

POLITECNICO DI TORINO



Master's Degree in Aerospace Engineering

Master Thesis

**Simulation model for the
Environmental Control System
of a turboprop two-seater aircraft**

Advisor:

prof. Paolo Maggiore

Candidate:

Eleonora Fico

UTC Aerospace Systems supervisor

Ing. Claudio Carpignano

Ing. Gabriele Maione

2017 - 2018

Acknowledgements

I would firstly like to thank my thesis advisor prof. Paolo Maggiore of *Politecnico di Torino*, for giving me the opportunity to have this learning experience of thesis in a company and for guiding me in its realization.

I wish to thank UTAS and my company supervisor Claudio Carpignano for allowing me to have this good internship experience and to carry out this thesis.

I would expecially like to express my deep gratitude to Ing. Gabriele Maione for supporting and helping me all along and for his patient guidance, enthusiastic encouragement and useful critiques of this master thesis work.

A very special thank goes to Zaf for always being there for me and for supporting me every day. Thank you for always believing in me, expecially when I do not believe in myself. Thank you for always making everything better and funnier. Thank you for being my person, thank you for being my rock.

I would also like to thank all my friends for making memorable this University experience. I am exceptionally grateful to Angela and Michele for our wonderful and unbreakable friendship, no matter how many kilometres will be between us. Particular thanks goes to Francesca, the first person I met in Turin, and to Biagio, Livia, Mattia e Pippi for always being there and for becoming my family in Turin. I would express my very profound gratitude to my mother and my father for providing me with unfailing support and continuous encouragement throughout my years of study, through the process of researching and writing this thesis and throughout my life. This accomplishment would not have been possible without them.

Special thanks are due to all my family, uncles and cousins, but above all to my grandparents Anna e Vincenzo and to my grandfather Angelo for sticking by me, although this far, and for encouraging me in all of my pursuits.

Abstract

This thesis project has been performed at *UTC Aerospace Systems*, in the factory of Luserna San Giovanni (TO), in cooperation with *Politecnico di Torino*.

The project is focused on the modeling at high level as lumped parameter of the Environmental Control System of an existing turboprop and two-seater aircraft in the Matlab/Simulink environment. In detail, particular attention has been paid to modeling of each component of the system and their connection in order to simulate the correct performances of the system in different conditions.

The need of the creation of this simulation model arises from the fact that the simulation modeling represents the realization and analysis of a digital prototype of a physical model to approximate different working conditions and to predict the system performances in the real world. As a consequence, it may help designers and engineers to implement the necessary changes in system's layout or performances also during preliminary design phases and to understand why a failure has occurred and how to solve issues (trouble shooting). At the same time, it can be useful to predict under which conditions and in which way the system or a part of it will behave.

Moreover, the Environmental Control System is a fundamental system on board an airplane: it is a survival system since it provides air in the cabin at temperature and pressure suitable for the human survival. As a consequence, a simulation model is utmost important to predict in advance the possible performances in every functioning condition or to solve in a timely manner any occurred problem. This task is usually carried out by a test rig especially designed.

The availability of a good simulator combined with the test rig allows to realize numerous engineering tasks much more effective and quick.

Thus, the realization of a simulation model is fundamental: it represents a significant asset for the company. In fact, after setting the required parameters, the simulation model can run autonomously without any human involvement or cost and it can also provide much faster results.

Moreover, once a simulation model has been implemented for an existing Environmental Control System, it may be validated by comparing its results with those obtained with the system test rig. Therefore, after its validation, it is possible to use the model also to simulate other similar systems, adapting it to the new system just with few changes.

This is particularly important, especially if a company has to design a new system, since the model allows to predict in advance the feasibility of the project and possibly its performances also without a physical test rig.

As far as concerns the thesis development, after a little introduction to the Environmental Control System, with particular attention to its importance on board and to the components that may be present in an aircraft system, the description of the test case is explained with an analysis of its principal features. A general schematic of the system is provided too, in order to help the reader to understand how the system is made of with all its main components.

The subsequent two chapters provide a detailed analysis of the two main subsystems modeling. Therefore, a detailed description of all the system components modeling and an explanation of the used mathematical equations are provided. Thereafter, the cockpit model and the control law used on board are described.

Finally, simulations in two different extreme working conditions have been performed and the results obtained are compared and validated with the test rig results in the same conditions.

Contents

Acknowledgements	III
Abstract	V
1 Introduction	1
1.1 Environmental Control System	1
1.1.1 Theoretical reasons	1
1.1.2 ECS general design	3
1.1.3 Engine bleed	4
1.1.4 Refrigeration systems	6
1.2 Matlab-Simulink Tool	9
2 Test case	11
2.1 System components description	12
2.1.1 Bleed Subsystem	12
2.1.2 Temperature Control Subsystem	16
2.1.3 Vapor Cycle Subsystem	17
2.2 System modeling	18
2.2.1 General setting of modeling	19
2.2.2 Components modeling	20
3 Bleed Subsystem modeling	21
3.1 ECS PRSOV	21
3.2 Precooler - By-Pass group	25

3.2.1	ECS Heat Exchanger	25
3.2.2	Temperature Control By-Pass Valve	31
3.2.3	Subassembly	35
3.3	Flow Modulating RSOV	39
3.4	Cabin Outlet Valve	42
3.5	Bleed Subassembly	47
4	Vapor Cycle Subsystem modeling	51
4.1	The mathematical model	53
4.1.1	Main equations	54
4.1.2	Density implementation	60
4.2	Cycle implementation	64
4.3	Moisture modeling	78
5	Complete ECS model and results	83
5.1	Cockpit model	83
5.2	Control law	87
5.3	Results	89
5.3.1	Heating	91
5.3.2	Refrigeration	95
6	Conclusions	100
	Bibliography	102

Chapter 1

Introduction

The main goal of this Thesis project is to create a simulation model of the Environmental Control System of an existing aircraft in the Matlab-Simulink environment. As a consequence, this chapter purpose is to introduce the latter two topics to the readers. Therefore, the following pages will explain the need for an Environmental Control System inside an aircraft, the different kinds of the existing systems and the differences between them. Finally, there will be a brief description of the tool used to realize the simulation model.

1.1 Environmental Control System

1.1.1 Theoretical reasons

The expression “Environmental Control” is generally referred to the complex handling system of the air in a specific ambient, in order to keep the crew and the passengers in comfortable conditions during all the aircraft flight phases, as far as concern temperature, pressure and moisture.

It is a primary need on board of an aircraft, because, as well as the passengers comfort, it provides a comfortable working environment to the aircraft technical staff. In so doing, it is possible to avoid a dangerous reduction in their attention and individual performances.

Moreover, modern ECS systems are able to provide suitable conditions for the avionic, fuel, electrical or hydraulic systems, too. In fact, these systems are often too hot and, as a consequence, they have to be cooled to guarantee their optimum performance.

The need to modify the thermo-physical properties of the air depends on the remarkable evolution of aircraft technologies. In fact, modern airplanes that fly at very high altitude at the same speed use less fuel compared to a lower altitude flight. In other words, an aircraft is more efficient at higher altitudes.

The atmospheric pressure and temperature decrease with increased altitude and, if the altitude is too high, some dangerous body issues may emerge.

As far as concerns the pressure, it is essential for both breathing and mechanical effects on the body. Generally, the majority of civil airplanes obtains the correct pressure to survive with an higher pressure on board than the outside one. Therefore, the pressure on board is maintained at an equivalent altitude of 8000 ft, as shown in Figure 1.1.

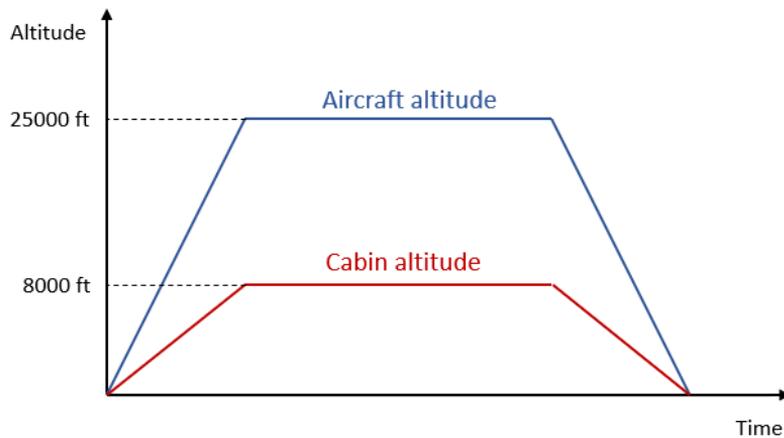


Figure 1.1. Difference of pressure between external and cabin altitude

On the other hand, in military aircraft the pressure is maintained at the equivalent altitude of 20000 ft since the correct pressure is provided to pilots through oxygen masks.

As far as concerns the temperature, instead, it is generally considered acceptable for values from 20°C to 24°C during the summer period and from 18°C to 22°C during the winter period, with a relative humidity from 30% and 70%.

All the conditions previously described have to be maintained on board in every flight condition. It means that they have to be guaranteed for both hot or cold days and for both on ground operations or in cruise.

1.1.2 ECS general design

The ECS design has to respect the required heat charge by balancing the effects of high heat charges on board due to the conduction from external air, the solar radiation, the electronic equipment and the body heat.

The main problem in cooling systems is the heat sources. In fact, we need to find a heat source to successfully cool passengers and eventually aircraft systems during both flight phases and on ground.

Heat has to be transferred by these sources to the ECS system and, then, rejected from the aircraft. Generally, the outside air is used either directly as ram air or indirectly as air bled from the engines. However, in the latter case, this air is usually at a higher temperature than the required one for passengers cooling, therefore some heat pumps are required.

There are two different options to obtain the requested temperature inside the cabin: it is possible to act on the air temperature and/or on the air flow rate entering the cabin.

The cabin inlet temperature cannot much change, in fact there is a maximum and a minimum temperature based respectively on the bleed air and on the cold air unit (CAU). Therefore, the parameter that allows to have a greater contribution to the temperature variation is the flow rate.

The chart in Figure 1.2 shows the variation of heat flow against the cabin temperature, where the slope of the straight line depends on the mass flow rate. Therefore, the chart shows the variation in the heat flow generation with the system for different mass flow rate entering the cabin.

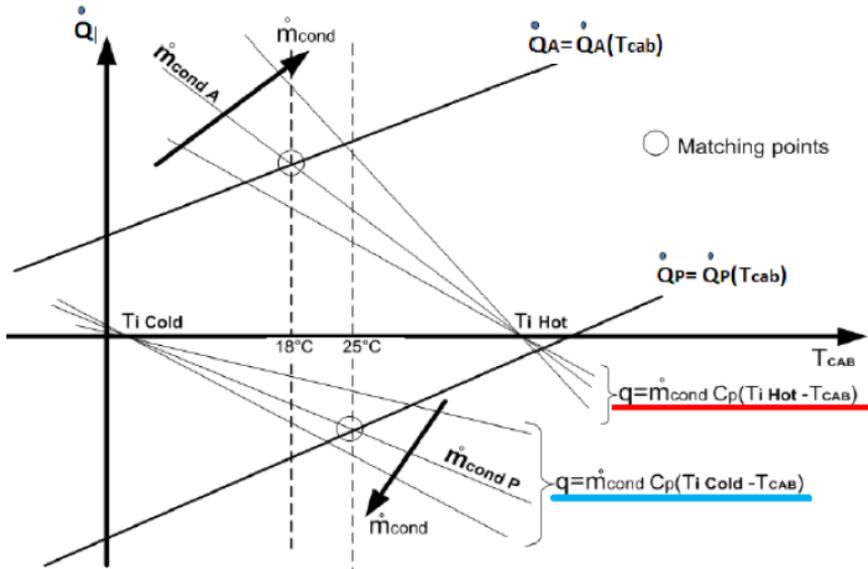


Figure 1.2. Chart of the heat flow against the cabin temperature

1.1.3 Engine bleed

For both civil and military aircraft, the main source of conditioning air is the bleed air from the engine compressor at high pressure that provides a source whenever the aircraft engines are running.

There are two different bleed air systems:

- Open loop
- Closed loop

The open loop ECS continually bleed large amounts of air from the engines, refrigerate it and, after that, use it to cool the passengers and eventually some aircraft equipment, before dumping the air overboard.

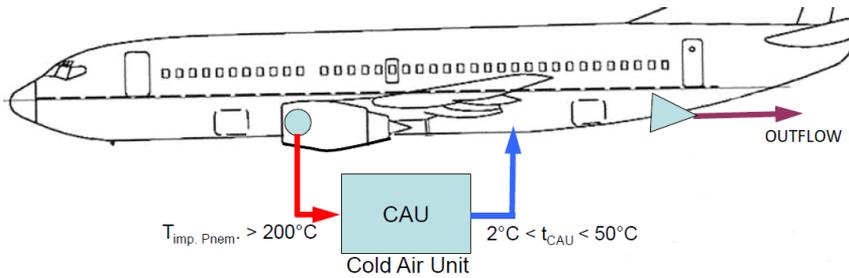


Figure 1.3. Open loop general schematic

The temperature of the air entering the CAU is generally greater than 200°C , therefore it has to be reduced before entering the cabin. In this respect, there are two temperature limits:

- Minimum temperature (maximum cold flow): 2°C . This is the lower limit because a lower temperature could determine ice formation and, as a consequence, it could obstruct the system ducts or damage some components
- Maximum temperature (maximum hot flow): 50°C . This is the upper limit to avoid possible burn in case of a direct contact of human skin with the cabin nozzle

Moreover, modern open loop ECS are evolving, including a partial recirculation of cabin air, in order to reduce the air mass flow rate bled by the engine. However, the reduction of the flow rate is possible under a condition: the temperature downstream the CAU have to be capable to be lower than 0°C . Generally this temperature is about -20°C . In so doing, as shown in Figure 1.2 it is possible to introduce a lower mass flow rate at the same heat flow in the cabin and it determine to have small size CAU.

This temperature is obviously too low to enter directly the cabin, in fact it is mixed with a bled flow rate at higher temperature to obtain an optimum temperature for the passengers conditioning.

On the other hand, the closed loop ECS does not dump the air overboard once it has been used for cabin conditioning, but refrigerate it and recycle it to be used again. In so doing, bleed air is used only to provide pressurization and this means

that such a system uses considerably less engine bleed air than an open loop system, increasing engine performances.

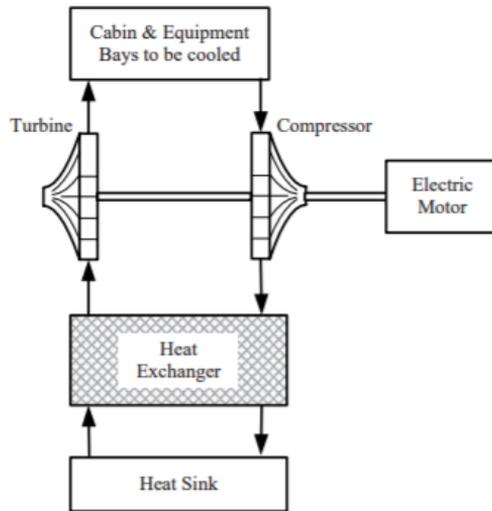


Figure 1.4. Close loop general schematic

Closed loop ECS have actually been used in a few aircraft applications. In fact, there are numerous practical difficulties of collecting and reusing the conditioning air, and they tend to be heavier and more expensive than equivalent open loop ECS, too. As a result the latter systems, are actually most common because of the use of air cycle refrigeration to cool engine bled air.

1.1.4 Refrigeration systems

There are two different kinds of refrigeration system:

- Air Cycle Refrigeration System
- Vapor Cycle Refrigeration System

The main difference between these two configurations is that the Vapor cycle is not able to autonomously pressurize the cabin, but it needs to be complemented

by a bleed line or to have a dedicated compressor to do it.

The Vapor Cycle System is a closed loop system characterised by the use of refrigerants with particular thermo-physical properties that make them suitable to the temperature and pressure ranges used in these systems.

These refrigerants are characterised by a high latent heat of vaporization and a high equilibrium vapor pressure. Therefore, if a suitable evaporator is going through air at a given temperature, the following refrigerant evaporation determines the reduction of the air temperature. In so doing, the refrigerated air can be used for cabin or eventually the electronic equipment cooling.

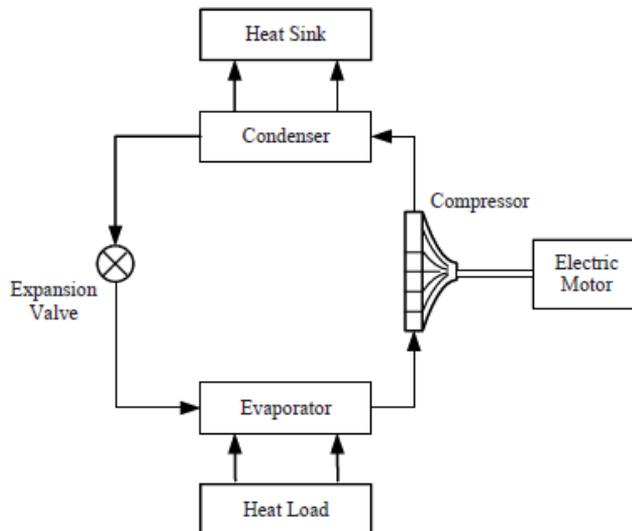


Figure 1.5. Vapor Cycle System general schematic

Figure 1.5 shows the general schematic of a Vapor Cycle System which typically consists of just four components: compressor, condenser, thermostatic expansion valve and evaporator. The circulating refrigerant enters the compressor where it is compressed to a high pressure, resulting, as consequence, at a higher temperature, too. This hot and compressed vapor is then cooled in a condenser where the heat is rejected to a heat sink. Thereafter, the refrigerant flows through an expansion valve where its pressure is reduced. Finally, it goes through an evaporator where

the heat load is absorbed by the refrigerant evaporation.

The Vapor Cycle Systems are generally very efficient, with performance coefficients about five times higher than a comparable Air Cycle System. Nevertheless, their applications are actually limited because of their limited temperature range and heavy weight.

On the other hand, the Air Cycle Systems generally have a different functioning principle. In these systems the heat is removed by a heat exchanger from compressed air which goes through a turbine driving a compressor: energy is transferred determining a temperature and pressure reduction.

The Air Cycle Systems are often combined with bleed systems and in these cases they are used to cool the engine bleed air down to the required temperatures in order to obtain the correct cabin and equipment conditioning.

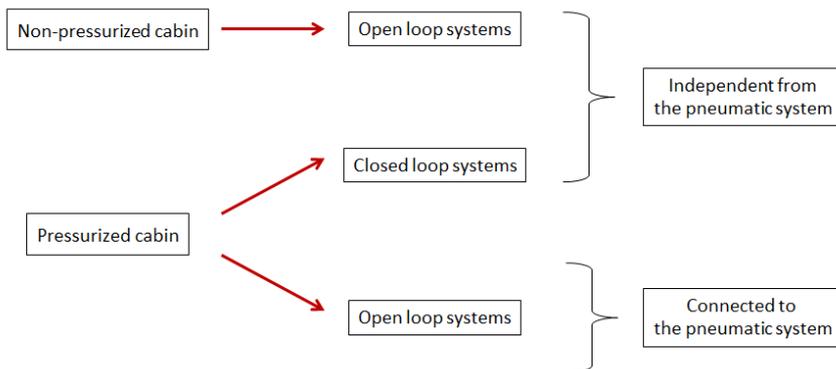


Figure 1.6. Different categories of Air Cycle System

The Air Cycle Systems can be divided in three broad categories, as shown in Figure 1.6:

- aircraft with non-pressurized cabin: this kind of air cycle system is an open loop system in which outdoor air enters the cabin and, thereafter, it flows outside the aircraft. In case of bleed air required, a heat exchanger is introduced. Therefore, a connection to the pneumatic system is not required
- aircraft with pressurized cabin and closed loop cycle: in the pressurized cabin, a closed loop cycle can be introduced. This kind of system use the same

recycling air after a chemical regeneration: using a dedicated compressor, the outdoor air is introduced in cabin at a pressure suitable to balance the losses due to the altitude. Furthermore, about temperature some components for the heating and refrigeration are used. Since a dedicated compressor is used, a connection to the pneumatic system is not required

- aircraft with pressurized cabin and open loop cycle: in the pressurized cabin, a open loop cycle can be introduced, too. In this kind of system an air flow rate is introduced in the pressurized cabin and after that it is ejected through the pressurization valve. A connection to the pneumatic system is required in this case, because the air has to be introduced in cabin at a pressure higher than the outdoor one. The pneumatic air is also at a very high temperature, thus it has to be refrigerate before enter the cabin: a CAU (Cold Air Unit) is required

The Air Cycle System are lighter, more compact and cheaper than the Vapor Cycle System. Nevertheless, very large air flows and high heat loads are required: as a consequence, these systems typically need large diameter ducts with the corresponding limitations in the system installation on board the aircraft. Moreover, because of the constant need for ram air cooling, large aircraft drag penalties incur, too.

1.2 Matlab-Simulink Tool

The thesis project has been carried out in the Matlab/Simulink environment.

Matlab is an acronym for MATrix LABoratory and it is a numerical computing environment whose proprietary programming language is developed by *MathWorks*. This tool is useful to manage matrix data, in order to generate algorithm or solve problems with numerical method.

On the other hand, *Simulink* is a graphical programming environment generally used to analyze, model and simulate multidomain dynamical systems: it allows to represent and simulate a mathematical model that describes a physical system. It is strongly integrate with the *Matlab* environment: *Simulink* can also be scripted

from *Matlab* or drive it. Its primary interface is a graphical block diagramming tool and a set of block libraries.

Moreover, *Simulink* allows to obtain one of the major advantages of the simulation and analysis of dynamic systems: it allows to quickly analyze the response of complicated systems that may be prohibitively difficult to analyze analytically. In fact, it is able to numerically approximate the mathematical models solutions, too complex to be solve “by hand”.

Simulink is typically used in automatic control, digital sign processing or Model-Based-Design applications. It has large applications in the aerospace world, too.

Chapter 2

Test case

The main aim of this thesis project is to realize a simulation model of an ECS (Environmental Control System) of a specific aircraft in the Matlab-Simulink environment.

The examined aircraft is a tandem two-seater and turboprop aircraft, that needs to be designed with an environmental control system. The latter is a primary system, therefore it has a wide range of purposes:

- Heating or cooling the cockpit environment in accordance with pilot indications
- Regulating cabin pressure in line with the flight altitude

The considered ECS aircraft has been split in three different subsystems to regulate the cabin temperature. These three subsystems are:

- Bleed Subsystem
- Temperature Control Subsystem
- Vapor Cycle Subsystem

The main aim of the Bleed Subsystem is to provide warm and pressured air, whose temperature is modulated by Temperature Control Subsystem in accordance with the pilot's requests. Consequently this air could be placed in the cockpit. Moreover, depending on the heat load a Vapor Cycle Subsystem can be switched on by

the pilot. Thus, if the cabin temperature appears to be too hot, the pilot will turn on the Vapor Cycle Subsystem to introduce colder air. In so doing, the environmental temperature will be modulated.

The three subsystems work alongside and independently, allowing to introduce in the cockpit two different air flows at different temperatures. These flows are mixed together to obtain temperature conditions requested by the pilot. In detail, the Temperature Control Subsystem task is exactly to regulate air temperature exiting by the Bleed Subsystem, in order to obtain the cabin specific conditions requested by pilot. On the other hand, the Vapor Cycle Subsystem task is to cool Bleed Subsystem air, too hot to enter directly the cabin.

2.1 System components description

Figure 2.1 shows a schematic of the general architecture of the aircraft environmental control system.

As you can see from the previous schematic, the two main subsystems (Bleed Subsystem and Vapor Cycle Subsystem) show completely different components, structures and functions. Therefore, various subsystems will be analyzed independently and described in each part, in order to provide a better understanding.

2.1.1 Bleed Subsystem

The main purpose of this part of the system is to provide the ventilation and heating functions to the system using the air bled from the aircraft engine compressor. Since the temperature and the pressure of the bled air are very high, it is necessary to regulate it to obtain the temperature and pressure requested.

Therefore, the Bleed Subsystem is made up of several components that are located along the red line, as shown by the schematic in Figure 2.1.

Following, these components are listed with the reference of the respective location of each one inside the schematic (the reference number for each component in the schematic in Figure 2.1 is indicated in the table below).

