	Pd, Td, dp	6	35
30	MLP	Pd, Td, dp	6	41	MLP	Td, dp	3	40

Table C1: Data Set I1 Initial Networks Searches

	"Extensive" Search, input subsets				"Extensive" Search, forced inputs			
Net.	Type	Inputs	Hidden	Error	Type	Inputs	Hidden	Error
1	MLP	Pd, Td, dp	3	25	MLP	All	9	23
2	MLP	Td, dp	6	26	MLP	All	10	27
3	MLP	Pd, Td, dp	7	32	MLP	All	7	25
4	MLP	Pd, Td, dp	6	36	MLP	All	5	23
5	MLP	Pd, Td, dp	4	28	MLP	All	12	28
6	MLP	Pd, Td, dp	5	29	MLP	All	13	20
7	MLP	Pd, Td, dp	9	30	MLP	All	10	26
8	MLP	Pd, Td, dp	5	38	MLP	All	11	22
9	MLP	Td, dp	6	31	MLP	All	17	23
10	MLP	Pd, Td, dp	5	28	MLP	All	15	24
11	MLP	Pd, Td, dp	11	29	MLP	All	10	19
12	MLP	Pd, Td, dp	9	37	MLP	All	4	24
13	MLP	Td, dp	6	25	MLP	All	8	23
14	MLP	Pd, Td, dp	5	24	MLP	All	8	23
15	MLP	Pd, Td, dp	5	37	MLP	All	10	26
16	MLP	Td, dp	9	31	MLP	All	6	24
17	MLP	Pd, Td, dp	6	35	MLP	All	6	24
18	RBF	Pd, Td, dp	7	43	MLP	All	4	22
19	MLP	Td, dp	6	44	MLP	All	6	26
20	MLP	Pd, Td, dp	7	40	MLP	All	4	27
21	MLP	Pd, Td, dp	7	40	MLP	All	13	26
22	MLP	Pd, Td, dp	5	38	MLP	All	8	26
23	MLP	Pd, Td, dp	4	34	MLP	All	8	25
24	MLP	Td, dp	7	37	MLP	All	8	26
25	MLP	Td, dp	9	40	MLP	All	5	23
26	MLP	Pd, Td, dp	6	37	MLP	All	10	25
27	MLP	Pd, dp	5	34	MLP	All	12	27
28	MLP	Td, dp	9	38	MLP	All	13	19
29	MLP	Pd, Td, dp	6	35	MLP	All	6	24
30	MLP	Td, dp	3	40	MLP	All	14	22

Table C2: Extensive Architecture Searches

	"Extensive" Search, forced inputs				"Extensive" Search, new splits			
Net.	Type	Inputs	Hidden	Error	Type	Inputs	Hidden	Error
1	MLP	All	9	23	MLP	All	6	26
2	MLP	All	10	27	MLP	All	9	26
3	MLP	All	7	25	MLP	All	14	18
4	MLP	All	5	23	MLP	All	6	35
5	MLP	All	12	28	MLP	All	17	30
6	MLP	All	13	20	MLP	All	7	27
7	MLP	All	10	26	MLP	All	8	22
8	MLP	All	11	22	MLP	All	5	38
9	MLP	All	17	23	MLP	All	9	31
10	MLP	All	15	24	MLP	All	5	28
11	MLP	All	10	19	MLP	All	17	31
12	MLP	All	4	24	MLP	All	8	47
13	MLP	All	8	23	MLP	All	10	35
14	MLP	All	8	23	MLP	All	11	33
15	MLP	All	10	26	MLP	All	6	30
16	MLP	All	6	24	MLP	All	9	32
17	MLP	All	6	24	MLP	All	3	36
18	RBF	All	4	22	MLP	All	7	32
19	MLP	All	6	26	MLP	All	6	52
20	MLP	All	4	27	MLP	All	16	25
21	MLP	All	13	26	MLP	All	4	34
22	MLP	All	8	26	MLP	All	8	34
23	MLP	All	8	25	RBF	All	19	33
24	MLP	All	8	26	MLP	All	13	40
25	MLP	All	5	23	MLP	All	3	24
26	MLP	All	10	25	MLP	All	3	30
27	MLP	All	12	27	MLP	All	19	34
28	MLP	All	13	19	MLP	All	8	26
29	MLP	All	6	24	MLP	All	5	29
30	MLP	All	14	22	MLP	All	8	33

Table C3: Forced Input Sets

References

[1] British Gas Transco, UK, "System Control", Ref. C047

[2] British Gas Transco, UK, "Network code – The Summary", 1996

[3] Gresh T.M., "Compressor Performance – Selection, Operation and Testing of Axial and Centrifugal compressors", Butterworth-Heinemann,

