

**The Validity of Using Neural Networks as a Predictor of the
Liquid-Mass-Fraction of the Refrigerant Exiting an
Evaporator**

James Solberg

DESC 9105
June 2000
Fergus Fricke
David Gunaratnam

PREFACE

This purpose of this document is to investigate the validity of using a neural network in a control system that regulates the liquid-mass-fraction (LMF) of the refrigerant exiting an evaporator. Part one of this document presents a brief background on vapor-compression refrigeration systems. Then it poses a problem with the current control system. A sensor and a control system are proposed that will solve this problem. If the reader is familiar with refrigeration control schemes and the thermal sensor presented, then (s)he should proceed to section 1.3. Section 1.3 attempts to use the data from the sensor to estimate the LMF of refrigerant, and to use that signal as the feedback signal for a controller. Part 2 of this document uses an alternative analysis tool to accomplish the same thing as was done in section 1.3. Part 2 jumps right into the neural network analysis.

ABSTRACT.....	4
Part 1: Background and Preliminary Analysis	5
1.1 Background	5
1.1.1 Expansion Devices.....	5
1.1.2 Operating with Finite LMF at the Exit of the Evaporator	6
1.1.3 Motivation	7
1.2 Sensor.....	9
1.2.1 Hot Wire Anemometry.....	9
1.2.2 Design Considerations.....	10
1.2.3 Constant Temperature Sensor.....	11
1.2.4 Theory	15
1.3 Estimation of LMF	17
1.3.1 Experimental Procedure	17
1.3.2 Data Collected	17
1.3.3 Relationships Among the Parameters	19
1.3.3.1 RMS.....	19
1.3.3.2 Constant Power	20
1.3.3.3 Constant LMF	22
1.3.3.4 Surface-to-Free-Stream Thermal Conductance (hA)	22
1.3.3.5 Mass Flow Rate Effects.....	26
1.3.3.6 Significance of Results	28
1.4 Precautions with this Analysis: Motivation for Better Analysis	29
Part 2: Using Neural Networks to Predict LMF	31
2.1 Relevant Variables.....	31
2.2 Missing Data.....	34
2.3 Choosing the Training, Verification, and Testing Data Sets	34
2.4 Statistical Information on the Various Case Types	37
2.5 Using “Bound” Cases	41
2.6 Correlation Matrix	41
2.7 Reduction of Inputs.....	46
2.8 Determining the Number of Hidden Layers.....	51

2.9 Comparing the Neural Network Model with Original Model.....	54
2.10 Further Simplification.....	60
2.11 Other Types of Networks	63
2.12 Confidence of Results	66
2.13 Improving the Analysis.....	67
Conclusion	70
BIBLIOGRAPHY	71

ABSTRACT

As the digital age engulfs our society, more and more devices surrender to the inevitable fate of digital control. Routinely, digital electronics replace traditional mechanical systems usually yielding an improvement in cost, size, weight, durability, performance, repeatability, and power consumption. As of the date of this document, no commercially available automobile is equipped with a digitally controlled throttling device for their air conditioning system. A primary reason for this is economics. Automotive manufactures cannot justify the additional costs associated with a microcontroller and an electronically controlled throttling device, even if they significantly improve performance and durability. As electronics become smaller, cheaper, “smarter”, and faster, electronic alternatives to traditional systems become increasingly prevalent.

Most techniques of actively controlling the performance of vapor-compression air conditioning system use evaporator superheat as the feedback parameter. Unfortunately, any amount of superheat causes the evaporator to operate at reduced capacity due to dramatically lower heat transfer coefficients in the superheated region. This document presents and defends a system that allows a vapor-compression air conditioning system to be stably controlled in a regime where liquid and vapor refrigerant are exiting the evaporator. The uniqueness of this system is attributed to the feedback transducer. The transducer is able to deliver a signal to the controller that is a function of the amount (by mass) of liquid droplets impinging on the transducer. By placing the transducer in the stream of refrigerant exiting the evaporator, a refrigerant throttling device can be manipulated to regulate the amount of liquid refrigerant that impinges the feedback transducer. With the signal from this transducer as the feedback in a control scheme, a controller can be constructed that essentially regulates system performance, and is able to control the system in regimes where superheat feedback is unable to operate.

Part 1: Background and Preliminary Analysis

1.1 Background

An understanding of several air conditioning and refrigeration concepts is essential for proper interpretation of the research presented in this document.

1.1.1 Expansion Devices

A schematic flow diagram showing the basic components of the vapor compression refrigeration system is shown in figure 1.1. Refrigerant fluid circulates through the piping and equipment in the direction shown.

Process 1 - 2. At point (1), the refrigerant is in the liquid state at a relatively high pressure and high temperature. It flows to (2) through a restriction, called the flow control device or expansion device. The refrigerant loses pressure going through the restriction. The pressure at (2) is low enough to allow a small portion of the refrigerant to flash (vaporize) into a gas. In order to vaporize, the refrigerant must absorb enough heat to overcome the refrigerant's heat of vaporization (which it takes from that portion of the refrigerant that did

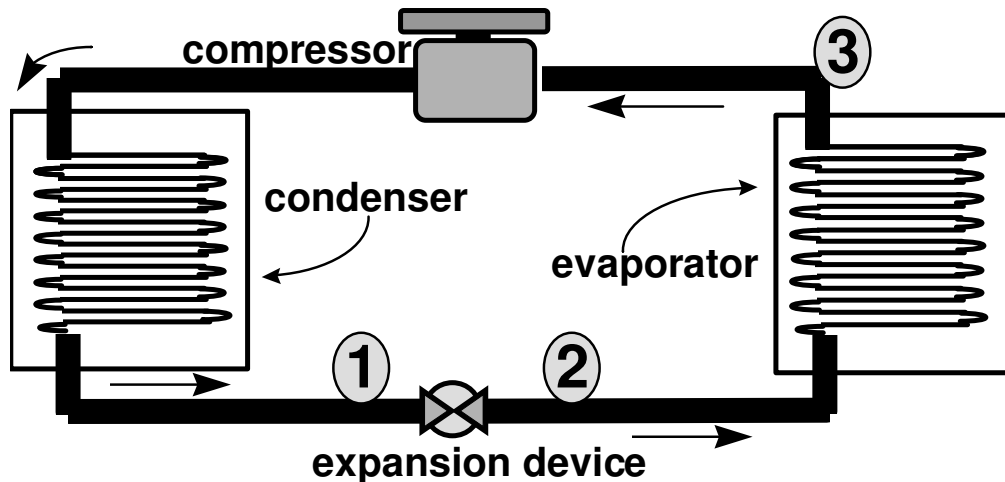


Figure 1.1. The vapor compression refrigeration system.

not vaporize), thus cooling the mixture and resulting in a low temperature at (2).

Process 2 - 3. The refrigerant passes through the evaporator. During this time, the refrigerant vaporizes, taking heat from the environment. Ambient fluid passing over the evaporating refrigerant will lose sensible heat. The heat is absorbed as latent heat and vaporizes the refrigerant.

The expansion device can play a dual role. The most obvious purpose of an expansion device is to expand some of the liquid into gas by imposing a pressure drop. The expansion device can also be responsible for regulating the refrigerant flow according to the load.

One of the two most common methods of refrigerant control in automobiles is the orifice tube. The orifice tube is only able to exhibit a fixed restriction, and is not able to actively control the flow of refrigerant. The orifice tube has been traditionally used because of its low cost and high reliability due to the fact that it has no moving parts.

The second traditional method of refrigerant control in automobiles is the thermostatic expansion valve (TXV). The TXV is able to actively change the flow of refrigerant as needed by the system. A bulb filled with a refrigerant fluid is in thermal contact with the outlet of the evaporator. It senses and responds to the temperature at that point. This bulb acts as a thermal motor such that increasing pressure of the fluid in the bulb due to vaporization tends to open the valve more, against a closing spring pressure. If the load in the system increases, then the refrigerant in the evaporator picks up more heat, and the temperature and pressure of the fluid in the bulb increases. This action is able to open the valve more to handle the increased load. The disadvantage of TXVs is that they are less reliable than orifice tubes due to their moving parts and their relatively fragile construction.

Electronic expansion valves have not been traditionally used for automotive applications due to the expense on such a system. An electronic expansion valve (EEV) system requires a computer to monitor the temperatures at the inlet and the outlet of the evaporator. From this feedback, the computer actively controls the valve setting. The versatility and efficiency of an EEV make it ideal for an automotive air conditioner, but, to date, the cost has been too great.

1.1.2 Operating with Finite LMF at the Exit of the Evaporator

The controlled variable in most evaporator control schemes (TXV, EEV, float valves, etc.) is the superheat at the evaporator exit. The reason is obvious. Superheat can be controlled by valve position and superheat can be determined by a simple temperature measurement. The superheat value used is a function of the evaporator type, load, and

distribution. Typical values are on the order of 4-8°C. A certain minimum superheat is needed for reliable operation of the system. If the degree of superheat is too small, the temperature signal from the evaporator outlet becomes very erratic leading to expansion valve hunting. Operation with superheated vapor exiting the evaporator is usually detrimental to evaporator performance since heat transfer is very poor in the superheated region.

Evaporator performance would be better if exit refrigerant had some low LMF. This principle is demonstrated in liquid overfeed evaporators. Evaporator operation without superheat or with some liquid droplets at the exit would potentially increase heat transfer on the refrigerant side and for a given system capacity, increase evaporating temperature. Of course, there must exist an optimum balance between an increase in evaporator capacity and lost refrigeration capacity due to liquid carry-over. Further, as evaporator exit LMF increases, a level that can damage the compressor will be reached.

1.1.3 Motivation

A traditional method of controlling evaporator superheat in a vapor compression air conditioning system is the thermostatic expansion valve (TXV). Such systems are often used in automotive applications. The TXV depends on superheat to adjust the valve opening. Unfortunately, any amount of superheat causes the evaporator to operate at reduced capacity due to dramatically lower heat transfer coefficients in the superheated region. In addition, oil circulation back to the compressor is impeded. The cold lubricant almost devoid of dissolved refrigerant is quite viscous and clings to the evaporator walls. A system that could control an air conditioner to operate with no superheat would either decrease the size of its existing evaporator while maintaining the same capacity, or potentially increase its capacity with its original evaporator. Also, oil circulation back to the compressor would be improved. To operate at this two-phase evaporator exit condition a feedback sensor would have to quantify the quality or the mass fraction of liquid in a liquid-superheated vapor stream of the refrigerant exiting the evaporator.

One of the most common control schemes for a vapor compression air conditioning system is the use of a thermostatic expansion valve (TXV). TXV systems use a remote thermal bulb at the exit of the evaporator. This bulb causes the TXV to open and close in response to changes in superheat of the refrigerant at the evaporator outlet. If the temperature of the refrigerant increases rapidly, as would be the case when the heat load was suddenly increased, the power element would open the valve and admit more liquid refrigerant to the

evaporator. Once in the evaporator, the liquid refrigerant absorbs heat by changing state from liquid to gas. By the time it leaves the evaporator, the gaseous refrigerant has been superheated a few degrees.

By allowing the evaporator to operate with some non-zero superheat at its exit, some portion of the evaporator will have only vapor flowing through it (no liquid). This situation decreases the refrigerant-side heat transfer. This portion of the evaporator is not able to vaporize refrigerant, and is only able to transfer heat via the sensible heating of the refrigerant. This process can reduce the capacity of the evaporator.

Any control scheme that uses superheat as its control signal (e.g. TXV systems) must have some non-zero superheat. Such a system is unable to control the plant to operate in a regime of saturated liquid/vapor at the exit of the evaporator. The minimum amount of superheat that such a system can use and maintain stability is dependent on the method of measuring the superheat.

The difficulty of a temperature measurement is in part due to the non-equilibrium flow of refrigerant as it exits the evaporator only slightly superheated. The flow is said to be non-equilibrium because saturated liquid droplets are entrained in superheated vapor. There is just not enough time for the liquid to vaporize and reach equilibrium. This phenomenon can be attributed to mal distribution of liquid/vapor refrigerant throughout the evaporator and to the nature of two-phase flow [1,29,34]. The saturated liquid droplets in superheated vapor flow regime cause temperature transducers to exhibit large variances.

In evaporators with imperfectly distributed exit streams, a mixture of superheated vapor and droplets often exits the evaporator. Some channels or circuits that are thermally overloaded have superheated vapor at the exit, while others where thermal loads are not sufficient to evaporate all liquid that enters will have some droplets at the exit. The mixture of these streams is in thermal non-equilibrium. After sufficient time (or length of pipe) the droplets would completely evaporate, reducing superheat. But if the sensible heat available in the superheated vapor is not enough energy to vaporize all droplets, then the exit stream is in the true two-phase quality region. Liquid-mass-fraction (LMF), which is the mass of liquid in vapor of any state, is a measure to describe the state at the evaporator exit, as described in Shannon, Hrnjak, and Leicht [23]. LMF is defined as

$$LMF = \frac{m_{liquid}}{m_{liquid} + m_{vapor}} \quad (\text{eq 1.1})$$

A temperature transducer measuring the temperature of refrigerant in this non-equilibrium flow regime can read the saturation temperature (if a liquid droplet is on the transducer), or can read the temperature of the superheated vapor (which may not be constant), or can read any value in between. A large variance in a control signal (e.g. superheat) can cause a controller to hunt. Since the non-equilibrium flow has superheated vapor along with liquid droplets, quality cannot be used to correctly describe the state of the refrigerant.

1.2 Sensor

The foundation of this research is based on the design and development of a sensor that will be used as a feedback signal in a controller. While sensors similar to the one described here have been developed, the document will highlight the uniqueness of this particular sensor. The physics of this sensor are not unique. Many of the principles used to describe the behavior of the sensor are found in other systems.

1.2.1 Hot Wire Anemometry

The hot-wire anemometer has been in common use in fluid mechanics research for more than seventy years. It has often been used for estimating instantaneous fluid velocities. Some of the fundamentals of the hot-wire will be utilized in the development of the feedback sensor. The hot-wire anemometer is a thermal transducer. The principle of operation is as follows. An electric current is passed through a fine filament that is exposed to a cross flow. As the flow rate varies, the heat transfer from the filament varies. This in turn causes a variation in the heat balance of the filament. The filament is made from a material which possesses a reasonably large temperature coefficient of resistance, i.e. if the temperature of the filament varies, so also does its resistance and hence the Joule heating. The variation of resistance give rise to signals related to the variations in flow velocity or flow temperature. The hot-wire method can therefore be used to measure instantaneous velocities and temperatures at a point in a flow.

There are two modes of operation of a hot-wire system. The first is the constant current mode. Here the current in the wire is kept constant and variation in wire resistance caused by the flow are measured by monitoring the voltage drop variation across the filament. The second is the constant temperature mode. Here the filament is placed in a feedback

circuit that tends to maintain the wire at constant resistance and hence constant temperature. Fluctuations in the cooling of the filament are seen as variation in wire current.

One of the more fundamental papers in the area of constant temperature anemometry was written by James Miller and titled “A Simple Linearized Hot-Wire Anemometer” [23]. Miller presents a design for a hot-wire system that exhibits high stability, simple bridge adjustment, and a linearizer having an adjustable exponent and very high transfer function accuracy. Measured frequency response is in excess of 100 kilohertz for the bridge and 7.5 kilohertz for the linearizer.

Another relevant paper is titled “A Study of the Hot-Wire Anemometer for Measuring Void Fraction in Two Phase Flow” which was written by H. Toral [30]. The performance of the constant temperature hot-wire anemometer as a local void fraction meter was studied with freely rising bubbles of air and vapor in ethanol. The principle of void fraction measurement by the hot-wire anemometer technique is to monitor the different rates of heat dissipation from the probe in vapor and liquid phases. The fraction of time during which the probe detects the vapor phase can be considered as the local void fraction, given a sufficiently long observation time. The hot-wire signal gave an indication of the rate of heat transfer from the transducer. From this relative heat transfer rate, the percentage of liquid and vapor was estimated.

1.2.2 Design Considerations

In order for the feedback sensor to meet desired specifications, many aspects of the sensor must be analyzed. The sensor must be robust enough to survive the harsh turbulent environment of high-speed refrigerant flow. The sensor must be able to deliver a measurable signal that is a function of relative LMF of the fluid. The dynamic response must be adequate for a controller to use the sensor’s signal as its feedback.

An introduction to related work is presented in sections 1.4 and 2.1.1.2 of this document. Much of the research done on this project is based on the work of Shannon, Hrjnak, and Miller [9,12,24,26]. They built on the idea of a constant current hot-wire anemometer to develop a MEMS device that is able to qualitatively estimate the LMF of a stream of refrigerant. The design of this project’s transducer began with the principles and results of Shannon, Hrjnak, and Miller. From there, variations in design specifications were imposed on a new design that would ultimately result in a transducer better suited for its purpose.

Although constant current and constant temperature designs can sometimes perform the same task, the separate architectures exhibit unique characteristics. The constant current architecture is passively controlled (open-loop) with a constant current supply. A constant current is fed to the filament, and after the filament has reached a uniform thermal equilibrium the voltage drop across the filament is an indication of the temperature of the filament. The speed of the transducer's response is a function of the convective heat transfer coefficient between the sensor and the free stream, the free stream mean temperature, the current through the sensor, and the thermal mass of the sensor. The convective heat transfer coefficient between the sensor and the free stream and the free stream mean temperature are fixed by the conditions of the flow and fluid properties. The current through the sensor is limited by the minimum amount of current needed to burn out the transducer for a given set of conditions. Under normal operating conditions forced convection usually provides enough heat transfer so the transducer does not burn out. But if the same current flowed through a transducer in an environment devoid of forced convection, the possibility of burning out the transducer greatly increases. The thermal mass of the transducer is a function the transducer's geometry and its material. For a constant current architecture the thermal mass is the only parameter that can be changed to alter its speed of response (frequency response).

A constant temperature architecture actively drives the temperature of the filament to a constant temperature. A more descriptive name for the circuitry is a constant resistance architecture, because negative feedback to an amplifier forces the filament to a fixed resistance. All of the factors that affect frequency response of the constant current architecture affect the response of the constant temperature (resistance) architecture. But, since the constant resistance circuitry is driven by active components, the active components (usually operational amplifiers) will influence the frequency response. Because of this, constant temperature transducers can be developed with a larger bandwidth than constant current transducers.

1.2.3 Constant Temperature Sensor

This project chose the path of the constant temperature architecture to drive the transducer. This design proved to more robust, yielded a greater bandwidth, and was inherently self-protected against burnout.

For a sensor to properly function as the feedback for a controller used for regulating the liquid-mass-fraction (LMF) at the exit an evaporator, the sensor has to meet many specifications.

The sensor has to be robust enough to handle an environment where high-speed two-phase refrigerant rushes over it. The sensor has to withstand the endless bombardment of high-speed liquid droplets without failing.

The filament of the transducer has to be situated in the stream of refrigerant in such a fashion to maintain isolation from ambient conditions (temperature and pressure). The transducer must be designed to be as non-invasive as possible. Neither the sensor nor its supporting structure should impose a significant pressure drop or alter the flow dynamics in any other significant manner.