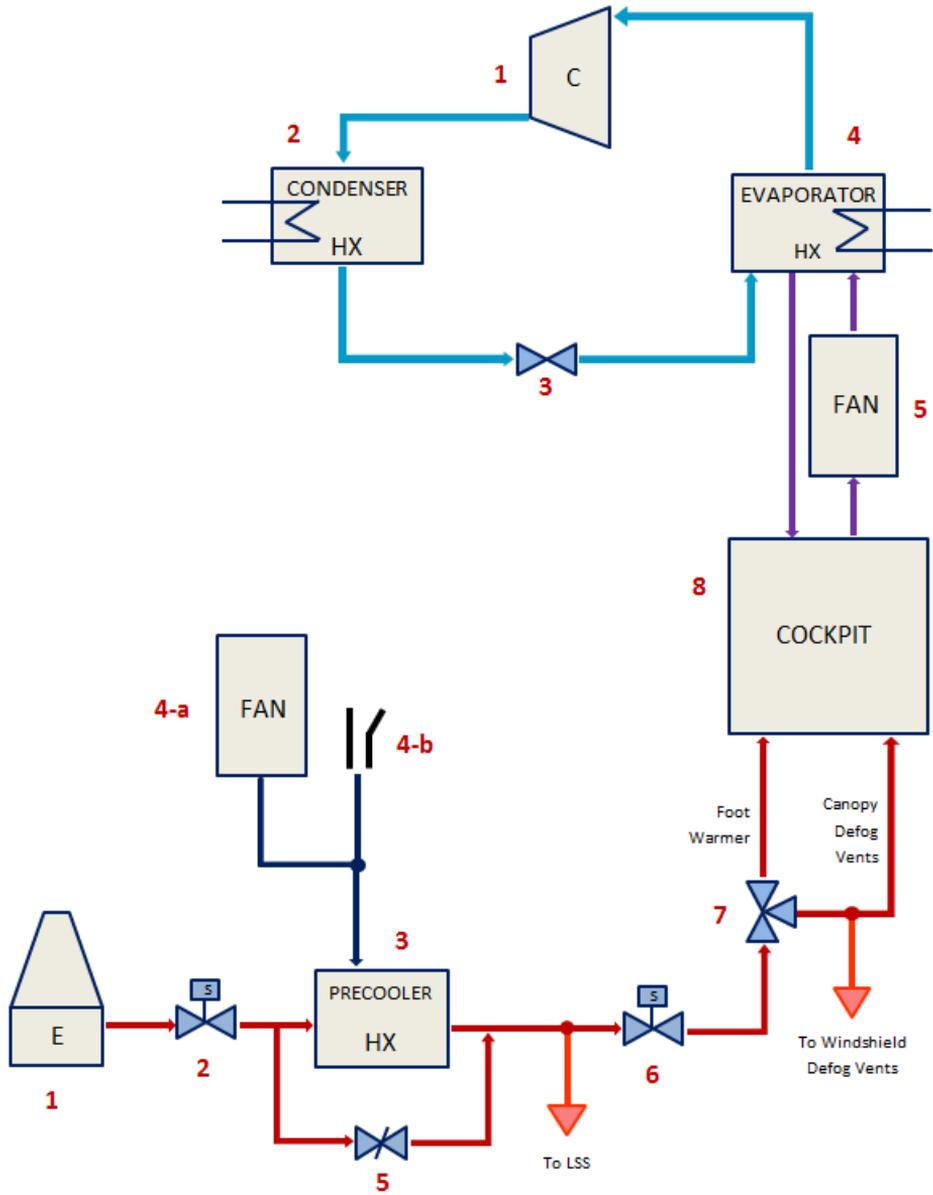


Figure 2.1. System schematic

Reference	Component
1	Engine
2	ECS PRSOV
3	ECS Heat Exchanger
4-a	Precooler Fan
4-b	Air Intake
5	Temperature Control By-Pass Valve
6	Flow Modulating RSOV
7	Cabin Outlet Valve
8	Cockpit

Table 2.1. Bleed Subsystem components legend

The hot and pressurized air is bled from the aircraft engine and then it is routed to the ECS PRSOV, a Pressure Regulating and Shut-Off Valve. It is a solenoid valve which provides two core functions:

- Regulating the downstream pressure at a fixed value, or at the maximum value defined in the performance design: in this way, pressure could be lower than the initial condition and it would determine a reduction of downstream air temperature inside the valve
- Acting as a safety device to stop the bleed air flow any time there are anomalous conditions either on the engine side (loss of power) or in the downstream system (hot air leakages, smoke, etc.)

The flow is then routed toward the Precooler, an Heat Exchanger whose main purpose is to cool down the hot air flow coming from the PRSOV valve. Inside the Precooler the bleed air flow is cooled down by the ram air introduced by an Heat Exchanger Fan on ground or by a dynamic air intake in flight. The temperature of this air flow is low enough to provide an appropriate heat exchanging and in the heat rejection it has to guarantee a maximum temperature of the bleed air of 70°C.

In parallel with this heat exchanger, there is also a Temperature Control By-Pass Valve. It is a throttle valve whose aim is to bypass the heat exchanger in order to have a hot line with the same temperature and pressure conditions which can be observed downstream the PRSOV valve.

Hot air flow rate can flow through the line along the Temperature Control By-Pass Valve depending on the opening of the throttle. In so doing, it is possible to modulate the air downstream flow temperature of Precooler – By-Pass subassembly. Flows arriving from the latter are mixed in another one that will have a variable temperature, depending on the intensity of the flow rate through the Temperature Control By-Pass Valve.

As Figure 2.1 shows, about 40% of mixed air flow resulting in output of Precooler – By-Pass subassembly is taken off and deployed for survival systems, such as the pilot oxygen mask. The purpose is to ensure the life support system in any operational condition and also in case of system failure.

The remaining percentage of the air flow goes through the FMRSOV, the Flow Modulating Regulating and Shut-Off Valve. The latter is a valve whose main purpose is to modulate the hot air flow rate; after passing through the Cabin Outlet Valve, this flow can be introduced directly inside the cockpit. The Flow Modulating Regulating and Shut-Off Valve is provided of a proportional solenoid whose aim is to control the pressure in the reference chamber. In particular, the pilot can select one of the four possible choices: each level corresponds to a different value of solenoid current. Depending on the level chosen by the pilot, a specific depressurization of the reference chamber is produced. Consequently, this entails a higher or lower reduction of air flow rate through the valve: the higher is the current value, the more significant is the depressurization inside the reference chamber resulting in a higher bleed flow rate.

Before entering the cockpit, hot air flow goes through one final system component, the Cabin Outlet Valve, named COV. It is an electro actuated three-way valve consenting a proper distribution of the air flow into the cabin environment, depending on the pilot and aircraft's needs.

The valve three-ways splits the air flow. Normally the major portion (about 70%) of inlet air flow goes into the cabin environment to provide the pilot comfort; the remaining portion (about 30%) goes to the canopy defog vents. When required, it is also possible to deploy all of the hot air flow to the canopy defog vents by closing the air flow passage towards the cockpit.

2.1.2 Temperature Control Subsystem

The main purpose of this subsystem is to regulate Bleed Subsystem air temperature entering the cabin, in accordance to the temperature value set by the pilot. It is generally provided by a Temperature Controller and Selector, through two different control modes.

The first control system mode is an automatic control of the cabin temperature. It compares the temperature value set by the pilot during the flight with the weighted average cockpit temperature value, obtained considering 25% of the duct inlet temperature read by a conditioned air duct temperature sensor and 75% of the cockpit discharge temperature read by a cabin temperature sensor.

The error between these two different temperature values will determine the correct opening (or closing) of the throttle valve, which is the Temperature Control By-Pass Valve. In particular, a lower temperature in the cabin environment is needed if the cockpit temperature is higher than the one set by the pilot. Therefore the valve will tend to close, so a lower flow rate will flow through the bypass duct. As a consequence, the resulting mixed flow at the end of the Precooler – By-Pass subassembly will be lower than the one of the previous condition. On the other hand, when the pilot sets the desired temperature and the cabin temperature value is lower, the throttle will tend to open and the resulting mixed flow temperature at the end of the Precooler – By-Pass subassembly will be higher.

The second control system mode is a manual control of the vents temperature and, in this case, only the value read by conditioned air duct temperature Sensor is considered to control the cabin temperature. This control mode has to be used only in case of failure of the cabin temperature sensor, determining that the other control system cannot be used.

The Temperature Control Subsystem has a fundamental role in temperature regulation. In fact, this subsystem works regulating only temperatures obtained by the Bleed Subsystem. On the other hand, the refrigerant function, carried out by Vapor Cycle, is independent from the rest of the system and it does not allow a direct temperature regulation. Thus, a control of Vapor Cycle temperature by Temperature Control Subsystem is not necessary.

Bleed Subsystem and Vapor Cycle Subsystem work at the same time and both of them have an impact on the cockpit temperature variation. The difference is

that the air flow from the Bleed Subsystem can be modulated in order to obtain the desired cabin temperature, whereas the Vapor Cycle Subsystem can only be modulated on two levels of air flow, flowing through the evaporator.

2.1.3 Vapor Cycle Subsystem

Finally, the Vapor Cycle Subsystem is a standard refrigerant cycle. Therefore, it consists of similar components of any other similar cycle, as it is possible to see along the light blue line in the schematic in Figure 2.1.

Following, these components are listed with the reference to the respective location of each inside the schematic (the reference component number in the schematic in Figure 2.1 is indicated in the table below).

Reference	Component
1	VCS Compressor
2	Condenser
3	TXV Valve
4	Evaporator
5	Evaporator Fan

Table 2.2. Vapor Cycle Subsystem components legend

The refrigerant used for this cycle is R134a, named Tetrafluoroethane, commonly known as Freon. It is a non-flammable compressed or liquefied gas used primarily as a high temperature refrigerant for air conditioning. It is also a kind of refrigerant with a low environmental impact, often used in compression refrigerant cycles. The refrigerant, in its gaseous state, has to be compressed by a compressor with a variable cylinder capacity. The compressor increases refrigerant pressure and sends it out to the condenser. The latter is a traditional condenser whose core is constituted by four layers in an arrangement of three plus one: the first three layers are in parallel and they are in series with the fourth one. This condenser rejects the refrigerant heat to the ambient air. The latent heat is removed from the refrigerant, going through the condenser, and then it is transferred to the external environment. Because of the reduction of enthalpy, the refrigerant condenses from

its gaseous state to its liquid one.

Therefore, liquid refrigerant goes through the TXV valve, named as Thermal Expansion Valve. This thermostatic valve is part of the evaporator subassembly which regulates the refrigerant flow through the evaporator core, in order to obtain the superheat requested. The superheat guarantees that R134a is evaporated entirely without any traces of liquid when it goes out and then enters the compressor. It is fundamental because the compressor works at the gaseous phase and a liquid ingestion could damage it. Thus, in order to avoid a malfunction or, worse, a breakage of the compressor, it is important to ensure a total liquid evaporation. This is possible by determining pressure and temperature of the fluid going out the heat exchanger. These two values have to be compared with the refrigerant saturation characteristics in order to obtain the correct superheat temperature value.

The main aim of the TXV valve is to open or close the orifice with an appropriate plunger position. This ensures the flowing of a higher or lower refrigerant flow rate in order to obtain the wanted superheat value: if a great refrigerant flow rate flows through the valve, the superheat will be lower; vice versa, if a lower refrigerant flow rate flows through the valve, the superheat will be higher for same conditions for air going through evaporator.

After the TXV regulation, the flow can finally go through the evaporator which absorbs the latent heat from the environment, allowing in this way the phases switching of refrigerant from its liquid state to its gaseous state. Because of this switching, the refrigerant becomes low-pressure overheated vapor. Therefore, the R134a, after completely expanding and vaporizing, is in the right conditions to go through the compressor again and restarting the cycle.

2.2 System modeling

The purpose of this thesis project is to realize, in the Matlab-Simulink environment, a simulation model that could be capable of simulating ECS system performance when different input parameters change.

There are no previous models, neither in Matlab-Simulink environment, nor in some others simulation environments. Therefore, it has to be simulated with the

basics. Only afterwards, the model will be improved. The model is also designed in order to provide the possibility to implement further changes and improvements. Following we will describe in detail how we have decided to set the modeling.

2.2.1 General setting of modeling

We decided to proceed with a progressive development of the model. In so doing, it is possible to obtain a model that will be completed in each part, but, at every stage of planning, it will be every time more complex and complete.

In particular, we decided to proceed through several model phases. Each of these represents a different modeling level:

- Phase 1: first of all, an elementary model of each components is built. These components are linked in the Matlab-Simulink environment to reproduce a steady-state model. We decided to work using this kind of model development in order to realize a complete model during the embryonic stage, too. At the same time, this way to proceed has been chosen because components data are limited, thus it appears too difficult to realize detailed models
- Phase 2: in a second step we add to the phase 1 model an elementary model of the cockpit in order to obtain a simple dynamic model. This one will be obtained with numerous assumptions and simplifications for each components. Because of these assumptions, the system performances described by the model in phase 2 will be similar to those in the final model, but not exactly the same: we will try to bypass this problem in phase 4
- Phase 3: after the dynamic model, we try to create a system control law capable to describe the correct distribution of the flow rate in order to obtain temperature and pressure requested by the pilot
- Phase 4: in this last phase we will improve the modeling of each components. The aim of this phase is to detail each component model (compared to the previous one), in order to create a model as realistic as possible and capable to simulate the real performances of the ECS system

It should be noted that modeling has been arranged in order to realize the four phases independently. A remarkable work and long times are needed to obtain a simulation model with all the previous design phases, thus, complete and detailed. Therefore, the main aim of this thesis project is to obtain a simulation model, whose components are completely modeled and in which at least phases 1, 2 and 3 are carried out. In so doing, it is possible to realize a simulation model fully functioning whose results are consistent, even if still not too accurate. Anyway, the user can develop also the fourth phase at a later date. It will be possible, since we set the modeling in this way.

2.2.2 Components modeling

As far as components modeling is concerned, we decided to proceed starting from simpler system components.

First of all, components of the Bleed Subsystem were modeled. We decided to start from the Bleed line, because its components have less critical issue than the rest of the system. After that, the Vapor Cycle was modeled, as well.

In detail, both in the Bleed Subsystem and in the Vapor Cycle Subsystem each component was enclosed in a Simulink subsystem. In so doing, the overall ECS system was made up of blocks with only the inputs and their respective outputs, in order to avoid to create too much confusion among blocks. Entering a specific block, it is also possible to observe the single component model and eventually to modify or improve it.

Furthermore, each component was modeled in an independent file. In this way, that component may be eventually simulated independently, thus, without the rest of the system.

We decided to use some model blocks, inside the entire system, instead of subsystem, in order to avoid to waste any changes in some components. *Model* blocks will be arranged in such a way as to be recalled in the global system.

Legal disclaimer: UTAS does not endorse this thesis.

Chapter 3

Bleed Subsystem modeling

As already stated in the previous chapter, the modeling of the ECS system started from the Bleed Subsystem. The latter includes the majority of system components, but, at the same time, it seems to be simpler to model than the Vapor Cycle components. Therefore, it was considered appropriate to model components with less critical issue first.

The different system components have been individually modeled and later they have been assembled in the overall system.

Thus, the evolution of the single components modeling will be described in detail.

3.1 ECS PRSOV

The PRSOV valve (Pressure Regulating and Shut-Off Valve) is the first Bleed Subsystem component modeled. This is the first component the air flow has to go through after it has been bled from the aircraft engine compressor and its main functions are to switch on/off the Bleed Subsystem and to regulate the pressure downstream the valve. Moreover, this valve is able to stop the bleed flow in case of over-pressure or over-temperature into the system.

A characteristic curve is required for this valve. For this purpose, a theoretical

curve has been used. The latter is generally made of some straight line that approximate the correct shape of a regulation curve of a valve. This theoretical curve is approximatively like the one shown in Figure 3.1:

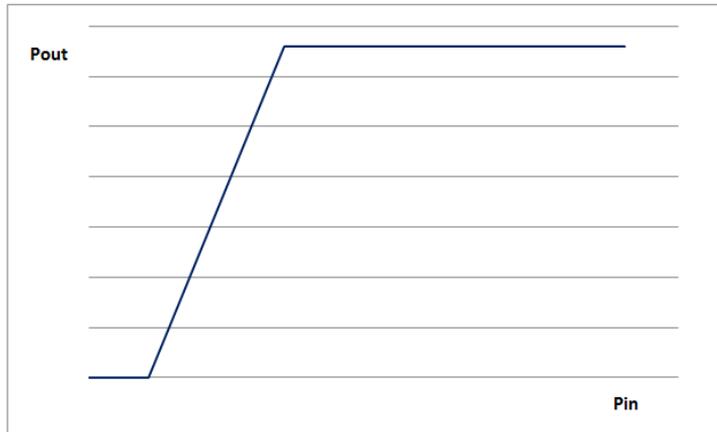


Figure 3.1. PRSOV theoretical curve

The theoretical curve can be used to simplify valve modeling, because the purpose is to realize a model capable to well approximate the real performances of the system, but, at the same time, to be as simple as possible to reduce the computational costs. When it is necessary to move on to an higher modeling level or to modify components in order to obtain a more realistic simulation model, the real curves have to be introduced.

As Figure 3.1 shows, as it can be expected with a similar valve, this kind of curve represents exactly the valve performances. At first, the valve is closed and only at a later stage it starts to open up: firstly, there is a zero outlet pressure, for a non-zero inlet pressure, too; this is due to the fact that this pressure is used to allow the first movement of the piston valve, thus it is needed an higher inlet pressure in order to obtain a non-zero pressure inside the chamber of the valve.

As far as concerns the modeling of this component, we firstly chose to create a .dat format file in which data have been set out in a table composed by two columns: the first column represents values of inlet pressure (Pin) and the second one represents values of outlet pressure (Pout). Then, we wrote a Matlab routine capable

to read that table data, to save them in an array and to implement a *while* cycle. Thus, this cycle will be responsible for finding the nearest pressure input data to the input value required by system and, then, for finding the respective outlet pressure value with an interpolation.

Therefore, a *Matlab Function* block has been created in the Simulink environment. This block creates a link with a *Matlab script* in which a *function* has been realized. Its inputs are the required inlet pressure, the ambient pressure, the ambient temperature and the solenoid state. On the other hand, the outputs are outlet pressure and temperature.

Moreover, there is an input called *State* that represents the solenoid state and it can have only two values: 1 or 0. If *State=1*, the solenoid is energized, thus the solenoid current is non-zero and positive ($I>0A$) and the valve reaches the open position: the piston is moved and the valve regulates the downstream pressure. Instead, if *State=0*, the solenoid is de-energized, so the solenoid current is zero ($I=0A$) and the valve is closed. Thus the outlet relative pressure is zero and the temperature is the ambient one.

A matrix constituted by three columns has been inserted inside the *PRSOV function*. The first column returns the values of inlet pressure; the second one states zero value for each row of that column; the third one returns the values of outlet pressure. Therefore, the outlet pressure can be achieved by a linear interpolation of matrix data.

With the last *function*, if *State=0* the pressure value is interpolated on the second column of the matrix that is made up of only zero value. In so doing, as expected, a zero outlet relative pressure is obtained for each inlet pressure, simulating the closed position. Instead, if *State=1*, the pressure value is interpolated on the third column, which is the column with the value obtained by the theoretical curve.

Moreover, as far as concerns the temperature, an important assumption has been made to obtain a simple model: an adiabatic system is considered.

Therefore, with this assumption, the outlet temperature of the valve can be calculated using the adiabatic introduced in the *Matlab Function*.

Eq. (3.1) and (3.2) show the equations used for the temperature calculation.

$$T \cdot p^{\frac{1-\gamma}{\gamma}} = cost \tag{3.1}$$

$$T_{in} \cdot p_{in}^{\frac{1-\gamma}{\gamma}} = T_{out} \cdot p_{out}^{\frac{1-\gamma}{\gamma}} \implies T_{out} = \frac{T_{in} \cdot p_{in}^{\frac{1-\gamma}{\gamma}}}{p_{out}^{\frac{1-\gamma}{\gamma}}} \quad (3.2)$$

If the selected *State* is *State=0*, pressure will have zero value. In so doing, system outlet temperature would be the same as that of the inlet one. It is not completely correct. In fact, the outlet temperature should have the ambient value: for this purpose, an *if* cycle has been introduced. Therefore, when the selected *State* is *State=0*, the outlet temperature reaches the one of the ambient temperature value with some iterations.

It should be noted that considering a zero value of outlet pressure ($P_{out} = 0$ kPag) and the outlet temperature equal to the ambient temperature ($T_{out} = T_{amb}$) is an important assumption. For this reason, it is fundamental to improve the modeling at a later stage, in order to obtain a more realistic simulation model.

It has been noted that, using the *Matlab Function* to model the PRSOV Valve, has a very high computational cost, too. Therefore, we decided to model the valve directly in the Simulink environment: a *Lookup Table* block has been used to interpolate the regulation curve. In so doing, it is not required to link a *Matlab Function* to the Simulink model and the calculation is easier.

A *Lookup Table 1-D* has been created. Referring to the example in Figure 3.1, data have been introduced in that *Lookup Table*. In so doing, Simulink is capable to interpolate the curve created by *Lookup Table* data. The block input is, in fact, the required pressure and Simulink will automatically interpolate the curve obtaining the outlet pressure.

Moreover, the adiabatic used to calculate the outlet temperature as previously described has been obtained using some elementary Simulink blocks.

The solenoid state has been introduced in the Simulink model in order to correctly simulate the valve functioning. At first, a Simulink block with the *if* logic was used to simulate the solenoid state. However, this block created some errors in the simulation, thus we decided to proceed by a logic approach. As far as concerns pressure, we decided to multiply the outlet pressure value by the solenoid state. The PRSOV valve, in fact, returns the calculated pressure if the solenoid is energized, but it returns a zero relative pressure if the solenoid is de-energized.

Therefore, the solenoid state value can be only 0 or 1. As a consequence, multiplying that value to the calculated pressure, these two options are obtained. A similar reasoning is also needed about temperature. In case of de-energized solenoid, the downstream outlet temperature has to be the same as the ambient one. Therefore, the logic is the same used for pressure calculation, thus the temperature calculated by the adiabatic is multiplied by the solenoid state value, but another term has been added to the resulting value of multiplication. This term has been obtained as a multiplication of the ambient temperature and the logic negation of the solenoid state value. In doing so, if the selected State is $State=0$, the simulation returns the ambient temperature. Vice versa, if the selected State is $State=1$, the temperature returned is the one calculated with the adiabatic.

3.2 Precooler - By-Pass group

After the PRSOV valve, the air flow rate goes through the Precooler – By-Pass group.

The outlet PRSOV flow rate shall be divided among the Precooler line and the By-Pass valve line. The flow rate passing through a line rather than the other one is determined by the opening of the throttle of the By-Pass valve. This opening is based on the Temperature Controller and Selector setting, depending on the evolution of the temperature and pressure conditions in the cockpit.

Following, there is a description of these components independently, showing their evolution during the modeling. After that, there will be a detailed description of the subassembly modeling: the latter is a fundamental issue because these two components works together and their interaction determines specific choices and modifications of modeling.

3.2.1 ECS Heat Exchanger

The ECS heat exchanger of the Bleed Subsystem is the Precooler. It receives the bleed air from the ECS PRSOV valve. The bleed air flow is cooled down by the ram air induced by the heat exchanger fan on ground or by the dynamic air intake

in flight.

Some performance maps were required to simulate Precooler performances in the Matlab-Simulink environment. We had both measured and simulated maps and it has been possible to extract from them an Excel table with data.

The curves of a Precooler performance map are like the ones in Figure 3.2.

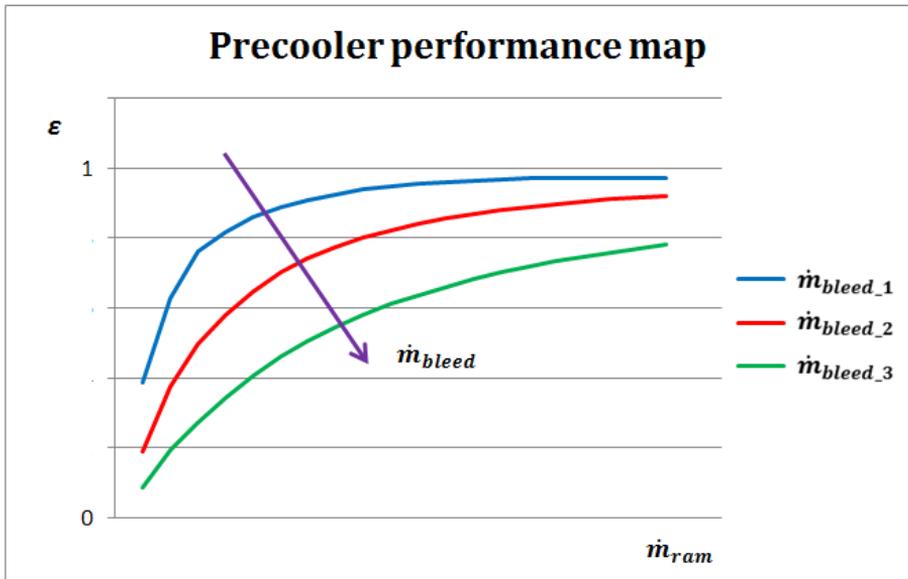


Figure 3.2. Precooler performance map

Where:

- ϵ is the efficiency
- \dot{m}_{ram} is the cold air mass flow rate of the ram line
- \dot{m}_{bleed} is the hot air mass flow rate of the bleed line, with $\dot{m}_{bleed_1} < \dot{m}_{bleed_2} < \dot{m}_{bleed_3}$

Figure 3.2 is built by plotting the mass flow rate \dot{m}_{ram} along the abscissa and the efficiency ϵ along the ordinate. Moreover, each curve corresponds to a specific mass flow rate \dot{m}_{bleed} value, as shown in the plot key.

