Nelson Mandela
Metropolitan
University

for tomorrow

Research Dissertation
For

Modelling and intelligent control of vehicle climatronic system

Submitted to

Faculty of Engineering, the Built Environment

and Information Technology

Department of Electrical Engineering

For the proposed research in the programme
Magister Technologiae: Engineering: Electrical

By
Student: J. Sun
Student No: 20423052

Promoter: Prof. Dr T. I. Van Niekerk, Prof. Dr H Holdack-Janssen
Project leader: Mr.Juergen Kranz

Copyright statement

Abstract

The modelling and control method of a vehicle climatronic system, based on MATLAB/SIMULINK, is presented. In order to achieve high modelling accuracy, a developed simulation model library is introduced. The modelling approach is described and the developed models are validated with some of experimental data obtained. The models are nonlinear, independent of fluid type and based on thermo-dynamic principles. Analysis of the cooling circuit modelling and empirical real-time control models are created by using Fuzzy logic controller and Stateflow. Both of control input and output are implemented essentially at original vehicle CAN-Bus system. Feasible digital automatic control strategy basic to fuzzy theory, hardware and software solution are given. The simulation experiment is achieved with the Hardware-in-Loop technology. This control methodology is easily operated and worth applying for any further studies or methods.

Acknowledgements

I would like to acknowledge the contributions made by the following people and institutions:

- My promoters- Professor Theo van Niekerk Professor Hinrich Holdack-Janssen and Mr-Juergen Kranz for their support, guidance and encouragement; In addition, I would like to thank my German teammate (Gritzner Sebastian, Smuda Daria, Schnell Stefan, Eitelbuss Tobias and Frohlich Martin) for their assistance; My most grateful thanks goes to my parents and Chinese colleagues (Shengkai Wu, Xiaopeng Wu), for their love and encouragement;

- Faculty of Engineering at the Nelson Mandela Metropolitan University for providing the opportunity for me to complete research work abroad;

- The staff of the Institute for Thermodynamics of the Technical University of Jiao Tong for their guidance and support during the project that I have undertaken.

Author's declaration

At no time during the registration for the Magister Technologiae Degree has the author been registered for any other Technikon or University degree. All sources used or referred to have been documented and recognized.

Signed...................................

Date.......................................

Nomenclature

α	Heat transfer coefficient	m	Mass
δ	Declination	p	Pressure
$\dot{m}$	Mass flow rate	s	Specific entropy
η	Efficiency	T	Temperature
τ	Time interval	t	Time
φ	Relative humidity	$\dot{Q}$	Heat
ρ	Density	L	Length
C_p	Heat capacity coefficient	V	Voltage; Volume
h, H	Specific enthalpy, Enthalpy	P	Cross-sectional area
D_h	Hydraulic diameter	A, a	Surface
Nu	Nusselt number	$\Pr$	Prandtl number
Re	Reynolds number	μ	Fluid viscosity
f	Fiction factor		

Indices

$''$	Saturated liquid	a	Air
$'$	Saturated vapour	$comp, c$	Compressor
i	Outside	$cond, d$	Condenser
o	Inside	$evap, e$	Evaporator
r	Refrigerant	w	Wall of tube

Table of contents

-Chapter 1

Introduction

Climatronic is a fully electronic climate control system. It was introduced on the Golf and Passat in the mid- 1990s when only about 10 percent of all newly registered vehicles were fitted with an air conditioning system. By 2000, air conditioners were being installed as a standard feature in more than one in four newly registered vehicles. Owing to customer demand, climatronic has become an integral part of the vehicle's specification. It is anticipated that climatronic systems will improve further through the development of the refrigerant circuit within air conditioners generally.

Traditionally, Heating, Ventilating, and Air Conditioning (HVAC) design alternatives were evaluated by testing prototypes in wind tunnel. This is a very time-consuming and expensive process. The use of an air-conditioned wind tunnel is inevitable when experiments need to be carried out with reproducible ambient conditions. This combined with high costs of installation and operation of a wind tunnel is a major problem. This project research outlines the efforts made to develop a dynamic cooling circuit's model of an air-conditioning system suitable for multivariable control design purposes. The significance of this simulation is that it can be utilized for testing or investigating independent components or entire refrigerant cycles of the future. The intelligent climate control algorithm enables a strategy for microprocessor, which can be incorporated into the existing CAN Bus-system.

1.1. Background

Currently the trend in automotive engineering is to develop control-oriented models for vehicle’s climatronic systems. The motivation for this work can be divided into three categories: efficiency, performance and a better understanding.

Billions of EUROs are spent on energy for academic, industrial and automotive air conditioning and refrigeration system. An increase of efficiency in HVAC systems would have a notable effect on the automotive economy generally as well as impacting on the cost of vehicles. Progress has been made to improve component efficiency in HVAC systems. With the increasing availability of inexpensive computing power, increased efficiency is possible using complex control techniques and multivariable control strategies.

When comparing commercial and vehicle application of HVAC systems, vehicles operate in a much harsher environment. Many parameters affect its performance, for instance, air temperature, surrounding surface temperatures, air velocity, humidity and direct solar flux. The thermal comfort for humans can be improved and used as a strategy to optimize HVAC systems. It is an important concern for occupants in an enclosed environment such as the passenger compartment of a vehicle. Multiple-Input Multiple-Output (MIMO) control strategies have to achieve multiple performance objectives within HVAC system. The solution to the problem again lies in model-based multivariable control strategies. By designing a general lumped parameter model, a large number of HVAC systems could be included within the framework. Multivariable control schemes developed using a form of this model would allow the control scheme to be adapted based on physical parameters of the individual systems. In this project, these control schemes could be designed to fit multiple performance objectives such as maximizing Compressor Performance (COP), exchanging the capacity when needed and controlling not only temperature, but also humidity.

Developing cooling circuit modelling control systems will increase our understanding of the dynamics system. A mathematical model of thermo fluid dynamics is complete. Little physical insight is gained by evaluating the general form of the governing equations. By modelling the cooling circuit, many dynamic phenomena can be neglected, but we have to find out the dominant dynamic phenomena of an air conditioning system.

Accurate vehicle HVAC systems are demanded for controller synthesis and a physical test rig is required for a portion of controller verification. The real performance test should be the inside of a vehicle cabin. Often the tasks of system identification, controller synthesis, and controller verification are done using various software and analysis tools that are not directly compatible with each other. This may lead to complications and errors when the data are transported between the various platforms. Furthermore, it is often necessary to write a custom code to implement different controllers. This is a time-consuming and error-prone task. In order to alleviate these problems, an integrated experimental setup will be developed to allow for Data Acquisition (DAQ), modelling, simulation, and controller design. Simulation and verification will take place in single integrated software/ hardware environment.

The following section will discuss how the proposed topic fits into the research area described as well as to discuss related work.

1.2. Overall aim

The overall aim of this study is to develop a modelling and simulation of the refrigerant cycle of a vehicle Air-Conditioning systems, and an intelligent control algorithm.

In order to improve the Thermal Comfort in the passenger compartment, a number of sub-problems need to be addressed:

i. Modelling

- How to make assumptions about cooling circuit modelling to minimize the effect on the control system design?
- To consider dynamic modelling of the components and the whole refrigerant cycle.

ii. Hardware and software

- How to build Hardware in Loop (HIL) mode to deal with the interfacing between hardware and software?
- How accurate time responses of the Real-Time Workshop (RTW) have to be acquired for software?

iii. Control

- How should the cooling circuit modelling be evaluated?
- What are the considerations for control objective onto vehicles air-conditioner systems?
- How to create strategy regulations for heat exchanging?

Effectively answering the above problems would ensure that the following objectives are achieved.

1.3. Hypothesis

A simulation that takes into account variation of the environment to be analyzed and and optimal controller optimal to control energy utilization within a vehicle in air conditioning systems. This model should also include the control object by which a refrigerant cycle would be realized; the solar-sensor and original built sensor information represent temperature distribution in the cabin. The self-tuning fuzzy PI controlling method would be used. The strategy of controller modelling will concern vehicle thermal comfort level, well-known thermal comfort has been evaluated by the performance of many more HVAC system.

1.4. Literature Review

This section describes background to the need for and means of a Fuzzy Logic PID controller in HVAC systems.

1.4.1. PID controller

A proportional-integral-derivative (PID) controller is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a desired set-point by calculating and then outputting a corrective action that can adjust the process accordingly.

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de}{dt}$$

The PID controller algorithm involves three separate parameters; the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral value determines the reaction based on the sum of recent errors and the Derivative value determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element.

There are several methods for determining the development of PID controller parameters. Some employ information about open-loop step response, for example Coben-Coon reaction curve method; other methods are knowledge of the Nyquist curve, for example, the Ziegler-Nichols frequency-response method. However, these methods use only a small amount of information about the dynamic behaviour of the system, and often do not provide good tuning.

Conventional PID controllers are widely used in air temperature control duet to their simplicity in arithmetic; ease of use; good robustness; high reliability; stabilization and zero steady state error. The common perception is that temperature control is a mature and largely unchanging area of technology. There are still other applications, which require not only precise temperature control, but also a faster warm-up/cool-off phase and quicker response to disturbances with minimal overshoot and under shoot when the set-point temperature has been reached. It is difficult to achieve desired tracking control performance since tuning and self-adapting adjusting parameters on line are a scabrous problem of PID controller. (Available: http://en.wikipedia.org/wiki/PID_controller [4, May 2008])

1.4.2. Fuzzy Logic

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. Just as in fuzzy set theory the set membership values can range between 0 and 1. In fuzzy logic the degree of truth of a statement can range between 0and 1 and is not constrained to the two truth values as in classic predicated logic. And when linguistic variables are used, these degrees may be managed by specific functions.
(Available: http://en.wikipedia.org/wiki/Fuzzy_logic [24, May 2008])

Fuzzy control in its basic form mimics a manual control process (Available: http://www.controleng.com/article/CA372359.html [20, May 2008]). A person regulates power applied to a heating element according the monetary temperature deviation from the set-point value and the rate of the temperature change. The entire process is governed more by a sense than by the knowledge of its physical or mathematical behaviour. Are the temperature deviation and rate of the temperature change high, low, or medium? Fuzzy control works exactly with the same states of process variables.

However, fuzzy logic controller with fixed scaling factors and fuzzy rules may not provide perfect control performance if the controlled plant is highly nonlinear and uncertain. And conventional fuzzy logic controller may result in steady state errors if the system does not have an inherent integrating property.

1.4.3. *Fuzzy-PID controller*

The simple process can become a complex task if additional features are needed. Implementing the enhanced, traditional PID controller can be a challenge, especially if auto-tuning capabilities to help find the optimal PID constants are desired. On the other option, fuzzy control seems to accomplish the same control quality with less complexity. Approximation of the second – order switching curve used in time-optimal control systems by a polynomial of the first or higher order makes fuzzy control a better candidate for time-optimal control applications.

In order to improve conventional fuzzy logic and PID controller, some controllers have been proposed in the literature such as *Fuzzy-PID controller*.

1.4.4. *Control Architectures*

This hybrid Fuzzy PID controller (see *Figure 1*) is not the same as the original fuzzy logic method which is mentioned by W.L.Bialkowshi. The difference between the two control architectures is additional to fuzzy rule B. The parameter of PI controller is kept a constant in the original control system, for this reason we add a fuzzy rule B which can affect Ki value during the on-line control. At the beginning of control processing, the value of Ki factor will be the biggest one, and nothing is affected by the PID controller; the whole running system depends on the fuzzy rule. A manual control process would be suit for any systems. When the system output closes to stabilization, the Ki factor correspondingly becomes smaller. It then seems as if the linear system is in control. Both systems can now be advantageously utilized.

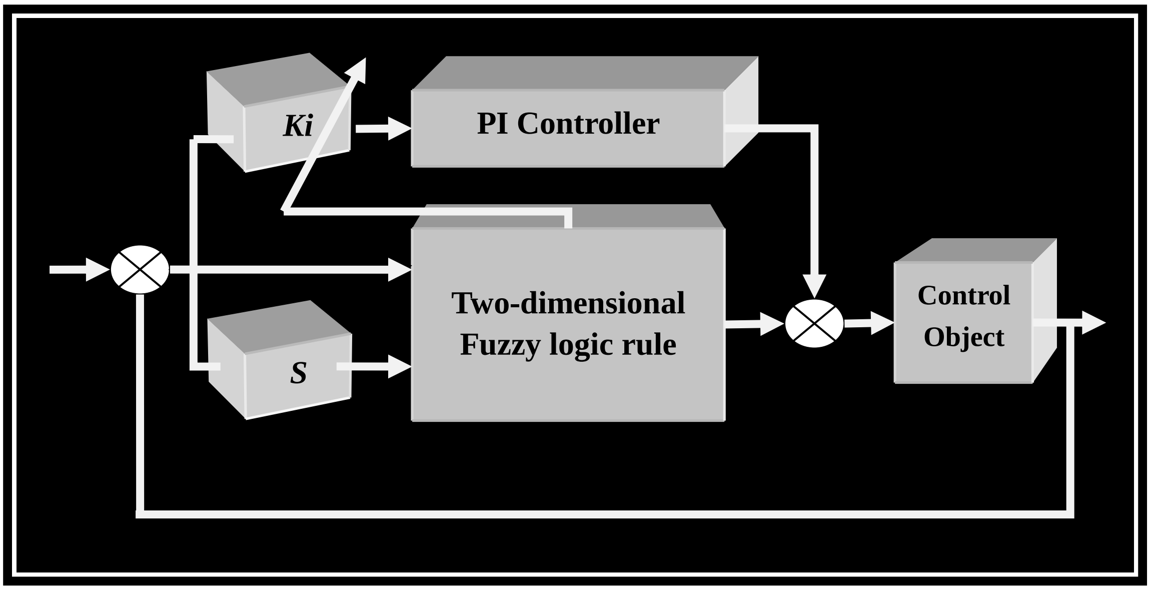


Figure 1.1: ***Hybrid self-tuning Fuzzy PID control architecture***

1.4.5. *Control Object*

The dynamitic matching character, during the refrigerant cycle that include evaporator, condenser, compressor and expansion valve, has direct effect on performance of a vehicle HVAC system. The control object involved with those components depends upon the existing hardware configuration on an automobile, and software control modelling has to be considered the realistic hardware. Normally refrigerant cycle system includes those factors such as compressor speed, electronic expansion valve voltage, evaporator fan speed and condenser fan speed. To achieve the optimization matching character among those four parts, it is necessary to study each structure and their operating characteristics.

The dynamics of the actuating component are considered to be fast relative to the dynamics of the heat exchangers. The actuating components are modelled with static relationships. The modelling of the other components is significantly more complex. Thus far, the teamwork project has done the simulation of compressor and expansion valve modelling. This paper sets out how to develop the core modelling parts of control object for heat exchanger in an HVAC vehicle system. A dynamic general model of condenser and evaporator has been built concerning the refrigerant R134a. It attempts to accurately describe the refrigerant working process; to consider of the numerical methods simulation model and possible operating conditions relative to another two parts of refrigerant cycle modelling. A simpler set of conserving ordinary differential equations is formulated by integrating separately over the three zones (super-cooled liquid, two-phase and super-heated vapour). All the ordinary differential equations have formed the heat exchanger dynamic mathematical model. Separated control volumes are considered for each of the fluid regions.

1.4.6. *Aim of Control Tactics*

There are a number of related projects to the proposed topic, either in how the projects include Fuzzy logic or PID controller simulation, or in the control strategy used to control the HVAC system.

Since the On/Off double – position control method occurred and was adopted in the 1960s. Automatic control of HVAC has been developed to the digital direct control; this control algorithm is mainly the PID control method, a self-adjusting control method based on the traditional control algorithm. However, with the HVAC system it is hard to establish the exact mathematical cooling circuit models. As a result of Fuzzy-neural network control, which based on experts knowledge and experienced operators, becomes a major research area in HVAC systems. Many research results indicate that fuzzy-neural network control in an effective method to solving control problems of HVAC systems.

In Multiple-Input Multiple-Output (MIMO) fuzzy control aspect, Tobi and Hanfusa create MIMO fuzzy controller by setting temperature and moistness of air conditioning system as input variables, and the controlling valve of heater, cooling coil, and moistener as output variables(Tobi T. & T. Hansfusa, 1991:333). Albert builds MIMO dynamic parameter identification/controller of air conditioning unit by using neural network (Albert T. P. etc, 1995:14), simulates and analyses the dynamic response characters and energy consumption in fuzzy control and PID control. In complex fuzzy control of air conditioning system aspect, Huang and Nelson research the fuzzy control problem of complex building air conditioning system (Huang S. and R.M. Nelson, 1994:844), which is composed by Variable Refrigerant Volume (VAV) air conditioning system, refrigerating system and room thermodynamic system. They put forward the method to build fuzzy rules model, which means fuzzy model should be symmetric with a centre point. Simulation and test prove this method of building model effective. Then, they put forward rules model self-adjusting fuzzy controller by using phase plane analysis (Huang S. and R.M. Nelson, 1999:206), and realized adjusting fuzzy control rules by using reference trace in phase plane. Simulation and test prove that this method has better control performances. In other proveniences, Qingwei Chen puts forward the laminar hierarchical fuzzy control method in workshop air conditioning system by combining laminar hierarchical control and fuzzy control (Qingwei Chen & Jianfang jiang & Weili Hu, 1994:14). In fuzzy model parameter identification aspect, Guoliang Ding builds the fuzzy model of compressor thermal character by combining fuzzy model and traditional mechanics model. Results of simulation indicate that mixed building model method has better precision and general performance than traditional model and simple fuzzy model (Guoliang Ding & Chunlu Zhang & Tao Zhan, 2000:1298). Jiejia Li introduces wavelet neural network into fuzzy control, which improves fuzzy control performances, considering sufficiently random disturbance and the impact on system in air conditioning control (Jiejia Li , 2004:37), when they solve fuzzy model parameter identification in VAV air conditioning system. In fuzzy rules online obtaining aspect, the author puts forward Mamdani rules double stages obtaining and self-organizing fuzzy control method and model reference rules self-organizing fuzzy control method of time delay thermal systems, and realize the fuzzy control course of air system and water system in radiator thermal performance test – board (Hui Liu, 2004).

We can know that fuzzy control has been used widely to automatic control of building HAVC. In design methods, fuzzy control has been developed from basic fuzzy control method to fuzzy PID, parameters self-optimizing, fuzzy neural network, fuzzy genetic algorithm, and wavelet fuzzy control. In order to accelerate the application and development of fuzzy control in refrigerating and air conditioning field, the author has researched systematically fuzzification method, fuzzy inference method, fuzzy control rules online obtaining and optimizing method, self-organizing fuzzy control method, fuzzy predictive method of time-lag process so many years.

1.4.7. Related Work and Success Story

It is generaylly impossible to fully test electronic ECUs with a large number of input values because of the enormous amount of the time required. Despite these constraints, Daimler$^{@}$ has achieved a high level of test coverage and good depth in tis software testing of electrical power steering units by using the methods of limit value and effects analysis to reduce the number of test cases. Its practical implementation involves simulating driving manoeuvres from the real world that are used as a reference.

That article (Available online at:
http://www.vector.com/portal/datei_mediendatenbank.php?system_id=474246&root=223 [5, May 2008]) describes how to reduce the set of all theoretically possible test cases and to implement tests with a development and testing tool. Practical testing would therefore requires that the sum of all possible input combinations be reduced to just a few representative test cases while achieving high test coverage. This can be achieved by combining and applying several test methods, such as the equivalency

class method, limit analysis and the pair-wise method. Daimler$^{@}$ uses the HIL test bench is effects analysis to find those signals that have the greatest impact on system behaviour. In addition to the three test methods mentioned, optimization by knowledge integration (*See Figure 1.2*) produces meaningful test cases and makes a valuable contribution toward increasing software quality and reduction process.

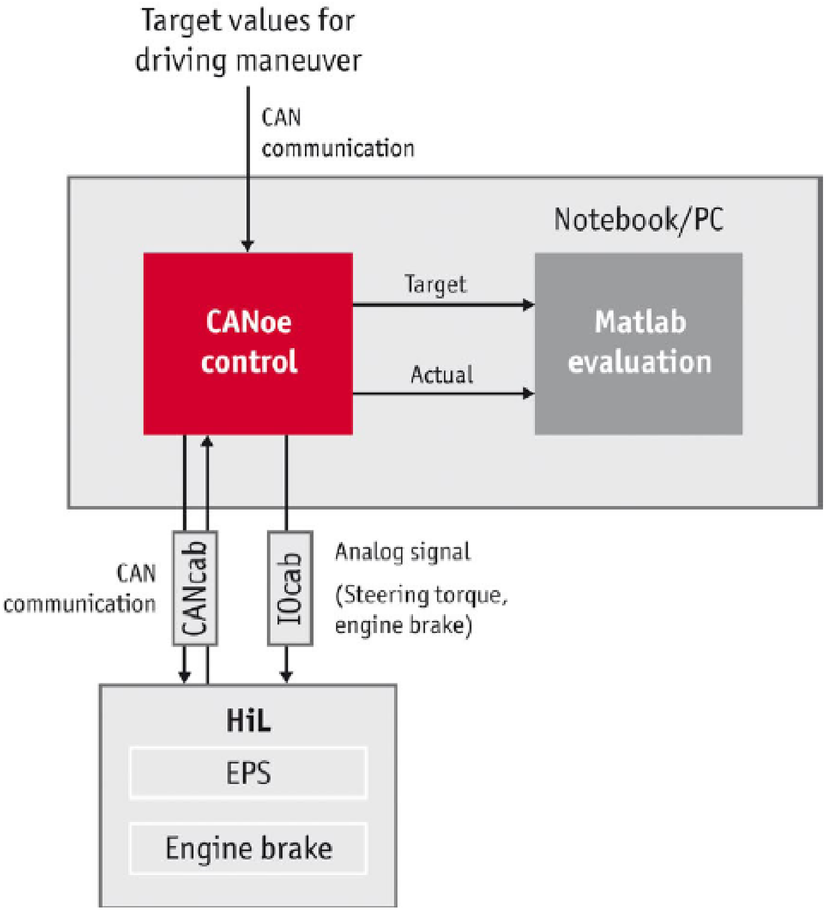


Figure 1.2: Test setup for logging real bus communication, driving the HIL and evaluating measured values [33]

In the Figure 1.2 MATLAB / Simulink have an overall functionality of building an integrated software environment, which serves as a rapid prototyping platform for designing, testing, and implementing a wide variety of control algorithms(Available online at: http://www.mathworks.com [1, May 2008]),which is the focus of this project.

Hardware testing and anglicizing environments on the CAN bus systems, CANoe is a universal development tools. It is made available to all project participants over the entire development process. The system producer is supported in functional distribution, functional checking and integration of the overall system. Additionally CANoe / MATLAB interface were also supported for design by Vector. This interface was designed to combine the strengths of both simulation tools, extend CANoe's node modelling capability by adding the strength of the MATLAB / Simulink environment.

(Available online at: http://www.vector-worldwide.com/vi_canoe_en.html [24, May 2008])

1.5. Delimitations

The following delimitations apply to this research:

- The simulation has to be implemented in MATLAB / Simulink
- The cooling circuit modelling is limited to vapour compression refrigerant cycles. No other refrigerant cycles are relevant for use in vehicles air-conditioning systems.
- The influence of the circulating lubricant in the refrigerant cycle is neglected. This delimitation is usually made for refrigerant cycle simulations and should mainly result in differences in suction density and enthalpy.
- According to the specification of researching vehicles which is focused on Volkswagen Polo, and in particular the self-control of expansion valve, there is nothing that can be controlled by our modelling. The radiator fan for condenser is still being controlled by an original control unit on the Volkswagen Polo.

1.6. Research Methodology

A Volkswagen Polo 2006 model will form the basis of the general simulation. Therefore the air conditioning system, to be constructed as an experimental setup will be copied and equipped with sensors and data acquisition hardware and software in order that a realistic representation of the system reflects the one built into the production vehicle. Further experimental tests will be performed on the vehicle itself to determine the various inherent vehicle parameters that are required to enable the overall vehicle simulation.

To establish of a performance simulation model for heat exchanger is an important task. It is based on the physical mechanism of the process, distributed parameters dynamic mathematical model of whole cooling circuit’s component as established.

It is important to understanding the function of the actuators and sensors in the testing vehicle, undertaking hardware and software DAQ for all of sensors. Therefore the input of controller modelling has to correspondingly allocate for each actuators supplied by MATLAB/SIMULINK, Thereafter active the interface of MATLAB/SIMULINK and CANoe to achieve actuators controlling, those system have to be running onto the real-time work. The simulation will be constructed to contain three major parts. The first will be the real-time value getting from CAN-Bus which included all of sensors, and classify each sensor value that will provide basis information regarding the expected controller, e.g. Temperature control have consider the input of temp error and change of temp error; the second will be controller modelling for various purpose or control actuators; the third one consist of the two main parts, one is cooling circuiting modelling which are considering the wobble plate input of compressor, others are the output from the controller which can dominant the actuators such as temperature flap, air flow flap, fresh air blower.

The literature review of the CANoe application will consider interfacing systems between MATLAB/SIMLINK software platforms with it components including CAN-Bus, and HIM design which was also included at the CANoe software design, this being the function of a panel editor. As sooner as the climatic control panel information has been obtained, it is a useful control for the MATLAB/SIMLINK modelling. The next vital step is to learn the MATLAB programming, which takes up the most of the research time. Although thermodynamics is a self-taught subject, one should collect all available and useful formulas that could explain the dynamic characteristics of the heat exchanger. After completion of the cooling circuit modelling, one should consider suitable controller system model. The last step in this process should include: testing and verification modelling check; information of findings, after which dissertation conclusions could be undertaken.

1.7. Proposed Dissertation Layout

The proposed layout of the dissertation is depicted in following:

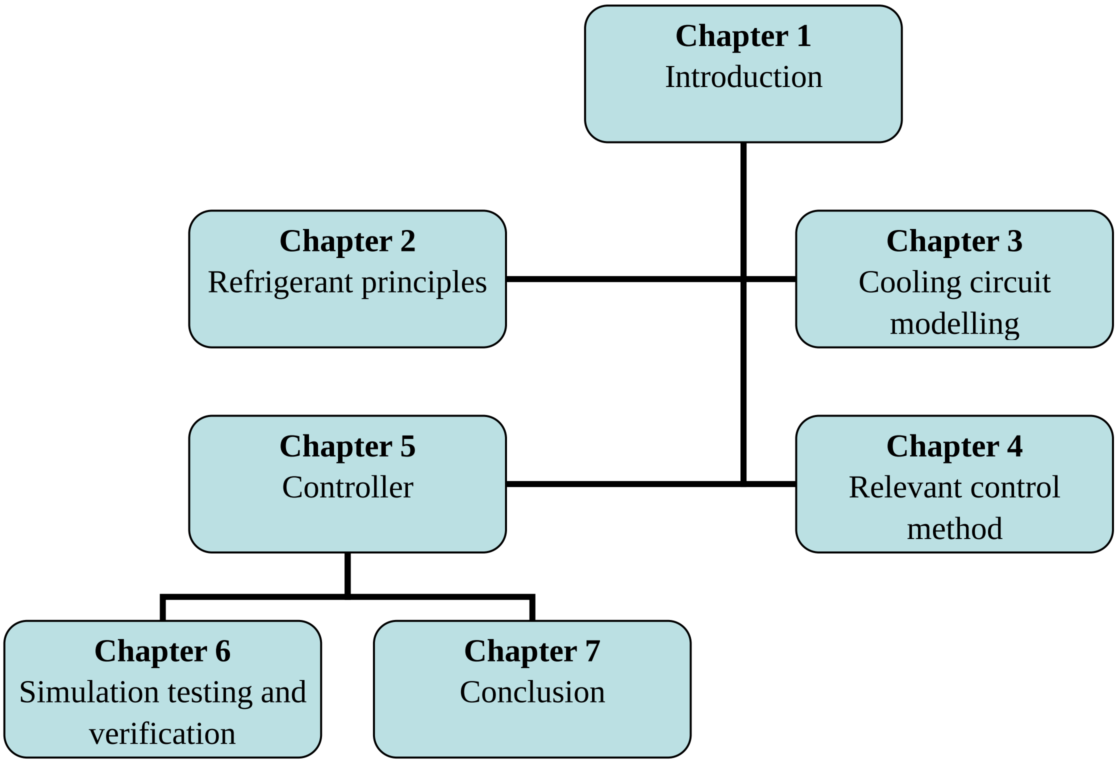


Figure 1.3: ***Proposed Dissertation Layout***

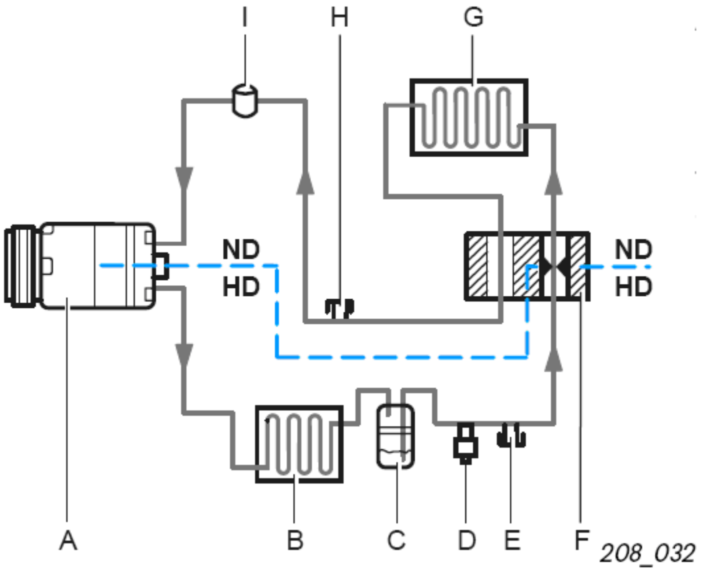
Chapter 1 provides some background into the problem area in order to describe the problem as the proposal. Chapter 2 contains some basic principles regarding air-conditioning to be discussed and some information will be provided concerning simulation of the refrigerant cycle in Chapter 3. All relevant control method will be discussed in Chapter 4 and setting the parameters of Fuzzy PID controller will be set out in Chapter 5. Chapter 6 will include discussion of the simulation performance and results. Chapter 7 concludes the dissertation with suggestions for further research or application.

-Chapter 2

Refrigerant Principles

2.1. Introduction

The schematic block diagram in *Figure 2.1* represents a simplified standard cycle as used in most air-conditioning applications; components may be more or less standardised. The physical properties of each component vary, depending on a vehicle’s requirements.



A	Compressor	G	Evaporator
B	Condenser (Gas Cooler)	H	Low-pressure service connection
C	Fluid container with drier	I	Damper
D	High-pressure switch	HD	High-pressure
E	High-pressure service connection	ND	Low-pressure
F	Expansion valve		

Figure 2.1: ***Automotive Cooling Cycle and most common components***

The fluid is compressed to a high pressure by the compressor and then enters the condenser. At this higher pressure, the fluid has a higher temperature than the ambient conditions, and, as a fan blows air across the condenser, heat is transferred through the air, and fluid condenses. The fluid exits the condenser as a sub-cooled liquid at a high pressure. The fluid then pass through an expansion device. At the exit of the expansion valve the fluid is generally two-phase and at a low pressure. The fluid then enters the evaporator. At this lower pressure the fluid has a lower temperature than ambient conditions and a fan blows air across the evaporator. Heat is transferred to the fluid and the fluid evaporates. The fluid exits the evaporator as a superheated vapour and enters the compressor. In accordance with this processing, those are the four standard components on which the cooling principle of an A/C systems rests. Therefore only these components will be discussed in the following section.

2.2. The Refrigerant

The thermodynamics of the cooling cycle can be analyzed, as show in *Figure 2.2*. a circulating refrigerant such as R134a enters the compressor as a vapour; this vapour is compressed at constant entropy and exits the compressor superheated; the superheated vapour travels through the condenser which first cools and removes the superheat and then condenses the vapour into a liquid by radiate additional heat at constant pressure and temperature; the liquid refrigerant goes through the expansion valve(also called throttle valve) where its pressure abruptly decreases, causing flash evaporation and auto-refrigeration.

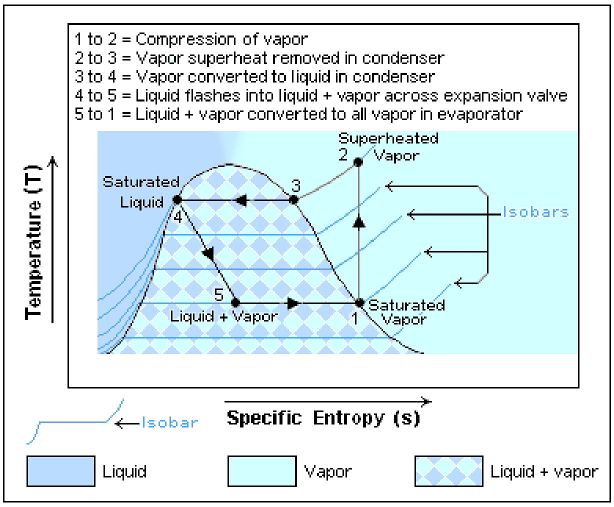

1 to 2 = Compression of vapor
2 to 3 = Vapor superheat removed in condenser
3 to 4 = Vapor converted to liquid in condenser
4 to 5 = Liquid flashes into liquid + vapor across expansion valve
5 to 1 = Liquid + vapor converted to all vapor in evaporator
Superheated Vapor
2
3
4
5
1
Saturated Liquid
Saturated Vapor
Liquid + Vapor
Isobars
Temperature (T)
Isobar
Specific Entropy (s)
Liquid
Vapor
Liquid + vapor

Figure 2.2: ***T-S Diagram of the vapour-compression refrigeration cycle***

The A/C systems implemented in the automotive industry are essentially refrigeration systems. In the current automotive A/C system only R134a (Appendix P.1) is used as a refrigerant. R134a (Terafluorethane) is a CFC (Chlorinated hydrocarbon) free refrigerant. The chemical properties are well suited for refrigeration purposes because of a low boiling point. Depending on the pressure and temperature condition of the refrigerant circuit, the refrigerant will either be a liquid; a gaseous state or two phase.

	R134a(Tetrafluorethane)	**Units**
Chemical Formula	CF_3CH_2F	[-]
Molar Mass	102	[kg/kmol]
Boiling Point (1.011bar)	246.85	[K]
Solidification Point	171.55	[K]
Critical Temperature	374.15	[K]
Critical Pressure	4.056	[MPa]
Colour	achromatic	[-]
Odour	Slight Ethereal	[-]

Table 2.1: ***Physical properties of R134a***

A number of thermodynamic cycles can be used to create low air temperatures. These include:

- Carnot cycle
- Lorenz cycle
- Ericson cycle
- Stirling cycle
- Joule cycle

The reversed Carnot cycle is the cycle most referred to in the automobile refrigeration industry. The cycle can however not be considered as a realistic model for refrigeration cycles. This is largely due to the fact that a liquid-vapour mixture is compressed. In practice the compressors are not able to compress a medium that consists of two phases. Thus the ideal vapour-compression refrigeration cycle is used as the ideal model with which to compare refrigeration cycles. It differs to the reversed Carnot cycle by replacing the turbine with an expansion valve or capillary tube, and the complete vaporization of the medium before compression.

2.3. Compressor is driven by the engine via a belt. An electromagnetic clutch on the compressor transmits the power from the pulley to the compressor crankshaft when the air conditioner is witched on. The compressor draws in refrigerant gas from the evaporator, compresses it and delivers it to the condenser.

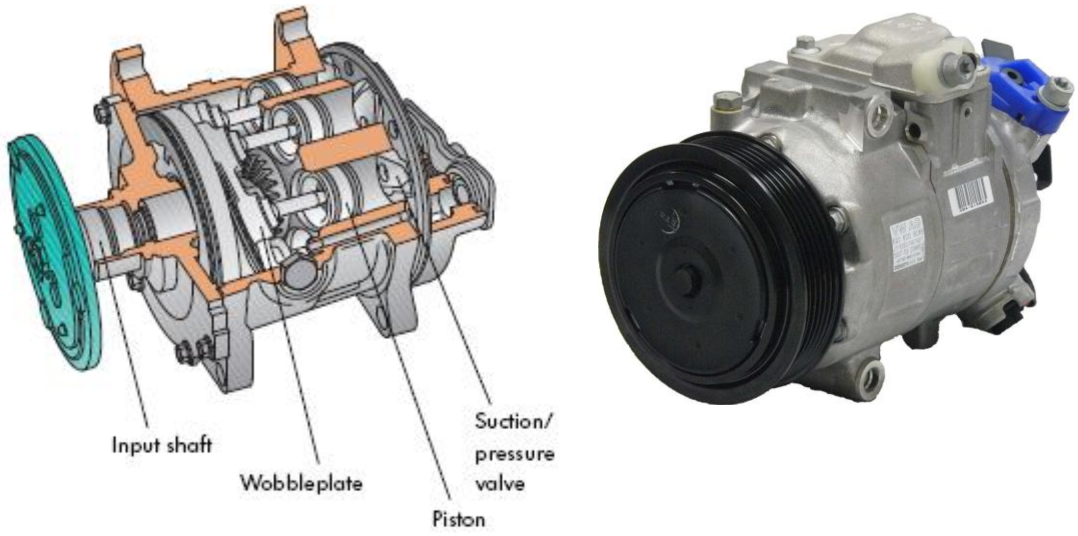


Figure 2.3: ***Construction of compressor [1] (L), Physical compressor @ VW Polo(R)***

In modern vehicle air-conditioning systems the compressors are oil-lubricated displacement compressors of the wobble-plate type. They require no magnetic clutch to switch the compressor on or off. The compressor is driven directly by the engine by means of a belt and is situated in the engine compartment fixed to the engine.

The function of the compressor is to increase the pressure of the refrigerant. With the rise in pressure, the temperature rises. The pressure rise is required in order to expand the refrigerant into the low-pressure side of the circuit and thus create low temperatures.

The wobble-plate is rotated by means of the input shaft. Fixed to the wobble-plate are 3 to 10 pistons. The inclination of the wobble-plate causes the pistons to move in a reciprocating manner. Each piston has a suction/pressure valve. The valves are automatically opened and closed during their respective suction and pressure strokes. The compressor operates between speeds varying between 0 and 6000 rpm. The compressor speed influences the flow rate and thus influences evaporator filling and cooling capacity. For this reason the wobble-plate (self regulating or externally regulated) compressors were developed. The inclination of the wobble-plate is adjusted by the pressure difference between the chamber and high-pressure side, that is the pressure difference between upper- and lower side of the piston. In addition the pressure difference can be regulated externally using a regulating valve. This enables the stroke to be varied and thus regulate the flow rate according to engine speed, ambient temperature or driver selected interior temperature.

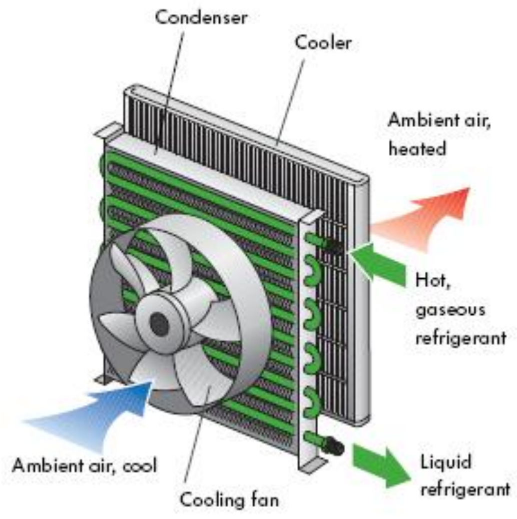
The oil used for lubrication is free of impurities such as sulphur, wax and moisture. The oil must be compatible with the refrigerant and cannot attack the seals and hoses used in the system. Oil properties be:

- High solubility in combination with the refrigerant
- Good lubrication properties
- Acid free
- Highly hygroscopic

Approx. half of the oil remains in the compressor while the other half is distributed throughout the system.

2.4. Condenser

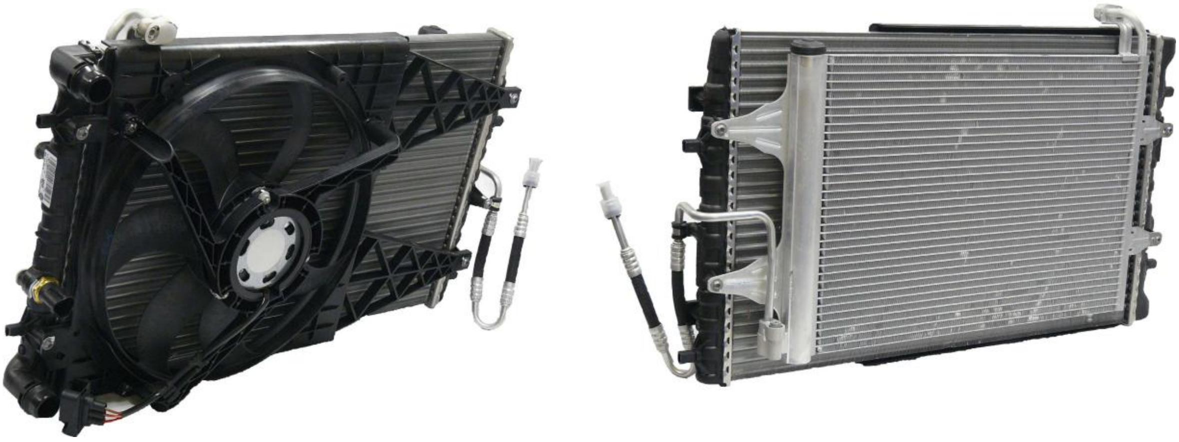
radiates heat from the compressed refrigerant gas to the surrounding air. In the process, refrigerant (R134a) gas condenses to a liquid.



In any A/C system the condenser is positioned in such a way that it will always have a good air flow across it to maximize the efficiency of the condenser in the refrigeration cycle. In the case of an automotive climatronic system, the condenser is located between the radiator and the cooling fan. This allows the headwind from driving to flow directly across the condenser, thereby ensuring a maximal flow of fresh air across it. Furthermore the fact that the cooling fan is located in front of the condenser and cooler, an auxiliary fan may still be used depending on type, it allows for the maintaining of the airflow across both the condenser and radiator during extended idle periods where the airflow from the head wind is reduced to almost zero.

The fan usually starts up when the air conditioner is switched on, in this case, fan switch-on will then be delayed until a specific pressure is reached. Impurities in the condenser reduce air flow and can also impair condenser capacity and engine cooling.

Hot gaseous refrigerant coming from the compressor at a temperature of approximately, 50 to-70 ℃ is injected into the compressor. The tubes and fins of the condenser absorb heat. Cool ambient air is ducted over the condenser, absorbing heat in the process and thereby cooling down the refrigerant. When the refrigerant cools down, it condenses at a specific temperature and pressure and becomes a liquid. At the bottom of the compressor, the refrigerant emerges from the condenser as a liquid.



{1} Back View **{2}** Front View

Figure 2.4: ***Physical Condenser, Radiator & Fan @ VW Polo***

Figure 2.4 shows, VW Polo’s fan. This is usually situated behind the engine radiator and next to the condenser, It specification and the dimension of the condenser will be mentioned in the next chapter.