The frequency response of the sensor must be fast enough to provide accurate measurement to the controller. If the dynamics of the sensor are too slow, the control system could lose significant phase, which could result in instabilities in the controller.

The sensor should be simple to construct. For industry to find interest in this design it must be able to be made economically.

The transducer must deliver a measurable signal that is an indication of the magnitude of the liquid-mass-fraction of the refrigerant.

As previously mentioned, the construction of the sensor was similar to the constant

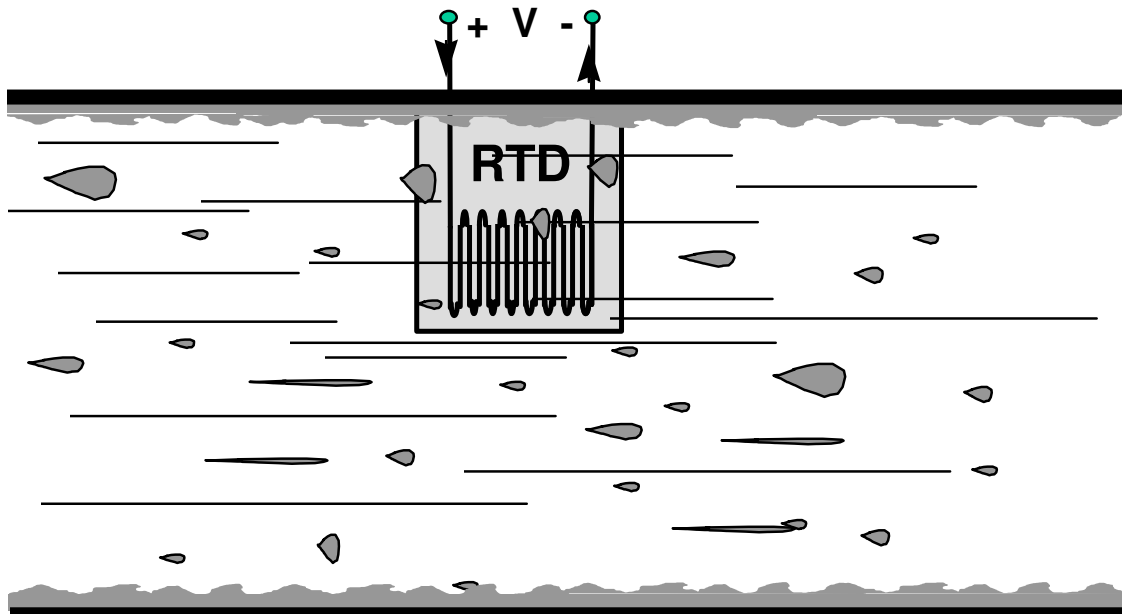


Figure 1.2. Cross-section view of the sensor placed in a pipe filled with moving refrigerant.

temperature hot-wire anemometer. Instead of a single isolated wire, this design used an RTD as the transducing element. The RTD is an off-the-shelf product, and therefore easy to obtain.

Figure 1.2 schematically shows what the sensor might look like in a stream of refrigerant. The RTD must be placed in the stream of refrigerant. Also, wires must carry voltage to and from the sensor while keeping the sensor sealed from ambient pressure so that high pressure refrigerant does not leak out. To accomplish this the wires to the RTD were fed through a 1/8 inch O.D. aluminum tube. The wires were 24-gauge and coated with polytetrafluoroethylene (trade name Teflon). Because of the limited space inside the 1/8 inch aluminum tube, TeflonTM coated wire is used because Teflon is an excellent insulator, and a larger percentage of the cross-sectional area of the wire can be used for conductor as opposed to insulation. Finally, Teflon acts as an excellent solid lubricant that easily allows the wires to be slid through the tube.

Figure 1.3 shows what the schematic of the device might look like. At one end of the tube the wires were soldered to the leads of the RTD. Epoxy was placed at the base of the RTD to seal it from the inside of the tube and to hold the RTD in place. No epoxy was placed on the transducing element of the RTD (filament head). Epoxy is also placed at the other end of the tube to seal off the ambient pressure from the inside of the tube. The tube itself is sealed off with SwedgelockTM compression fittings.

As a result of the physical nature of the conduction of electricity, electrical resistance of a conductor or semiconductor varies with temperature. Using this behavior as the basis for temperature measurement is extremely simple in principle, and leads to two basic classes of resistance thermometers: resistance temperature detectors (conductors) and thermistors (semiconductors). Resistance temperature detectors (RTD) may be formed from a solid metal wire, which exhibits an increase in electrical resistance with temperature. The physical basis for the relationship between resistance and temperature is the temperature dependence of the resistivity of the material.

The most common material chosen for the construction of RTDs is platinum. The platinum RTD provides a means for the measurement of temperature, and historically provided the first interpolation standard for an internationally acceptable temperature scale.

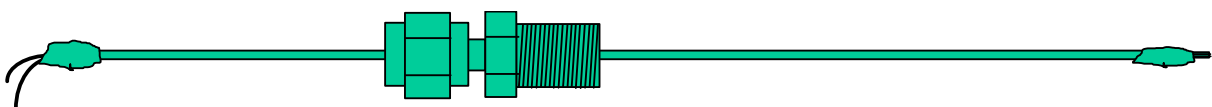


Figure 1.3. *schematic of the sensor.*

The RTD relies on the change in electrical resistance of a platinum wire to provide a precise measure of temperature. The linear approximation for the relationship between temperature and resistance is valid over a wide temperature range, and platinum is highly stable.

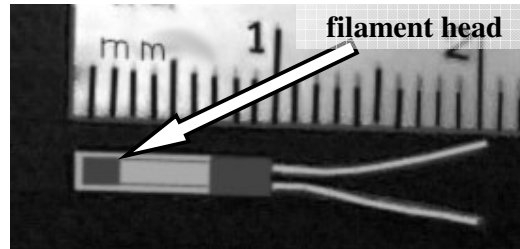


Figure 1.4 Photograph of the thin-film RTD used in the sensor. The dimensions of the RTD are 10mm x 2mm x 1mm thick.

The specific RTD used in the construction of the sensor was purchased by Omega. Omegafilm Platinum RTD (catalog number F3101) are small, thin, flat ceramic encased elements with strain relieved silver palladium alloy leads approximately 0.4" long. Temperature rating from -50 to 600°C. Calibration is to DIN standard 43760, 0.1% tolerance. It has a single winding configuration with 100-Ohm resistance at 0°C. The alpha (the temperature coefficient of resistivity) is 0.00385. The dimensions of the RTD (in millimeters) are 10 long, by 2 wide, by 1 thick. The filament head is the portion of the RTD that has the actual transducing element exposed. The filament head is 3mm long by 2mm wide.

For the transducer to behave correctly circuitry must be used to keep the RTD at a constant temperature (see Figure 1.5). This circuit tries to keep the resistance of the RTD equal to R_{set} . The voltage V_o is then directly proportional to the current needed to achieve this condition. The power removed by heat transfer into the refrigerant stream is, of course, the square of the current flowing through the RTD times the RTD resistance (or R_{set}). This circuit uses an operational amplifier as the medium for feedback. The op-amp uses the feedback to maintain its inputs at constant voltage while drawing very little current. This is what forces the resistance of the RTD to be equal to the resistance of R_{set} . Traditionally, an RTD is used to measure temperature by measuring the resistance of the RTD as it changes with temperature. But, this circuit forces the resistance of the RTD to be equal to R_{set} . The circuit compensates by heating up the RTD until the resistance (and thus the temperature) of the RTD is equal to R_{set} .

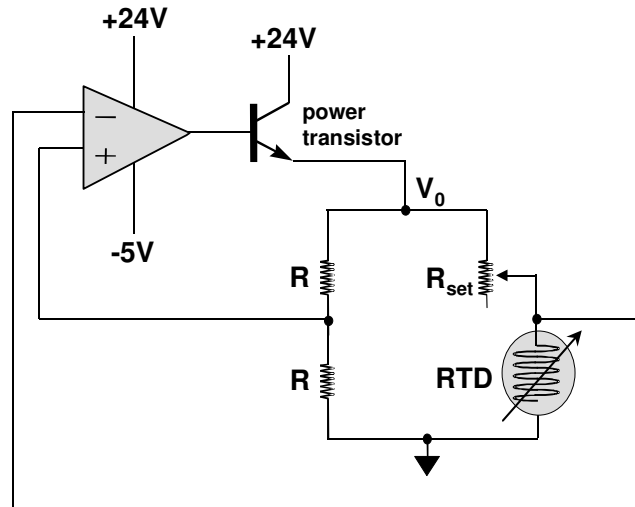


Figure 1.5 Constant sensor

1.2.4 Theory

Compared to the constant current architecture, this constant temperature design has a much wider bandwidth (that is, it will respond to much higher rate variations in heat flux). The reason is as follows. Constant current excitation requires that the transducer temperature changes for any change in transducer resistance and hence signal to be observed. This is an inherently slow (relatively long time constant) process dominated by the thermal capacity of the transducer body. Constant resistance operation implies that the circuitry varies the transducer current so that the transducer stays at a constant resistance and hence a constant temperature. The thermal energy stored in the transducer body does not change. This technique is used with hot wire anemometers and provides very broadband performance (bandwidths up to 0.5 MHz). The technique also has the advantage of protecting the sensor from overheating.

The circuit (as shown in Figure 1.5) maintains the voltage drop across the RTD equal to half of V_0 . Since R_{set} is equal to the resistance of the RTD, the power dissipated through the RTD can be determined. By measuring the temperature of the refrigerant passing over the sensor and inferring the temperature at the surface of the RTD from R_{set} , the difference of these temperatures can be found. This document refers to this temperature difference as *overheat*. The *overheat* represents the driving potential that allows power to be dissipated through the sensor.

Figure 1.6 shows the RTD sensor in the stream of refrigerant. Current is driven through the filament of the RTD. The i^2R power loss through the RTD is transferred to the

refrigerant. The voltage drop across the RTD is measured. The power dissipated through the sensor is a function of its voltage drop and its resistance (which is known). The temperature of the free-stream refrigerant is measured by a thermocouple. These measured and computed parameters will be used to estimate the LMF of the refrigerant passing the sensor.

The ratio of the power dissipated to this temperature difference can be interpreted as "the surface-to-free-stream thermal conductance" between the RTD and refrigerant. It is essentially the convection heat transfer coefficient multiplied by the effective surface area (hA). This surface-to-free-stream thermal conductance (hA) does not depend on the effective surface area of the sensor because neither the geometry nor the orientation of the sensor varies. This hA parameter is particularly sensitive in the high quality/low superheat region (low LMF). As the LMF of a fluid increases, so does its hA .

As a droplet of saturated liquid refrigerant clings to the surface of the RTD, the RTD circuitry will do what it can to raise its temperature back its set point (which is determined by R_{set}). To do this the RTD must transfer enough energy to the refrigerant to overcome its latent heat of vaporization. As the LMF of the fluid decreases, less energy is dissipated through the RTD. When the fluid becomes all vapor, all of the energy flux through the RTD goes to sensible heat that is needed to raise the temperature of the RTD to its set point.

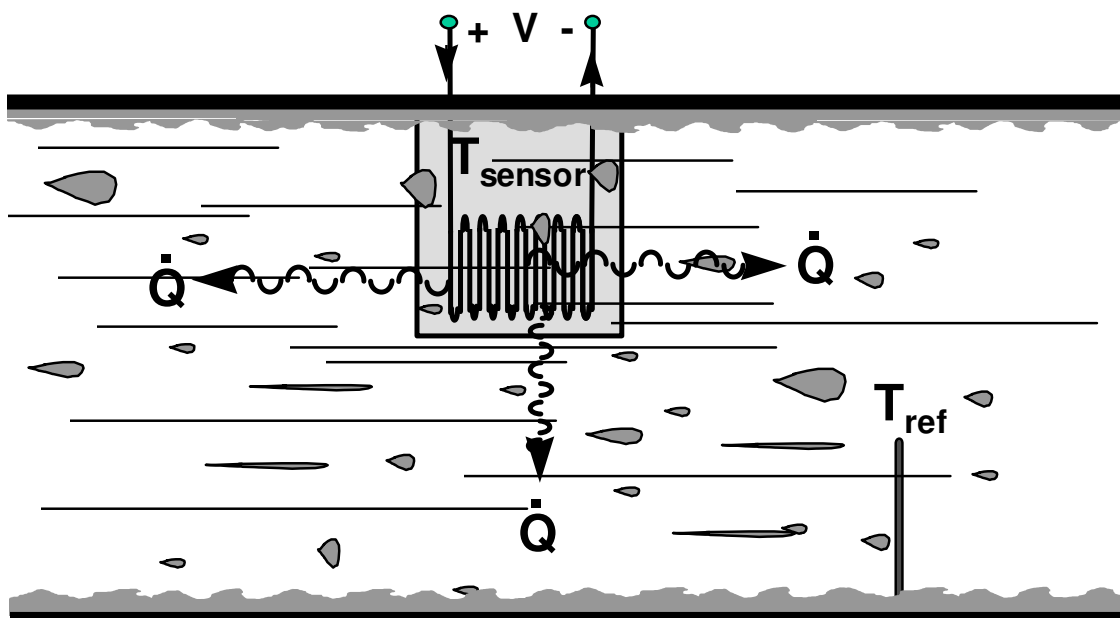


Figure 1.6 schematic of the sensor in the flow of refrigerant.

1.3 Estimation of LMF

In order for the sensor to be useful some relationship between refrigerant LMF, overheat (temperature difference), and power dissipated through the sensor needs to be developed. LMF is determined from the enthalpy of the refrigerant entering the calorimeter. Overheat is the difference between the temperature of the RTD and the temperature of the refrigerant. The power dissipated through the RTD is determined by the square of the RTD's voltage drop divided by its resistance

1.3.1 Experimental Procedure

In principle the experiments used in this study were straightforward. The loop was run under various operating conditions that yielded a non-zero LMF of refrigerant exiting the evaporator. The LMF of the refrigerant passing over the sensor was assumed to be equal to the LMF of the refrigerant entering the calorimeter. The power dissipated through the sensor was estimated as the square of the voltage across the sensor divided by its resistance. The temperature of the sensor is known because its resistance is known, and the temperature of the free-stream refrigerant is measured with a thermocouple. The difference between the temperature of the sensor and the temperature of the free-stream refrigerant is an estimate of the overheat.

During operation of the loop the LMF, power dissipated via the sensor, and overheat are measured and recorded using the data acquisition system described in [27]. These parameters will initially be assumed independent with respect to each other. Then correlations will be formed to approximate the interdependence of the variables. These correlation(s) will be used to describe how the overheat and the power dissipated through the sensor can be used to predict the LMF of refrigerant passing over the sensor.

1.3.2 Data Collected

Data was primarily collected during operating conditions under which the RTD sensor would most likely be operating. Table 1.1 shows all the data collected with a refrigerant mass

flow rate of approximately 50 g/sec. The voltage refers to the voltage across the RTD sensor, and Rsensor is the resistance of the RTD (which also infers the temperature).

Table 1.1 All trials taken with a refrigerant mass flow rate of 50 g/sec.

	<i>overheat (C)</i>	<i>voltage (V)</i>	<i>Rsensor (ohms)</i>	<i>power (Watts)</i>	<i>LMF</i>
	7.0	9.42	103.0	0.861	0.221
	4.7	7.23	102.0	0.512	0.22
	3.2	5.29	103.0	0.271	0.139
	5.8	7.64	103.0	0.567	0.139
	5.1	6.01	102.3	0.352	0.124
	9.2	8.76	104.0	0.738	0.118
	5.0	6.64	103.0	0.428	0.114
	2.4	4.47	102.0	0.195	0.113
	7.4	8.01	103.8	0.618	0.112
	5.3	6.07	102.0	0.361	0.1
	4.2	5.99	102.0	0.352	0.078
	8.5	8.49	104.0	0.692	0.077
	5.3	6.03	102.6	0.354	0.07
	5.8	5.73	102.0	0.322	0.07
	6.5	7.27	103.0	0.512	0.062
	5.6	5.73	102.0	0.321	0.06
	10.6	8.65	103.8	0.721	0.06
	8.7	8.26	103.8	0.657	0.059
	9.4	8.45	104.6	0.682	0.042
	6.4	4.88	102.0	0.233	0.03
	11.4	7.66	103.8	0.565	0.03
	12.1	7.79	103.8	0.585	0.03
	13.6	10.17	105.9	0.977	0.03
	5.89	6.04	103.3	0.353	0.027
	10.17	8.35	105.0	0.664	0.026
	6.44	6.05	103.6	0.353	0.01
	11.5	6.91	103.8	0.459	0.01
	12.2	6.81	103.8	0.446	0.001
	17.2	8.71	105.9	0.716	0.001
average	7.81	7.15	103.34	0.513	0.075
std dev	3.43	1.43	1.11	0.196	0.058
max	17.20	10.17	105.90	0.977	0.221
min	2.40	4.47	102.00	0.195	0.001

1.3.3 Relationships Among the Parameters

The data was collected in hopes of developing some relationship the parameters. During the developmental stages of the experiments there was no accurate model to describe the relationships among the parameters. Once the data was collected, statistical correlations were established, then a model was developed to incorporate those correlations.

1.3.3.1 RMS

The root-mean-square (RMS) is a mathematical way to characterize the variance of a waveform. It represents the average value of the square of the signal over the measured time period. The RMS value of any continuous analog variable $y(t)$ over the time, $t_2 - t_1$, is expressed as

$$y_{RMS} = \sqrt{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} y^2 dt} \quad (\text{eq 1.2})$$

Early in the development of a control scheme to regulate the LMF of refrigerant exiting the evaporator, it was thought that the RMS value of the signal coming from the RTD sensor would be an indication of the LMF of the refrigerant. The reasoning was as follows. During superheated vapor flow the signal from the RTD sensor would be fairly constant because the refrigerant hitting the RTD sensor is single phase. As the first droplets hit the sensor, the RMS signal would be expected to become significant because every time a droplet hits the RTD sensor, the sensor has to deliver more power to its surface to vaporize the droplet. As more and more droplets hit the sensor, the RMS signal would increase until the RTD sensor would become unable to deliver enough power to vaporize all the liquid on the sensor.

In order to test this theory the data acquisition system also collected RMS data from the RTD sensor. Figure 1.7 shows the RMS of the voltage signal from the RTD sensor for various LMFs. The plot shows that the RMS of the voltage signal from the RTD sensor is not dependent on the LMF of the refrigerant.