Therefore, a *Lookup Table 2-D* has been created in the Matlab-Simulink environment. Unlike with the one constructs for the PRSOV valve, this *Lookup Table* has been realized directly importing data from the Excel file. In so doing, a performance map similar to the one in Figure 3.2 has been created.

As far as concerns the Excel file, the data have been distributed in vectors and the *Lookup Table 2-D* properly recalls the name of file and the created vectors.

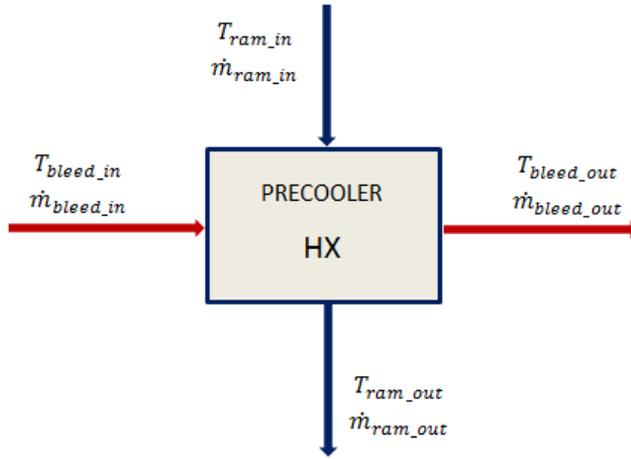


Figure 3.3. Precooler schematic

As Figure 3.3 shows, the Precooler has two different inputs, therefore the *Lookup Table* has to be a *2-D* one. The cold air mass flow rate \dot{m}_{ram} has been linked to the input u_1 ; instead, the hot air mass flow rate \dot{m}_{bleed} has been linked to the input u_2 . Thus, the output is the efficiency ϵ , used to obtain the Precooler outlet temperature of the bleed line. This temperature has been calculated through the following equation:

$$\epsilon = \frac{T_{in,bleed} - T_{out,bleed}}{T_{in,bleed} - T_{in,ram}} \quad (3.3)$$

Thus, the outlet temperature has been taken from the following mathematical equation.

$$T_{out,bleed} = T_{in,bleed} - [\epsilon \cdot (T_{in,bleed} - T_{in,ram})] \quad (3.4)$$

Along with the Precooler outlet temperature, the other parameter that has to be assessed is the Precooler outlet pressure.

We decided to realize some curves that can well approximate the Precooler pressure drop against the volume flow rate, in order to obtain outlet pressure from the pressure drop.

We have three Precooler functioning points, corresponding to three different conditions. Consequently, it has not been possible to obtain a curve with three points. Thus, it was assumed to design three different curves, each one corresponding to a specific functioning point.

It was assumed that each one of these curves was a parabola passing through that specific functioning point and through the origin of the axes. This curve has to have a horizontal tangent in the origin, too. Again, these are important assumptions, but, in the absence of other options, it seems the greater and the more realistic solution, at least in this modeling phase.

Therefore, a mathematical system has been constructed for each functioning point. The system is like the following generic one:

$$\begin{cases} y(q, \Delta p) = ax^2 + bx + c \\ y(0,0) = ax^2 + bx + c \\ \frac{d}{dx} (y(0,0)) = 0 \end{cases} \quad (3.5)$$

This mathematical system describes all the curve characteristics: the passage through the functioning point (the first equation); the passage through the origin of the axes (the second equation); horizontal tangent (the third equation).

Coefficients a , b and c has been calculated from the mathematical system for each functioning point. Thus, the three parabola equations have been plotted using a Matlab script in order to compare them.

It has been noted from the plot in Figure 3.4 that two of the functioning point curves are approximately overlapped and the third one deviates slightly from the others. Therefore, the curve passing through the three functioning points could

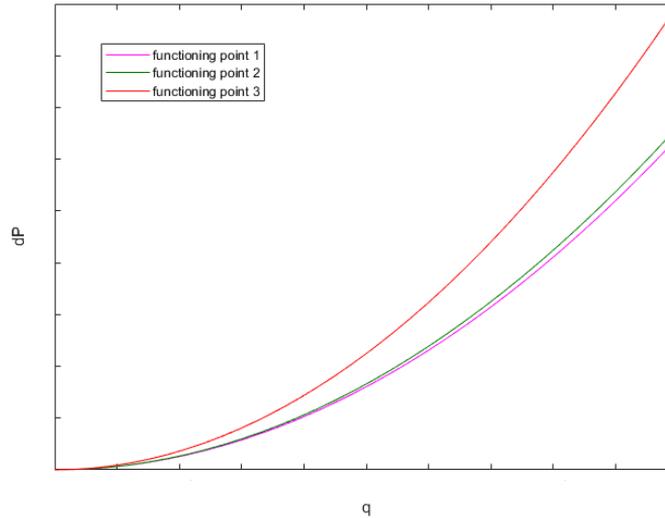


Figure 3.4. Precooler pressure drop curves

approximate well enough the real Precooler performance.

Comparing the plot of this equation and the three others, you can see that, for higher value of volume flow rate, this curve seems to be roughly overlapped to the other two similar (Figure 3.5), but, for lower value of flow rate, the approximation curve is rather different. In fact, it does not pass through the origin of axes.

Figure 3.6 shows that the approximation curve (the one passing through the three functioning points) provides an error for low volume flow rate value, because the characteristic of the passage through the origin of the axes is not respected. Fortunately, this is not a real problem because the curve deviation can be noted just for very low value of volume flow rate and generally such minimal flow rate is out of heat exchanger range. Moreover, this error is in the range of about 0,1 kPa, thus it is a negligible error.

Nevertheless, we decided to still use that approximation curve to describe Precooler performances.

Outlet Precooler pressure and outlet Precooler temperature that have been calculated are just referred to the bleed line. A similar reasoning is applied to the ram

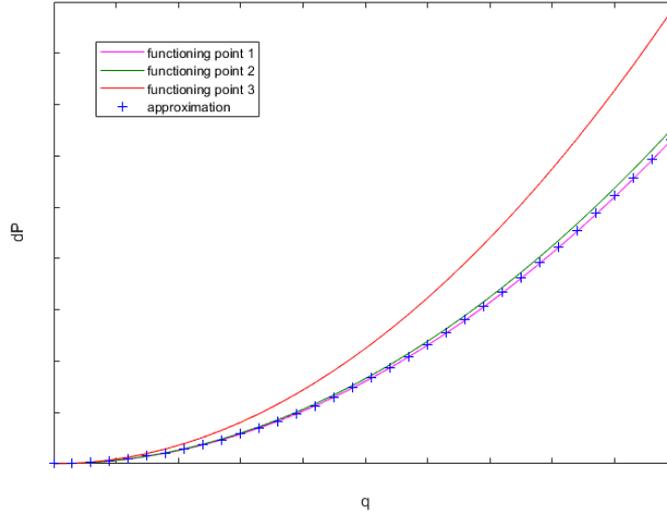


Figure 3.5. Pre-cooler pressure drop curves and the approximation curve

line by considering inlet pressure and inlet temperature introduced as inputs to the system.

Therefore, outlet temperature of the ram line has been calculated using the efficiency ϵ obtained as an output from the *Lookup Table*. In this case, the equation is:

$$\epsilon = \frac{T_{out,ram} - T_{in,ram}}{T_{in,bleed} - T_{in,ram}} \quad (3.6)$$

Output temperature has been calculated from the last equation, as follows:

$$T_{out,ram} = T_{in,ram} + [\epsilon \cdot (T_{in,bleed} - T_{in,ram})] \quad (3.7)$$

On the other hand, the coefficients of the parabola passing through the three different functioning points have been calculated in order to obtain the outlet pressure. Contrary to the previous case, the three parabolas seem to be different, thus we decided to consider the intermediate curve equation.

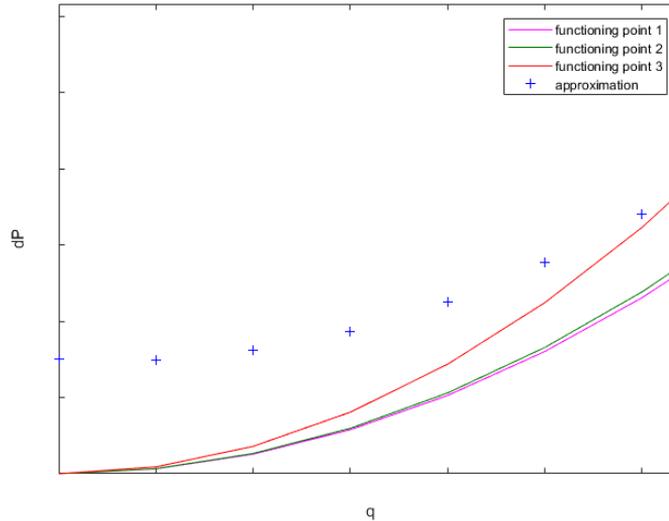


Figure 3.6. Zoom of re cooler pressure drop curves for low flow rates

It has to be noted that, if the user wants to replace the Precooler, it will be enough to recalculate the parabola coefficients and to change them inside the model.

3.2.2 Temperature Control By-Pass Valve

The Temperature Control By-Pass Valve is a throttle valve placed on the bleed line that by-passes the Heat Exchanger. Its main function is to by-pass the Precooler to less refrigerate in order to achieve the pilot request.

The outlet PRSOV flow rate should be divided among the Precooler line and the By-Pass valve line.

We had a functioning point referred to the fully open condition of the valve ($\alpha = 90^\circ$), in order to model this valve. However, it was necessary to find a mathematical equation capable to describe the valve performances in every throttle opening conditions, because a linear law that could describe it doesn't exist. Initially, we thought to use the functioning point to obtain a curve that can well approximate the By-Pass valve pressure drop against the mass flow rate. Although,

just a single functioning point is insufficient to describe the valve performances, thus we thought to assume some other characteristics of this curve:

- the curve could be a parabola
- zero pressure drop corresponds to a zero mass flow rate
- it has to have a horizontal tangent in the origin of axes

Afterwards, we tried to find a mathematical equation to describe valve performances. Therefore, a general functioning equation of a throttle valve has been considered.

$$\Delta p = A \cdot \rho \cdot T \cdot \frac{q^2}{p} \cdot \frac{MaxArea(\alpha = 90^\circ)}{Area(\alpha)} \quad (3.8)$$

Where:

- A is a coefficient specifically for each valve, thus a different value has to be used depending on the considered valve. In this case, it will be evaluated by the valve plot
- $MaxArea(\alpha = 90^\circ)$ is the maximum possible section that corresponds to the completely throttle open condition, thus to an opening angle of $\alpha = 90^\circ$
- $Area(\alpha)$ is the throttle section depending on the opening angle
- T and p are inlet temperature and inlet pressure of the By-Pass valve, which means PRSOV outlet temperature and pressure

Instead, the density ρ is determined with another important assumption: the equation of state of ideal gas has been used to calculate the density:

$$p = \rho RT \quad (3.9)$$

The latter can be rewritten as follows:

$$\frac{\rho T}{p} = \frac{1}{R} \quad (3.10)$$

Starting from the equation written in this way, it can be possible to note that the term $\frac{\rho T}{p}$ features in Eq. (3.8). Therefore, it can be directly replaced by $\frac{1}{R}$. The latter term is more convenient since R is the ideal gas constant, thus it is always a constant value.

Moreover, the sections can be defined as follows:

$$MaxArea(\alpha = 90^\circ) = \pi r^2 \quad (3.11)$$

$$Area(\alpha) = \pi r^2 - (\pi r (r \cdot \cos \alpha)) \quad (3.12)$$

where r is the valve section radius. Moreover, $Area(\alpha)$ has been obtained viewing the free section as a subtraction of the section occupied by the throttle from the total one. The section occupied by the throttle can be approximated to an ellipse, whose area is $\pi \cdot b_1 \cdot b_2$ (b_1 and b_2 are the two semi-axes of the ellipse). Thus, $b_1 = r$ and $b_2 = r \cdot \cos \alpha$ are used in order to calculate the valve section.

Therefore, (3.7) can be rewritten including the previous considerations, as follows:

$$\Delta p = \frac{A}{R} \cdot q^2 \cdot \frac{\pi r^2}{\pi r^2 (1 - \cos \alpha)} = \frac{A}{R} \cdot q^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.13)$$

As a result, the equation implemented in the Simulink environment will be the following one:

$$\Delta p = \frac{A}{R} \cdot q^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.14)$$

Using elementary blocks, the previous equation has been modeled and pressure drop Δp has been obtained. Moreover, the definition of pressure drop has been considered, too, in order to calculate the downstream valve pressure. Thus, using the equation $\Delta p = p_{in} - p_{out}$, the outlet pressure p_{out} has been obtained.

Regarding Eq. (3.13), the gas constant R value considered is:

$$R = 0,28705 \frac{kPag \cdot m^3}{kg \cdot K} \quad (3.15)$$

The latter value has been chosen in order to respect the units of measurement of the system, in fact:

$$\frac{\rho T}{p} = \frac{kg}{m^3} \cdot \frac{1}{kPag} = \frac{kg \cdot K}{kPag \cdot m^3} = \frac{1}{R} \quad (3.16)$$

Generally, the gas constant is the following one:

$$R = 287,05 \frac{J}{kg \cdot K} = 287,05 \frac{Pa \cdot m^3}{kg \cdot K} \quad (3.17)$$

In this case the pressures are expressed in *kPag*, thus the previous value has to be divided by 1000 in order to compare the units of measurement.

Therefore, in view of all these considerations, the chosen *R* value is the one in Eq. (3.14).

Instead, constant *A* has been calculated from the functioning point, as anticipated. It depends on the specific valve used, too. If the valve changes, the constant *A* will be recalculated considering the new valve characteristics.

After the connection of Precooler and By-Pass valve, the model of the Temperature Control By-Pass Valve had to be modified. This modeling change has been necessary, because the connection of the two components showed the importance of their simultaneously modeling: their flow rates are related. In fact, the flow rate downstream the PRSOV valve is divided in Precooler flow rate and By-Pass valve flow rate depending on the throttle opening.

We need to introduce a calculation to obtain the two different flow rates, thus, a complementary and simultaneous modeling of the components as a subassembly is required.

The subassembly modeling is described in Paragraph 3.2.3. Therefore, you can see the next changes in the Temperature Control By-Pass Valve modeling directly in that paragraph.

3.2.3 Subassembly

Precooler and Temperature Control By-Pass Valve works together. They have the same inlet pressure and inlet temperature and return a single output for pressure and temperature.

Therefore, a single Simulink block that includes both these components has been realized inside the model. From now on, all reasoning will be done on the sub-assembly.

As already said, the hot air flow rate of the Bleed Subsystem, after PRSOV valve is divided in order to obtain the correct distribution of flow rates through the ducts of the two components. After going through the components, the flow rates are mixed and, as a result, they flow back together as a single flow rate that will enter the FMRSOV valve.

Thus, a calculation to correctly divide the flow rates has been introduced inside the unified block Precooler – By-Pass. We decided to think about pressure drop to obtain that calculation: pressure drop has to be the same in the two components. Following there are the equations of the two components pressure drop, as described in 3.2.1 and 3.2.2.

$$\Delta p_{precooler} = a q_{precooler}^2 + b q_{precooler} + c \quad (3.18)$$

$$\Delta p_{by-pass} = \frac{A}{R} \cdot q_{by-pass}^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.19)$$

Matching these two mathematical equation, the following one was calculated:

$$\Delta p_{precooler} = \Delta p_{by-pass} \quad (3.20)$$

Therefore:

$$a q_{precooler}^2 + b q_{precooler} + c = \frac{A}{R} \cdot q_{by-pass}^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.21)$$

After converting it in volume flow rate, the flow rate has been replaced in the

previous equation:

$$a \left(\frac{\dot{m}_{precooler}}{\rho} \right)^2 + b \left(\frac{\dot{m}_{precooler}}{\rho} \right) + c = \frac{A}{R} \cdot \left(\frac{\dot{m}_{by-pass}}{\rho} \right)^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.22)$$

Recalling that:

$$\dot{m} = \dot{m}_{precooler} + \dot{m}_{by-pass} \quad (3.23)$$

The By-Pass mass flow rate has been expressed as a function of the total mass flow rate and the Precooler one:

$$a \left(\frac{\dot{m}_{precooler}}{\rho} \right)^2 + b \left(\frac{\dot{m}_{precooler}}{\rho} \right) + c = \frac{A}{R} \cdot \left(\frac{\dot{m} - \dot{m}_{precooler}}{\rho} \right)^2 \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.24)$$

The Precooler mass flow rate has been calculated from the last equation, as follows:

$$\begin{aligned} \dot{m}_{precooler} = & \frac{- \left(\frac{b}{\rho} + \frac{2\dot{m}A}{R(1-\cos \alpha)\rho^2} \right)}{2 \left(\frac{a}{\rho^2} - \frac{A}{R(1-\cos \alpha)\rho^2} \right)} + \\ & + \frac{\sqrt{\left(\frac{b}{\rho} + \frac{2\dot{m}A}{R(1-\cos \alpha)\rho^2} \right)^2 - 4 \left(\frac{a}{\rho^2} - \frac{A}{R(1-\cos \alpha)\rho^2} \right) \left(c - \frac{A\dot{m}^2}{R(1-\cos \alpha)\rho^2} \right)}}{2 \left(\frac{a}{\rho^2} - \frac{A}{R(1-\cos \alpha)\rho^2} \right)} \end{aligned} \quad (3.25)$$

The Precooler mass flow rate has been realized using elementary blocks in the Matlab-Simulink environment and the other one (By-Pass mass flow rate) has been obtained from this, as follows:

$$\dot{m}_{by-pass} = \dot{m} - \dot{m}_{precooler} \quad (3.26)$$

The two mass flow rates can be introduce into the respective component as an

input.

After the block Precooler – By-Pass, the pressure output has to be just one and it has to be calculated as a combination of the two outputs. About temperature, the output has to be again just one.

Precooler outlet pressure and By-Pass valve outlet pressure have to be the same, thus the output port was linked to only one of the two.

Instead, about temperature, a weighted average of Precooler temperature and By-Pass valve one has been calculated as follows:

$$T_{out} = \frac{\dot{m}_{precooler}T_{out,precooler} + \dot{m}_{by-pass}T_{out,by-pass}}{\dot{m}_{precooler} + \dot{m}_{by-pass}} \quad (3.27)$$

It has been noted that, these last are bleed outlet pressure and temperature. Ram outlet dimensions do not depend on the Temperature Control By-Pass Valve, thus their calculation is the one described in Paragraph 3.2.1.

Although, because of running simulation difficulties, we decided to change our approach to modeling: instead of a repartition of flow rates, we preferred to use an algebraic loop. The latter seems to be a better solution in the Simulink environment, because it is able to increase automatically the dimensions until reaching the required condition.

Before changing the model, we preferred to align component models using *Lookup Table* for each of these. The Precooler was still modeled with *Lookup Table*; instead, the Temperature Control By-Pass Valve model has been modified.

Therefore, as it had been done for the Precooler, a *Lookup Table 2-D* has been created in the Matlab-Simulink environment to model the Temperature Control By-Pass Valve. This *Lookup Table* has been realized directly importing data from the Excel file in which there are data to interpolate. In so doing, a map of the pressure drop Δp against the volume flow rate, changing the opening angle of the throttle α , has been created.

In the Excel file, data have been distributed in vectors and the *Lookup Table 2-D* properly recalls the name of file and the created vectors.

The By-Pass valve has two different inputs: the volume flow rate going through the valve has been linked to the input u_1 ; instead, the opening angle of the throttle α has been linked to the input u_2 .

After adding that *Lookup Table*, model changes have been implemented. Component models do not change, but an algebraic loop has been introduced in the place of the flow rate calculation block described above. The purpose of this algebraic loop is to calculate the Precooler mass flow rate on the basis of bleed flow rate and the same Precooler mass flow rate.

First of all, it is necessary to match the mathematical equations of Δp of the two components (Eq. (3.20) and Eq. (3.21)).

Therefore, the following equation is achieved:

$$\dot{m}_{precooler} = \frac{-\frac{b}{\rho} + \sqrt{\left(\frac{b}{\rho}\right)^2 - 4\left(\frac{a}{\rho^2}\right)\left(c - \frac{A(\dot{m} - \dot{m}_{precooler})^2}{R(1 - \cos \alpha)\rho^2}\right)}}{2\left(\frac{a}{\rho^2}\right)} \quad (3.28)$$

Although, even this modeling solution didn't work properly. In fact, during the running with some initial conditions, sometimes the radicand became negative and its modulus is required: it is not physically correct, thus an alternative modeling solution had to be found.

We tried some different solutions, but it has been demonstrated that everyone of these had still low applicability. Thus, we decided to try a completely new modeling logic.

Therefore, we decided to use the By-Pass mass flow rate $\dot{m}_{by-pass}$ as input to the *Lookup Table* of the Temperature Control By-Pass Valve. The $\Delta p_{by-pass}$ value has been calculated by the latter *Lookup Table*, and, considering that the two pressure drops have to match ($\Delta p_{precooler} = \Delta p_{by-pass}$), the calculated $\Delta p_{by-pass}$ can be used as input to a *Lookup Table* of the Precooler. The latter has been realized by inverting Excel vectors, compared to the By-Pass *Lookup Table*, in order to obtain $\Delta p_{by-pass}$ as input and $\dot{m}_{precooler}$ as output. In so doing, in fact, it was possible to calculate the Precooler mass flow rate $\dot{m}_{precooler}$. Finally, the $\dot{m}_{precooler}$ has been subtracted to the bleed mass flow rate, that is the one flowing from the PRSOV valve, in order to obtain $\dot{m}_{by-pass}$ to restart the algebraic loop with.

The schematic in Figure 3.7 shows the general architecture of the algebraic loop. It has been noted that the Precooler *Lookup Table* mentioned about algebraic loop modeling is not the same one described above. The latter provided Precooler pressures, instead in this case the system needs a relation between pressure drop and

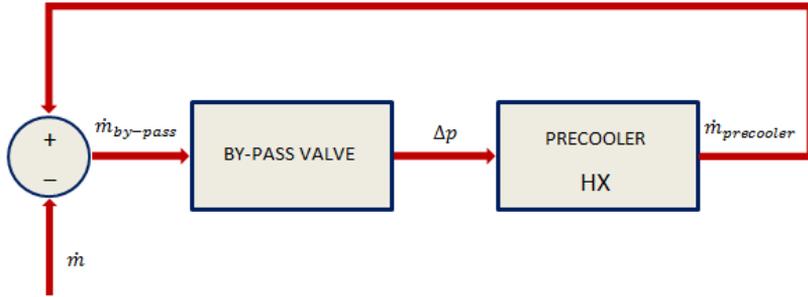


Figure 3.7. Precooler - By-Pass group schematic

mass flow rate. In particular, it is a *Lookup Table 1-D* specifically created for the algebraic loop and it complements the other one.

Thus, we decided to use the algebraic loop because, running the simulation, reasonable flow rate and outlet temperature values have been obtained without mathematical or physical mistakes.

As far as concerns temperature and pressure outputs of the combined Precooler – By-Pass group, they have been evaluated as previously described. Thus, outlet pressure is the Precooler outlet one (it is the same of the By-Pass valve); outlet temperature is calculated as follows:

$$T_{out} = \frac{\dot{m}_{precooler} T_{out,precooler} + \dot{m}_{by-pass} T_{out,by-pass}}{\dot{m}_{precooler} + \dot{m}_{by-pass}} \quad (3.29)$$

3.3 Flow Modulating RSOV

The FMRSOV valve (Flow Modulating Regulating and Shut-Off Valve) is the penultimate component that the air flow has to go through before entering the cockpit. It is quite similar to the PRSOV valve but its main function is to modulate the hot air flow that will enter the cockpit after passing through the cabin outlet valve. It has been noted that about 40% of mixed air flow resulting in output of Precooler – By-Pass subassembly is took off and deployed for survival systems.

Thus, just about 60% of that air flow is able to enter the FMRSOV valve.

This valve has been modeled together with the PRSOV valve, thus the evolution of the modeling is the same.