2.5. Expansion valve atomises the refrigerant flowing in and controls the flow rate so that depending on the heat transport, the vapour first becomes gaseous at the evaporator outlet.

In an automotive climatronic system, the expansion valve is located inside the so-called Heating, Cooling and Air-conditioning box in close proximity to the evaporator. The function of the expansion valve is to regulate the flow of refrigerant into the evaporator. It forms the interface between the high-pressure side and low-pressure side of the refrigerant circuit. Thermostatic expansion valves have become common practice in automotive A/C systems. These types of expansion valves are generally controlled by the refrigerant gas temperature at the outlet of the evaporator and are very efficient at regulating refrigerant flow into the evaporator .*Figure 2.5* shows the construction of a thermostatic expansion valve, with real component used in VW Polo.

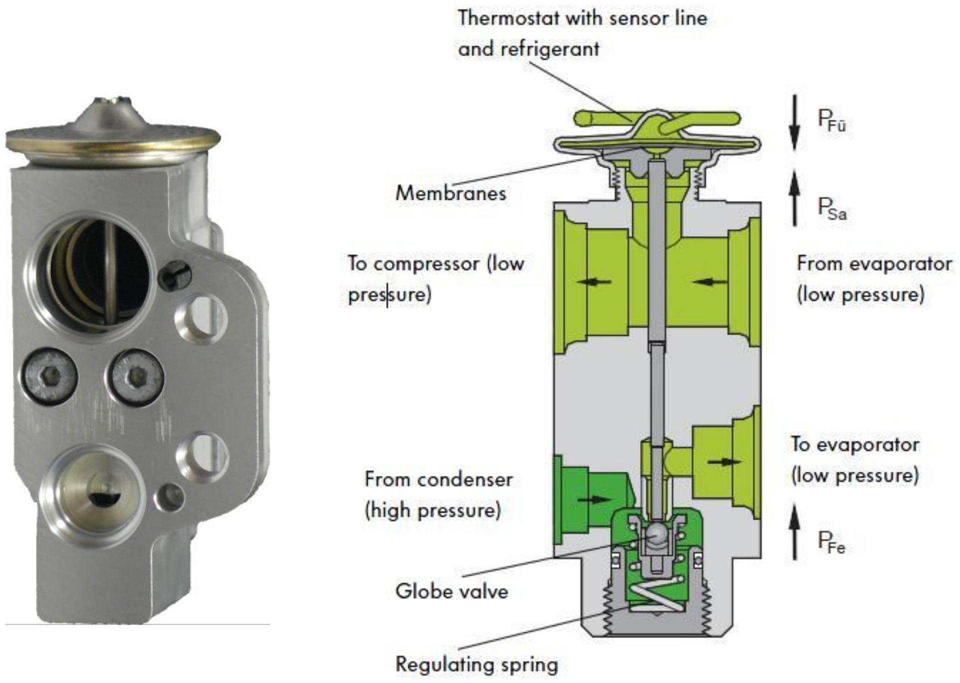

Thermostat with sensor line and refrigerant
$P_{Fü}$
Membranes
P_{Sa}
To compressor (low pressure)
From evaporator (low pressure)
To evaporator (low pressure)
From condenser (high pressure)
P_{Fe}
Globe valve
Regulating spring

Figure 2.5: ***Physical expansion valve @ VW Polo (L), Construction of expansion valve [1] (R)***

The expansion valve regulates the refrigerant flow into the evaporator. It is controlled by the exit temperature of the vapour at evaporator exit. The globe valve is controlled by the push rod that is attached to a membrane. The membrane is filled on one side with a special gas. The other side is supplied with refrigerant exiting the evaporator via pressure-equalizing holes. Depending on the temperature of the refrigerant, the membrane move up or down. If the globe valve opens, refrigerant is passed through and injected into the evaporator.

2.6. Evaporator

the liquid refrigerant vaporises in the coils of the evaporator. The heat required for this purpose is taken from air streaming along the fins of the evaporator, so that the air is cooled in this manner. The refrigerant vaporises and is drawn in together with the absorbed heat by the compressor.

The boiling point of the refrigerant is well below zero degrees Celsius and therefore the liquid refrigerant from the expansion valve cools the evaporator rapidly upon entering. The construction of the evaporator dictates effective heat transfer between the refrigerant and the ambient air flowing across the evaporator. This allows the refrigerant to extract heat from the air effectively in order to evaporate, thereby effectively cooling the air flow that passes through the evaporator fins. Condensation occurs on the outside of the evaporator because of its low surface temperature, causing the air not only to be cooled but also dehumidified. On the following page *Figure D.1* shows the refrigerant and air flow directions through and across the evaporator in an automotive application.

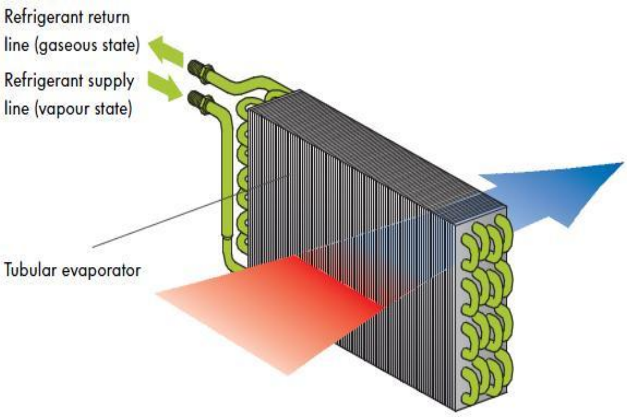
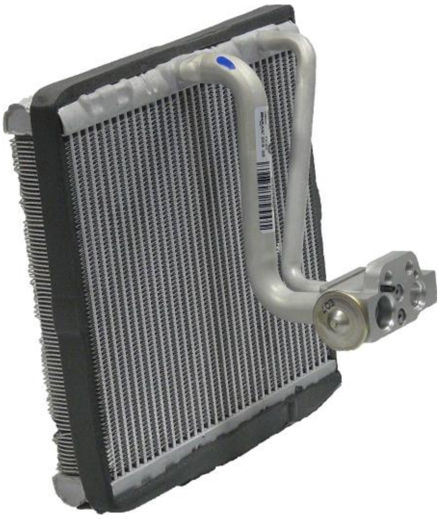

Refrigerant return line (gaseous state)
Refrigerant supply line (vapour state)
Tubular evaporator

Figure 2.6: ***Physical evaporator @ VW Polo***



It is necessary to regulate the flow of refrigerant through the evaporator with the expansion valve in order to prevent the surface temperature from becoming too low. Should the temperature of the evaporator become low enough for the condensed vapour from the ambient air to freeze on the surface, it will dramatically affect the heat transfer capacity of the evaporator, thereby making the system inefficient. The refrigerant exits the evaporator in a gaseous form to be returned to the suction side of the compressor where the cycle is repeated.

2.7. HVAC Box

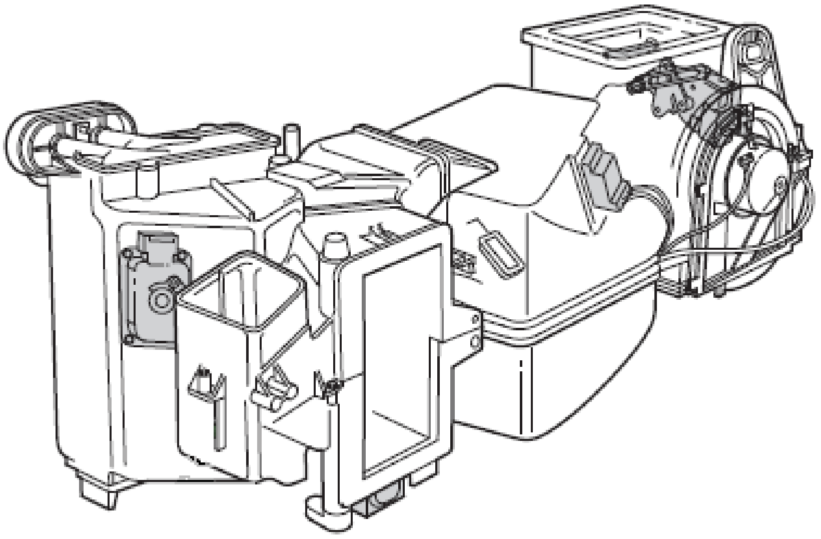


Figure 2.7: ***HVAC box*** **[1]**

The final component that allows for satisfactory operation of the system within the constraints of a vehicle is the HVAC box. In an automotive air conditioning system, both the expansion valve and the evaporator are located inside the so-called Heating, Cooling and Air-conditioning box. The HVAC box forms an integral part of the overall system and is located behind the dashboard of the vehicle. It not only houses the evaporator and allows for effective air flow across it, but also incorporates components like the fan, a heat exchanger for heating the cabin and the pollen filter. The construction is a complex series of housings for components. It also includes channels to form the air ducting and to enable effective airflow regulation and distribution throughout the vehicle cabin. In *Figure 2.7* a common construction of an HVAC box can be seen. In the next chapter, a description of all the components inside the HVAC box by mathematical analysis will be outlined.

-Chapter 3

Cooling circuit modelling

Simulation is the imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviours of a selected physical or abstract system. Computer simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behaviours of the system.
(Available: http://en.wikipedia.org/wiki/Simulation [24, May 2008])

3.1. Refrigerant *R134a*

In the cooling circuit of an HVAC system, one should consider what are the core issues on its behaviours? It seems suitable to think of this system as a corporal body. Then obviously, “refrigerant” would serve as “blood” which is circulated around the body through pipes that act as “blood vessels” which by their pumping action supply the heart, and thus has the same function as a compressor. High pressure and Low pressure refrigerant could be assumed as “arterial blood” and “venous blood”. In this case, analyzing and modelling refrigerant would start as the first step for the whole cooling circuit system.

3.1.1. General equation for R134a

As mentioned before in chapter 2, currently in the most popular automotive A/C system R134a is used as a refrigerant. The following general equations and constants are valid:

$$\Theta = 1 - \frac{T}{T_C}, \quad \tau = \frac{T_C}{T}, \quad \delta = \frac{\rho}{\rho_C}, \quad R = 81.488856\frac{J}{kg \cdot K}, \quad T_c = 374.18K, \ \rho_C = 508\frac{kg}{m^3}$$

3.1.2. *Equation for the vapour*

The valid limits for the scope are: $195K \le T \le 455K$ and $p \le p_S(T) < 3.5MPa$.

❖ Pressure $p(\tau,\delta)$

$$\frac{p(\tau,\delta)}{\rho RT} = 1 + \delta(b_1\tau^{1/2} + b_2\tau^{7/4} + b_3\tau^4 + b_4\tau^{12})$$
$$+ 2\delta^2(c_1\tau^{1/4} + c_2\tau^{5/2})$$
$$+ 3\delta^3(d_1\tau^{7/2} + d_2\tau^{12} + d_3\tau^{20})$$

$(Eq.3.1)$

❖ Enthalpy $h(\tau,\delta)$

$$\frac{h(\tau,\delta)}{RT} = a_1\tau + m_1 + \frac{2}{3}m_2\tau^{-1/2} + \frac{4}{7}m_3\tau^{-3/4}$$
$$+ \delta(\frac{1}{2}b_1\tau^{-1/2} + \frac{11}{4}b_2\tau^{7/4} + 5b_3\tau^4 + 13b_4\tau^{12})$$
$$+ \delta^2(\frac{1}{2}c_1\tau^{-1/4} + \frac{9}{2}c_2\tau^{5/2})$$
$$+ \delta^3(\frac{13}{2}d_1\tau^{7/2} + 15d_2\tau^{12} + 23d_3\tau^{20})$$

$(Eq.3.2)$

❖ Entropy $s(\tau,\delta)$

$$\frac{s(\tau,\delta)}{R} = a_0{}^* + (m_1 - 1)(1 - In\,\tau) - In\,\delta + 2m_2\tau^{-1/2} + \frac{4}{3}m_3\tau^{-3/4}$$
$$+ \delta(-\frac{3}{2}b_1\tau^{-1/2} + \frac{3}{4}b_2\tau^{7/4} + 3b_3\tau^4 + 11b_4\tau^{12})$$
$$+ \delta^2(-\frac{5}{4}c_1\tau^{-1/4} + \frac{3}{2}c_2\tau^{5/2})$$
$$+ \delta^3(\frac{5}{2}d_1\tau^{7/2} + 11d_2\tau^{12} + 19d_3\tau^{20})$$

$(Eq.3.3)$

❖ Spec. heat capacity $c_P{}^0(\tau)$

$$\frac{c_P{}^0(\tau)}{R} = m_1 + m_2\tau^{1/2} + m_3\tau^{3/4} \qquad (Eq.3.4)$$

With the coefficients:

$b_1 = 0.22012860$ $\quad b_2 = -1.26740200$ $\quad b_3 = -0.30849910$

$b_4 = -3.546135 \cdot 10^4$ $\quad a_1^* = 9.047135$ $\quad a_0^* = 1.019535$

$m_1 = -0.629789$ $\quad m_2 = 7.292937$ $\quad m_3 = 5.154411$

$d_1 = -0.06356633$ $\quad d_2 = 0.08120353$ $\quad d_3 = -0.03699965$

$c_1 = 0.01387313$ $\quad c_2 = 0.33531550$

3.1.3. *Equation for saturated vapour*

❖ Vapour pressure $p_s(T)$

Valid for $169.85K \leq T \leq 374.18K$ with $p_0 = 4.056318MPa$

$$In\frac{p_s(\Theta)}{p_0} = \frac{1}{1-\Theta}[a_1\Theta + a_2\Theta^{3/2} + a_3\Theta^2 + a_4\Theta^4] \qquad (Eq.3.5)$$

With the coefficients:

$a_1 = -7.7057291$ $\quad a_2 = 2.4186313$ $\quad a_3 = -2.1848312$

$a_4 = -3.4530733$

❖ Density of saturated vapour pressure $p'(\Theta)$ (liquid)

Valid for $185K \leq T \leq 373K$

$$\frac{p'(\Theta)}{kg/m^3} = q_0 + q_1\Theta^{1/3} + q_2\Theta^{2/3} + q_3\Theta^{13/4} \qquad (Eq.3.6)$$

With the coefficients:

$q_0 = 518.236$ $\quad q_1 = 885.538$ $\quad q_2 = 482.517$ $\quad q_3 = 192.157$

- The saturated liquid density $p''(\Theta)$

  Can be determined with an iterative procedure where $p = p(\tau, \delta)$

- Saturated vapour enthalpy $h'(\Theta)$

  Valid for $185K \leq T \leq 373K$ with $h_0 = 384.07kJ/kg$

$$In \frac{h'(\Theta)}{h_0} = n_1\Theta^{1/2} + n_2\Theta^{5/4} + n_3\Theta^3 + n_4\Theta^4 + n_5\Theta^{10} \qquad (Eq.3.7)$$

With the coefficients:

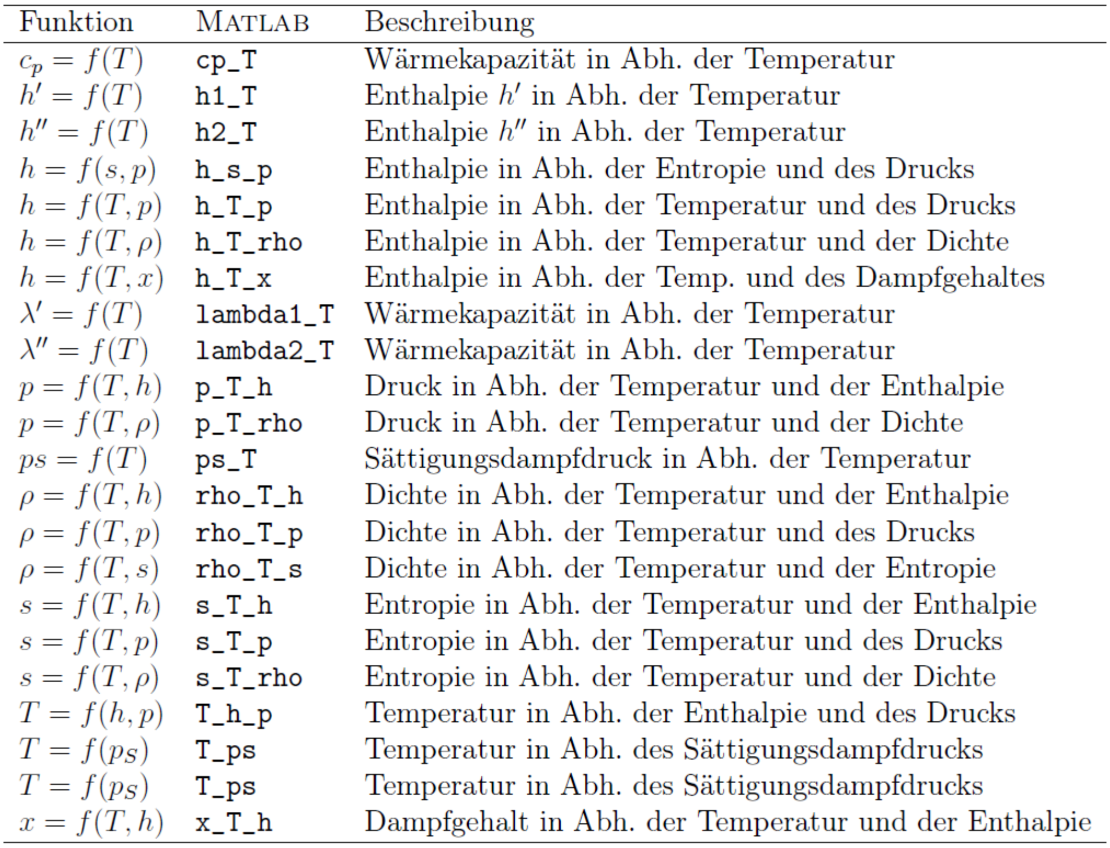
$$n_1 = -0.534728 \quad n_2 = -1.757777 \quad n_3 = -0.928326$$

$$n_4 = -2.666564 \quad n_5 = -30.82570$$

3.1.4. *Numerical methods*

Instant determination is only possible for the following properties, with the given independent property combinations:$p = p(T,\rho)$; $h = h(T,\rho)$; $s = s(T,\rho)$; $ps_s = p_s(T)$; $\rho' = \rho'(T)$; $h' = h'(T)$; The solution for any other combination of dependent and independent properties can be achieved using a numerical method. In a case with limited scope, the dichotomizing or binary search is useful. With this method the equation is equated to zero and the starting points for the search have to be assumed with one point being a zero point. The determination of these starting points can be difficult in that there may be more than one zero point in the range of the interval of the starting points, by definition of the dichotomizing algorithm. Hence, these starting points were found by examining the results of the calculations within the stated limits for the equation. Using these two starting points the marked interval can be reduced as long as the remaining error is greater than the error limit.

The programming and solution for those equations are implementing onto MATLAB-Files (m-files). Mr. Tobias Eitelbubs who is my German teammate wrote the m-files, in Table 3.1, all functions used in MATLAB / Simulink are programmed. The source code to the functions is included in the Appendix-.



Funktion	MATLAB	Beschreibung
$c_p = f(T)$	cp_T	Wärmekapazität in Abh. der Temperatur
$h' = f(T)$	h1_T	Enthalpie h' in Abh. der Temperatur
$h'' = f(T)$	h2_T	Enthalpie h'' in Abh. der Temperatur
$h = f(s, p)$	h_s_p	Enthalpie in Abh. der Entropie und des Drucks
$h = f(T, p)$	h_T_p	Enthalpie in Abh. der Temperatur und des Drucks
$h = f(T, \rho)$	h_T_rho	Enthalpie in Abh. der Temperatur und der Dichte
$h = f(T, x)$	h_T_x	Enthalpie in Abh. der Temp. und des Dampfgehaltes
$\lambda' = f(T)$	lambda1_T	Wärmekapazität in Abh. der Temperatur
$\lambda'' = f(T)$	lambda2_T	Wärmekapazität in Abh. der Temperatur
$p = f(T, h)$	p_T_h	Druck in Abh. der Temperatur und der Enthalpie
$p = f(T, \rho)$	p_T_rho	Druck in Abh. der Temperatur und der Dichte
$ps = f(T)$	ps_T	Sättigungsdampfdruck in Abh. der Temperatur
$\rho = f(T, h)$	rho_T_h	Dichte in Abh. der Temperatur und der Enthalpie
$\rho = f(T, p)$	rho_T_p	Dichte in Abh. der Temperatur und des Drucks
$\rho = f(T, s)$	rho_T_s	Dichte in Abh. der Temperatur und der Entropie
$s = f(T, h)$	s_T_h	Entropie in Abh. der Temperatur und der Enthalpie
$s = f(T, p)$	s_T_p	Entropie in Abh. der Temperatur und des Drucks
$s = f(T, \rho)$	s_T_rho	Entropie in Abh. der Temperatur und der Dichte
$T = f(h, p)$	T_h_p	Temperatur in Abh. der Enthalpie und des Drucks
$T = f(ps)$	T_ps	Temperatur in Abh. des Sättigungsdampfdrucks
$T = f(ps)$	T_ps	Temperatur in Abh. des Sättigungsdampfdrucks
$x = f(T, h)$	x_T_h	Dampfgehalt in Abh. der Temperatur und der Enthalpie

Table 3.1: ***List of thermal-dynamic properties in MATLAB***

All the algorithms use the call "Bisection method". h_T_p.m and h_s_p.m are exemplarily for explaining this function.

❖ h_T_p.m: This function determines the enthalpy in J/kg depending on the entropy in J/ (kgK) and the pressure in Pa for the refrigerant R134a, following step is required.

1. Determines the density in Kg/m^3 depending on the temperature in K and the pressure in Pa for the refrigerant R134a using the bisection method, and Equation 3.1 converted into zero.
2. The result of Equation 3.1 for density can compute the temperature which is in Equation 3.2, and can also work out the enthalpy "h" in the overheat area.

With a maximum error $\varsigma = 10^{-5}$; normally exceed 20 iterations are needed.

- h_s_p.m: This function determines the enthalpy in J/kg depending on the entropy in J/ (kgK) and the pressure in Pa for the refrigerant R134a using the bisection method, the following step is demanded:
  1. Define two limits for the maximum and the minimum temperature. One should the minimum temperature T1 by inserting the given pressure in equation 3.5. The maximum temperature is defined with T2 = 393K.
  2. With these two temperatures and pressures you should use equation 3.1 to obtain density 1 and density 2 (this has been outlined above).
  3. The two density values, together with the belonging temperatures, have to be inserted in equation 4.3 and deliver the entropy s1 and entropy s2.
  4. Because the two interval boarders (s1, s2) and the exact given value s are known, the temperature can be calculated by using a numerical method. This numerical methods works similar to the bisection method. Hereby, the temperatures T1 and T2 are moved more and more together until the equation s(T;p) 4 minus the given value s is smaller than the defined left failure. enthalpy h with the equation h_T_p.m is obtained.

With a left failure = 10^2, an average of 300 iterations is necessary because the density is also calculated with the bisection method. As this is an immense numerical work, this equation will be replaced with the 2D Lookup table (*Figure 3.1*). The values for the table are calculated with the help of the equations and, thereafter, saved. The advantage of a Lookup table is the relative easy calculation of the wanted values. While with the above mentioned equations iterative methods need to be used, the Lookup table interpolates between the neighbour values. Therefore, instead of using the equations T_h_p.m and h_s_p.m, the Lookup table is used instead. This shortens the simulation time immensely. Because the simulation takes place in MATLAB/Simulink, the m-files need to be included with so called Embedded Matlab Functions (*Figure 3.2*). Embedded Matlab Functions are used with Simulink when the algorithm can be described more easily with the help of code and graphical programming. An additional advantage of these functions is that automatically a fast C code will be generated and compiled during the first start. This is made by Embedded Matlab 5. S-functions can also be generated.

In table 3.1, all functions are listed which can be programmed in Matlab/Simulink. The source code for the equations can be found in the affix. Embedded Matlab Functions have no access to m-files. For example, if the equation rho_T_p is needed within the equation s_T_p, it has to be included as a sub-function directly into the Matlab Embedded Function. This is where it differs from the m-files.

Once all of the considerable numerical results have been obtained, MATLAB / Simulink provide an expediently function by a 2D lookup table as show in (*Figure 3.1.1*) This Lookup Table (2-D) block computes an approximation to some function z=f(x,y) given x,y,z data points. Each vector for the table are identified and saved. The major advantage of a lookup table is that the relatively simple calculation for the sought values could be done by the MATLAB computation. On the other hand the functions described above iterative methods must be used; an interpolated lookup table only between the adjacent values in the simulation, therefore the function T_h_p.m and h_s_p.m are used instead of by the lookup tables function. The simulation time is greatly reducing comparison with those two, once the simulation of the model in MATLAB/Simulink has been done; the m-files on the so-called Embedded MATLAB Functions (*Figure 3.1.2*) are embedded.

Lookup
Table (2-D)
u
fcn
y
Embedded
MATLAB Function

Figure3.1: ***Lookup Table Block and Embedded Matlab Function***

Embedded MATLAB Functions are always used in Simulink. Where the algorithm works it is best to refer to the helping system, the code as defined by graphic programming can be described. Another advantage of these functions is that when they start off the model the first time, it automatically generates fast c code and compiled. This is done using Embedded MATLAB, which is called as s-function generation.

3.1.5. **log** *p,h-diagram*

This vapour pressure curve shows the variations of properties during phase-change processes. Intuitively a “p,h-diagram” is the best way to study and understand the property of refrigerant, an excerpt from the different state of refrigerant R134a for a similar vehicle air-conditioner.

At *Figure 3.2* the property diagram is also extremely useful in analyzing the thermodynamic processes, heat transferred and efficiencies, for the cycle. With this diagram, the heat transferred is simply proportional to the enthalpy change across the corresponding process. Alternatively, a T,s-diagram as seen in *Figure 2.2* in section two may be used for this analysis, but the heat transferred is more difficult to determine, in that the heat transferred is equal to the area under the corresponding process. Different absolute values are possible in dependence upon the demand of a vehicle type for refrigeration capacity.

With the described property equations it is possible to create a log p,h-diagram and the generation of the saturation lines, and is-lines with MATLAB script-files.

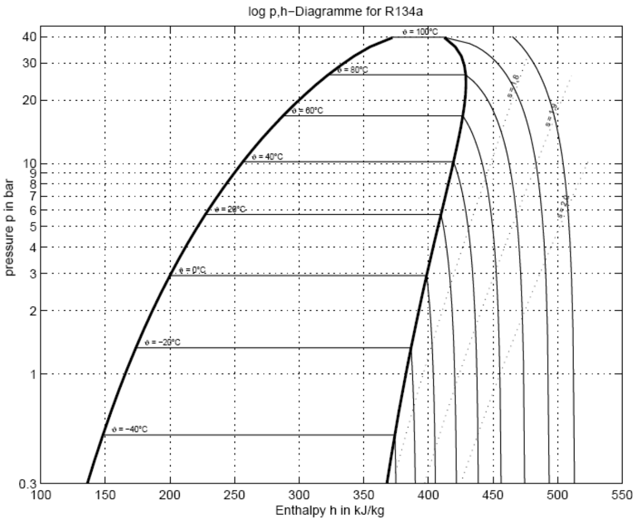


Figure 3.2: log *p,h-diagram for R134a*

With the 3.1.2 section show in numerical methods, *Figure 3.3* shows the diagram, if there are enough values in the chart calculation, under the help of MATLAB commands "meshgrid" and "pht griddata" a three-dimensional design chart. The function "griddata" interpolate adjacent the calculated values. The various interpolations in MATLAB can help with translation. Since the m-files as many values can be calculated, simply as a linear interpolation. The commands on these derived coordinates (x, y, z) can be used to create a 2-D lookup table see (*Figure. 3.1*) in Simulink to generate. This requires only that coordinates values in the lookup table block be entered.

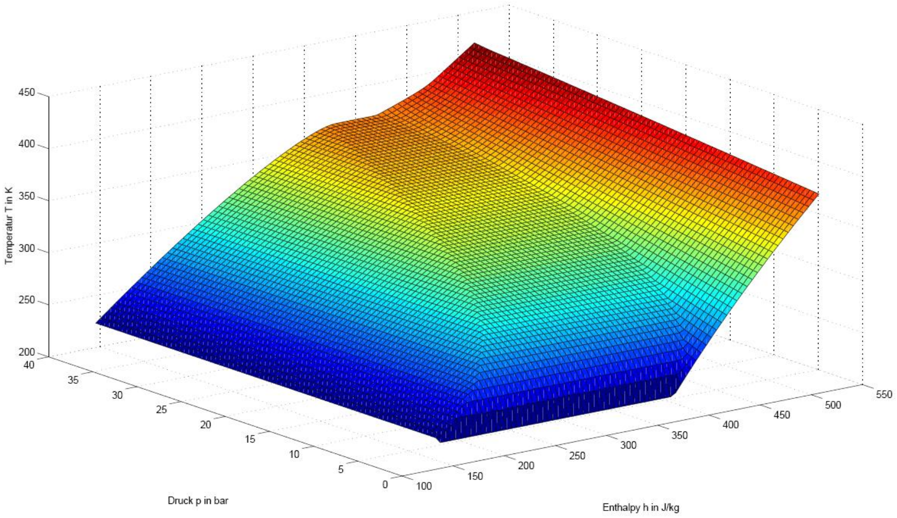


Figure 3.3: ***the three-dimensional diagram for p_h_t***

3.2. Modelling overview

The simplest cycle of the HVAC-R134a system as mentioned above, operates with four components see (*Figure 3.4*): compressor, condenser, expansion valve, and evaporator. In the cooling circuit, various factors could influence cooling performance, and one of the components effect with the other. In order to achieve this modelling system, it is necessary to extract the matching parameters, representative of conforms to the cooling circuit in the system. This parameter is separated by two parts that are self-impact and mutual-impact parameters. If the parameter only affects one manner of component and does not affect others, it is called a self-impact parameter; mutual-impact parameter means one aspect would affect the whole system.

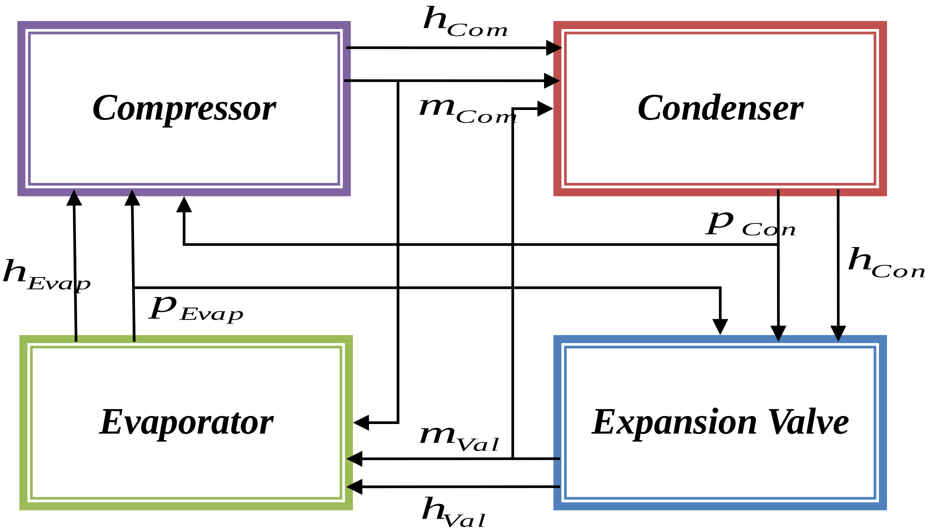


Figure 3.4: ***Framework structure of modelling***

Mutual-impact parameters link each pact achieving cooling circuit. The value involved in this numerical method must be found in the condition or at the end of modelling process, which means that iterative algorithm is demanded. In the use of this modelling, mutual-impact parameter has been included: “mass-flow rate, enthalpy and pressure", will reveal the reason for this choice of these three parameters in each part of modelling.

Figure 3.4 shows in the subcritical system how the fluid enters the compressor as a superheated vapour at a low pressure. m_{com} and h_{com} are the output from the compressor, which respectively represent the mass-flow and the enthalpy of compressor. This fluid output enters the condenser. As a fan blows air across the condenser, heat is transferred in the air, and the fluid condenses. h_{com} represents the enthalpy output for condenser. The fluid exits the condenser as a sub-cooled liquid at a high pressure, and passes through an expansion valve. Output values m_{val} and h_{val} used for the input condition of evaporator. At this lower pressure the fluid has a lower temperature than ambient conditions, and as a fan blows air across the evaporator, heat is transferred to the fluid, and fluid evaporates. h_{evap} and p_{evap} would be used as an initial condition for the compressor, Closed- loop modelling is then achieved.

In the subcritical system, each Input / Output mutual parameter is difficult to determine during the closed-loop computation. There is only one way to calculate those parameters which are required, and that is by using the breakpoint and feedback to evaluate the initial value, at the beginning of the calculation. So, it is therefore better to start with the compressor model as the breakpoint.

3.3. Compressor

The modelling of the compressor is divided into several parts. Firstly, the compressor power and mass flow, as a function of compressor speed, can be simulated and on the other hand, the static behaviour in the coolant loop calculated by using the equations from the previous (Chapter 3.1) can be calculated.

3.3.1. *Compressor power and mass-flow*

The power of the compressor, depending on the speed which is engine driven is normally implement as a lookup tables. The values of the lookup tables derived from the specifications of the compressor from VW and are subject to confidentiality. All data are expressed as a function of the speed n in min-1. The specifications are tables for the section pressure $p_1$for the output pressure p_2, the cooling power $\dot{Q}_0$as well as for the compressor power P_k . A problem of the tables is the low number of values. Partially, only three values for a chart before. All other points must be interpolated. This is the MATLAB command interp1 used. For p_2, $P_k$and $\dot{Q}_0$, and p_1 for the Spline-interpolation are used. This interpolation is only an approximation of real behaviour. With more interpolated values could be better and more accurate results obtained. *Figure 3.5* shows the compressor power as a function of speed; *Figure 3.6* reveals the dependence of the cooling speeds.

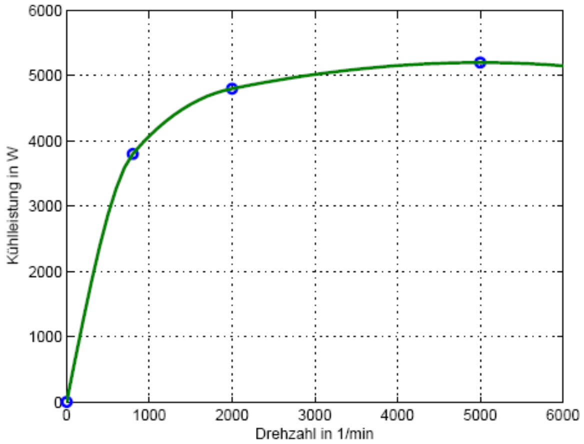


Figure 3.5: *Cooling power $\dot{Q}_0$*

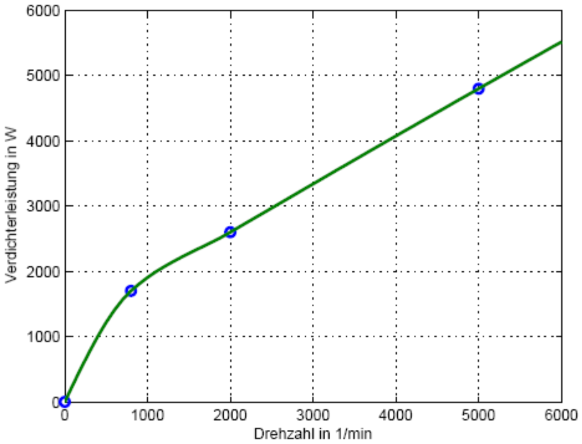


Figure 3.6: *Compressor performance P_k*

The compressor performance P_k is sponsored by the compressor refrigerant mass flow m_k and increase of the specific enthalpy-difference $h_2 - h_1$ determined. $h_2$is the enthalpy of the gaseous refrigerant in the condenser, h_1 the enthalpy of the evaporated refrigerant at the evaporator outlet.

$$P_k = \dot{m} \cdot (h_2 - h_1) \quad (Eq.3.8) \qquad \dot{Q}_0 = \dot{m}_k \cdot (h_1 - h_3) \quad (Eq.3.9)$$

The definition of cooling capacity specified in the datasheets. The cooling power $\dot{Q}_0$ is supported by the compressor refrigerant mass flow m_k and increase of the specific enthalpy-difference $h_1 - h_3$ determined in evaporator. h_3 is the enthalpy of the liquid refrigerant before the expansion valve.

$$\dot{m}_k = \frac{P_k}{h_2 - h_1} = \frac{P_k}{(h_{2is} - h_1)} \cdot \eta_{is} \qquad (Eq.3.10)$$

In this equation, the compressor performance is applied to calculate the mass flow of compressor. A further addition equation includes another factor, which is isentropic efficiency $\eta_{is} = f(w)$, since the effect of this factor is very small and the dependence can only be estimated, this extension will be waived, instead of isentropic efficiency with $\eta_{is} = 0.8$ as a constant.

$$\dot{m}_k = \rho_1 \cdot \dot{V}_k = \rho_1 \cdot V_k \cdot n \cdot v_k \qquad (Eq.3.11)$$

In equation.3.11 is the mass flow from the flow of the compressor is calculated. The flow rate $\dot{V}_k$ depends on delivery level of compressor v_k. This supply level is turning dependent on the ratio is given by the next equation where the spatial relationship with harmful $\ni_s = 0.04$ is approved and the Isentropic exponent *Kappa* (k) depending on the temperature is calculated.

$$v_k = 1 - \ni_s \left(\left(\frac{p_2}{p_1}\right)^{\frac{1}{k}} - 1\right) \quad (Eq.3.12) \qquad k = \frac{c_p(T)}{c_v(T)} = \frac{c_p(T)}{c_p(T) - R_i}$$

Since the dynamics of the compressor modelling is still not known, there is a simple link PT1 approximated. The time constant of the compressor is much smaller than evaporation, due to changing ratio difference. Therefore a dynamics of the overall model is needed in evaporation. The time constant of compressor Tk =2 is defined.

3.3.2. *Suction-pressure and outlet pressure*

As mentioned above, another mutual-impact parameter is the pressure. The specifications are also measured for the suction-pressure and outlet-pressure in correspondence to the speed and the gradients are drawn in *Figure 3.7* and *Figure 3.8*

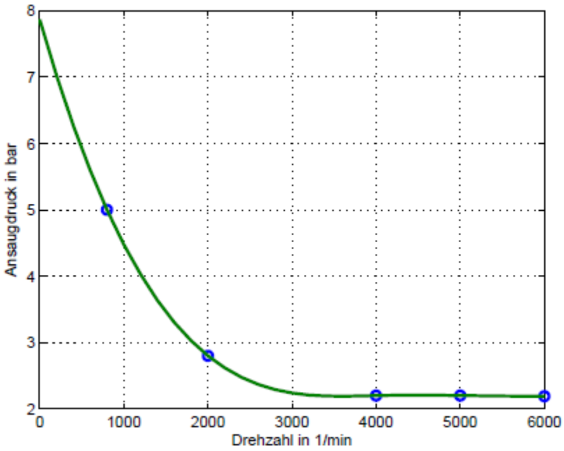


Figure 3.7: ***Suction-pressure*** P_1

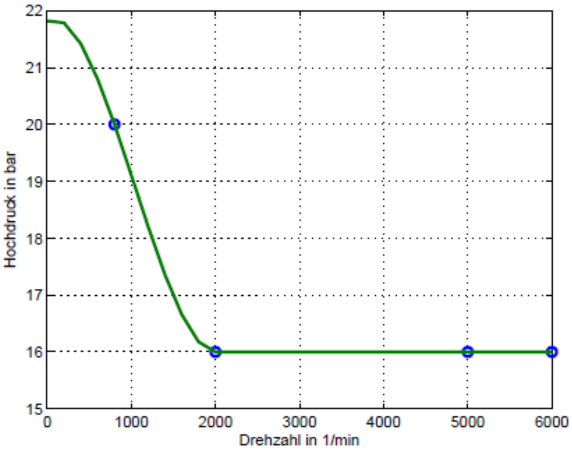


Figure 3.8: ***Outlet-pressure*** P_2

From those two diagrams, it is very easy to find a common point which both of pressure value keeps constant at the higher speed. A point of reference is needed to indicate the conditions found under the ensured valve control. The Lookup Table in MATLAB application is highly recommended. It was used to control the this block operation, especially because its superior calculation method was proven when used to establish either the pressure in the evaporator or for the condenser.

3.3.3. *Wobble plate and control valve*

About the wobble-plate, the mass flow and hence the performance of compressor effectively regulated, basic function mentioned in previous chapter mechanical construction. This may support the volume of between 2% and 100%. The information from the specification relating to the swept volume is listed in Table 3.2

summarizes:

Table 3.2: Capacity

Funding volume at 100% stroke	$140cm^3$
Minimum Stroke	$< 2\%$
Capacity at the turn	$< 2\%$

The controls of the wobble-plate are accessed via signal sending by microprocessor. The relationship between the control rule and the suction power is linear. Tables were written for the low and high pressure from Section 3.3.2 by an expected regulated behaviour, the arrangement of the wobbleplate only while calculating the mass flow, was used the calculated value for the mass flow simply by a factor between 0.02 < x<1.