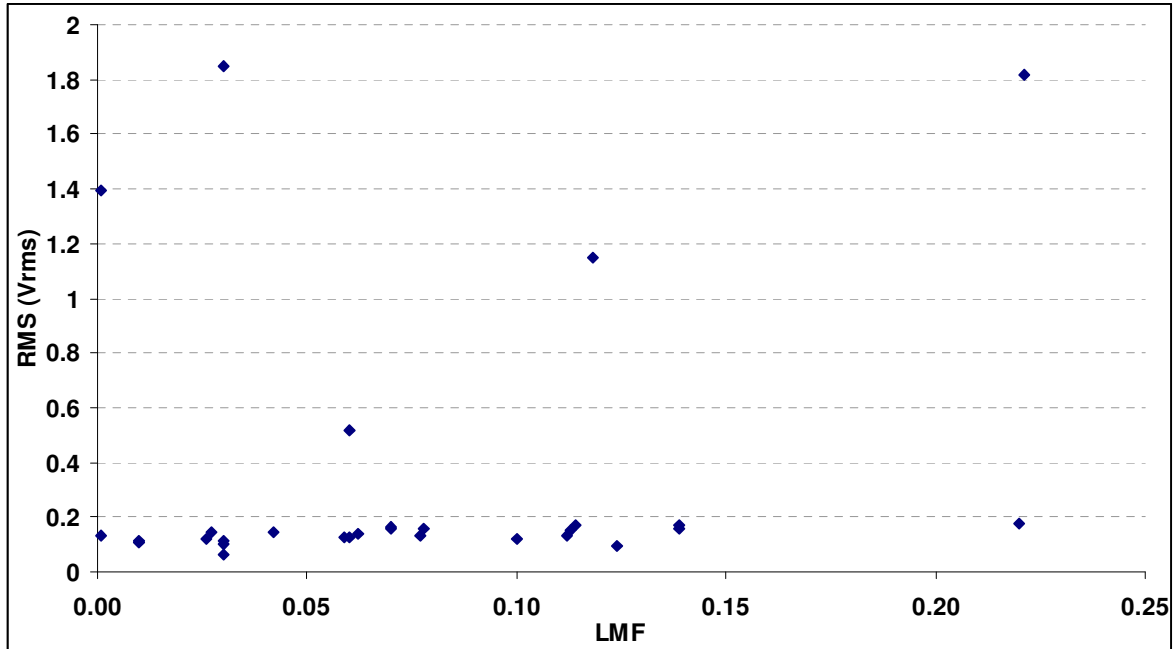


Figure 1.7 The RMS of the voltage signal from the RTD sensor as a function of the LMF of the refrigerant.

1.3.3.2 Constant Power

Recall that the purpose of these experiments was to develop some relationship between overheat, power through the sensor, and LMF of refrigerant. One of the most effective procedures for developing relationships between parameters is to simply plot them on independent axis, and make visual observations. Many times these visual observations quickly reveal the interdependence of the variables. Computer graphics have made it easier to visualize three independent dimensions on a single graph by artificially creating the illusion of 3-D objects projected onto a 2-D plane (computer monitor or paper). But it is still much easier to see trends in 2-D. In order to analyze only two of the three parameters at a time one of the parameters must be held constant as the other two change. Then the data can be plotted in two dimensions.

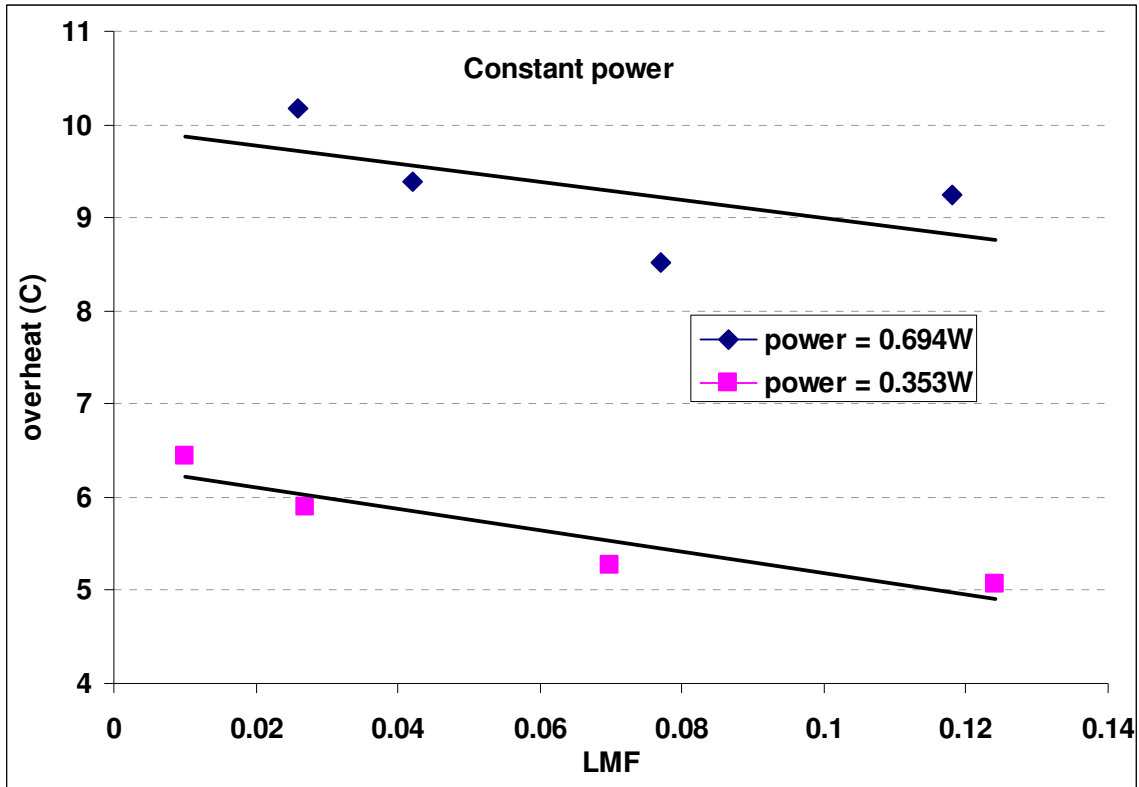


Figure 1.8 *Overheat vs. LMF for constant sensor power dissipation*

Figure 1.8 shows overheat as a function of the LMF of the refrigerant while the power dissipated through the sensor is held constant. Two important conclusions can be inferred from figure 1.8. First, for a constant amount of power dissipated through the RTD, as the LMF increases, the overheat decreases. Second, for a given refrigerant LMF, if the power dissipated through the sensor is increased, then the overheat must be increased.

Both of these conclusions can be explained from the same phenomenon as follows. Suppose the overheat represents the heat transfer driving potential. (Recall that the overheat is the difference between the temperature at the surface of the RTD sensor and the temperature of the free-stream refrigerant.) As the temperature difference between any two points increases, the potential for heat transfer between those two points also increases. But, increasing the amount of liquid refrigerant in the stream (increase in LMF) also increases the potential for heat transfer, because liquid refrigerant can absorb heat much better than vapor refrigerant. These are competing effects when the power dissipated through the RTD (the amount of heat transfer) is held constant. As a result of this logic, the overheat and LMF of refrigerant have a negative correlation.

By similar logic it can be shown that for a given LMF, if the power dissipated is increased, then the overheat must increase. Without the LMF affecting the heat transfer rate the only way that the amount of heat transferred (power dissipated) can increase is to increase the overheat (driving potential for heat transfer).

1.3.3.3 Constant LMF

Experiments were conducted that would demonstrate the dependence of sensor power and overheat while the LMF was held constant. Figure 1.9 shows the results of such experiments. A linear least-squares line was fit to each set of data. The graph shows four different trials, where each trial has a unique LMF. While the lines do not perfectly fit each data set, they offer some insight to the qualitative behavior of the parameters. Also, the equation of each line is explicitly stated on the plot.

Stretching the linear interpretation of the data farther, it could be said that the data agrees with the convection heat transfer model that takes the form:

$$q = h \cdot A \cdot (T_s - T_\infty) \quad (\text{eq. 1.3})$$

T_s is the fixed temperature at the surface of the RTD. T_∞ is the temperature of the free-stream refrigerant passing over the sensor. The power dissipated can be modeled as the energy transfer q . The overheat in the system is analogous to $(T_s - T_\infty)$. And the slope of the line represents hA . hA is the product of the convection heat transfer coefficient and the effective surface area.

While this is the first time that this model has been formally stated, it has been alluded to in the previous section and throughout this document. Note that as the LMF decreases, the slope of its fit line also decreases. While neither the overheat nor the power by themselves are able to predict the LMF, *the ratio of the power dissipated through the RTD to the overheat does seem to give an indication of the LMF.*

1.3.3.4 Surface-to-Free-Stream Thermal Conductance (hA)

While equation 1.3 may seem like a valid model for the data, it does not totally describe what is happening at the surface of the RTD sensor. Recall that the RTD sensor has both liquid and vapor hitting its surface. These two different states of refrigerant remove heat

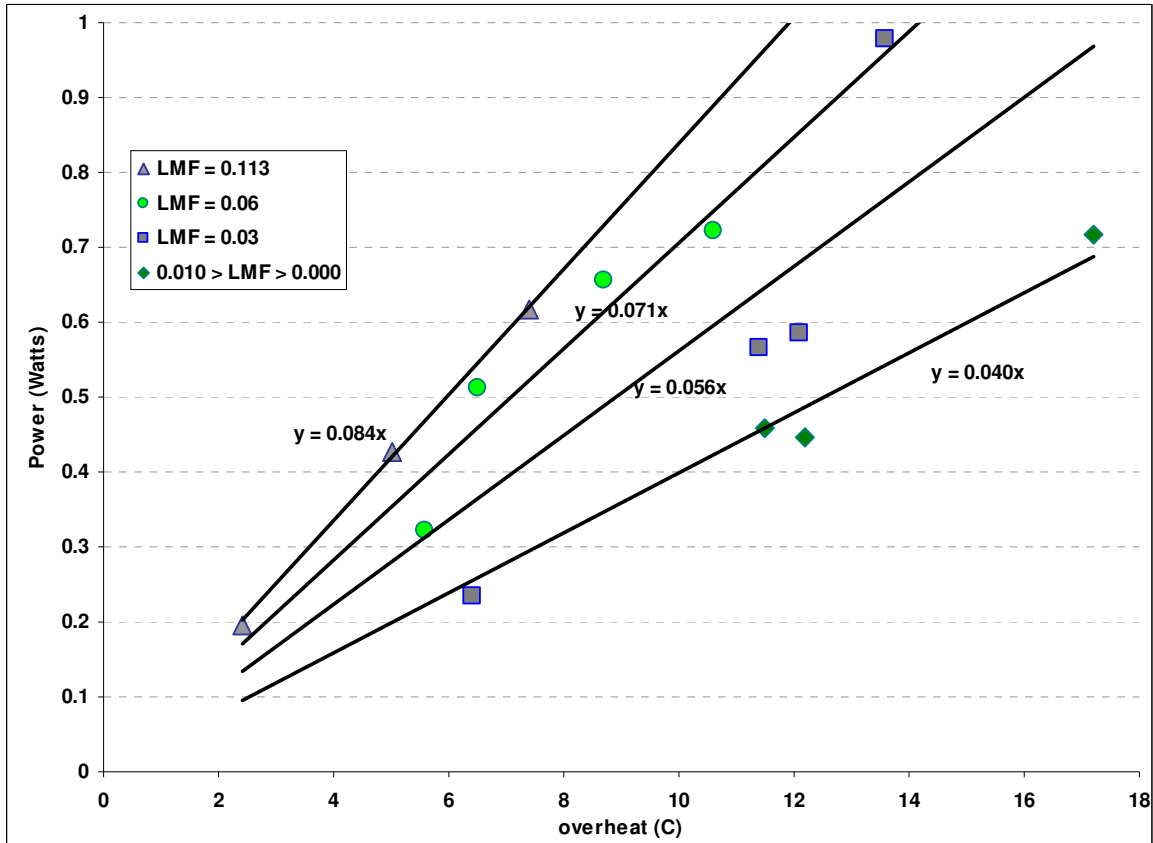


Figure 1.9 power dissipated through the sensor as a function of overheat while LMF was held constant for each trial.

from the transducer in very different ways. Perhaps a more accurate variation of equation 1.3 would be:

$$q_{total} = q_{vapor} + q_{latent} = h_v A_v (T_s - T_{mean}) + \dot{m}_{evap} h_{lv} \quad (\text{eq 1.4})$$

Shannon et al initially proposed this equation [23,24]. h_v is the convective heat transfer coefficient between the sensor and vapor. A_v is the effective surface area between the sensor and the vapor refrigerant. T_s is the temperature at the surface of the RTD. T_{mean} is the average temperature of the free stream. \dot{m}_{evap} is the liquid mass evaporation rate. And h_{lv} is the latent heat from liquid to vapor.

The heat transferred to the vapor is used to heat up the vapor, and the heat transferred to the saturated liquid is used to vaporize the liquid. While the different phases of refrigerant use the heat differently, the mode of heat transfer to each of the phases is fundamentally similar. Using this information, equation 1.4 can be restated:

$$q_{total} = q_{vapor} + q_{latent} = h_v A_v (T_s - T_{mean}) + h_l A_l (T_s - T_{sat}) \quad (\text{eq 1.5})$$

h_l and A_l are the convective heat transfer coefficient and effective surface area for the sensor/liquid interface. T_{sat} is the saturation temperature for the refrigerant.

Recall that the object of this project was to design a simple and robust control system capable of regulating the LMF of refrigerant exiting the evaporator. The purpose of designing this RTD sensor is to create a device that is able to qualitatively estimate the relative LMF of refrigerant passing over it. In order to make the control system as simple as possible it would be beneficial if a single feedback signal was fed into the controller. This signal would be derived from measurements taken from the RTD sensor. Section 1.3.3.3 showed that the slopes of the lines in figure 1.10 gave an indication of the refrigerant's LMF. As previously stated, this slope represents the ratio of the power dissipated through the sensor to the overheat. This ratio will be defined as the "surface-to-free-stream thermal conductance", and its formal definition will take the form:

$$hA = \frac{\text{power}_{\text{dissipated through RTD}}}{T_{\text{sensor}} - T_{\text{mean}}} \quad (\text{eq 1.6})$$

The power dissipated through the RTD and temperature of the sensor are quantities that can be measured quite accurately. It is not as easy to get an accurate measurement for the average temperature of the refrigerant stream. T_{mean} is measured by a thermocouple that is immediately up stream from the RTD sensor.

Once it was established that hA could be used as a measure of LMF, the next task was to develop a relationship between LMF and hA . Figure 1.10 shows the results of an experiment where the RTD sensor was subjected to various LMFs. LMF was measured by using a calorimeter in the method described earlier.

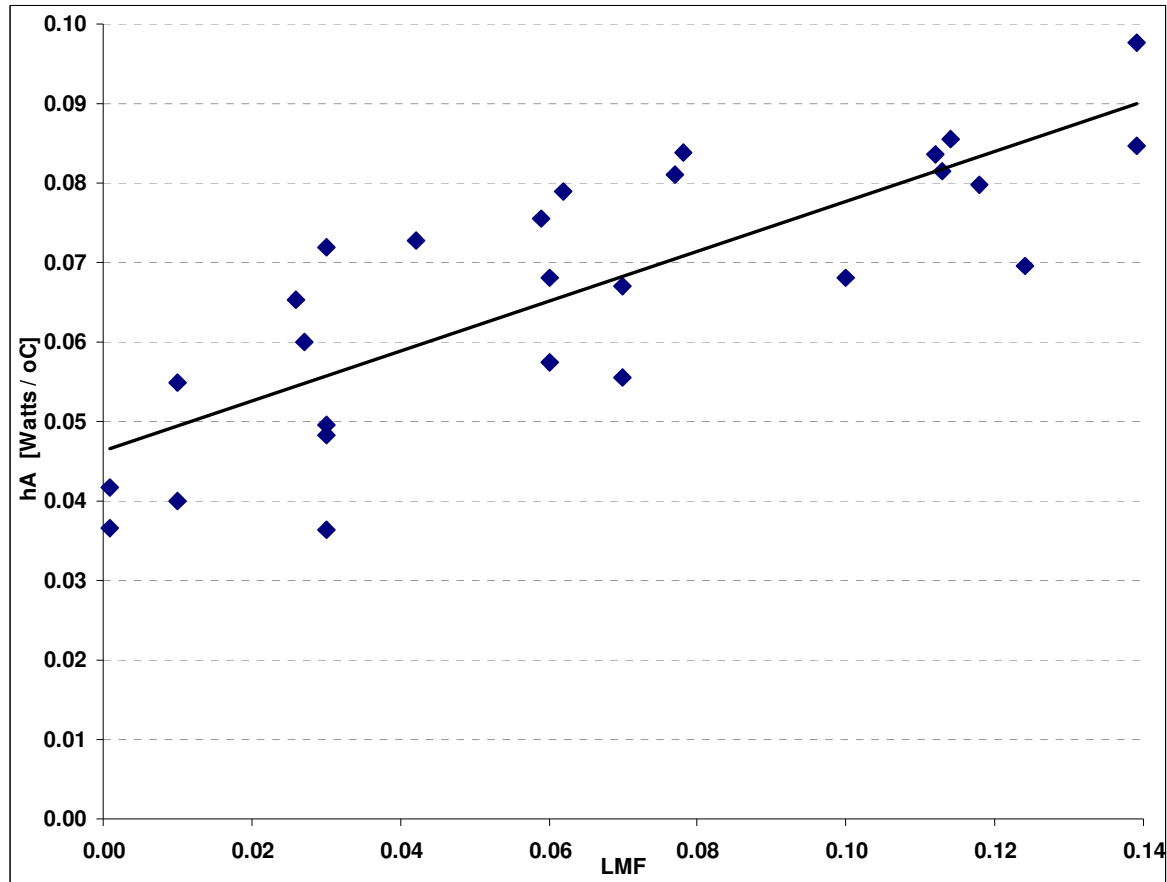


Figure 1.10 The surface-to-free-stream thermal conductance as an estimate of LMF.

As LMF increases more and more liquid droplets hit the sensor. This demands more power to be dissipated through the sensor in order for the sensor to maintain its constant temperature. At the same time the temperature of the refrigerant is becoming closer to being fixed at its saturation temperature. So the ratio of the dissipated power to the overheat (hA) increases as LMF increases. This theory is supported by the data shown in figure 3.11.

The hA defined in this document does not refer to the $h_v A_v$ or $h_l A_l$ in equation 1.5. Nor will it be defined analytically as a function of $h_v A_v$ or $h_l A_l$. hA is defined only as stated in equation 1.6. The objectives of this project do not encompass deriving an analytical solution for hA .

1.3.3.5 Mass Flow Rate Effects

All of the data presented so far have a consistent mass flow rate. The next set of experiments was conducted to investigate the effects of varying the mass of the refrigerant. All previous trials had a refrigerant mass flow rate of approximately 50 g/sec. The experiments presented in this section had a refrigerant mass flow rate of approximately 25 g/sec. The results of the experiments are summarized in table 1.2.

The data can also be presented as it was in figure 1.10. Figure 1.11 shows hA as a function of LMF for two different mass flow rates. The plot shows that lower mass flow rate will yield a lower hA for the same LMF. This is expected because for the same LMF, there will be less droplets hitting the sensor. Less droplets hitting the sensor mean a lower hA .