The characteristic curves are similar to the PRSOV valve one, but, in this case, there are four curves (instead of only one) and each one corresponds to a different value of the current passing through the solenoid. In fact, the pilot can choose a specific position in order to obtain a different temperature and pressure inside the cockpit:

- Position 0: the solenoid is de-energized and, as a consequence, the valve in closed position
- Position 1: the solenoid is energized and it is crossed by a little current value. Thus, the regulation will be on the respective curve
- Position 2: the solenoid is energized and it is crossed by another current value (higher than the previous one). Thus, the regulation will be on the respective curve
- Position 3: the solenoid is energized and it is crossed by an high current value. Thus, the regulation will be on the respective curve

Since the FMRSOV model has been realized along with the PRSOV one, also in this case, a *Matlab Function* block has been created in the Matlab-Simulink environment, and it has been subsequently replaced by a *Lookup Table*.

First of all, the *Matlab Function* block has been realized. Its *Matlab script* is the same described in Paragraph 3.1 about the PRSOV one. The input State, named *Curr* for this valve, can have a position value from 0 to 3 and, in the model, it shows the possibility of the interpolation on a specific column of the matrix created by the *Matlab script*.

About FMRSOV valve, in fact, the matrix consists of five columns: the first one is the column of inlet pressures and the other four correspond to the four position selectable by pilot.

The same modeling of the PRSOV one is provided about outlet temperature calculation.

However, at a later stage, exactly as for PRSOV modeling, a *Matlab Function* to

model the valve has a very high computational cost. Therefore, we decided to model it directly in the Simulink environment: a *Lookup Table* block is used in order to interpolate the regulation curves. In so doing, it is not required to link a *Matlab Function* to the Simulink model and the calculation is easier and quicker. A *Lookup Table 2-D* has been chosen for FMRSOV valve modeling and it has been realized directly importing data from an Excel file.

Since there were no functioning point data, an acceptance test has been carried out and its result data have been used to obtain the Excel file for the *Lookup Table*. During the acceptance test, air flow rate and outlet pressure have been measured with increased inlet pressure and it has been carried out for each FMRSOV current value. Then, the measurements have been plotted and data have been saved in the Excel file.

Moreover, the *Lookup Table 2-D* has two inputs: the inlet pressure p_{in} has been linked to the input u_1 ; instead, the solenoid state has been linked to the input u_2 , in order to choose the correct curve to interpolate depending on the pilot selecting. As for PRSOV, a logic approach has been considered to calculate the FMRSOV valve outlet pressure and temperature. However, the FMRSOV valve returns the calculated pressure if the solenoid is energized ($State=1$, $State=2$ or $State=3$), but it returns a zero relative pressure if the solenoid is de-energized ($State=0$). Therefore, unlike the PRSOV valve, only a multiplication is insufficient and we needed to introduce the logic block *Compare to 0* downstream the solenoid state input. In so doing, if the solenoid is de-energized, the operational logic block returns 1 , instead, if the solenoid is energized, the operational logic block returns 0 . This value is multiplied by the logic operator *NOT* and it is multiplied by the calculated pressure: in so doing, the four options are obtained.

A similar reasoning is also needed about temperature. The latter is evaluated in the same way as the PRSOV valve, as described in Paragraph 3.1.

Moreover, after a first FMRSOV simulation, it was noted that, downstream the valve, an important decrease of the temperature has been shown: for this valve the adiabatic transformation is unable to well approximate the valve performance. Because of that, a constant temperature transformation was preferred.

3.4 Cabin Outlet Valve

The Cabin Outlet Valve (COV) is the last component of the Bleed Subsystem that the air flow has to go through before entering the cockpit. In fact, its main purpose is to consent the correct distribution of air into the cabin.

This valve is designed so that the pilot can select one of the following valve position:

- Normal Condition
- Defog Condition

If the selected position is the first one (Normal Condition), the major portion of the inlet air flow (about 70%) goes into the cockpit environment close to the pilot, going through the *footwarmer* line; the remaining portion of the inlet air flow (about 30%) goes through the *defog* line, thus it goes to the canopy defog vents. On the other hand, if the selected position is *Defog Condition*, all the inlet air flow goes through the *defog* line.

About COV modeling, first of all we decided to calculate pressure drop using the simplified parabola equation $\Delta p = k \cdot q^2$, where k is a generic coefficient that is a specific one for each Δp .

Using the functioning points about different operative COV conditions, k coefficients of the parabola equation in the different functioning conditions have been evaluated using an Excel code. Therefore, different coefficients for each operational modes have been obtained:

- A_1 \longrightarrow *footwarmer* line coefficient in *Normal Condition*
- A_2 \longrightarrow *defog* line coefficient in *Normal Condition*
- B_1 \longrightarrow *defog* line coefficient in *Defog Condition*

Pressure drop downstream the two outlet line (*footwarmer* and *defog* lines) can be evaluated using these coefficients. However, it was showed that the calculated pressure drop value is not the same for the two lines and it is due to the fact that there is a piping connecting each COV output to the cockpit. In fact, before entering the cockpit, the air goes through this pipe in which there is an additional pressure drop.

The pressure downstream the first piece of pipe has been calculated as $p_{out} = p_{in} - \Delta p$, in order to evaluate the pressure drop inside the pipe. The calculated p_{out} is the same pressure entering the second additional pipe: it is possible to obtain the air density at this point of the pipe in order to calculate the volume air flow rate going through the pipe, using the previously calculated pressure p_{out} . In so doing, the pipe pressure drop has been calculated and the coefficient of the parabola equation in the known functioning point condition has been evaluated from this.

- $C \rightarrow$ additional pipe coefficient

Considering this additional pressure drop, the pressure outgoing from the two lines are essentially the same.

Once the coefficients were found, we needed to decide how to proceed to valve modeling, in particular how to calculate the flow rates entering the two valve outlet lines.

First of all, we thought to determine them by matching the pressure drop equations and explaining the flow rates, as we did for the Precooler – By-Pass group. The pressure drop in Normal Condition are:

$$\Delta p_{footwarmer} = A_1 q_{footwarmer}^2 + C q_{pipe1}^2 \quad (3.30)$$

$$\Delta p_{defog} = A_2 q_{defog}^2 + C q_{pipe2}^2 \quad (3.31)$$

The following equation has been obtained, matching the latter pressure drop:

$$\Delta p_{footwarmer} = \Delta p_{defog} \quad (3.32)$$

Therefore:

$$A_1 q_{footwarmer}^2 + C q_{pipe1}^2 = A_2 q_{defog}^2 + C q_{pipe2}^2 \quad (3.33)$$

The mass flow rate has been explained in the latter equation. As a result, the

following one has been obtained:

$$A_1 \left(\frac{\dot{m}_{footwarmer}}{\rho} \right)^2 + C \left(\frac{\dot{m}_{footwarmer}}{\rho_{pipe}} \right)^2 = A_2 \left(\frac{\dot{m}_{defog}}{\rho} \right)^2 + C \left(\frac{\dot{m}_{defog}}{\rho_{pipe}} \right)^2 \quad (3.34)$$

We can already observe by the latter equation that it is quite complex and difficult to match the pressure drop equation, because of the two different densities: one of the air in the first pipe and one of the air going through the additional pipe. In particular, the latter density should be obtained using pressure and temperature depending on the Δp : the Eq. (3.34) could be developed in a huge equation.

Another solution has to be thought.

First of all, we tried to realize a *while* cycle in the Matlab-Simulink environment. Its purpose is to cyclically calculate flow rates $\dot{m}_{footwarmer}$ e \dot{m}_{defog} until we succeeded in matching the outlet pressure.

Therefore, a model capable to calculate the *footwarmer* line pressure drop and the *defog* line pressure drop has been created in the Matlab-Simulink environment and the two pressure drop have been compared to a very little value ϵ . After that, a *while iterator* block has been included in the model and it has been set as *do-while* to carry out the while cycle and to improve the flow rate with a *memory* block.

Unfortunately, also this modeling solution cannot be used because the *while iterator* block keeps on improving the flow rate also after achieving the requested condition: in other words, the flow rate is increased until the end of the simulation. This is because Simulink is a dynamic simulation tool and the use of an algebraic loop to improve the flow rate is a better solution.

Therefore, a completely different approach to the COV modeling has been adopted. First of all, the COV *Normal Condition* has been modeled; after that, the *Defog Condition*, too.

We decided to use the same logic adopted for the flow rate repartition of the Pre-cooler – By-Pass subassembly, to distribute the flow rates between the *footwarmer* and the *defog* lines. However, this model is more complex because of the additional pipe pressure drop in both the lines.

Two *Lookup Table 2-D*, one for each line, have been used and they have been realized directly importing data from an Excel file. On the other hand, we preferred

to model the additional pipes with elementary blocks in the Matlab-Simulink environment.

Therefore, the logic used can be summarized in the following schematic:

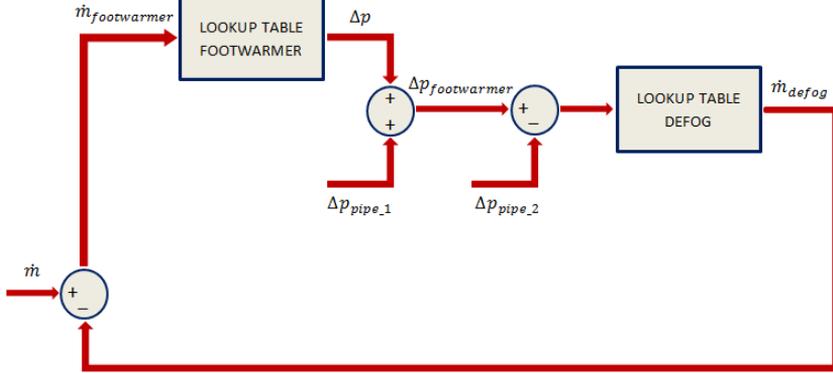


Figure 3.8. COV flow rates schematic

In detail, the *footwarmer* mass flow rate represents the input to the *Lookup Table* constructed for the *footwarmer* line. Therefore, the pressure drop downstream the first pipe of this line is the output of the *Lookup Table*, that, added to the additional pipe pressure drop, provides the overall *footwarmer* line pressure drop.

The additional pipe pressure drop has been evaluated using the simplified parabola equation $\Delta p = k \cdot q^2$, in which the coefficient in the known functioning point condition is the additional pipe coefficient C previously seen and the flow rate has been calculated considering the pressure and temperature downstream the first pipe.

Thus, the overall *footwarmer* line pressure drop has to match the overall *defog* line one, thus the used approach is exactly the opposite compared to the previous one: a subtraction between the overall pressure drop and the additional pipe pressure drop evaluated for the *defog* line has been carried out, thus the pressure drop downstream the first pipe of the *defog* line has been obtained. The latter enters the *Lookup Table* constructed for the *defog* line as an input and it returns the *defog* mass flow rate. The latter, removed to the bleed mass flow rate, provides the new *footwarmer* mass flow rate which allows to restart the cycle.

The latter additional pipe pressure drop has been obtained using the same equation ($\Delta p = k \cdot q^2$) in which the coefficient is always the same but the flow rate has been calculated again considering this pipe inlet condition.

The main difficulty in this component modeling consists on the second part of the algebraic loop. In fact, we needed a lot of physical dimensions which are not calculated yet, since we decided to retrace the calculations in reverse. This entails that internal algebraic loops are needed.

The previous description is about the *Normal Condition*. On the other hand, the *Defog Condition* modeling is turned out to be easier, because in this situation all the flow rate goes through the *defog* line and its distribution among the two lines is not required.

A repartition in two different subsystem models for the two conditions has been preferred, in order to obtain a more clear model. However it is just a single component, thus these two subsystems have been included into a single Simulink block and their outputs have been combined in order to obtain a single output for each dimension (pressure and temperature).

A *Lookup Table 1-D* has been created inside the *Defog Condition* subsystem. Its input is the bleed flow rate and, with the flow rate, the *Lookup Table* interpolates data and the pressure drop Δp in this condition has been obtained. The additional pipe pressure drop has been added to the *Defog Condition* Δp , so outlet pressure and temperature have been calculated exactly the same as *Normal Condition*.

The COV outlet physical dimensions have to be four, because there are two different pipes: pressure and temperature in output from the *footwarmer* line, entering the cockpit, and pressure and temperature in output from the *defog* line, going out the aircraft to the canopy defog vents.

Therefore, the outputs have been properly combined and the definitive outlet dimensions have been obtained considering the contribution of all different COV conditions.

It has been noted that, if the *Defog Condition* is selected, the *footwarmer* line outputs will be characterized by a zero pressure and ambient temperature.

The *Model* block could not be used for the COV modeling because of the presence of an algebraic loop. In fact, the latter is not supported by the *Model* block: COV is the only component modeled directed in the ECS model.

3.5 Bleed Subassembly

The modeling of each component of the Bleed Subsystem has been completed, therefore it has been possible to link them and to think how to calculate the Bleed flow rate in order to obtain the condition $p_{outCOV} = p_{cockpit}$.

First of all we decided to calculate the flow rate by matching the subtraction between the upstream and the downstream pressure of the Bleed Subsystem and the sum of all the components pressure drop. Thus, the following equation has been obtained:

$$p_{in} - p_{out} = \sum \Delta p_{system} = p_{in} - p_{cockpit} \quad (3.35)$$

Explaining the previous summation, it has been achieved to following one:

$$p_{in} - p_{cockpit} = \Delta p_{PRSOV} + \Delta p_{precooler-bypass} + \Delta p_{FMRSOV} + \Delta p_{COV} \quad (3.36)$$

The $\Delta p_{precooler-bypass}$ and the Δp_{COV} are the only pressure drop depending on the flow rate that can be expressed in the following way:

$$\Delta p_{pb} = \left[a \left(\frac{\dot{m} - \dot{m}_{bypass}}{\rho_{pb}} \right)^2 + b \left(\frac{\dot{m} - \dot{m}_{bypass}}{\rho_{pb}} \right) + c \right] \quad (3.37)$$

$$\Delta p_{COV} = B_{COV} \left(\frac{\dot{m}}{\rho_{COV}} \right)^2 \quad (3.38)$$

where:

- $\dot{m} - \dot{m}_{by-pass}$ corresponds to the precooler mass flow rate $\dot{m}_{precooler}$
- \dot{m} is the Bleed subsystem mass flow rate

- the subscript pb shows a precooler - by-pass line dimension

It has been noted that, since Precooler branch and By-Pass valve branch are in parallel and, as described earlier, they have the same value, just one of those can be used to calculate component Δp_{pb} (Eq. (3.37)).

The $\dot{m}_{by-pass}$ can be explained using Eq. (3.19) as follows:

$$\Delta p_{by-pass} = \frac{A}{R} \cdot \frac{\dot{m}_{by-pass}^2}{\rho_{pb}^2} \cdot \frac{1}{(1 - \cos \alpha)} \quad (3.39)$$

$$\implies \dot{m}_{by-pass} = \rho_{pb} \sqrt{\Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha)} \quad (3.40)$$

After introducing Eq. (3.40) in Eq. (3.37), performing the calculations and replacing it in the completed Eq. (3.36), the following equation has been obtained:

$$\begin{aligned} & \left(\frac{B_{COV}}{\rho_{COV}^2} + \frac{a}{\rho_{pb}^2} \right) \dot{m}^2 + \left(\frac{b}{\rho_{pb}} - \frac{2a}{\rho_{pb}} + \sqrt{\Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha)} \right) \dot{m} + \\ & + \Delta p_{PRSOV} + \Delta p_{FMRSOV} + a \Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha) + \\ & - b \sqrt{\Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha)} + c - \Delta p_{cockpit} = 0 \end{aligned} \quad (3.41)$$

The Eq. (3.41) is a quadratic equation and \dot{m} is its unknown. Thus, the equation can be rewritten calling the different terms as follows:

- $A = \left(\frac{B_{COV}}{\rho_{COV}^2} + \frac{a}{\rho_{pb}^2} \right)$
- $B = \left(\frac{b}{\rho_{pb}} - \frac{2a}{\rho_{pb}} + \sqrt{\Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha)} \right)$
- $C = +\Delta p_{PRSOV} + \Delta p_{FMRSOV} + a \Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha) +$
 $- b \sqrt{\Delta p_{by-pass} \frac{R}{A} (1 - \cos \alpha)} + c - \Delta p_{cockpit}$

In so doing, the \dot{m} has been evaluated:

$$\dot{m} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (3.42)$$

Then, we decided to find an alternative solution to model the distribution of the Bleed mass flow rate, because Eq. (3.42) is excessively burdensome to model. Therefore, we thought to enclose all the Bleed Subsystem components in a single Simulink block.

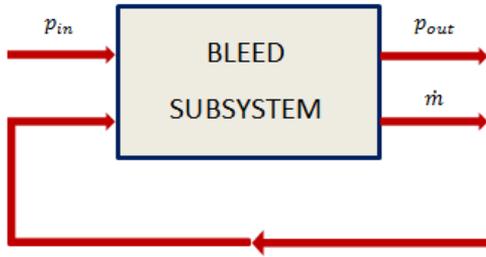


Figure 3.9. Bleed Subsystem general schematic

Imagining the entire Bleed Subsystem like a single block, it has been possible to obtain the overall Δp of the Bleed line with just a single coefficient. Thus, the following mathematical relation has been obtained:

$$\Delta p = p_{in} - p_{out} = A \cdot \left(\frac{\dot{m}}{\rho}\right)^2 \implies A = (p_{in} - p_{out}) \cdot \left(\frac{\dot{m}}{\rho}\right)^2 \quad (3.43)$$

A block for the calculation of the constant A has been introduced in the Matlab-Simulink environment. Thus, the constant A has been calculated at each simulation step with this block: in so doing, the pressure $p_{footwarmer}$ is used for the p_{out} value and the density ρ is obtained by the current outlet *footwarmer* pressure and outlet COV temperature.

After obtaining the constant A , it has been possible to calculate the Bleed Subsystem mass flow rate with an algebraic loop realized in the Matlab-Simulink environment using the equation:

$$\dot{m} = \sqrt{\frac{p_{in} - p_{cockpit}}{A}} \cdot \rho^2 \quad (3.44)$$

Finally, a *memory* block has been introduced to close the flow rate algebraic loop.

Chapter 4

Vapor Cycle Subsystem modeling

After the Bleed Subsystem, the Vapor Cycle Subsystem has been modeled.

That is a common refrigerant cycle in which the R134a, the tetrafluoroethane, is the refrigerant used. The cycle consists of just few components: compressor, condenser, thermostatic expansion valve (TXV) and evaporator. Nevertheless, the Vapor Cycle Subsystem is more complex than the other Subsystem for two reasons: first of all, because components like the condenser and the evaporator include a phase change of the refrigerant and secondly, because all the dimensions of a component depend on the dimensions of the previous component and, for this reason, a lot of nested algebraic loop have been required.

A Vapor Cycle can be plotted on the Mollier Diagram of the R134a refrigerant, as shown in Figure 4.1. The Mollier Diagram is a graphic representation of the relationship between pressure and enthalpy of a substance that allows a state modification. It can be possible to identify some different region bordered by the bell-shaped curve on this diagram:

- the region inside the bell-shaped curve is the one in which liquid and vapor exist together
- the region to the right of the bell-shaped curve is the vapor one

- the region to the left of the bell-shaped curve is the liquid one

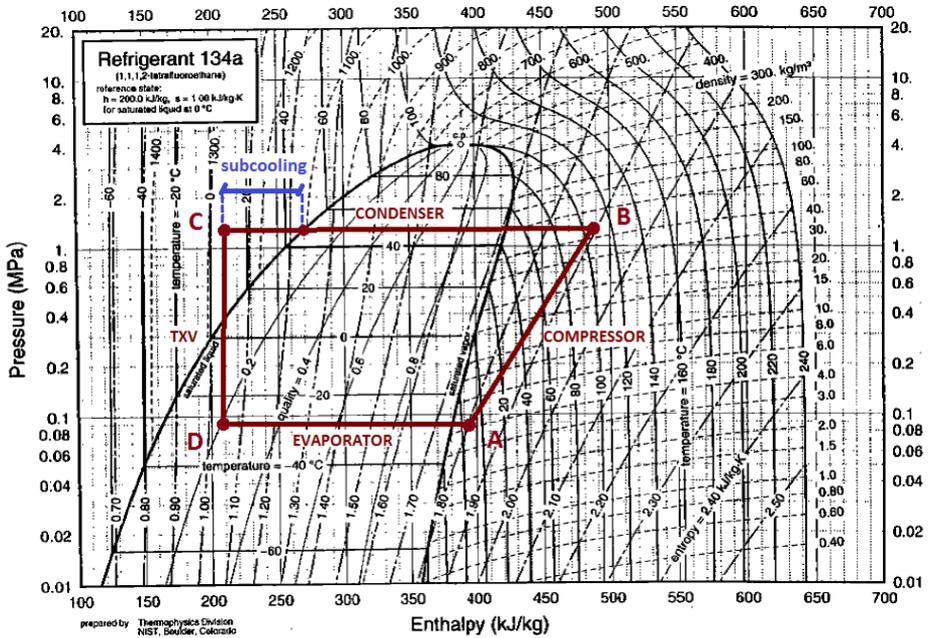


Figure 4.1. Ideal Vapor Cycle on the Mollier Diagram of the R134a refrigerant

We preferred to plot on the Mollier Diagram the ideal cycle instead of the real one in order to simplify the situation and the related calculations. In fact, in the ideal cycle, the evaporator and the condenser generally work at constant pressure and the TXV valve works at constant enthalpy.

The situation shown in Figure 4.1 is just one of the possible conditions of the cycle: each point of the cycle could be found in different regions of the Mollier Diagram and in different conditions. Thus, because of the phase exchange, it is particularly difficult to model the Vapor Cycle Subsystem with the classical modeling in which the various components are individually modeled and then they are connected to each other.

Moreover, because of the high computational cost of a similar system modeling, an high level modeling is required, so we needed to find an alternative way to calculate

the required dimensions to completely describe the cycle.

Therefore, we thought to use a mathematical model to realize the Vapor Cycle Subsystem model. In detail, the Subsystem model was created by calculating the dimensions of each cycle operating point, which corresponds to calculate the dimensions upstream and downstream each component.

Thus, we tried to find some mathematical equations capable to link different dimensions in order to obtain those still unknown. In detail, we checked for the following kind of equations:

- an equation that links the saturation pressure p_{sat} and the saturation temperature T_{sat}
- $h = f(p, T)$
- $\rho = f(p, h)$
- $q = f(p, h)$ where q is the quality

It is quite difficult to find these equations because the great majority of the traditional thermodynamic equations are not able to consider the phase change. In fact, they calculate the variation of those dimensions, thus they consider just the inlet and the outlet values without evaluating the intermediate phase change.

For this reason, we tried to create a specific model for R134a refrigerant: it will be described in Paragraph 4.1.

4.1 The mathematical model

A research about the thermodynamic properties of the R134a refrigerant was carried out by Reiner Tillner-Roth and Hans Dieter Baehr in 1994 at the *Institut fuer Thermodynamik* of the Hannover University [9]. This research shows a fundamental equation of state for the R134a (1,1,1,2 - tetrafluoroethane) which is valid for particular temperature and pressure conditions:

- temperatures between 170 K and 455 K
- pressures up to 70 MPa

The structure of the fundamental equation of state has been obtained, in this research, using a linear regression analysis and a non-linear least squares fitting technique, based on the most accurate available measurements of pressure, density, temperature, speed of sound, vapor pressure and heat capacity.

This research has been also selected by the International Energy Agency (IEA) as an international standard formulation for the thermodynamic properties of R134a refrigerant. In fact, it represents nearly all selected experimental data within their estimated accuracy.

Moreover, by using the fundamental equation of state, other equations have been obtained by Reiner Tillner-Roth and Hans Dieter Baehr. They represent the relations between the dimensionless free energy and thermodynamic properties. In detail, the equations set out in the research allow to obtain a dimension as a function of other two dimensions for a specific functioning point.

In doing so, it is possible to calculate the needed dimension in a specific point of the cycle independently of the region of the Mollier Diagram in which that point is.

Therefore, considering the difficulties in the VCS subsystem modeling because of the phase exchange of the refrigerant, the research by Reiner Tillner-Roth and Hans Dieter Baehr has been used. This choice is due to the fact that the equations of this research allow to calculate the thermodynamic dimensions for the points *A*, *B*, *C*, *D* of the VCS cycle on the Mollier Diagram (Figure 4.1) independently of the phase exchange, because the research equations already include it.

Therefore, a different kind of modeling has been adopted for this subsystem: we decided to model, at high level as lumped parameter block, each component considering its effects on the refrigerant conditions at its inlet and outlet ports.

4.1.1 Main equations

The main equation of the research carried out by Reiner Tillner-Roth and Hans Dieter Baehr [9], used to model the VCS cycle of the system, have been presented in this paragraph.

