Politecnico di Torino

Department of Mechanical, Aerospace
and Automotive Engineering

Master of Science in Automotive Engineering

Master Thesis

Predictive control strategies for electric vehicles thermal management system

Supervisors
Prof. Daniela Anna Misul
Dr. Federico Miretti
Dr. Matteo Acquarone

Candidate
Michele Barale

Academic Year
2024-2025

Abstract

Optimization of the thermal management in an electric vehicle (EV) is a crucial challenge aimed to improve the overall energy efficiency and ensuring optimal operating conditions for both battery system and passengers comfort. The Battery Thermal Management (BTM) system and the Heating, Ventilation and Air Conditioning (HVAC) system are among the most energy-demanding systems in an electric vehicle, significantly impacting the total driving range.
Made these considerations, the thesis focuses on the development and implementation, in Matlab/Simulink environment, of an adaptive Model Predictive Control (MPC) strategy to enhance the energy efficiency of these systems while maintaining both battery and cabin temperatures within the desired limits.
The work is structured in five chapters.
The first one presents a deep analysis and reverse engineering of the Simscape model developed by MathWorks and used for simulation, providing a detailed description of components, specially the ones directly affected by the developed AMPC controller, having then clear the reference model used for evaluating the performances of the different control strategies.
The second chapter introduces the initial control logic, which is based on reactive and PID strategies, and lays the theoretical fundamentals for MPC.
In the third chapter is presented the hearth of this thesis work, describing first of all the two control configurations explored, one prioritizing the battery and the other the cabin temperature regulation, then analyzing relatives predictive models build and controls configuration. This dual-priority control allows the system to dynamically adjust control objectives based on current operating conditions.
In the fourth chapter is presented an extensive performance evaluation through the simulations results, in which is demonstrated that the proposed adaptive MPC strategy achieves a reduction in cooling system energy consumption of 28% on the UDDS test cycle and 18% on the WLTC cycle compared to baseline control logic, without compromising the thermal comfort or battery safety. This translates into an overall 2.7% reduction in total vehicle energy consumption under the tested condition of 26 °C environmental temperature, which paves the way for even greater percentage savings in more extreme environmental conditions, where the cooling system has a greater impact on the total consumption of the vehicle.
Finally, the fifth chapter discusses the conclusions and potential future developments, underlining how these simulations confirm that adaptive MPC offers a significant advantage in therms of energy efficiency thanks to the prediction of the system's states evolution allowing for the calculation of the optimal control sequence, which convert into the ability for an EV to cover greater mileage with the same battery capacity.

Index

List of Figures

List of Tables

Listings

Nomenclature

Variables

β_{nom} nominal compression ratio

$\Delta u_{min/max}$ minimum/maximum possible value for the MVs variation

$\dot{m}_{b,pump}$ battery pump mass flow rate

$\dot{m}_{blower}$ current blower mass flow rate

$\dot{m}_{coolant\_chiller\_in}$ coolant mass flow rate at the chiller inlet

$\dot{Q}_{cab,conv}$ heat exchange by convection between cabin and external environment

$\dot{Q}_{chiller}$ heat exchange in the chiller

$\dot{Q}_{evap}$ heat exchange in the evaporator

$\dot{Q}_{met}$ passengers metabolic heat generation

$\dot{Q}_{ptc}$ ptc heat generation

$\eta_{isentropic}$ compressor isentropic efficiency

η_{vol} nominal compression volumetric efficiency

ρ_{ε} weight on constraint violation

ε_k slack variable

a Joule losses corrective factor

c_{p,air_cabin} specific heat of the air in the cabin

c_{p,air_out_evap} specific heat of the air at the evaporator exit

c_{p,b_cell} specific heat of the battery cell

$c_{p,coolant}$ specific heat of the coolant

E_{BTM} BTM energy consumption

E_{comp} compressor energy consumption

i current prediction step

i_b total battery current

i_{b_BTM} battery current for the BTM

$i_{b\_traction}$ battery current for traction

j current variable

$J(z_k)$ cost function

$J_u(z_k)$ MVs contribution on the cost function

$J_y(z_k)$ output variables contribution on the cost function

$J_{\Delta u}(z_k)$ MVs variation contribution on the cost function

$J_\varepsilon(z_k)$ constraint violation contribution on the cost function

k current control interval

m_b battery mass

$m_{air,cabin}$ mass of the air in the cabin

m_{cell} mass of a single battery cell

n_{cell} total battery cells

p prediction horizon

P_{comp} current compressor power

P_{max_ptc} maximum ptc power

$p^{in}_{nom,comp}$ nominal compressor inlet pressure

pc control horizon

s^u scale factor for the MVs

s^y scale factor for the output variable

$s^{\Delta u}$ scale factor for the MVs variation

T_b battery temperature

T_c cabin temperature

T_s MPC sample time

$T_{air\_out\_evap}$ temperature of the air at the evaporator exit

$T_{b,target}$ target battery temperature

$T_{coolant\_chiller\_in}$ temperature of the coolant at the chiller inlet

$T_{coolant\_chiller\_out}$ temperature of the coolant at the chiller outlet

T_{env} environmental temperature

T_{in_evap} temperature of the air at the evaporator inlet

$T_{target,HVAC}$ target cabin temperature

$u_{min/max}$ minimum/maximum possible value for the MVs

V_{b_nom} nominal battery voltage

V_{min} constraint relaxation factor on the lower limit

V_{min} constraint relaxation factor on the upper limit

w^u weight on the MVs

w^y weight on the output variable

$w^{\Delta u}$ weight on the MVs variation

$y_{min/max}$ minimum/maximum possible value for the output variable

z_k QP optimal sequence

Acronyms

AMPC Adaptive Model Predictive Control

BTM Battery Thermal Management

BTMS Battery Thermal Management System

CAV Connected and Automated Vehicles

COP Coefficient Of Performance

CT Continuous Time

DP Dynamic Programming

DT Discrete Time

EM Electric Motor

EV Electric Vehicle

HVAC Heat, Ventilation and Air-Conditioning

ICE Internal Combustion Engine

LIB Lithium Ion Battery

LTI	Linear Time-Invariant
LTV	Linear Time Variant
MD	Measured Disturbance
MO	Measured Output
MPC	Model Predictive Control
MV	Manipulated Variable
OV	Output Variable
PID	Proportional-Integral-Derivative
PTC	Positive Temperature Coefficient
QP	Quadratic Programming
RMS	Root Mean Square
RMSE	Root Mean Square Error
SOC	State Of Charge
SS	State Space
UD	Unmeasured Disturbance
VCR	Vapor Compression Refrigeration