What was not expected is the shallower slope of the fit line. The explanation of this phenomenon relates to limitations of the experimental facility. Recall that the RTD sensor is in a section of the loop that is glass. The glass allows observation of the flow patterns around the RTD sensor. During the course of the experiments it was observed that the flow became stratified, and much of the refrigerant flowed beneath the sensor. For the previous experiments with refrigerant mass flow rates of 50 g/sec the flow was well distributed across the cross-section of the pipe, so the RTD sensor was subjected to a representative sample of the refrigerant flow composition. But, with lower mass flow rates the sensor did not get a representative sample of the refrigerant flow composition because most of the liquid was at the bottom of the pipe. As a result, the sensor lost sensitivity.

Table 1.2 Data collected with low mass flow rate.

	<i>overheat (C)</i>	<i>voltage (V)</i>	<i>power (Watts)</i>	<i>Rsensor</i>	<i>LMF</i>	<i>hA [W / °C]</i>
	7.61	4.96	0.239	102.9	0.114	0.0313
	10.10	5.62	0.304	103.9	0.115	0.0301
	15.09	6.90	0.450	105.9	0.112	0.0298
	20.08	7.89	0.577	107.8	0.115	0.0287
	25.10	8.48	0.655	109.8	0.113	0.0261
	30.04	9.17	0.753	111.7	0.112	0.0251
	9.13	5.31	0.271	103.9	0.077	0.0297
	14.22	6.45	0.392	105.9	0.070	0.0276
	19.95	6.92	0.444	107.8	0.073	0.0222
	24.29	7.79	0.552	109.8	0.069	0.0227
	29.30	8.77	0.688	111.7	0.074	0.0235
	8.80	4.57	0.201	103.9	0.031	0.0228
	13.73	5.65	0.302	105.9	0.036	0.0220
	18.75	6.39	0.378	107.8	0.033	0.0202
	23.71	7.06	0.454	109.8	0.029	0.0191
	28.72	7.97	0.569	111.7	0.030	0.0198
	6.38	3.14	0.095	103.9	0.006	0.0149
	12.70	5.58	0.294	105.9	0.007	0.0232
	17.55	6.50	0.392	107.8	0.007	0.0223
	22.67	7.52	0.515	109.8	0.010	0.0227
	28.09	8.40	0.631	111.7	0.010	0.0225
average	18.38	6.71	0.436	107.6	0.059	0.024
std dev	7.67	1.54	0.174	3.0	0.042	0.004
max	30.04	9.17	0.753	111.7	0.115	0.031
min	6.38	3.14	0.095	102.9	0.006	0.015

The important conclusion drawn from these experiments is that the RTD sensor must be in a flow pattern that is representative of the composition of the entire cross-sectional flow. One possible fix for the problem is to place the sensor closer to the exit of the evaporator. Usually refrigerant exiting the evaporator is turbulent and well mixed. As the refrigerant travels along a long horizontal tube, the liquid refrigerant falls to the bottom of the pipe. Another possible fix is to place the sensor in a vertical section of pipe. In this geometry the liquid has no preference across the cross-section of the pipe.

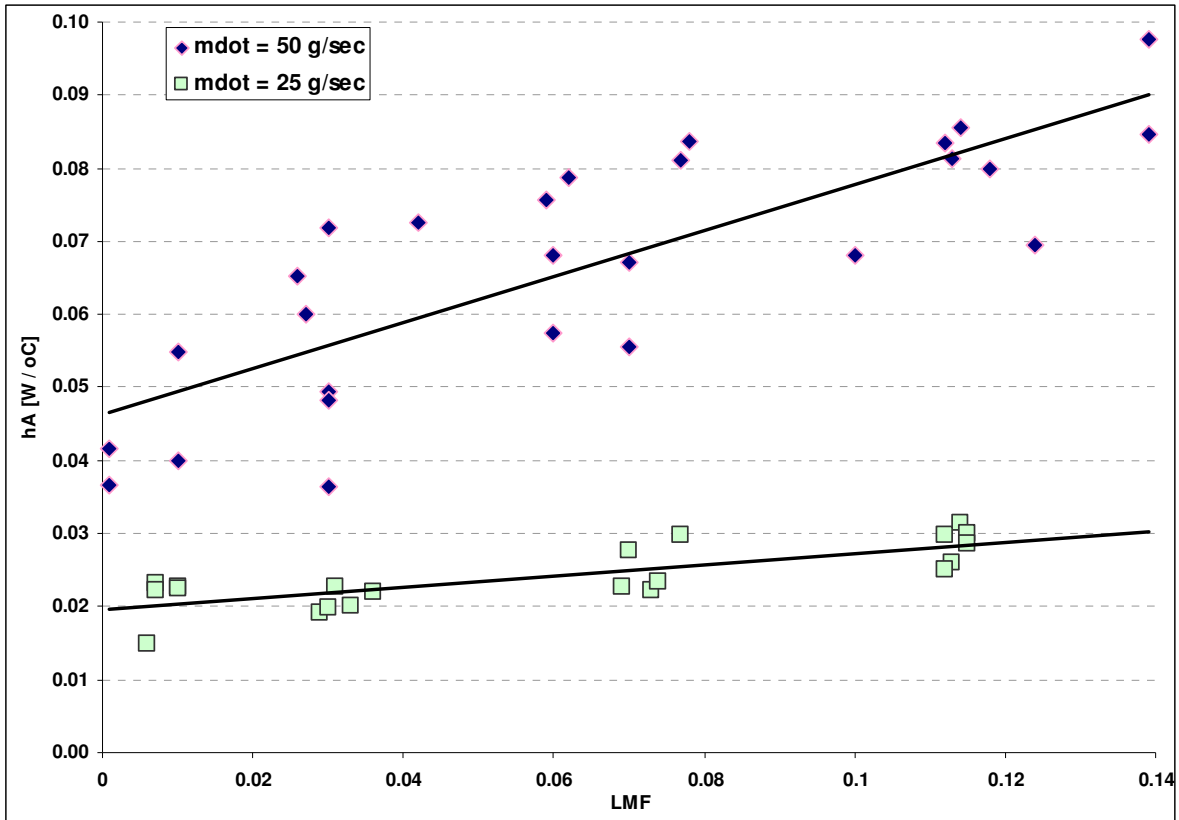


Figure 3.12 hA as a function of LMF for two different refrigerant mass flow rates.

1.3.3.6 Significance of Results

The experiments and data presented in the previous sections demonstrate that the RTD sensor could be used as the feedback sensor in a controller able to control the LMF of refrigerant exiting an evaporator. This is because the sensor is capable of estimating the LMF of refrigerant.

hA was defined so that it is simple to calculate from measurable parameters, and it is a measure of LMF. The important conclusion drawn from these sets of data is that hA is dependent on LMF. The exact relationship between hA and LMF is not important for the purposes of this project. What is important is that the RTD sensor can be empirically calibrated to estimate state parameters of the refrigeration loop. For example, if the RTD sensor is calibrated to system COP (coefficient of performance), then a controller can be designed to regulate the refrigeration loop at its maximum COP.

1.4 Precautions with this Analysis: Motivation for Better Analysis

The data presented along with the discussion presented in this paper demonstrate the ability of this sensor to predict the LMF of refrigerant for that set of unique operating conditions. The data presented in this document is specific to the facilities and conditions described in this paper. Not enough data has been taken to generalize the sensor's behavior. The sensor's behavior can significantly change if it is used in another system. Parameters such as sensor orientation, distance to the evaporator, refrigerant type, pipe cross-section, and evaporator type, all of which were held relatively constant for each set of experiments, would create inconsistent results. All of these parameters mentioned would change the heat transfer dynamics between the refrigerant and the RTD. Recall that hA is intimately related to the heat transfer dynamics between the refrigerant and the RTD.

Assuming the output of the sensor can be calibrated to estimate LMF of the liquid exiting the evaporator, there is no guarantee that the calibration will stay. Many factors will change the calibration including:

- **sensor orientation** It is important that the effective surface area of the sensor remains constant. That means once the sensor is calibrated, the sensor cannot be rotated. It is suggested that the sensing filament of the sensor be oriented to maintain the maximum effective surface area with respect to the refrigerant flow.
- **distance from the evaporator** The nearer the sensor is to the exit of the evaporator, the more turbulent the flow will be. It is recommended that the RTD be placed as close to the exit of the evaporator as possible in order to obtain a representative sample of the cross-sectional flow.
- **refrigerant type** Refrigerant characteristics such as heat capacity, density, and heat of vaporization all affect the heat transfer characteristics. Recall that hA is highly dependent on the convective heat transfer coefficient between the RTD and the refrigerant.
- **pipe cross-section** It is important to maintain consistent sensor mass flux. Mass flux is the amount of mass that passes through a given cross-sectional area (2-D plane). If the pipe diameter is increased, then for the same mass flow rate, the RTD will have a lower mass flux.
- **type of evaporator** Different evaporators create different flow dynamics at their exits.

The refrigerant must be well mixed as it passes over the sensor in order for the results to be valid. The liquid droplets need to be evenly distributed throughout the cross-section of the pipe. This ensures that the sensor is getting a representative sample. In reality a small

portion of the refrigerant is being carried as an oil-refrigerant mixture that flows attached to the tube walls. The sensor cannot detect that portion of the flow. Our experience demonstrates that the sensor should be placed very near the evaporator exit, preferably in a vertical section of tube. Recall the lesson learned in section 1.3.3.5. Beware of situations where the RTD is not getting a representative sample of the cross-sectional flow.

The sensor can only be used in its region of sensitivity. The sensor must be used in the low LMF/high quality regimes. It cannot operate where traditional TXV systems operate.

Shannon et al [23,24] had hypothesized that a non-linear saturation response can occur with liquid refrigerant present in the stream. If liquid refrigerant completely coats the RTD and the influx of liquid is higher than the evaporation rate, then a thin film of boiling liquid can cover the surface. When saturation occurs, the sensor cannot provide further information to estimate LMF.

The numerous problems associated with this type of analysis are obvious. The system uses a limited amount of information to linearly combine relevant parameters in attempt to estimate the LMF of the refrigerant. So, the analysis assumes a linear behavior of a very non-linear system. There is an obvious need for a tool that reduces variables in a non-linear fashion. Neural networks seem to be the perfect tool for the job. The only major drawback in the practicality of using neural networks is that it must be fast if it is to be implemented in a real-time control system. The sample rate of this particular control system is once a second. This is particularly slow for a typical system. But, as microprocessors become faster, cheaper, and smaller, it may become practical to use neural networks in this type of control system.

Part 2: Using Neural Networks to Predict LMF

The previous section presented a method for estimating the liquid-mass fraction (LMF) of the refrigerant exiting the evaporator. The assumptions used in this analysis violated some fundamental engineering doctrines. While that crude method of data reduction was able to stably control the expansion valve, the system was not nearly generic enough to be applied to an arbitrary air conditioning or refrigeration system.

This section will develop and implement a neural network capable of more accurately predicting the LMF of refrigerant. The output of the neural network will be the actual LMF (as opposed to a control signal that could be inferred as a function of the LMF)

2.1 Relevant Variables

The previous model reduced three parameters (temperature of the sensor, temperature of the refrigerant, and the voltage across the sensor) to estimate the LMF. Three parameters were used because the reduction of data had to be as simple as possible. But, the use of neural networks allows an arbitrary number (within reason) of parameters, each of which have arbitrary relations to each other and to the output. Initially, every recorded variable that influenced the LMF of the refrigerant was used in the analysis. These included:

T_{sensor} [°C]. (The temperature of the sensor) This was inferred from the resistance of the sensor. The resistance of the sensor is known, because it is equal to R_{set} (see section 1.2.3). The temperature of an RTD is a linear function of its resistance.

V_{DC}. (The DC voltage across the sensor) This is measured by a multimeter on the HP data acquisition system.

V_{RMS}. (The root-mean square of the voltage of the voltage across the sensor) The RMS voltage was measured with specific hardware on the data acquisition system, but could have been estimated using equation 1.2 (refer to section 1.3.3.1).

T_{ref_out} [°C]. (The temperature of the refrigerant coming out of the evaporator) This is the refrigerant immediately before it passes over the sensor. This may also be referred to as the “free stream” temperature.

m_{dot} [grams/sec]. (the mass flow rate of the refrigerant) This is measured by a Micromotion™ Coriolis flow meter.

Pevap [kPa]. (the absolute pressure in the evaporator) This pressured is assumed to be same as the pressure of the refrigerant flowing over the sensor.

Tref_in [°C]. (the temperature of the refrigerant going into the evaporator)

Tmix [°C]. (The temperature of the refrigerant after it has passed over the sensor and has been well mixed) The refrigerant passing over the sensor is not in thermal equilibrium. As a result the reading from the temperature sensor (thermocouple) may not be accurate, because it is not known what it being read (the temperature of the possibly superheated vapor, or the temperature of the saturated liquid?). Tmix is in thermal equilibrium, and is a more stable measurement.

dPevap [dkPa]. (differential pressure across the evaporator) The pressure drop across the evaporator is an indication of the velocity of the refrigerant.

Pmix [kPa]. (pressure of the refrigerant after it has passed over the sensor and come to thermal equilibrium)

The output of the neural network is:

LMF. (liquid-mass-fraction of the refrigerant exiting the evaporator) Liquid-mass-fraction (LMF), which is the mass of liquid in vapor of any state, is a measure to describe the state at the evaporator exit, as described in Shannon, Hrnjak, and Leicht [28]. LMF is defined as

$$LMF = \frac{m_{liquid}}{m_{liquid} + m_{vapor}} \quad (\text{eq 1.1})$$

The complete list of the data is in Table 2.1.

Table 2.1 All the data used to construct the neural network

<u>Tsensor</u>	<u>DC</u>	<u>RMS</u>	<u>Tref,out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref,in</u>	<u>Tmix</u>	<u>DPEvap</u>	<u>Pmix</u>	<u>LMF</u>
10.00	10.62	0.439	0.87	0.0256	506.9	0.30	0.65	17.45	502.1	0.077
15.00	12.89	0.640	0.78	0.0255	505.3	0.20	0.56	17.43	500.4	0.071
20.00	13.83	0.961	0.79	0.0256	505.2	0.20	0.56	17.44	500.4	0.073
25.00	15.57	1.512	0.71	0.0255	504.1	0.12	0.50	17.40	499.3	0.069
30.00	17.53	2.208	0.70	0.0256	503.7	0.11	0.48	17.45	499.0	0.074
30.00	16.79	1.295	1.91	0.0255	516.8	0.88	2.48	17.79	512.2	0.011
25.00	15.03	0.518	2.33	0.0255	518.1	0.97	3.23	17.83	513.4	0.010
20.00	13.00	0.345	2.45	0.0255	518.8	1.01	3.92	17.78	514.0	0.007
15.00	11.16	0.335	2.30	0.0255	518.5	0.98	4.01	17.86	513.6	0.007
10.00	6.28	0.228	3.62	0.0254	515.5	0.81	5.07	17.89	510.6	0.006
10.00	9.13	0.413	1.20	0.0255	510.0	0.48	1.01	17.94	505.2	0.031
15.00	11.30	0.532	1.27	0.0256	511.3	0.56	1.06	17.88	506.6	0.036
20.00	12.77	0.601	1.25	0.0256	511.4	0.56	1.08	17.86	506.6	0.033
25.00	14.11	0.591	1.29	0.0255	510.5	0.52	1.05	17.78	505.7	0.029
30.00	15.94	1.531	1.28	0.0255	511.1	0.55	1.07	17.79	506.3	0.030
30.00	18.34	2.405	-0.04	0.0257	492.8	-0.54	-0.26	17.10	488.2	0.112
25.00	16.86	2.167	-0.10	0.0257	491.8	-0.60	-0.32	17.09	487.2	0.113
20.00	15.77	1.151	-0.08	0.0257	492.0	-0.58	-0.30	17.10	487.4	0.115
15.00	13.80	0.609	-0.09	0.0257	491.8	-0.59	-0.31	17.20	487.2	0.112
10.00	11.24	0.440	-0.09	0.0257	491.8	-0.60	-0.32	16.99	487.2	0.115
7.50	9.91	0.360	-0.11	0.0257	491.5	-0.61	-0.34	16.94	487.0	0.114
7.69	18.83	1.815	-0.55	0.0483	490.9	-0.34	-1.06	17.03	483.6	0.221
5.13	14.46	0.175	-0.32	0.0484	493.9	-0.11	-0.83	17.15	486.8	0.220
7.69	10.57	0.159	1.80	0.0514	527.8	1.56	1.29	17.79	519.2	0.139
7.69	15.28	0.171	1.72	0.0515	526.1	1.49	1.19	17.69	517.6	0.139
5.90	12.01	0.135	0.82	0.0493	509.7	0.96	0.37	17.28	502.1	0.124
10.26	17.52	1.147	1.01	0.0494	512.9	1.13	0.57	17.41	505.3	0.118
7.69	13.28	0.172	2.52	0.0513	539.5	2.23	2.03	18.15	530.8	0.114
5.13	8.93	0.094	2.58	0.0514	540.7	2.29	2.09	18.16	532.0	0.113
9.74	16.02	0.154	2.63	0.0514	541.3	2.33	2.15	18.12	532.7	0.112
5.13	11.98	0.155	0.76	0.0494	508.8	0.86	0.28	18.10	500.7	0.078
10.26	16.97	0.133	1.72	0.0489	524.3	1.77	1.32	17.61	516.6	0.077
6.67	12.06	0.164	1.38	0.0478	518.1	1.39	0.95	17.69	510.5	0.070
7.69	14.53	0.137	0.97	0.0495	511.9	1.03	0.52	18.18	503.9	0.062
9.74	16.52	0.127	1.31	0.0496	517.6	1.37	0.87	18.25	509.5	0.059
11.79	16.89	0.143	2.40	0.0490	534.9	2.38	2.04	17.74	527.2	0.042
8.46	12.08	0.142	2.56	0.0488	537.8	2.51	2.20	18.04	530.1	0.027
12.82	16.70	0.117	2.64	0.0488	538.7	2.55	2.28	18.06	531.0	0.026
9.23	12.10	0.108	2.78	0.0488	540.7	2.67	2.46	18.24	533.1	0.010
5.13	12.13		0.29	0.0494	501.1	0.43	-0.16	14.14	493.2	0.100
5.13	11.46		0.70	0.0491	507.5	0.79	0.24	13.95	499.4	0.070
5.13	11.45		0.62	0.0490	506.8	0.69	0.17	13.84	498.4	0.060
9.74	17.30		0.62	0.0490	506.8	0.69	0.17	13.84	498.4	0.060
5.13	9.75		1.36	0.0488	518.7	1.34	0.93	13.67	510.4	0.030
9.74	15.31		1.36	0.0488	518.7	1.34	0.93	13.67	510.4	0.030
9.74	15.58		1.48	0.0491	519.8	1.50	1.06	13.71	511.8	0.030
15.13	20.34		1.36	0.0488	518.7	1.34	0.93	13.67	510.4	0.030
9.74	13.81		2.13	0.0489	529.9	2.03	1.77	13.45	522.1	0.010
9.74	13.61		2.21	0.0488	525.0	1.69	4.60	13.73	516.7	0.001
15.13	17.41		2.21	0.0488	525.0	1.69	4.60	13.73	516.7	0.001

2.2 Missing Data

Eleven cases have missing RMS data (40 through 50). *ST Neural Networks* has the capability to deal with missing data values, both during network training and execution. Missing values are handled by substituting a special value (or, in the case of one-of-N encoded nominal variables, a set of values). The special values used to patch missing values are determined from the training set when the network is trained.