First of all, two equations of the research [9] have been considered in order to obtain both pressure and enthalpy from temperature and density ($p = f(T, \rho)$ and

$$p = f(T, \rho).$$

Before showing the latter mathematical expressions, the fundamental equation of state for R134a refrigerant has to be explained. The latter, presented in the research [9], is a dimensionless Helmholtz free energy equation and it has the following form:

$$\Phi(\tau, \delta) = \Phi^\circ(\tau, \delta) + \Phi^r(\tau, \delta) \quad (4.1)$$

Where:

- $\tau = \frac{T^*}{T}$ is the inverse reduced temperature, in which the critical parameter $T^* = 374,18K$ is used
- $\delta = \frac{\rho}{\rho^*}$ is the reduced density, in which the critical parameter $\rho^* = 508 \frac{kg}{m^3}$ is used
- $p_0 = 1MPa$

As you can see in Eq. (4.1), the dimensionless form Φ of the fundamental equation of state is split into an ideal part Φ° that describes the ideal gas properties and into a residual part Φ^r that takes in account the behaviour of the real fluid.

The ideal part Φ° is analytically derived from the ideal gas law.

$$\Phi^\circ(\tau, \delta) = a_1^\circ + a_2^\circ \tau + a_3^\circ \ln \tau + \ln \delta + a_4^\circ \tau^{-\frac{1}{2}} + a_5^\circ \tau^{-\frac{3}{4}} \quad (4.2)$$

The coefficients a_i° are in Table 4.1.

a_1°	-1,019535
a_2°	9,047135
a_3°	-1,629789
a_4°	-9,723916
a_5°	-3,927170

Table 4.1. Coefficients a_i° of the equation (4.2)

On the other hand, the mathematical expression for the residual part Φ^r of the fundamental equation of state of R134a is much more complex.

$$\Phi^r = \sum_{i=1}^8 a_i \tau^{t_i} \delta^{d_i} + e^{-\delta} \sum_{i=9}^{11} a_i \tau^{t_i} \delta^{d_i} + e^{-\delta^2} \sum_{i=12}^{17} a_i \tau^{t_i} \delta^{d_i} + e^{-\delta^3} \sum_{i=18}^{20} a_i \tau^{t_i} \delta^{d_i} + a_{21} e^{-\delta^4} \tau^{t_{21}} \delta^{d_{21}} \quad (4.3)$$

The coefficients a_i are shown in Table 4.2; the coefficients t_i in table 4.3; finally, the coefficients d_i in Table 4.4.

$\mathbf{a_1}$	$0,5586817 \cdot e^{-1}$	$\mathbf{a_8}$	$-0,4781652 \cdot e^{-1}$	$\mathbf{a_{15}}$	$0,2057144 \cdot e^0$
$\mathbf{a_2}$	$0,4982230 \cdot e^0$	$\mathbf{a_9}$	$0,1423987 \cdot e^{-1}$	$\mathbf{a_{16}}$	$-0,5000457 \cdot e^{-2}$
$\mathbf{a_3}$	$0,2458698 \cdot e^{-1}$	$\mathbf{a_{10}}$	$0,3324062 \cdot e^0$	$\mathbf{a_{17}}$	$0,4603262 \cdot e^{-3}$
$\mathbf{a_4}$	$0,8570145 \cdot e^{-3}$	$\mathbf{a_{11}}$	$-0,7485907 \cdot e^{-2}$	$\mathbf{a_{18}}$	$-0,3497836 \cdot e^{-2}$
$\mathbf{a_5}$	$0,4788584 \cdot e^{-3}$	$\mathbf{a_{12}}$	$0,1017263 \cdot e^{-3}$	$\mathbf{a_{19}}$	$0,6995038 \cdot e^{-2}$
$\mathbf{a_6}$	$-0,1800808 \cdot e^{+1}$	$\mathbf{a_{13}}$	$-0,5184567 \cdot e^0$	$\mathbf{a_{20}}$	$-0,1452184 \cdot e^{-1}$
$\mathbf{a_7}$	$0,2671641 \cdot e^0$	$\mathbf{a_{14}}$	$-0,8692288 \cdot e^{-1}$	$\mathbf{a_{21}}$	$-0,1285458 \cdot e^{-3}$

Table 4.2. Coefficients a_i of the equation (4.3)

$\mathbf{t_1}$	$-1/2$	$\mathbf{t_8}$	2	$\mathbf{t_{15}}$	6
$\mathbf{t_2}$	0	$\mathbf{t_9}$	1	$\mathbf{t_{16}}$	10
$\mathbf{t_3}$	0	$\mathbf{t_{10}}$	3	$\mathbf{t_{17}}$	10
$\mathbf{t_4}$	0	$\mathbf{t_{11}}$	5	$\mathbf{t_{18}}$	10
$\mathbf{t_5}$	$3/2$	$\mathbf{t_{12}}$	1	$\mathbf{t_{19}}$	18
$\mathbf{t_6}$	$3/2$	$\mathbf{t_{13}}$	5	$\mathbf{t_{20}}$	22
$\mathbf{t_7}$	2	$\mathbf{t_{14}}$	5	$\mathbf{t_{21}}$	50

Table 4.3. Coefficients t_i of the equation (4.3)

The equation of state for the pressure performed in the research [9] can be now

d_1	2	d_8	2	d_{15}	1
d_2	1	d_9	5	d_{16}	2
d_3	3	d_{10}	2	d_{17}	4
d_4	6	d_{11}	2	d_{18}	1
d_5	6	d_{12}	4	d_{19}	5
d_6	1	d_{13}	1	d_{20}	3
d_7	1	d_{14}	4	d_{21}	10

Table 4.4. Coefficients d_i of the equation (4.3)

introduced.

$$\frac{p(\tau, \delta)}{RT} = \rho(1 + \delta\Phi_\delta^r) \quad (4.4)$$

On the other hand, Eq. (4.5) is the equation of the enthalpy.

$$\frac{h(\tau, \delta)}{RT} = 1 + \tau(\Phi_\tau^\circ + \Phi_\tau^r) + \delta\Phi_\delta^r \quad (4.5)$$

Moreover, the equations of the isochoric heat capacity and the isobaric heat capacity have been introduced, too.

$$\frac{c_v(\tau, \delta)}{R} = -\tau^2(\Phi_{\tau\tau}^\circ + \Phi_{\tau\tau}^r) \quad (4.6)$$

$$\frac{c_p(\tau, \delta)}{R} = \frac{c_v}{R} + \frac{(1 + \delta\Phi_\delta^r - \delta\tau\Phi_{\delta\tau}^r)^2}{1 + 2\delta\Phi_\delta^r + \delta^2\Phi_{\delta\delta}^r} \quad (4.7)$$

In the previous equations, R is the ideal gas constant of the R134a refrigerant and it is obtained by $R = \frac{R_m}{M}$ where R_m is the universal gas constant and M is the molar mass of the R134a.

$$R = \frac{R_m}{M} = \frac{8,314471 \cdot 10^{-3} \frac{kJ}{molK}}{0,102032 \frac{kg}{mol}} = 0,08148885644 \frac{kJ}{kgK} \quad (4.8)$$

Both in Eq. (4.4) and Eq. (4.5), other three terms can be noted: Φ_τ° , Φ_τ^r and Φ_δ^r .

The latter expressions are some derivatives of the fundamental equation of state Φ . In detail, the derivatives of the ideal part Φ° and of the residual part Φ^r are required.

In accordance with Eq. (4.2), the first and second derivatives of the ideal part with respect to τ have been considered.

$$\frac{d}{d\tau}(\Phi^\circ) = \Phi_\tau^\circ = a_2^\circ + \frac{a_3^\circ}{\tau} + \sum_{j=4}^{N^\circ} a_j^\circ t_j^\circ \tau^{t_j^\circ - 1} \quad (4.9)$$

$$\frac{d^2}{d\tau^2}(\Phi^\circ) = \Phi_{\tau\tau}^\circ = -\frac{a_3^\circ}{\tau^2} + \sum_{j=4}^{N^\circ} a_j^\circ t_j^\circ (t_j^\circ - 1) \tau^{t_j^\circ - 2} \quad (4.10)$$

On the other hand, the first (Eq. (4.11) and Eq. (4.12)) and second (Eq. (4.13) and Eq. (4.14)) derivatives of the residual part (Eq. (4.3)) with respect to δ and τ have been considered.

$$\frac{d}{d\delta}(\Phi^r) = \Phi_\delta^r = \sum_{i=1}^{N_0} a_i d_i \delta^{d_i - 1} \tau^{t_i} + \sum_{k=1}^4 \left(e^{-\delta^k} \cdot \sum_{i=N_{k-1}+1}^{N_k} a_i (d_i - k\delta^k) \delta^{d_i - 1} \tau^{t_i} \right) \quad (4.11)$$

$$\frac{d}{d\tau}(\Phi^r) = \Phi_\tau^r = \sum_{i=1}^{N_0} a_i t_i \delta^{d_i} \tau^{t_i - 1} + \sum_{k=1}^4 \left(e^{-\delta^k} \cdot \sum_{i=N_{k-1}+1}^{N_k} a_i t_i \delta^{d_i} \tau^{t_i - 1} \right) \quad (4.12)$$

$$\frac{d^2}{d\delta^2}(\Phi^r) = \Phi_{\delta\delta}^r = \sum_{i=1}^{N_0} a_i d_i (d_i - 1) \delta^{d_i - 2} \tau^{t_i} +$$

$$+ \sum_{k=1}^4 \left(e^{-\delta^k} \cdot \sum_{i=N_{k-1}+1}^{N_k} a_i [d_i^2 - d_i - k\delta^k (2d_i + k - 1 + k\delta^k)] \delta^{d_i - 2} \tau^{t_i} \right) \quad (4.13)$$

$$\begin{aligned} \frac{d^2}{d\tau^2} (\Phi^r) &= \Phi_{\tau\tau}^r = \sum_{i=1}^{N_0} a_i t_i (t_i - 1) \delta^{d_i} \tau^{t_i-2} + \\ &+ \sum_{k=1}^4 \left(e^{-\delta^k} \cdot \sum_{i=N_{k-1}+1}^{N_k} a_i t_i (t_i - 1) \delta^{d_i} \tau^{t_i-2} \right) \end{aligned} \quad (4.14)$$

Finally, the mixed derivative of the ideal part has been considered.

$$\begin{aligned} \frac{d^2}{d\delta d\tau} (\Phi^r) &= \Phi_{\delta\tau}^r = \sum_{i=1}^{N_0} a_i d_i t_i \delta^{d_i-1} \tau^{t_i-1} + \\ &+ \sum_{k=1}^4 \left(e^{-\delta^k} \cdot \sum_{i=N_{k-1}+1}^{N_k} a_i (d_i - k\delta^k) \delta^{d_i-1} \tau^{t_i-1} \right) \end{aligned} \quad (4.15)$$

About the implementation of Eq. (4.4) and Eq. (4.5) in the Matlab-Simulink environment, we thought to implement in Matlab and then recall them in the Simulink environment using a *Matlab Function* block.

Although the higher computational cost, we preferred to implement them in Matlab because they are really complex equations, with a large number of summations, which means a lot of iteration. Therefore, a large number of nested algebraic loop would be required.

For this reason, a *Matlab Function* has been created in the Matlab-Simulink environment for each equation.

After that, we thought to realize the inverse calculations, too. Thus, we tried to obtain $\rho = f(p, T)$ and $T = f(p, \rho)$ from Eq. (4.4) and $\rho = f(h, T)$ and $T = f(h, \rho)$ from Eq. (4.5). As a consequence, other four *Matlab Function* blocks (one for each calculation) have been created in the Matlab-Simulink environment.

In detail, these inverse equations have been obtained within each *Matlab Function*, by using an iterative calculation on the required dimension.

Comparing the results obtained from the *Matlab Function* and the experimental results presented in the research [9], it has been possible to note that all of them matched, except the ones concerning the two density calculations. In detail, the density results matched only if the calculated point was in the vapor region of

the Mollier Diagram (to the right of the bell-shaped curve), but, if the calculated point was in the other two regions, the results would not be reliable. This is due to the fact that the density is not a monotonic function, therefore the two *Matlab Functions* had to be modified.

4.1.2 Density implementation

As explained in 4.1.1, the density calculation cannot be realized simply with the inverse *Matlab Function*, but a dedicated one is required.

Therefore, we decided to split the problem into the three situations and to model the density in a different way, depending on the region of the Mollier Diagram where the generic functioning point is. In fact, since the density is not a motonic function, the Mollier Diagram has to be virtually divided in regions:

- the region of the liquid to the left of the bell-shaped curve
- the region of liquid and vapor existing together inside the bell-shaped curve
- the region of the vapor to the right of the bell-shaped curve

It has been noted that the three regions are separated by the bell-shaped curve which identifies the saturation conditions, therefore we thought to introduce some other equations presented in the research [9] for the calculation of those conditions.

$$\vartheta \ln \frac{p_{sat}}{p_c} = -7,686556 \theta + 2,311791 \theta^{\frac{3}{2}} - 2,039554 \theta^2 - 3,583758 \theta^4 \quad (4.16)$$

Where:

- $\vartheta = \frac{T}{T^*}$ is the reduced temperature, in which the critical parameter $T^* = 374,18K$ is used
- $\theta = 1 - \vartheta$
- $p_c = 4,05629MPa$

Eq. (4.16) allows to calculate the saturation pressure for a specific temperature value. Moreover, following there are the mathematical expression of the saturated liquid density (Eq. (4.17)) and the one of the saturated vapor density (Eq. (4.18)).

$$\rho_{saturated\ liquid} = 518,20 + 884,13 \theta^{\frac{1}{3}} + 485,84 \theta^{\frac{2}{3}} + 193,29 \theta^{\frac{10}{3}} \left[\frac{kg}{m^3} \right] \quad (4.17)$$

$$\ln \left(\frac{\rho_{saturated\ vapor}}{\rho_0} \right) = -2,837294 \theta^{\frac{1}{3}} - 7,875988 \theta^{\frac{2}{3}} + 4,478586 \theta^{\frac{1}{2}} +$$

$$-14,140125 \theta^{\frac{3}{4}} - 52,361297 \theta^{\frac{11}{2}} \quad (4.18)$$

After finding the equations for the calculation of the saturation conditions, the *Matlab Function* has been realized.

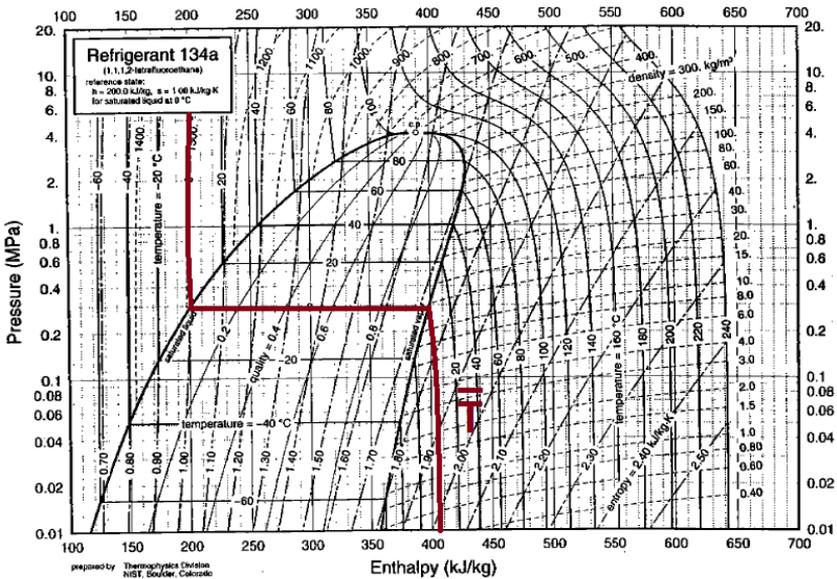


Figure 4.2. Shape of the curve of a generic temperature \bar{T} on the Mollier Diagram of the R134a refrigerant

Figure 4.2 shows the evolution of the curve for a generic temperature on the Mollier Diagram. It has been noted that the temperature is a monotonic function and, in particular, inside the bell-shaped curve the pressure has a constant value, that is the saturation pressure.

Therefore, the density has been calculated introducing the three regions of the Mollier Diagram by the use of an iterative calculation with an *if* cycle on the inlet pressure. In fact, referring to Figure 4.2, the evolution of temperature for a selected temperature \bar{T} has been divided depending on the respective pressure \bar{p} compared to the saturation one. Therefore:

- if \bar{p} value is lower than the saturation pressure, the functioning point is to the right of the bell-shaped curve, thus it is in the vapor region. Therefore, the density in this region can be evaluated by an iterative calculation with the inverse *Matlab Function*, where the iteration is carried out between two extreme values of density: $\rho_{min} = 0 \frac{kg}{m^3}$ and $\rho_{max} = \rho_{saturated\ vapor}$
- if \bar{p} value is greater than the saturation pressure, the functioning point is to the left of the bell-shaped curve, thus it is in the liquid region. Therefore, the density in this region can be evaluated by an iterative calculation with the inverse *Matlab Function*, where the iteration is carried out between two extreme values of density: $\rho_{min} = \rho_{saturated\ liquid}$ and $\rho_{max} = 2000 \frac{kg}{m^3}$ (the value of ρ_{max} has been chosen because it is an enough high value: as you can see on the Mollier Diagram of the R134a (Figure 4.2), all the possible functioning points must have a lower density)
- if \bar{p} value is exactly equal to the saturation pressure, the functioning point is inside the bell-shaped curve, in which both pressure and temperature are constant but density and enthalpy are not, therefore a different logic has to be implemented

The latter condition of the previous list is the more complex to model.

The density and the entalpy, which are the only dimensions changing under the bell-shaped curve of the Mollier Diagram, take a linear variation in this region of the diagram. Therefore, we thought to obtain a mathematical equation with these two dimensions in order to calculate the correct density in a specific functioning point.

First of all, we needed to obtain the two extreme enthalpies, which means to calculate the saturated liquid enthalpy and the saturated water vapor enthalpy. Thus, Eq. (4.5) (the calculation of enthalpy from temperature and density) has been used, considering saturated liquid density and the saturated water vapor density respectively.

Therefore, knowing the extreme values of both enthalpy and density, and considering the linear evolution of these dimensions, we thought to use mathematical proportions. The latter has been set up considering the density and enthalpy range under the bell-shaped curve for a selected temperature and saturation pressure as a percentage.

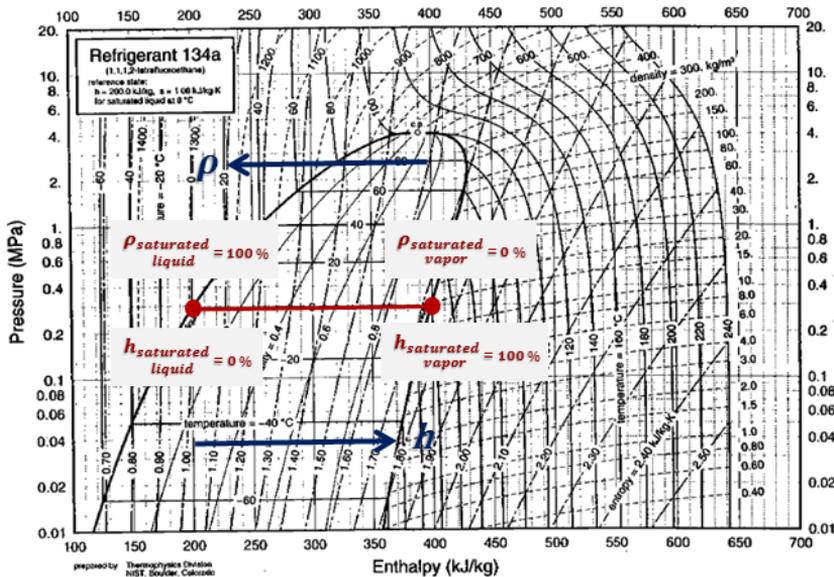


Figure 4.3. Distribution of density and enthalpy inside the bell-shaped curve on the Mollier Diagram of the R134a refrigerant

In fact, as you can see in Figure 4.3, the percentage for enthalpy and density has to be splitted because their evolutions are inversely proportional: the density decreases with increasing enthalpy, and vice versa. Therefore, the enthalpy has been considered with 0% value at the point on the bell-shaped curve in which there are saturated liquid conditions and with 100% value at the point on the bell-shaped curve in which there are saturated vapor density and enthalpy has been considered

at 100% value. On the other hand, the density has been considered with inverted values.

After the identification of the percentages, the proportions have been written.

$$H : H\% = r_H : 100 \quad (4.19)$$

$$\rho : \rho\% = r_\rho : 100 \quad (4.20)$$

Eq. (4.19) is the mathematical proportion for the enthalpy, in which H is the required adapted enthalpy $H = H_{required} - H_{saturated\ liquid}$, $H\%$ is the percentage of the required enthalpy and r_H is the range of the enthalpy $r_H = H_{saturated\ vapor} - H_{saturated\ liquid}$. Eq. (4.20) is the mathematical proportion for the density, in which ρ is the required adapted density, $\rho\% = 100 - H\%$ is the percentage of the required density and r_ρ is the range of the density $r_\rho = H_{saturated\ liquid} - H_{saturated\ vapor}$.

Therefore, the required density has been obtained by combining Eq. (4.19) and Eq. (4.20), thus it has been implemented in the density *Matlab Function*.

Moreover, another condition has to be considered, yet. In fact, all the conditions listed above occur if the selected temperature \bar{T} is lower than the critical value ($T^* = 374,18K$). But, if the selected temperature is higher than the critical value, the functioning point is above the bell-shaped curve, therefore there is not saturation. For this reason, in this condition the density can be calculated with the inverse *Matlab Function*, in which the iteration is carried out between the two extreme values of density: $\rho_{min} = 0 \frac{kg}{m^3}$ and $\rho_{max} = 2000 \frac{kg}{m^3}$.

4.2 Cycle implementation

After creating all the necessary *Matlab Function* for the calculation of the cycle conditions, it has been possible to think to its implementation in the Matlab-Simulink environment.

Referring to the ideal cycle in Figure 4.1, we chose to start the cycle from *point A*, which is the one downstream the evaporator air side. In fact, the air flow rate

from the cockpit goes through an inlet port entering the evaporator. After the heat exchange between the latter air flow and the R134a refrigerant, the air leaves the evaporator and enters the cockpit again. Therefore, *point A* represents the point in which the refrigerant ideally starts a new cycle before the heat exchange with the air flow.

Moreover the refrigerant mass flow rate depends on the density of the refrigerant entering the compressor and it is constant for all the cycle, until it restarts at *point A* again. Therefore, for each cycle, the density ρ_A and, as a consequence, the respective mass flow rate have to be recalculated exactly at *point A*.

For these reasons, starting the cycle from *point A* seems to be the correct choice. First of all, in addition to the mathematical model explained in 4.1, some other equations were required to describe the components functioning of the cycle.

Considering that we had the Mollier Diagram of the R134a in which the enthalpy is shown and it is calculated with the *Matlab Functions*, Eq. (4.21), Eq. (4.22) and Eq. (4.23) were found.

$$\Delta H_{evaporator} = H_A - H_D = \frac{\dot{Q}_{evaporator}}{\dot{m}_{R134a}} \quad (4.21)$$

$$\Delta H_{condenser} = H_C - H_B = \frac{\dot{Q}_{condenser}}{\dot{m}_{R134a}} \quad (4.22)$$

$$\Delta H_{compressor} = H_B - H_A = \frac{\dot{L}_{compressor}}{\dot{m}_{R134a}} \quad (4.23)$$

Where the enthalpy of all the four functioning points are used and $\dot{Q}_{evaporator}$ and $\dot{Q}_{condenser}$ are respectively the heat flows of the evaporator and the condenser. On the other hand, $\dot{L}_{compressor}$ is the compressor power.

$$\dot{Q}_{condenser} = f(T_{saturation\ condenser}, \dot{m}_{air\ condenser}, T_{air\ condenser}) \quad (4.24)$$

$$\dot{Q}_{evaporator} = f(T_{saturation\ evaporator}, \dot{m}_{air\ evaporator}, T_{air\ evaporator}) \quad (4.25)$$

Moreover, as previously described, the mass flow rate of the R134a refrigerant

shown in Eq. (4.21), Eq. (4.22) and Eq. (4.23) is constant and it can be calculated as expressed in Eq. (4.26).

$$\dot{m}_{R134a} = \eta \cdot cc \cdot RPM \cdot \rho_A \quad (4.26)$$

Where, η is the efficiency of the compressor, cc is the cylinder displacement expressed in $[m^3]$, RPM is the speed of the compressor expressed as revolution per minute and ρ_A is the density of the refrigerant at *point A*.

Finally, an isentropic transformation has been considered to describe the compressor behaviour. In this respect, Eq. (4.27) has been used.

$$T \cdot P^{\frac{1-\gamma}{\gamma}} = constant \quad (4.27)$$

Where γ is the ratio of specific heat coefficient and it can be obtained as $\gamma = \frac{c_p}{c_v}$. Moreover, in addition to all the latter equations, we had a TXV performance map, too. It is similar to the one in Figure 4.4.