Introduction

In recent years, the automotive industry has changed drastically, starting to look for alternative solutions to traditional internal combustion engines (ICE). Driven by the need to reduce greenhouse gas emissions and the search for more sustainable energy solutions, electric vehicles (EVs) revealed to be one of the most appealing alternative at the point that governments and companies have invested heavily in developing advanced technologies to improve the efficiency, reliability, and range of EVs, which started to spread exponentially in the last few years arriving to represent in 2023 the 18% of the cars sold worldwide [6]. However, despite the progress, several technological challenges still need to be addressed to make the adoption of EV even more widespread and effective, such as the higher cost on average of a new EV compared to an ICE one, the lack of a widespread and reliable network of charging stations and the so called autonomy anxiety [7]. Two of the critical systems of the vehicle which impact more on the energy consumption are the battery thermal management system (BTMS) and the HVAC (Heating, Ventilation, and Air Conditioning) system. The temperature of the battery directly affects the performance, safety, and life of the energy storage system. Excessively high or low temperatures can significantly reduce battery efficiency, accelerate degradation, and compromise vehicle safety. At the same time, the HVAC system is essential to ensure passenger comfort, but its intensive use can negatively impact the vehicle's range.
In this context, this thesis work aims to develop and implement an advanced control system based on an adaptive Model Predictive Control (MPC) for the simultaneous management of battery cooling and HVAC system. This approach allows optimizing the use of available energy, balancing battery thermal needs and passenger comfort, taking into account operating conditions and future forecasts.
In the following chapters, details of the adopted methodology, implemented modeling and control strategies, and performances obtained through simulations and tests will be discussed. Before going into the merits of the MPC, an overview of the purposes and characteristics of the thermal management currently used in electric vehicles and the main challenges associated with their optimization will be provided.

0.1. BTMS and HVAC purpose and characteristics

The BTMS has the main objective of guaranteeing the correct cooling or heating of the battery pack, ensuring the constant operation of the cells in the optimal temperature conditions specified by the supplier. Generally, lithium-ion batteries (LIBs) are used, because of their high power and energy density.
These kinds of batteries must work on a specific window of temperature, between 5°C and 40/45°C, with an optimal operating temperature range that goes from 25

to 35 °C.

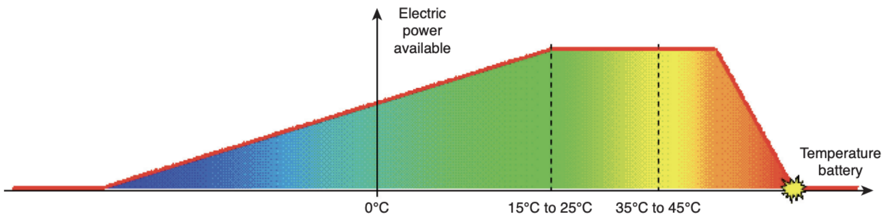

Figure 1: Battery optimal operating range [1]

The operation of batteries out of the optimal temperature range can cause different effects mainly related to the chemical reactions and to the material used, furthermore also the ionic conductivity of the electrodes are influenced by the temperature. Both high and low temperatures influence the battery performance and operating life, in particular:

- **High temperatures**: excessively high temperatures may cause the effects of aging and thermal runaway. The aging accelerates the deterioration of the battery capacity from the designed one which is a combination of cycle and calendar aging. While the thermal runaway is a phenomenon that may trigger exothermic reactions inside the battery, promoting for a further uncontrolled increase of the temperature. When this heat generation overlaps the thermal resistance of the battery, fires and in the worst case explosion may occur [8].

- **Low temperatures**: in general, a decrease in LIB performance is observed for temperatures below 0 ° C. Low temperatures affect the properties of the electrolyte, increasing its viscosity and reducing its ionic conductivity. These variations result in an increase in the internal resistance of the cell. Another typical phenomenon that occurs at low temperatures is lithium plating, in which metallic lithium forms around the anode during charging, causing a reduction in the battery capacity. Furthermore, metallic lithium exists in dendritic form, which can penetrate the separators and cause an internal short circuit [8].

Therefore the BTMS plays the crucial role of maintaining the battery in a safe temperature range, furthermore, in EVs, this system is integrated with the HVAC responsible for heating and cooling of the passenger compartment, this is done because the cooling of the battery provided by the radiator itself in hot summer environmental conditions is not sufficient, therefore and active system is needed to reach the cooling compliance. The cooling is regulated through a vapor compressor refrigeration (VCR) cycle, which usually use as refrigerant the R134a. The refrigerant absorbs heat from be battery coolant through the chiller and from the air entering the passengers compartment through the evaporator.