3.6.1. Simulink-model Compressor

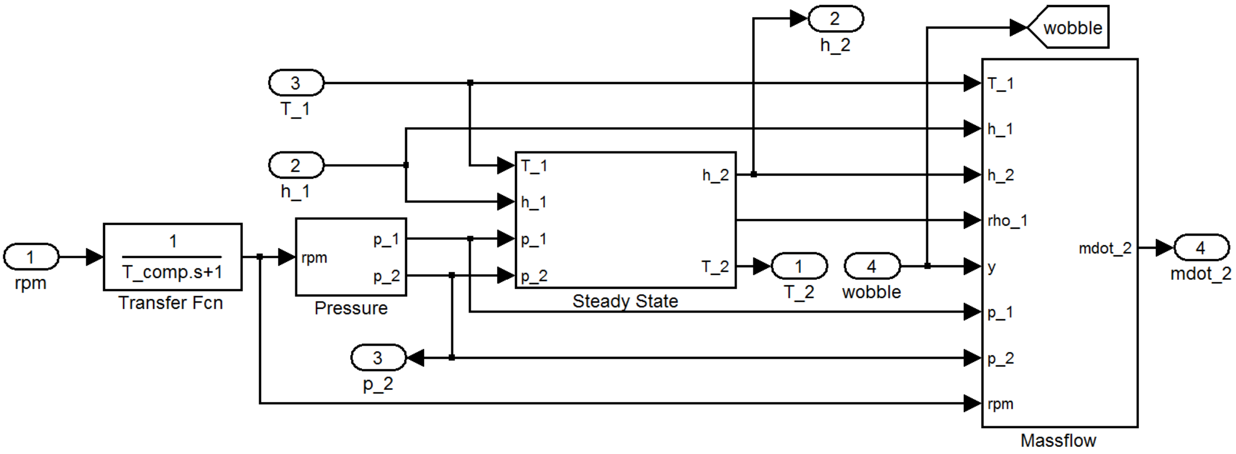


Figure 3.9: ***Compressor Modelling***

Figure 3.9 shows the main model of the compressor, which has been subdivided into different subsystems. The subsystems are as following:

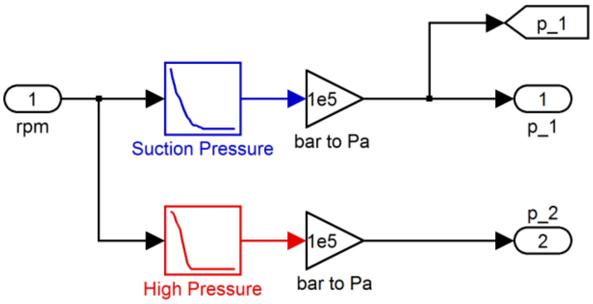


Figure 3.10: *Subsystem Pressure*

This subsystem pressure is from the interpolated *Figure 3.10* inlet and outlet pressure as described in Section 3.3.2

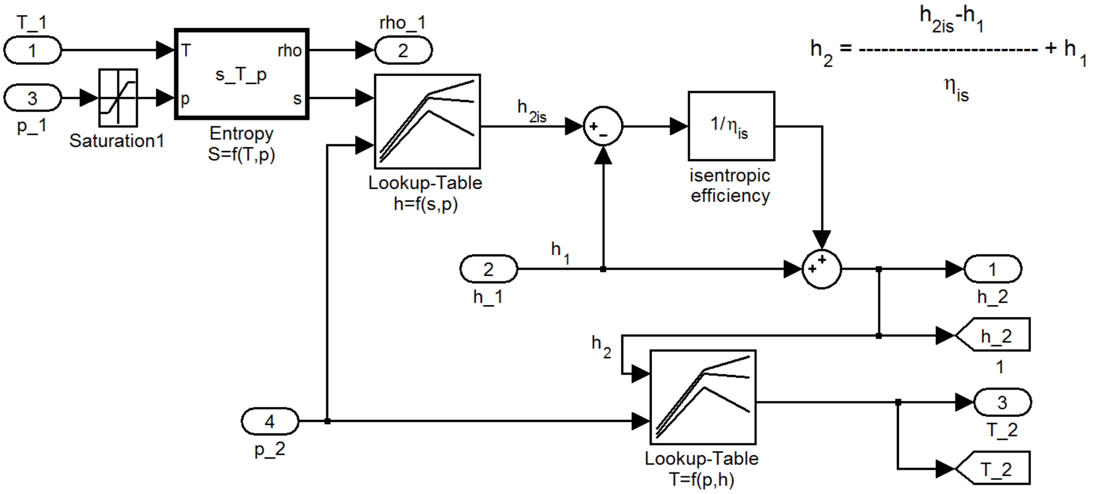


Figure 3.11: *Subsystem Steady State*

The calculation in MATLAB/Simulink will be held for steady state subsystem:

The states T1 and h1 are known from the evaporator. With the suction P1 can be accessed via the function s = f (T, p), the density rho_1 as the entropy s1 can be determined.

Because s1 =s2 then the enthalpy h_{2is}can be calculated by S2 and P2 in function h = f (s, p). The value of P2 is for condenser.

The Equation shows in *Figure 3.11* shows how the real enthalpy outlet for compressor h2 calculated. h2 and P2 can be used by reference to the 2-D lookup table T = f (p, h), temperature can be interpolated.

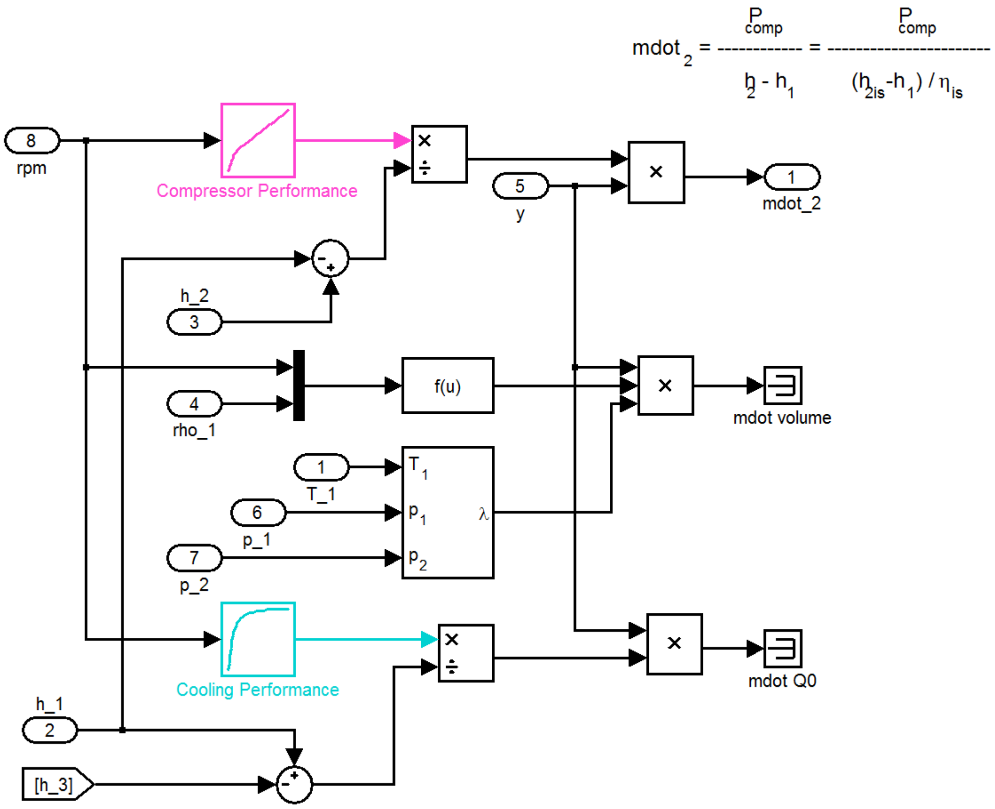


Figure 3.12: *Subsystem Massflow*

In the subsystem Massflow calculates the mass flow by the method already described in Section 3.3.1. The currently used mass flow resulting from the compressor performance is calculated Pk, while the rest of the sink block is terminated (mdot volume & mdot Q0).

3.4. Condenser

A condenser is normally used for the high temperature heat exchanger device in the HVAC system, the heat energy is produced from the compressed high pressure refrigerant. The working status of this device in the cooling mode mainly includes two conditions, namely the steady condition and the unsteady condition. A steady condition runs most of the time at refrigerant R134a as a superheat vapour state at the inlet of the condenser. After the vapour has condensed, a two phase state which combines vapour and liquid of R134a occurs, as R134a becomes more condensed, a sub-cooled liquid will be found at the outlet of the condenser. Should an unacceptable or faulty condenser design become apparent, and insufficient mass of R134a occurs, this will prevent a sub-cooling liquid at the outlet of the condenser. An unsteady condition is emitted. This condition R134a always changes corresponding with the time domain. This means is analysis of the refrigerant in a condenser will always be concerned about this initial status, until all mutual-impact parameter become stabilized. Once the compressor modelling project had been successfully undertaken, it was establish that relationship between suction pressures versus compressor speed was linear; this was taken into account during discussion and decided that focus would continue to centre on the steady condition of condenser modelling.

In the steady parameter modelling of a condenser it was found that different parameter complexities cause different modelling. If the modelling concerned contained more detail for fluid heat transferring, then 3-D modelling would be essential. One needs to note that there is some disadvantage could be attached to 3-D modelling which requires huge computation; this, in turn could mean worse stability of calculation. The precision solution may not solved as respected, the reason being that the accuracy result does not only ask for fluid heat-transferring, but actually required through ask the better calculation of the heat transfer coefficient. Any numeral simulations need optimizing which depends on this assumption, due to simplified calculation or facilitate analyzing.

3.4.1. Modelling assumptions

The modelling methods to be presented require several assumptions about the fluid flow in the heat exchangers. These assumptions were commonly used in past modelling efforts and are outlined as follows:

- The heat exchanger is a long, thin, horizontal tube.
- The refrigerant flowing through the heat exchanger tube can be modelled as a

one-dimensional fluid flow.

- Axial conduction of refrigerant is negligible.
- Pressure drop along the heat exchanger tube is due to momentum change in refrigerant and viscous friction is negligible (refrigerant pressure along the entire heat exchanger tube can be assumed to be uniform). Thus the equation for conservation of momentum is not needed.

3.4.2. Thermo-element Schema

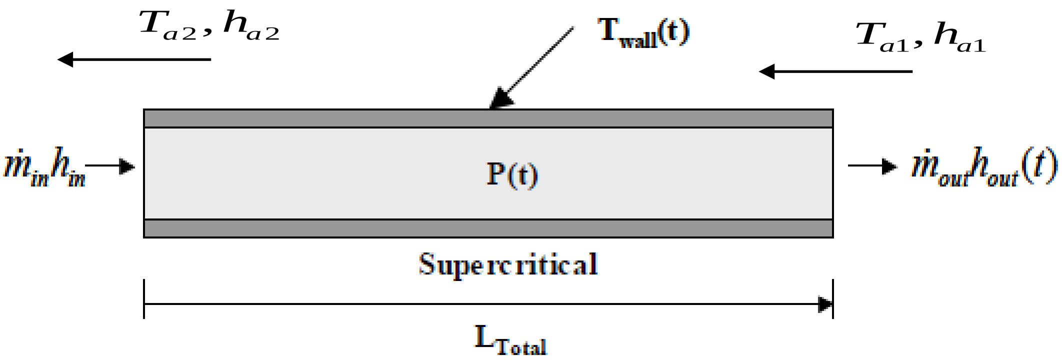


Figure 3.13: ***Thermo-element schema***

The modelling of condenser has to consider three states: superheat vapour (V), two-phase (TP) and sub-cooled liquid (L). Each state could divide into a number of thermo-elements; these elements are classified as temperature or thermal enthalpy, due to the pressure drop which is negligible. Thermo-element schema showing *Figure 3.13* is above.

3.4.3. The equation used in energy transfer:

Air-side heat transfer equation: $$Q_a = m_a (h_{a2} - h_{a1}) \qquad (Eq.3.13)$$

Refrigerant-side heat transfer equation: $Q_r = m_r(h_{r2} - h_{r1})$

$$(Eq.3.14)$$

Tube inside / outside the energy conservation: $Q_a = \varepsilon \cdot Q_r$

$$(Eq.3.15)$$

Thermal-element conductivity equation: $Q_r = U \cdot A_i(T_{rm} - T_{am})$

$$(Eq.3.16)$$

Refrigerant-side average temperature: $$T_{rm} = \frac{T_{r1} + T_{r2}}{2} \qquad (Eq.3.17)$$

Air-side average temperature: $$T_{am} = \frac{T_{a1} + T_{a2}}{2} \qquad (Eq.3.18)$$

Length of tube equation: $$L = \frac{A_i}{\pi d_i}$$

$$(Eq.3.19)$$

Notes: Equation.3.15 the leakage thermal coefficient ε [1], according to the experimental determinate as 0.8 < ε< 1, applied on condenser modelling this value is normally 0.9

3.4.4. *The equation for the tube:*

$$U = (\frac{1}{a_i} + \frac{A_i}{a_o A_o})^{-1} \qquad (Eq.3.20)$$

In the Equation.3.20 U is the heat transfer coefficient for the tube; $a_i$is the heat transfer coefficient of refrigerant; a_o is the heat transfer coefficient of air. $\frac{A_i}{A_o}$is the proportion of area, which includes internal surface of the tube for R134a and external surface of the tube for air. Those values are calculated from hydraulic diameter (Dh), which is a commonly used term when handling flow in noncircular tubes and channels. In the Equation.3.21 where A is the cross section area and P is the wetted perimeter of the cross-section.

$$D_H = \frac{4A}{P} \qquad (Eq.3.21)$$

If the condition of inlet and outlet tube is known, the length of each thermo-element in Equation3.22 can be determined.

$$L = \frac{(\frac{1}{a_i} + \frac{A_i}{a_o \cdot A_o})m_r(h_{r1} - h_{r2})}{(T_{rm} - T_{am})\pi d_i} \qquad (Eq.3.22)$$

3.4.5. *Heat transfer coefficient*

The heat transfer coefficient, in thermodynamics and in mechanical and chemical engineering, is used in calculating the heat transfer, typically by convection or phase change between a fluid and a solid. It is the proportionality coefficient between the heat flux, and the thermodynamic driving force for the flow of heat. There are numerous methods for calculating the heat transfer coefficient in different heat transfer modes, different fluids, flow regimes, and under different thermo-hydraulic conditions. Often it can be estimated by dividing the thermal conductivity of the convection fluid by a length scale.

A common and particularly simple correlation useful for many applications is the Dittus-Boelter heat transfer correlation for fluid in turbulent flow. This correlation is applicable when forced convection is the only mode of heat transfer that is, there is no boiling, condensation, significant radiation, and etcetera. the accuracy of this correlation is anticipated to be 15%.

Dittus-Boeler relational equation: $Nu_i = 0.023\,\mathrm{Re}^{0.8}\,\mathrm{Pr}^n$ (*Eq*.3.23)

$$\mathrm{Re} = \frac{G_i \cdot d}{\mu} \qquad \mathrm{Pr} = \frac{C_p \cdot \mu}{\lambda}$$

Notes: In the equation.3.23, n value is different during condensation or evaporation.

- Condensation n=0.3
- Evaporation n=0.4

For the signal state (super-heat vapour, sub-cooled liquid), heat transfer coefficient a_i can be calculated by Dittus-Boeler relational equation Nu.

$$Nu_i = \frac{a_i \cdot d_i}{\lambda}$$

For the double state (two-phase), heat transfer coefficient a_{TP} defined as Shah relational equation [2] in the equation.3.24:

$$a_{TP} = a_i[(1-x)^{0.8} + \frac{3.8x^{0.76}(1-x)^{0.04}}{\Pr^{0.38}}] \quad (Eq.3.24)$$

Notes: x is possession of the proportion between liquid and vapour in the two-phase state.

For air side heat transfer coefficient a_o, it is important to find out what types of condenser are used on the prototype. VW Polo utilized the louvered condenser which is produced by BEHR. Due to air side heat transfer coefficient is strongly restrict by the structural condenser. It cannot use the traditional Nusselt number (Nu) method to calculate heat transfer coefficient, so Devenport C J. [3] researched for heat transfer and friction characteristics of a louvered fin, and summarized a J-factor equation as follows:

$$j = \frac{Nu}{\mathrm{Re}_{Pl} \cdot \Pr^{1/3}} = 0.249\,\mathrm{Re}_{Pl}{}^{-0.42} H_L{}^{0.33}(\frac{L_L}{H_F})^{1.1} H_F{}^{0.26}$$

$$(Eq.3.25)$$

The value used in Equation.3.24 refers to *Figure.3.14*

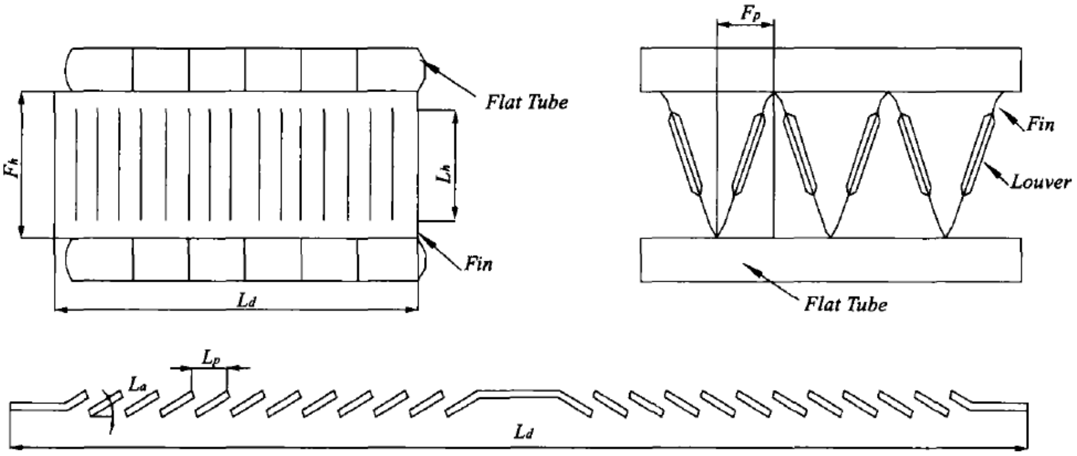


Figure 3.14: ***Definition of geometrical parameters for a louver heat exchanger***

This specification is the only one record on the condenser's datasheet which is confidential, so all of the values used in the modelling are measured values.

3.4.6. Algorithm design for condenser modelling

From the basic equation mentioned above, Iterative algorithms have to be used in calculating the length of a condenser tube. According to the Simulation modelling, inlet of air and refrigerant are necessary elements as an input condition for condenser modelling. Outlets of that parameter are the simulation target. Flowchart for modelling is shown in *Figure 3.14*

A dichotomous iterative algorithm used in this modelling, process is described as follow:

Set up an outlet enthalpy for one thermo-element. The range of outlet enthalpy may be explained as: lower limit of R134a enthalpy should correspond with the inlet of air enthalpy; the upper limit of R134a enthalpy should correspond with air outlet, the temperature should be set as required. Lower or upper enthalpy is as the initial value for the dichotomy algorithm and the arithmetic average of enthalpy used for the initial iterative value.

The specified outlet enthalpy predicates at which state the refrigerant is. A note the different refrigerant of state corroding to different thermo-element schema in *Figure 3.13*, then separately calculate the length of the condenser tube belongs to each parts. Compare the calculated length with actual length of tube, if calculated length is longer than the actual one, then reset a smaller outlet enthalpy, until these two matches each other, then output the value of enthalpy.

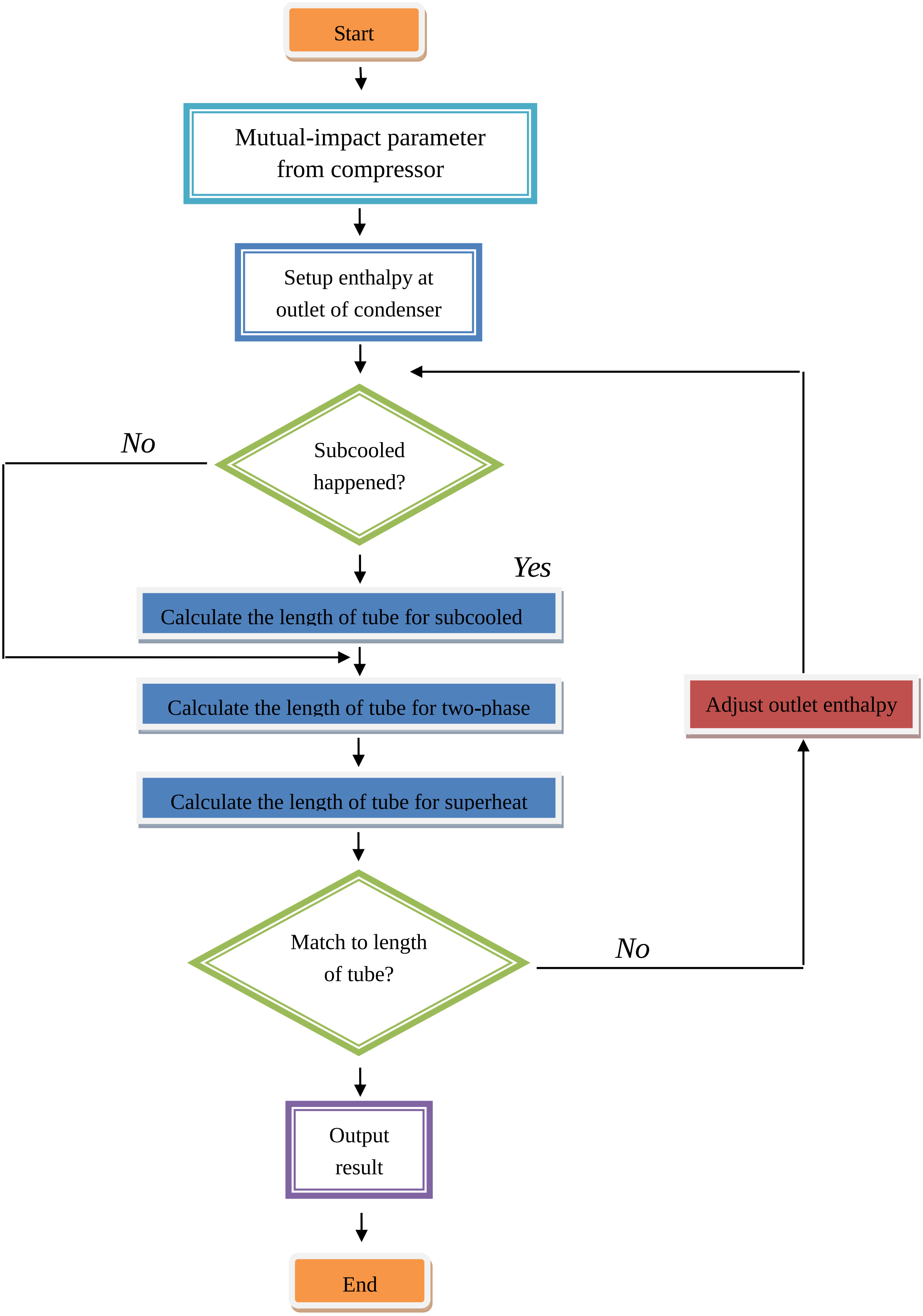

Start
Mutual-impact parameter from compressor
Setup enthalpy at outlet of condenser
Subcooled happened?
No
Yes
Calculate the length of tube for subcooled
Calculate the length of tube for two-phase
Calculate the length of tube for superheat
Adjust outlet enthalpy
Match to length of tube?
No
Output result
End

Figure 3.15: ***Flowchart of condenser modelling***

3.4.7. *Simulink model of condenser*

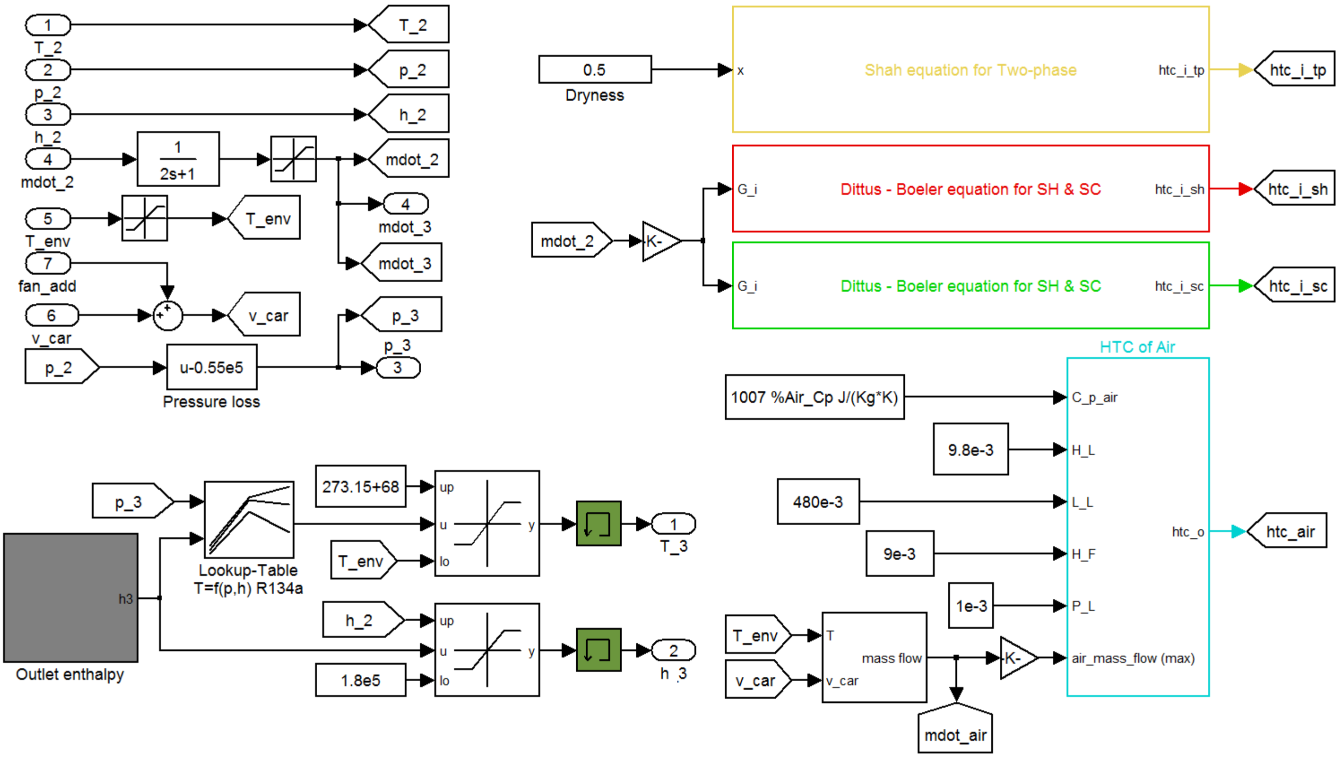


Figure 3.16: *Condenser Modelling*

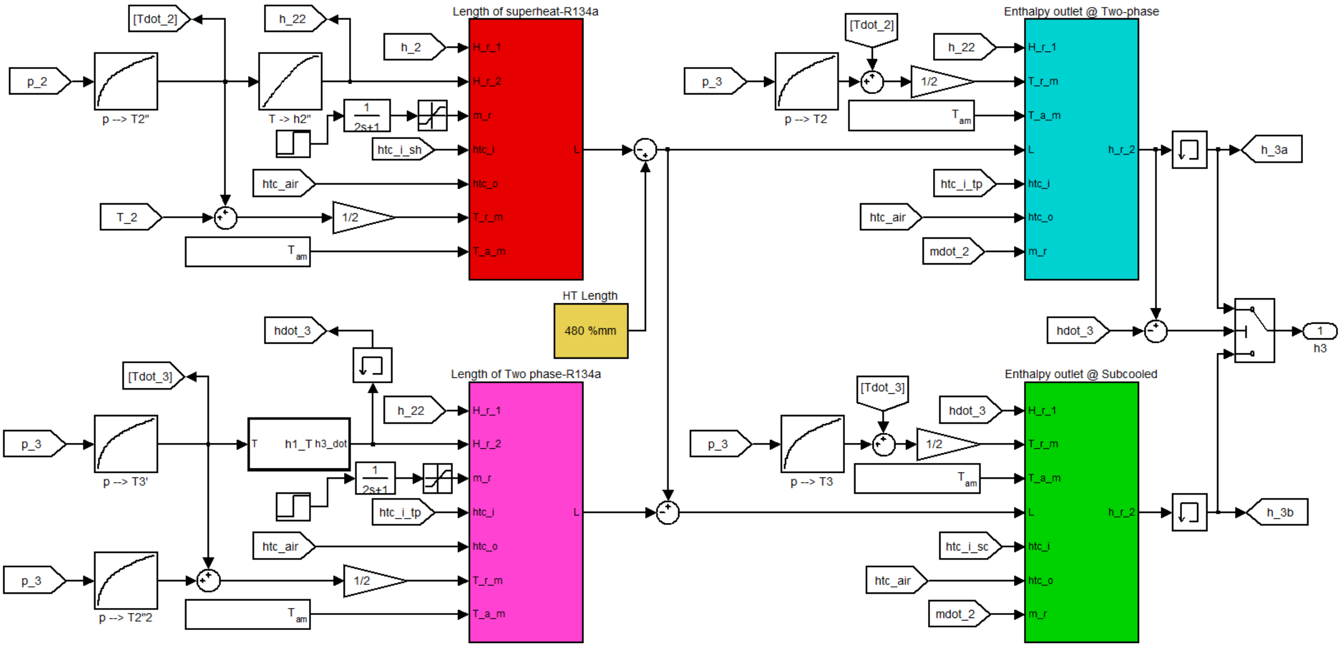


Figure 3.17: *Outlet Enthalpy*

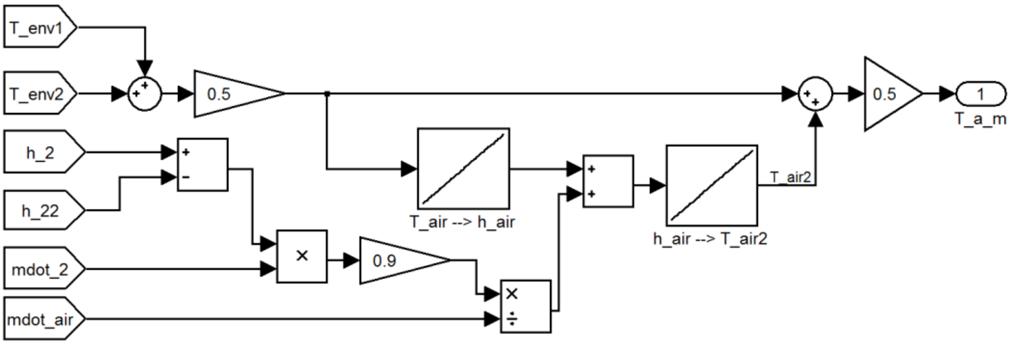


Figure 3.18: *Air-side average temperature*

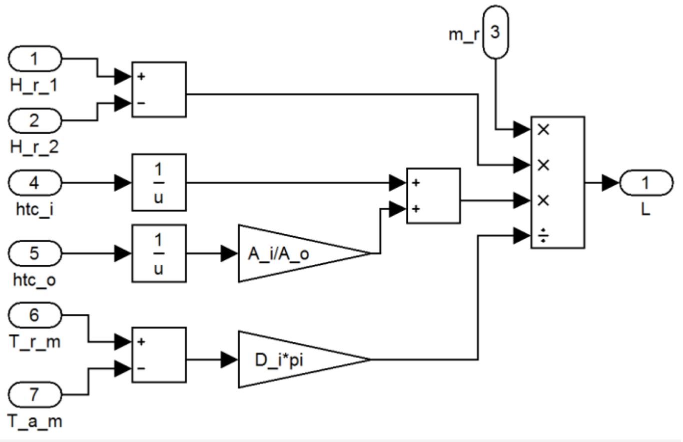


Figure 3.19: *Length of thermo-element*

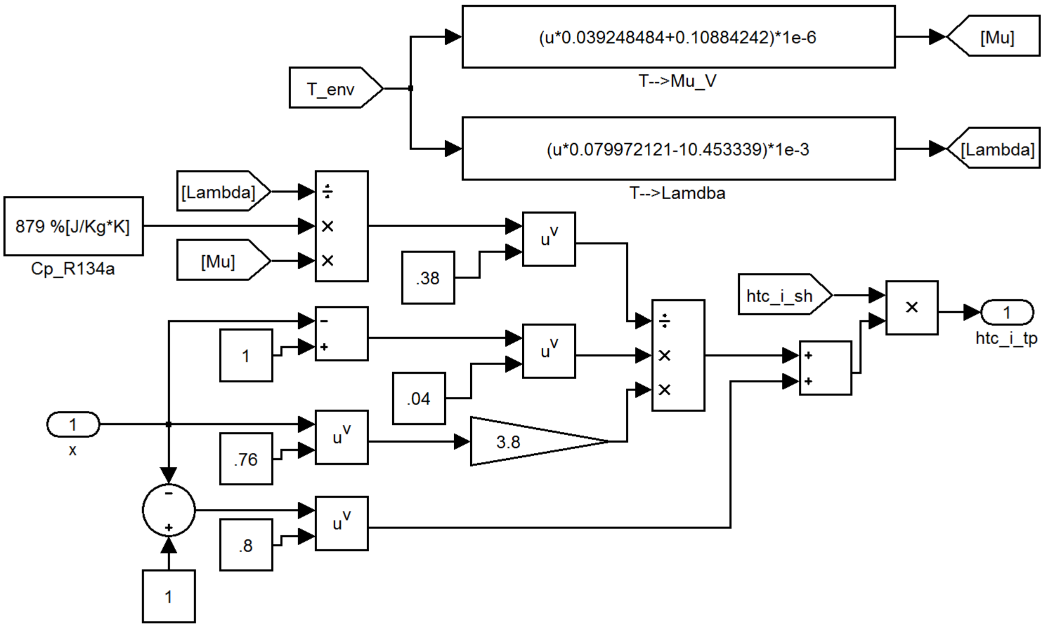


Figure 3.20: *Shah Equation for TP heat transfer coefficient*

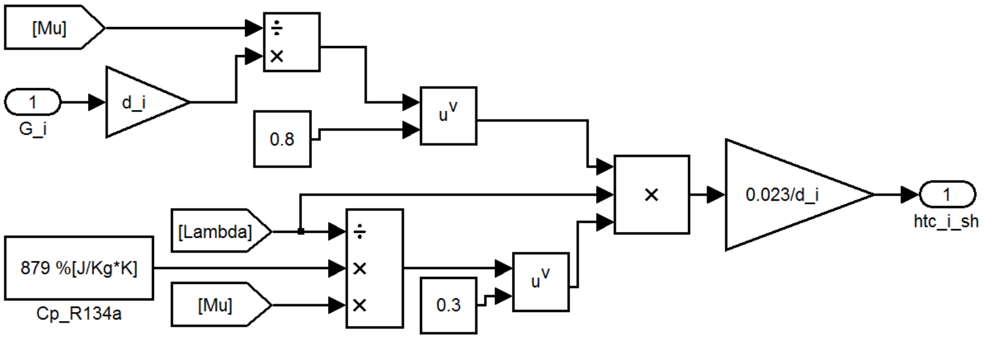


Figure 3.21: *Dittus-Boeler Equation*

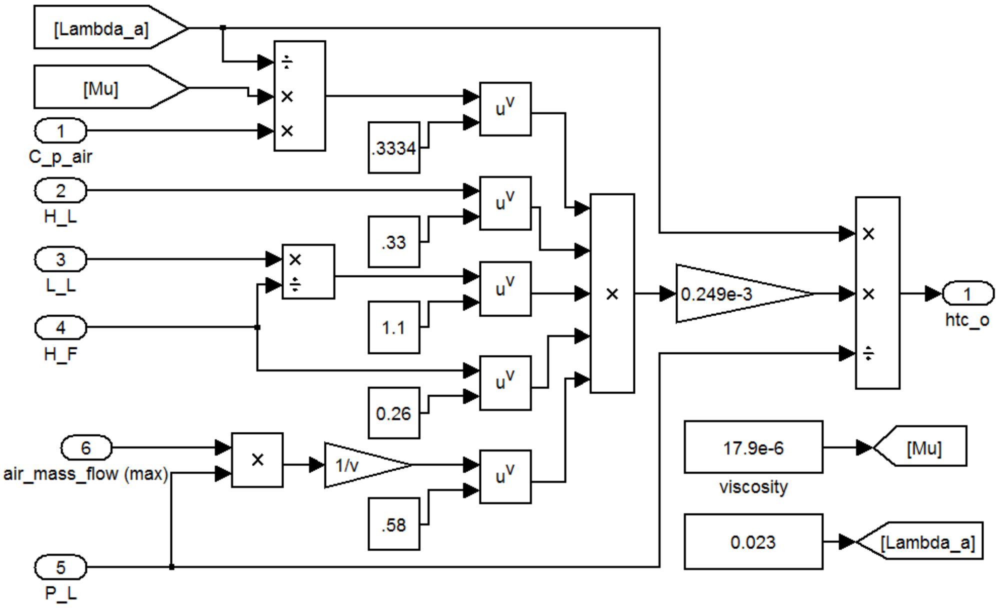


Figure 3.22: *Devenport Equation for air heat transfer coefficient*

3.5. Expansion valve

As mentioned in Chapter 2.3, the expansion valve forms the interface between the high-pressure side and low-pressure side of the refrigerant circuit.

The VW Polo installed expansion valve belongs to a thermal valve without any additional electronic control unit. It regulates the refrigerant flow into the evaporator, and is controlled by the exit temperature of the vapour at evaporator exit. This regular temperature provides the overheating of the refrigerant at evaporator. The thermal regime as the function of the overheating temperature is in accordance with *Figure 3.23* the overheating temperature ΔT calculated from the difference between the actual temperatures T_1 and the temperature T_1' of the saturated vapour pressure at p_1

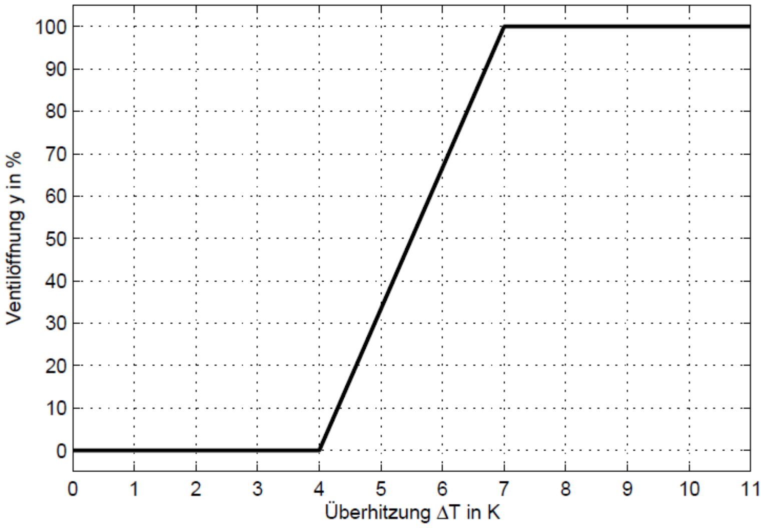


Figure 3.23: ***Overheating temperature vs. Valve opening***

Since the dependence of the flow from the valve position is not known, in a logarithmic model, a linear relationship is deposited. The selection of one of the two curves is in the block valve position. It is also possible in this block, to determine the maximum and minimum flow rate.

3.5.1. *Volume flow*

$$Y_v = \zeta_v \cdot \frac{c^2}{2} \qquad (Eq.3.26)$$

$$\Delta p_v = \rho \cdot Y_v = \rho \cdot \zeta_v \cdot \frac{c^2}{2} \qquad (Eq.3.27)$$

Equation.3.26 is the resistance formula according to the Darcy flow velocity c. resistance ζ_v involved. This resistance number is fitted for each part, related to the exit velocity, determined by experimental result . For the pressure difference Δp_v is between throttle input and output, in this case $\Delta p_v = p_3 - p_4$. The volume flow rate $\dot{V}_v$ can be calculated by Equation.3.28.

$$\dot{V}_v = A \cdot c_3 = A \cdot \sqrt{2 \cdot \frac{p_3 - p_4}{\rho_3 \cdot \zeta_v}} \qquad (Eq.3.28)$$

A is the area of the narrowest point in the valve. It follows from the mass flow of the valve on the general equation.

$$\dot{m}_v = \rho_3 \cdot \dot{V}_v = A \cdot \sqrt{2 \cdot \rho_3 \frac{p_3 - p_4}{\zeta_v}} \qquad (Eq.3.29)$$

As same with the compressor modelling, the momentum of the expansion valve is much greater than the condenser's and evaporator's momentum. Therefore the dynamics of the valve by a PT1 with a time constant Tv = 0.2 s approximated.

3.5.2. *Static behaviour*

The expansion process is an is-enthalpy process, inlet and outlet enthalpy for the expansion valve should be equal, so

$$h_3 = h_4 \qquad (Eq.3.30)$$

The pressure drop during the expansion process could seem to have as a static behaviour; this could be due to the pressure difference between the inlet and outlet of two-heat exchangers with an almost constant value. But only one issue has to be considered in the cooling circuit, and that is the pressure drop for the evaporator at the two-phase state of R134a. Because of the pressure drop normally exceeds 1 bar, and the density of refrigerant ρ_1, ρ_2 and therefore a great difference occurs. Subscript 1 and 2 represent inlet and outlet for thermo-element methods of the two-phase state R134a, see (*Figure 3.4.1)*. This thermo-element method fits into both of the two heat exchangers.

$$\frac{p_1 - p_2}{L} = \frac{4fG_r^2}{\rho_1 d_i} + \frac{G_r^2}{L}\left(\frac{1}{\rho_2} - \frac{1}{\rho_1}\right) \qquad (Eq.3.31)$$

In a typical example; one should consider the pressure drops on the superheated state of the evaporator tube. The following value would confirm the argument in the last paragraph:

Inlet pressure of the expansion valve is at 20.328bar; temperature at 45.0℃; Outlet pressure of evaporator is at 6.6264bar; temperature at 9.42℃; Mass flow rate for R134a is 62.34 kg/h. Hydraulic diameter of evaporator is 9mm, and Length of two-phase R134a is 12m and length of superheat is 2m.

According to Equation 3.31, the pressure drop can be determined, two-phase area has 0.6bar, superheat area has 0.05bar. If the pressure drop superheated area is neglected, the superheat temperature difference is only 0.12℃. So the magnitude of pressure value on the superheat area is small, and the influence of temperature at the outlet of evaporator may be ignored.

***3.5.3.** Simulink-Model of expansion valve*

Superheat control temperature ΔT on the valve position (*Figure 3.24*) is the throttling range function block as seen in *Figure 3.25* below.