[4] Eurogas, Annual Report 1998, http://www.eurogas.org

[5] International Gas Union, Statistical Data 1999, http://www.igu.org

[6] Transcanada website, http://www.transcanada.com

[7] White M.H., "Surge Control for Centrifugal Compressors", Chemical Engineering, pp 54-62, December 25 1972,

[8] Staroselsky N. and Ladin L., "Improved Surge Control for Centrifugal Compressors", Chemical Engineering, pp 175-184, May 21 1979

[9] Boyce M. P., "Tutorial Session on Practical Approach to Surge and Surge Control Systems", ASME Proceedings of the 12th Turbomachinery Symposium

[10] Batson B.W., "Invariant Co-ordinate Systems for Compressor Control", ASME 41st International Gas Turbine and Aeroengine Congress & Exhibition, Birmingham U.K., June 10-13, 1996

[11] Botros K.K. and Henderson J.F, "Developments in Centrifugal Compressor Surge Control – A Technology Assessment", ASME Journal of Turbomachinery, Vol 116, pp 240-249, 1994

[12] Botros K.K., Glover A., "Neural Networks & Fuzzy Logic – Overview of Pipeline Applications Part II: Fuzzy Logic, International Conference on Offshore Mechanics and Arctic Engineering, OMAE 99, 18th, St. John's, Newfoundland, July 11-16, 1999

[13] Vachtsevanos G., Kang H., Cheng J., Kim J., "Detection and Identification of Axial Flow Compressor Instabilities", ASME Journal of Guidance, Control and Dynamics, Vol. 15, No.5, September – October 1992

[14] Cordiner S.A., "Optimising Turbo-compressors Using Fuzzy Logic Control", M. Phil Dissertation, Napier University, 1997

[15] Gas Processors Suppliers Association, "Engineering Data Book", 1980

[16] Harman R.T.C., "Gas Turbine Engineering", MacMillan Press, 1981

[17] Douglas J.F., Gasiorek J.M., Swaffield J.A., "Fluid Mechanics", Longman, 1987

[18] Sapiro L., "Centrifugal Gas Compressors Aero-Thermodynamics Concepts", Solar Turbines Inc., San Diego, 1982

[19] Eastop T.D., McConkey A, "Applied Thermodynamics", Longman, 4th Edition, 1992

[20] Starling K.E. editor, "Compressibility and Supercompressibility for Natural Gas and other Hydrocarbon Gases", American Gas Association, Transmission Measurement Committee Report No. 8, (2nd printing for 1992 edition), 1994

[21] ISO6976:1995, "Natural Gas - Calculation of calorific values, density, relative density and Wobbe index from composition"

[22] ASME Report No. PTC10, "Power and Test Codes, Compressors and Exhausters", 1965

[23] Gent D.W., "The calculation of Isentropic Exponent for the Metering of Natural Gas", North Sea Flow Metering Workshop, NEL. East Kilbride, 1988

[24] Nova Chemicals' News, http://www.novachem.com, 14/10/99

[25] Liptak B., "Instrument Engineers' Handbook", 3rd Edition, Butterworth-Heinemann, 1995

[26] ESD Simulation plc, "Control and Operation of Centrifugal Gas Compressors", IchemE Continuing Education Programme Short Course Notes, Aberdeen, 1996

[27] Staroselsky N. and Minsky S., "Gas Processing Plants: Automation of compressors and Expanders to optimize Plant Performance", www.ccc.com, June 2000.

[28] "Series 3 Antisurge Controller – Operating Manual", Compressor Controls Corporation, Des Moines, Iowa, 1990

[29] deSa D., Maalouf S., "The Operating and Control Philosophy of Turbo-compressors, Measurement & Control", Volume 29, April 1996.

[30] Gysling D.L. and Greitzer E.M., "Dynamic Control of Rotating Stall in Axial Flow compressors Using Aeromechanical Feedback", ASME Journal of Turbomachinery, Vol. 117, pp307-319, 1995

[31] Jungowski W.M., Weiss M.H., Price G.R., "Pressure Oscillations Occurring in a Centrifugal Compressor System With and Without Passive and Active Surge Control", ASME Journal of Turbomachinery, Vol. 118, pp29-40, 1996

[32] Henderson D.S. and Pearson W.N., "An Improved Control Algorithm for an Electronic Load Governor", Journal of Measurement and Control, Vol. 30, No. 10, pp 293-296, December 1997.