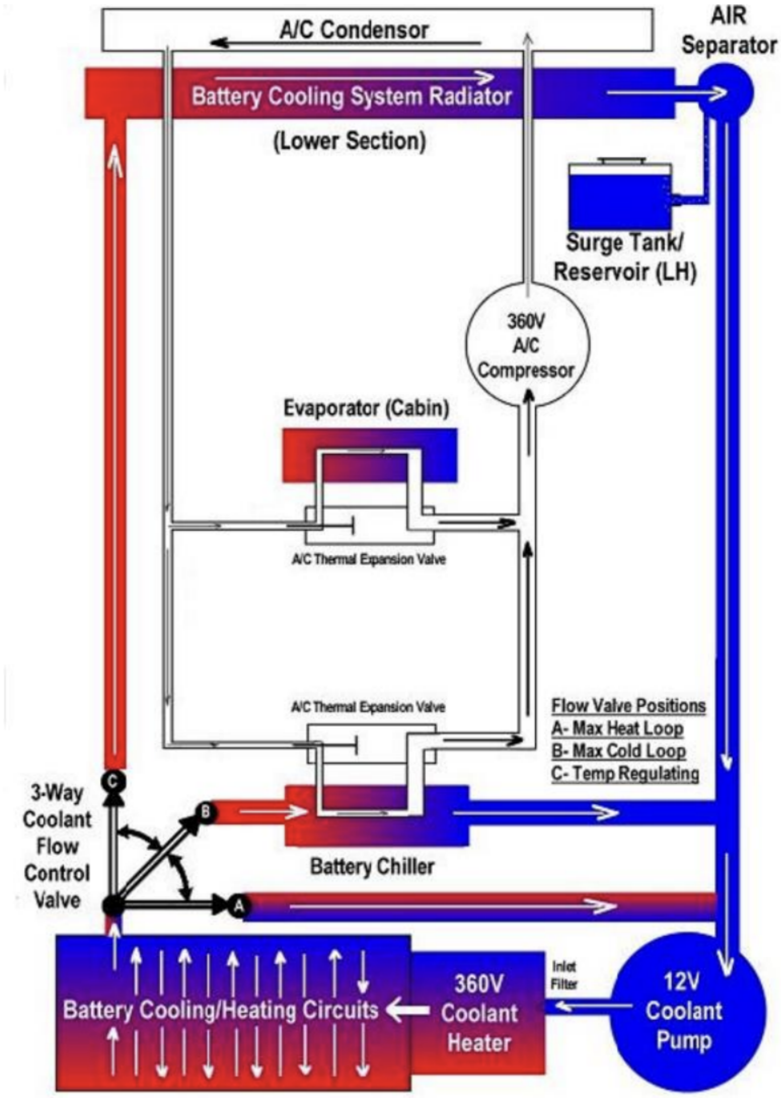

Figure 2: General BTM and HVAC scheme [2]

The coolant used is a mixture of deionized water and ethylene glycol, typically in a 50/50 ratio. The addition of ethylene glycol is essential for lowering the freezing point: with this proportion, the coolant remains fluid down to -36°C. Moreover, it plays a crucial role in preventing pipe corrosion. Due to these properties, this mixture is widely used in automotive cooling circuits. In hybrid vehicles, the use of deionized water is necessary not only to prevent corrosion issues but also to ensure proper high-voltage insulation. It is interesting to note that the addition of ethylene glycol significantly alters some key properties of the coolant compared to pure water. Specifically, the dynamic viscosity increases, leading to higher pressure losses within the pipes. The density is also higher, while the specific heat capacity is lower. At room temperature, the mixture's density is approximately 1.077 times that of pure water, while its specific heat capacity drops to 0.815 [9]. Consequently, to maintain the same heat transfer efficiency with identical inlet and outlet temperatures, the

volumetric flow rate must be increased by approximately 15%. A typical scheme of the these cooling systems for EVs is reported in figure 2. What was described above is the scenario in which a cooling effect is needed, on the counter when the environmental temperature are too low the heat produced by the motor is used to increase the battery temperature, but thanks to their high efficiency, electric motors (EM), may not produce enough heat at the point that in very cold conditions an additional coolant heater is necessary. Same for the cabin where the heat is provided by the positive temperature coefficient (PTC) heating element.
Together BTM and HVAC can impact up to 20% the total electric energy consumption [10][11], The integration and optimized control of these two systems is a crucial aspect in the design of modern electric vehicles, with the potential to increase significantly the range of the vehicle.

0.2. BTM and HVAC control strategies

Software strategies can significantly contribute in the optimization and consumption reduction of these two system. Is clear that the two main target of the aforementioned systems are: first to maintain the optimal temperature range for each component in every operating condition, in order to provide reliability, durability and comfort in case of the cabin; on the other hand this want to be achieved utilizing as little energy as possible. These two targets are opposite, therefore a good trade-off is needed.
Typical control strategies used are:

- On/off control: these type of control are called Reactive since they activate the cooling or heating only when the observed component overlapped the temperatures limits imposed. The robustness and simplicity of these type of controller are the main pros, but they are not able to perform any kind of energy optimization.
- PID control: regulate in a dynamic way the input of the different components of the BTM, proportionally to the error from the desired values. It is reliable and quite easy to be tuned, without required the detailed knowledge of the system's mathematical model. But in systems with slow response time, such as thermal one characterized by a large thermal inertia, they can not be optimal in the control.
- Predictive control: based on the deep knowledge of the system behavior, they are able to optimize the cooling components usage, predicting the future behavior of the system through an internal mathematical model of it, being able to act in advance.

This thesis focuses on the usage of the predictive control strategies, in particular the adaptive MPC (AMPC). From literature it was evident how in recent years more and more interest to optimal control strategies for BTM and HVAC are being investigated, thanks to the consumption potential it can be obtained from these

systems. It was studied how the connected and automated vehicles (CAV) can be exploited to optimize the cooling using the traffic flow information for predicting the future scenario by using the MPC [12]. Often a two stage optimization, one based on a long term high sampling time optimization using CAV data to predict the traffic condition, and a second stage with shorter prediction horizon and low sample time to accurate tracking the battery temperature based on actual vehicle speed profile; the optimization in these case was made using the Dynamic programming (DP) and the MPC [12][13][14]. Some researches where conducted studying only the HVAC system going to develop and extreme accurate model for the cabin prediction state without considering at the same time the BTM [15]. Other implementation of the only cabin control where conducted using a linear time variant (LTV) MPC [16].
To summarize it was difficult to find in literature some studies investigating the usage of the MPC for both the BTM and the HVAC systems, furthermore most of the prediction models results to be very precise but built using specific parameters of the system used. What is proposed to be done in this thesis is a prediction model able to control both the aforementioned systems, using as feedback variables value coming from the system's sensors and not tuned ad hoc for the specific system simulated; this is done to try to make the controller easily implementable on any cooling system having the same structure, without depending on the nature of the specific components.