Calculation of volume flow $\dot{V}_v$by Equation 3.28 and mass flow $\dot{m}_v$ Equation 3.29

(*Figure 3.24* middle); *Figure 3.25* shows the subsystem valve position; Calculation of pressure loss on the evaporator for low pressure cycle by Equation 3.31

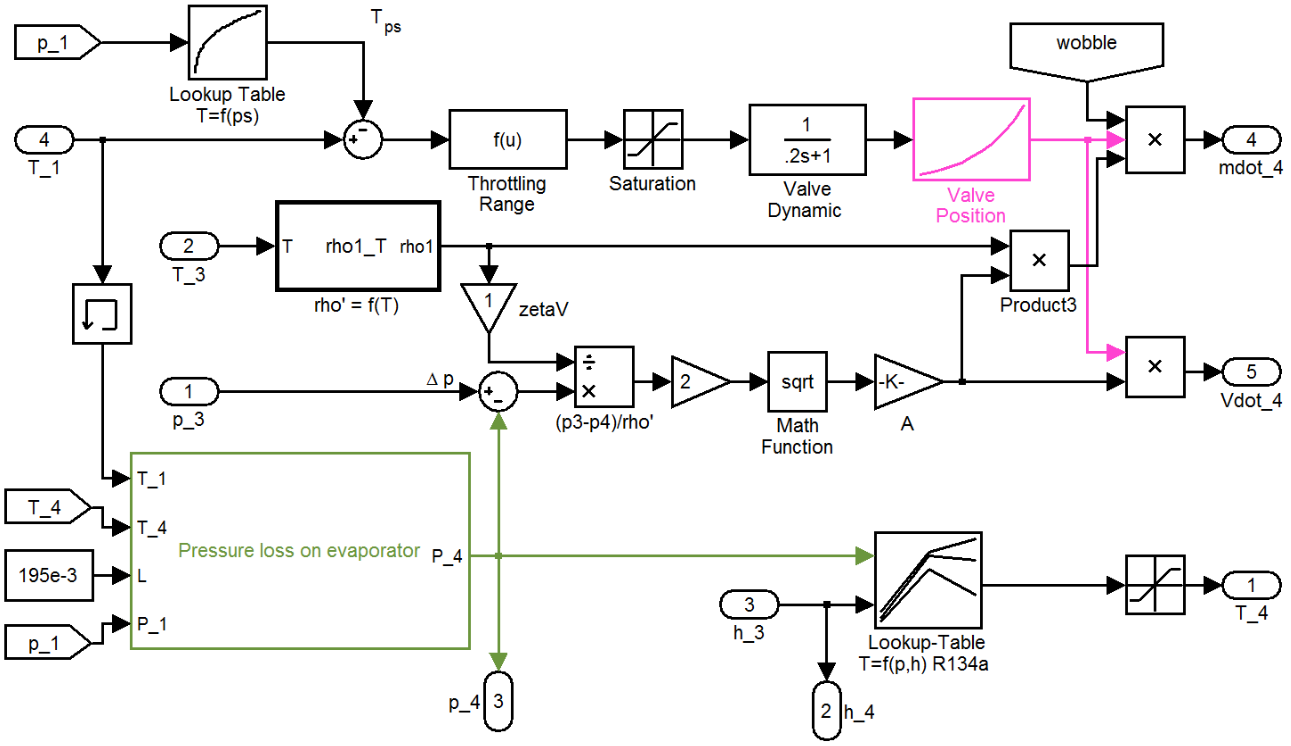


Figure 3.24: ***Expansion valve modelling***

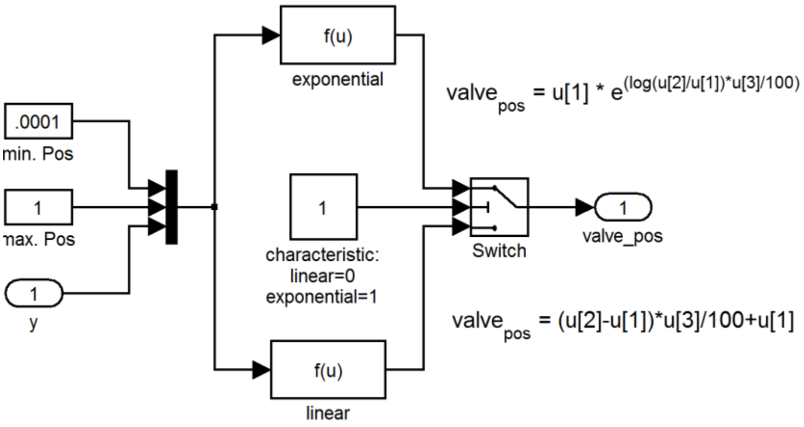


Figure 3.25: ***Subsystem valve position***

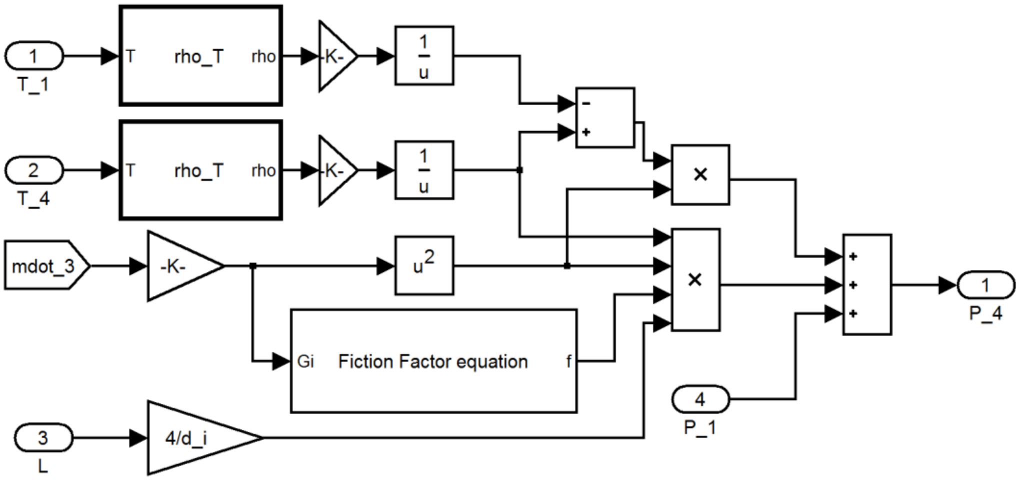


Figure 3.26: *Subsystem Pressure loss*

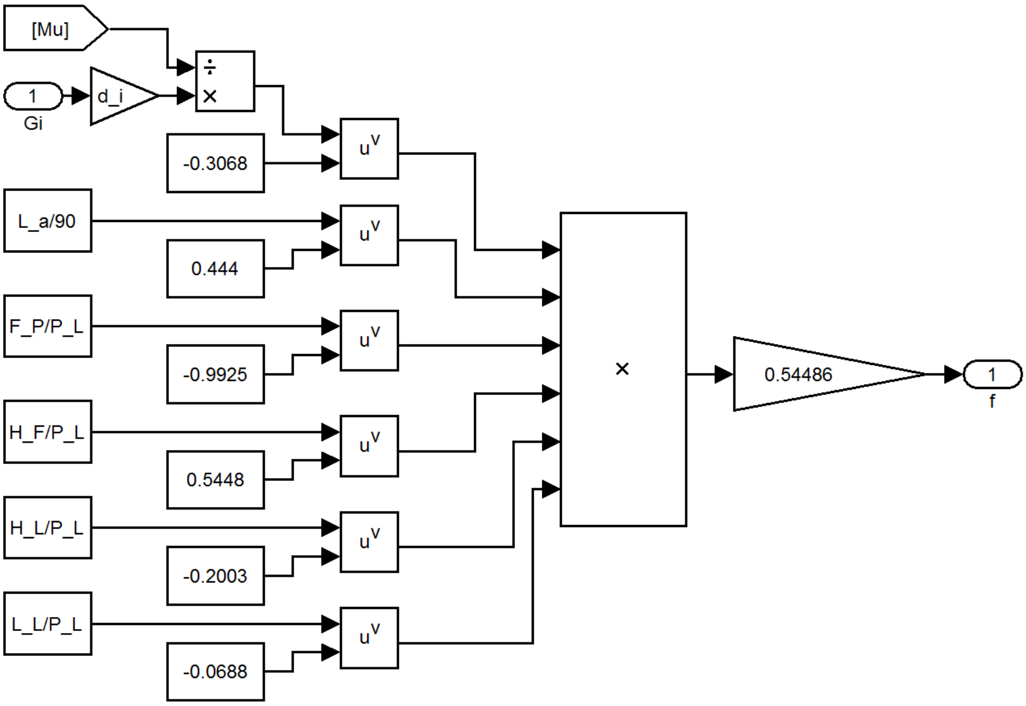


Figure 3.27: *Subsystem Friction factor*

f was defined as Fiction Factor in the equation.3.31. In fluid dynamics, this value is used to calculate the pressure drop and flow rates in a section of a uniform pipe running for a certain distance. The pipe is assumed to be relatively straight, such that changes in pressure are due mostly to elevation changes and wall friction.

$$f = -0.54486\,\mathrm{Re}^{-0.3068}\left(\frac{L_a}{90}\right)^{0.444}\left(\frac{F_p}{L_p}\right)^{-0.9925}\left(\frac{F_h}{L_p}\right)^{-0.2003}\left(\frac{L_d}{L_p}\right)^{-0.0688}$$

$(Eq.3.32)$

In the section.3.4.2, the heat transfer coefficients and pressure drop data for different fin pitch Fp and fin height Fh were reported in terms of Colburn j-factor and Fanning friction factor f, as function of Re. Both J and F factor equations have also been summarized by experimental studies at Shanghai Jiaotong University, where over 20 types of multi-louvered fin and flat tube heat exchangers were performed. A series of tests were conducted for air Reynolds numbering from 200 to 3000 based on the louver pith with different fin pitch, fin height, fin thickness, fin louver angle and flow length. The tests were kept at a constant tube –side flow rated, and the air side thermal performance data were analyzed using the effectiveness-NTU method. Equation.3.32 for f factor has RMS errors of $\pm 10\%$ and $\pm 12\%$, and the mean deviations of 4.1% and 5.6%, respectively , in the Re number range of 200 to 3000.

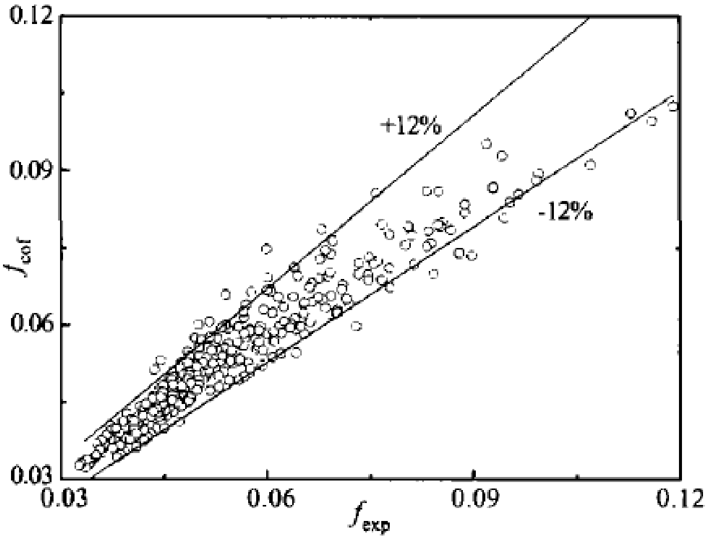


Figure 3.28 ***Comparison of experiment data and correlation for f factor***

3.6. Evaporator

An evaporator is normally used for the low temperature heat exchanger device in the HVAC system. During the cooling mode this energy is utilized for heat exchanging; during the evaporation mode, superheat refrigerant or expanded refrigerant absorbs the air energy to cool the high temperature air. Extracted energy is obtained from warm, fresh air which cools down when air passes through the evaporator fins. Disguised heat is the heat exchange method which always occurs in the evaporator, due to the maximum latent heat the refrigerant could reach in this processing. The evaporator is assumed to operate in several conditions: with one, two or three fluid regions.

Generally, this device which runs at a steady rate with the fluid entering the evaporator is two-phase or in the form of a sub-cooled liquid whereas the fluid exiting the evaporator is superheated vapour. The evaporator is modelled with two regions in mind: a two-phase region, and a superheated region. The boundary between these regions is a moving interface and although extremely difficult to measure physically, it is the variable that holds the greatest interest. If a faulty evaporator design occurred or if there as insufficient mass of R134a during injection phase, the outlet of evaporator R134a would not become superheated vapour. As mentioned before, a thermo expansion valve will avoid this phenomena and to make this device more efficient. In this chapter, factors to determine the efficiency of evaporator will be discussed?

For an unsteady condition, the condition R134a changes always correspond with the time domain. For instance, an air-conditioning system lasts a long time even if unused in the vehicle, R134a in the evaporator is filled in by superheated vapour, when AC system start running, the compressor rapidly extracts the refrigerant, and as the pressure drop the thermo expansion valve opens, changing an unsteady condition gradually to becomes steady. The hunting occurs due to this changed condition, and evaporating pressure balanced. When the compressor switch off, as a result of the pressure difference, balance occurs during the high and low pressure cycle, two-phase of R134a at the evaporator, if temperature of this region is higher than the air, it will case evaporator radiate the heat to air. When the compressor is switched on or off, transformation between unsteady condition and steady condition is really difficult to discern.

In accordance with the achieved modelling expectations in the cooling circuit, the evaporator model will only assume on the steady condition. The same modelling assumptions about the fluid flow used for the condenser derivation are also applied here. Additionally the assumption of a pressure drop on the evaporation process is used, which is defined in the model of the expansion valve.

3.6.1. *Thermo-element Schema*

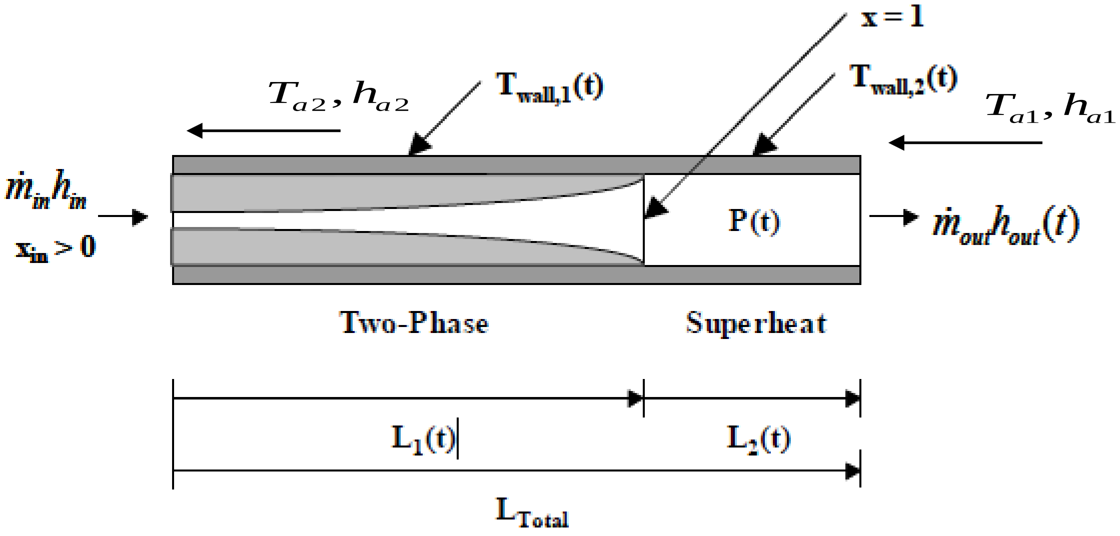


Figure 3.29: ***Thermo-element schema***

The modelling of evaporator has to consider two states: superheat vapour (V), and two-phase (TP). Each state could divide into a number of thermo-elements; those elements are classified as temperature or thermal enthalpy, although the resultant pressure drop is negligible. One element schema is shown (*Figure 3.6.1*)

3.6.2. *The equation used in energy transfer:*

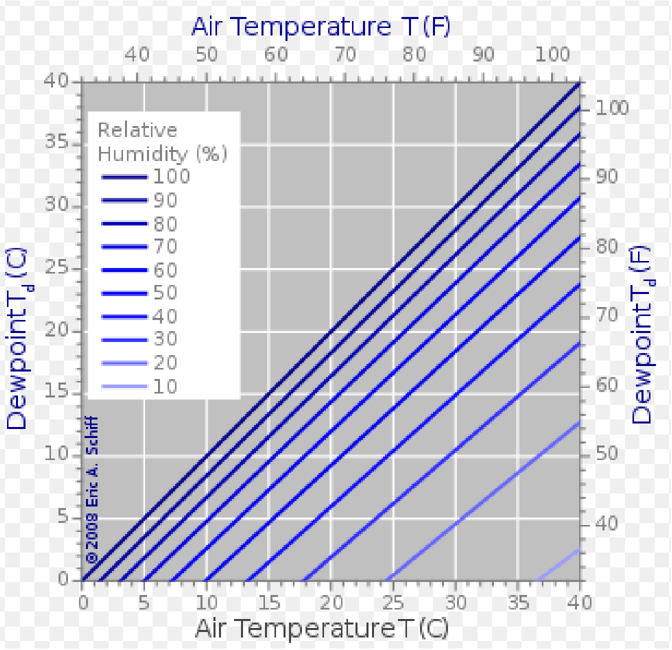
The same equation used in energy transfer on the condenser derivation are applied here, such as [Eq.3-13 to Eq.3-19]. Additionally the energy transfer equation is used in Eq.3-33 and Eq.3-34.

$$Q_r = m_r(h_{r2} - h_{r1}) = \alpha_i A_i (T_w - T_{rm}) \qquad (Eq.3.33)$$

$$Q_a = m_a(h_{a2} - h_{a1}) = \xi \alpha_o A_o (T_{am} - T_w) \qquad (Eq.3.34)$$

Compared to the performance of condenser and evaporator, the difference is not only found at the behaviour of refrigerant, but depends on the heat transfer coefficient. For the cooling mode of the evaporator, the evaporation temperature is normally lower than Dew point, which is the temperature to which a given parcel of air must be cooled at constant barometric pressure, for water vapour to condense into water. (Available: http://en.wikipedia.org/wiki/Dew_point [2, May 2009])

Figure 3.30: *Dewpoint--RH*



Refer to the *Figure 3.30*, when the air blowing through evaporator, humidity of air will change, as the evaporation temperature lower than freezing point, precipitation of water droplets in the air will be solidified, this phenomena is called frosting.

ξ in Eq.3.35 is defined as wet-coefficient. Sometimes this coefficient would improve the ability of heat transfer on the evaporator, as long as the frosting happened, this coefficient would affect lessen the ability of heat transfer.

3.6.3. *The equation for the tube:*

$$L = \frac{Q_r}{(T_w - T_r)\pi d_i \alpha_i} = \frac{\varepsilon \cdot Q_r}{(T_a - T_w)\chi \pi d_i \alpha_o}$$

(*Eq*.3.35)

Notes: Equation.3.35 shows the leakage thermal coefficient ε [1], according to which the experimental determinate as 0.8 < ε < 1, applied on evaporator modelling this value normally is 0.9

3.6.4. *Heat transfer coefficient*

The same equation used in calculating the Superheat region Heat transfer coefficient on the condenser derivation is again applied here, that is [Eq.3-23]. Special attention should be given to the Dittus-Boelter equation. When calculation is for evaporation, value n =0.4. Additionally, the coefficient equation for the two-phase region is applied in Eq.3-36.

$$\frac{\alpha_{Tp}}{\alpha_i} = C_1 (C_0)^{C_2} (25 Fr_l)^{C_5} + C_3 (B_0)^{C_4} F_{fl}$$

(*Eq*.3.36)

Detail for Eq.3-36.

$$\alpha_l = 0.023 \left(\frac{g(1-x)D_i}{\mu_l} \right)^{0.8} \frac{\mathrm{Pr}_l^{\ 0.4} \lambda_l}{D_i}$$

$$C_0 = \left(\frac{1-x}{x} \right)^{0.8} \left(\frac{\rho_g}{\rho_l} \right)^{0.5}$$

$$B_0 = \frac{q}{gr} \qquad Fr_l = \frac{g^2}{9.8 \rho_l^{\ 2} D_i}$$

Notes: Equation.3.36. C_1, C_2, C_3, C_4, C_5 is constant, which is the value depending on the . C_0

When $C_0 \le 0.65$ ， $C_1 = 1.1360, C_2 = -0.9, C_3 = 667.2, C_4 = 0.7, C_5 = 0.3;$

When $C_0 > 0.65$ ， $C_1 = 0.6683, C_2 = -0.2, C_3 = 1058.0, C_4 = 0.7, C_5 = 0.3;$

The Eq.3.36 is called Kandlikar relational equation, which was developed after many thermo-dynamic experiments had been undertaken. The theoretical model specifies that determination of flow two-phase region heat transfer on fin-tube evaporator of R134a unit.

For air side heat transfer coefficient a_o , which is kept the same as used in a condenser. Because of the category, condenser and evaporator remain the same.

3.6.5. *Algorithm design for evaporator modelling*

In accordance with Expansion valve modelling, the following is used as input conditions for evaporator: inlet enthalpy and mass flow rate of refrigeration, the condition of air and volume flow. The output parameter for this modelling only focuses on the outlet enthalpy and the heat-energy used in transfer of R134a. The flowchart is shown on (*Figure 3.31)*. The logic of evaporator modelling is as the logic used for condensers. Dichotomy iterative algorithm is still important for estimating the value of outlet enthalpy, the goal of iteration and convergence condition is to make sure that computed tube length of the evaporator matches for the real tube length, only difference is between evaporator’s and condenser’s modelling. Additionally wet analysis and pressure drop takes place at the two-phase length of tube, this is for calculating the pressure drop that occurred in the expansion valve, but this value is iterative on the evaporator modelling as well. More detail about algorithm design is listed according to the following steps:

Set up an outlet enthalpy for one thermo-element, the range of outlet enthalpy should conform as: lower limit of R134a enthalpy and correspond with the inlet of air enthalpy; the upper limit of R134a enthalpy should correspond with the outlet of air, as should the temperature. Those two values are dominated at either the two-phase region or the superheat region on the log h-p diagram as shown in (*Figure 3.2*). Normally outlet enthalpy is found in the region of superheat. By this means lower or upper enthalpy is the same as the initial value found for the dichotomy algorithm, and the arithmetical average of enthalpy is used for the initial iterative value.

Set up pressure drop on the two-phase region, so that this value can be estimated where the superheat region is.

Make a comparison for both of enthalpy and pressure drop together, and double check the findings.

Specified outlet enthalpy predicate which state of the refrigerant it is. Note the different refrigerant states corroding to different thermo-element schema in (*Fig.3.29)*, and then calculate the length of the evaporator tube.

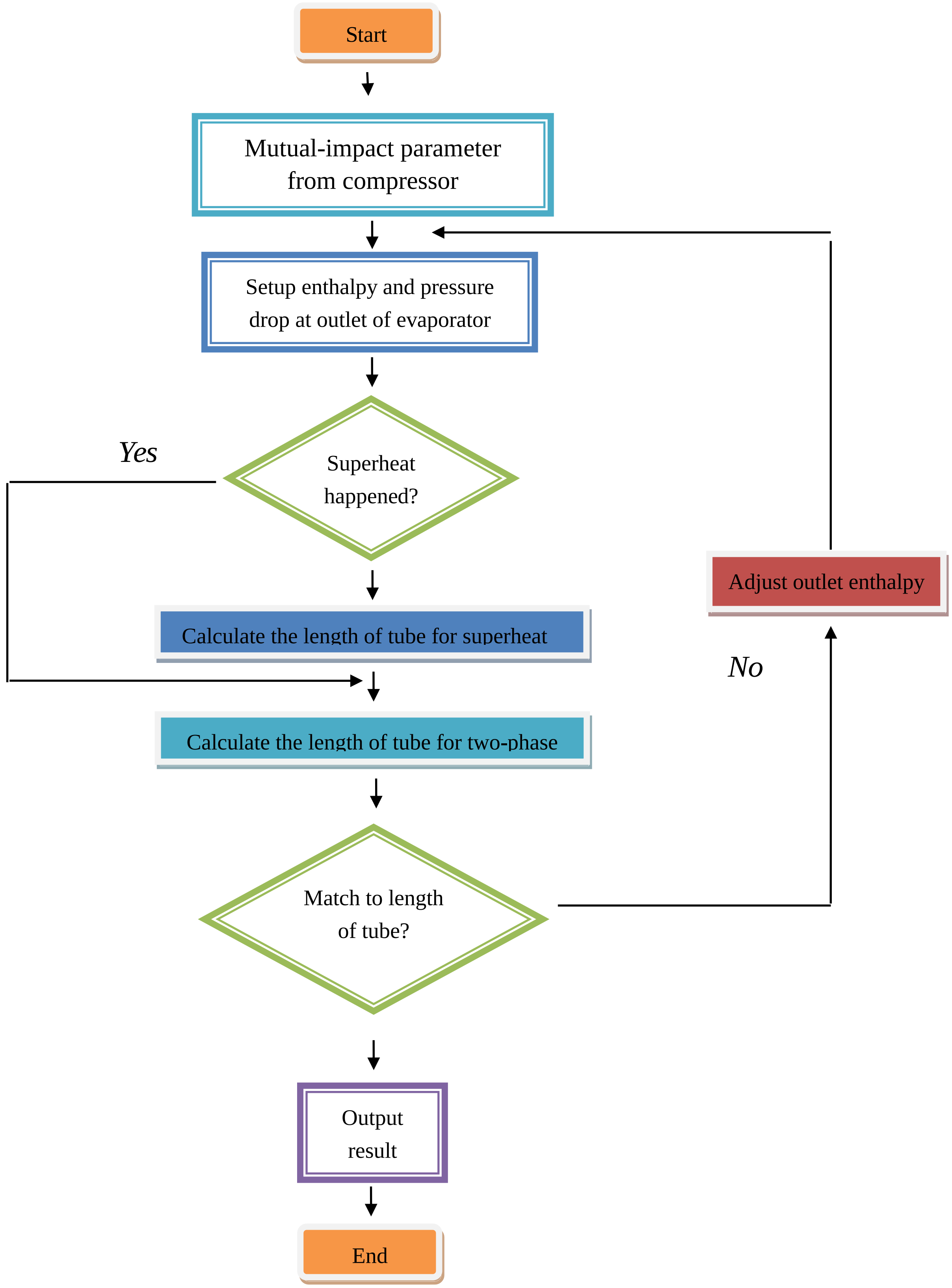

Start
Mutual-impact parameter from compressor
Setup enthalpy and pressure drop at outlet of evaporator
Yes
Superheat happened?
Adjust outlet enthalpy
Calculate the length of tube for superheat
No
Calculate the length of tube for two-phase
Match to length of tube?
Output result
End

Figure 3.31: ***Flowchart of evaporator modelling***

3.6.6. *Simulink model of evaporator*

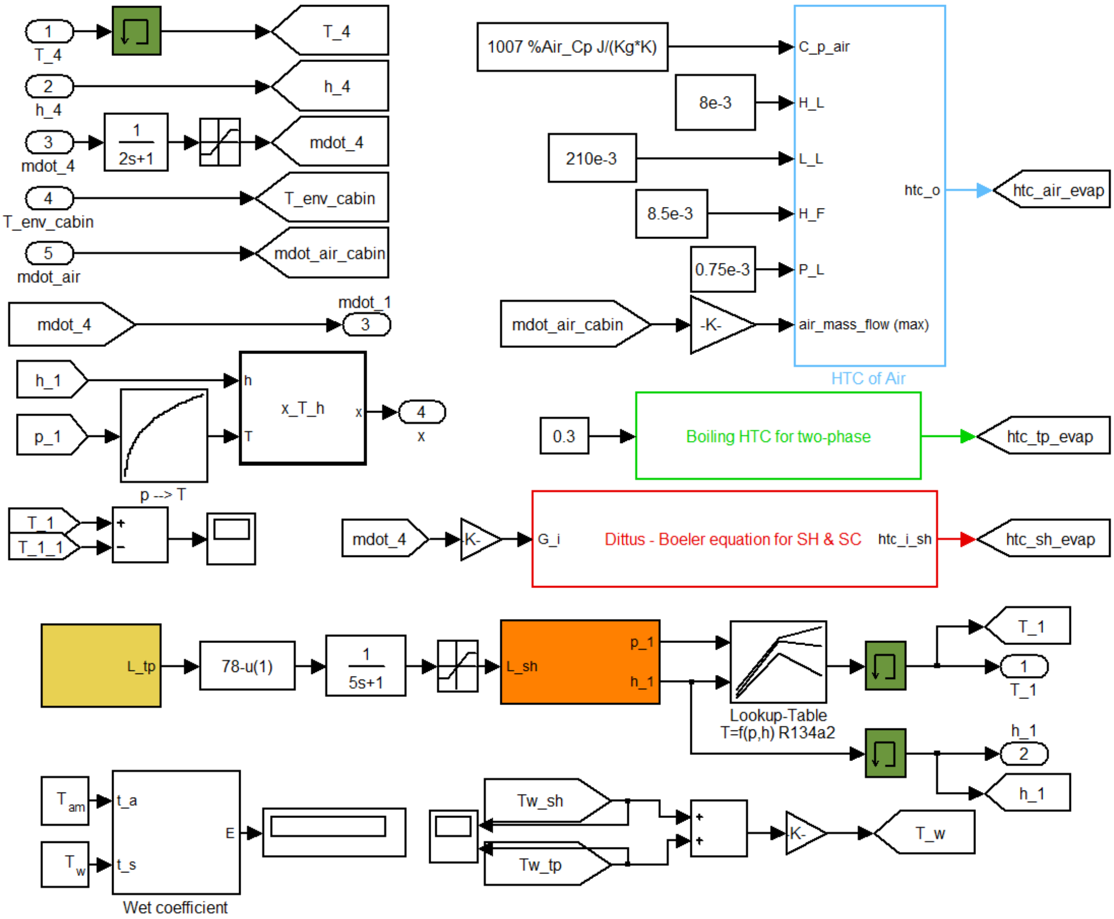


Figure 3.32: *Evaporator Modelling*

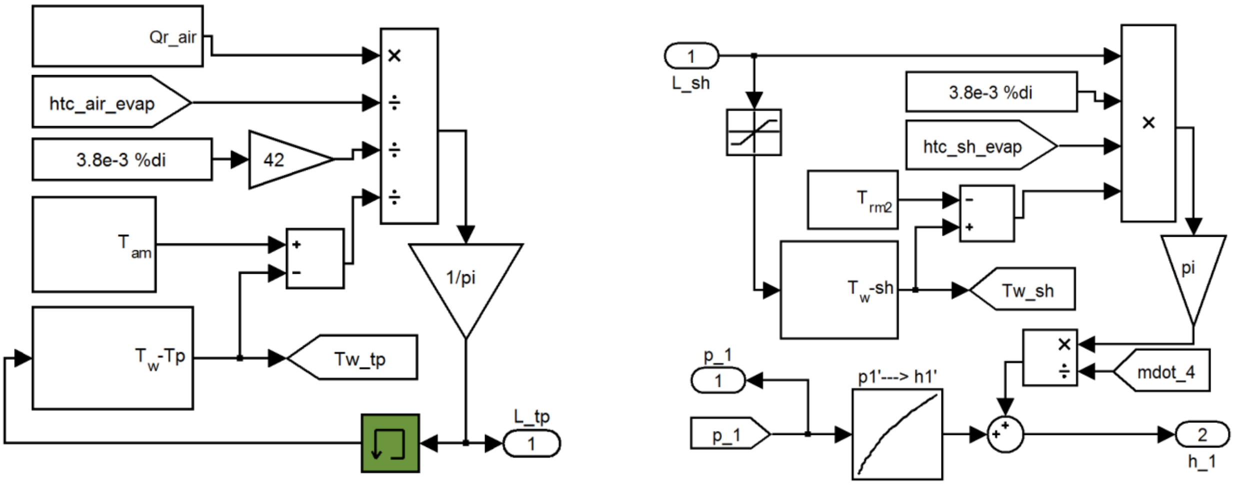


Figure 3.33: *Outlet Enthalpy*

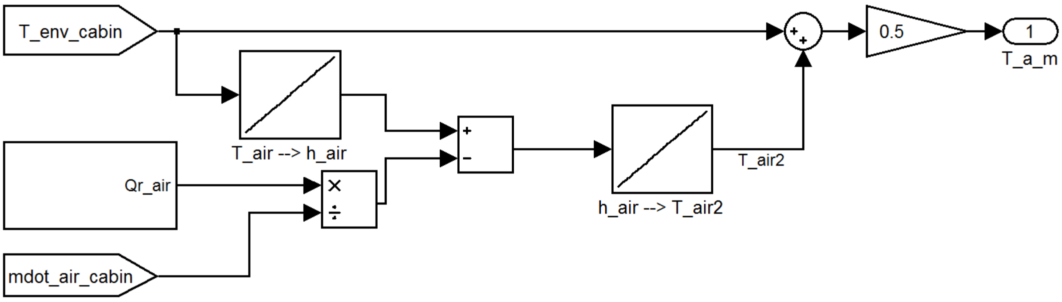


Figure 3.34: ***Air-side average temperature***

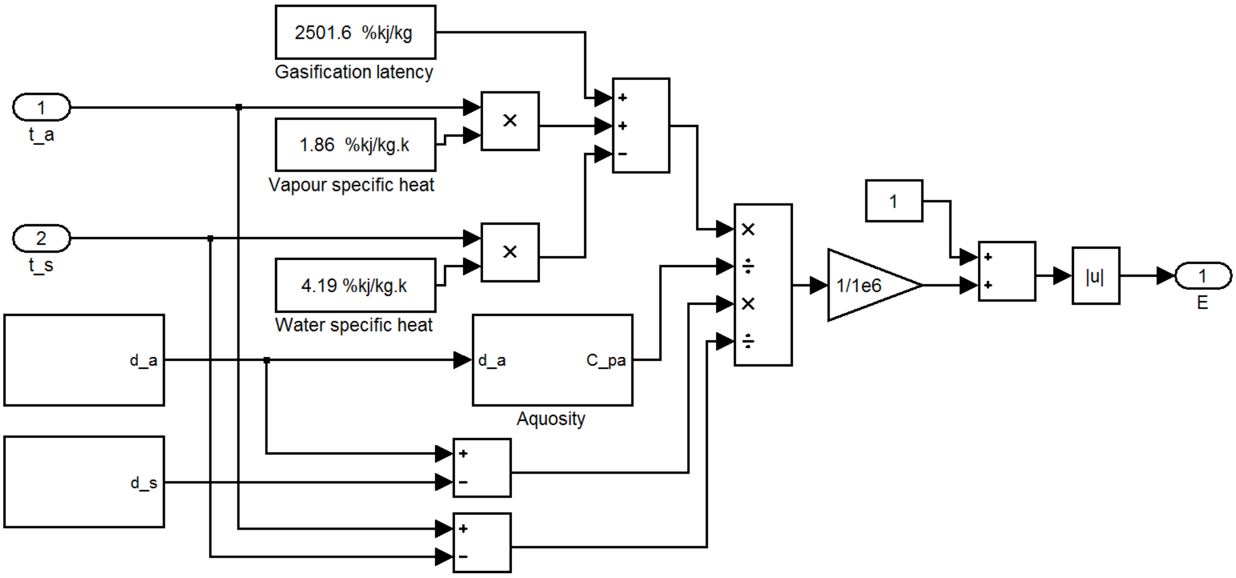


Figure 3.35: ***Wet-coefficient***

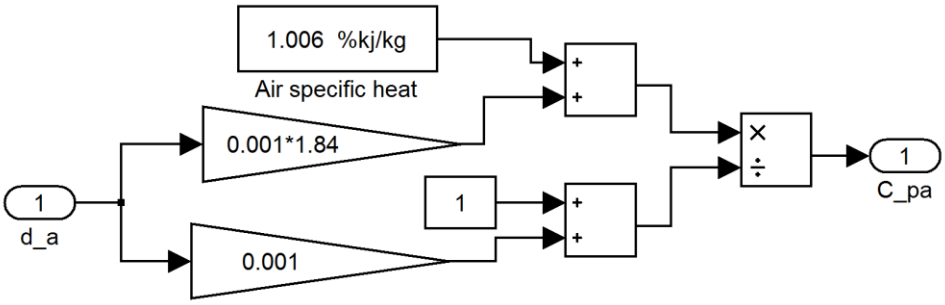


Figure 3.36: ***Subsystem for Aquosity***

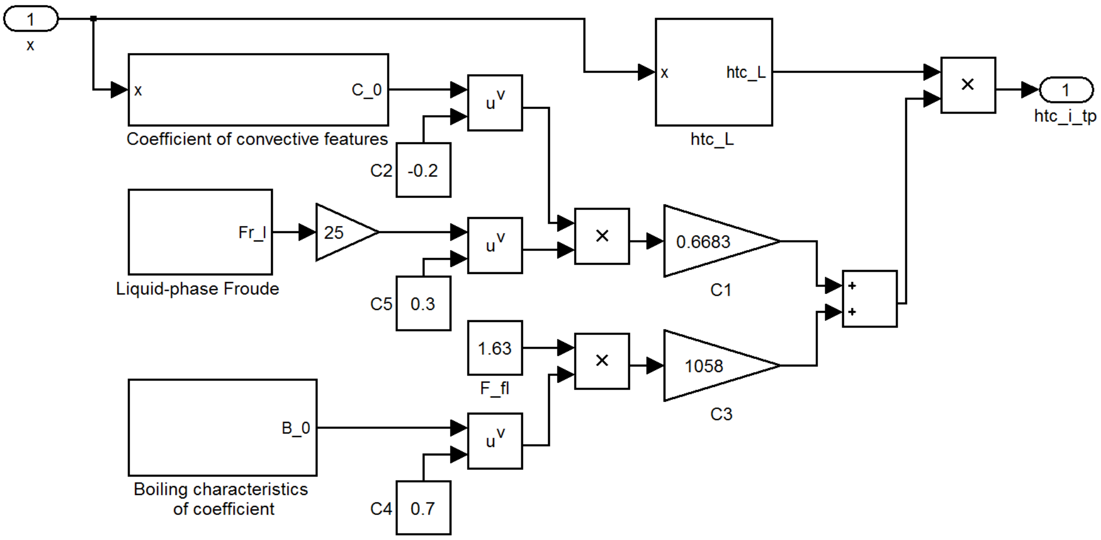


Figure 3.37: ***Kandlikar equation for TP heat transfer coefficient***

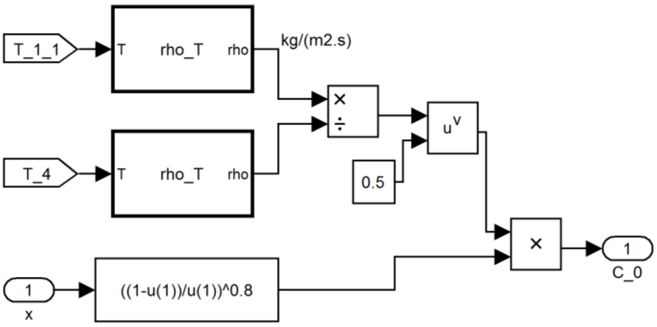


Figure 3.38: ***Subsystem for coefficient of convective features***

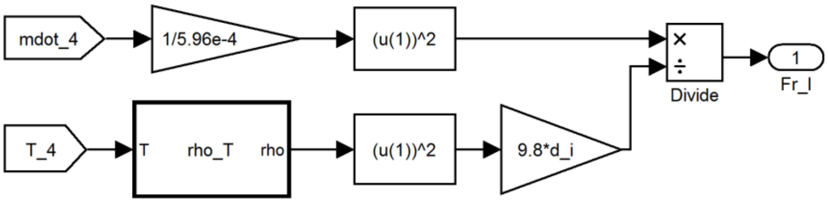


Figure 3.39: ***Subsystem for Liquid-phase Froude***

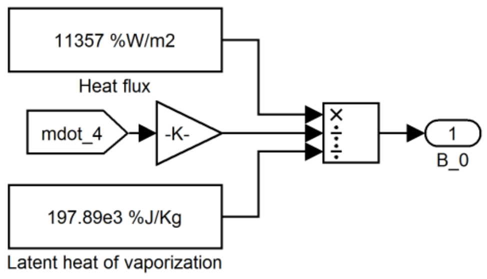


Figure 3.40: ***Subsystem for boiling characteristics***

3.7. Cooling circuit modelling overview

As soon as each part has been achieved, simply refers to the framework structure of modelling in *Figure 3.4*, mutual-impact parameter link each part, combine the whole subsystem into one model, *Figure 3.41* shows the cooling circuit modelling in MATLAB/Simulink. Every parameter can be observed by a 4-channel scope at each end of the components. A subsystem of “Engine” is based on the model of M. Konz, in A Generic Simulation of Energy consumption of Automobile Air conditioning systems. The New European Driving Cycle (NEDC), which provides the values for the driving speed and selected gear. NEDC is a combination of four repeating cycles with low driving speeds (city cycles) and one with a higher driving speed (highway cycle). Since then an off-line test has been undertaken. Testing and verification on this off-line modelling should be the next step in the process.

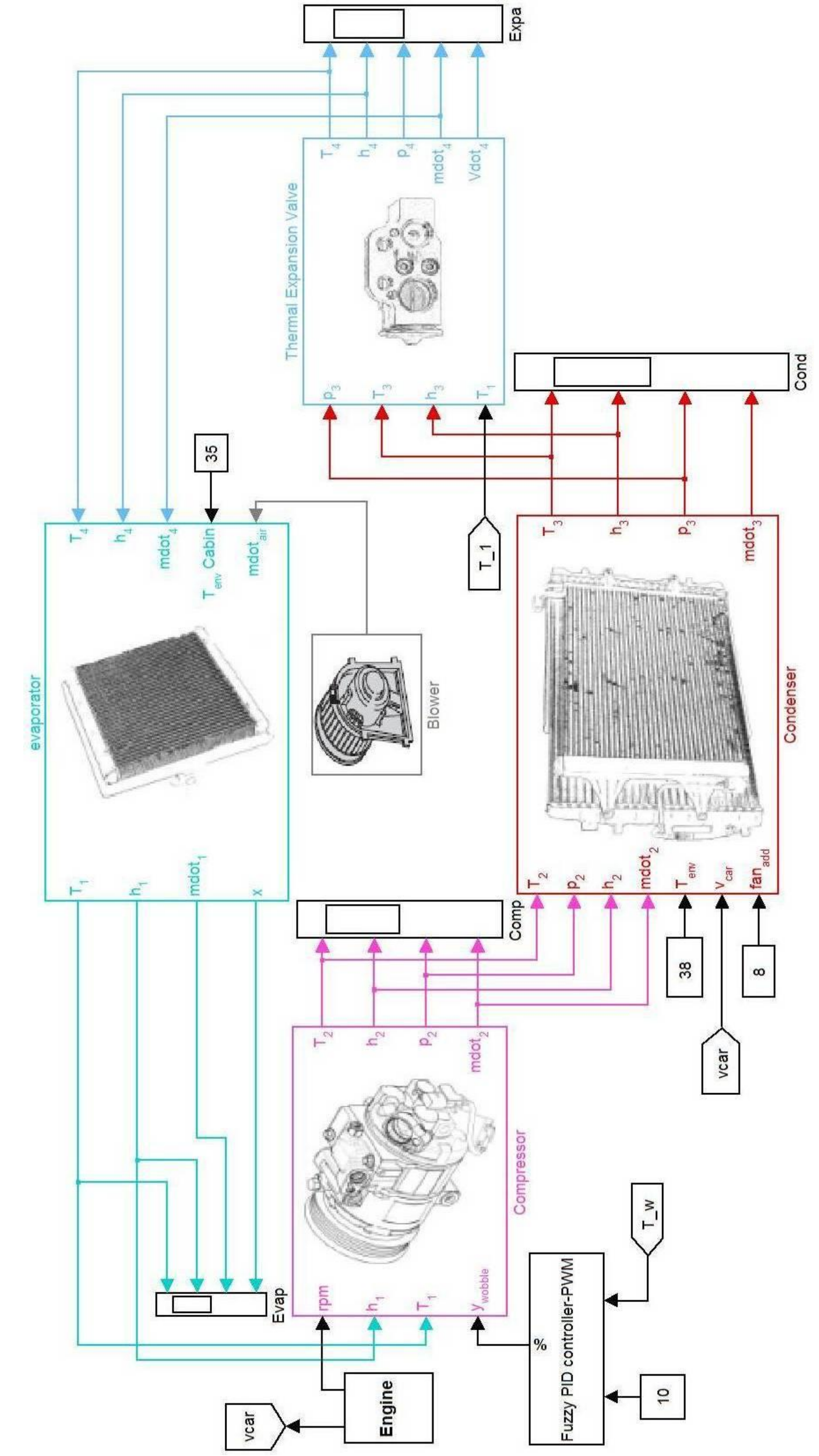


Figure 3.41: ***Cooling circuit modelling***

-Chapter 4

Control strategy and modelling verification

This chapter discusses the experimental testing equipment and the resulting data, which is used for model validation and something factor involved on instability of AC system with a VDC will investigated on the dynamic modelling and avoided by control strategy.