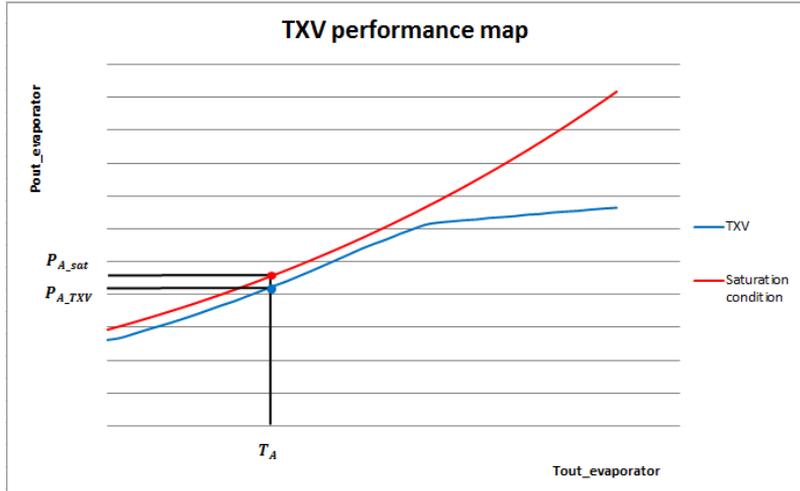


Figure 4.4. TXV performance map compared to the saturation condition of the refrigerant

The blue curve in Figure 4.4 represents the performance curve of the TXV valve. This curve shows an incrementation of the evaporator outlet pressure corresponding to the incrementation of its outlet temperature until the valve is fully-open. When this condition occurs, a slope reduction arises, because the valve can no longer be opened and, as a consequence, it determines a lower increasing of the evaporator outlet pressure.

On the other hand, the red curve represents the saturation condition of the R134a refrigerant. It can be possible to note that, the evaporator outlet pressure of the TXV valve is always lower than the correspondent saturation value, for any evaporator outlet temperature. For this reason, it is ensured that the *point A* of the cycle is always to the right of the bell-shaped curve, therefore the refrigerant is certainly a vapor downstream the evaporator.

There were a compressor performance map, too, and it is similar to the one in Figure 4.5, from which isentropic effectiveness and volumetric effectiveness can be retrieved.

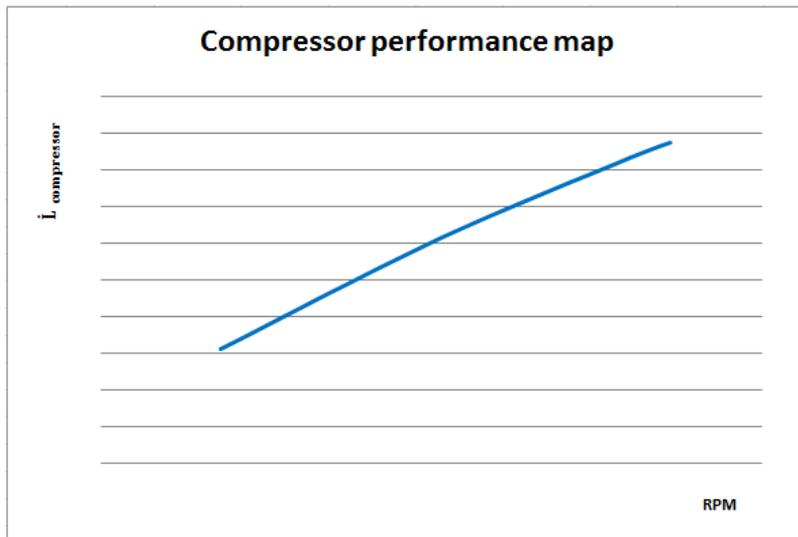


Figure 4.5. Compressor performance map

Finally, we had a condenser and an evaporator performance maps. They are similar to the ones in Figure 4.6 and Figure 4.7.

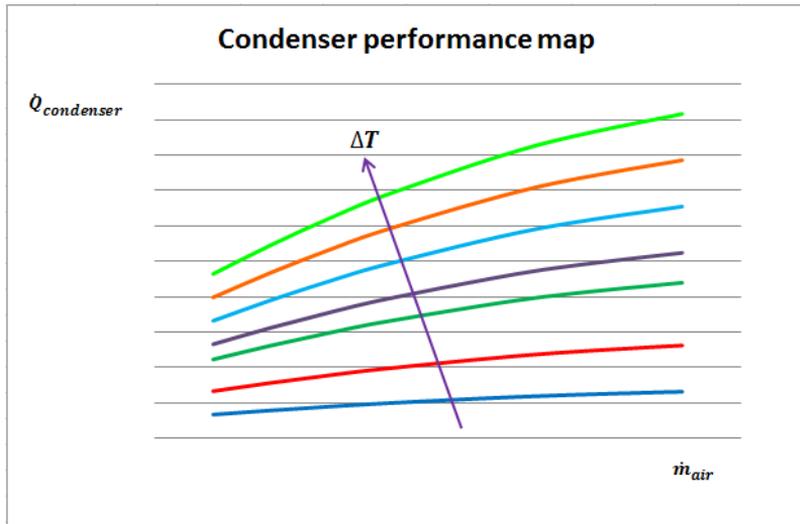


Figure 4.6. Condenser performance map

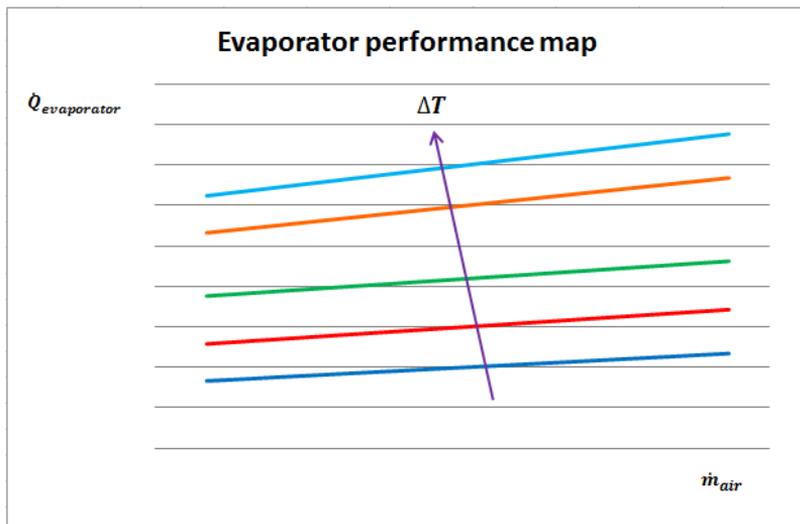


Figure 4.7. Evaporator performance map

Thus, after gathering the main equations and the performance maps to describe the cycle, a way to implement and close it, had to be found.

First of all, we thought to use the subcooling as dominant parameter. It shows how far is *point C* from the bell-shaped curve. In fact, a well defined subcooling value is required by the cycle, so it is reasonable to consider it as dominant parameter. In detail, the subcooling for the functioning conditions of the examined aircraft Vapor Cycle Subsystem is zero and it means that *point C* has to be exactly on the bell-shaped curve.

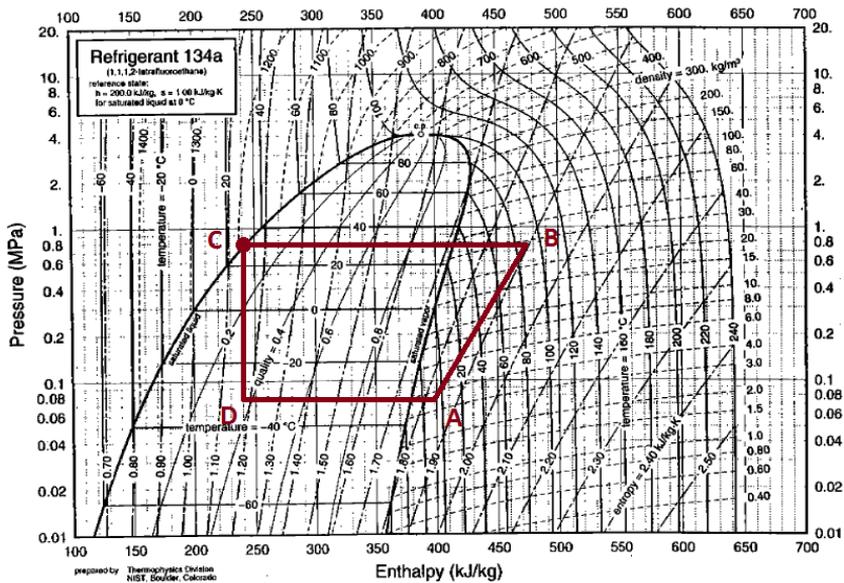


Figure 4.8. Generic ideal cycle with the requested zero subcooling

Moreover, an algebraic loop had to be implemented. In so doing, using the equations of the research [9] described in Paragraphs 4.1.1 and 4.1.2, the cycle had to calculate all the dimensions at every time steps starting from a trying condition for *point A*, until a convergence with the requested subcooling was obtained.

In detail, a trying temperature for *point A* had been considered and, with it, the respective pressure P_A had been obtained from the performance map of the TXV implemented with a *Lookup Table 1-D*. Therefore, the density ρ_A and the enthalpy H_A had been calculated, too, using the respective *Matlab Function*.

After the calculation of the *point A* condition density, the mass flow rate of the R134a refrigerant \dot{m}_{R134a} could be evaluated from Eq. (4.26).

Thereafter, the thermodynamic transformation from *point A* to *point B* of the compressor had to be implemented. The isentropic transformation of Eq. (4.27) should have been used to model it, but, because of its high computational cost, its implementation had been postponed. In fact, we thought to implement at first a simpler and less accurate, but with speedier simulation times solution to gain a clear overview of the Vapor Cycle Subsystem and analyze more easily how to model it, and only when the model will be complete, the isentropic transformation will be implemented.

Therefore, at first the compressor performance map (Figure 4.5) had been preferred to the modeling. It had been used to obtain the compressor power $\dot{L}_{compressor}$ knowing its speed *RPM*. In fact, a *Lookup Table 1-D* had been built in the Matlab/Simulink environment.

Thus, the enthalpy of *point B* had been evaluated:

$$H_B = H_A + \frac{\dot{L}_{compressor}}{\dot{m}_{R134a}} \quad (4.28)$$

After that, because neither the pressure nor the temperature at *point B* were known, we decided to calculate some dimensions at *point D*. In fact, the pressure of the *point D* was already known and it is the same at *point A* and, because *point D* is always inside the bell-shaped curve, P_D is always a saturated pressure.

Therefore, the saturated temperature T_D was obtained by using the *Matlab Function* for the calculation of the saturated temperature from the saturated pressure. Thus, a *Lookup table 2-D* was built in the model with the evaporator data: the temperature variation ΔT between the evaporator air temperature and the saturated temperature at *point D* was been linked to the input u_1 ; instead, the evaporator air mass flow rate was been linked to the input u_2 . Thus, the output was the heat flow of the evaporator $\dot{Q}_{evaporator}$ and the enthalpy at *point D* was obtained with it using Eq. (4.21).

Moreover, the enthalpy at points *D* and *C* were assumed the same, since an adiabatic expansion inside the TXV valve had been considered. Therefore, after building another *Lookup table 2-D* similar to the previous one, but for the condenser, the heat flow of the condenser $\dot{Q}_{condenser}$ could be obtained implementing Eq. (4.22). It can be noted that the compressor power value obtained had to be evaluated

for the specific situation considered. For this reason it had to be scaled with the pressure ratio.

$$\beta = \frac{P_{out}}{P_{in}} = \frac{P_{condenser}}{P_{evaporator}} \quad (4.29)$$

Moreover, it can be also noted that the *Lookup Table* data were related to a specific compressor functioning point, therefore, before using it, it had to be scaled again with the pressure ratio of that specific point (β_0).

$$\dot{L}_{compressor} = \dot{L}_{compressor}(\beta_0) \cdot \frac{\beta}{\beta_0} \quad (4.30)$$

Because of this remark, the latter cycle implementation could not be used because the condenser pressure for the pressure ratio β was not yet known at this cycle step. Thus, the cycle had to be thought again.

We decided to start the cycle from *point A* with the calculation of all the dimensions of this point. Then, considering $P_A = P_D$, the temperature of *point D* had been obtained from the performance map of the TXV valve. Using this T_D , the heat flow of the evaporator $\dot{Q}_{evaporator}$ had been obtained from the respective *Lookup Table* and the enthalpy H_D had been calculated.

Moreover, we thought to use the *Matlab Function* for the calculation of the saturated pressure and the saturated temperature from the respective saturated enthalpy. In fact, as seen before, the subcooling for this system is zero, therefore *point C* is exactly on the bell-shaped curve and its enthalpy is the saturated one for a specific saturated pressure value. Thus, knowing that $H_D = H_C$, the pressure P_C and the temperature T_C had been obtained.

The compressor power had been obtained with the compressor performance map, too, and it had been scaled as described in Eq. (4.30). The latter value had been used to calculate the enthalpy of the *point B* from Eq. (4.28).

Thereafter, the heat flow of the condenser $\dot{Q}_{condenser}$ had been obtained and the enthalpy of a new *point C* in the cycle had been calculated with it.

$$H_{C_{new}} = H_B - \frac{\dot{Q}_{condenser}}{\dot{m}_{R134a}} \quad (4.31)$$

Considering the adiabatic expansion, the enthalpy of the new *point D* were known,

too. In fact, as already said, $H_{D_{new}} = H_{C_{new}}$.

Moreover, the evaporator heat exchange had to be the same, therefore we had $\Delta H_{evaporator} = \Delta H_{evaporator_new}$. Thus, the enthalpy of the new *point A* could be obtained.

$$\Delta H_{evaporator} = H_A - H_D = H_{A_{new}} - H_{D_{new}} \quad (4.32)$$

$$\implies H_{A_{new}} = \Delta H_{evaporator} - H_{D_{new}} \quad (4.33)$$

After that, we needed to obtain the temperature of this new *point A* to close the algebraic loop.

In this respect, we thought to create a new *Matlab Function* with a relation between temperature pressure and enthalpy to calculate $T_{A_{new}}$. A similar direct equation did not exist in the research [9], but it could be obtained combining some of its other equation seen in Paragraph 4.1.1.

Therefore, after obtaining the calculation for the temperature $T_{A_{new}}$, the algebraic loop were closed. All the previous equation were implemented in the Matlab-Simulink environment with elementary blocks and with the *Matlab Functions* for the necessary calculations to create the algebraic loop implemented with a *Unit Delay* block on the temperature of the *point A*.

However, the cycle thus created was not able to converge sistematically. This was because just a single loop on the temperature T_A was not enough for the convergence of a similar loop. As a consequence, we thought to implement another loop in the cycle.

Since the subcooling for the functioning conditions of the examined VCS, as already explained, is zero, thus *point C* has to be exactly on the bell-shaped curve, we thought to creat the other algebraic loop exactly on this condition.

The cycle had been rewritten.

We started the cycle again from *point A* with the calculation of all the dimensions of this point. The calculation of the R134a refrigerant mass flow rate had been obtained with the density ρ_A as shown in Eq. (4.26). Then, after the determination of $\Delta H_{compressor}$, the enthalpy H_B was calculated. After that, a lookup table for the condenser was implemented in the cycle obtaining the condenser heat exchange

and the condenser enthalpy variation $\Delta H_{condenser}$, too. Thus, using the enthalpy H_B previously calculated, the enthalpy H_C and as a consequence H_D ($H_D = H_C$) were obtained.

After that, we decided to introduce the loop on *point C* condition. It had been assumed that the calculated enthalpy H_C could be the saturated liquid one, thus a calculation of temperature T_C in order to consider it as the saturated one corresponding to the saturated liquid enthalpy H_C , had been introduced. In this respect, a *Matlab Function* were created with this loop.

After the achievement of the enthalpy H_C convergency, its value was used to obtain the enthalpy of the new *point A* with Eq. (4.33), and the temperature of this new *point A* to close the algebraic loop was calculated by the pressure-temperature-enthalpy *Matlab Function*.

Running the model thus built, it was not able to complete the algebraic loop and provide some result. The calculations of a loop similar to that one, led to obtain a *point C* really close to the triple point and the simulation were terminated quite early.

Even this approach to the VCS modeling was conceptually incorrect.

Thereafter we tried to find a mathematical correlation between the pressure P_C and enthalpy H_C at *point C* and the liquid saturated enthalpy H_{sat_liquid} corresponding to P_C .

In this respect, Eq. (4.34) were found.

$$P_{C_{new}} = P_C \left(1 - \frac{H_{sat_liquid} - H_C}{H_{sat_liquid}} \right) \quad (4.34)$$

Therefore, the calculation of *point C* condition were replaced by the latter equation, introduced in the model with elementary block. In doing so, the model run until the convergence between the *point C* enthalpy and the liquid saturated one were obtained. In fact, P_C was equal to $P_{C_{new}}$ when $H_{sat_liquid} = H_C$:

$$P_{C_{new}} = P_C \left(1 - \frac{H_{sat_liquid} - (H_C \equiv H_{sat_liquid})}{H_{sat_liquid}} \right) = P_C (1 - 0) = P_C \quad (4.35)$$

On the other hand, if the *point C* was to the left of the saturated liquid curve

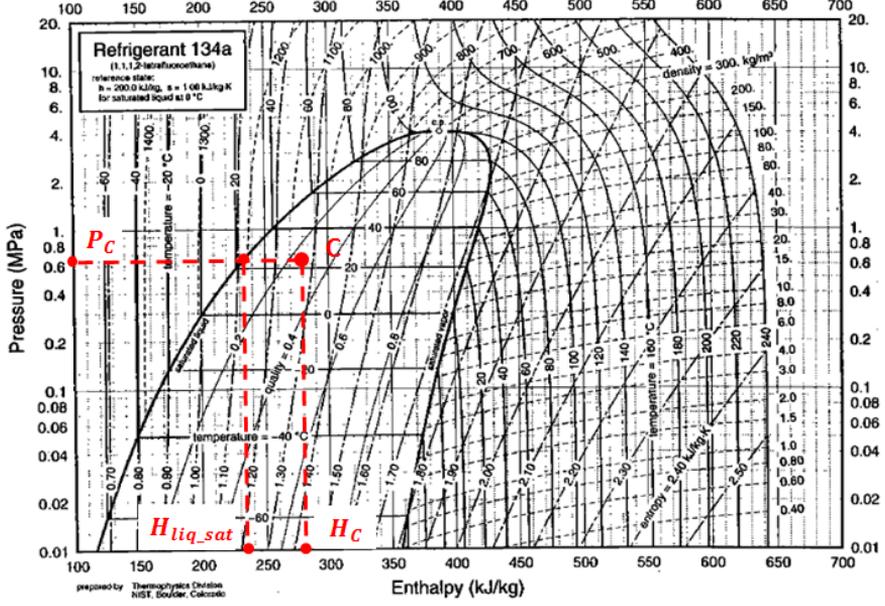


Figure 4.9. Hypothetical *point C* and correlation between P_C , H_C and H_{sat_liquid} on the Mollier Diagram

$H_{sat_liquid} > H_C$, therefore $\frac{H_{sat_liquid} - H_C}{H_{sat_liquid}} > 0$ and $P_{C_{new}} < P_C$ and vice versa.

Running this new model, the simulation did not stop as was the case with the previous model, but, it was still incorrect, because at the end of the simulation there was not a convergence between P_C and $P_{C_{new}}$. Moreover, the achievement of a convergence temperature value $T_{A_{new}}$ required very long simulation times and a high computational costs. As a consequence, the model thus built was unable to provide a convergence within the selected simulation time.

In order to solve the latter problem, a change to the model had been provided. In fact, the calculation of the enthalpy $H_{A_{new}}$ with the implementation of Eq. (4.33) determined greater variations of the enthalpy obtained at two subsequent time steps. Therefore, we thought to add a calculation implementing a weighted average.

$$H_{A_{calculated}} = 0.9 \cdot H_A + 0.1 \cdot H_{A_{new}} \quad (4.36)$$

In so doing, Eq. (4.33) was still used allowing to the calculated enthalpy to move closer the convergence solution, but, at the same time the enthalpy is close enough to the previous H_A to find the convergence value within any selected simulation time.

On the other hand, as far as concerns the pressure $P_{C_{new}}$, the problem previous described occurred because the algebraic loops for $T_{A_{new}}$ and for $P_{C_{new}}$ were nested: it means that the $P_{C_{new}}$ loop had to run for a specific T_A value and only when the convergence was found, a new value of temperature could be used until the convergence of the second loop was found, too.

We tried to solve the problem of the two nested loop creating the calculation for the pressure $P_{C_{new}}$ directly in the *Matlab Function* for the calculation of T_A : an *if* cycle was implemented to determine when the new value of *point A* could be used. After applying the changes, the model was run and the evolution of the pressure $P_{C_{new}}$ was plotted.

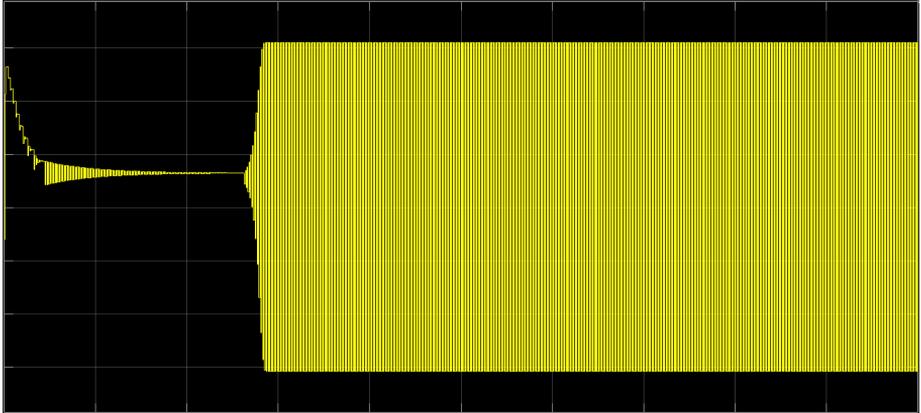


Figure 4.10. Simulation plot of the pressure P_C evolution

Figure 4.10 shows a particular evolution of the pressure P_C . In fact, the curve tended to the convergence value and after achieving it, the pressure became unstable, thus it started to oscillate between two opposite values. This was due to the equation used to calculate the pressure (Eq. (4.34)). In fact, when the calculation

arrived in neighbourhood of the solution, the enthalpy of the *point C* was alternatively to the right or to the left of the saturated liquid enthalpy since it was not able to get closer to the convergence solution.

Analyzing the equation, it was noted that the subtraction $H_{sat_liquid} - H_C$ is alternatively positive and negative and it explains why the solution oscillated.

Therefore, we thought to proceed with a logical approach: Eq. (4.34) has been replaced by the implementation of an iterative calculation realized with elementary blocks in the Simulink model. The iteration is carried out between a maximum and a minimum value of pressure properly chosen depending on the functioning conditions.

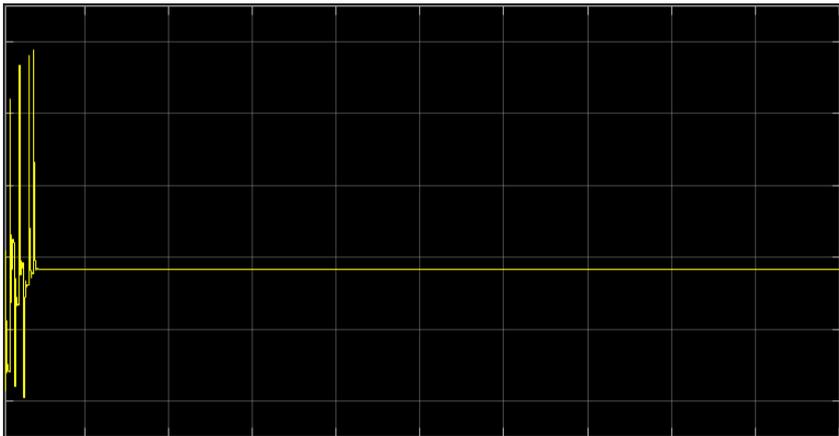


Figure 4.11. Simulation plot of the pressure P_C evolution with the iterative calculation

With the iterative calculation, the pressure is become stable (as shown in Figure 4.11). But, in so doing, as a result, the temperature $T_{A_{new}}$ is become instable.

Therefore, an iterative calculation has been implemented on the temperature $T_{A_{new}}$, too. In so doing, the respective *Matlab Function* for the calculation of that temperature has been replaced by this iteration and it has determined a resultant decreases in the computational costs.

Moreover, an *if action subsystem* block has been introduced in the temperature iterative calculation in order to implement the calculation only when a convergence of the pressure $P_{C_{new}}$ is found.

Running the latter model, it finally provides correct results.

Moreover, as previously anticipated, after the completion of the model, the isentropic transformation (Eq. (4.27)) has been implemented in place of the compressor performance map to model the transformation from *point A* to *point B*.