Chapter 1

Model analysis

In this chapter will be introduced the Simscape model that simulate the cooling system of an EV. The system to be controlled, as mentioned, was developed by MathWorks in Simscape [17], which is a virtual environment that allows the development of physical systems in Simulink such as multi-domain systems such as BTMS.
In Figure 3 is shown the Simscape model used. Subsystems present here are:

- Scenario drive cycle
- Controls
- Measurements
- AMPC model

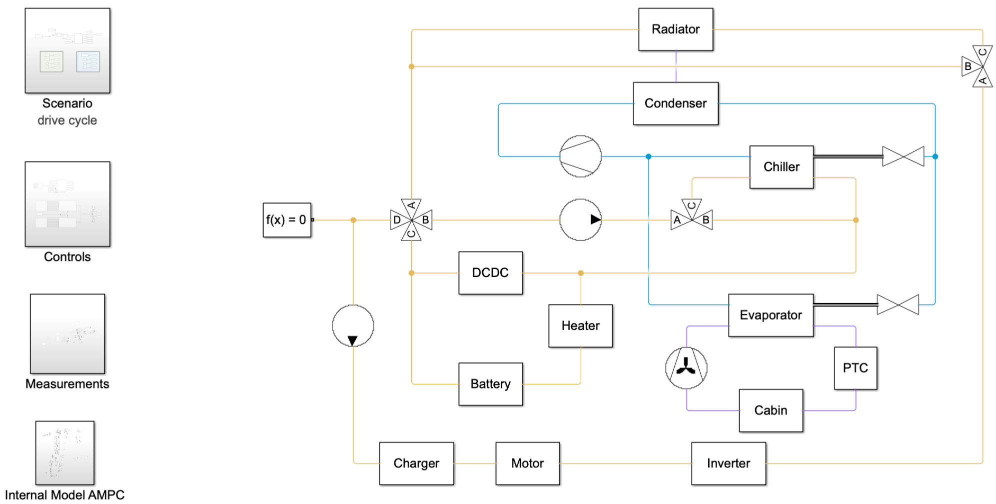

Figure 3: Simscape model

Scenario subsystem will be briefly described below, Controls will be deeply analyzed in Chapter 2, while measurements block is just used to extract and save in MATLAB data needed for the post processing of the simulation so it will not be further analyzed. About AMPC (Adaptive Model Predictive Control) model it is used to check the precision of the prediction model used in the control system described in Chapter 3 .

In the following Scenario subsystem will be analyzed, followed by the main valves of the system and its working modes and to finish a focus on vapor compression refrigeration (VCR) cycle's, battery cooling loop's and HVAC's components will be done. Motor's coolant loop components will not be analyzed since they are not purpose of this thesis.

1.1. Scenario subsystem

Scenario subsystem presented in figure 5 has three main purposes:

- Computing the powertrain current as well as the heat generated by motor, inverter, charger ad DC-DC converted: first of all the driving cycle to be simulated is selected then by using look up tables the values mentioned before are computed (figure 4).

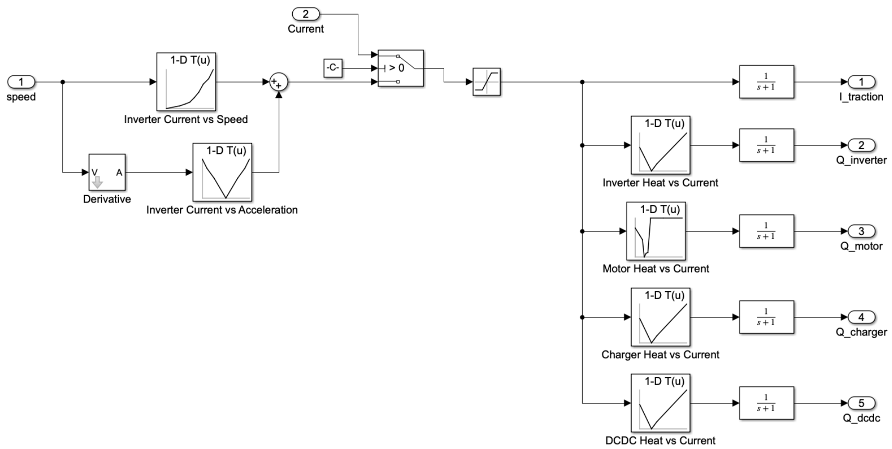

Figure 4: Powertrain demand

- Set environmental conditions: as pressure, temperature, relative humidity and CO_2 fraction.

- Set HVAC parameters: target temperature, AC on/off, cabin recirculation on/off and number of occupants which will be used to calculate heat generated in the cabin as will be showed below.