4.1. Test bench

All of experimental data coming from one of my teammate Marin Freoehlich who was designed a test stand at the IFBW (Institut für Fahrzeugbau Wolfsburg) in Germany. The schematic of the test rig essentially refers to the *Figure 4.1*

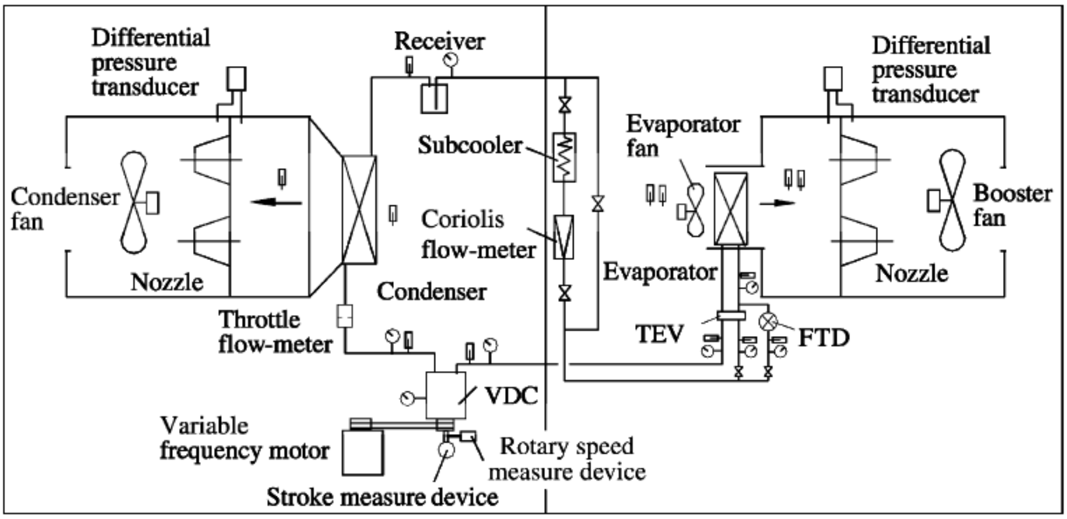


Figure 4.1: ***Schematic of test rig***

The air conditioning system that is used to construct the test rig is from the Polo 9N series. Each standard part from the supplier Volkswagen, it is of great importance that the original parts are used to simulate the air conditioner as close to the actual system as possible and rule out errors that could occur from using non original parts.

It was also possible to obtain results from real test vehicle by CAN-bus, but the information only comes from existing sensor assembled in the vehicle. The combination of the real CAN-messages and the test bench results could utilized to summary control strategy. *Figure 4.2* shows the CAD-model of the complete test bench and real photo. The recording hardware is supplied by National Instruments. The data is fed to the computer by the interfaces on the PCMCIA.

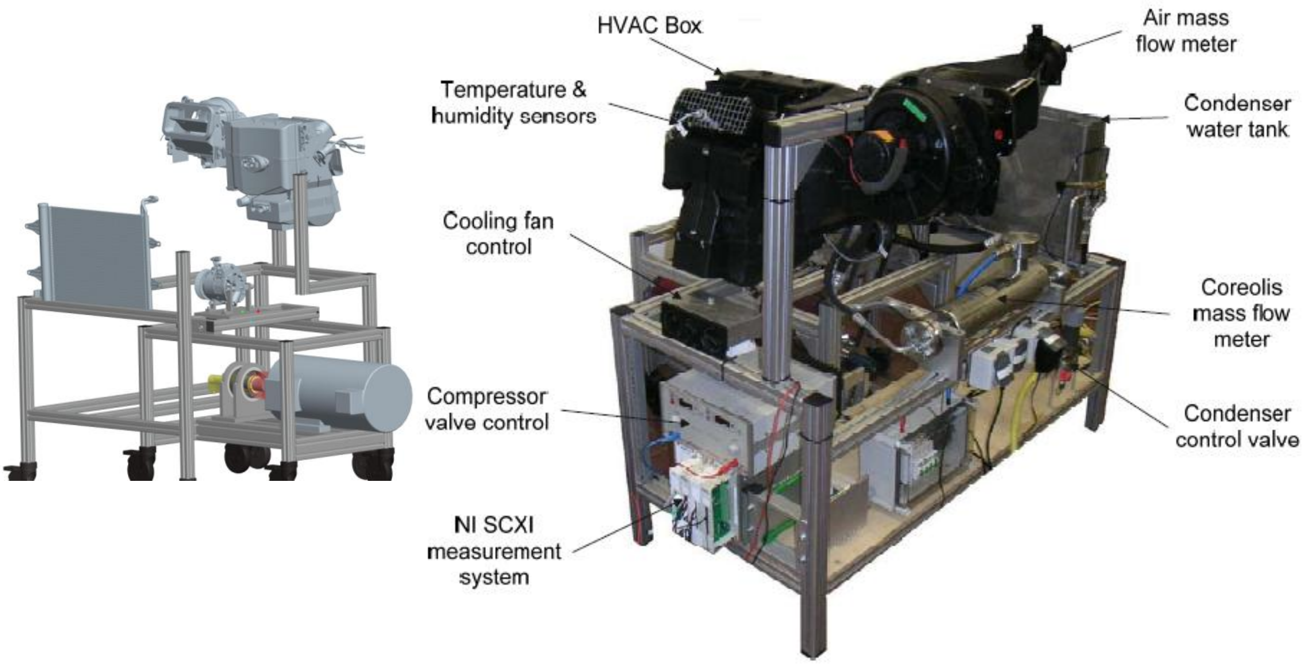
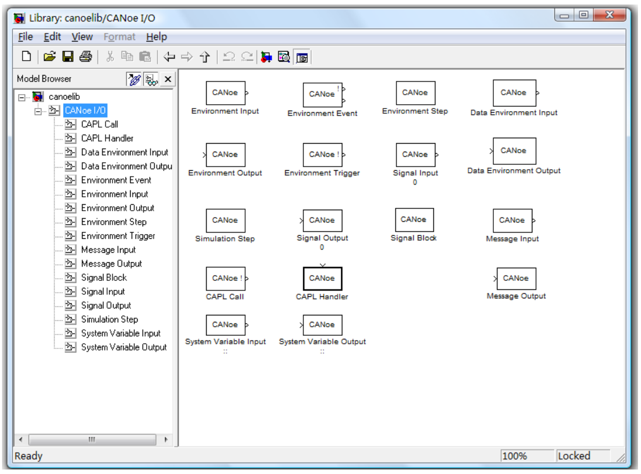


Figure 4.2: ***CAD-model of test rig and real photo [3]***

4.2. Polo CAN-bus

The real time data acquisition (DAC) on the vehicle, Canoe which supply by Vector Company is the professional option for the vehicle CAN-bus, it is used to create simulation models that emulate the behaviour of the ECUs. Over the course of ECU development these models serve globally as a foundation for analysis, testing. CANoe AddOn MATLAB @ interface is the document which describes the CANoe /

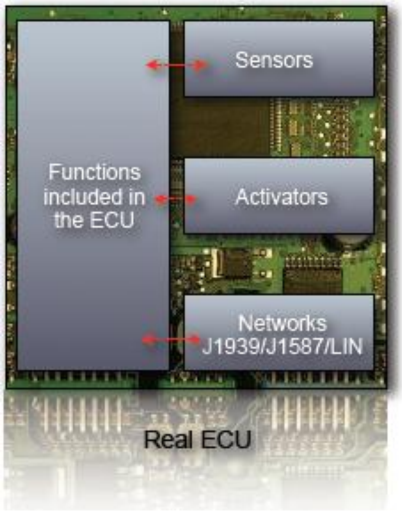
MATLAB@ interface. The purpose of this interface is to extend CANoes' node modelling capability by adding the strength of the MATLAB/Simulink environment. After properly software installation process, CANoe I/O library is attached into the Simulink library by a collection of S-Functions blocks shows in *Figure 4.3*

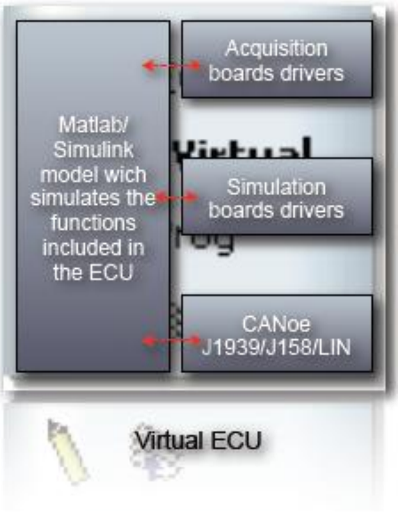


Notes: MATLAB needs to be associated with C compiler in order to generate the dll files for the HIL mode. The following compilers are currently installed:
Microsoft Visual C++@6.0

Figure 4.3: ***CANoe I/O library***

The Blocks are used as sources or sinks for Simulink models. They provide the current value of the according CANoe element, and the block set is a very easy way to extend MATLAB/Simulink models with the ability to communicate with real bus devices. CANoe provides network management and interaction layer functionality so that the models are focused on the application logic and are separated from hardware access functions. After understanding the structure of ECU, as the Figure shows the function block of real ECU corresponding with Virtual node on CANoe.




Acquisition boards drivers
Matlab/ Simulink model wich simulates the functions included in the ECU
Simulation boards drivers
CANoe J1939/J158/LIN
Virtual ECU

Pick up all of the useful sensor value from the vehicle CAN-bus, and make a real data acquisition under MATLAB environment, all of the CAN-bus messages recorded in the Workspace for farther analysis. Due to the database (dbc .file) is highly confidential information, so there is no screenshot on CANoe DBC files, only by Simulink modelling.

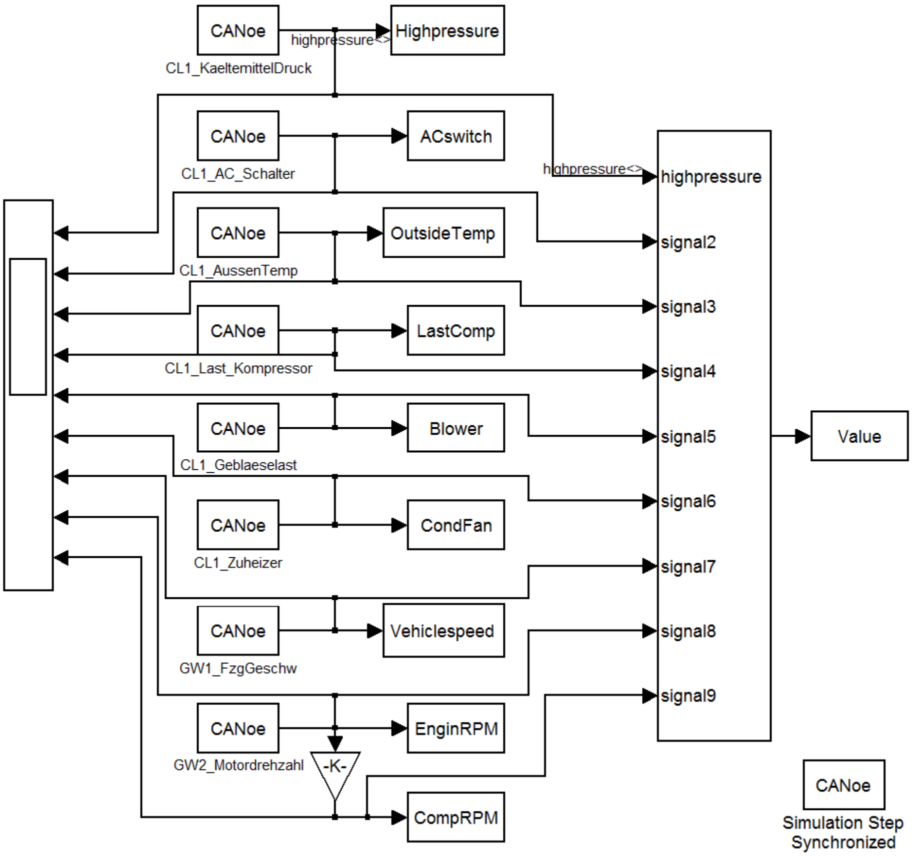


Figure 4.4: ***DAC modelling interface with CANoe***

4.3. Data verification

The compressor that is used for Polo system is a Variable Displacement Reciprocating Compressor (VDRC) that is manufactured by Sanden and is operated at rotational speeds of between approximately 800 and 6000 revolutions per minute. This compressor is operated without a clutch system and only depending on the piston stroke control.

The pressure at compressor discharge end (point B) during the rpm of compressor changing, *Figure 4.5* shows the refrigerant high-pressure changing in the cycle in the real vehicle environment, compare this value on the real DAC modelling with the value from compressor modelling in *Figure 3.8*, it has too much difference. Because some of the reason is condensing pressure has not considering on the high-pressure cycle of refrigerant or the data provided modelling for compressor pressure cycle from VW documentation is only theoretic value. So rest of data validation needs to do it by individually parts or using measured suction and high pressure for father model testing.

In according to *Figure 4.6* theoretical refrigeration cycle in pressure-enthalpy chart, there are four sharp peaks: point A is the refrigerant state at evaporator outlet (or compressor suction end), point B is the refrigerant state at compressor discharge end (or condenser inlet), point C is the refrigerant state at throttling device inlet (or condenser outlet), and point D is the refrigerant state at throttling device outlet (or evaporator inlet).

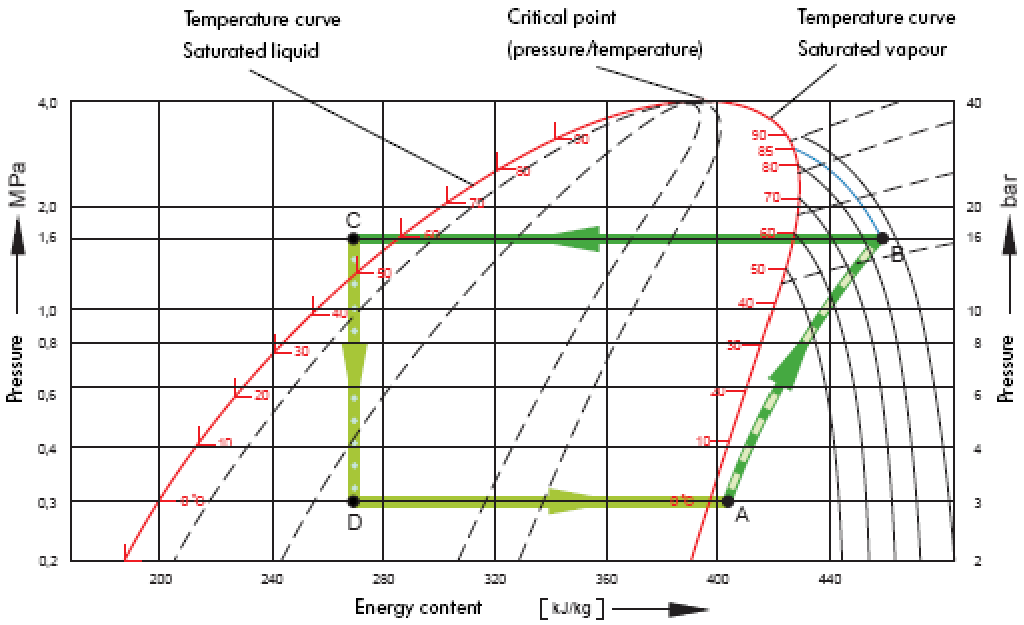


Figure 4.6: ***Theoretical refrigeration cycle***

The refrigerant at compressor discharge end is always superheated vapour whose degree of superheat is usually very high, far away from the saturation vapour line. The refrigerant at the throttling device outlet is always two-phase state. It is almost impossible to make phase transition for these two sharp peaks, so the large external disturbance cannot cause larger per-disturbance for the refrigeration system through these two sharp peaks. However, the refrigerant at the throttling device inlet is near the saturation liquid line, and the refrigerant at the evaporator outlet near the saturation vapour line, and both saturation line are at the vapour-liquid junction. The normal working condition of the refrigeration system is that liquid refrigerant at the throttling device and superheated vapour at the evaporator outlet. For the large external disturbance, if the refrigerant at the evaporator outlet does not pass the saturation vapour lien and keeps superheated vapour state, the pert-disturbance to the refrigeration system is little. But once the liquid refrigerant at the throttling device inlet passes the saturation liquid line, or the saturation vapour at the evaporator outlet passes the saturation vapour line due to the large external disturbance, there is a phase change for the refrigerant at the throttling device inlet or at the evaporator outlet, which makes a large disturbance to the system. For the AC system with a

Variable Displacement Compressor (VDC) and Thermo Expansion Valve (TEV), an overshoot of thermo expansion valve due to the large disturbance and time lag of its operation to the degree of superheat at the evaporator outlet due to the transition process of thermal bulb cause the evaporator and expansion value control loop and then the system to go into hunting, the large disturbance cause by difference of the refrigerant mass flow rate. To make sure the AC system keep in a stable condition, first of control is the mass flow rate by control the VDC.

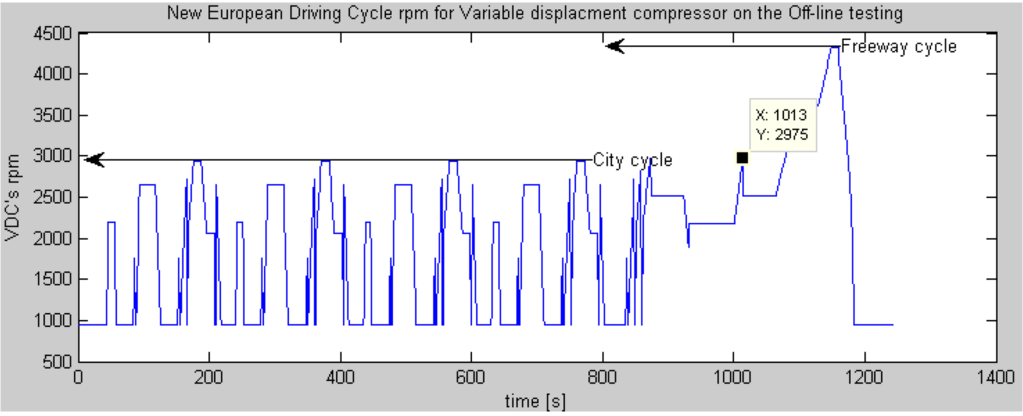


Figure 4.7: ***Simulated NEDC- cycle for compressor***

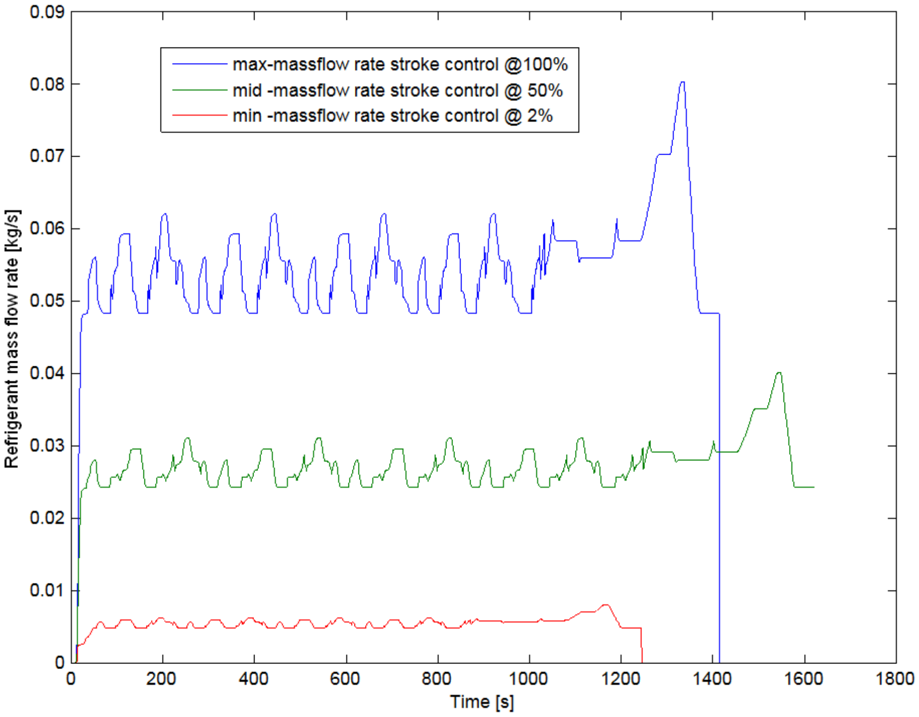


Figure 4.8: ***Simulated Refrigerant Mass flow***

Figure 4.8 shows the Eq.3.10 and 3.11 calculated mass flows in the compressor corresponding with the rpm from NEDC cycle supplied for compressor. Due to data are unavailable, no decision which course of rather correspond to reality. This selection needs to validate by other research, Yuefei Miao who wrote the thesis about “Characteristics of Swash Plate Variable Displacement Compressor and its Automotive Air conditioning system”, on his experimental investigations, the test on VDC during the maximum stroke control, the relationship between refrigerant mass flow and suction pressure shows on *Figure 4.9* right side rotary speed keep constant at 2000 rpm, compare with *Figure 4.8*, the mass flow rate can be validated.

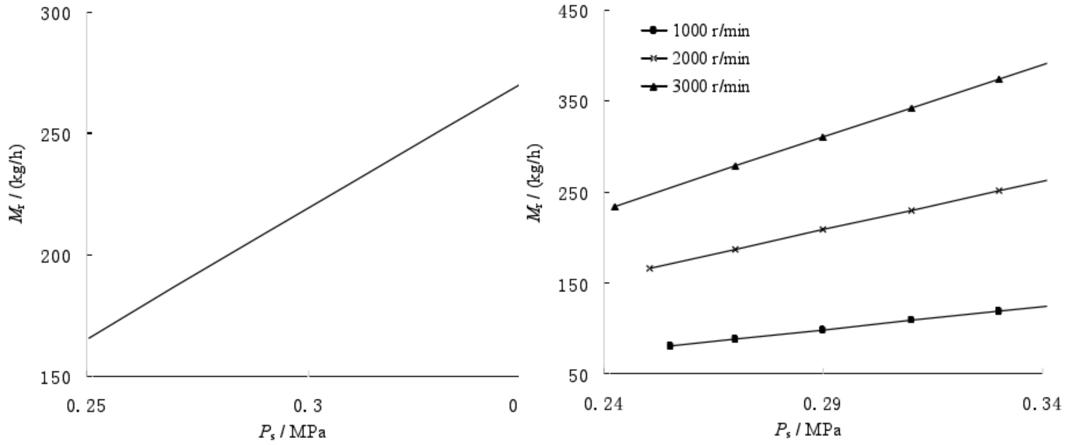


Figure 4.9: ***Measured Refrigerant Mass flow***

Stationary measuring result from the test beach could use for two heat exchanger modelling validation. The temperature and pressure sensor located at inlet and outlet of heat exchanger, basic function reference on *Figure 4.1*, except there is only one throttling devices installed in the test beach, the parameters controlled in the test beach are the compressor rotary speed, and air conditioning load. Variable frequency motor driven and adjusting compressor; the air conditioning load is adjusted by the air flow rate through the evaporator and controlled by the blower, so the measuring result categorized by different rpm of compressor and level of evaporator’s blower. Using Matlab load those xls.file, simply find out the pressure at inlet and outlet of compressor almost same, due to all of the test is a stationary measuring without any condensing pressure drop, so under lowest rpm condition test result is the best of enthalpy analysing and validating. *Figure 4.10* shows Matlab plot the xls.file for pressure data, sent those values in compressor Simulink modelling to test condenser and evaporator modelling.

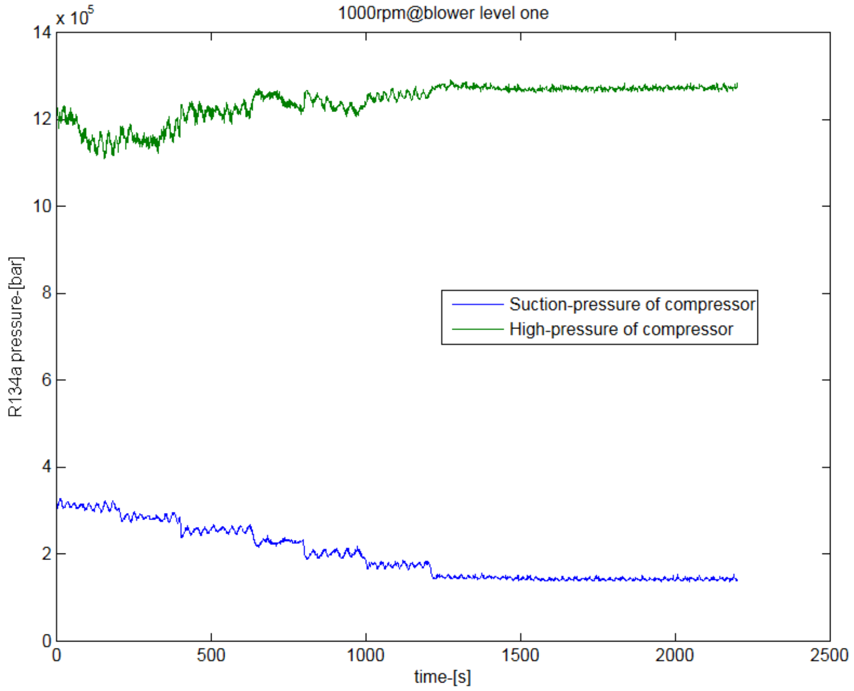


Figure 4.10: ***Measured inlet and outlet of compressor pressure from test beach***

Due to the physical parameters of the system components are not given, all of the parameters used in the simulation are shown in the *Table 4.1*. The effective cross-sectional areal was originally calculated using the hydraulic diameter. The selected value for cross-sectional area was calculated using the measured internal volume of the heat exchanger divided by the length of fluid flow, for all of heat transfer coefficient was calculated by given HTC equation in chapter 3, the value shows on the *Table 4.1* is the typical one, later discussion will focus on how to determinate the heat transfer coefficient.

Component	Parameter	Units	Value	Comment
Condenser	Specific Heat	[KJ/KG/K]	0.879	Aluminium
	Exterior Fluid Specific Heat	[KJ/KG/K]	1.007	Air
	Hydraulic Diameter	[m]	2.90E-03	Calculated
	Internal Surface Area	[m^2]	4.00E-01	Calculated
	External Surface Area	[m^2]	7.90E+00	Calculated
	Total Fluid Flow Length	[m]	4.80E-01	Measured
	Louver height	[m]	9.80E-03	Measured
	Louver width	[m]	4.80E-01	Measured
	Fin height	[m]	9.00E-03	Measured
	Louver space	[m]	1.00E-03	Measured
	Two-phase region HTC	[KW/m^2/K]	1.97E+02	Calculated
	Superheat region HTC	[KW/m^2/K]	6.58E+01	Calculated
	Subcooled region HTC	[KW/m^2/K]	1.63E+01	Calculated
	Exterior Fluid HTC	[KW/m^2/K]	1.26E-02	Calculated
	Number of Parallel tubes	[-]	3.60E+01	Measured
Evaporator	Specific Heat	[KJ/KG/K]	0.879	Aluminium
	Exterior Fluid Specific Heat	[KJ/KG/K]	1.007	Air
	Hydraulic Diameter	[m]	3.80E-03	Calculated
	Internal Surface Area	[m^2]	8.00E-01	Calculated
	External Surface Area	[m^2]	4.60E+00	Calculated
	Total Fluid Flow Length	[m]	2.10E-01	Measured
	Louver height	[m]	8.00E-03	Measured
	Louver width	[m]	2.10E-01	Measured
	Fin height	[m]	8.50E-03	Measured
	Louver space	[m]	7.50E-04	Measured
	Two-phase region HTC	[KW/m^2/K]	1.57E+03	Calculated
	Superheat region HTC	[KW/m^2/K]	1.89E+02	Calculated
	Subcooled region HTC	[KW/m^2/K]	0.00E+00	Calculated
	Exterior Fluid HTC	[KW/m^2/K]	3.87E-03	Calculated
	Number of Parallel tubes	[-]	42	Measured

Table 4.1: ***Parameter for Model Validation***

The available inputs of condenser air flow rate, expansion valve opening, evaporator air flow rate. For model validation the outputs of pressure and enthalpy on evaporator and condenser, evaporator superheat, condenser sub-cool are compared. Calculated refrigerant mass flow rate (0.018kg/s or 64kg/h) is ultimately matched for real at 1000rmp maximum stroke. Problem is the relationship between mass flow rate and the suction pressure, in the *Figure 4.9* during the suction pressure decrease, mass flow should also decrease, but the in the *Figure 4.11* the relationship is opposite.

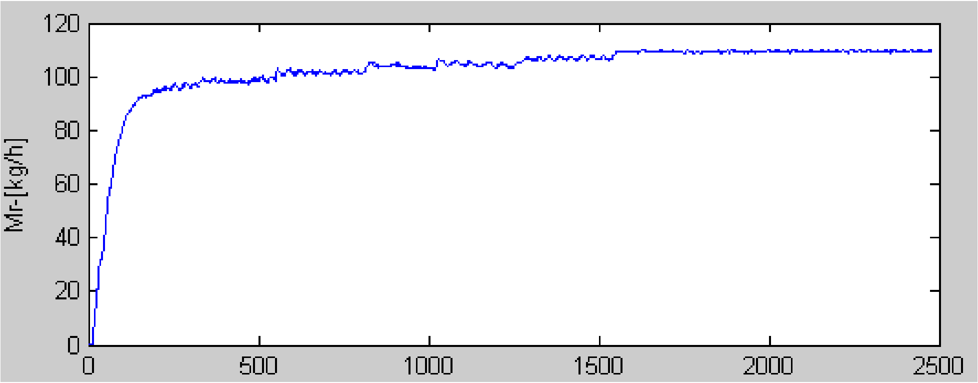


Figure 4.11: *Simulated Mr @ 1000rpm*

The enthalpy dominated the temperature very precisely, especially when the pressure have been confirmed. From the *Figure 4.12 and 4.13* temperature curves are matching perfectly. By the modelling both of heat exchanger, outlet enthalpy can be calculated more accurate. So cooling circuit modelling can be represented stationary measuring result very well. Due to the suction and high pressure cycle is unable to simulated, dynamic performance of cooling circuit modelling still need to improve.

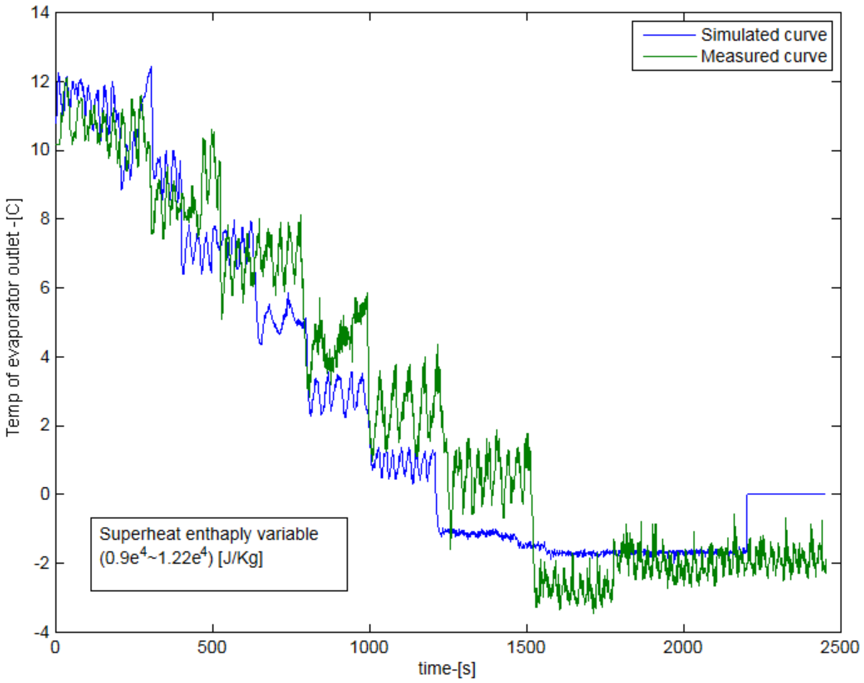


Figure 4.12: *Simulated and Measured outlet temp of evaporator*

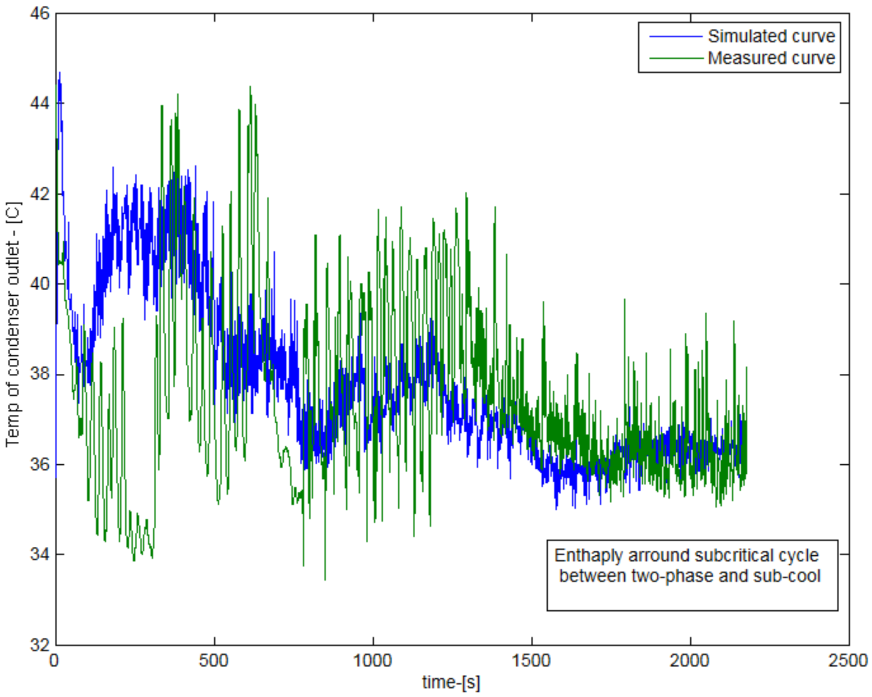


Figure 4.13: *Simulated and Measured outlet temp of condenser*

Figure 4.12 and 4.13 indicate outlet of temperature for two heat exchangers, Enthalpy outlet of condenser around subcritical cycle between two-phase and sub-cool region. It means at 1000 rpm of compressor, maximum refrigerant mass flow rate is not enough to make R134a absolute condensing, according to the whole cooling cycle, if the energy radiated insufficient on the condenser, and insufficient mass flow rate to drive the refrigerant evaporating properly, so stroke control for compressor is normally keep maximum value at lower rpm, in order to provide enough mass flow of refrigerant inject to heat exchanger. However, while the vehicle speed up or the condensing fan started exterior air fluid heat transfer coefficient increasing show in *Figure 4.14*. Correspondingly the condensing pressure has a hunting period, finally as the pressure reducing, the temp and enthalpy decreases as well. Outlet enthalpy reaches to the sub-cooled region of R134a, makes cooling cycle a better performance.

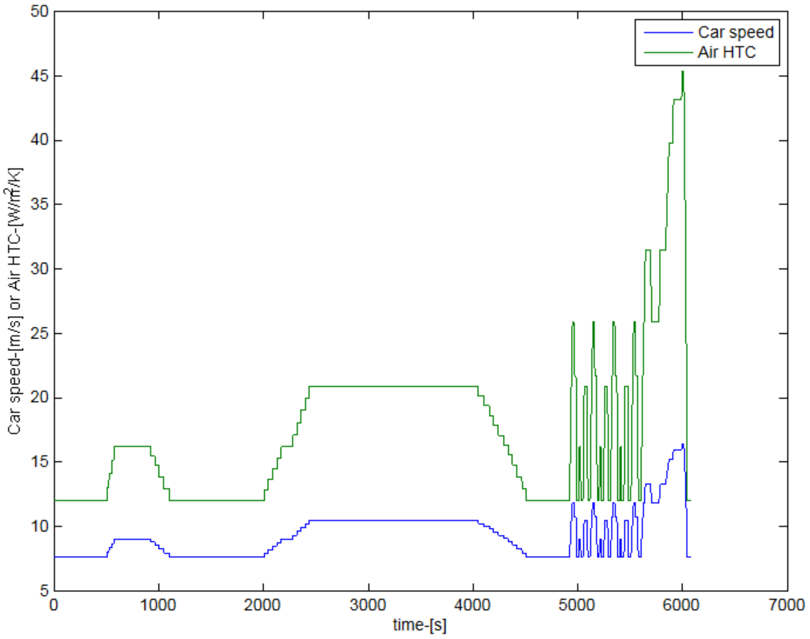


Figure 4.14: ***Simulated Air HTC vs. Car speed***

4.4. Data analyzing

Last paragraph belongs to author's understanding refers the cooling circuit modelling, unfortunately this modelling could not test under dynamic characteristics for vehicle AC system, a excerpt needs to be used for validating my opinion, it is still mention on the Yuefei Miao's dissertation, a serial test on adjusting characteristics of VDC. VDC test beach which installed in Tsinghua University, school of Architecture. Following three measuring schematic is also basic on the *Figure 4.1*, those three factors have to be proofed, which are the most of effect stroke control on the compressor. Keep relative constant mass flow rate so that the cooling power match to the Air-con load. Hereby the control strategy conforms how to adjust those characteristics of VDC and summarize the variation is significant.

Notes: some of the measuring curve has been zoomed and parallel shift in *Figure 4.15~4.16*

4.4.1. Variable rotary speed adjusting test on VDC

VDC test beach running at variable rotary speed and all of sensor value shows on the *Figure 4.15*. Initially fully stroke working condition of VDC running at 600 rpm as usual, and then increase the rotary speed to 3600 rpm, finally make the rotary speed back to 600 rpm, during the rotary speed is changing and keep the rest of parameters unchanging, make a record for each sensor after the system stabilized.

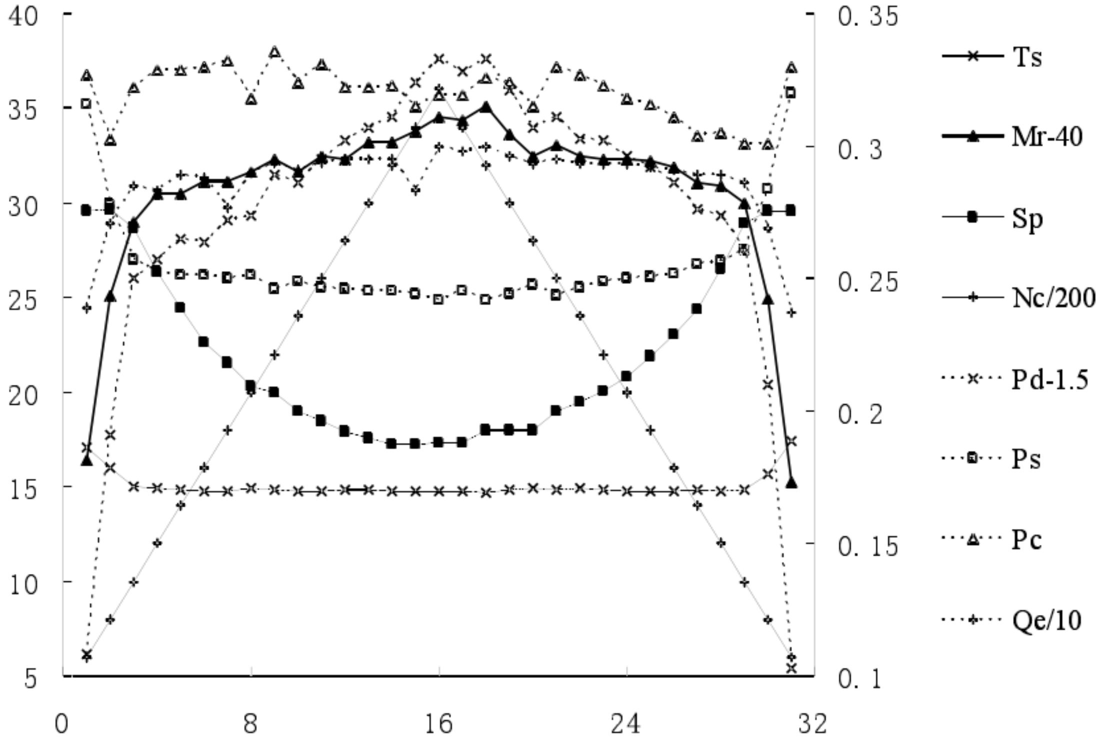


Figure 4.15-a: ***Measuring point begins at the Pd @ 16bar***

In the *Figure 4.15-a* rotary speed start at 600 rpm, during the rpm speed up, suction pressure (Ps) reducing, when changed suction pressure is so small that is impossible to make stroke distance (Sp) shifted, wherefore the mass flow rate (Mr) and cooling power (Qe) rapidly rising, this moment characteristics of VDC seems as a Fixed Displacement Compressor (FDC)'s. As Ps decreasing once more and the stroke distance shifted and pressure crate (Pc) enhanced, the characteristics of VDC returns to be adjustable. During 1000 rpm to 3600 rpm and back to 1000 rpm, according to

Fig.4.13 stroke distance shifted start at 1 to 0.58 and then back to 1. Qe is essentially keeping at 2.8~3KW, Mr holding at 69~75 kg/h (0.019~0.021kg/s), Ps (2.42~2.6 bar), Suction temperature (Ts) of compressor at 14.75~15 ℃. Due to condenser pressure (Pd) keep constant at 16bar, so Ps is changing too small.

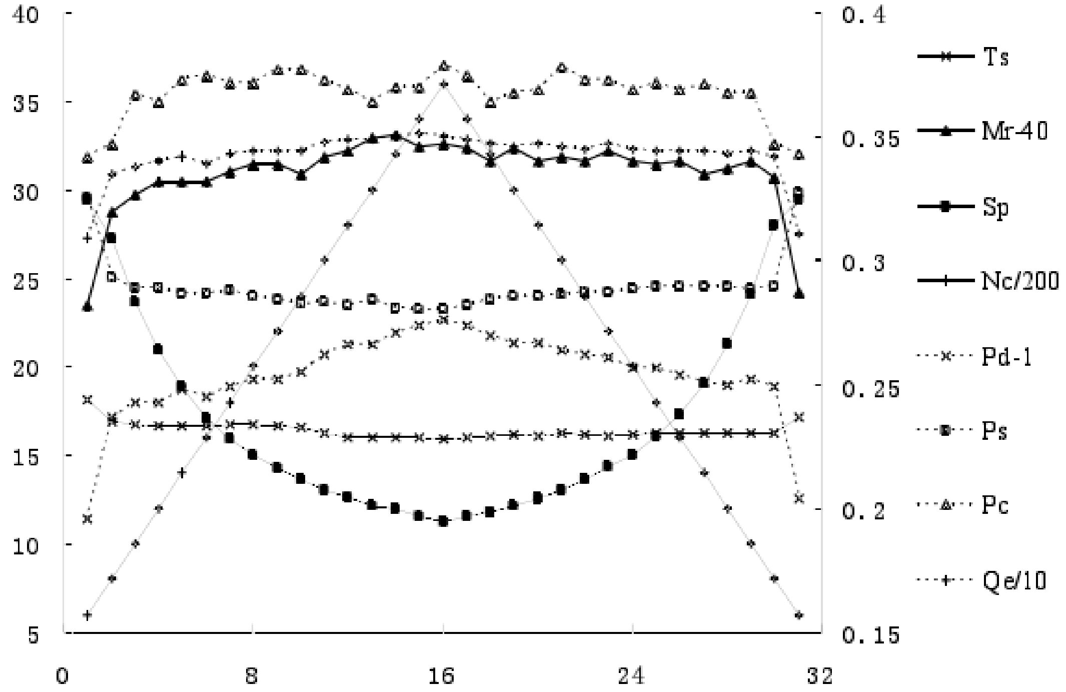


Figure 4.15-b: ***Measuring point begins at the Pd @ 12bar***

In the *Figure 4.15-b*, all of measuring point begins at Pd @ 12 bar, comparison with those two measurements, as rotary speed rising, VDC start shifting stroke more earlier than *Figure 4.15-a*.