The available methods are:

Mean. For numeric variables, the mean of the variable's values in the training set is used. For nominal variables, the proportion of training cases in each class is used. This is usually the most appropriate method.

Median. For numeric variables, the median of the values in the training set is used. For nominal variables, this is treated identically to mean substitution. May be superior to mean substitution for a numeric variable which contains strong outliers.

Minimum. For numeric variables, the minimum of the values in the training set. For nominal variables, the class assigned is that with the lowest frequency in the training set.

Maximum. For numeric variables, the maximum of the values in the training set. For nominal variables, the class assigned is that with the highest frequency in the training set.

Zero. A zero value is substituted.

For this particular set of data it would be most beneficial to use the mean.

2.3 Choosing the Training, Verification, and Testing Data Sets

Iterative algorithms need both a starting point and a stopping rule. The starting point is usually taken to be a random set of weights. Some care is needed that they are not taken to be too large, because the hidden units may start in a saturated state. The earliest idea was to stop when the error became small. This is often fine in logical problems, where no example is ever mis-classified, but can result in poor generalisation. Very many ad hoc stopping rules have been proposed. One that seems popular is to have a validation set, and stop training when the error measure on the validation set starts to rise. This is dangerous, as we have often encountered examples in which after an initial drop the error on the validation set rises slowly for a large number of iterations, then falls dramatically to a small fraction of its previous minimum. Thus, one can never know if the minimum error on the validation set has yet been attained. It is also not uncommon to use the test set rather than a validation set. In

attempt to obtain the best neural network possible training, verification, and test sets will be used.

The data set is divided into three sections: the training set, verification set, and test set. The training set is the data shown to the network to adjust the weights. The verification set is used to check the progress of training. The test set is never shown to the network while training. During training and testing, the verification and test set are treated differently to the training set. Any statistics reported are calculated separately for these three parts of the data set, and training algorithms do not use the verification or test sets to adjust network weights. The verification set may optionally be used to track the network's error performance, to identify the best network, and to stop training if over-learning occurs. The test set is not used in training at all, and is designed to give an independent assessment of the network's performance when an entire network design procedure is completed.

Ideally, you would like to show the network as many diverse cases as possible. This would allow the network to become more accurate over a wide range of possible inputs. But, if all the data is used for the training set, then there is no way of checking to see if the network is performing well. A phenomenon call “over training” can occur if the network is trained too much. Over training the network forces the network to adapt to only the data given to it. So, if new data is inputted to the network, the network will not be able to respond with the proper output. Without any verification data or test data there is no way to know if the network has been over trained.

Dividing the data set up into the four sections is not an exact science. In order to experiment with the number of training, verification, and test cases, an arbitrary MLP network was created with one hidden layer containing five neurons (the best number of neurons will be explored later) Initially, 25 cases were chosen for training, 12 cases for verification, and the remaining 13 were used for testing. While keeping the network architecture constant and the number of epochs constant (2,000), the same thing was done for three different permutations of the number of cases given to each section. The results are summarized in the following table.

cases			RMS error		
<u>training</u>	<u>verification</u>	<u>test</u>	<u>training</u>	<u>verification</u>	<u>test</u>
25	12	13	0.0101	0.0476	0.0355
38	7	5	0.0189	0.0071	0.0047
20	17	13	0.0043	0.0254	0.0456
27	9	14	0.0088	0.0215	0.0487

For each trial the cases were distributed differently throughout training, verification, and test. The right half of the table displays the RMS error of the network across the training, verification and test sets. This is the Root Mean Square of the individual case errors, using the network's error function for each case, adjusted for the network's post-processing scaling factor. This is normally equal to the RMS error of the residuals. The exception is the case of multiple output regression problems, where the individual residuals are effectively weighted to give equal importance to each output variable irrespective of its original scale. These error values can be used to directly compare the performance of various networks, even of different types, providing that they have the same network error function.

The ambiguity of the results begs the fundamental question: which is the best network? How can a network be assessed given that data? The third trial had the lowest training RMS error, but it also had the least amount of training cases, so it was relatively easier for the network to accommodate those 20 training cases. But the 13 independent cases (test cases) of that same network did relatively poorly in the network. Too few training cases is most probably a bad thing.

The second trial uses the other extreme of having almost all of its cases committed to training the network. This network seems to perform very well. The independent testing cases performed very well in the network. This shows that the network can accurately predict results for cases that it has never seen before (This is was the network is designed to do.). But the numbers may be deceiving. There were only five test cases that the network used. What if these five cases happened to be five cases that the network did well with? The fact that the RMS error for the test and verification sets is lower than for the training set is some indication that something may be misleading. So, to verify the data three more trials were conducted using the same scenario. While the same number of training, verification, and test sets were used, the cases that were used for each type were changed for each case. For example if case

26 was used for verification in trial 1, then it would be used for training or testing in the next trial. The results are summarized in the following table.

RMS error		
<u>training</u>	<u>verification</u>	<u>test</u>
0.0085	0.074	0.073
0.0193	0.0279	0.012
0.0213	0.0088	0.0272

Note how much variance is present between trials for the verification and test RMS error. This is due to the fact that there are only five cases for each.

So, probably the ‘best’ combination of cases would either be (25,12,13) – trial 1 or (27,9,14) – trial 4.

2.4 Statistical Information on the Various Case Types

Various implementations of neural networks were explored in attempt to find the “best” network. Best is in quotations because it is a subjective assessment of the performance of the network. There are numerous deterministic quantifiers used to estimate the performance of a network. So, optimisation of a network is often difficult and is dependent on the specific needs of the application. Nonetheless, there are a few basic deterministic parameters that characterise the network and can be used to estimate the performance of the network. For this exercise *Statistica’s* output values will be used to estimate the performance of each case tried.

In regression problems, the purpose of the neural network is to learn a mapping from the input variables to a continuous output variable, or variables. A network is successful at regression if it makes predictions more accurate than a simple estimate.

The simplest way to construct an estimate, given training data, is to calculate the mean of the training data, and use that mean as the predicted value for all previously unseen cases. The average expected error from this procedure is the standard deviation of the training data. The aim in using a regression network is therefore to produce an estimate that has a lower prediction error standard deviation than the training data standard deviation.

ST Neural Networks automatically calculates the mean and standard deviation of the training and verification subsets, when the entire data set is run. It also calculates the mean

and standard deviations of the prediction errors. The ratio of the prediction to data standard deviations is displayed; if this is 1.0, then the network does no better than a simple average. A lower ratio indicates a better estimate. In addition, *ST Neural Networks* displays the standard Pearson-R correlation coefficient between the actual and predicted outputs. A perfect prediction will have a correlation coefficient of 1.0. A correlation of 1.0 does not necessarily indicate a perfect prediction (only a prediction which is perfectly linearly correlated with the actual outputs), although in practice the correlation coefficient is a good indicator of performance. It also provides a simple and familiar way to compare the performance of your neural networks with standard least squares linear fitting procedures.

Case 1. The first case tested for this exercise has ten neurons in the input layer, one hidden layer with five neurons, and a single neuron in the output layer. Back propagation was used with 27 cases for training, 9 for verification, and 14 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0741	0.0804	0.0496
Data S.D.	0.0439	0.0633	0.0576
Error Mean	-0.0021	0.0025	0.0430
Error S.D.	0.0028	0.0149	0.0527
Abs E. Mean	0.0029	0.0112	0.0578
S.D. Ratio	0.0647	0.2350	0.9143
Correlation	0.9979	0.9748	0.5047

The regression statistics datasheet displays the statistics for the data set.

Data Mean. Average value of the target output variable.

Data S.D. Standard deviation of the target output variable.

Error Mean. Average error (residual between target and actual output values) of the output variable.

Abs. E. Mean. Average absolute error (difference between target and actual output values) of the output variable.

Error S.D. Standard deviation of errors for the output variable.

S.D. Ratio. The error:data standard deviation ratio.

Correlation. The standard Pearson-R correlation coefficient between the target and actual output values.

Case 2. The second case tested for this exercise has ten neurons in the input layer, one hidden layer with seven neurons, and a single neuron in the output layer. Back propagation was used with 30 cases for training, 10 for verification, and 10 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0700	0.0586	0.0734
Data S.D.	0.0528	0.0456	0.0592
Error Mean	0.0001	0.0009	0.0023
Error S.D.	0.0187	0.0090	0.0278
Abs E. Mean	0.0140	0.0065	0.0190
S.D. Ratio	0.3539	0.1979	0.4693
Correlation	0.9358	0.9803	0.8979

Case 3. The second case tested for this exercise has ten neurons in the input layer, one hidden layer with three neurons, and a single neuron in the output layer. Back propagation was used with 26 cases for training, 14 for verification, and 10 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0753	0.0519	0.0734
Data S.D.	0.0526	0.0450	0.0592
Error Mean	-0.0064	0.0077	-0.0059
Error S.D.	0.0357	0.0305	0.0393
Abs E. Mean	0.0310	0.0253	0.0273
S.D. Ratio	0.6777	0.6785	0.6641
Correlation	0.7641	0.7356	0.7585

Case 4. The second case tested for this exercise has ten neurons in the input layer, two hidden layers each with four neurons, and a single neuron in the output layer. Back propagation was used with 27 cases for training, 13 for verification, and 10 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0728	0.0554	0.0734
Data S.D.	0.0532	0.0448	0.0592
Error Mean	-0.0315	0.0021	-0.0215
Error S.D.	0.0508	0.0500	0.0616
Abs E. Mean	0.0459	0.0432	0.0414
S.D. Ratio	0.9544	1.1154	1.0396
Correlation	0.4023	-0.2342	-0.0714

Here is a summary of the average absolute error and the standard Pearson-R correlation for each of the cases:

		<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Case 1	Abs E. Mean	0.0029	0.0112	0.0578
	Correlation	0.9979	0.9748	0.5047
Case 2	Abs E. Mean	0.0140	0.0065	0.0190
	Correlation	0.9358	0.9803	0.8979
Case 3	Abs E. Mean	0.0310	0.0253	0.0273
	Correlation	0.7641	0.7356	0.7585
Case 4	Abs E. Mean	0.0459	0.0432	0.0414
	Correlation	0.4023	-0.2342	-0.0714

Case 1 seems to exhibit good statistics for the training and verification sets. But the statistics are significantly worse for the testing set. This is in general a bad sign for a network. The network seemed to learn the system well for cases that it has seen (used for verification). But the network has trouble generalizing its “knowledge” to the test cases that it hasn’t seen before. Case 2 performed much better for the training data. The correlation for the test set is almost 0.9, which is pretty impressive. Case 4 was an interesting experiment because it used two hidden layers in its network. For whatever reason case 4 performed poorly compared to the other cases explored in this section. Determining the best number of hidden layers is not a straightforward problem (This topic will be explored in greater detail later in this paper.). Using only the statistics explored in this section, case 2 seems to have performed the best.

2.5 Using “Bound” Cases

Bound cases are often used in training and verification of neural networks when examples of extreme data are naturally present. Creating bound cases is also a way to artificially increase your data and to increase your input domain. Bound cases may be used to check to see if an established network is able to handle extreme inputs, or they can be used to train the network to handle some artificially imposed extreme inputs. Whatever these cases are used for, the creator of these cases must have a fairly extensive intuition of the behaviour of system that is being modelled by the network.

Due to the extreme complexity of the interaction of the variables in this particular set of data, it would be difficult to develop bound cases that would accurately describe the system. No one variable could take an extreme value that would guarantee a particular value of the output. One possibility is that if the combination of the temperature of the refrigerant coming out of the evaporator and the pressure in the evaporator dictate that the refrigerant should have a very high superheat (Superheat is defined as the difference between the refrigerant’s temperature and its saturation temperature for the given pressure.), then the LMF should be zero. But, this may not always be the case. The problem is that the liquid/vapor mixture that is exiting the evaporator is not in thermal equilibrium. All thermodynamic properties of fluids assume that the fluid is in equilibrium. In reality, there is saturated liquid droplets entrained in superheated vapor. It is not clear what the instrumentation is reading (the temperature of the vapor or of the liquid?).

If the T_{mix} and P_{mix} indicate that the refrigerant is subcooled, then it must be the case that the LMF is unity. But, this situation should never occur. I have never seen a case where the LMF of the refrigerant exiting the evaporator was more than about 0.2. In my opinion it is physically impossible for the LMF ever to be greater than 0.8 in any situation. Also, the purpose of this neural network is to create a predictor of LMF that will be used as a feedback signal for a controller. This controller will be used to regulate the LMF at about 0.02 or so. So, I don’t think a case that has the LMF at unity would do the network any good.

2.6 Correlation Matrix

A correlation matrix is a useful statistical tool. The matrix will give insight to the linear dependence that the variables have on each other. A correlation of 1 (or -1) between two variables means that those variables have perfect linear correlation. A correlation of zero

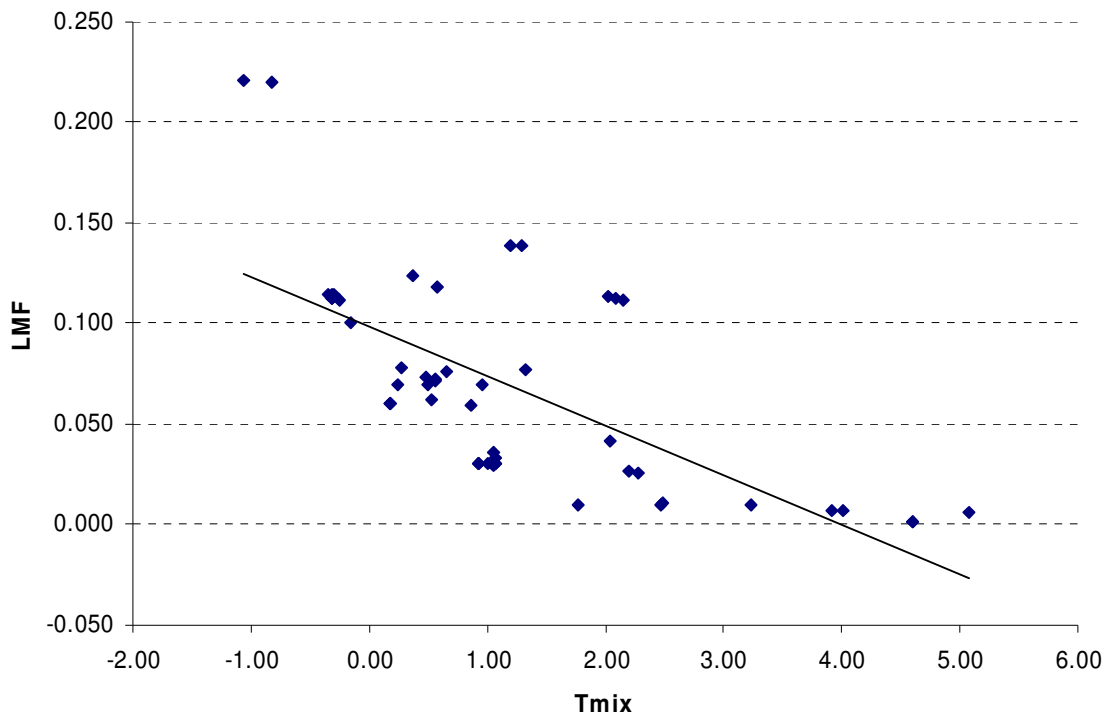
means that the variables have no correlation at all. One motivation for the construction of a correlation matrix is to reduce the dimensionality of the system. For every set of two variables that have a strong linear dependence the dimensionality of the network system can be reduced by one. This will be discussed in the next section.

The correlation matrix for the data is as follows:

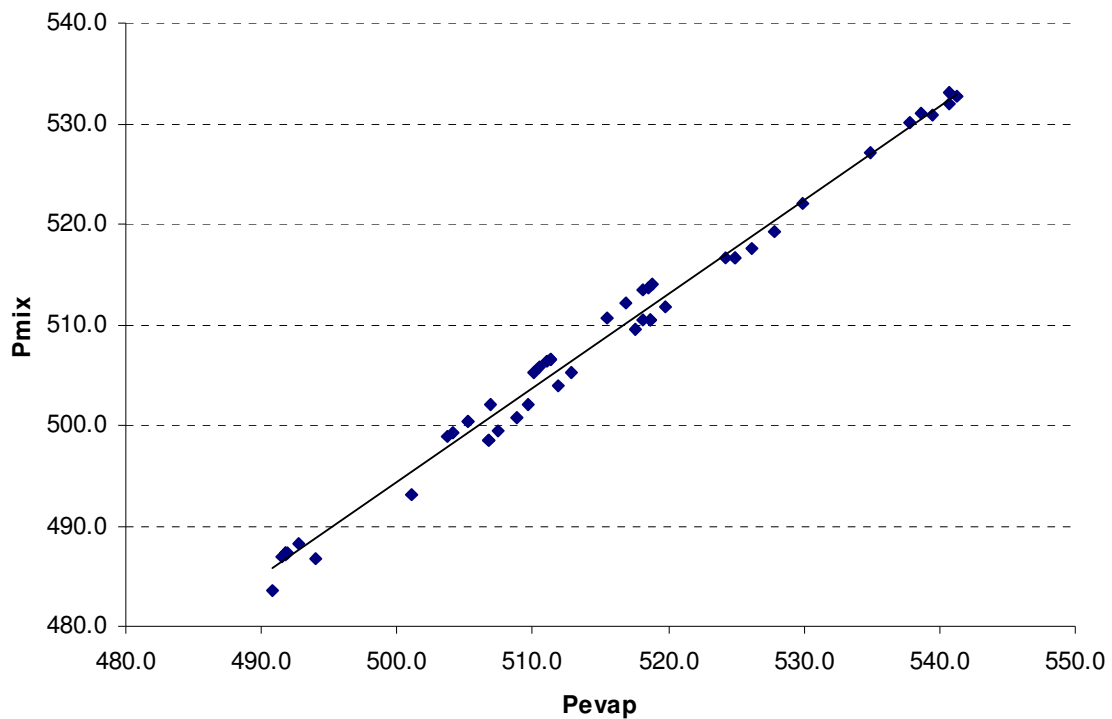
	Tsensor	VDC	VRMS	Tref_out	mdot	Pevap	Tref_in	Tmix	dPevap	Pmix	LMF
Tsensor	1	<u>0.436</u>	<u>0.689</u>	-0.077	<u>-0.716</u>	-0.253	<u>-0.364</u>	0.069	0.237	-0.183	-0.259
VDC	<u>0.436</u>	1	<u>0.485</u>	-0.199	0.139	-0.062	-0.015	-0.181	-0.126	-0.080	0.115
VRMS	<u>0.689</u>	<u>0.485</u>	1	<u>-0.492</u>	<u>-0.438</u>	<u>-0.545</u>	<u>-0.573</u>	<u>-0.336</u>	-0.111	-0.525	0.155
Tref_out	-0.077	-0.199	<u>-0.492</u>	1	0.233	<u>0.897</u>	<u>0.836</u>	<u>0.887</u>	0.145	<u>0.920</u>	<u>-0.615</u>
mdot	<u>-0.716</u>	0.139	<u>-0.438</u>	0.233	1	<u>0.528</u>	<u>0.666</u>	0.023	<u>-0.349</u>	<u>0.440</u>	0.169
Pevap	-0.253	-0.062	-0.545	<u>0.897</u>	<u>0.528</u>	1	<u>0.976</u>	<u>0.663</u>	0.077	<u>0.995</u>	<u>-0.427</u>
Tref_in	<u>-0.364</u>	-0.015	<u>-0.573</u>	<u>0.836</u>	<u>0.666</u>	<u>0.976</u>	1	<u>0.596</u>	-0.028	<u>0.954</u>	<u>-0.406</u>
Tmix	0.069	-0.181	<u>-0.336</u>	<u>0.887</u>	0.023	<u>0.663</u>	<u>0.596</u>	1	0.040	<u>0.697</u>	<u>-0.673</u>
dPevap	0.237	-0.126	-0.111	0.145	<u>-0.349</u>	0.077	-0.028	0.040	1	0.132	0.207
Pmix	-0.183	-0.080	<u>-0.525</u>	<u>0.920</u>	<u>0.440</u>	<u>0.995</u>	<u>0.954</u>	<u>0.697</u>	0.132	1	<u>-0.467</u>
LMF	-0.259	0.115	0.155	<u>-0.615</u>	0.169	<u>-0.427</u>	<u>-0.406</u>	<u>-0.673</u>	0.207	<u>-0.467</u>	1