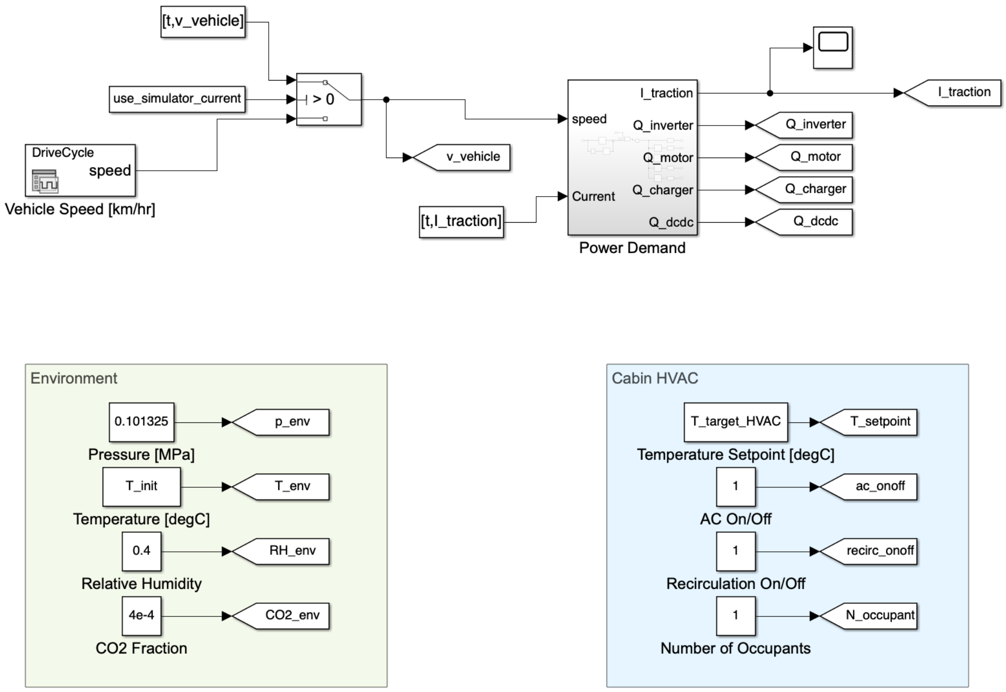

Figure 5: Scenario subsystem

1.2. Working modes and main valves

The original model from MathWorks (figure 6) presents different loops:

- In yellow are defined motor and battery coolant loops using as thermal liquid a mixture of water and ethylene glycol present at 0.5 in volume fraction
- In turquoise is identified the VCR loop with R-134a as refrigerant
- In purple the air loop of cabin's HVAC

Valves present are chiller and radiator bypass valve and four way valve, the latter is responsible for the operating mode of the system which will be described below.

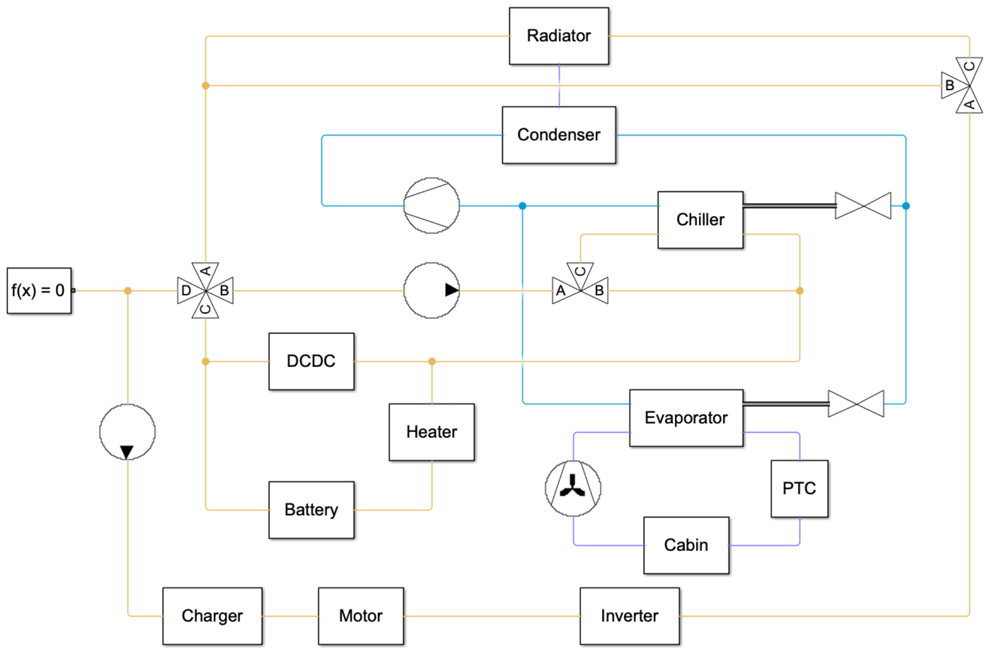

Figure 6: Simscape thermal loops

1.2.1 Four way valve

As mentioned above, the purpose of this valve is to select the operating mode of the system. In particular the two possible operating modes are the following:

- serial mode: in this case the thermal fluid enters from port A and exits from port B and consequently it enters from port C and exits from port D. This configuration can be used in cold weather conditions in which the heat from the motor is exploited to heat up the battery with the help of the heater in case temperatures are very low; in this condition both the chiller and the radiator are bypassed since no cooling effect is needed. This mode can still be used in warm weather, cooling the wall system trough the radiator supported by the chiller when necessary (detail of the control strategy will be analyzed in Chapter 2).

- parallel mode: in this case the thermal liquid enters from port A and exits from port D while it enters from port C and exits from port B. This configuration allows to keep separate the battery and the motor cooling circuit, it is particularly indicated in case of hot weather conditions to exploit maximum cooling capacity of the system, dedicating chiller for battery and radiator for motor cooling.

In detail is observable in figure 7 how this valve is controlled by the signal *cmd_parallel_serial*, it regulate the displacement of the valve between -0.0063 m (paralel) and 0.0063 m (serial); the control logic will be analyzed in Chapter 2. Finally the motor and battery tanks heat exchanges with the external environment are modeled with a convective heat transfer Simscape block.