From those two test results, VDC represents double characteristics, under same Air-con load condition, stroke control variety between 0.58~1 to make sure the refrigerant mass flow rate keeps corresponding constant.

4.4.2. Variable vehicle velocity adjusting test on VDC

This test simulated as real vehicle condition for VDC system, when a vehicle start running, the rotary speed and road speed changing synchronously, air flow of condenser increasing during the velocity speed up, so in the test beach slightly increasing the frequency of motor driving the compressor, and also for the condenser's fan, in order to enhance the air flow rate as enhance the rotary speed.

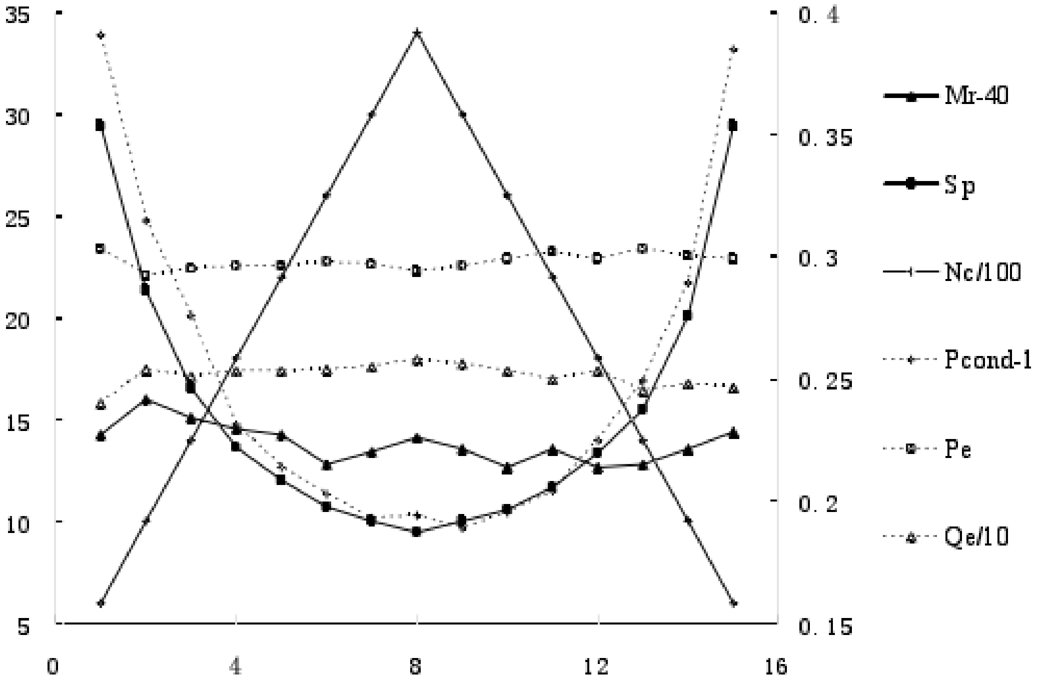


Figure 4.16-a: ***Measuring point, rotary speed (600~3400~600 rpm)***

In the *Figure 4.16-a*, input condition of air exit temp behind evaporator side at 27 ℃, humidity is 50% and keep the blower as same level, air flow rate is 300 m^3/h. air temp forward condenser side at 38 ℃.

When the vehicle speed up, rotary speed and air flow rate for condenser rising, condenser pressure drop, because at the evaporator side nothing changed for air flow rate by vehicle speed up, so stroke adjusting reduced shift distance; when the vehicle speed down, condenser pressure back to expected value, stroke increasing. During the measurement, refrigerant mass flow rate, cooling power and evaporator pressure keep in a short range. Compare with *Figure 4.16 and 4.15*, the difference are the condenser pressure and evaporator pressure.

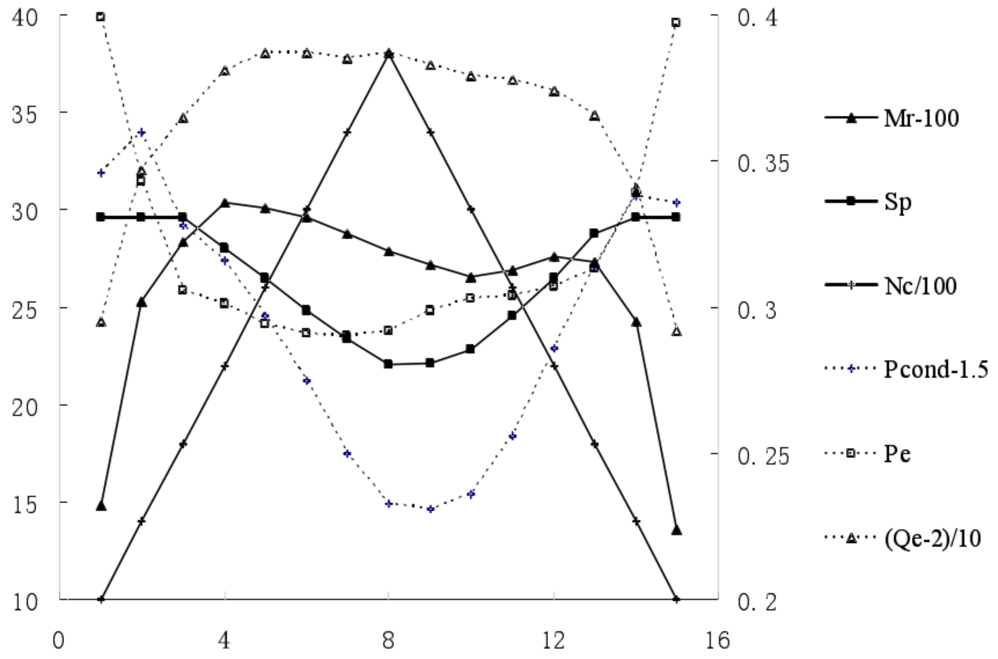


Figure 4.16-b: ***Measuring point, rotary speed (600~3800~600 rpm)***

In the *Figure 4.16-b*, input condition of air temp behind evaporator side at 35 ℃, humidity is 70% and keep the blower as same level, air flow rate is 500 m^3/h. air temp forward condenser side at 40 ℃.

Due to the Air-con load is great than *Figure 4.16-a*, evaporator pressure is higher than previously, when the vehicle speed up, evaporator pressure reducing, stroke distance shift quite short. As the rotary speed rising, refrigerant mass flow rate and cooling power increased. Compare with those two conditions, when the Air-con load rises up, refrigerant mass flow have to be increasing to keep a better cooling performance. Next test will proof this point.

4.4.3. *Variable Air-Con load adjusting test on VDC*

In this measurement, compressor keeps the rotary speed at 2000 rpm, air temp forward condenser side at 40 ℃, air flow rate for evaporator keep as 300 m^2/h, initial condition of air exit temp behind evaporator side at 41 ℃ and humidity is 50%, and then cooling around to 20 ℃. Second measurement release air-con load, which make the air temp behind evaporator side return backs to 41 ℃.

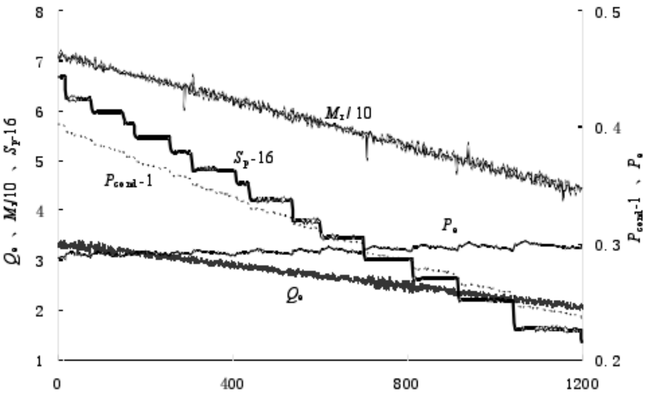


Figure 4.17-a: ***Decreases Air-con load***

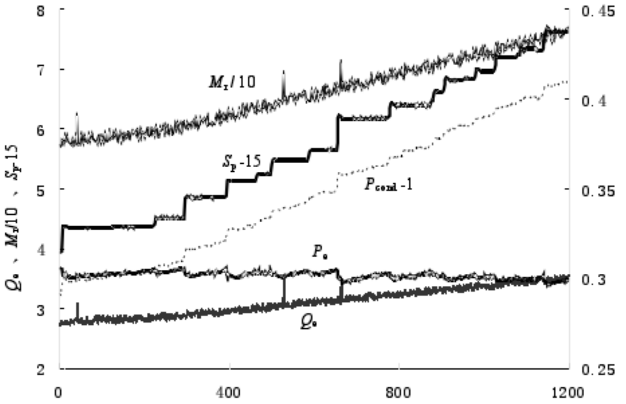


Figure 4.17-b: ***Increases Air-con load***

From *Figure 4.17-a*, Air-con load decreasing, condenser pressure, stroke and cooling power all reduced, evaporator pressure keeps in a short arrange of variety. Stroke shift decreasing like a step-down pulse, each step division is quite small, so there is no effect the variable of mass flow rate and cooling power. In the *Figure 4.17-b* processing, all of parameter changes as opposite trend.

4.5. Control strategy

Refers to the measurement result for characteristics of VDC, all of parameter have been conform the relationship during the dynamic testing. Variable rotary speed, variable vehicle velocity and variable air-con load adjusting test indicate the cooling circuit modelling analyse result is quite reasonable. Distinguish between VDC and FDC, VDC provide a suitable adjustment stroke control which deal with different air-con load and variable speed condition, but the all AC system running at VDC performance is nonlinear control, in order to achieve the changing of cooling power and refrigerant mass flow rate and suction pressure stability.

According to a nonlinear control system, Fuzzy logic control is the best solution for it, another advantage of fuzzy controller is no need control object modelling, because of cooling circuit modelling is disable for dynamic performance as so far. In the next chapter, a summarization of stroke control which is based on the measurement result in previous section will be utilized for fuzzy rules writing.

-Chapter 5

Fuzzy controller modelling for VDC

This chapter discusses the control strategy and Fuzzy logic controller implementation, according to the experiment result in last chapter, control algorithm for VDC stroke control only can be fixed at closed-loop for fuzzy controlling. HIL controller design is based on the existing sensors and actuators.

5.1. Polo electronically controlled AC system

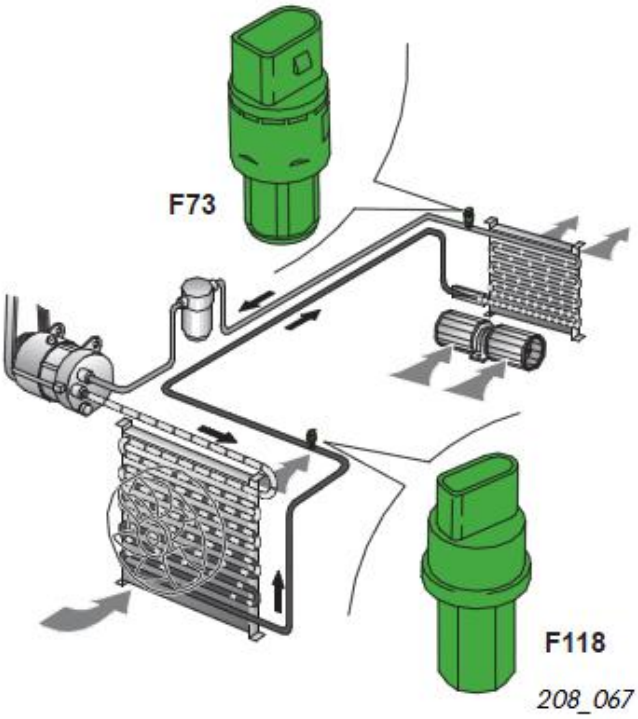
HIL controller design must consider the existing sensors and actuators. Due to the limitation of researching time, there is no more extra sensor refit on the vehicle. So understanding of the existing sensor is important to determinate feedback factor for the closed-loop control.

Sensors	Actuators
Temperature sensor Dash panel	Foot well/defroster flap positioning motor
Dash panel Blower	Central Flap positioning motor
Ambient temperature sensor	Temperature flap position motor
Fresh air intake duct temperature sensor	Air flow flap positioning motor
Foot well vent temperature sender	Air recirculation flap positioning motor
Air conditioner pressure switch	Fresh air blower Control unit, Blower
Auxiliary signals: Road speed signal Engine speed signal ‘time parked’ single	Auxiliary signals: engine control unit control unit with display unit in dash panel insert
Radiator fan thermo switch	Radiator fan and Compressor control unit

Table 5.1: ***Electronically controlled components***

The temperature is regulated evenly at the left and right–hand side of the passenger cabin as show *Table 5.1* using the Polo as an example. Hardware configuration shows in *Table 5.1*, location of sensors placed response for the functions mapped in the cooling cycle.

Figure 5.1: ***Position of high pressure sensor***



In Polo AC system, High pressure switch F118 in *Figure 5.1*, mainly function of this device to make sure when the pressure exceeds approx.30bar and switches off the compressor, due to VDC without magnetic clutch, so stroke control for compressor should turn to lowest value. As this Fig shows the position of pressure sensor, this value representing compressor pressure plus condensing pressure drop. Last chapter have mentioned during the vehicle speed up, even different radiator fan speed controlled by thermo switch, high pressure cycle can be defined by this value under dynamic variable. Auxiliary signals and high pressure signals are all available by Polo CAN-bus. Following list the available CAN-massage involved on HIL controller modelling.

- ✓ Rotary speed [rpm]– Engine control unit
- ✓ High pressure [Bar] – Climatronic control unit
- ✓ Compressor load [Nm] – Climatronic control unit
- ✓ Blower load [%]– Climatronic control unit
- ✓ Ambient temperature [C]– Gateway

From control strategy summarized last chapter, variable rotary speed, variable vehicle velocity and variable air-con load adjusting test indicated the relationship between pressure and rotary speed corresponding to stroke control. During the rotary speed increasing under same cooling pressure, the stroke distance reducing; during the cooling pressure increasing under same rotary speed , the stroke distance reducing as well. VW documentation “Funktionshplan PQ24 Kima- Halbautomat” also mention by Kompressor – Drehoment, which defined by compressor torque load. Using MATALB by interpolate function, *Figure 5.2* shows as following.

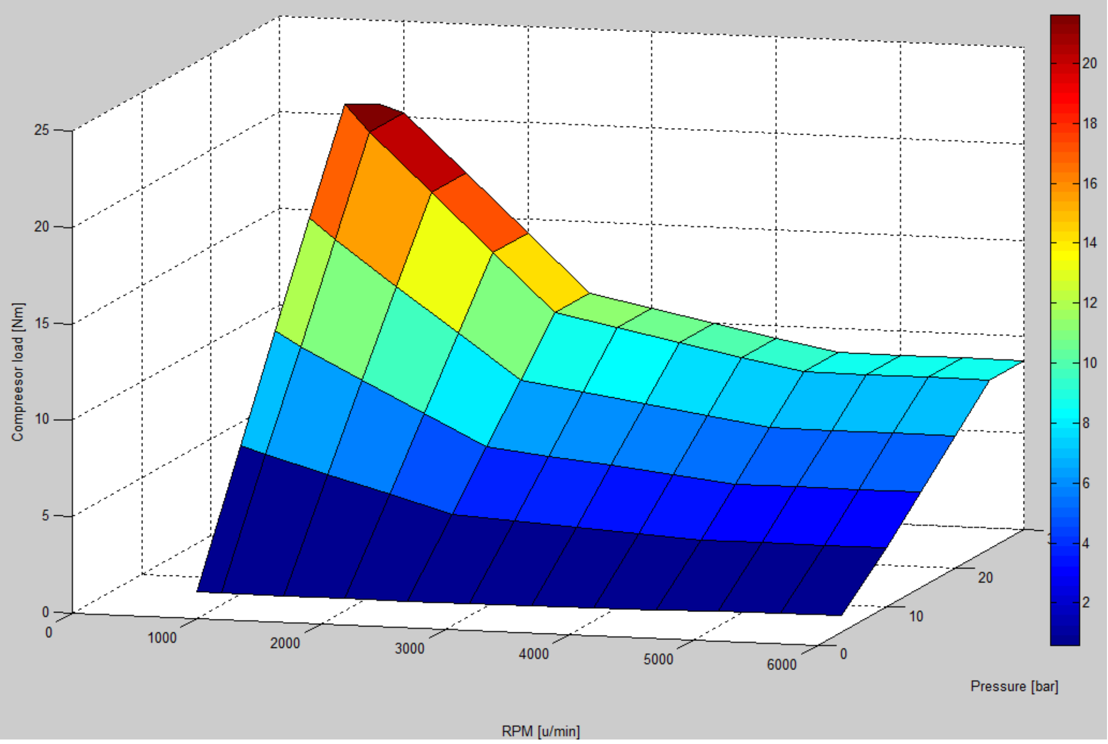


Figure 5.2: ***Drehomentkennfeld PQ24 for Polo***

If the official document describes the relationship between high pressure and rotary speed as a compressor torque load, this variable can be validated the trend of stroke shift, so related high pressure and rotary speed will be the fuzzy input parameters. By another way, transfer compressor torque load to the compressor power load, simply use this equation: $P = M \bullet n$

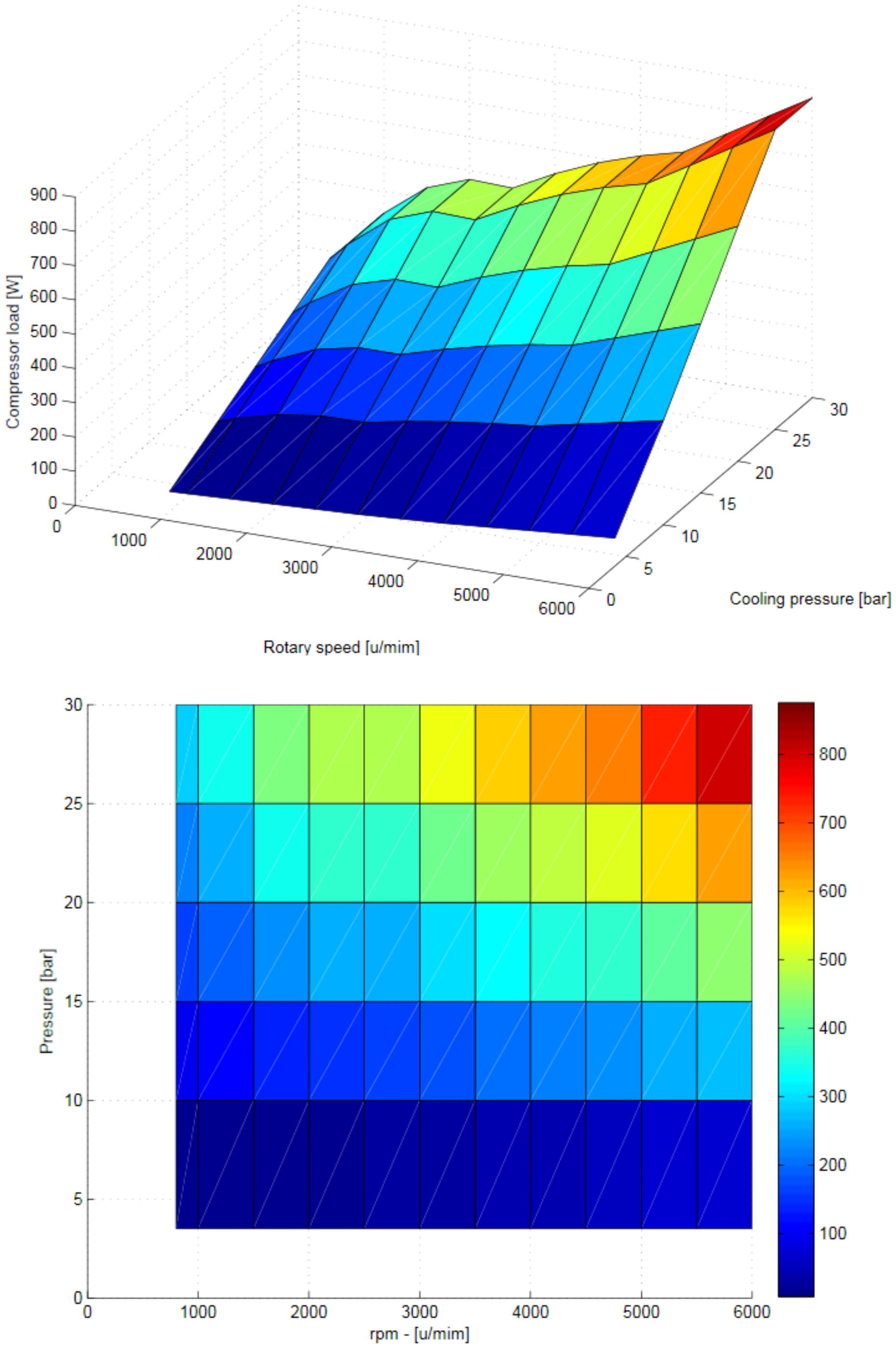


Figure 5.3: ***3-D and 2-D compressor load [W] for Polo***

Compressor power load can be determinate by Torque and Rotational speed, refers to the *Figure 5.2*, 3-D and 2-D of Power load [W] interpolated by MATLAB shows in the *Figure 5.3*. From 2-D Table, the colour bar represents the load by Watt. While the lowest power load of compressor, the highest stroke distance in order to make sure sufficient coolant mass flow rate injection.

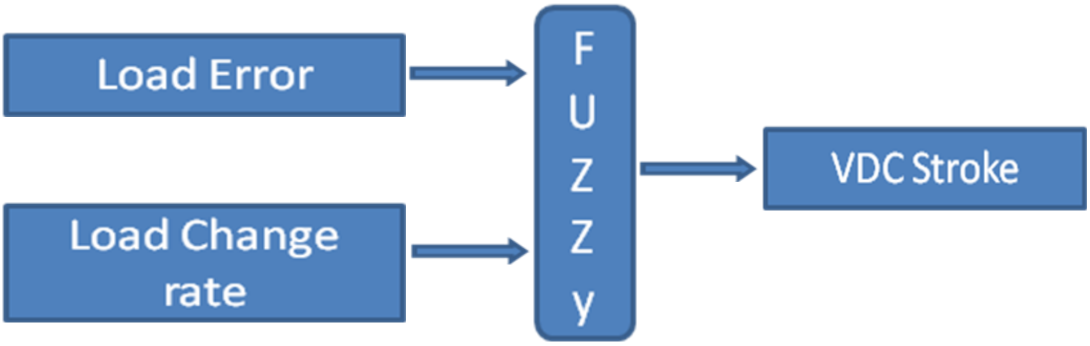
5.2. Fuzzy Logic Theory in MATLAB

In the chapter one, literature review has included the brief abstract of Fuzzy logic theory, in this theory defined Fuzzy logic (FL) has two different meanings. In a narrow sense, FL is a logical system, which is an extension of multi-valued logic. However, in a wider sense FL is almost synonymous with the fuzzy sets, this theory which relates to classes of objects with un-sharp boundaries in which membership is a matter of degree. In this perspective, FL in its narrow sense is a branch of FL. Even in its more narrow definition, FL differs both in concept and substance from traditional multi-valued logical systems. The basic concept in FL plays a central role in most of its applications, is that of a fuzzy if-then rule or, simply, fuzzy rule. Although rule-based systems have a long history of used in Artificial Intelligence (AI), what is missing in such systems is a mechanism of dealing with fuzzy consequents and fuzzy antecedents. In fuzzy logic, this mechanism is provided by the calculus of fuzzy rules. The calculus of fuzzy rules serves as a basis for what might be called the fuzzy Dependency of Command Language (FDCL). Although FDCL is not used explicitly in the MATLAB/Simulink toolbox, it is effectively one of its principal constituents. In most of applications of fuzzy logic, a fuzzy logic solution is, in reality, a translation of a human solution into FDCL.

MATLAB technical computing for Fuzzy Logic Toolbox software as a tool for solving control algorithms, FL is a fascinating area of research because it does a good job of trading off between significance and precision-something that humans have been managing for a very long time.

5.3. Fuzzy Logic Controller design for VDC

Fuzzy logic controller can be utilized for controlling the stroke distance, in order to provide related refrigerant mass flow rate to from VDC, like the FL theory describes, FL controller design is supposed to fix any conditions, such as different rotary speed, Air-con load, and so on. The compressor power load is a fuzzy input parameter in FL control diagram shows below.



The Stroke control for VDC refers to compressor load have been proofed by pervious section, rapidly making VDC from instable to stable which same as the variable approaches to setting point.

5.3.1. *Fuzzy inference*

Mamdani-type and Sugeno-type, these two types of inference systems vary somewhat in the way outputs are determined. Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Comparison with those two, the advantages of Mamdani method is more intuitive, widespread acceptance and suited to human input.

5.3.2. *Membership Functions*

The Fuzzy toolbox includes 11 built-in membership functions types. These 11 functions are, in turn, built from several basic functions:

- Piece-wise linear functions
- The Gaussian distribution function
- The sigmoid curve
- Quadratic and cubic polynomial curves

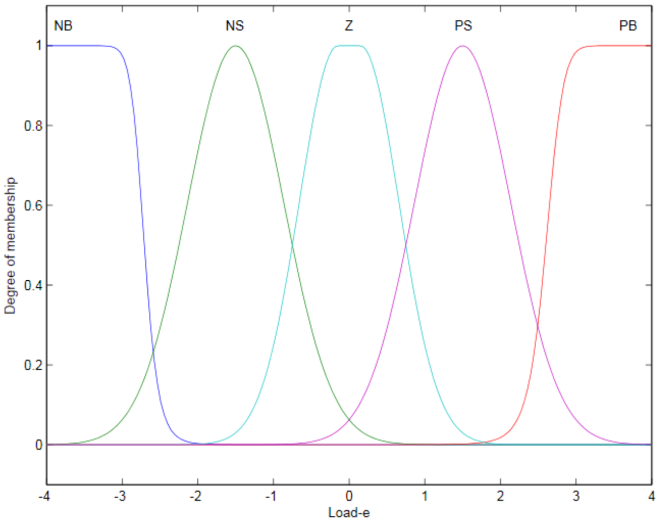
For smoothness and concise notation purpose, Gaussian and bell membership functions are popular methods for specifying fuzzy sets, gaussmf, gauss2mf and gbellmf will be the Membership Function (MF) for input and output setup.

5.3.3. Defuzzification

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is single number. Perhaps the most popular defuzzification method is the centroid calculation, which returns the centre of area under the curve. The defuzzification chooses centroid method in the modelling.

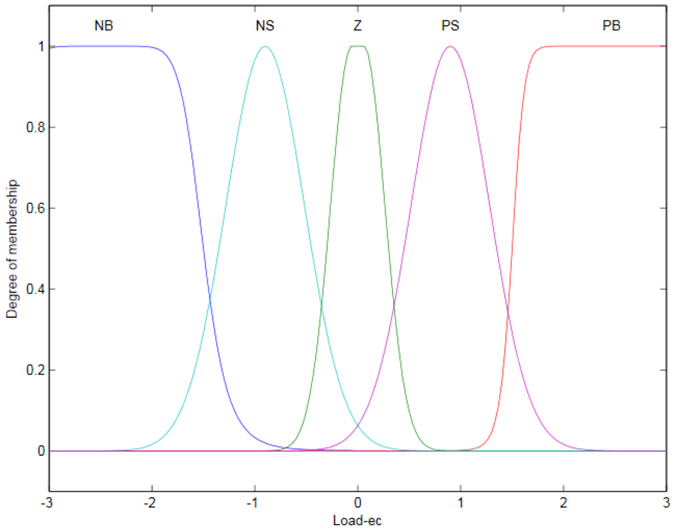
5.3.4. Load Error (FL input one)

This variable defined as the difference between setting compressor load and real-time detected compressor load on CAN, according to Fuzzy theory, the first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions – Fuzzify input. The input is always a crisp numerical value limited to the universe of discourse of the input variable (interval between -4 and 4), the MF associated with a given variable as follows:



5.3.5. Load change rate (FL input two)

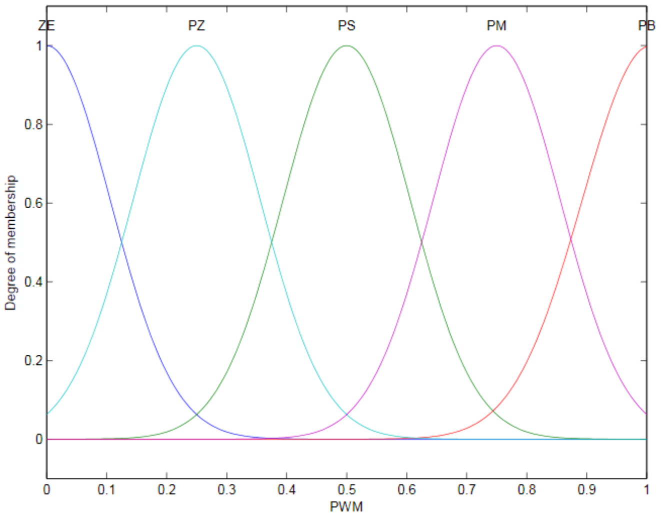
This variable defined as the load error changing rate, input variable (interval between -3 and 3). The MF associated with a given variable as follows:



Notes: Negative Big (NB); Negative Small (NS); Zero (Z); Positive Small (PS); Positive Big (PB)

5.3.6. VDC stroke (FL output)

Real stroke control relates to a PWM signal, variable duty cycle indicated the controlling output and the range of duty cycle defined as a percent value, output variable (interval between 0 and 1). The MF associated with a given variable as follows:



Notes: Zero (ZE); Positive Zero (PZ); Positive Small (PS); Positive Middle (PM); Positive Big (PB)

5.3.7. VDC stroke control rule

According to the different fuzzy I/O port, vehicle climatronic system obtain a variable power from engine, compressor load which involved on VDC torque load and engine rotary speed. Fuzzy control rules writing in the *Table 5.2*.

U		Load change rate				
		PB	PS	Z	NS	NB
Load error	PB	ZE	ZE	ZE	ZE	ZE
	PS	ZE	ZE	ZE	PZ	PZ
	Z	PZ	PZ	PS	PM	PM
	NS	PM	PM	PM	PB	PB
	NB	PB	PB	PB	PB	PB

Table 5.2: *Fuzzy control table*

When the load error is NB and whatever the load change rate it is, climatronic system needs a maximum mass flow rate to get rapidly cooling, so the stroke distance should keep as a max value PB. If the load error switches to NS, as different change rate, stroke control needs get down, in order making the mass flow rate keep a related constant. When the load error became to PB, it means climatronic system running at high power load, stroke control have to be a minimum value. The gensurf function in MATLAB, the surface view of Fuzzy rules in *Figure 5.4*

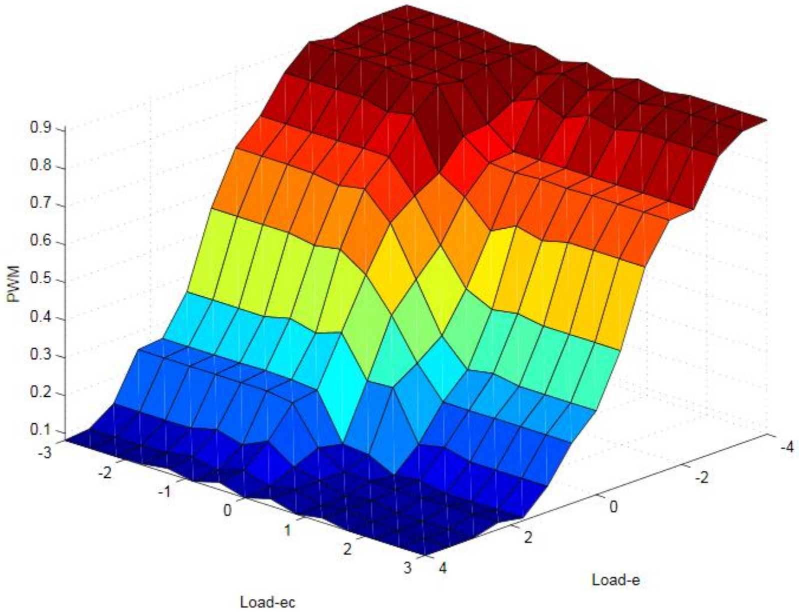


Figure 5.4: ***Surface of control rules***

5.4. Stateflow for Logic Control

Stateflow extends Simulink@ with a design environment for developing state machines and flow charts. Stateflow provides the language elements required to describe complex logic in a natural, readable, and understandable form. It is tightly integrated with MATLAB and Simulink, providing an efficient environment for designing embedded systems that contain control, supervisory, and mode logic. (Available:http://www.mathworks.com/products/stateflow/description1.html [1, May 2009])

All of CAN signals involved on the existing climatronic control unit has some certain logic condition with each other. To consider the HIL simulation those logic relationships are necessary. Whatever from safety reason or regular execution, this logic control has to achieve by MATLAB / Simulink environment which is assisting Fuzzy controller works in the real-time. Stateflow is the best option for logic control, the reason explains by Mathworks portal in last paragraph. Another reason is Stateflow in Simulink@ could be easy compiled for the third application software, such as CANoe, once the language in Stateflow compiled by real time workshop@ under Simulink, a C-code or a DLL file can be attach to the CANoe simulation and drive CANoe running without MATLAB

Prerequisite of enable compressor on are list as following, this is quoted on the VW document “Funkionsplan Klima-Halbautomat”.

- Vehicle ignition signal
- Blower load signal (Manually nonzero level)
- AC switch signal (Manually On)
- Ambient Temperature (more than 5 degree)
- Coolant pressure (less than 33 bar)
- Rotary speed (more than 400 rpm)

In the *Figure 5.5* shows the Stateflow chart, two subsystems for RPM and PRESSURE is the limitation for stroke control. Considering safety reason, under high RPM and exorbitant coolant pressure, stroke control value has to be down. P and R factor for compressor is the percentage of control value refers to output.

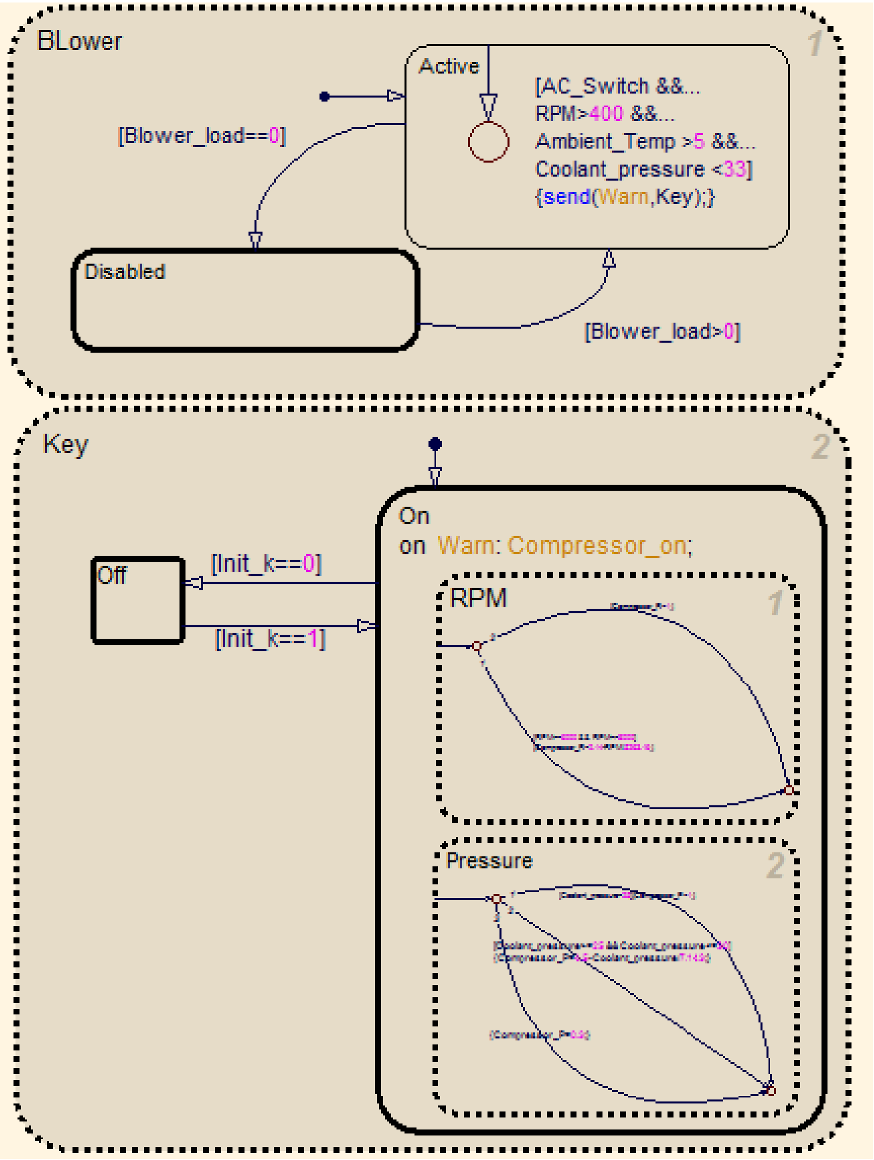


Figure 5.5: ***State flow Chart for compressor on***

As so far, combine with Stateflow chart and Fuzzy controller together, this whole simulated controller have done under MATLAB / Simulink, simply select the singles involved on this controller model from CAN database, completely modelling shows on Figure.5.6.

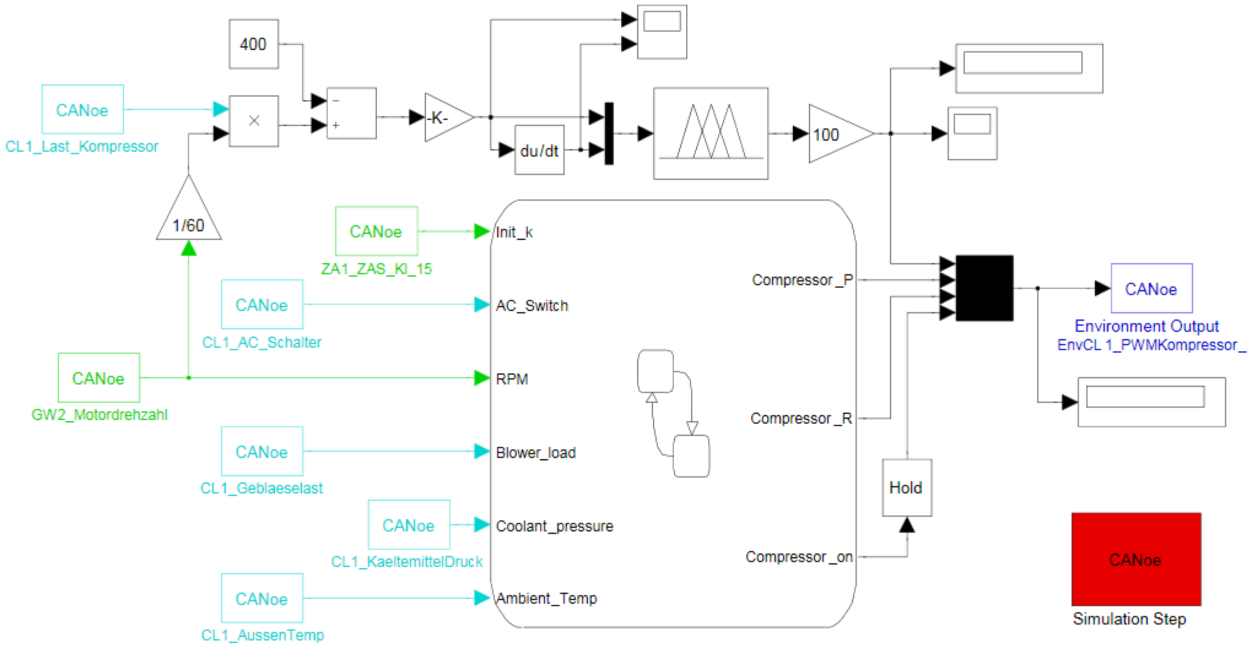


Figure 5.6: ***Final controller modelling***

5.5. CANoe / MATLAB setup

The first important setting of a model is the solver setting. The simulation is supposed to run under HIL mode, the RTW requires a fixed step solver. In the MATLAB configuration parameters, the CANoe target file must be selected in the RTW settings.

Simulation utilizing the CANoe / MATLAB interface can be operated in three different modes.

- **Offline Mode**

In this mode the simulation is run in the MATLAB / Simulink environment. CANoe is operated in slave mode whereas MATLAB / Simulink are the simulation master. The simulation is controlled from the MATLAB Simulink environment. CANoe’s slave mode is a special simulation mode whereby CANoe is simulating the CAN bus and takes its measurement time base from Simulink. Due to no real-time simulation is provided with this mode. This mode only can be used as a early design stages.

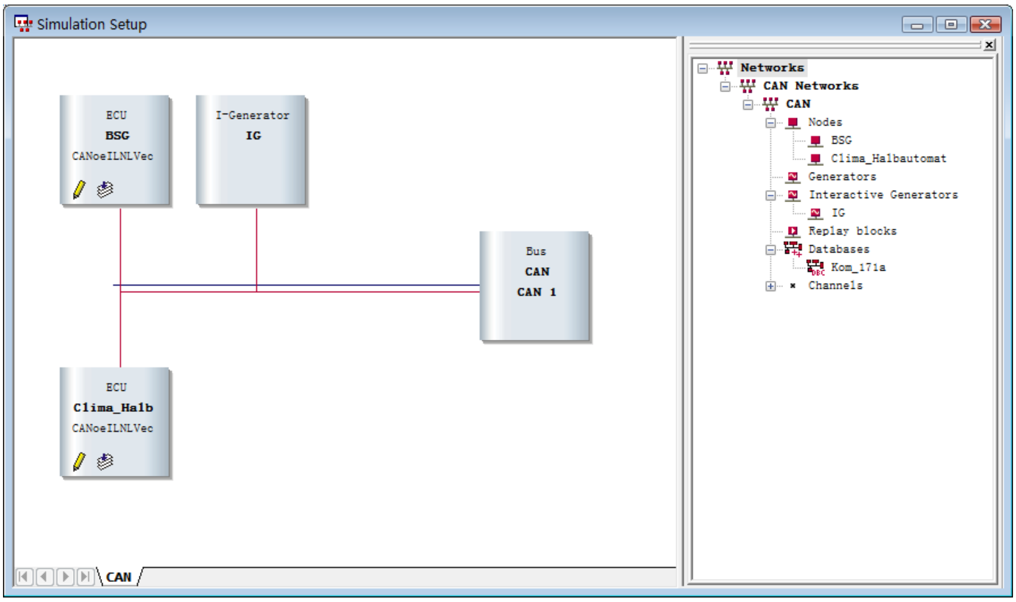
- **Synchronized Mode**

In this mode the simulation is run in the MATLAB / Simulink environment. This mode is run in almost real-time (typical resolution is about 1ms depending on the model). It is necessary that MATLAB / Simulink can compute the model faster than real-time in order to use this mode. This mode can be used for interaction with CAN card (hardware devices). It is highly recommend using a multi core processor for best simulation results. Communication between MATLAB / Simulink can CANoe takes place over Microsoft@ DCOM for configuration and shared memory for data exchange.