Marked correlations are significant at $p < .05000$

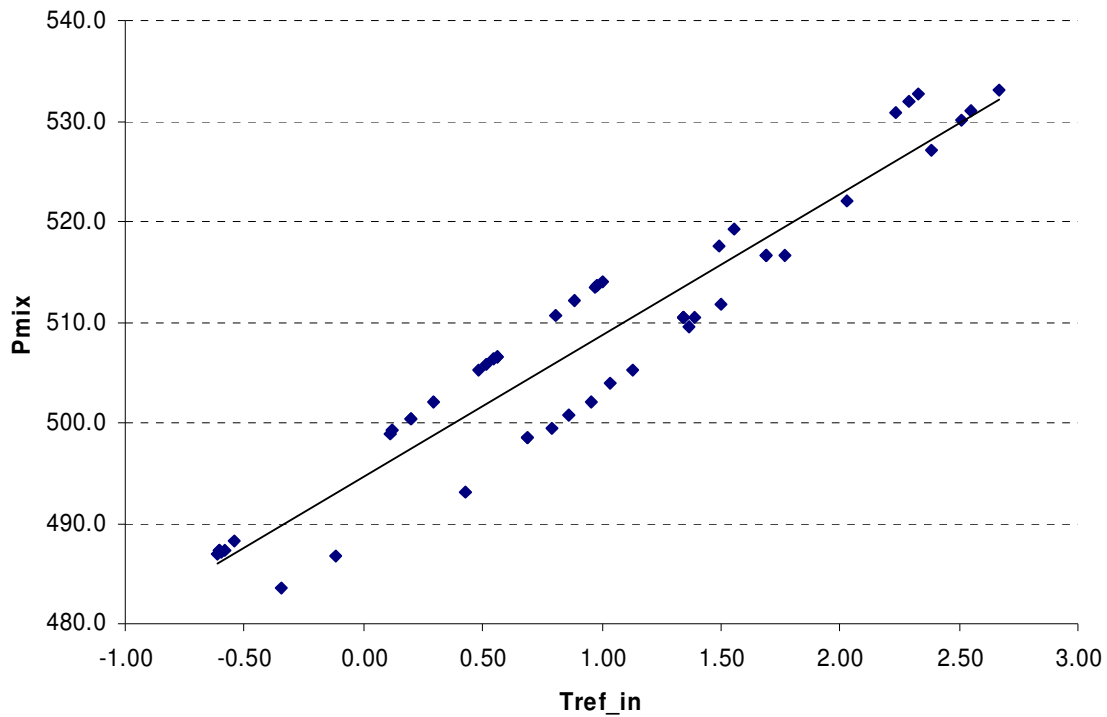
There are quite a few significant pairs. The single best variable that linearly correlates with LMF is Tmix. As seen here:



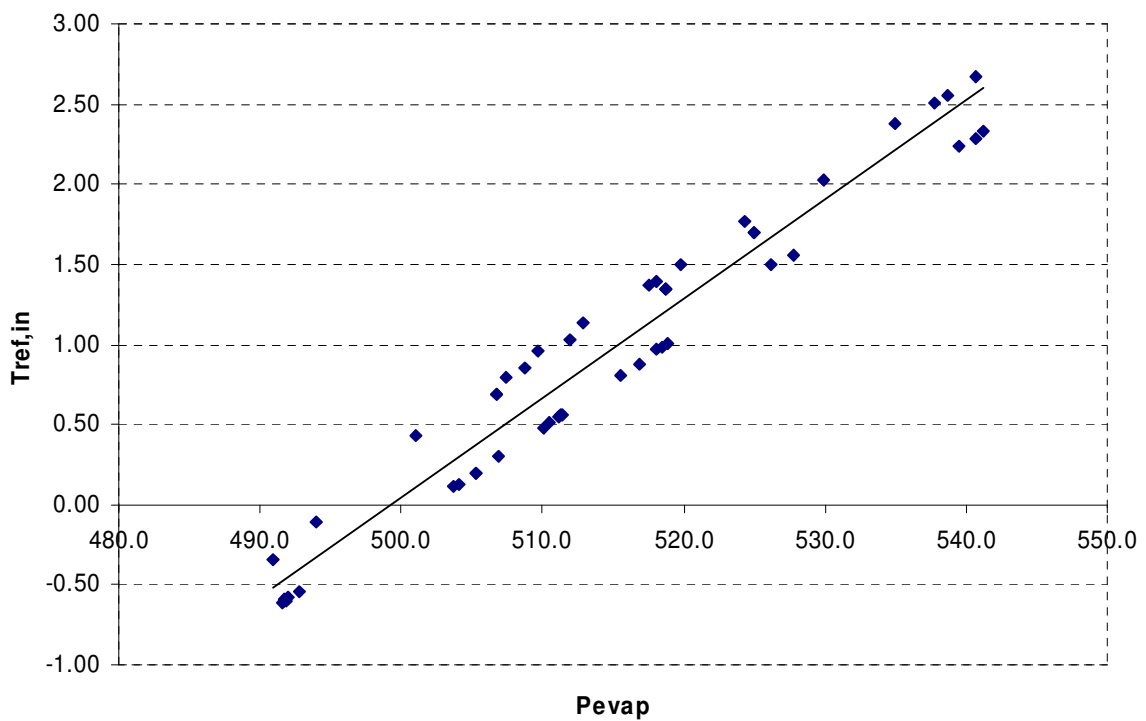
A few pairs of variables have an extremely high linear correlation (> 0.9). Coefficients of this magnitude make it easy for dimensionality reduction. The highest correlation was between P_{mix} and P_{evap} (0.995). Not only is the correlation high, but their numerical values are almost identical. P_{evap} is measured before the refrigerant reaches the sensor (in the evaporator) and P_{mix} is measured after the sensor. The only pressure drop between them is due to frictional losses from the wall of the piping. Due to the correlation, at least one of them will have to be removed during the dimensionality reduction analysis.



Another very high correlation was between P_{mix} and T_{ref_in} . T_{ref_in} is the temperature of the refrigerant going into the evaporator. The refrigerant going into the evaporator is saturated liquid/vapor, and therefore, fixed at its saturation temperature and pressure. The mixed refrigerant downstream of the sensor is usually saturated liquid/vapor as well. Measurements taken at these points are steady and fairly accurate. Because both are saturated and only separated by a constant pressure drop, they should be intimately related.



There is a strong linear correlation between Tref,in and Pevap. It was shown that Tref,in is strongly correlated with Pmix. And Pmix was shown to be strongly correlated with Pevap. So, it is logical that Tref,in is strongly correlated with Pevap.



2.7 Reduction of Inputs

A sensitivity analysis on the inputs to a neural network indicates which input variables are considered most important by that particular neural network. Sensitivity analysis can give important insights into the usefulness of individual variables. It often identifies variables that can be safely ignored in subsequent analysis, and key variables that must always be retained. However, it must be deployed with some care, for reasons that are explained below.

Input variables are not, in general, independent – that is, there are interdependencies between variables. Sensitivity analysis rates variables according to the deterioration in modeling performance that occurs if that variable is no longer available to the model. In so doing, it assigns a single rating value to each variable. However, the interdependence between variables means that no scheme of single ratings per variable can ever reflect the subtlety of the true situation. There may be interdependent variables that are useful only if included as a set. If the entire set is included in a model, they may be accorded significant sensitivity, but this does not reveal the interdependency. Worse, if only part of the interdependent set is included, the sensitivity of the others will be zero, as they carry no important information.

Sensitivity analysis does not rate the "usefulness" of variables in modelling in a reliable or absolute manner. You must be cautious in the conclusions you draw about the importance of variables. Nonetheless, in practice it is extremely useful. If a number of models are studied, it is often possible to identify key variables that are always of high sensitivity, others that are always of low sensitivity, and "ambiguous" variables that change ratings and probably carry mutually redundant information.

A datasheet was constructed to estimate the importance of each variable. It indicates the sensitivity of each variable. Sensitivity is reported separately for training and verification subsets – the consistency of the sensitivity ratings across the two subsets is a good initial cross check on the reliability of the sensitivity analysis. The sensitivity is reported in three rows – the Rank, Error, and Ratio. The basic sensitivity figure is the Error. This indicates the performance of the network if that variable is "unavailable." Important variables have a high error, indicating that the network performance deteriorates badly if they are not present. The Ratio reports the ratio between the Error and the Baseline Error (i.e. the error of the network if all variables are "available"). If the Ratio is one or lower, then making the variable "unavailable" either has no effect on the performance of the network, or actually enhances it!

The Rank simply lists the variables in order of importance (i.e. order of descending Error), and is provided for convenience in interpreting the sensitivities.

One of the better networks that I have explored had one hidden layer with five neurons. 27 cases were used for training, 9 for verification, and 14 for testing. A datasheet was constructed for this network:

		<u>Tsensor</u>	<u>VDC</u>	<u>VRMS</u>	<u>Tref_out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref_in</u>	<u>Tmix</u>	<u>dPevap</u>	<u>Pmix</u>
training	Rank	7	10	8	6	4	2	1	9	3	5
	Error	0.0228	0.0164	0.0227	0.0293	0.0337	0.0511	0.0706	0.0202	0.0433	0.0305
	Ratio	1.449	1.045	1.446	1.863	2.147	3.254	4.496	1.284	2.755	1.943
verification	Rank	7	10	9	6	4	2	1	8	3	5
	Error	0.0321	0.0192	0.0219	0.0387	0.0427	0.0641	0.0721	0.0291	0.0448	0.0390
	Ratio	1.549	0.924	1.056	1.865	2.061	3.093	3.477	1.405	2.161	1.883

The rankings of the training sets and the verification sets are pretty consistent. The training and verification sensitivity analysis ranks Tref_in as the most important variable. The verification and training sensitivity analysis ranks VDC as the least important variable (so bad, in fact, that, according to the verification data, the network would perform better without it). But these numbers are dependent on the construction of the network. A few other networks will be analyzed to investigate the consistency of the results.

The next network that will be looked at has seven hidden neurons in a single layer. 30 cases are used for training, 10 for verification, and 10 for testing.

		<u>Tsensor</u>	<u>VDC</u>	<u>VRMS</u>	<u>Tref_out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref_in</u>	<u>Tmix</u>	<u>dPevap</u>	<u>Pmix</u>
training	Rank	3	9	6	7	5	8	1	10	4	2
	Error	0.0334	0.0190	0.0250	0.0230	0.0278	0.0198	0.0753	0.0159	0.0286	0.0502
	Ratio	2.208	1.251	1.652	1.519	1.833	1.305	4.969	1.049	1.885	3.312
verification	Rank	2	7	10	6	4	9	1	8	5	3
	Error	0.0416	0.0222	0.0189	0.0294	0.0330	0.0190	0.0432	0.0222	0.0326	0.0352
	Ratio	1.933	1.032	0.880	1.369	1.536	0.883	2.010	1.031	1.514	1.637

There seems to be a bit more variation between the training and verification sets.

The third network examined had three hidden neurons in one hidden layer. There were 30 training, 10 verification, and ten test cases.

		<u>Tsensor</u>	<u>VDC</u>	<u>VRMS</u>	<u>Tref_out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref_in</u>	<u>Tmix</u>	<u>dPevap</u>	<u>Pmix</u>
training	Rank	6	9	7	10	3	4	1	8	2	5
	Error	0.0175	0.0065	0.0141	0.0040	0.0627	0.0583	0.1083	0.0092	0.0642	0.0542
	Ratio	4.787	1.785	3.854	1.097	17.172	15.961	29.679	2.528	17.583	14.854
	Rank	6	8	9	10	2	5	1	7	3	4

verification	Error	0.0185	0.0112	0.0111	0.0108	0.0589	0.0440	0.0976	0.0135	0.0506	0.0443
	Ratio	1.815	1.094	1.087	1.056	5.762	4.308	9.552	1.321	4.957	4.340

This network showed preference to a slightly different set of variables. None of the variables in this particular network seems to degrade the performance of the network.

The last network that will be looked at in this exercise has four hidden layers, each with three neurons. There are 30 training, 10 verification, and 10 training cases.

		<u>Tsensor</u>	<u>VDC</u>	<u>VRMS</u>	<u>Tref_out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref_in</u>	<u>Tmix</u>	<u>dPevap</u>	<u>Pmix</u>
training	Rank	2	7	10	6	1	4	5	9	3	8
	Error	0.0512	0.0222	0.0056	0.0229	0.0793	0.0378	0.0334	0.0085	0.0457	0.0137
	Ratio	16.662	7.217	1.813	7.444	25.809	12.304	10.878	2.777	14.884	4.473
verification	Rank	2	5	8	4	1	10	7	6	3	9
	Error	0.0523	0.0270	0.0171	0.0307	0.0870	0.0150	0.0229	0.0240	0.0409	0.0167
	Ratio	2.965	1.529	0.970	1.743	4.939	0.851	1.300	1.361	2.320	0.946

This particular network didn't like Pevap, Pmix, or VRMS.

The average over the four trials presented in this experiment:

		<u>Tsensor</u>	<u>VDC</u>	<u>VRMS</u>	<u>Tref_out</u>	<u>mdot</u>	<u>Pevap</u>	<u>Tref_in</u>	<u>Tmix</u>	<u>dPevap</u>	<u>Pmix</u>
training	Rank	4.5	8.75	7.75	7.25	3.25	4.5	2	9	3	5
	Error	0.0312	0.0160	0.0168	0.0198	0.0509	0.0417	0.0719	0.0135	0.0454	0.0372
	Ratio	6.276	2.824	2.191	2.981	11.740	8.206	12.505	1.909	9.277	6.145
verification	Rank	4.25	7.5	9	6.5	2.75	6.5	2.5	7.25	3.5	5.25
	Error	0.0361	0.0199	0.0173	0.0274	0.0554	0.0355	0.0589	0.0222	0.0422	0.0338
	Ratio	2.066	1.145	0.998	1.508	3.574	2.284	4.085	1.280	2.738	2.202

The only variable that can be pruned with confidence is VDC. The next closest variable that the sensitivity analysis says should be pruned is VDC. But, VDC is actually a very important variable (questionably the most important). Without having knowledge of the system pruning VDC would appear to benefit the network. But, VDC will not be pruned because of its importance.

Recall from the previous section that many pairs of variables had very large correlations with each other. Some of these correlations are so large (>0.9) that some of these variables can be removed from the network without having the network lose much available information. Pmix had a very strong linear correlation with Pevap and Tref_in. Therefore, Pevap and Tref_in can be removed from the input of the network without having the network suffer. The remaining seven inputs to the network are:

1. Tsensor
2. VDC
3. Tref_out
4. mdot
5. Tmix
6. dPevap
7. Pmix

A few trials will be run to see how the network performs with and without the three pruned variables.

Case 1. The first case tested for this exercise has one hidden layer with five neurons and a single neuron in the output layer. Back propagation was used with 27 cases for training, 9 for verification, and 14 for testing. Learning rate was 0.1 and momentum was 0.3. The results of using all ten variables were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0741	0.0804	0.0496
Data S.D.	0.0439	0.0633	0.0576
Error Mean	-0.0021	0.0025	0.0430
Error S.D.	0.0028	0.0149	0.0527
Abs E. Mean	0.0029	0.0112	0.0578
S.D. Ratio	0.0647	0.2350	0.9143
Correlation	0.9979	0.9748	0.5047

And with only seven variables:

Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.011	-0.001	0.013
Error S.D.	0.020	0.025	0.032
Abs E. Mean	0.018	0.020	0.027
S.D. Ratio	0.407	0.385	0.698
Correlation	0.913	0.937	0.724

The network with only seven inputs performed better with the test cases.

Case 2. The second case tested for this exercise has one hidden layer with seven neurons and a single neuron in the output layer. Back propagation was used with 30 cases for training, 10 for verification, and 10 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0700	0.0586	0.0734
Data S.D.	0.0528	0.0456	0.0592
Error Mean	0.0001	0.0009	0.0023
Error S.D.	0.0187	0.0090	0.0278
Abs E. Mean	0.0140	0.0065	0.0190
S.D. Ratio	0.3539	0.1979	0.4693
Correlation	0.9358	0.9803	0.8979

And with seven inputs:

Data Mean	0.063	0.084	0.069
Data S.D.	0.051	0.061	0.048
Error Mean	0.012	0.003	0.013
Error S.D.	0.017	0.028	0.041
Abs E. Mean	0.017	0.023	0.034
S.D. Ratio	0.342	0.454	0.857
Correlation	0.940	0.911	0.516

For this particular network it did better when it had all ten of the inputs.

Case 3. The second case tested for this exercise has one hidden layer with three neurons and a single neuron in the output layer. Back propagation was used with 26 cases for training, 14 for verification, and 10 for testing. Learning rate was 0.1 and momentum was 0.3. The results of this case were as follows:

	<u>Tr. LMF</u>	<u>Ve. LMF</u>	<u>Te. LMF</u>
Data Mean	0.0753	0.0519	0.0734
Data S.D.	0.0526	0.0450	0.0592
Error Mean	-0.0064	0.0077	-0.0059
Error S.D.	0.0357	0.0305	0.0393
Abs E. Mean	0.0310	0.0253	0.0273
S.D. Ratio	0.6777	0.6785	0.6641
Correlation	0.7641	0.7356	0.7585

And with only seven inputs

Data Mean	0.071	0.069	0.061
Data S.D.	0.050	0.059	0.052
Error Mean	0.009	0.001	0.004
Error S.D.	0.021	0.028	0.037
Abs E. Mean	0.018	0.021	0.029
S.D. Ratio	0.422	0.479	0.697
Correlation	0.907	0.879	0.718

Seven inputs had better training and verification statistics, but slightly worse test cases statistics. In reality, there is little difference between the trials with all ten inputs or with the

seven inputs. Because the addition of three inputs makes very little difference to the network, it would best if they were not used at all.