In this respect, Eq. (4.27) has been rewritten, depending on the conditions of that specific transformation.

$$T_A \cdot P_A^{\frac{1-\gamma_A}{\gamma_A}} = T_{B_{is}} \cdot P_B^{\frac{1-\gamma_B}{\gamma_B}} \quad (4.37)$$

Where T_A , P_A and γ_A are the dimensions at *point A*; P_B and γ_B are the same dimensions at *point B* and $T_{B_{is}}$ is the temperature of the isentropic transformation at *point B*. Thus, the latter temperature at *point B* can be obtained.

$$T_{B_{is}} = \frac{T_A \cdot P_A^{\frac{1-\gamma_A}{\gamma_A}}}{P_B^{\frac{1-\gamma_B}{\gamma_B}}} \quad (4.38)$$

The ratio of specific heat coefficient $\gamma = \frac{c_P}{c_V}$ has been obtained with the implementation of Eq. (4.6) and Eq. (4.7) of the research [9]. Both the specific heat coefficients are expressed as a function of temperature and density of that point. In this respect, γ_B has to be calculated as a function of $T_{B_{is}}$, therefore an iterative calculation has to be implemented to obtain it.

After calculating the convergence value of $T_{B_{is}}$, it has been used to obtain the respective enthalpy with Eq. (4.5). Then the isentropic variation of enthalpy has been obtained.

$$\Delta H_{is} = H_{B_{is}} - H_A \quad (4.39)$$

Thereafter, dividing the latter value to the isentropic efficiency, the real variation of enthalpy has been calculated and, with it, the real enthalpy.

$$\Delta H_{real} = \frac{\Delta H_{is}}{\eta_{is}} \implies H_B = \Delta H_{real} + H_A \quad (4.40)$$

It can be seen that the computational cost of the latter iterative calculation is very

high since all these calculations have to be carried out with a *Matlab Function* nested in the other two nested loop of the model (P_C and T_A). In detail the algebraic loop for the $T_{B_{i,s}}$ is nested inside the algebraic loop for the P_C and in turn the latter is nested inside the algebraic loop for the T_A : the simulation is really much longer.

4.3 Moisture modeling

Finally, before connecting the Vapor Cycle Subsystem model to the completed ECS system, it is necessary to calculate the air temperature that enter the cockpit, considering the humid air conditions, too.

In fact, the calculations described in Pragraph 4.2 show the model of the VCS on the refrigerant side. Therefore, the performance of the VCS on the air side has to be implemented in order to describe the refrigeration of the cockpit. In detail, it is important to analyze the heat exchange between refrigerant and air in the cycle, determining the air temperature entering the cockpit.

For this purpose, Eq. (4.41) has been used and the moisture has to be introduced in it.

$$H_{out} = H_{in} - \frac{\dot{Q}}{\dot{m}_{air}} \quad (4.41)$$

Where H_{out} and H_{in} are respectively the outlet air and the inlet air enthalpies, \dot{Q} is the heat exchange between refrigerant and air and \dot{m}_{air} is the air mass flow rate that, after refrigerating, enter the cockpit.

The moisture can be seen as a mixture of dry air and water vapor. In detail, dry air is a mixture of all the gasses in the air, except water vapor. Therefore, the enthalpy can be written as follows.

$$H = H_{da} + H_v = m_{da} \cdot h_{da} + m_v \cdot h_v \quad (4.42)$$

Where h_{da} and h_v are the specific enthalpies [J/kg] of, respectively, dry air and

water vapor; and m_{da} and m_v are their masses.

Generally, the enthalpy H is evaluated considering the flow of dry air, because it remains constant despite the changes in the amount of water vapor, in order to simplify the calculations for the thermodynamics of humid air.

$$H = m_{da} \cdot h \quad (4.43)$$

Therefore, combining Eq. (4.42) and Eq. (4.43), Eq. (4.44) can be obtained.

$$m_{da} \cdot h = m_{da} \cdot h_{da} + m_v \cdot h_v \quad (4.44)$$

Dividing Eq. (4.44) by the dry air mass, the specific enthalpy has been obtained.

$$h = h_{da} + \frac{m_v}{m_{da}} \cdot h_v \quad (4.45)$$

As you can see, in Eq. (4.45) there is the expression of quality (or absolute humidity) defined as a ratio between the mass of water vapor and the mass of dry air.

$$x \equiv \frac{m_v}{m_{da}} \quad (4.46)$$

Moreover, the specific enthalpy can be expressed as a function of the temperature. Therefore, after combining the latter equations, the specific enthalpy can be rewritten, obtaining Eq. (4.47).

$$h = c_{p_{da}} T + x \cdot (c_{p_v} T + r) \quad (4.47)$$

Where:

- $c_{p_{da}}$ is the specific heat at constant pressure of dry air and it depends on the temperature, but generally in the temperature range of $263K - 313K$, its average value is about $c_{p_{da}} = 1,005 \frac{kJ}{kgK}$

- c_{p_v} is the specific heat at constant pressure of water vapor and it depends on the temperature, but generally in the temperature range of $263K - 313K$, its average value is about $c_{p_v} = 1,875 \frac{kJ}{kgK}$
- r is the evaporation latent heat of water at $273 K$ and generally it is $r = 2501 \frac{kJ}{kg}$
- x is the quality, defined as in Eq. (4.46)

Therefore, Eq. (4.48) can be obtained, by explaining the terms of Eq. (4.41) with the previous mathematical relations.

$$(c_{p_{da}} T_{out} + x_{out} (c_{p_v} T_{out} + r)) = (c_{p_{da}} T_{in} + x_{in} (c_{p_v} T_{in} + r)) - \frac{\dot{Q}}{\dot{m}_{air}} \quad (4.48)$$

Finally the equation can be rewritten obtaining the mathematical expression for the outlet air temperature

$$T_{out} = \frac{1}{c_{p_{da}} + x_{out} c_{p_v}} \cdot \left(c_{p_{da}} T_{in} + x_{in} (c_{p_v} T_{in} + r) - \frac{\dot{Q}}{\dot{m}_{air}} - x_{out} r \right) \quad (4.49)$$

The quality x in the previous equations can be expressed as a function of the partial dry air pressure and partial water vapor pressure. Then, these two partial pressure can be rewritten as a function of the saturated pressure and the relative humidity, too.

$$x = 0,622 \cdot \frac{p_v}{p_{da}} = 0,622 \cdot \frac{p_v}{p_{in} - p_v} = 0,622 \cdot \frac{\phi p_{v_sat}}{p_{in} - \phi p_{v_sat}} \quad (4.50)$$

Where:

- the coefficient 0,622 is obtained from the ratio $\frac{M_v}{M_{da}}$ for the standard air, therefore it is $\frac{M_v}{M_{da}} = \frac{\text{water vapor molar mass}}{\text{dry air molar mass}} = \frac{0,0180153 \frac{kg}{mol}}{0,028964 \frac{kg}{mol}} = 0,622$
- p_{v_sat} is the saturated water vapor pressure

- p_{in} is the inlet pressure of the Vapor Cycle Subsystem
- ϕ is the relative humidity and it is defined as $\phi = \frac{p_v}{p_{v_sat}}$. It can take values from $\phi = 0$ that is 0% of humid air, thus the air is dry air, to $\phi = 1$ that is 100% of humid air, thus the air is saturated

The values of the saturated water vapor pressure of the Mollier Diagram are generally summarized in tables. Nevertheless, an approximation of those tables has been preferred in order to simplify the modeling (the tables are generally complex and quite long) [14]. The approximation can be obtained from the following equation.

$$\log(p_{v_sat}) = 28,59051 - 8.2 \cdot \log(T) + 0,0024804 \cdot T - \frac{3142,31}{T} \quad (4.51)$$

Where the saturated water vapor pressure p_{v_sat} is evaluated in [bar] and the temperature in [K].

Moreover, in Eq. (4.49) the calculation of outlet air temperature T_{out} is complicated by the quality outlet conditions. In fact, the saturated water vapor pressure and, as a consequence, the quality for the outlet temperature, depends on the calculation of outlet air temperature.

Therefore, an algebraic loop is required for its Matlab-Simulink implementation. Thus we decided to calculate the outlet air temperature using Eq. (4.48) in which the left-hand side of the equation represents the outlet condition, and the right-hand side of the equation represents the initial condition. All the dimensions of the initial condition are known, thus the equation is known, too, and, for the sake of clarity, can be rewritten as follows:

$$H_{out} = H_{in} - \frac{\dot{Q}}{\dot{m}_{air}} = (c_{p_{da}}T_{in} + x_{in}(c_{p_v}T_{in} + r)) - \frac{\dot{Q}}{\dot{m}_{air}} \quad (4.52)$$

Then, an algebraic loop has been implemented in order to recalculate the outlet condition until a convergence to the initial condition is obtained.

First of all, a maximum temperature and a minimum one have been considered. These two values of temperature are the extreme values within the solution can cycle. In detail, the maximum temperature has been considered as the initial value of air temperature that enter the evaporator, because the aim of the VCS is to

reduce the cockpit temperature, therefore it cannot be higher than the initial one. On the other hand, the minimum temperature has been considered as the outlet temperature calculated in dry condition, obtained assuming $x_{in} = x_{out}$. This value for minimum temperature has been chosen because the moisture increases the outlet temperature, therefore it cannot be lower than the one in dry condition. Thus, the trying outlet temperature has been calculated by an average:

$$\left(T_{trying} = \frac{T_{max} + T_{min}}{2} \right) \quad (4.53)$$

After that, an algebraic loop for each of the two extreme temperature values has been introduced, in order to obtain the two new extreme temperature values for each simulation time step and to calculate with them the new outlet temperature. It is obtained in the Matlab-Simulink environment with a *Unit Delay* block for the two extreme temperature T_{max} and T_{min} .

Moreover, as anticipated, the relative humidity could not be greater than $\phi = 1$. If it is, which means that the partial pressure of water vapor is greater than the saturated water vapor pressure at the corresponding temperature T , the water vapor condensates until an equilibrium corresponding to that saturated water vapor pressure is reached.

The possibility of condensate formation has to be considered, too. In this respect, the quality in maximum relative humidity condition ($\phi = 1$) has been evaluated for the trying outlet temperature. Then, using the minimum value between the latter quality and the initial quality, the trying outlet enthalpy has been calculated and compared with the one obtained for the initial condition (Eq. (4.52)). Therefore, after the comparison, the extreme temperature values of the algebraic loop for the next simulation time step, can be:

- $T_{max_{t+1}} = T_{max_t}$ and $T_{min_{t+1}} = T_{trying}$ if the trying outlet enthalpy is greater than the initial condition: there is the condensate
- $T_{max_{t+1}} = T_{trying}$ and $T_{min_{t+1}} = T_{min_t}$ if the trying outlet enthalpy is lower than the initial condition: there is not the condensate

Therefore, the convergence outlet air temperature obtained by the algebraic loop can be introduced directly into the cockpit, refrigerating it.

Chapter 5

Complete ECS model and results

After the Bleed Subsystem and the Vapor Cycle Subsystem modeling, they have to be connected.

Before we do that, a dynamic model of the cockpit and the implementation of a control law to determine the correct opening of the Temperature Control By-Pass Valve are required.

Therefore, the aim of this chapter is to provide a description of these two additional model and of the interaction of them with the models of the two subsystems.

Finally, after connecting all the models, obtaining the complet model, the results of the Environmental Control System will be provided.

5.1 Cockpit model

As seen in the previous chapters, the analysed ECS system has been modeled splitting the system in the two main subsystems (Bleed Subsystem and Vapor Cycle Subsystem). Both these models are able to simulate the behavior of the system in static conditions.

Therefore, to obtain a dynamic model of the system, a dynamic model of the

cockpit is required.

The dynamic equation for the air volume of the cockpit is described in Eq. (5.1).

$$\begin{bmatrix} V \left(\frac{\partial \rho}{\partial P} \right)_T & V \left(\frac{\partial \rho}{\partial T} \right)_P \\ m \left(\frac{\partial h}{\partial P} \right)_T - V & m \left(\frac{\partial h}{\partial T} \right)_P \end{bmatrix} \begin{Bmatrix} \left(\frac{\partial P}{\partial t} \right) \\ \left(\frac{\partial T}{\partial t} \right) \end{Bmatrix} = \begin{Bmatrix} \sum_i \frac{\partial m_i}{dt} - \rho \frac{dV}{dt} \\ \sum_i \frac{\partial m_i}{dt} h_i - \sum_i \frac{\partial m_i}{dt} h + \delta Q \end{Bmatrix} \quad (5.1)$$

The following assumptions can be made:

- $\frac{\partial P}{\partial t} = 0$
- $\dot{m}_{in} = \dot{m}_{out}$
- $\frac{dV}{dt} = 0$
- Air is an ideal gas $\frac{\partial h}{\partial T}_P = c_P$
- $\frac{\partial m_i}{dt}$ is the flow rate at port i (inlet/outlet)

Therefore, considering the latter assumptions, the system in Eq. (5.1) can be rewritten obtaining Eq. (5.2).

$$c_P m \frac{dT}{dt} = c_P \frac{dm}{dt} (T_{in} - T_{out}) + \delta Q \quad (5.2)$$

Where the c_P is the specific heat capacity at constant pressure and δQ is the heat flow rate.

Moreover, Eq. (5.2) can be rewritten in order to include both the Bleed Subsystem and the Vapor Cycle Subsystem contribution.

$$c_P m \frac{dT}{dt} = c_P \frac{dm_{bleed}}{dt} (T_{in_{bleed}} - T_{out}) + c_P \frac{dm_{VCS}}{dt} (T_{in_{VCS}} - T_{out}) + \delta Q \quad (5.3)$$

The heat flow rate has been calculated as follows:

$$\delta Q = \frac{dQ}{dt} = (UA)_{OAT} (OAT - T_{ckp}) + (UA)_{Fr} (T_{Frame} - T_{ckp}) \quad (5.4)$$

Where:

- UA is the global heat transfer coefficient expressed as a multiplication between the overall conductance U and the surface area A where the heat transfer takes place
- OAT is the outside air temperature
- T_{ckp} is the cockpit temperature
- T_{Frame} is the overall frame temperature

Therefore, the cockpit has been simulated as described in Figure 5.1.

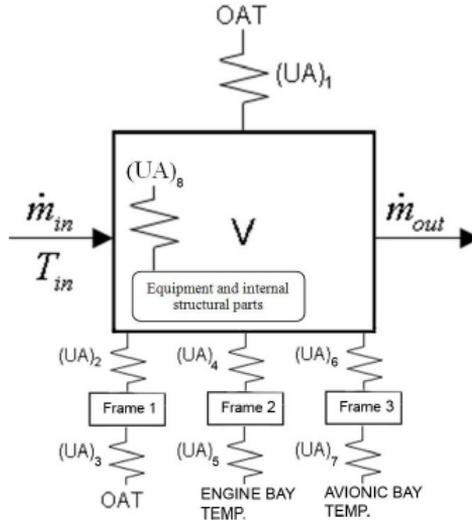


Figure 5.1. Simplified mathematical model of the aircraft cockpit

Figure 5.1 shows the situation described in Eq. (5.4) with a detailed explanation of the term T_{Frame} . In fact, it can be noted that, the cockpit air volume exchanges the heat with the outside air temperature ($(UA)_1$), with the equipment and the internal structural parts ($(UA)_8$) and with three frames, that interface with as follows:

- *Frame 1* is the external structure of the aircraft fuselage, therefore it determines a heat exchange between the cockpit air volume and this structure

$((UA)_2)$, and a heat exchange between the structure and the outside air temperature $((UA)_3)$

- *Frame 2* is the structure of the engine bay, therefore it determines a heat exchange between the cockpit air volume and this structure $((UA)_4)$, and a heat exchange between the structure and the engine bay temperature $((UA)_5)$
- *Frame 3* is the structure of the avionic bay, therefore it determines a heat exchange between the cockpit air volume and this structure $((UA)_6)$, and a heat exchange between the structure and the avionic bay temperature $((UA)_7)$

Explaining all these terms, Eq. (5.4) can be rewritten:

$$\begin{aligned} \delta Q = & (UA)_1 (OAT - T_{ckp}) + (UA)_2 (T_{Frame1} - T_{ckp}) + (UA)_4 (T_{Frame2} - T_{ckp}) \\ & + (UA)_6 (T_{Frame3} - T_{ckp}) + (UA)_8 (T_{internal\ parts} - T_{ckp}) \end{aligned} \quad (5.5)$$

Therefore, Eq. (5.5) has been introduced in the cockpit model in the Matlab-Simulink environment using elementary block and the different temperatures of that equation are obtained with algebraic loop by an implementation of the heat exchange with the specific temperature.

In detail, Eq. (5.6) is the equation for the implementation of the algebraic loop related to the *Frame 1*.

$$c\ m_{Frame1} \frac{dT_{Frame1}}{dt} = (UA)_3 (OAT - T_{Frame1}) + (UA)_2 (T_{ckp} - T_{Frame1}) \quad (5.6)$$

Eq. (5.7) is the equation for the implementation of the algebraic loop related to the *Frame 2*.

$$c\ m_{Frame2} \frac{dT_{Frame2}}{dt} = (UA)_5 (T_{ENG.BAY} - T_{Frame2}) + (UA)_4 (T_{ckp} - T_{Frame2}) \quad (5.7)$$

Eq. (5.8) is the equation for the implementation of the algebraic loop related to the *Frame 3*.

$$c m_{Frame3} \frac{dT_{Frame3}}{dt} = (UA)_7 (T_{AV.BAY} - T_{Frame3}) + (UA)_6 (T_{ckp} - T_{Frame3}) \quad (5.8)$$

Finally, Eq. (5.9) is the equation for the implementation of the algebraic loop related to the *Equipment and internal structural parts*.

$$c m_{internal\ parts} \frac{dT_{internal\ parts}}{dt} = (UA)_8 (T_{ckp} - T_{internal\ parts}) \quad (5.9)$$

In the previous equations, c is the specific heat capacity of the alluminium, that is the material of all the frames considered.

Moreover, for all the global heat transfer coefficients, two different kind of values have been considered to represent both the two operative conditions: cooling heating operations and ground heating operations.

5.2 Control law

Finally, the Temperature Control Subsystem has been modeled.

As described in Paragraph 2.1.2, it is a simple and totally analogue controller based on few logic conditions and its main aim is to control and regulate the Bleed air temperature entering the cockpit. Thus, it consists of a control law that regulates α , which is the opening angle of the throttle of the Temperature Control By-Pass Valve.

This control law provides temperature detection realized primarily by a comparison between the *effective temperature* and the temperature selected by pilot, where the latter *effective temperature* can be obtained by the 75% of the cockpit temperature and the 25% of the bleed air temperature.

The control law logic can be simply summarized in the schematic in Figure 5.2.

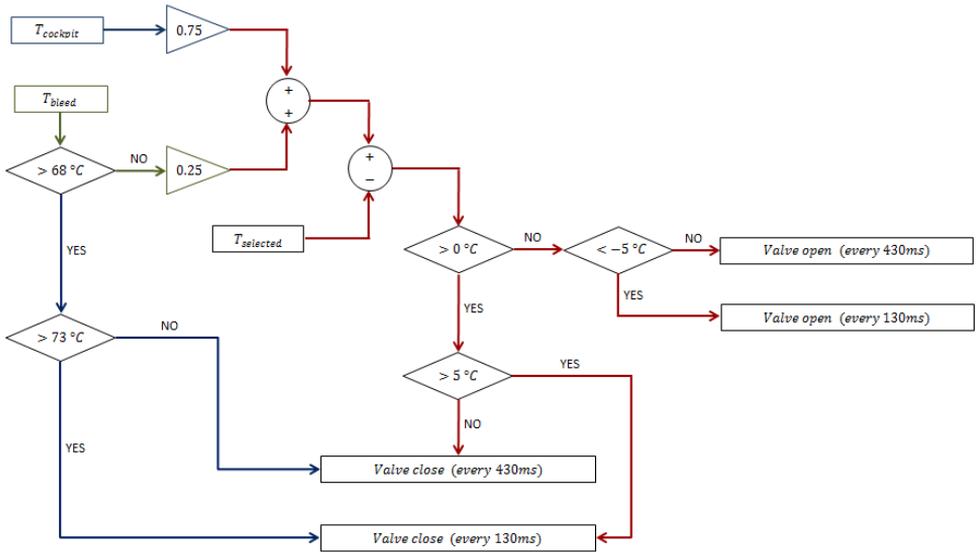


Figure 5.2. Control law schematic

First of all, the law controls if the bleed air temperature is higher than 68°C . If this logic condition is verified the blue line has to be followed and it means that the throttle has to be directly closed. Otherwise, if the condition is not verified, the red line has to be followed, opening or closing the throttle depending on the situations.

In detail, if the bleed temperature is lower than 68°C , the *effective temperature* is compared with the value selected by pilot. Thus, if the *effective temperature* is higher than the selected one, the control law commands to close the throttle; vice versa, if it is lower, the throttle is opened.

Moreover, the control law commands a pulse to open or to close the throttle for different simulation times:

- if the *effective temperature* is too far from the temperature selected by pilot, the throttle receives a pulse every 130 ms of the simulation with an amplitude of 22 ms, both for the opening or the closing
- if the *effective temperature* is quite similar to the selected one, the throttle

receives a pulse every 430 ms of the simulation with an amplitude of 22 ms

As a consequence, the control law is able to independently command the correct opening or closing speed of the throttle of the Temperature Control By-Pass Valve, in order to obtain a quicker (every 130 ms) or a slower (every 430 ms) variation of the cockpit temperature.

As far as concerns the modeling of this control law in the Matlab/Simulink environment, it has been modeled with elementary blocks to implement the logic conditions show in Figure 5.2.

Moreover, a mathematical equation has been implemented in the model to obtain the calculation of the α angle.

$$\alpha_{t+1} = \alpha_t - x \cdot \Delta\alpha + \bar{x} \cdot \Delta\alpha \quad (5.10)$$

Where α_{t+1} is the α angle calculated for a generic simulation time step and α_t is the angle calculate at the previous simulation time step. x and \bar{x} represent the logic signals in output from the logic operations. In other words x is the signal to close the throttle and \bar{x} is the signal to open it.

Thus, an algebraic loop has been used to implement Eq. (5.10) in the Simulink model and it has been obtained using a *Unit Delay* block.

On the other hand, the variation of the α angle has been realized with a *Pulse Generator* block with different setting parameters to obtain the two different speed of the throttle.

5.3 Results

After modeling all the components, the system has been assembled.

First of all, the Bleed Subsystem model and the cockpit model have been connected: the COV outlet temperature and mass flow rate have been used as inputs for the cockpit model in order to calculate the outlet cockpit temperature for each time step.

Thereafter, the control law has been connected, too. In fact, depending on the

difference between the calculated cockpit temperature and the one selected by pilot, the algebraic loop implemented by the control law commands the opening or closing of the By-Pass valve.

Finally, the VCS Subsystem has been connected to the cockpit model, too. In so doing, both contributions for the heating and the refrigeration are included in the complete system.

Moreover, a further change has been carried out to the complete system. Referring to the schematic in Figure 1.2 in Para. 2.1, the cockpit air temperature enters the evaporator fan before the heat exchange in the VCS evaporator. The presence of the Evaporator Fan determines that there is a further heat absorbed and, as a consequence, the air temperature becomes higher before the VCS evaporator. Therefore, this temperature contribution has been added to the simulation model. Since the model is now complete, some simulations can be performed.

Before running a simulation, the Simulink simulation parameters have been configured.

First of all, as simulation solver, a *Variable-Step Solver* has been chosen: this kind of solver changes the step size during the simulation in order to increase accuracy when a model's states change rapidly and, on the other hand, to avoid taking unnecessary steps when the model's states change slowly. Therefore, since the examined model is quite complex, this kind of solver has been preferred.

Moreover, the *ode45 (Dormand-Prince)* has been chosen. The latter solver computes the model's state at the next time step using an explicit Runge-Kutta (4,5) numerical integration. *Ode45* is a one-step solver, therefore it only needs the solution at the preceding time point. This solver has been preferred because it is the *Variable-Step Solver* with the lower computational cost, but it is accurate enough to well approximate the system.

After selecting the simulation parameters, the simulations has been performed.

It has been decided to simulate the two extreme conditions on board the airplane, therefore the maximum heating and maximum refrigeration conditions has been simulated. The results of those simulations have been compared to the respective results of the tests carried out on the company system test rig.

5.3.1 Heating

The first situation analyzed is the maximum heating.

Both the simulation model and the test rig are configured with the same initial parameters. The cockpit is in a very cold ambient: it is at an initial temperature of -30°C and the temperature selected by the pilot is 20°C , therefore the cockpit temperature has to be incremented by the ECS system in order to tend to the selected one.

A simulation of 30 minutes has been performed.