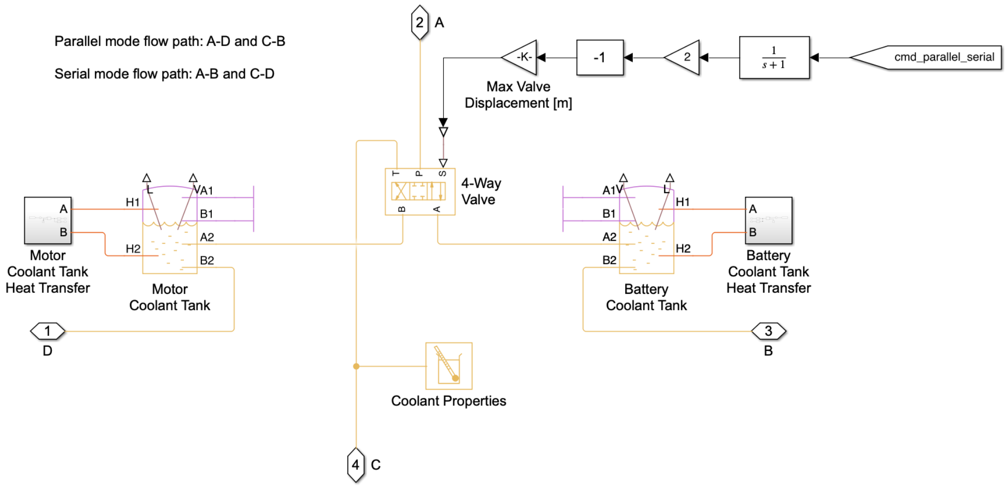

Figure 7: Four way valve detail

1.2.2 Chiller bypass valve

This valve has a similar working principle as the four way valve. Its position is regulated by *cmd_chiller_bypass* which switch between -0.0063 m (no bypass) and 0.0063 m (bypass). Note that *cmd_chiller_bypass* can be 0 or 1 then it is multiplied by two and summed to a bias of -1, so the final value will be -1 or 1 that multiplied by 0.0063 allow to find the valve displacement position.

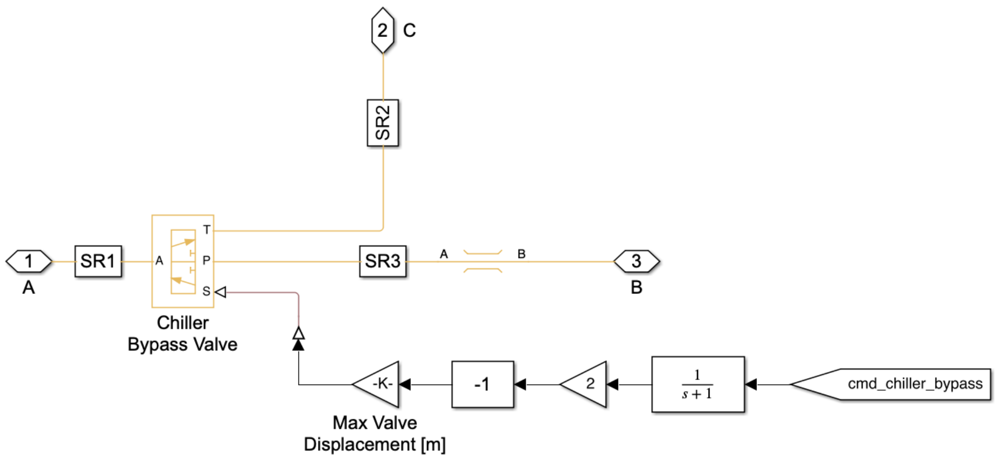

Figure 8: Chiller bypass valve

1.2.3 Radiator bypass valve

Here we have the same working principle described for the other valves. Valve position is switched between -0.0063 (no bypass) and 0.0063 (bypass). The switching logic is described in Chapter 2 and depends from the temperature of the coolant exiting the inverter coolant jacket.

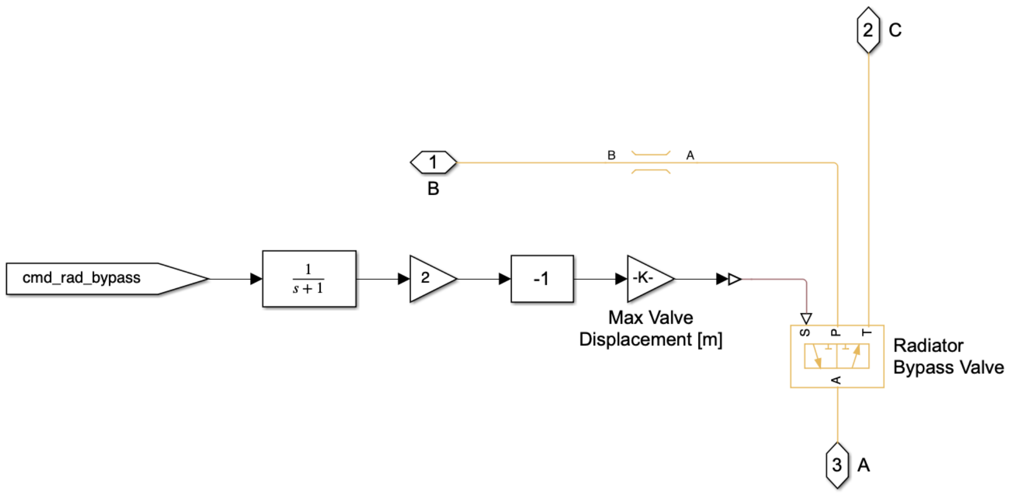

Figure 9: Radiator bypass valve

1.3. VCR components

The vapor compression refrigeration cycle represents the heart of the cooling system. An ideal case of the cycle's transformations is shown in figure 10. Processes are:

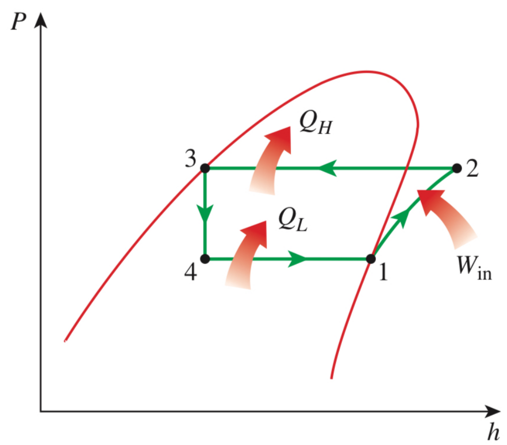

Figure 10: VCR ph diagram [3]

- 1 - 2 represent the isentropic compression made by the compressor: in reality the entropy of the refrigerant is not constant but can increase due to frictional effects and increase or decrease depending on the heat transfer direction[3].
- 2 - 3 represent the constant pressure heat rejection in the condenser: in reality some pressure drops are present in the condenser and in the pipe connecting it with the expansion. Furthermore it is difficult to ensure that the fluid exiting the condenser is exactly at saturated liquid phase, so what is done to avoid

refrigerant at gaseous state at the expansion valve inlet is to slightly subcooling it[3].

- 3 - 4 represent the isenthalpic expansion in the expansion valve: in reality the process is considerable isenthalpic but some entropy generation is present due to the viscous dissipation inside the fluid during the process[3][18].
- 4 - 1 represent the constant pressure heat absorption in evaporator and chiller: in reality it is very difficult to control the process so precisely so that the refrigerant enter the compressor as saturated vapor. Therefore chiller and evaporator are designed to ensure that the refrigerant is slightly superheated at the compressor inlet to avoid possible damage to it[3].

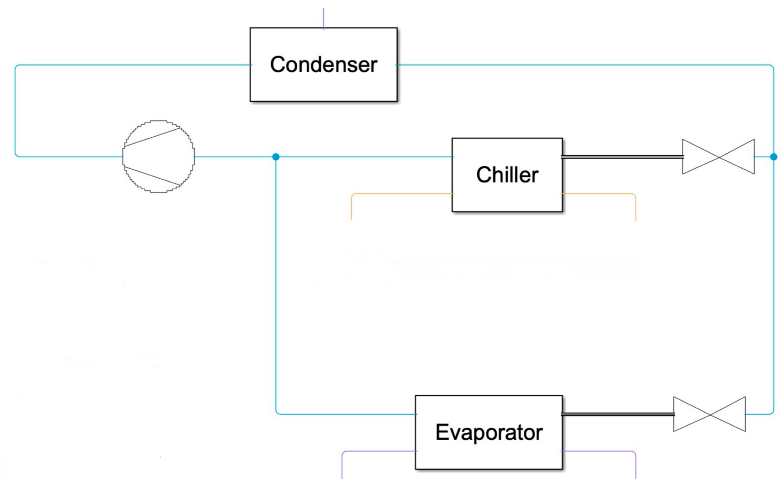

Figure 11: VCR Simscape

1.3.1 Compressor

The compressor is the component that allows to regulate the cooling power through the VCR cycle. It is regulated with the *cmd_comp*, which give as input the angular velocity. As shown in figure 12, the compressor is modeled as a positive displacement two-phase compressor (name of the Simscape block), which corresponds to a scroll compressor, a typical solution used in BTM system[19]. Some characteristics of the compressor are:

- compressor displacement: 80 cm^3/rev
- nominal compression ratio: $\beta_{nom} = 4.6$
- nominal inlet pressure: $p^{in}_{nom,comp} = 0.3MPa$
- isentropic efficiency: $\eta_{isentropic} = 0.65$
- volumetric efficiency: $\eta_{vol} = 0.9$

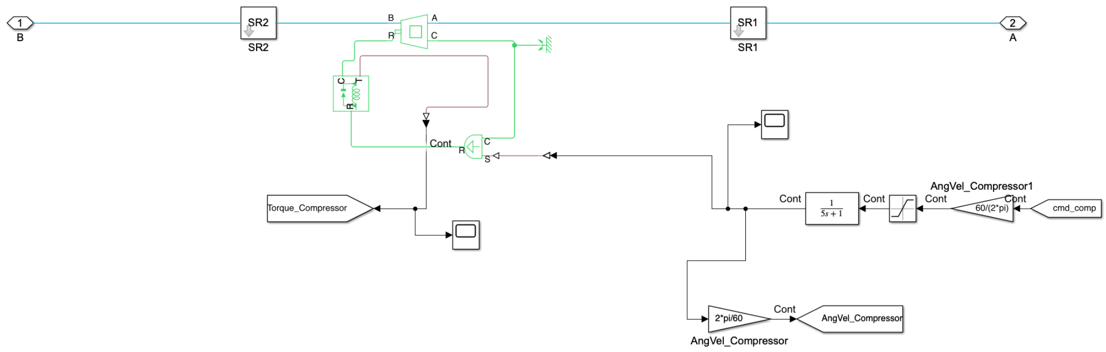

Figure 12: Compressor

Compressor is modeled in mechanical and thermal domain, electrical domain related to the electrical actuation of the compressor. This is done to avoid unnecessary computations during the simulation, also due to the fact that in the original model the only electrical load of the battery came from the motor. In this way the electrical power demand of the compressor is calculated multiplying its angular velocity by the actual torque (measured by the ideal torque sensor block) and then dividing it by 0.8 which is the conventional electromechanical efficiency.
This approach is used for all the part of the BTM.

1.3.2 Condenser

The condenser is the component of the VCR cycle responsible for the cooling of the refrigerant that comes from the compressor in gaseous state, allowing its condensation. It is modeled in Simscape as a cross-flow non-mixed fluids heat exchanger.The refrigerant is cooled exchanging heat with the environmental air, the heat exchange is directly proportional to the air mass flow rate which is regulated by the vehicle and fan speeds.