2.8 Determining the Number of Hidden Layers

One of the simplest and most commonly asked question is: ‘How many hidden units should I use? Are there any rules of thumb?’ There are rules of thumb, but these are as unreliable as those for the complexity of a multiple or polynomial regression. The answer depends on the unknown underlying function that the neural network is approximating. There are three ways to control the complexity of the functions represented by a network: to cut out links, to change the number of hidden units, and to change the regularisation parameter.

Selecting the number of hidden units in a neural network is in principle no different from selecting regressors in a linear regression or the order of a polynomial regression. The main ideas that have been developed in that field are pruning by small steps (backward selection), incremental construction (forward selection), and minimizing some measure of performance over the class of possible models.

All the routines aim to predict the performance on a test set, and so to select the model with the best performance on the test set. The most general idea is to use cross-validation for neural nets. How do we move from the whole set to a subset for training? Do we start at the fitted weights for the whole data? This could bias the procedure. If we start at another random starting point, we could end up at a very different solution even with the whole data, let alone with a subset. This shows that the learning procedure for a neural network is not well defined, as there are often multiple local minima of rather different performance.

Once again our encapsulated software saviour, *Statistica Neural Networks*, comes to our rescue. *Statistica* does its best to make to the best decisions for our network, but no known algorithm is able to optimize the network perfectly. But, nonetheless, we trust logic of the software.

One way to investigate the effect of the number of neurons in the hidden layer is to run a series of networks where the only thing that changes between the networks is the number of hidden neurons. The base network will use seven inputs (see the previous section), 28 cases for testing, 10 for verification, and 12 for testing. The regression statistics will be shown for each case.

One hidden neuron:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0060	-0.0002	0.0033
Error S.D.	0.0145	0.0235	0.0232
Abs E. Mean	0.0126	0.0185	0.0180
S.D. Ratio	0.3081	0.3434	0.4576
Correlation	0.9514	0.9393	0.8970

Two hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0029	0.0004	-0.0025
Error S.D.	0.0215	0.0277	0.0320
Abs E. Mean	0.0161	0.0225	0.0201
S.D. Ratio	0.4567	0.4051	0.6301
Correlation	0.8896	0.9361	0.7977

Three hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0036	0.0009	-0.0012
Error S.D.	0.0206	0.0273	0.0333
Abs E. Mean	0.0160	0.0221	0.0216
S.D. Ratio	0.4374	0.3991	0.6557
Correlation	0.8993	0.9361	0.7641

Four hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	-0.0004	-0.0043	-0.0042
Error S.D.	0.0211	0.0278	0.0319
Abs E. Mean	0.0160	0.0204	0.0198
S.D. Ratio	0.4476	0.4066	0.6296
Correlation	0.8944	0.9345	0.7953

Five hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0043	0.0006	-0.0007
Error S.D.	0.0200	0.0285	0.0327
Abs E. Mean	0.0157	0.0228	0.0206
S.D. Ratio	0.4247	0.4166	0.6454
Correlation	0.9053	0.9225	0.7649

Six hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0077	0.0026	0.0019
Error S.D.	0.0199	0.0308	0.0297
Abs E. Mean	0.0165	0.0240	0.0204
S.D. Ratio	0.4237	0.4499	0.5857
Correlation	0.9058	0.9079	0.8114

Seven hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0068	0.0013	0.0044
Error S.D.	0.0201	0.0314	0.0326
Abs E. Mean	0.0162	0.0251	0.0238
S.D. Ratio	0.4273	0.4597	0.6426
Correlation	0.9041	0.9050	0.7728

Eight hidden neurons:

	<u>Train</u>	<u>Verif</u>	<u>Test</u>
Data Mean	0.0720	0.0724	0.0564
Data S.D.	0.0471	0.0684	0.0507
Error Mean	0.0076	0.0015	0.0004
Error S.D.	0.0196	0.0320	0.0336
Abs E. Mean	0.0164	0.0269	0.0241
S.D. Ratio	0.4159	0.4680	0.6633
Correlation	0.9096	0.8873	0.7525

To help reduce the data further some of the testing regression data was plotted on the same graph for the various numbers of hidden neurons. The test set was used because it is not used in training at all, and is designed to give an independent assessment of the network's performance when an entire network design procedure is completed. The two statistics that were plotted were:

1. **Abs. E. Mean.** Average absolute error (difference between target and actual output values) of the output variable.
2. **Correlation.** The standard Pearson-R correlation coefficient between the target and actual output values

A perfect prediction will have a correlation coefficient of 1.0. A correlation of 1.0 does not necessarily indicate a perfect prediction (only a prediction which is perfectly linearly correlated with the actual outputs), although in practice the correlation coefficient is a good

indicator of performance. It also provides a simple and familiar way to compare the performance of your neural networks with standard least squares linear fitting procedures. Obviously, a lower error mean and average absolute error is better. Using these guidelines,



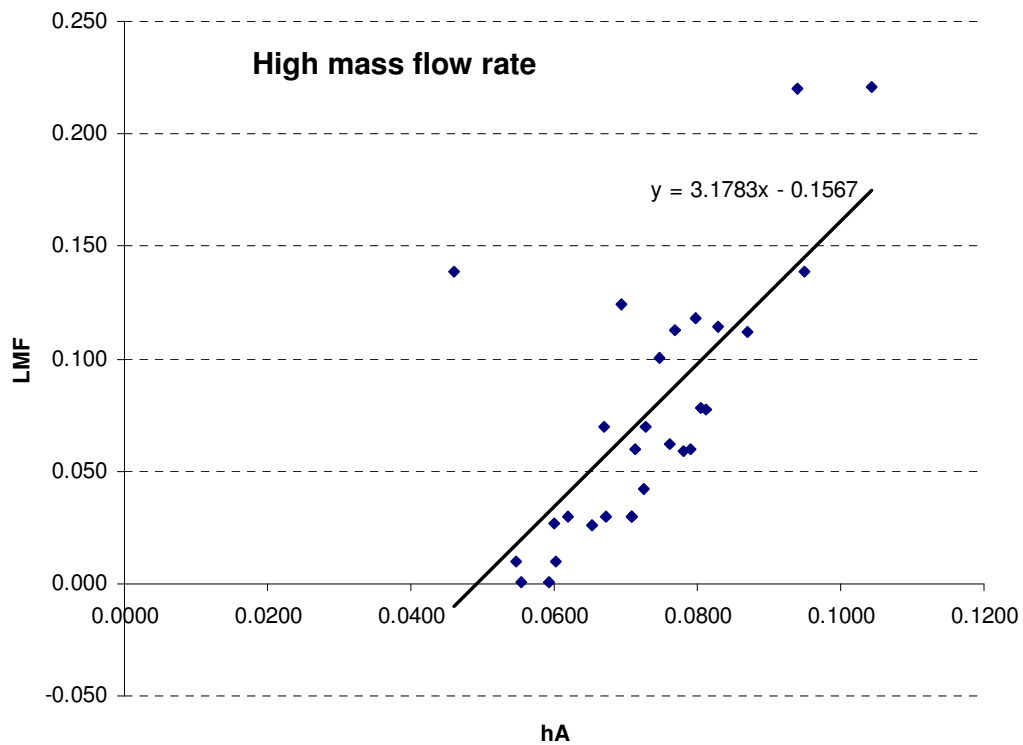
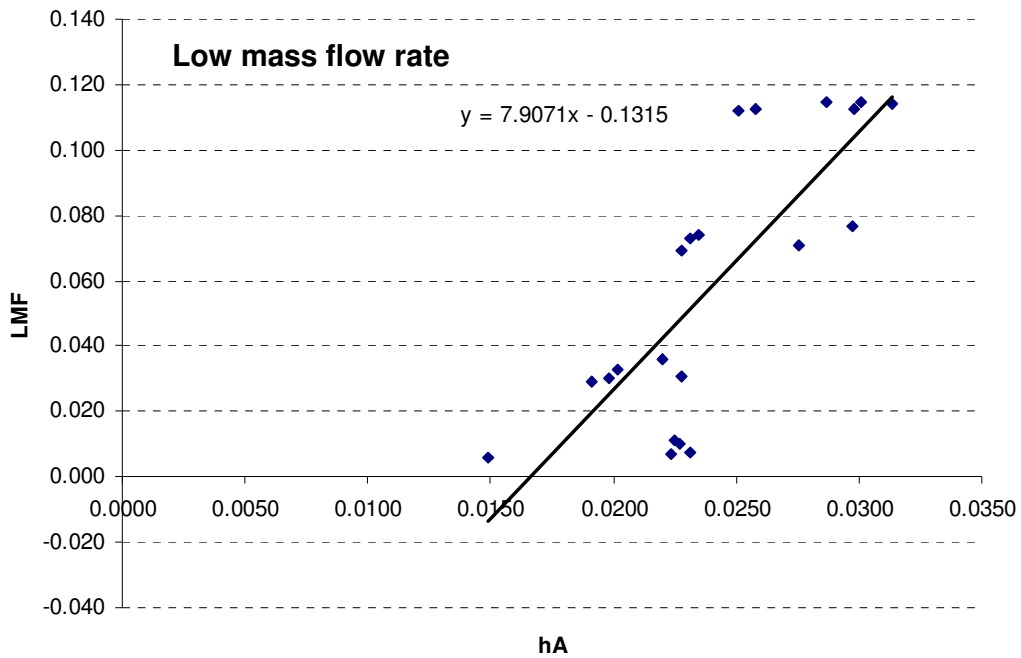
the optimal number of hidden neurons for this network with these parameters is actually one.

2.9 Comparing the Neural Network Model with Original Model

During the course of the research that has been presented in this document a simplified linear model was adapted. The purpose of the model was to render an output that would have relationship with the LMF of the refrigerant exiting the evaporator. This model would be incorporated in a control scheme that would be used to control the position of an electronic expansion valve. The model was based on the rate of heat transfer off the surface of sensor. As the rate of heat transfer coefficient on the surface of the sensor increased, it was assumed

that the LMF of the refrigerant also increased. This rate of heat transfer coefficient (called hA) was the feedback signal in the control loop (see section 1.3.3.4 for a more detailed explanation). This document presented the construction of a neural network used to serve the same purpose. This analysis begs the obvious question: ‘which method is a better predictor of LMF?’

Data from the original model was plotted as hA versus LMF.



The data was broken into two sets. One set of data had about 50 grams/sec of refrigerant flowing over the sensor. And the other set had about 25 grams/sec of refrigerant flowing over the sensor. In each case a straight line was fit to the data using the least-squares constraints to minimize the errors. This line was then used as a predictor of LMF. The LMF was predicted for each, and the error was calculated:

<u>hA</u>	<u>mdot [kg/sec]</u>	<u>LMF</u>	<u>predicted LMF</u>	<u>error</u>
0.0298	0.0256	0.077	0.1037	0.027
0.0276	0.0255	0.071	0.0866	0.016
0.0231	0.0256	0.073	0.0511	-0.022
0.0227	0.0255	0.069	0.0483	-0.021
0.0235	0.0256	0.074	0.0541	-0.020
0.0225	0.0255	0.011	0.0461	0.035
0.0227	0.0255	0.010	0.0480	0.038
0.0223	0.0255	0.007	0.0451	0.038
0.0232	0.0255	0.007	0.0516	0.044
0.0149	0.0254	0.006	-0.0136	-0.019
0.0228	0.0255	0.031	0.0487	0.018
0.0220	0.0256	0.036	0.0422	0.006
0.0202	0.0256	0.033	0.0280	-0.005
0.0191	0.0255	0.029	0.0198	-0.009
0.0198	0.0255	0.030	0.0250	-0.005
0.0251	0.0257	0.112	0.0667	-0.045
0.0258	0.0257	0.113	0.0725	-0.040
0.0287	0.0257	0.115	0.0955	-0.019
0.0298	0.0257	0.112	0.1042	-0.008
0.0301	0.0257	0.115	0.1065	-0.008
0.0314	0.0257	0.114	0.1165	0.002
0.1044	0.0483	0.221	0.1752	-0.046
0.0940	0.0484	0.220	0.1421	-0.078
0.0460	0.0514	0.139	-0.0104	-0.149
0.0949	0.0515	0.139	0.1449	0.006
0.0694	0.0493	0.124	0.0639	-0.060
0.0798	0.0494	0.118	0.0969	-0.021
0.0828	0.0513	0.114	0.1064	-0.008
0.0768	0.0514	0.113	0.0873	-0.026
0.0869	0.0514	0.112	0.1195	0.007
0.0805	0.0494	0.078	0.0992	0.021
0.0811	0.0489	0.077	0.1011	0.024
0.0670	0.0478	0.070	0.0562	-0.014
0.0762	0.0495	0.062	0.0855	0.023
0.0780	0.0496	0.059	0.0912	0.032
0.0726	0.0490	0.042	0.0739	0.032
0.0599	0.0488	0.027	0.0336	0.007
0.0652	0.0488	0.026	0.0506	0.025
0.0548	0.0488	0.010	0.0175	0.007
0.0746	0.0494	0.100	0.0804	-0.020
0.0727	0.0491	0.070	0.0742	0.004
0.0713	0.0490	0.060	0.0700	0.010
0.0790	0.0490	0.060	0.0945	0.035
0.0618	0.0488	0.030	0.0398	0.010
0.0673	0.0488	0.030	0.0573	0.027
0.0707	0.0491	0.030	0.0682	0.038
0.0709	0.0488	0.030	0.0687	0.039
0.0603	0.0489	0.010	0.0350	0.025
0.0592	0.0488	0.001	0.0315	0.030
0.0554	0.0488	0.001	0.0193	0.018

The average absolute error for this predictor (both high mass flow rate and low mass flow rate) was 0.0258, which is a significant error, but not bad for the amount of scatter in the data. The correlation coefficient between the actual LMF and the predicted LMF is 0.736

Similarly, data was run through the neural network to see how it performed:

<u>actual LMF</u>	<u>predicted</u>	<u>error</u>
0.077	0.084	0.007
0.071	0.079	0.008
0.073	0.078	0.005
0.069	0.079	0.010
0.074	0.076	0.002
0.011	0.004	-0.007
0.010	-0.004	-0.014
0.007	-0.005	-0.012
0.007	-0.005	-0.012
0.006	-0.007	-0.013
0.031	0.057	0.026
0.036	0.057	0.021
0.033	0.057	0.024
0.029	0.045	0.016
0.030	0.049	0.019
0.112	0.107	-0.005
0.113	0.110	-0.003
0.115	0.108	-0.007
0.112	0.112	0.000
0.115	0.118	0.003
0.114	0.124	0.010
0.221	0.221	0.000
0.220	0.225	0.005
0.139	0.118	-0.021
0.139	0.096	-0.043
0.124	0.111	-0.013
0.118	0.088	-0.030
0.114	0.085	-0.029
0.113	0.101	-0.012
0.112	0.074	-0.038
0.078	0.117	0.039
0.077	0.068	-0.009
0.070	0.086	0.016
0.062	0.096	0.034
0.059	0.082	0.023
0.042	0.050	0.008
0.027	0.062	0.035
0.026	0.048	0.022
0.010	0.051	0.041
0.100	0.089	-0.011
0.070	0.074	0.004
0.060	0.078	0.018
0.060	0.063	0.003
0.030	0.063	0.033
0.030	0.051	0.021
0.030	0.044	0.014
0.030	0.043	0.013
0.010	0.033	0.023
0.001	-0.006	-0.007
0.001	-0.006	-0.007

The characteristics of this particular network were derived from the analysis presented in this document. One hidden layer with one neuron was used. 28 cases were used for training, 10 were used for verification, and 12 were used for testing. The regression statistics are as follows:

	<u>Train</u>	<u>Verification</u>	<u>Test</u>
Data Mean	0.072	0.072	0.056
Data S.D.	0.047	0.068	0.051
Error Mean	0.006	0.000	0.003
Error S.D.	0.016	0.024	0.024
Abs E. Mean	0.014	0.019	0.019
S.D. Ratio	0.333	0.347	0.474
Correlation	0.943	0.938	0.888

The mean of the absolute value of the errors in all three sets of data is significantly less than 0.0258. The correlation coefficient was much higher for all the sets of data in the neural network than 0.736 (correlation coefficient for the original model).

So, it appears (from this analysis) that a neural network is a better predictor of the LMF of refrigerant than the original model.

2.10 Further Simplification

In order for a neural network to be practical in this particular application, it must be able to estimate the LMF of refrigerant a rate faster than the sample period of the controller. The controller must have an accurate prediction of the LMF before it can send out its control signal to the electronic expansion valve. Reducing the number of inputs to controller is one way to simplify the algorithm. A sensitivity analysis was performed on the network in attempt to reduce the number of inputs (see section 2.7). Only one input was pruned as a result of that analysis. Then, correlations between the variables were used to prune two more variables. These methods are based on numerical statistics. While these methods do work, they are not able to prune all of the variables that are not absolutely needed. Intuitive knowledge about the data set can help logically reduce the data without crunching any numbers.

From the seven remaining inputs I will prune the inputs one by one. The order of the pruning is purely subjected, and based on my expertise on the domain in question. The control case in this experiment will use the seven inputs that were deduced in section 2.7 and the network structure found in section 2.8 (1 hidden neuron). Recall that this network has the following regression statistics:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.072	0.072	0.056
Data S.D.	0.047	0.068	0.051
Error Mean	0.006	0.000	0.003
Error S.D.	0.016	0.024	0.024
Abs E. Mean	0.014	0.019	0.019
S.D. Ratio	0.333	0.347	0.474
Correlation	0.943	0.938	0.888

The first variable that I have chosen to discard is Tmix. Tmix is numerically close to Tref_out. Tmix can also usually be determined by knowing what Pmix. It is usually the case that when Tmix is measured that the refrigerant is in a saturated state. Thus, Tmix is a function of Pmix (and vice versa). The regression statistics for the same network but with only six inputs are:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.072	0.072	0.056
Data S.D.	0.047	0.068	0.051
Error Mean	0.006	0.001	0.004
Error S.D.	0.017	0.024	0.026
Abs E. Mean	0.014	0.019	0.020
S.D. Ratio	0.354	0.356	0.510
Correlation	0.935	0.936	0.873

The statistics are very similar, but maybe a bit worse. The abs error mean for the test cases went from 0.019 to 0.020. For all of the networks used in this section the same cases were used for training, verification, and testing.