[33] Henderson D.S. and Pearson W.N., "E.L.G – Application of Derivative Control Action for Improving Transient Response", Presented at 33rd Universities Power Engineering Conference, Napier University Sept. 1998.

[34] Botros K.K., Campbell P.J., Mah D.B., "Dynamic Simulation of Compressor Station Operation Including Centrifugal Compressor and Gas Turbine", ASME Journal of Engineering for Gas Turbines and Power, Vol. 113, pp 300 – 311, April 1991.

[35] Botros K.K. and Petela G., "Use of Method of Characteristics & Quasi-Steady Approach in Transient Simulation of Compressor Stations", presented at the 1994 ASME Fluids Engineering Division Summer Meeting, Lake Tahoe, Nevada June 19-23 1994.

[36] Botros K.K., Richards D.J., Brown R.J., Stachniak D.M., "Effects of Low Power Turbine/Compressor Rotor Inertia During Shutdown", presented at the 1993 Symposium on the Industrial Application of Gas Turbines, Banff, Alberta October 13-15 1993.

[37] Botros K.K., Jones B.J., Richards D.J., "Re-cycle Dynamics During Centrifugal compressor ESD, Start-up and Surge Control", presented at the 1st International Pipeline conference, Calgary, Alberta, June 9-13 1996.

[38] Botros K.K., Jungkowski W.M., Richards D.J., "Compressor Station Recycle System Dynamics During Emergency Shutdown", ASME Journal of Engineering for Gas

Turbines and Power, Vol. 118, pp 641-653, July 1996

[39] Boroomand M., "Simulation and Measurement of Transients in Pipes and Compressors", Ph.D. Thesis, Cranfield Institute of Technology, April 1992

[40] Greitzer E.M., "Surge and Rotating Stall in Axial Flow Compressors Part I: Theoretical Compression System Model", ASME Journal of Engineering for Power, pp 191-198, April 1976

[41] E.M.Greitzer, "Surge and Rotating Stall in Axial Flow Compressors Part II: Experimental Results and Comparison with Theory", ASME Journal of Engineering for Power, pp 199-217, April 1976

[42] Hansen K.E., Jorgensen P., Larsen P.S., "Experimental and Theoretical Study of Surge in a Small Centrifugal Compressor", ASME Journal of Fluids Engineering, Vol.. 103, pp 391-395, September 1981.

[43] Badmus O.O., Chowdhury S., Eveker K.M., Nett C.N., "Control-Oriented High-Frequency Turbomachinery Modelling: Single Stage Compression System One-Dimensional Model", ASME Journal of Turbomachinery, Vol. 117, pp 47 – 61, January 1995.

[44] Badmus O.O., Eveker K.M., Nett C.N., "Control-Oriented High-Frequency Turbomachinery Modelling: General One Dimensional Model Development", ASME Journal of Turbomachinery, Vol. 117, pp 321 – 335, July 1995

[45] Godse A.G., "Predict compressor performance at new conditions", Hydrocarbon Processing, June 1989

[46] Buckingham E., "On Physically Similar Systems: Illustration of the use of Dimensional Equations", Physics Review, Vol. 4, No. 4, pp 345 - 376, 1914

[47] White F.M., "Fluid Mechanics", 2nd Edition, McGraw Hill, Toronto

[48] Gurney K., "Neural Networks", UCL Press, 1997.

[49] Picton P., "An Introduction to Neural Networks", Macmillan, 1994

[50] McCulloch W.S. and Pitts W., "A Logical Calculus of the Ideas Imminent in Nervous Activity", Bulletin of Mathematical Biophysics, 5, pp 115 – 133, 1943

[51] Caudill M. and Butler C., "Understanding Neural Networks", Volume 2: Advanced Networks, MIT, 1993.

[52] Haykin S., "Neural Networks – A Comprehensive Foundation", Macmillan, 1994.

[53] Rosenblatt F., "Principles of Neurodynamics", Spartan Books, New York, 1962

[54] Widrow B., Hoff Jr. M. E., "Adaptive Switching Circuits", IRE WESCON Convention Record, 1960. Reprinted in "Neurocomputing – Foundations of Research", edited by Anderson A., Rosenfeld E., MA:MIT Press, Cambridge 1988.

[55] Hinde C.J., Fletcher G.P., West A.A., Williams D.J., "Neural Networks", ICL Systems Journal, January 1997.