Thereafter, all the required simulation parameters have been implemented and, in detail, the VCS Subsystem has been set in *OFF mode*. In fact, it is the maximum heating situation and it means that the cooling effect of the VCS is not required.

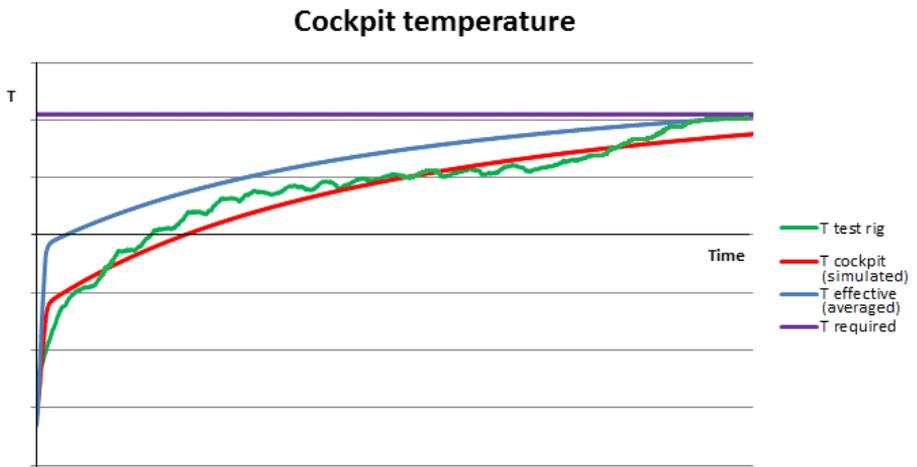


Figure 5.3. Comparison between the evolution of the cockpit temperature for the simulation model and the test rig in the heating condition

Figure 5.3 shows the evolution of the temperature in the maximum heating condition.

There are four different curves and the violet one is the temperature selected by the pilot, in fact all the other curves tend to it.

The red curve represents the evolution of the cockpit temperature obtained with

the simulation model. On the other hand, the green curve shows the evolution of the cockpit temperature measured on the test rig, that shows some oscillations due to the controller of the test rig.

It can be noted that the latter two curves have a similar evolution. Therefore, in both the curves there is a transient in which there is a significant increasing in temperature and, after that, both the temperatures increase but less than in transient.

Nevertheless, the red curve has the correct evolution, but it is generally few degrees less than the test rig temperature, in fact it doesn't achieve the required temperature. However, it is correct: as described in Para. 5.1, the temperature detection by the control law is provided on the *effective temperature* that, as previously defined, is obtained by the 75% of the cockpit temperature and the 25% of the bleed air temperature. It means that the control law shall ensure that the latter *effective temperature* achieves the selected temperature, but the real cockpit temperature will be always few degrees lower.

The *effective temperature* is represented by the blue curve in Figure 5.3. As previously described, the gap between the latter curve and the calculated cockpit temperature for each time step is shown. Actually, this temperature evolution shows the same performance of the real ECS on board the airplane, therefore it represents a limit for the control law and the model simulates exactly the control law performances.

It can be also noted that, as expected, the *effective temperature* approximates well enough the measured test rig curve, achieving the selected temperature. Figure 5.4 shows the opening angle α of the FMRSOV valve and the calculated cockpit temperature. It can be possible to note that the transient in the cockpit temperature is due to a progressive opening of the valve, that means a greater increasing of the temperature.

However, when the valve is fully open, the transient ends and temperature increases but less than the transient, in fact the curve slope is lower.

Moreover, the situation analyzed requires a considerable increasing in cockpit temperature since the initial one is very low. Therefore, the system has to constantly increase the temperature: after the transient, as expected, the valve remains fully open during all the simulation, since the required temperature is still not achieved

at the end of the simulation.

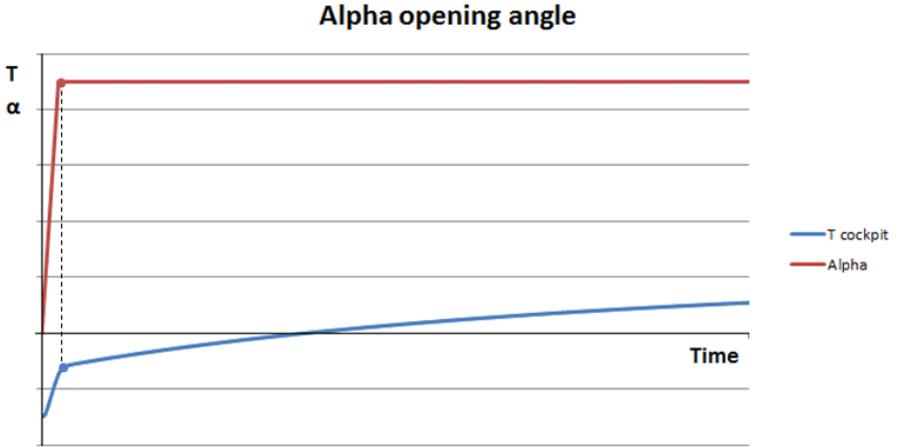


Figure 5.4. Alpha opening angle compared to the cockpit temperature in the heating condition

The bleed mass flow rate has been compared, too.

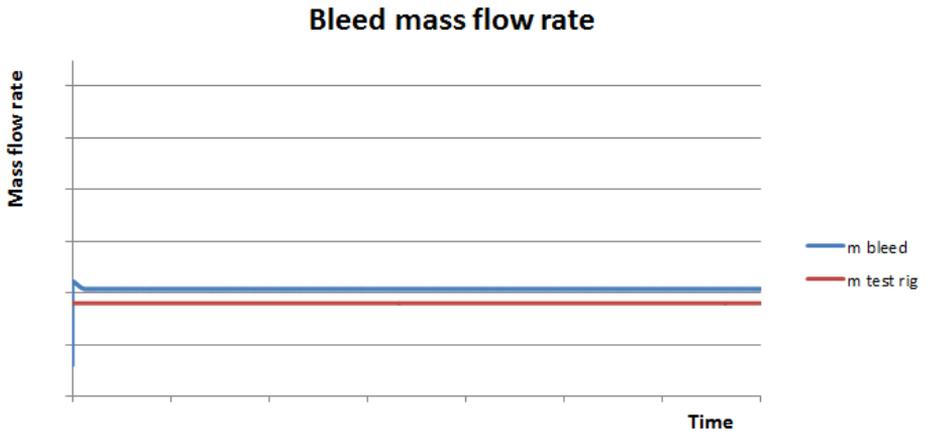


Figure 5.5. Comparison between the evolution of the bleed mass flow rates for the simulation model and the test rig in the heating condition

As expected, the evolution of the simulated bleed mass flow rate shown in Figure

5.5 approximates quite well the respective performance of the test rig. In fact, despite of a little gap on the flow rate value, the evolution of the two curves is the same: both the simulated and the test rig evolutions remain quite constant during all the test.

It is also possible to note a little variation in the first few seconds of the simulation on the simulated curve. It is consistent with the evolution of the opening angle α shown in Figure 5.4. In fact, in the first few seconds of the simulation the throttle of the valve is opening and it determines a variation of the bleed mass flow rate; on the other hand, when the valve is fully open, the convergence of the flow rate is achieved, too. Therefore, it remains constant after the transient.

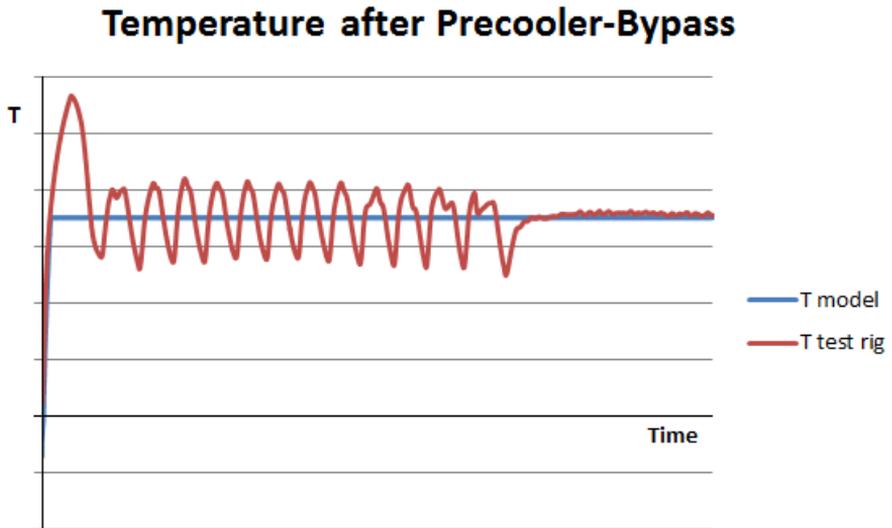


Figure 5.6. Comparison between the evolution of the Precooler-Bypass outlet temperature for the simulation model and the test rig in the heating condition

Finally, one last dimension has been compared, in order to verify the performances inside the Bleed Subsystem. Thus, the temperature after the Precooler - By-pass group has been compared.

As shown in Figure 5.6, the temperature curve of the simulation model well approximates the performances of the test rig. In fact, the convergence value is almost

the same.

However, it has to be noted that the temperature of the test rig oscillates about the convergence value. This particular evolution is due to the controller of the test rig. In fact, the trend line is almost comparable to the model results during all the simulation test.

5.3.2 Refrigeration

The second situation analyzed is the maximum refrigeration.

Both the simulation model and the test rig are configured with the same initial parameters. The cockpit is in a very hot ambient, in fact it is at an initial temperature of $70^{\circ}C$ and the temperature selected by the pilot is $20^{\circ}C$, therefore the cockpit temperature has to be reduced by the ECS system in order to tend to the selected one.

A simulation of 30 minutes has been performed.

All the required simulation parameters has been implemented. In detail, the Bleed Subsystem has been set in *OFF mode*. In fact, it is the maximum refrigeration condition and it means that the bleeding effect is not required.

The first dimension analyzed is the cockpit temperature, as shown in Figure 5.7.

As expected, both the cockpit temperature of the simulation model and on the test rig decrease with time, but, at the end of the simulation, only the test rig temperature converges to the one required by the pilot. The latter is represented in the plot by the green curve. In fact, the temperature resulting by the simulation model is still some degrees over the required one.

First of all, it should be noted that the transient is quite different in the two curves. The transient of the test rig curve shows a temperature decreasing much more gradually and progressively than the other one, in fact the slope of the test rig curve is lower then the other one. Moreover this gradual variation determines a greater reduction of the temperature, too.

Nevertheless, after the transient, the slope of both two temperature curves matches, in fact, referring to Figure 5.8, the error in the two measured slopes is only 0,6% (ΔT_2 and ΔT_3 differ of 0,6%).

Moreover, the transient is notoriously very difficult to model, especially in this

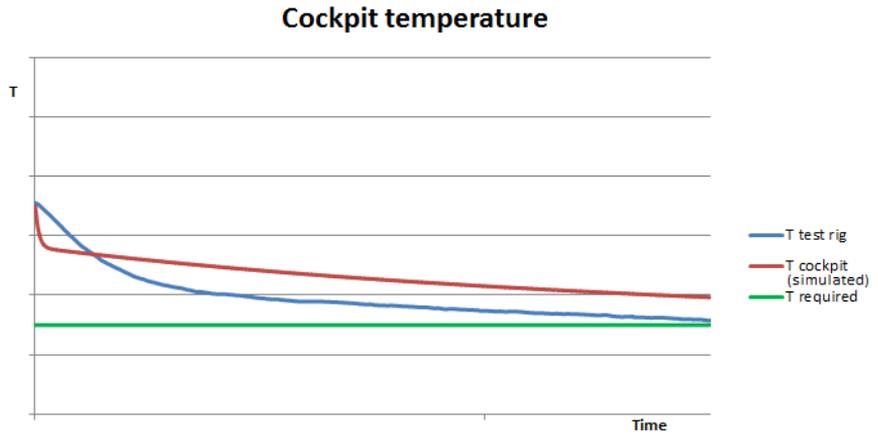


Figure 5.7. Comparison between the evolution of the cockpit temperature for the simulation model and the test rig in the refrigeration condition

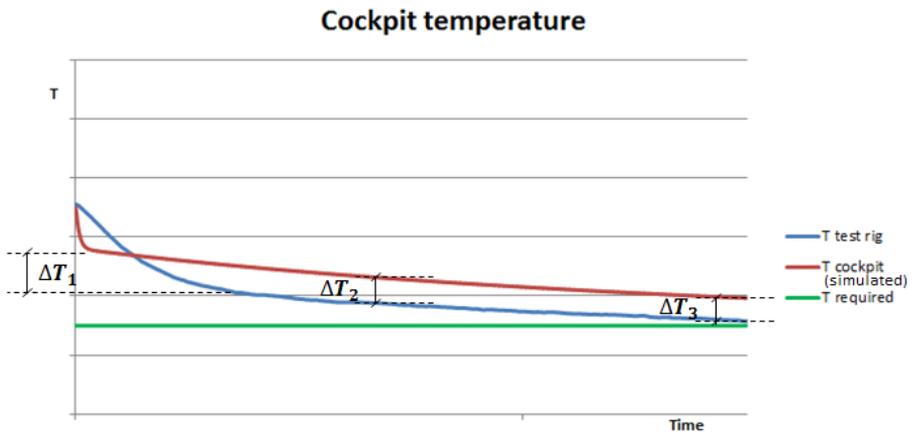


Figure 5.8. Temperature variation in the cockpit temperature plot in the refrigeration condition

situation because the control law does not regulate the VCS transient. In fact, as shown in Figure 5.4, the transient for the Bleed Subsystem exactly corresponds to

the opening of the valve, but it is not the same for the VCS Subsystem. However, overcome the transient, the curve performances are correct. The final temperature gap between the simulated and test rig curves is due to the incorrect transient, too.

In fact, as shown in Figure 5.8, the error on the temperature variation measured at the end of transient (ΔT_1), about in the middle (ΔT_2) is about 4,4%. It means that improving the transient modeling, probably the evolution during the transient could be more gradual, with a resultant more constant temperature decrease in this phase. As a consequence, all the rest of the curve could be few degrees lower, allowing a match with the test rig results.

Therefore, this result can be considered acceptable at this modeling level and the improvement of the transient modeling will be definitely a further development of the project.

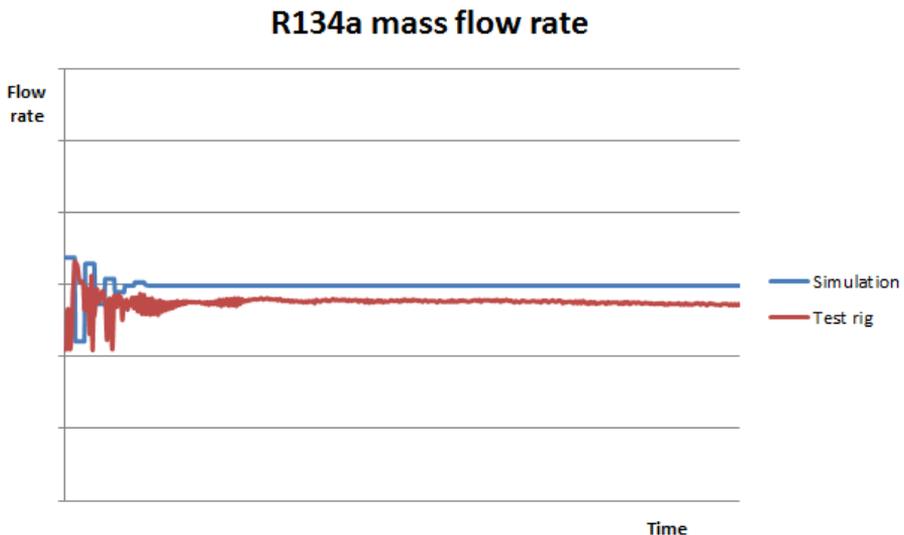


Figure 5.9. Comparison between the evolution of the refrigerant mass flow rate for the simulation model and the test rig in the refrigeration condition

Another dimension compared is the R134a refrigerant mass flow rate.

The evolution of this dimension supports what we said about temperature. In fact, the evolution of the curve is correct also for the mass flow rate, but there is a gap

on the convergence value. Here again, this gap is due to the transient modeling that determines a lower decreasing in temperature and, as a consequence, also the mass flow rate in the test rig is lower.

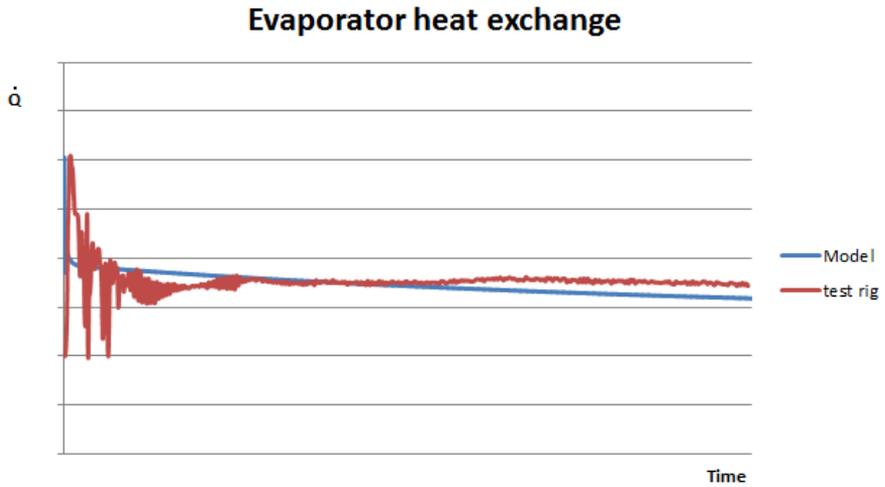


Figure 5.10. Evolution of evaporator heat exchange in the refrigeration condition

Figure 5.10 and Figure 5.11 show respectively the evolution of the evaporator and condenser heat exchange. Both these two plots show a comparable evolution between simulated and test rig curves.

It can also be noted that both the plots show oscillations about the convergence values for the test rig curve. As previously described, these oscillations are due to the controller of the test rig.

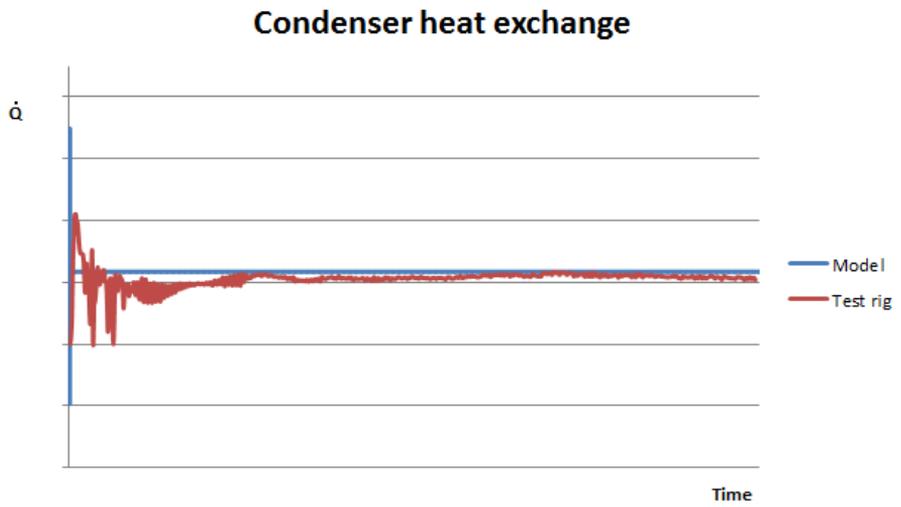


Figure 5.11. Evolution of condenser heat exchange in the refrigeration condition

Chapter 6

Conclusions

In this thesis project, a simulation model at high level as lumped parameter of the Environmental Control System of an existing aircraft has been developed in the Matlab/Simulink environment.

The purpose is to obtain a model capable to predict and analyze the system performances in different conditions in order to reduce simulation times and costs that would occur using a test rig. To achieve this goal, the simulations performed by the model have to be compared and validated with the results obtained by the test rig in the same analyzed conditions.

In this respect, two extreme condition tests have been performed both on the test rig and with the simulation model: maximum heating and maximum refrigeration. Moreover, since the tests have been performed in extreme conditions, it has been possible to analyze individually the modeling, the functioning and the performances of the two main subsystems: the VCS Subsystem has been set in *OFF mode* for the heating test and the Bleed Subsystem has been set in *OFF mode* for the refrigeration test.

As far as concerns the Bleed Subsystem analyzed with the heating test, the simulation model approximates reasonably well the test rig performances: the model shall be capable to provide a considerable increasing in cockpit temperature allowing it to tend to the one selected by the pilot. Moreover, the increasing in cockpit temperature is exactly proportional to the opening of the FMRSOV valve and it means

that the mass flow rate flowing through the Bleed line is correctly regulated. This is also confirmed by the congruence between the curves of the Precooler-Bypass outlet temperature respectively obtained by the simulation model and by the test rig which confirms the correct functioning of the model also inside the system.

On the other hand, the VCS Subsystem was more complex since the early stages of modeling. Two of its component (condenser and evaporator) include a phase change of the refrigerant whose modeling is notoriously very difficult: this has caused much longer modeling times. Furthermore, these difficulties have been reflected on the results, too. Nevertheless, the results obtained by the refrigeration test have been sufficiently satisfactory: as expected the model shall be capable to provide a considerable increasing in cockpit temperature allowing it to tend to the one selected by the pilot. However the evolution of the transient is too rapid and it means that the convergence temperature value is some degrees over the one required by pilot. Nevertheless, except for the transient, the evolution of the temperature in refrigeration condition is correct, in fact the refrigerant mass flow rate and the heat exchanges of evaporator and condenser show simulated evolutions comparable to the test rig.

As a conclusion, at this modeling stage the simulation model can be considered as validated: both heating and refrigeration tests have shown satisfying results.

As far as concerns the future perspectives of the work, some aspects could be detailed and improved. First of all, the VCS transient evolution could be definitely investigated in order to improve the modeling and, as a consequence, its performance.

Another important aspect could be the simulation time. At this stage, a simulation of 30 minutes runs for about 2 hours, therefore the actual model could be optimized in order to reduce this long simulation times.

Moreover, a further development may be to adapt and, then, to validate the model also for other similar systems or in other different functioning conditions.

Bibliography

- [1] I. Moir, A. Seabridge, *Aircraft Systems - Mechanical, electrical and avionics subsystems integration*, 3rd Edition, John Wiley & Sons, 2008
- [2] Y. A. Cengel, A. J. Ghajar, *Heat and Mass Transfer*, 5th Edition, McGraw Hill Education, 2015
- [3] A. Bejan, A. D. Kraus, *Heat Transfer Handbook*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003
- [4] M. Behbahani-Nejad, A. Bagheri, *A MATLAB Simulink Library for Transient Flow Simulation of Gas Networks*, World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol.2, No.7, 2008
- [5] J. Rutgers, *Dynamic modeling of a heat exchanger*, Internship Report at Powerspex, Hengelo, 2016
- [6] P. Nevřiva, S. Ozana, L. Vilimec, *Simulation of the Heat Exchangers Dynamics in MATLAB & Simulink*, Department of Measurement and Control, Department of Energy Engineering, VŠB - Technical University of Ostrava, 2009
- [7] L. Costiuc, V. Popa, *Simulink Model for a Heat Exchanger*, The 3rd International Conference on “Computational Mechanics and Virtual Engineering”, Transilvania University of Brasov, n. 51, pp. 859-864, 2009
- [8] V. Mangubat, *Variable Displacement Compressor - How it works*, 2014
- [9] R. Tillner-Roth, H. D. Baehr, *An International Standard Formulation for the Thermodynamic Properties of 1,1,1,2-Tetrafluoroethane (HFC-134a) for Temperatures From 170 K to 455 K and Pressure up to 70 MPa*, Institut fuer Thermodynamik, Universitaet Hannover, Germany, Vol.23, No.5, pp. 657-720 1994

- [10] M. L. Huber, M. O. McLinden, *Thermodynamic Properties of R134a (1,1,1,2-Tetrafluoroethane)*, International Refrigeration and Air Conditioning Conference, Paper 184, pp. 453-462 1992
- [11] C. Lu, Y. Cheng, H. Liu, *An approach to fault detection and isolation for control components in the aircraft environmental control system*, 2013
- [12] P. Tronville, *Energetica dell'edificio - Course slides*, Department of Energy (DENERG), Politecnico di Torino 2014
- [13] M. J. Lampinen, *Thermodynamics of Humid Air*
- [14] M. J. Lampinen, *Termodynamiikan teoreettiset perusteet*, 1987
- [15] P. Maggiore, *Sistemi di bordo aerospaziali - Course slides*, Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, 2015
- [16] P. Maggiore, M. Dalla Vedova *Modellazione, simulazione e sperimentazione dei sistemi aerospaziali - Course slides*, Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, 2017
- [17] UTC Aerospace System, *Company Technical Document*, (Proprietary, not disclosed)