The next variable to be removed in dPevap. This is an important variable in estimating the speed of the refrigerant flowing over the sensor. Since the sensor actually measures the heat transfer coefficient of its surface, the speed of the refrigerant is an important factor to know. But, we still have the mass flow rate of the refrigerant (mdot). The speed of the refrigerant can be inferred from mdot. The regression statistics for the same network with five inputs (dPevap and Tmix taken out):

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.072	0.072	0.056
Data S.D.	0.047	0.068	0.051
Error Mean	0.007	0.001	0.009
Error S.D.	0.016	0.025	0.031
Abs E. Mean	0.013	0.020	0.024
S.D. Ratio	0.333	0.362	0.612
Correlation	0.943	0.932	0.791

Once again the statistics are very similar to those in the previous network. Thus far, removing these two variables has had little effect on the performance of the network. Let's try to remove only more variable. Pmix is the next logical variable to remove. Pmix was very strongly correlated with the other variables. At this point we should really start to see the performance of the network suffer. The statistics for the network with only four variables are as follows:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.072	0.072	0.056
Data S.D.	0.047	0.068	0.051
Error Mean	0.002	0.000	-0.001
Error S.D.	0.028	0.038	0.046
Abs E. Mean	0.020	0.031	0.034
S.D. Ratio	0.586	0.562	0.901
Correlation	0.811	0.841	0.435

The statistics do show that there is a significant degrade in the performance of the network without this variable. At this point the abs error mean for the test cases for this network has actually dropped below the abs error mean for the original model (0.0258). But, the training cases still perform better than the original model. The training cases would actually be a more fair comparison with the original model. Because with the original model, all of the data points were used to fit a line through the data points. Then those same data points were used to assess the performance of the model. So there is no measure of the original model's ability to generalize to other cases.

Also, in defense of the neural network model the original model used two separate models for the two different mass flow rates. To do a completely fair comparison between the two types of model, we should construct each model with the same domain of knowledge. Two neural networks will be constructed (one for high mass flow rate and the other for the lower mass flow rate), and the neural networks will only have the inputs that the original model had (Tsensor, VDC, and Tref_in). The networks will have one hidden neuron. The regression statistics for the network for the cases with the lower mass flow rate are:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.057	0.075	0.052
Data S.D.	0.042	0.051	0.041
Error Mean	0.000	0.000	-0.003
Error S.D.	0.002	0.003	0.006
Abs E. Mean	0.002	0.002	0.004
S.D. Ratio	0.057	0.059	0.148
Correlation	0.998	0.998	0.989

The network performed extremely well. The abs error mean for the test cases was only 0.004. The corresponding abs error mean for the original model (for just the low mdot data) is 0.021; more than five times greater.

A similar network was constructed for the high mass flow rate data. The results went as followed:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.069	0.087	0.077
Data S.D.	0.058	0.077	0.047
Error Mean	-0.011	0.000	-0.021
Error S.D.	0.026	0.028	0.057
Abs E. Mean	0.022	0.020	0.042
S.D. Ratio	0.441	0.362	1.219
Correlation	0.898	0.933	0.332

This network didn't perform nearly as well. The abs error mean for the test cases was 0.042. The corresponding abs error mean for the original model (for just the high mdot data) is 0.029. So, the original model actually performed better than the neural network did for the test cases. But, the neural network performed better in the training cases (abs error mean of 0.022).

2.11 Other Types of Networks

Up to this point only Multilayer Perceptron networks have been considered. Other types of neural networks include:

- Radial Basis Function network
- Kohonen network
- Linear network
- (Bayesian) Probabilistic network
- (Bayesian) Regression network
- Perceptron
- ADALINE

Given the nature of the relationships between the variables and the output, the best type of network to try on this set of data is the linear network. The system of variables has a much more simple nature than was originally thought. Many of these networks work well for complex systems, and others only work for classification problems.

Linear networks have only two layers: an input and output layer, the latter having linear PSP and activation functions. Many problems cannot be solved (or solved well) by linear techniques; however, many others can, and it is poor practice to neglect a simple technique in favor of more complex ones without comparison. You should always train a linear network, as a standard of comparison for the more complex non-linear ones. Linear networks are best trained using the pseudo-inverse technique: this optimizes the last layer in any network, providing it is a linear layer. Pseudo-inverse is fast, and guaranteed to find the optimal solution. You may also use back propagation, quick propagation, or Delta-bar-Delta, to optimize a linear network, if you desire. This is not usually good practice, although occasionally the pseudo-inverse technique suffers from numerical problems, in which case iterative training provides a fall-back position.

The first linear network constructed used all ten of the original inputs. The regression statistics are as follows:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.000	-0.006	0.000
Error S.D.	0.010	0.014	0.017
Abs E. Mean	0.007	0.013	0.013
S.D. Ratio	0.201	0.211	0.381
Correlation	0.980	0.979	0.926

Recall that the MLP network with the same number inputs had the following regression statistics:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.012	0.000	0.012
Error S.D.	0.013	0.020	0.020
Abs E. Mean	0.015	0.015	0.018
S.D. Ratio	0.261	0.308	0.430
Correlation	0.966	0.966	0.903

Note that the linear network performs significantly better than its MLP counter part.

The next linear network constructed used the seven inputs that were found in section 2.7. The regression statistics are as follows:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.000	-0.018	0.002
Error S.D.	0.028	0.030	0.030
Abs E. Mean	0.021	0.023	0.023
S.D. Ratio	0.572	0.461	0.669
Correlation	0.820	0.911	0.807

There was a significant degrade in the performance of the network after the three inputs were removed. The abs error mean of the test cases dropped from 0.013 to 0.023 in the linear networks. But the corresponding MLP networks had no significant change in the abs error mean of the test cases (0.018).

The next linear network that will be looked at will have Tmix input removed. This was the next variable that was pruned in section 2.10. The following are regression statistics for a six input linear network:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.000	-0.018	0.002
Error S.D.	0.028	0.030	0.030
Abs E. Mean	0.021	0.023	0.023
S.D. Ratio	0.572	0.461	0.669
Correlation	0.820	0.911	0.807

There is virtually no change in the performance of the network from removing Tmix. The MLP network still performed better with six inputs (abs error mean of the test cases of 0.020).

The next linear network will remove the dPevap variable. The regression statistics are as follows:

	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.067	0.091	0.056
Data S.D.	0.050	0.066	0.045
Error Mean	0.000	-0.015	0.006
Error S.D.	0.032	0.043	0.039
Abs E. Mean	0.024	0.034	0.031
S.D. Ratio	0.649	0.648	0.865
Correlation	0.761	0.765	0.502

At this point the performance of the network is beginning to drop below acceptable levels, so no more variables will be removed. The corresponding MLP network had a abs error mean of 0.024 for the test cases.

2.12 Confidence of Results

Determining confidence is difficult. One possible way to estimate confidence is to train and verify with different sets of data and see how close the network weightings are in each case. If they are the same then you could have a high level of confidence. This exercise will use a network having seven inputs (see section) and one hidden neuron. There will be five different variations in what data is used for training, verification, and testing.

trial 1	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.0653	0.0577	0.0811
Data S.D.	0.0541	0.0309	0.0589
Error Mean	-0.0112	-0.0006	-0.0131
Error S.D.	0.0187	0.0175	0.0200
Abs E. Mean	0.0176	0.0138	0.0159
S.D. Ratio	0.3451	0.5655	0.3397
Correlation	0.9386	0.8783	0.9414

trial 2	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.0782	0.0601	0.0548
Data S.D.	0.0610	0.0398	0.0363
Error Mean	-0.0072	-0.0006	0.0097
Error S.D.	0.0395	0.0336	0.0287
Abs E. Mean	0.0318	0.0255	0.0235
S.D. Ratio	0.6471	0.8453	0.7917
Correlation	0.8014	0.5449	0.6123

trial 3	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.0619	0.0490	0.0934
Data S.D.	0.0416	0.0423	0.0678
Error Mean	-0.0068	-0.0003	-0.0284
Error S.D.	0.0202	0.0115	0.0456
Abs E. Mean	0.0168	0.0089	0.0357
S.D. Ratio	0.4853	0.2716	0.6719
Correlation	0.8743	0.9697	0.7780

trial 4	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.0714	0.0731	0.0595
Data S.D.	0.0505	0.0649	0.0487
Error Mean	-0.0024	-0.0005	-0.0091
Error S.D.	0.0144	0.0163	0.0226
Abs E. Mean	0.0123	0.0117	0.0196
S.D. Ratio	0.2839	0.2516	0.4642
Correlation	0.9589	0.9679	0.8903

trial 5	<u>train</u>	<u>verify</u>	<u>test</u>
Data Mean	0.0646	0.0723	0.0731
Data S.D.	0.0508	0.0425	0.0621
Error Mean	0.0062	0.0001	0.0059
Error S.D.	0.0097	0.0137	0.0156
Abs E. Mean	0.0086	0.0109	0.0128
S.D. Ratio	0.1910	0.3230	0.2508
Correlation	0.9817	0.9489	0.9688

There does seem to be significant variation in the data. For example the abs error mean values for just the test cases are {0.0128, 0.0159, 0.0196, 0.0235, 0.0357}, and the same statistic for the training cases was as high as 0.0318, and as low as 0.0086. Nonetheless, neural network is never an exact science, and variance such as these is inevitable.

2.13 Improving the Analysis

There is much more that can be done to improve the analysis. All of networks presented in this report use the default activation functions. These are usually best suited for most problems, but different activation functions can improve the performance of the network. *ST Neural Networks* supports a wide range of activation functions. Only a few of these are used by default; the others are available for customization.

Linear. The activation level is passed on directly as the output. Used in a variety of network types, including linear networks, and the output layer of radial basis function networks.

Logistic. This is an S-shaped (sigmoid) curve, with output in the range (0,1). The most commonly-used neural network activation function.

Hyperbolic. The hyperbolic tangent function: a sigmoid curve, like the logistic function, except that output lies in the range (-1,+1). Often performs better than the logistic function because of its symmetry. By default, not used in any network types. Ideal for customization of multilayer perceptrons.

Exponential. The negative exponential function. Ideal for use with radial units. The combination of radial PSP function and negative exponential activation function produces units which model a Gaussian (bell-shaped) function centered at the weight vector. The standard deviation of the Gaussian is given by the formula below, where d is the "deviation" of the unit stored in the unit's threshold:

Softmax. Exponential function, with results normalized so that the sum of activations across the layer is 1.0. Used in the output layer of specialized multilayer perceptrons for classification problems, so that the outputs can be interpreted as probabilities of class membership.

Square root. Used to transform the squared distance activation in a Kohonen network to the actual distance as an output.

Sine. Possibly useful if recognizing radially-distributed data; not used by default.

Ramp. A piece-wise linear version of the sigmoid function. Relatively poor training performance, but fast execution.

Step. Outputs either 1.0 or 0.0, depending on whether the PSP value is positive or negative.

One way to possibly improve a network is to artificially inject noise into the system. Adding noise can have similar benefits to shuffling, in allowing the network to escape local minima during training. It can also provide better generalization performance, by preventing the network from overfitting the training data.

There are wide variety of different types of learning algorithms that were not explored at all. Training algorithms are divided into three types.

Initialization algorithms. These are not really training algorithms at all, but methods to initialize weights (usually randomly) prior to training proper. They do not require any training data.

Supervised learning. These algorithms alter weights and/or thresholds, using sets of training cases that include both input and target output values.

Unsupervised learning. Alter weights and/or thresholds using sets of input training cases (output values are not required, and if present are ignored). The unsupervised learning techniques are mainly used to assign weights and thresholds in radial units; the supervised techniques are mainly used to assign weights and thresholds in linear PSP units.

Some network types require a combination of supervised and unsupervised training algorithms; for example, the radial units in a radial basis function network must first be assigned using unsupervised techniques, with the linear units in the output layer being subsequently assigned using supervised techniques. In general, different techniques can be

freely combined in training any particular network, although each technique may have restrictions over where it can be used (e.g., the Kohonen algorithm can only be used if the first hidden layer consists of radial units).

ST Neural Networks also features some auxiliary features associated with training. This includes the random initialization of weights, the definition of stopping conditions for the training algorithms (in particular, training can be stopped once the error performance of the network begins to deteriorate) and the retention of the best network discovered during an iterative training run.

More could have been done with dimensionality reduction. The only systematic forms of dimensionality reduction that were looked at in detail were sensitivity analysis and reduction by using the correlation matrix. But, we could have searched based on network performance using genetic algorithm, forward stepwise selection, or backward stepwise selection. By creating a pre-processing network the effective dimensions on the inputs could have been reduced by essentially compressing the data before it is fed through the main network.

Conclusion

Two ways of reducing data to estimate the LMF of refrigerant were presented in this document. The first method (in section 1.3) used classical techniques to form linear combinations of the data to approximate the desired result. The second method used neural networks to develop non-linear relationships between the variables. In most cases the neural networks performed better. The 'best' neural network found was a linear type that used all ten of the inputs. Another good network was a MLP network that used the seven inputs found in section 2.7.

BIBLIOGRAPHY

1. Barnhart, J.S. & Peters, J.E., (1992). *An Experimental Investigation of Flow Patterns and Liquid Entrainment in a Horizontal -Tube Evaporator*, ACRC Technical Report #28, December, 234 pp.
2. Berlin, Howard M. & Frank C. Getz, Jr, (1988). **Principles of Electronic Instrumentation and Measurement**, Macmillan Publishing Co., New York, NY, 1988.
3. Buck, C. B., (1995). *Measurements of Short-Term Flow Processes in Refrigeration Systems*. Published by Elsevier Science Ltd and IIR, 1996. Printed in Great Britain. 0140-7007(96)00087-9.
4. Collins, C.D., N.R. Miller, & W.E. Dunn, (1996). *Experimental Study of Mobile Air Conditioning System Transient Behavior*. ACRC Technical Report 102, July 1996.
5. Davis, M. R. (1970). *The Dynamic Response of Constant Resistance Anemometers*, Journal of Physics E: Scientific Instruments 1970. Printed in Great Britain.
6. Figliola, Richard S. & Donald E. Beasley (1991). **Theory and Design for Mechanical Measurements**, John Wiley & Sons, Inc., 1991.
7. Forsythe, George E., Michael A. Malcolm, & Cleve B. Moler, (1977). **Computer Methods for Mathematical Computations**, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.
8. Gayakward, Ramakant A., (1993). **Op-Amps and Linear Integrated Circuits**, Regents/Prentice Hall, Englewood Cliffs, NJ, 1983, 1988, & 1993.
9. Hrnjak, P.S., M. A. Shannon, T. M. Leicht, & N.R. Miller (2000). *Detection of Liquid Mass Fraction at the Evaporator Exit of Refrigeration Systems*, To appear in International Journal of Thermal Sciences - Revue Generale de Themique, 2000.
10. Horowitz, P., W. Hill, (1996). **The Art of Electronics**, Press Syndicate of the University of Cambridge, Copyright 1980 & 1989.
11. Johnston, C. E., N. R. Miller, & W. E. Dunn, (1997). *Refrigeration Charge Loss Detection for a Mobile Air Conditioning System*, ACRC Technical Report 125, July 1997.
12. Kuo, Benjamin C., (1980). **Digital Control Systems**, Holt, Rinehart and Winston, Inc., 1980.
13. Ljung, Lennart, (1991). *System Identification Toolbox User's Guide*, for use with MATLAB, The Math Works, Inc., 1986-1991.
14. Lomas, C. G., (1986). **Fundamentals of Hot Wire Anemometry**, Cambridge University Press.

15. Luo, F. L., Unbehauen, R. (1997). **Applied Neural Networks for Signal Processing**, Cambridge University Press.
16. Miller, James A. (1976) *A Simple Linearized Hot-Wire Anemometer*, Journal of Fluids Engineering, December 1976.
17. Moran, Michael J. & Howard N. Shapiro, (1995). **Fundamentals of Engineering Thermodynamics**, Third Edition, John Wiley & Sons, Inc., 1995.
18. Ogata, Katsuhiko, (1997). **Modern Control Engineering**, Prentice Hall, Upper Saddle River, NJ, 1997, 1990, & 1970.
19. Page, G. F., Gomm, J. B., Williams, D. (1993). **Application of Neural Networks to Modelling and Control**, Chapman & Hill
20. Perry, A. E. (1982). **Hot-Wire Anemometry**, Oxford Science Publications, Oxford University Press, 1982.
21. Ripley, B. D.(1996). **Pattern Recognition and Neural Networks**, Cambridge University Press.
22. Rubio-Quero, J.E., W.E. Dunn, & N.R. Miller, (1995). *A Facility for Transient Testing of Mobile Air Conditioning Systems*, ACRC Technical Report 80, June 1995.
23. Shannon, M. A., P. S. Hrnjak, & T. M. Leicht. *Exploratory Research on MEMS Thechnology for Air Conditioning and Heat Pumps* EPRI Report TR-1111699.
24. Shannon, M. A., T.M. Leicht, P.S. Hrnjak, N.R. Miller, & F.A. Kahn (2000). *Thin-Film Resistance Sensor for Measuring Liquid Mass Fraction in Super-Heated Refrigerant*, Sensors and Actuators A - Physical in Dec 1999.
25. Simpson, R. L., K. W. Heizer, & R. E. Nasburg, (1979). *Performance Characteristics of a Simple Linearized Hot-Wire Anemometer*, Journal of Fluids Engineering, September 1979.
26. Solberg, J. R., N. R. Miller, & P. S. Hrnjak, (1999). *A Sensor for Estimating the Liquid Mass Fraction of the Refrigerant Exiting an Evaporator*, SAE paper 2000-01-0976, 2000.
27. Solberg, J. R. (2000) *Regulation of the Liquid-Mass-Fraction of the Refrigerant Exiting an Evaporator*, Master of Science Thesis, University of Illinois at Urbana-Champaign.
28. Spina, E. F. & C. B. McGinley, (1994). *Constant-Temperature Anemometry in Hypersonic Flow: Critical Issues and Sample Results*, Experiments in Fluids 17 (1994) 365-374. Springer-Verlag 1994.
29. Stoecker, W. F., (1966), *Stability of and Evaporator-Expansion Valve Control Loop*, ASHRAE Transactions, Vol. 72, Part II.

30. Toral, H., (1981). *A Study of the Hot-Wire Anemometer for Measuring Void Fraction in Two Phase Flow*, Journal of Physics E: Sci. Instrum., Vol 14, 1981. Printed in Great Britain.
31. US Patents: #5289692 (1994), #5335513 (1994), #5390897 (1995), #5460349 (1995), #5477701 (1995), #5522231 (1996).
32. Wandell, E.W., W.E. Dunn, & N.R. Miller ,(1997). *Experimental Investigation of Mobile Air Conditioning System Control for Improved Reliability*, ACRC Technical Report 128, August 1997.
33. Wedekind, G. L., (1965), *Transient response of the mixture -vapor transition point in two-phase horizontal evaporating flow*, Ph.D.
34. Wedekind, G. L. & Stoecker W. F., (1966), *Transient Response of the Mixture - Vapor Transition Point in Horizontal Evaporation Flow*, ASHRAE Transactions, Vol. 72, Part II.
35. Weston, P.G., W.E. Dunn, & N.R. Miller, (1996). *Design and Construction of a Mobile Air-Conditioning Test Facility for Transient Studies*, ACRC Technical Report 97, May 1996.
36. Whitchurch, M. R., W. E. Dunn, & N.R. Miller, (1997). *Humidity Effects in Mobile Air-Conditioning Systems*, ACRC Technical Report 126, August 1997.