- **HIL Mode**

In this mode the simulation is run in the CANoe execution environment. With the Simulink@ Real-Time Workshop@ can target CANoe and produce a Windows@ DLL which can be loaded in CANoe's simulation environment. One DLL must be generated per node. With this approach it is possible to test and verify a design with Real CAN card in a networked environment. The simulation can be used as a simulation of remainder of the bus in a real-time environment.

In the final controller modelling, the signal from two ECUs that has to be simulated simultaneously. Following shows the CANoe Simulation setup.



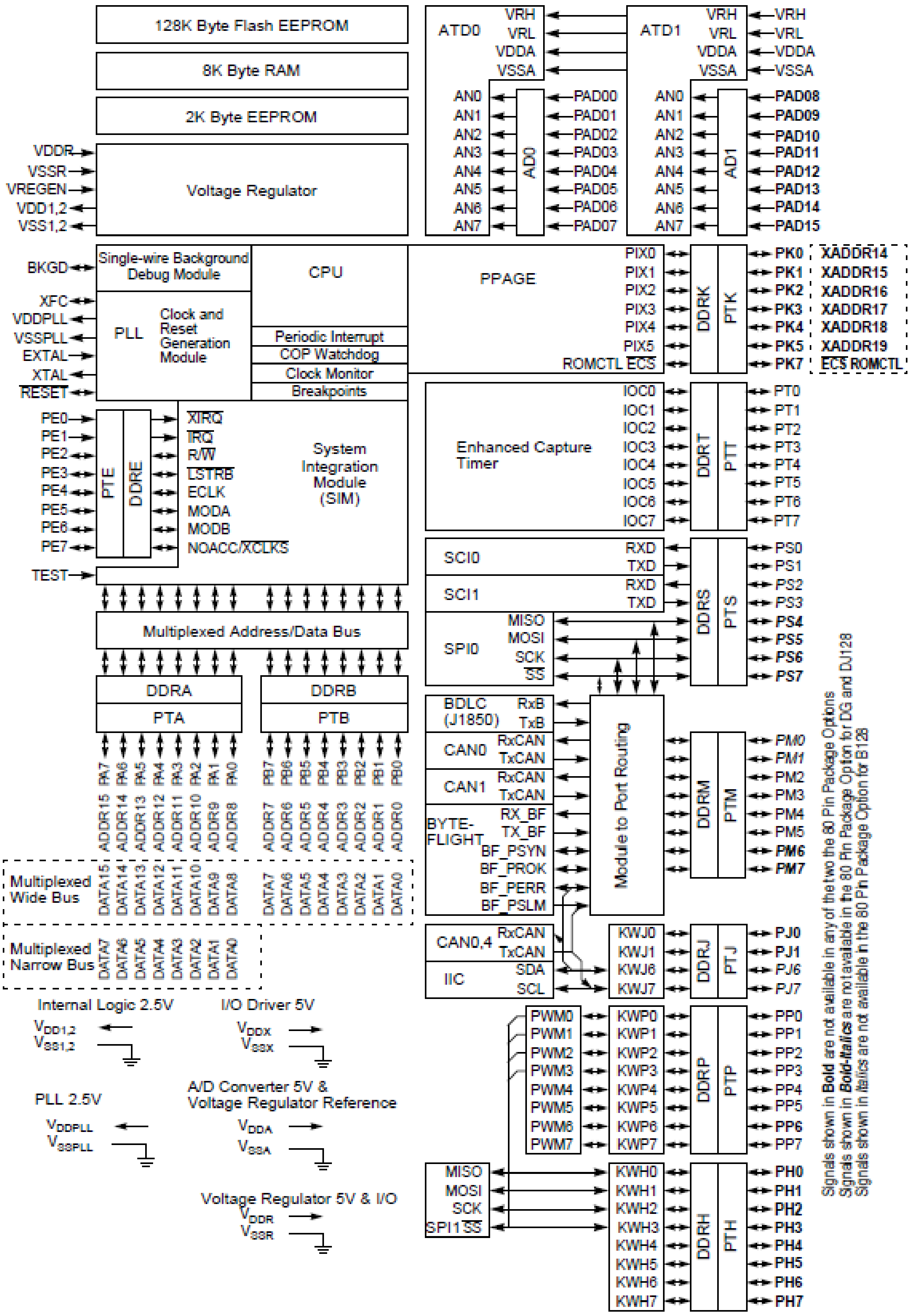
In the CANoe Simulation setup, two network nodes and one interactive generator which represent the vehicle ECU and one output for controller. Each network node includes the real involved CAN signals, massage and environment variables. CAN Application Programming Language (CAPL) is based on C-programming language, which allow people write the function for signals, massage and environment variables. CANoe tools provide an easy function: CAPL generator, this software tool generate the CAPL automatically by the database.

Due to the output variable for controller is not in the existing CAN database, so this additional environment variable have to be added in the new database, correspondingly add one environment function for this variable in the CAPL browser. Then the interactive generator could send the expected controller output variable embedded on mCLima messages by the same CAN channel.

The VDC stroke is controlled externally through a Compressor regulating valve show right-hand side. This valve is controlled using a Pulse width modulation (PWM) signal. With PWM an electrical signal is broken up in a controlled manner. Using constant frequency, voltage and current, the electrical energy can be controlled thus altering the effective current at constant voltage. So a hardware interface has to be designed for achievement.

5.6. Hardware interface

Freescale HCS12 microcontroller unit (MCU) provide high-performance 16-bit control for automotive applications. The MC9S12DG128 MCU composed of standard on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), 128K bytes of Flash EEPROM, 8K bytes of RAM, 2K bytes of EEPROM, two asynchronous serial communication interface (SCI), two serial peripheral interface (SPI), an 8-channel pulse-width modulator (PWM), a digital Byte Data Link Controller (BDLC), 29 discrete digital I / O channels (Port A,B,K and E), 20 discrete digital I / O lines with interrupt and wakeup capability, three CAN 2.0 A, B software compatible modules (MSCAN12), a Byte flight module and an Inter-IC bus. The MC9S12DG128 has full 16-bit data paths throughout. However, the external bus can operate in an 8-bit narrow mode so single 8-bit wide memory can be interfaced for lower cost systems. The inclusion of a Phased-Locked Loop (PLL) circuit allows power consumption and performance to be adjusted to operational requirements.



128K Byte Flash EEPROM
8K Byte RAM
2K Byte EEPROM
Voltage Regulator
VDDR
VSSR
VREGEN
VDD1,2
VSS1,2
Single-wire Background Debug Module
BKGD
CPU
PLL
Clock and Reset Generation Module
XFC
VDDPLL
VSSPLL
EXTAL
XTAL
RESET
Periodic Interrupt
COP Watchdog
Clock Monitor
Breakpoints
PTE
DDRE
PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
XIRQ
IRQ
R/W
LSTRB
ECLK
MODA
MODB
NOACC/XCLKS
System Integration Module (SIM)
TEST
Multiplexed Address/Data Bus
DDRA
PTA
DDRB
PTB
PA7 PA6 PA5 PA4 PA3 PA2 PA1 PA0
PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0
ADDR15 ADDR14 ADDR13 ADDR12 ADDR11 ADDR10 ADDR9 ADDR8
ADDR7 ADDR6 ADDR5 ADDR4 ADDR3 ADDR2 ADDR1 ADDR0
Multiplexed Wide Bus
DATA15 DATA14 DATA13 DATA12 DATA11 DATA10 DATA9 DATA8
DATA7 DATA6 DATA5 DATA4 DATA3 DATA2 DATA1 DATA0
Multiplexed Narrow Bus
DATA7 DATA6 DATA5 DATA4 DATA3 DATA2 DATA1 DATA0
Internal Logic 2.5V
$V_{DD1,2}$
$V_{SS1,2}$
I/O Driver 5V
V_{DDX}
V_{SSX}
PLL 2.5V
V_{DDPLL}
V_{SSPLL}
A/D Converter 5V & Voltage Regulator Reference
V_{DDA}
V_{SSA}
Voltage Regulator 5V & I/O
V_{DDR}
V_{SSR}
ATD0
VRH VRL VDDA VSSA
AN0 AN1 AN2 AN3 AN4 AN5 AN6 AN7
AD0
PAD00 PAD01 PAD02 PAD03 PAD04 PAD05 PAD06 PAD07
ATD1
VRH VRL VDDA VSSA
AN0 AN1 AN2 AN3 AN4 AN5 AN6 AN7
AD1
VRH VRL VDDA VSSA
PAD08 PAD09 PAD10 PAD11 PAD12 PAD13 PAD14 PAD15
PPAGE
PIX0 PIX1 PIX2 PIX3 PIX4 PIX5
ROMCTL ECS
DDRK
PTK
PK0 PK1 PK2 PK3 PK4 PK5 PK7
XADDR14 XADDR15 XADDR16 XADDR17 XADDR18 XADDR19 ECS ROMCTL
Enhanced Capture Timer
IOC0 IOC1 IOC2 IOC3 IOC4 IOC5 IOC6 IOC7
DDRT
PTT
PT0 PT1 PT2 PT3 PT4 PT5 PT6 PT7
SCI0
RXD TXD
SCI1
RXD TXD
SPI0
MISO MOSI SCK SS
DDRS
PTS
PS0 PS1 PS2 PS3 PS4 PS5 PS6 PS7
BDLC (J1850)
RxB TxB
CAN0
RxCAN TxCAN
CAN1
RxCAN TxCAN
BYTE-FLIGHT
RX_BF TX_BF BF_PSYN BF_PROK BF_PERR BF_PSLM
Module to Port Routing
DDRM
PTM
PM0 PM1 PM2 PM3 PM4 PM5 PM6 PM7
CAN0,4
RxCAN TxCAN
IIC
SDA SCL
KWJ0 KWJ1 KWJ6 KWJ7
DDRJ
PTJ
PJ0 PJ1 PJ6 PJ7
PWM0 PWM1 PWM2 PWM3 PWM4 PWM5 PWM6 PWM7
KWP0 KWP1 KWP2 KWP3 KWP4 KWP5 KWP6 KWP7
DDRP
PTP
PP0 PP1 PP2 PP3 PP4 PP5 PP6 PP7
SPI1
MISO MOSI SCK SS
KWH0 KWH1 KWH2 KWH3 KWH4 KWH5 KWH6 KWH7
DDRH
PTH
PH0 PH1 PH2 PH3 PH4 PH5 PH6 PH7

Signals shown in **Bold** are not available in any of the two the 80 Pin Package Options
Signals shown in ***Bold-Italics*** are not available in the 80 Pin Package Option for DG and DJ128
Signals shown in *Italics* are not available in the 80 Pin Package Option for B128

Figure 5.7: ***MC9S12DT128 Block Diagram [7]***

5.6.1. *Power Supply Block*

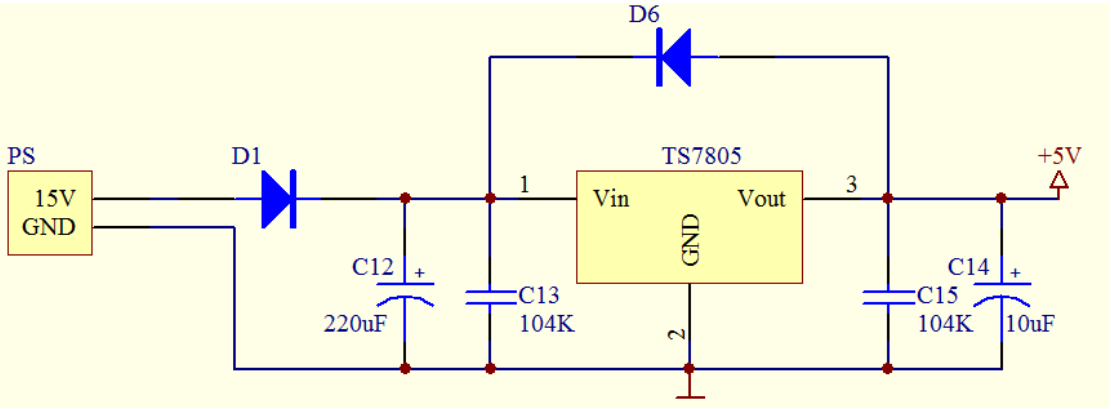
According to the Table power supply pins' description , VDDX,VSSX are the external power and ground I/O drivers; VDDR, VSSR are input to the internal voltage regulator. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high- frequency characteristics and place them as close to the MCU as possible. Bypass requirements depend on how heavily the MCU pins are loaded.

Mnemonic	**Pin No.**	**Voltage**	**Description**
VDD1,2	13,65	2.5V	Internal power and ground generated by internal regulator.
VSS1,2	14,66	0V	
VDDR	41	5.0V	External power and ground, supply to pin drivers and internal voltage regulator.
VSSR	40	0V	
VDDX	107	5.0V	External power and ground, supply to pin drivers.
VSSX	106	0V	
VDDA	83	5.0V	Operating voltage and ground for the ADC and the reference for the internal voltage regulator, allows the supply voltage to the A/D to be bypassed independently
VSSA	86	0V	
VRL	85	0V	Reference voltage for the ADC.
VRH	84	5.0V	
VDDPLL	43	2.5V	Provides operating voltage and ground for the PLL. This allows the supply voltage to the PLL to be bypassed independently. Internal power and ground generated by internal regulator.
VSSPLL	45	0V	
VREGEN	97	5V	Internal Voltage Regulator enable/disable.

Table 5.3: ***MC9S12DT128 Power and Ground connection summary***

Power is supplied to the MCU through VDD and VSS. This 2.5V supply is derived from the internal voltage regulator. There is no static load on those pins allowed. The internal voltage regulator is turned off, if VREGEN is tied to ground. VDDA, VSSA are the power supply and ground input pins for the voltage regulator and the analog to digital converter (ADC). It also provides the reference for the internal voltage regulator. This allows the supply voltage to the ATD and the reference voltage to be bypassed independently. VDDPLL, VSSPLL provide operating voltage and ground for the Oscillator and PLL. This allows the supply voltage to the Oscillator and PLL to be bypassed independently. This 2.5V voltage is generated by the internal voltage regulator.

So the whole MCU power supply only needs an external +5V for I/O logic drivers. Normal vehicle battery has +12V supply, consider the power load for all of hardware interfacing application, normal 7805 three terminal positive regulator are available, input voltage range of 7805 is 5V~18V; peak current is 2.2 A; output voltage around 4.9~5.1V. 7805 employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provide, they can deliver over 1A output current. Schematic circuit 5.1 for power supply shows below.



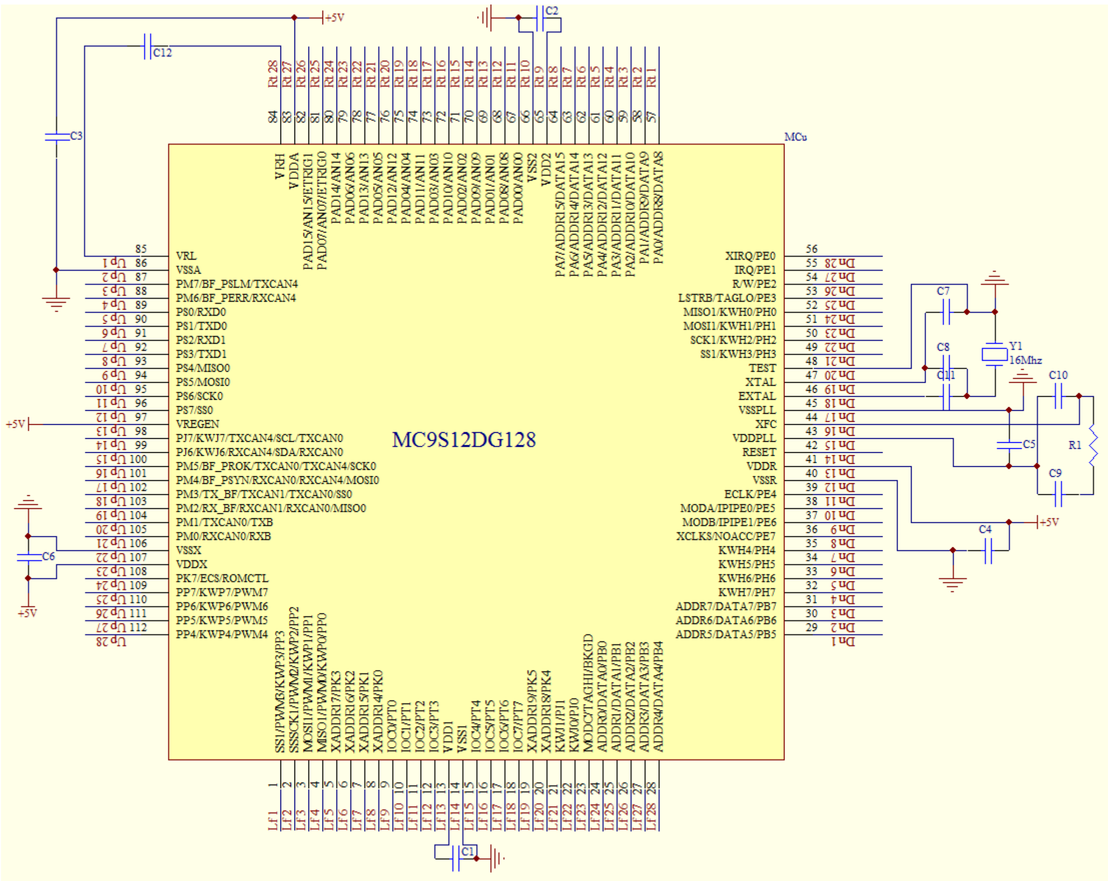
Schematic circuit 5.1: ***Power supply for MCU***

5.6.2. Oscillator Block

EXTAL and XTAL are the crystal driver and external clock pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output. The XCLKS is an input signal which controls whether a crystal in combination with the internal Colpitts (low power) oscillator is used or whether

Pierce oscillator / external clock circuitry is used. The state of this pin is latched at the rising edge of RESET. If the input is logic low the EXTAL pin is configured for an external clock drive. If input is logic high an oscillator circuit is configured on EXTAL and XTAL. Since this pin is an input with a pull-up device during reset, if the pin is left floating, the default configuration is an oscillator circuit on EXTAL and XTAL.

The hardware interfacing application use Pierce Oscillator connections, so the XCLKS left open and use default configuration setup. The oscillator provides the reference clock for the PLL, The PLL's Voltage Controlled Oscillator (VCO) is also the system clock source in self clock mode. PLL loop filter circuit comply with MC9S12DG128's datasheet. Minimum MCU schematic circuit 5.2 and recommended PCB layout show below.



Schematic circuit 5.2: ***Minimum MCU element***

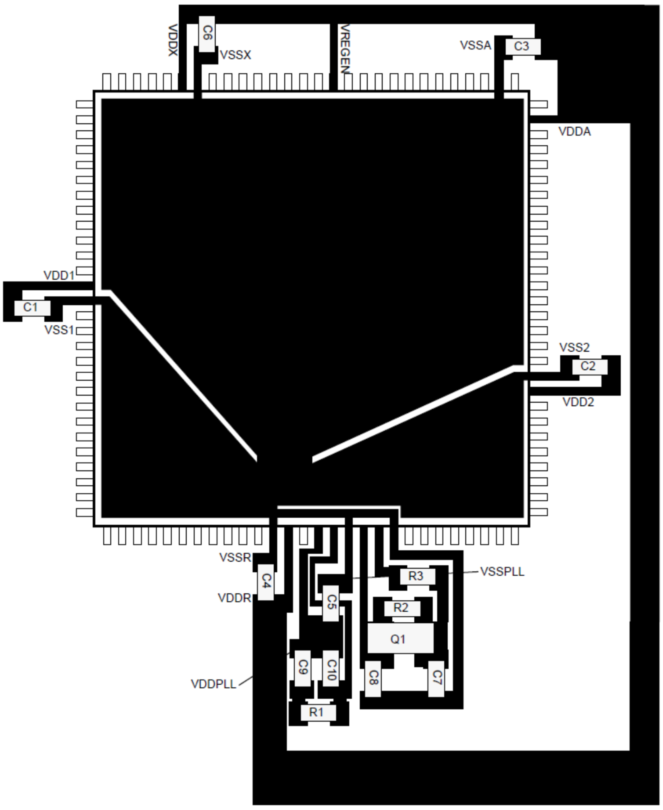


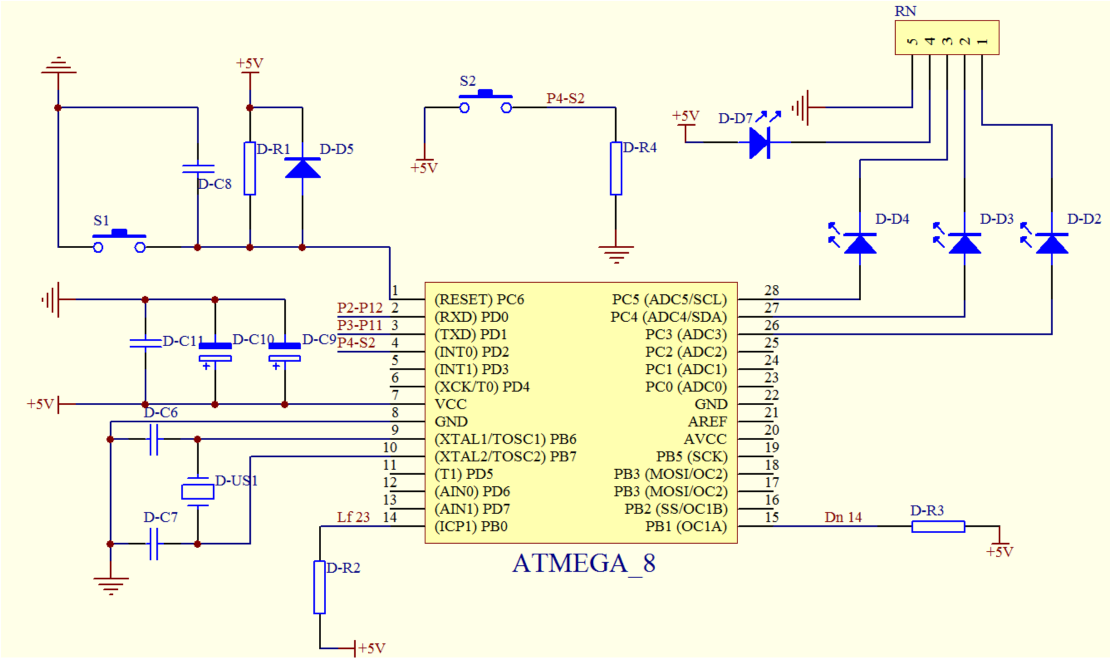
Figure 5.9: ***Recommended PCB layout for minimum MCU with Pierce Oscillator***

5.6.3. *Background Debug Module (BDM)*

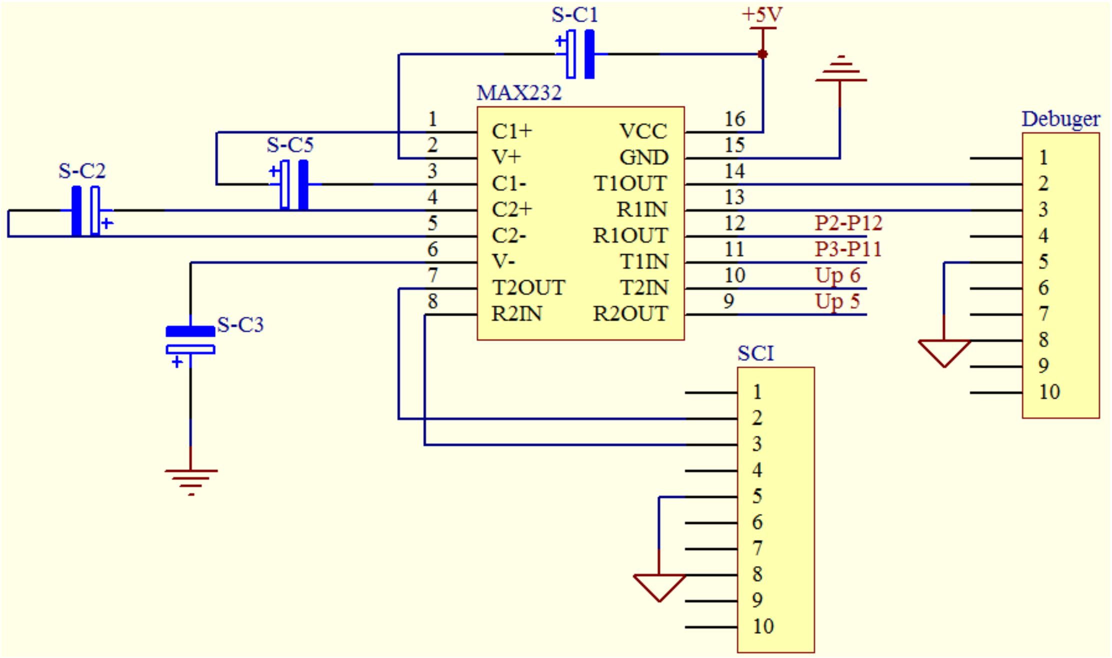
The BDM sub-block is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin. BDM has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to show the clock rate and a handshake signal to indicate when an operation is complete.

Mr. Juergen Kranz designed a serial monitor for HCS12 hardware debugger by an 8 bit controller, which are including a MAX232 dual - serial driver and an ATmega16. This hardware BDM debugger could run under HCS12 package of CodeWarrior4.7

serial monitor mode. Due to MAX232 is a dual-channel driver, so one channel have to be used for debugging and other channel could used for SCI. An assembly-code for ATmega16 references at his master thesis. Following shows the schematic of hardware debugger circuit and serial receiver.



Schematic circuit 5.3: *Hardware debugger*



Schematic circuit 5.4: *Serial receiver*

5.6.4. CAN module

The Motorola Scalable Controller Area Network (MSCAN) definition is based on the MSCAN12 definition which is the specific implementation of the Motorola Scalable CAN concept targeted for the HCS12. The module is a communication controller implementing the CAN 2.0 A/B protocol as defined in the BOSCH specification dated September 1991. MSCAN utilizes an advanced buffer arrangement resulting in a predictable real-time behaviour and simplifies the application software. *Figure 5.10* shows MSCAN block diagram.

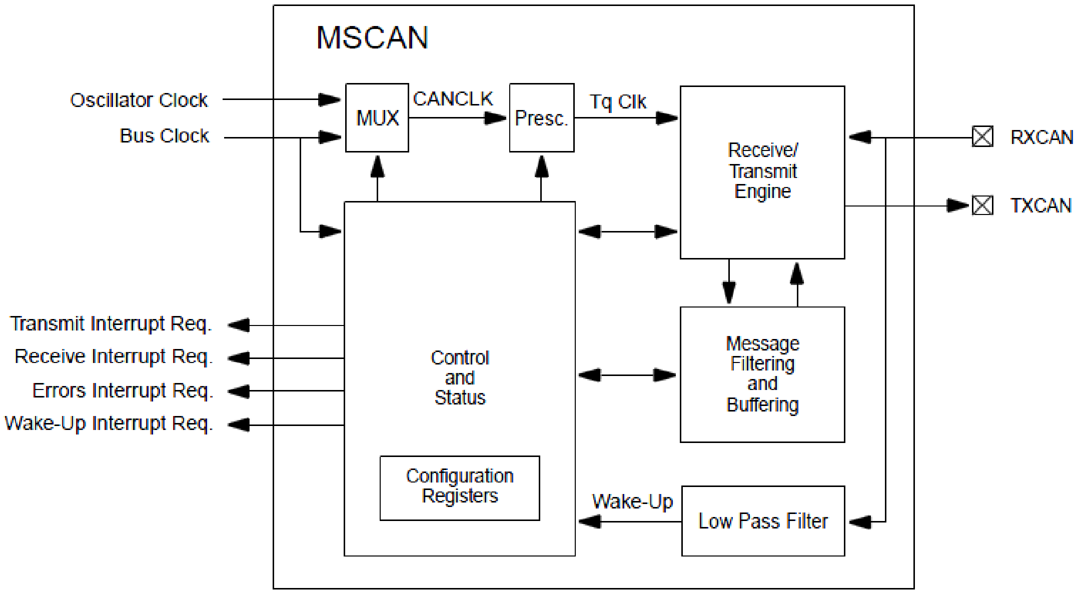


Figure 5.10: *MSCAN Block Diagram [9]*

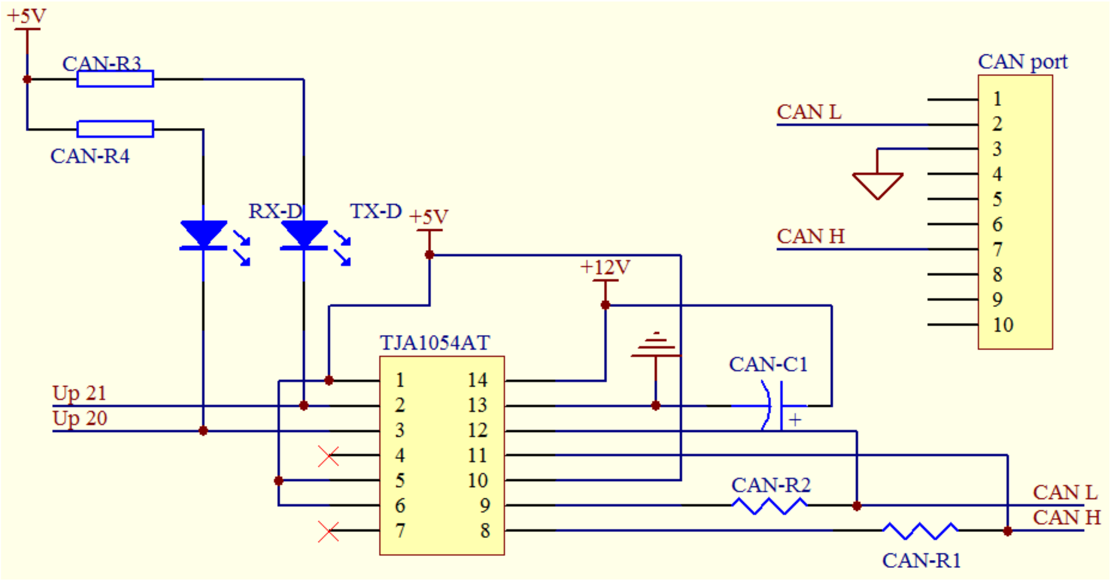
The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol – Version 2.0 A/B
  - Standard and extended data frames
  - 0~8 bytes data length
  - Programmable bit rate up to 1Mbps
  - Support for remote frames
  - 5 receive buffer with FIFO storage scheme
- 3 transmit buffer with internal prioritization using a “local priority” concept

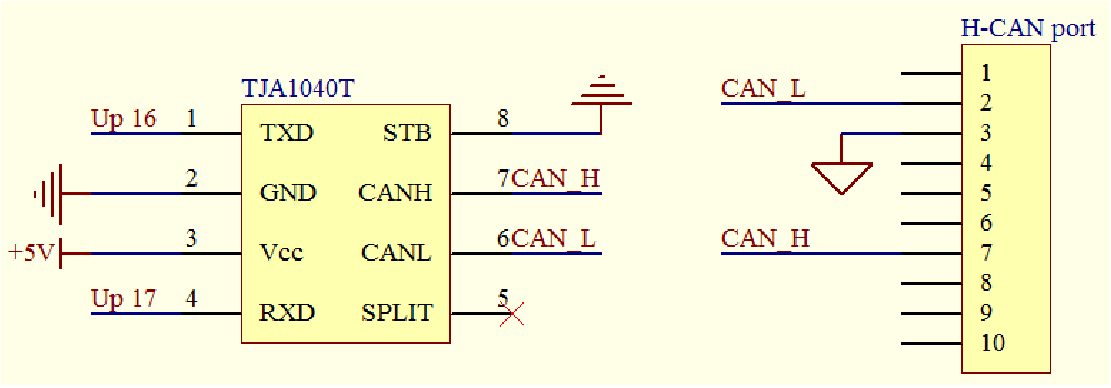
- Flexible maskable identifier filter supports two full size extended identifier filter (two 32-bit) or four 18-bit filters or eight 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter
- Programmable loop back mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (Warning, Error Passive, Bus-Off)
- Programmable MSCAN clock source either Bus Clock or Oscillator Clock
- Internal timer for time-stamping of received and transmitted messages
- Three low power modes: Sleep, Power Down and MSCAN Enable
- Global initialization of configuration registers

To get communication by CAN bus, the protocol of CAN was primarily, but not only, designed to be used on the vehicle CAN bus, this hardware interface controller board have been connected all of three channels. Two channels for vehicle low speed CAN; one for vehicle high speed CAN. “Konzern-Lastenheft CAN im Komfort- und infotainment- Bereich” makes a definition about the baud rate for different speed CAN. Philips semiconductor TJA1054 is the interface between the protocol controller and the physical bus wires in a CAN. It is primarily intended for low-speed applications up to 125kBaud in passenger cars. It provides differential receive and transmit capability but will switch to single-wire transmitter or receiver in error conditions. TJA1040 is primarily intended for high speed applications, up to 1Mbaud. Schematic 5.5 and 5.6 for both CAN transceivers are showing below.

Notes: schematic circuit of CAN transceiver set the STB bit and EN bit always at logic high and leave the WAKE bit at logic low, so CAN module is no wake-up function at all.



Schematic circuit 5.5: ***CAN transceiver for low-speed***



Schematic circuit 5.6: ***CAN transceiver for high-speed***

C-Code forCAN module

The registers involved on MSCAN module list at following:

- CONTROL 0 REGISTER (CANCTL0) provides for various control of the MSCAN module. Initialization Mode Request (INITRQ) at bit0, when this bit is set by the CPU, the MSCAN skips to Initialization Mode, any ongoing transmission or reception is aborted and synchronization to the bus is lost.
- CONTROL 1 REGISTER (CANCTL1) provides for various control and handshake status information of the MSCAN module. MSCAN Enable (CANE)

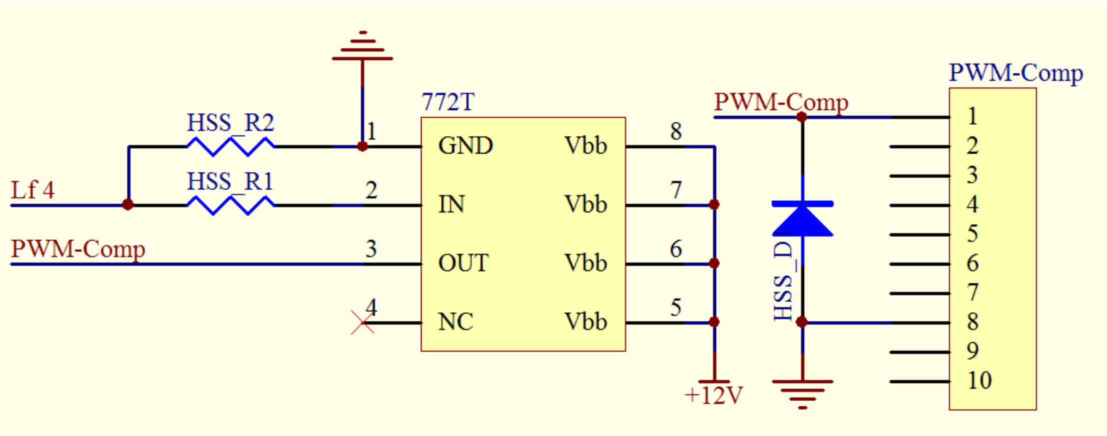
at bit7, after INITRQ and INITAK set to 1 enable the CAN. Loop Back Self Test Mode (LOOPB) at bit5, when this bit is set, the MSCAN performs an internal loop back which can be used for self test operation. In this state, the MSCAN ignores the bit sent during the Acknowledge (ACK) slot in the CAN frame ACK field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- MSCAN BUS TIMING REGISTER 0 (CANBTR0) provides for various bus timing control of the MSCAN module. Baud Rate Prescaler (BRP [5~0]) at bit5~0 determine the time quanta (Tq) clock which is used to build up the individual bit timing.
- MSCAN BUS TIMING REGISTER 1 (CANBTR1) also provide for various bus timing control of the MSCAN module. Sampling (SAMP) at bit7 determines the number of samples of the serial bus to be taken per bit time. Regular one preceding samples using a majority rule. Time Segment 2 (TSEG22~TSEG20) and Time Segment 1 (TSEG12~TSEG10) within the bit time fix the number of clock cycles per bit time and location of the sample point.
- MSCAN RECEIVER FLAG REGISTER (CANRFLG), this flag can only be cleared when the condition which caused the setting is no longer valid and can only be cleared by software.
- MSCAN RECEIVER INTERRUPT ENABLE REGISTER (CANRIER) contains the interrupt enable bits for the interrupt flags.
- MSCAN IDENTIFIER ACCEPTANCE REGISTERS (CANIDAR0~7), on reception, each message is written into the background receive buffer. The CPU is only signalled to read the message if it passes the criteria in the identifier acceptance and identifier mask registers (accepted); otherwise, the message is overwritten by the next message (dropped).
- MSCAN IDENTIFIER MASK REGISTERS (CANIDMR0~7) specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering.

5.6.5. *PWM module*

In the HCS12 PWM definitions, this module contains the basic features from the HC11 with some of the enhancements incorporated on the HCS12, that is centre aligned output mode and four available clock sources. In the HCS12 module has eight channels with independent control of left and centre aligned outputs on each channel. Each of the eight channels has a programmable period and duty cycles as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources to be used with the counters. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%. The PWM outputs can be programmed as left aligned outputs or centre aligned outputs.

Compressor regulating valve requires a +12V PWM input, PWM output from MCU needs a +5V to +12V driving circuit. Infineon BSP772T as a smart power high-side-switch, which is N channel vertical power FET with charge pump, ground referenced CMOS compatible input and monolithically integrated in smart SIPMOS technology. Schematic circuit 5.7 shows below:



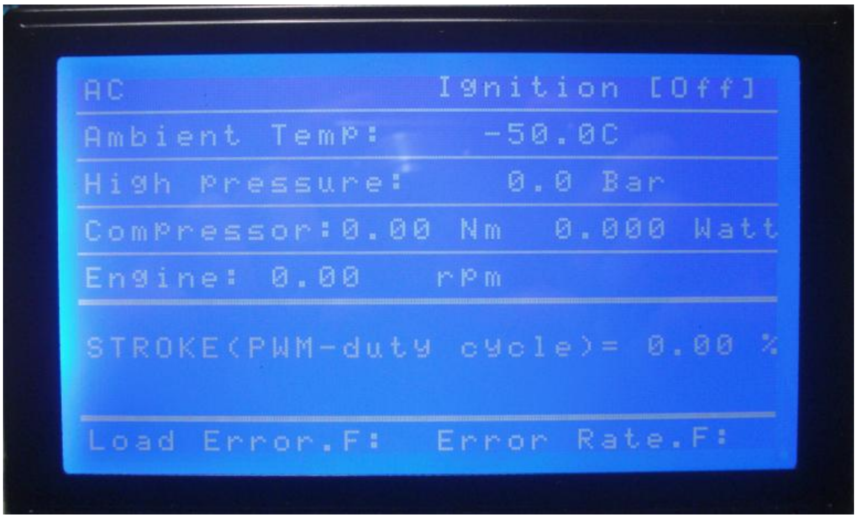
Schematic circuit 5.7: *+12V PWM driver*

The registers involved on PWM module list at following:

- PWM ENABLE REGISTER (PWME): each PWM channel has an enable bit to start its waveform output. If all PWM channels are disabled, the Prescaler counter shuts off for power savings.
- PWM CLOCK SELECT REGISTER (PWMCLK): each PWM channel has a choice of two clocks to use as the clock source for that channel.
- PWM PRESCALE CLOCK SELECT REGISTER (PWMPRCLK) selects the prescale clock source for clocks A and B independently.
- PWM CENTER ALIGN ENABLE REGISTER (PWMCAE) contains eight control bits for the selection of centre aligned outputs or left aligned outputs for each PWM channel.
- PWM CONTROL REGISTER (PWMCTL) provides for various control of the PWM module. CON67, CON45, CON23 and CON01 are the 4 bit provides the function for concatenating or separating adjacent channels.
- PWM CHANNEL PERIOD REGISTERS (PWMPER), the value in this register determine the period of the associated PWM channel.
- PWM CHANNEL DUTY REGISTERS (PWMDTY), the value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and output changes state.