[56] Trajan Software Ltd., Trajan 4.0 Neural Networks, County Durham, UK, 1999

[57] Hornick K., Stinchcombe M., White H., "Multi-Layer Feed Forward Networks are Universal Approximators", Neural Networks 2,5 1989

[58] Park J. and Sandberg I.W., "Universal Approximation Using Radial-Basis-Function

Networks", Neural Computation, Vol 3 , No. 1, Spring 1991

[59] Werbos P.J., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences", PhD Thesis, Harvard University, 1974

[60] Rumelhart D.E., McClelland J.L., "Parallel Distributed Processing", MIT Press, 1986

[61] Bishop C.M., "Neural Networks for Pattern Recognition", Clarendon Press, 1995

[62] Kohonen T., "The Self Organising Map", Proceedings of IEEE Vol. 78, pp 1464-80, 1990

[63] Cheng B. and Titterington D.M., "Neural Networks: A Review from a Statistical Perspective", Statistical Science, Vol. 9, No. 1, 1994

[64] Bothe H.H., "Artificial Neural Networks and Fuzzy Methods", (English draft), Springer Verlag, 1999

[65] Pedrycz W., "Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks", IEEE Transactions on Neural Networks, Vol. 9, No 4, July 1998.

[66] Joseph B., Wang F.H., Shieh D. S-S., "Exploratory Data Analysis: A Comparison of Statistical Methods with Artificial Neural Networks", Computers in Chemical Engineering, Vol. 14 No. 4, pp 413-423, 1992

[67] Bishop C.M., "Graphical models", Proceedings of 9th International Conference on Artificial Neural Networks, late paper presentation, Edinburgh, UK, 17th - 10th September 1999.

[68] Hinton G.E., "Product of Experts", Proceedings of 9th International Conference on Artificial Neural Networks, Vol. 1, pp 1-6, Edinburgh, UK , 7th-10th September 1999.

[69] Gerstner W., "Rapid Signal Transmission by Population of Spiking Neurons", Proceedings of 9th International Conference on Artificial Neural Networks, Vol. 1, pp 7-12, Edinburgh, UK, 17th - 10th September 1999.

[70] Neural Computer Sciences Ltd, NCS NeuFrame v 3.0.0.0, Southampton, UK, 1996

[71] Promised Land Technologies, "Braincel – an Excel97 add in", http://www.promland.com, 2000.

[72] Griffiths D., Stirling W.D., Weldon K.L., "Understanding Data - Principles & Practice of Statistics", Wiley, 1998.

[73] Lapin L.L., "Probability and Statistics for Modern Engineering", PWS-Kent, International Student Edition 1990.

[74] Foxboro Company, "Introduction to Process Control", 1986

[75] Shinskey F.G., "Process Control Systems", McGraw-Hill, 1979

[76] Hagglund T., "Process Control in Practice", Chartwell-Bratt, 1992

[77] White D.A., Sofge D.A., editors, "Handbook of Intelligent Control", Van Nostrand Reinhold, 1992

[78] Zurada J.M., Marks R.J., Robinson C.J., "Computational Intelligence Imitating Life", IEEE Press, 1994

[79] Vanek T.W., "Fuzzy Guidance Controller for an Autonomous Boat", IEEE Control Systems, April 1997.

[80] Shenoi S., Ashenayi K., Timmerman M., "Implementation of a Learning Fuzzy Controller", IEEE Control Systems, June 1995

[81] Kim J., Moon Y., Zeigler B.P., "Designing Fuzzy Net Controllers Using Genetic Algorithms", IEEE Control Systems, June 1995.

[82] Papa M., Tai H-M., Shenoi S., "Cell Mapping for Controller Design and Evaluation", IEEE Control Systems, April 1997

[83] Gonzalez A.J., Dankel D.D., "The Engineering of Knowledge Based Systems", Prentice Hall, 1993

[84] Kandadai R.M. and Tien J.M., "A Knowledge-Base Generating Hierarchical Fuzzy-Neural Controller", IEEE Transactions on Neural Networks, Vol. 8 No. 6, November 1997.

[85] Travé-Massuyès L. and Milne R., "Gas Turbine Condition Monitoring Using Qualitative Model-Based Diagnosis", IEEE Expert, May/June 1997.

[86] Antsaklis P.J., "Neural Networks in Control Systems", IEEE Control Systems Magazine, April 1990.

[87] Armitage A.F., "Neural Networks in Measurement and Control", Measurement & Control, Vol. 28, September 1995.

[88] Werbos P.J., "Handbook of Neural Computation", Section F1.9 Release 97/1, Oxford University Press, 1997

[89] Agarwal M., "A Systematic Classification of Neural-Network-Based Control", IEEE Control Systems, April 1997

[90] Nguyen D.H., Widrow B., "Neural Networks for Self-Learning Control Systems", IEEE Control Systems Magazine, April 1990

[91] Bhat N.V., Minderman Jnr P.A., McAvoy T., Wang N.S., "Modelling Chemical Process Systems via Neural Computation", IEEE Control Systems Magazine, April 1990

[92] Hoskins J.C., Himmelblau D.M., "Process Control via Artificial Neural Networks and Reinforcement Learning", Computers and Chemical Engineering, Vol. 16 No. 4, 1992.

[93] Bloch G., Sirou F., Eustache V., Fatrez P., "Neural Intelligent Control for a Steel Plant", IEEE Transactions on Neural Networks, Vol. 8 No. 4, July 1997

[94] Bhat N.V., McAvoy T.J., "Use of Neural Nets for Dynamic Modelling and Control of Chemical Process Systems", Proceedings of American Control Conference, 1989

[95] Chen F-C., "Back-Propagation Neural Networks for Non Linear Self-tuning Adaptive Control", IEEE Control Systems Magazine, April 1990

[96] Willis M.J., Montague G.A., di Massimo C., Tham M.T., Morris A.J., "Artificial Neural Network based Predictive Control", International Federation of Automatic Control, ADCHEM91 Symposium, Toulouse, France 1991.

[97] Hernandez E., Arkun Y., "Study of the Control Relevant Properties of Backpropagation Neural Network Models of Nonlinear Dynamical Systems", Computers and Chemical Engineering, Vol. 16 No. 4, 1992.

[98] Park Y-M., Choi M-S., Lee K.Y., "An Optimal Tracking Neurocontroller for Nonlinear Dynamic Systems", IEEE Transactions on Neural Networks, Vol. 7 No. 5, September 1995

[99] Fabri S., Kadirkamanathan V., "Dynamic Structure Neural Networks for Stable Adaptive Control of Nonlinear Systems", IEEE Transactions on Neural Networks, Vol. 7 No. 5, September 1996

[100] Horn J., "Feedback Linearisation Using Neural Process Models", 9th International Conference on Artificial Neural Networks, Edinburgh, 7th-10th September 1999.

[101] Miller III W.T., Sutton R.S., Werbos P.J., "Neural Networks for Control", MIT Press, 1995

[102] Joubert H.W., Theron P.L., Lange T., "The use of Neural Networks for Gasloop Process Optimisation", Elektron, Journal of South African Institute of Electrical Engineers, August 1996

[103] El-Sayed M.A.H., "Artificial Neural Network Based Control of Thermal Power Plants", 33rd Universities Power Engineering Conference, Napier University, 8th-10th September 1998.

[104] Kleymenov G., Krutovertsev S., Piskounov A., Vdovichev S., "Genetic Learning of Neural Networks For Calibration in Temperature and Humidity", Eurosensors XII, University of Southampton, 13-16th September 1998.

[105] Livneh B., "Neural Networks and Expert System for Process Control", http://www.kbe.co.za

[106] Seskin A., "Expert System used in Pipeline Scheduling for Petronet", Elektron, Journal of South African Institute of Electrical Engineers, March 1996

[107] Moore R., Rosenhof H., Stanley G., "Process Control Using a Real time Expert", IFAC Triennial World congress, Estonia, 1990

[108] Toral H., "Multiphase Flow Rate Identification by Pattern Recognition at Shell Auk Alpha Platform", North Sea Flow Measurement Workshop, 2000

[109] Henderson D.S., "A three phase electronic load governor for micro hydro generation", Ph.D. Thesis, University of Edinburgh, June 1992

[110] Pearson W.N. and Buchanan W.J., "Intelligent Control of Large Scale Plant", PREP'99 Conference, Manchester 1999.

[111] Irwin G.W., "Computing & Control: Back to the Future", Computing & Control Engineering Journal, February 1998.

[112] Davidson C. and McWhinnie J., "Stepping off the Ladder", IEE Review, September 1997.

[113] Honeywell Inc., "Total Plant Solution System Overview", Publication TP01100, November 1998.

[114] Bond A., "An Enabling Technology for Fundamental Change", IEE Review, November 1996.

[115] Bodington C.E., "Planning, Scheduling and Control Integration in the Process Industries", McGraw Hill, 1995

[116] Yazdi H., "Control and Supervision of Event Driven Systems", Ph.D. Thesis, Denmark Technical University, February 1997

[117] McLeister, L.D., "Surge Prevention Control System for Dynamic Compressors", US Patent No. US5798941, 25/5/98