5.6.6. LCD dispaly module

The T6963C graphics embedded controller of 240*128 Liquid Crystal Display (LCD) used for hardware interface. It is interfaced with a number of different 8-bit MUC, T6963 embedded controller chip provides a 128 character-Generator ROM (CG-ROM), the capability to control up to 64K bytes of external display RAM (VRAM) and generates the necessary timing and data signals for the LCD driver circuits. Combine test and graphics data formats support to LCD module. This LCD display module only for auxiliary monitoring purpose, *Figure 5.11 shows* the message involved on the Fuzzy controller. C-code for LCD display driver list into the *Appendix C.1*

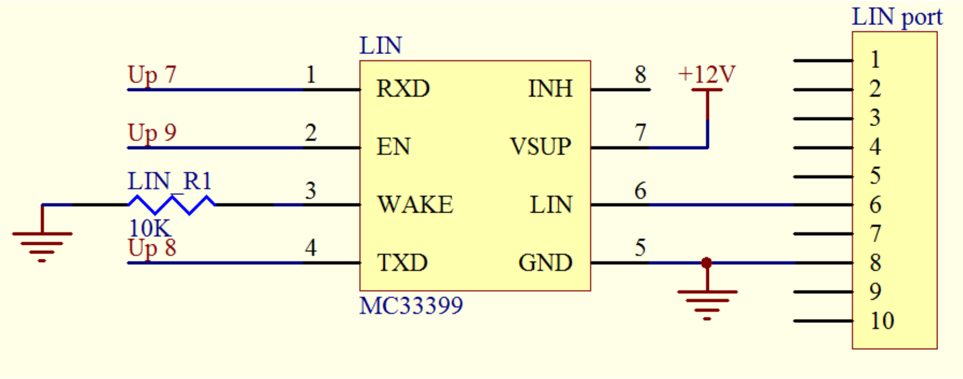


```
AC                Ignition [Off]
Ambient Temp:      -50.0C
High pressure:      0.0 Bar
Compressor:0.00 Nm  0.000 Watt
Engine: 0.00   rpm

STROKE(PWM-duty cycle)= 0.00 %

Load Error.F:  Error Rate.F:
```

Figure 5.11: ***240*128 LCD display***

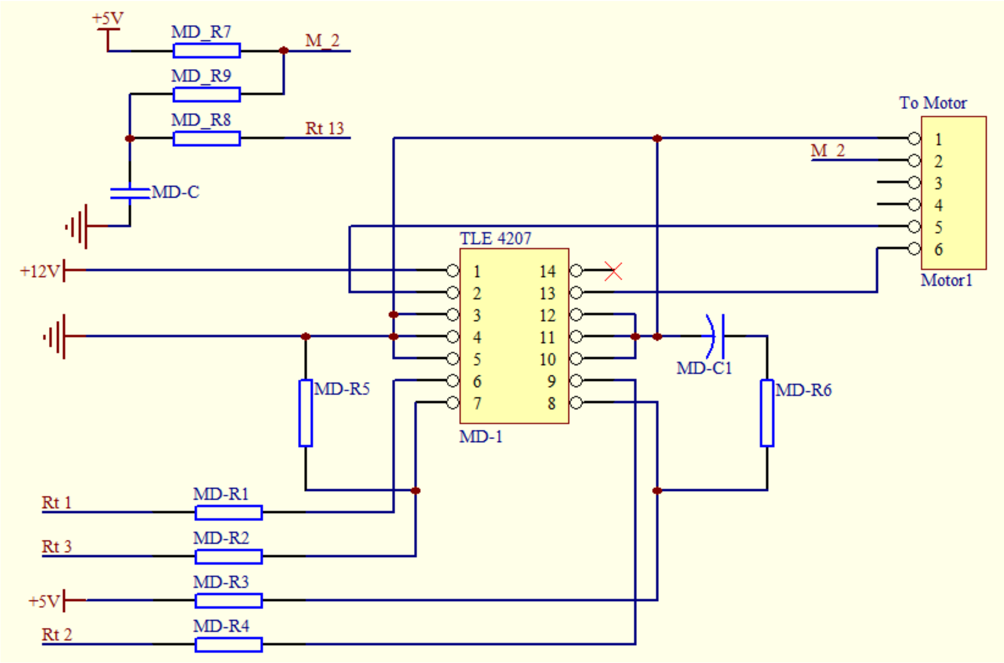
5.6.7. Other module and finial PCB

Due to the time limitation of my research, rest of the module have already designed onto the hardware interfacing board, but not complete testing yet. Those modules will be available for feature study. Local Interconnect Network (LIN) interface chip MC33399 normally used for controlling the Fan speed of blower, because of the power of blower is nearly 200W, a PWM driving control signal will always occur electro-magnetic interference, so MCU and terminal Fan speed controller are batter separated, and LIN bus will be the best network for this medium. The Schematic circuit shows 5.8 below:



Schematic circuit 5.8: *LIN Interface*

Flap position control module, TLE4207 and TLE4208 are fully protected Bridge Driver designed especially for automotive motion control applications. DC motor can be connected direct between the half-bridges. Operation modes forward, reverse, brake and high impedance are invoked from a standard interface. The Schematic circuit 5.9 shows below:



Schematic circuit 5.9: *Flap position driver*

There are a few more functions for hardware interfacing board is not introduced such as a Inter-Integrated Circuit (I^{2}C) interrupt controller, 4 analog sensor input channels.

Integrated all of schematic circuits, final Printed Circuit Board (PCB) can be created by the software of Altium Designer 6. Three-dimensional view of PCB layout shows below.

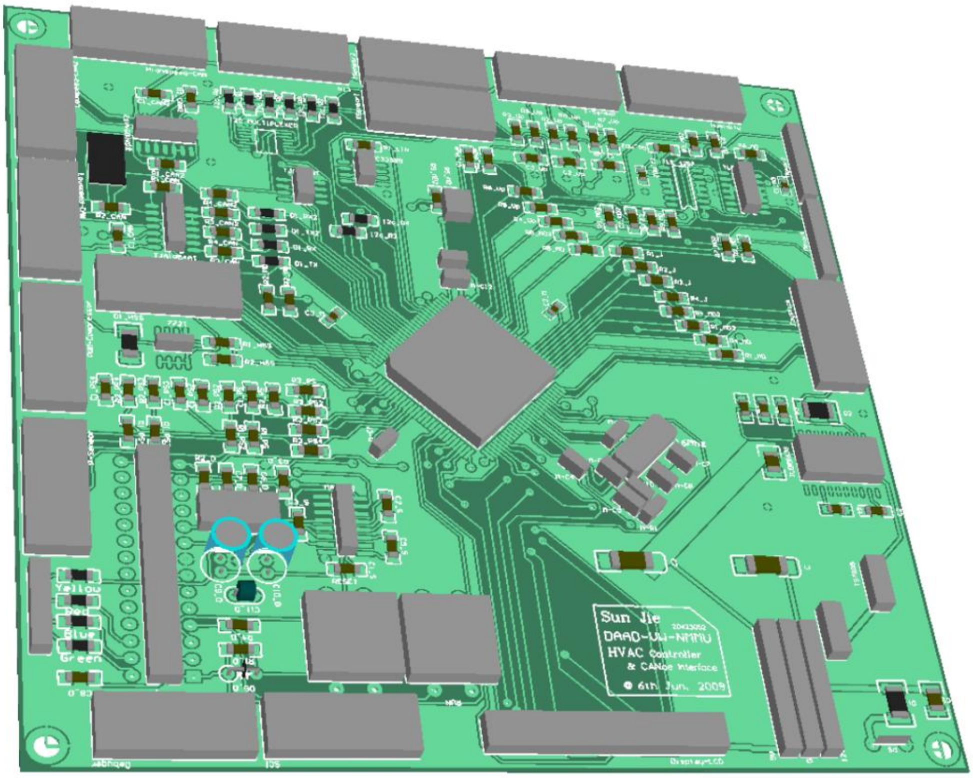


Figure 5.12: ***3D-view for PCB layout***

-Chapter 6

Conclusion

This thesis makes several contributions to the control system on instability of automotive air conditioning with a variable displacement compressor. First, an existing modelling approach is applied to an R134a subcritical cycle (*Appendix P.1*). Second, four different modelling approach is presented each component on the cooling circuit. The heat-energy based approach is more straightforward to derive and simpler conceptually. Third, the resulting from cooling cycle modelling are validated using experimental data and recommendation are made for improving the model validation. Fourth, data analysis and comparison onto the simulation modelling was given, an empirical real-time controller models constructed using by Fuzzy theory and Stateflow logic.

Each input and output have to be estimated the parameters for Fuzzy controller, those information is possible to derive from original vehicle CAN-Bus. Combine with hardware and software solution by Vector-CANoe system. MATLAB / Simulink integrated with this CAN issue, Hardware-in-Loop simulation is achieved. In addition to a hardware interfacing, a real-time control system for variable displacement compressor is completed. A flexible adjustment of real-time simulation controller is easy to change the inner parameter on-line, in order to achieve different human comfort level.

This research has many aspects that have yet to be explored. A few of these are mentioned here, including improvements in model validation, model reduction, new method of controller design, and modelling of complex systems.

- Model validation

  In chapter 4 many observations were made regarding problems with experimental data. A new VDC modelling has more fit with dynamic performance; reconsider the sensor location onto the test beach; need for unfiltered mass flow measurements at throttling device; misdistribution of refrigerant in the prototype evaporator. To truly test the validity of vehicle cooling cycle, it could be experimentally verified on automotive systems. This obviously requires a large amount work.

- Model reduction

  A logical part of future work is to validate the modelling approach and explore possibilities for model reduction, and suitable for real-time control object.

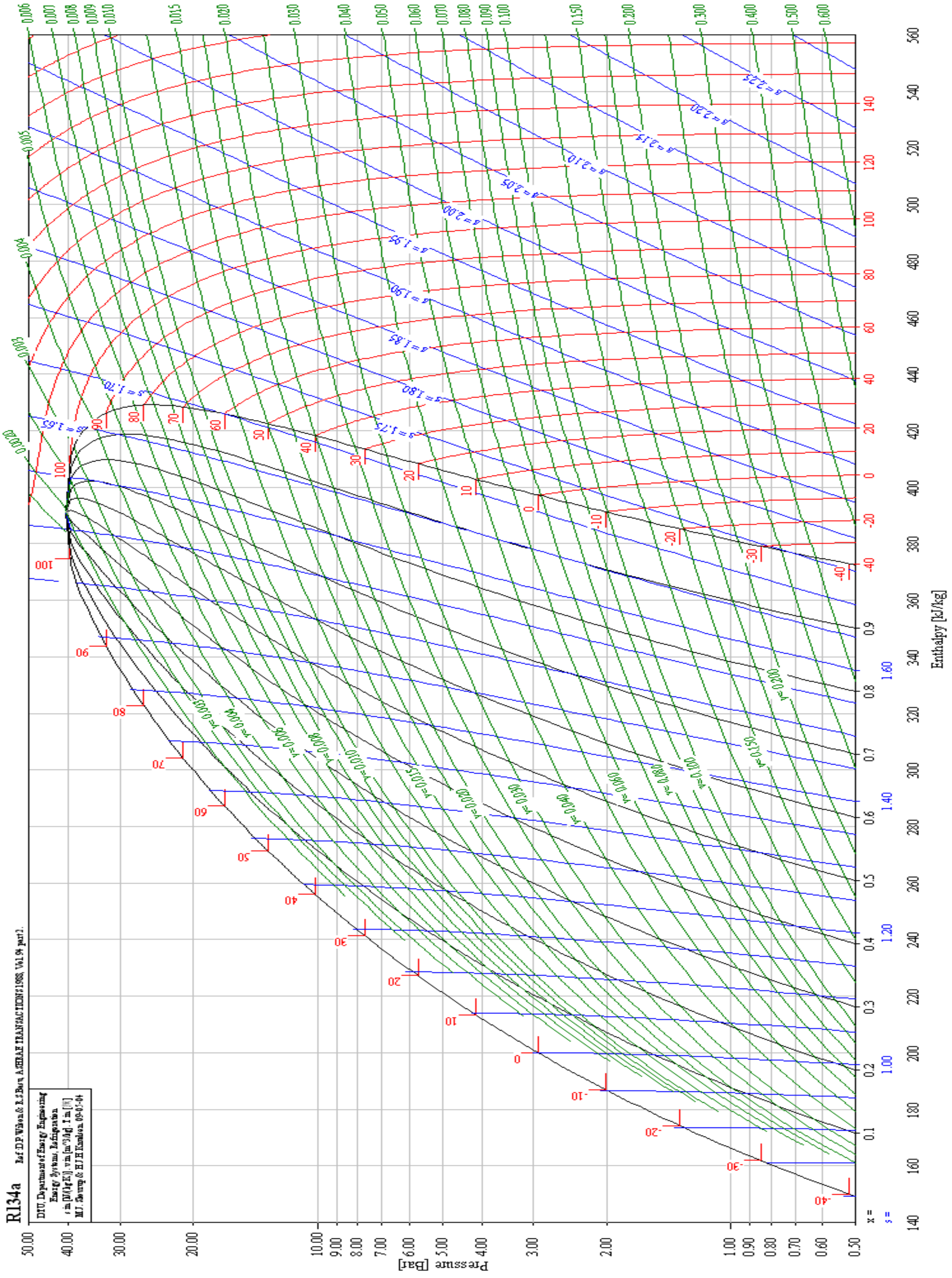
- New controller design

  As a new accuracy or more empirical control object is obtain. New type controller such as Nerve network could be utilized.

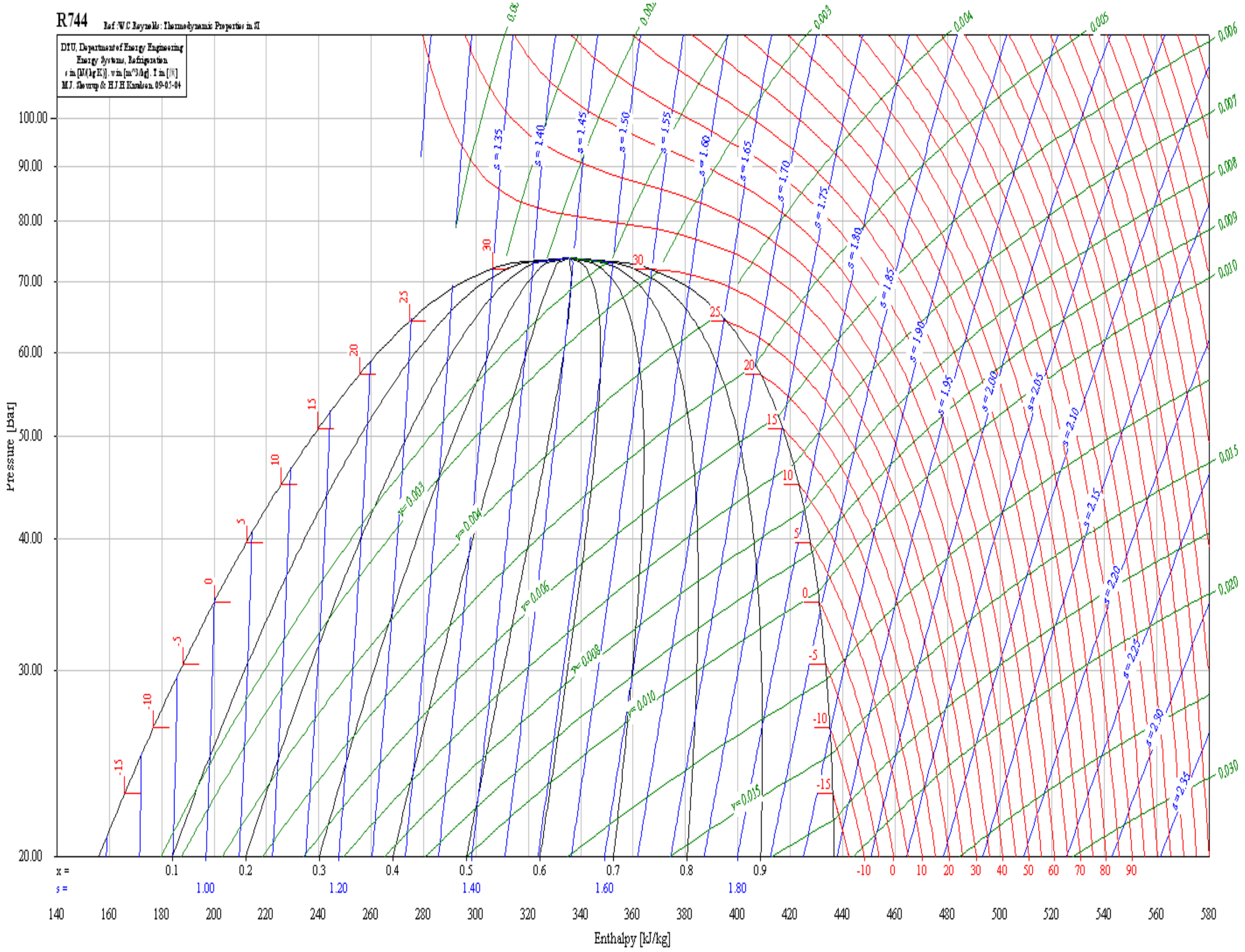
- Complex systems

  Finally, a largely unexplored area of research is the comfort control of more complex multi-component HVAC systems. This whole systems has the potential to benefit for investigation new generation climatronic control unit into the vehicle, the more complex system such as DSPACE system, it is a professional real-time embedded system for all vehicle solution. The approach presented in this thesis of component level modelling makes the transition to more complex systems easy and straightforward, by simply appropriately defining the component input-output relationships to form the overall system model.

Appendix P.1


R134a
Ref D.P. Wilson & R.S Basu, ASHRAE TRANSACTIONS 1988, Vol 94 part 2.
DTU, Department of Energy Engineering
Energy Systems, Refrigeration
s in [kJ/(kg K)]. v in [m^3/kg]. T in [ºC]
M.J. Skovrup & H.J.H Knudsen 09-05-04
Pressure [Bar]
50.00
40.00
30.00
20.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.90
0.80
0.70
0.60
0.50
Enthalpy [kJ/kg]
140
160
180
200
220
240
260
280
300
320
340
360
380
400
420
440
460
480
500
520
540
560
x =
s =
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.00
1.20
1.40
1.60
s = 1.65
s = 1.70
s = 1.75
s = 1.80
s = 1.85
s = 1.90
s = 1.95
s = 2.00
s = 2.05
s = 2.10
s = 2.15
s = 2.20
s = 2.25
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
90
100
120
140
0.0020
0.0030
0.0040
0.0050
0.006
0.007
0.008
0.009
0.010
0.015
0.020
0.030
0.040
0.050
0.060
0.070
0.080
0.090
0.100
0.150
0.200
0.300
0.400
0.500
0.600
v=0.0003
v=0.0004
v=0.006
v=0.008
v=0.010
v=0.015
v=0.020
v=0.030
v=0.040
v=0.060
v=0.080
v=0.100
v=0.150
v=0.200

Appendix P.2


R744
Ref :W.C.Reynolds : Thermodynamic Properties in SI
DTU, Department of Energy Engineering
Energy Systems, Refrigeration
s in [kJ/(kg K)] . v in [m^3/kg] . T in [ºC]
M.J. Skovrup & H.J.H Knudsen. 09-05-04
Pressure [Bar]
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
Enthalpy [kJ/kg]
140
160
180
200
220
240
260
280
300
320
340
360
380
400
420
440
460
480
500
520
540
560
580
x =
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
s =
1.00
1.20
1.40
1.60
1.80
-10
0
10
20
30
40
50
60
70
80
90
-15
-10
-5
0
5
10
15
20
25
30
s = 1.35
s = 1.40
s = 1.45
s = 1.50
s = 1.55
s = 1.60
s = 1.65
s = 1.70
s = 1.75
s = 1.80
s = 1.85
s = 1.90
s = 1.95
s = 2.00
s = 2.05
s = 2.10
s = 2.15
s = 2.20
s = 2.25
s = 2.30
s = 2.35
v= 0.003
v= 0.004
v= 0.006
v= 0.008
v= 0.010
v=0.015
0.00
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.010
0.015
0.020
0.030

Appendix C.1

```
/*------------------------------------T6963.h---------------------------------*/
#define     LcdDataPort  PORTB
//#define     FS        PORTK_BIT5
#define     WR        PORTK_BIT0
#define     RD        PORTK_BIT1
#define     CE        PORTK_BIT2
#define     CD        PORTK_BIT3
#define     RST       PORTK_BIT4

#define TEXT_HOME_ADDR 0000    //starts 0x1700 ending 0x1BFF -> 5120 bytes
#define GRH_HOME_ADDR  2048    //starts 0x0000 ending 0x13FF -> 1280 bytes
#define CG_HOME_ADDR   0x1C00    //starts 0x1C00 ending 0x1FFF -> 128 bytes
#define GraphSize      5120      //4096
#define TextSize       1280      //2048
#define COLUMN         32
#define Graphic        1
#define TXT            0
#define LcmLengthDots  240       //The physical matrix width  (x axis)
#define LcmWidthDots   128       //The physical matrix length (y axis)
#define linechar       30        //240/8=30

// T6963C Commond bit
#define LC_CUR_POS    0x21
#define LC_CGR_POS    0x22
#define SetAddress    0x24
#define LC_TXT_STP    0x40
#define LC_TXT_WID    0x41
#define LC_GRH_STP    0x42
#define LC_GRH_WID    0x43
#define LC_MOD_OR     0x80
#define LC_MOD_XOR    0x81
#define LC_MOD_AND    0x82
#define LC_MOD_TCH    0x83
#define LC_DIS_SW     0x90
#define LC_CUR_SHP    0xA0
#define AutoWrite     0xB0
#define AutoRead      0xB1
#define AutoReset     0xB2
#define DataWriteIncr 0xC0
#define DataReadIncr  0xC1
#define DataWriteDecr 0xC2
```

```
#define DataReadDecr  0xC3
#define DateWriteNull 0xC4
#define DateReadNull  0xC5
#define LC_SCN_RD     0xE0
#define LC_SCN_CP     0xE8
#define LC_BIT_OP     0xF0
//-------------------------------------------------------------------------------//
/*-------------------------------------T6963.c----------------------------------*//
void Dly_ms(int ms) {
int ii,jj;
  if (ms<1) ms=1;
  for(ii=0; ii<ms; ii++)
     for(jj=0; jj<2670; jj++);           //busclk:16Mhz --1ms
}
void sReadFlag(void)
{
  DDRB = 0x00;
  RD=0;
  CD=1;
  WR=1;
  CE=0;
  rCount1=PORTB;

  #asm
    nop
  #endasm
}

void CheckRWCD(void)
{
  for(;;)
  {
  sReadFlag();
     if((rCount1&0x03)==0x03)break;
          CE=1;
  }
}

void CheckAutoWrite(void)
{
     for(;;)
     {
   sReadFlag();
  if((rCount1&0x08)==0x08)break;
  CE=1;
    }
}

void CheckScreen(void)
{
     for(;;)
     {
          sReadFlag();
     if((rCount1&0x40)==0x40)break;
     CE=1;
  }
}
//-------------------------------------------------------------------------
void LcmWriteData(uchar Data)
{
CheckRWCD();
LcdDataPort=Data;
#asm
  nop
#endasm
 DDRB=0xFF;
 RD=1;
 CD=0;
 WR=0;
 CE=0;
#asm
  nop
#endasm
CE=1;
}
//-----------------------------------------------------------------------
void LcmWriteCommand(unsigned char Command)
{
   CheckRWCD();
   LcdDataPort=Command;
   DDRB=0xFF;
   RD=1;
   CD=1;
   WR=0;
   CE=0;
   #asm
      nop
   #endasm
   CE=1;
}
//function sends a byte to display using autowrite mode
void WriteDisplayByteA(unsigned char data){
  CheckAutoWrite();
  LcdDataPort=data;
  #asm
    nop
  #endasm
  DDRB=0xFF;
  RD =1;
  CD =0;
  WR =0;
  CE =0;
  #asm
    nop
  #endasm
  CE=1;
}
//----------------------------------------------------------------------
void LcmWriteCommandWith1Par(uchar Parameter,uchar Command)
{
     LcmWriteData(Parameter);
     LcmWriteCommand(Command);
}

void LcmWriteCommandWith2Par(uchar Parameter1,uchar Parameter2,uchar Command)
{
     LcmWriteData(Parameter1);
     LcmWriteData(Parameter2);
     LcmWriteCommand(Command);
}
//----------------------------------------------------------------------
void LcmInit(void)
{
     LcmWriteCommandWith2Par(0x00,0x00,LC_TXT_STP);
     LcmWriteCommandWith2Par(LcmLengthDots/8,0x00,LC_TXT_WID);
     LcmWriteCommandWith2Par(0x00,0x08,LC_GRH_STP);
     LcmWriteCommandWith2Par(LcmLengthDots/8,0x00,LC_GRH_WID);
     LcmWriteCommand(0xA7);
     LcmWriteCommand(LC_MOD_XOR);
     LcmWriteCommand(0x9C);
     LcmClear(0x00);
}
```

```
//------------------------------------------------------------------------
void LcmClear(uchar FillByte)
{
     uint i = 32768;
     if(FillByte)
     {
             i -= 2*1024;
             LcmWriteCommandWith2Par(0x00,0x08,SetAddress);
     }
     else
             LcmWriteCommandWith2Par(0x00,0x00,SetAddress);
     LcmWriteCommand(0xB0);
     while(i--)
     {
          WriteDisplayByteA(FillByte);
     }
     LcmWriteCommand(0xB2);
}

//-------------------------------------------------------------------------
void LocateXY(uchar x,uchar y,uchar mode)
{
     uint  temp;
     temp= LcmLengthDots/8*y + x;
     if(mode)
     {
     temp+=0x0800;
     }
     LcmWriteCommandWith2Par(temp&0xff,temp/256,SetAddress);
}
//---------------------------------------------------------------------------
void line1(uchar lin)
{
uchar i;
uint StartAddr;
StartAddr=0x800+(uint)lin*linechar;
LcmWriteCommandWith2Par((uchar)(StartAddr), (uchar)(StartAddr>>8), SetAddress);
for(i=0;i<linechar;i++)
{
LcmWriteCommandWith1Par(0xFF,0xc0);
}
}
//function sends a whole display command (inclusive data)
void SendDisplayCommand(uchar cmd, uchar* ptr, uchar lng){
  if (lng == 0)
    LcmWriteCommand(cmd);
  else
    LcmWriteData(ptr[0]);
    LcmWriteCommand(cmd);
}


void SendDisplayStringA(signed char* ptr,uchar lin,uchar column){
  unsigned char c;
  uint iPos;
  iPos = lin * 30 + column;
  gCurRow = lin;
  gCurCol = column;
  LcmWriteCommandWith2Par(iPos&0xFF,iPos/256, SetAddress);
  SendDisplayCommand(AutoWrite,NULL,0);
  while ((c =*ptr++) != '\0'){
     if (c < 0x80)         //translate to display ASCII
              c -= 0x20;
         WriteDisplayByteA(c);
  }
  SendDisplayCommand(AutoReset,NULL,0);
}

void SendDisplayuchar(unsigned char reg, uchar lin, uchar column){
    unsigned char column3;
    float bg;
   bg =reg*100.05/250;
     if(bg<100){
      (void)sprintf(DisVal,"%.1f",bg);
       if(bg<10){
       (void)sprintf(DisVal,"%.2f",bg);
       }
     }
     else
      (void)sprintf(DisVal,"%.0f",Power);

     Dly_ms(150);
     SendDisplayStringA(DisVal,lin,column);
}

void SendDisplayucharAC(unsigned char reg, uchar lin, uchar column){
   uchar AC;
   AC = reg -128;
   if(AC==2)SendDisplayStringA("[On ]",lin,column);
   else if(AC==0)SendDisplayStringA("[Off]",lin,column);
}

void SendDisplayucharIg(unsigned char reg, uchar lin, uchar column){
   if (reg >1)SendDisplayStringA("[On ]",lin,column);
   else SendDisplayStringA("[Off]",lin,column);
}

void SendDisplayucharBar(unsigned char reg, uchar lin, uchar column){
  float Bar;
  unsigned char column3,column4;
  column3=column+3;
  column4=column+4;
    SendDisplayStringA("Bar",lin,column4);
   Bar = (reg *50.9)/255+0.035;
    (void)sprintf(DisVal,"%.1f",Bar);     // %u
     Dly_ms(150);
     SendDisplayStringA(DisVal,lin,column);
     if(Bar<10)SendDisplayStringA(" ",lin,column3);
}

void SendDisplayucharAmTemp(unsigned char reg, uchar lin, uchar column){
 unsigned char column1,column4,column3;
 float AmTemp,NAmTemp;
    column1=column-1;
    column3=column+3;
    column4=column+4;
    SendDisplayStringA("C",lin,column4);
      if (reg<100) {
        NAmTemp= 50-reg *127.5/255 ;
        (void)sprintf(DisVal,"%.1f",NAmTemp);
        Dly_ms(150);
        SendDisplayStringA(DisVal,lin,column);
        SendDisplayStringA("-",lin,column1);
        if(NAmTemp<10)SendDisplayStringA(" ",lin,column3);
        }
        else {
        AmTemp = reg *127.5/255-50;
        (void)sprintf(DisVal,"%.1f",AmTemp);     // %u
        Dly_ms(150);
        SendDisplayStringA(DisVal,lin,column);
        SendDisplayStringA(" ",lin,column1);
        if(AmTemp<10)SendDisplayStringA(" ",lin,column3);
        }
}

void SendDisplayucharTorq(unsigned char reg, uchar lin, uchar column){
    uchar column5,column4;
    column4=column+4;
    column5=column+5;
    Torq = reg * 63.75 / 255;
    (void)sprintf(DisVal,"%.2f",Torq);// %u
     Dly_ms(150);
     SendDisplayStringA(DisVal,lin,column);
     SendDisplayStringA("Nm",lin,column5);
     if(Torq<10)SendDisplayStringA(" ",lin,column4);
}

 void SendDisplayucharPow(uchar lin, uchar column){
     unsigned char column6;
     column6=column+6;
     SendDisplayStringA("Watt",lin,column6);
     Power = Torq * rpm/60;
     if(Power<100){
      (void)sprintf(DisVal,"%.2f",Power);
       if(Power<10){
       (void)sprintf(DisVal,"%.3f",Power);
       }
     }
     else
      (void)sprintf(DisVal,"%.1f",Power);

     Dly_ms(150);
     SendDisplayStringA(DisVal,lin,column);

 }

void SendDisplayucharRPM(unsigned char reg,unsigned char reg1,uchar lin, uchar column){
  unsigned char column4,column5,column6,column7;
    column4=column+4;
    column5=column+5;
    column6=column+6;
    column7=column+7;
    SendDisplayStringA("rpm",lin,column7);
    rpm = (reg1 * 63.75/255) + reg*64;

    (void)sprintf(DisVal,"%.2f",rpm);
```

```
     Dly_ms(150);
     SendDisplayStringA(DisVal,lin,column);
     if(rpm<1000) {
      SendDisplayStringA(" ",lin,column6);
      if(rpm<100){
        SendDisplayStringA(" ",lin,column5);
        if(rpm<10){
          SendDisplayStringA(" ",lin,column4);
        }
      }
     }
}

void SendDisplayString(signed char* ptr, unsigned int Adr){
  unsigned char c,fld[2];
  fld[0]= (unsigned char)Adr;
  fld[1]= (unsigned char)(Adr>>8);
  SendDisplayCommand(SetAddress,fld,2);    //set address
  SendDisplayCommand(AutoWrite,NULL,0);
  while ((c =*ptr++) != '\0'){
     if (c < 0x80)          //translate to display ASCII
              c -= 0x20;
         WriteDisplayByteA(c);
  }
  SendDisplayCommand(AutoReset,NULL,0);
}
/*-------------------------------------------------------------------------------------*
/
```

Appendix C.2

```
void main(void) {
   DisableInterrupts;

  //!!!!!!!!!!!!!!!!!!!
  // for PLL initialization, don't change/remove this code snippet!!!
  // don't even think about it!!!
  if (!(CLKSEL&0x80)) {
    CLKSEL = 0;
    PLLCTL |= 0x40;
    SYNR = 0x02;
    REFDV = 0x01;
    while(!(CRGFLG&0x08));
    CLKSEL |= 0x80;
  }
  //!!!!!!!!!!!!!!!!!!

  initCAN1();                //initialize CAN channel 1
  sInitPort();               //initialize Controller
  initPWM1();                //initialize PWM1 module
  LcmInit();                 //initialize LCD module
  EnableInterrupts;

  for(;;) {

  PWMDTY0=Clima_Buf[0]*102/250;                //Signal convert

     SendDisplayucharAC(mClima1_Buf[6],0,3);
     SendDisplayStringA("AC",0,0);
     SendDisplayucharIg(mZAS_Buf[0],0,24);
     SendDisplayStringA("Ignition",0,15);
     line1(11);
     SendDisplayucharAmTemp(mClima1_Buf[1],2,18);
     SendDisplayStringA("Ambient Temp:",2,0);
     line1(27);
     SendDisplayucharBar(mClima1_Buf[2],4,18);
     SendDisplayStringA("High pressure:",4,0);
     line1(43);
     SendDisplayucharTorq(mClima1_Buf[3],6,11);
     SendDisplayStringA("Compressor:",6,0);
     SendDisplayucharPow(6,20);
     line1(59);
     SendDisplayucharRPM(mGW2_Buf[2],mGW2_Buf[1],8,8);
     SendDisplayStringA("Engine:",8,0);
     line1(75);
     line1(76);
     SendDisplayStringA("%",11,29);
     SendDisplayStringA("STROKE(PWM-duty cycle)=",11,0);
     SendDisplayuchar(Clima_Buf[0],11,24);
     line1(115);
     line1(116);
     SendDisplayStringA("Load Error.F:",15,0);
     SendDisplayStringA("Error Rate.F:",15,15);
  }
}
```

The C- code for initiating PWM mode list at following:

```
/*-------------------------------sets up PWM0 module----------------------------------*/
void initPWM1(){
PWME=0X00;                        //stop PWM output
PWMCTL=0X00;                       //channel separeate
PWMPRCLK=0X66;                    //ClockA=ClockB=Bus/64
PWMCLK=0XFF;                      //Clock source switch between ClockSA,ClockSB
PWMSCLA=0x4;                      //PWMSCLA=PWMSCLB
PWMSCLB=0x4;                      //ClockSA=CLockSB
PWMPOL=0XFF;                      //PWM inital output at high
PWMCAE=0X00;                      //Align right edges
PWMPER0=0x65;                     //Output time duriation =(1/48Khz)x(101+1)=2.127ms
PWME=0X01;                        //enable PWM0
}

/*-------------------------------------------------------------------------------------*/
```

The C- code for initiating CAN1 mode list at following:

```
/*------------------------------sets up CAN1 module---------------------------------*/
void initCAN1() {
  CAN1CTL0 = 0x01;                    // Initialization mode
  while(!(CAN1CTL1&0x01));            // wait for Initialization mode
  CAN1CTL1 = 0x80;                   // enable CAN, crystal clock as source, Loop Back mode off
  CAN1BTR0 = 0x0F;                    // 100k bits/s, RJM = 1
  CAN1BTR1 = 0x25;                   // TSEG1 = 6, TSEG2 = 3, single sampling
  CAN1IDAC = 0x10;                   // 4 x 16 Bit Filters  (only one filter in use) 0x10
  CAN1IDAR0 = (unsigned char)( Clima >> 3);
  CAN1IDMR0 = 0x00;                           //exact match
  CAN1IDAR1 = (unsigned char)( Clima << 5);
  CAN1IDMR1 = 0x00;                           //exact match
  CAN1IDAR2 = (unsigned char)( mClima1 >> 3);
  CAN1IDMR2 = 0x00;                           //exact match
  CAN1IDAR3 = (unsigned char)( mClima1 << 5);
  CAN1IDMR3 = 0x00;                           //exact match
  CAN1IDAR4 = (unsigned char)( mGW2>> 3);
  CAN1IDMR4 = 0x00;                           //exact match
  CAN1IDAR5 = (unsigned char)( mGW2<< 5);
  CAN1IDMR5 = 0x00;                           //exact match
  CAN1IDAR6 = (unsigned char)( mZAS>> 3);
  CAN1IDMR6 = 0x00;                           //exact match
  CAN1IDAR7 = (unsigned char)( mZAS<< 5);
  CAN1IDMR7 = 0x00;                           //exact match
  CAN1CTL0 = 0x00;                            // exit Initialization mode
  while((CAN1CTL1&0x01));                     // wait until Initialization mode has been left
  while((CAN1CTL1&0x10));                     // Wait until synchronized to the CAN bus
  CAN1RFLG = 0xC3;                            // clear CAN Flags
  CAN1RIER = 0x01;                            //enable receiver interrupts
  CAN1RXIDR1_IDE=0;
}
/*------------------------------------------------------------------------------------*
/
```

The C- code for receiving CAN1 message and interrupt subroutines list at following:

```
void ReceiveMessagemClima1(){
  unsigned char* iew = &CAN1RXDSR0;     //set a pointer to the start of message register block
  char i;                               //mClima1 Message
  for (i=0;i<7;i++)
    mClima1_Buf[i] = iew[i];            //read″mClima1″ message from control unit
  CAN1RFLG |= 0x01;                     //clear RXF flag
}
```

```
void ReceiveMessageClima(){
  unsigned char* iew = &CAN1RXDSR0;     //set a pointer to the start of message register block
  char i;                               //Clima Message
  for(i=0;i<7;i++)
    Clima_Buf[i] = iew[i];               //read″Clima″ message from control unit
  CAN1RFLG |= 0x01;                      //clear RXF flag
}

void ReceiveMessagemGW2(){
  unsigned char* iew = &CAN1RXDSR0;     //set a pointer to the start of message register block
  char i;                               //mGW2 Message
  for (i=0;i<7;i++)
    mGW2_Buf[i] = iew[i];                //read″mGW2″ message from control unit
  CAN1RFLG |= 0x01;                      //clear RXF flag
}

void ReceiveMessagemZAS(){
  unsigned char* iew = &CAN1RXDSR0;      //set a pointer to the start of message register block
  char i;                                //mZAS Message
  for (i=0;i<7;i++)
    mZAS_Buf[i]  = iew[i];                //read″mZAS″ message from control unit
  CAN1RFLG |= 0x01;                        //clear RXF flag
}
/*-------------------------------------------------------------------------------------*/
/*--------------------------------CAN1 receiver interrupt-----------------------------*/
#pragma CODE_SEG __NEAR_SEG INTERRUPT_ROUTINES
void interrupt CAN1_IR(void) {
 if(CAN1RXIDR0==124) ReceiveMessagemClima1();
 if(CAN1RXIDR0==225) ReceiveMessageClima();
 if(CAN1RXIDR0==106) ReceiveMessagemGW2();
 if(CAN1RXIDR0==78) ReceiveMessagemZAS();
};

#pragma CODE_SEG DEFAULT

/*-------------------------------------------------------------------------------------*/
```

Bibliography

[1] Volkswagen AG: *Self-Study Programme 208: Air Conditioner in the Motor Vehicle*, Wolfsburg, 1998.

[2] Volkswagen AG: *Konzern-Lastenheft CAN im Komfort- und Infotainment-Bereich,* Wolfsburg, 1998.

[3] PREH automotive: *Funktionhplan: PQ24 Klima-Halbautomat*, Wolfsburg, 1998.

[4] Rasmussen, B.: *Control-oriented modelling of transcritical vapour compression systems,* University of Illinois at Urbana-Champaign, 2002.

[5] Konz, M.: *A Generic Simulation of Energy Consumption of Automobile Air Conditioning Systems*, Nelson Mandela Metropolitan University, 2007.

[6] Holdack-Janssen, H.: *Skript zur Vorlesung: Grundlagen der Fahrzeugklimatisiserung, Kältekreislauf für R134a und Komponenten*, Fachhochschule Wiesbaden.

[7] Motorola Inc: 9S12DT128DGV2/D. *MC9S12DT128 Device User Guide,* V02.15, 05 OCT 2005.

[8] Motorola Inc: S12DBMV4/D. *Background Debug Module (BDM) V4,* Rev.4.05, 05 OCT 2004.

[9] Motorola Inc: S12MSCAN2/D. *MSCAN Block Guide V02.15,* 15 JUL 2004.

[10] Tobi T. & T. Hanafusa.1991. A practical application of fuzzy control for an air-conditioning system. *Int. J. of Approximate Reasoning,* (5):331-348.

[11] Albert T. P. & W. L. Chan & T. T. Chow, etc.1995. A neural network based identifier/controller for modern HVAC control. *ASHRAE Transactions,* 101(2):14-31.

[12] Huang S. & R.M. Nelson. 1994. Rule development and adjustment strategies of a fuzzy logic controller for an HVAC system: part one—analysis. *ASHRAE Transactions,* 100(2):841-850.

[13] Huang S. & R.M. Nelson. 1994. Rule development and adjustment strategies of a fuzzy logic controller for an HVAC system: part two—experiment. *ASHRAE Transactions,* 100(2):851-856.

[14] Huang S. & R. M. Nelson. 1999. Development of a self-tuning fuzzy logic

controller. *ASHRAE Transactions: research,* 105(part1):206-213.

[15] Qingwei Chen & Jianfang jiang & Weili Hu. 1994. The application of laminar hierarchical fuzzy control method in automatic control of HVAC. *Journal of Nanjing University Science and Technology,* (4):14-18.

[16] Guoliang Ding & Chunlu Zhang & Tao Zhan. 2000. Fuzzy modeling method of thermodynamic performance of refrigeration compressor. *Journal of Shanghai Jiaotong University,* 34(9):1298-1300.

[17] Jiejia Li & Jinxiang Pian & Donghua, Zhu. 2004. Application of adopting fuzzy neural network adaptive control in VAV temperature control system based on wavelet neural network. I*nstrument technique and sensor,* (7):37-38.

[18] Hui Liu. Rules self-organizing fuzzy control method of time delay thermal systems. *Harbin Institute of Technology,* 2004.06

[19] Hui Liu & Jili Zhang & Dexing Sun. 2005. Two-stage rules extraction and self-organizing fuzzy control method. *Journal of Harbin Institute of Technology,* 37(9):1189-1192.

[20] Hui Liu & Jili Zhang & Dexing Sun. Test research of self-organizing fuzzy control method in air system of air conditioning. HV&AC.

[21] Guoqing & Cao Chengzhi Lou & Dawei An. 2004. Application of fuzzy self-tuning PID control to air conditioning systems.HV&AC, 34(10):106 -109

[22] Aiguo Wu & Chunyan Du & Xiaoqiang Song. 2004. A fuzzy controller with self-optimizing parameters and Its application in the central air-conditioning system. *Engineering science,* 6(11):84-87.

[23] Dawei An & Wangjiang Wang. 2004. The simulation and research of the fuzzy neural network control in the temperature of air conditioning system. *Refrigeration and air conditioning,* 4(1):41-44.

[24] Huiping Chen & Yannian Rui & Juntao Li. 2004. Research on method of intelligent inverter air-conditioner based on fuzzy genetic algorithm. *Journal of Suzhou Institute of Silk Textile Technology,* 24(5):111-116. (In Chinese)

[25] Mamdani E. H. & Assilian S. 1974. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Machine Studies,* (7):1-13.

[26] Tzafestas S.G. 1994. Fuzzy systems and fuzzy expert control: an overview. *The Knowledge Engineering Review,* 9(3):229-268.

[27] Jili Zhang & Jinping Ou & Daren Yu. 2003. Method of rule self-tuning fuzzy control based on trajectory performance of phase plane. *Control theory & applications.* 20(4): 607-611.

[28] Jili Zhang & Jinping Ou & Daren Yu. 2000. The Method of the Real-time Multi-step Prediction and Control of BP Network for the Delayed-time System. *Journal of Harbin institute of technology,* (2): 82-86.

[29] Liang Wang, Reza Langari. Building Sugeno – type Models Using Fuzzy Discretization and Orthogonal fuzzy controller of directily modify control rules [J]. Acta electronic sinica, 1992, 20(2):10-15. *Parameter Estimation Techniques [J]*.IEEE Trans. 1995, 3(4):454-458. (In Chinses)

[30] Lee, C.C., "Fuzzy Logic in Control Systems: Fuzzy logic controller - Part I", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20 No. 2, pp. 408-418, 1990.

[31]J.S.Roger Jang, Ned Gulley. MATLAB Fuzzy Logic Toolbox. [Online] 5 May 2008. http://www.mathworks.com

[32] T.J. Skinner, R.L. Swadner, V-5 automotive variable displacement air conditioning compressor, *Soc Automotive Eng* 1985.

[33] Andreas Herp, Michael Herbert, Oliver Falkner. Semi - Synthetic Regression Tests with Real-World Data, Technical Article [Online] 5 May 2008. http://www.vector.com/

[34] Anthony Faucogney, St éphane Martin.Verification, Virtual validation with CANoe / Matlab, Technical presentation [Online] 6 May 2008. http://www.vector.com/

[35] Wolfgang Hartig, Albert Habermann, Jurgen Mottok. Model – Based Testing for Better Quality. Technical Article [Online] 1 May 2008. http://www.vector.com/