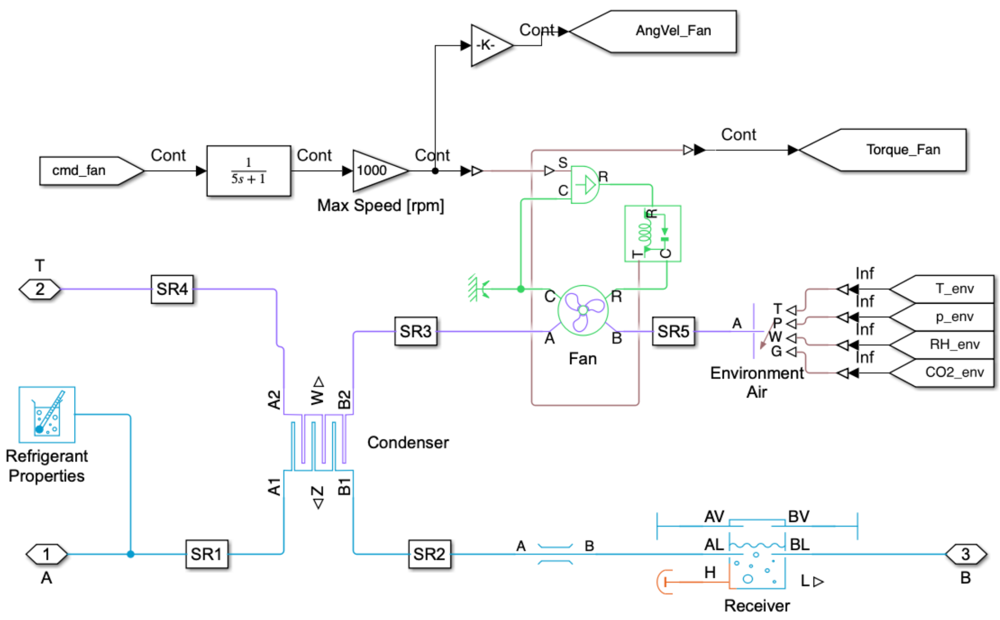

Figure 13: Condenser

1.3.3 Thermal expansion valves

This component receives the refrigerant from the condenser at saturated liquid state. The thermal expansion valve is responsible for the regulation of the refrigerant flow rate guaranteeing a certain amount of superheat at the exit of chiller and evaporator. At the exit of the latter is positioned the sensing bulb, a device that is filled with a liquid whose thermodynamic properties are similar to those of the refrigerant. This bulb is thermally connected to the output of the evaporator so that the temperature of the refrigerant that leaves the evaporator can be sensed. The gas pressure in the sensing bulb provides the force to open the valve, and as the temperature drops this force will decrease, therefore dynamically adjusting the flow of refrigerant into the chiller and evaporator. The spring is adjustable and the closing force generated regulates the superheat[20].

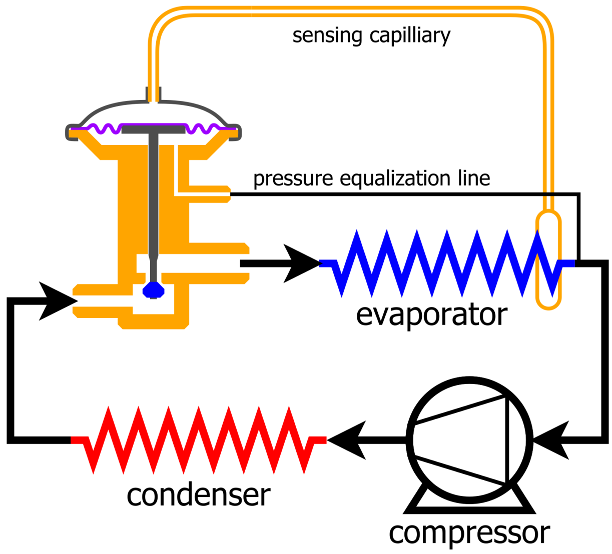

Figure 14: Thermal expansion valve scheme[4]

1.3.4 Chiller

Chiller represented in figure 15, as said previously has the duty to cool the coolant in the battery's loop. It is designed in Simscape as a cross-flow unmixed-fluid heat exchanger.

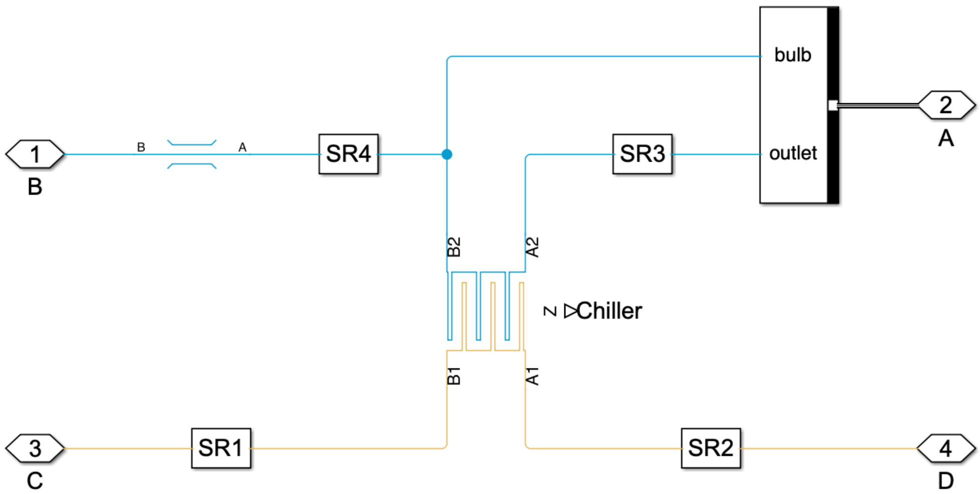

Figure 15: Chiller

1.3.5 Evaporator

Evaporator is exactly the same as the chiller, the only difference is that it is used to cool the HVAC circuit, with the refrigerant absorbing heat from the air entering the cabin, instead that from the coolant.

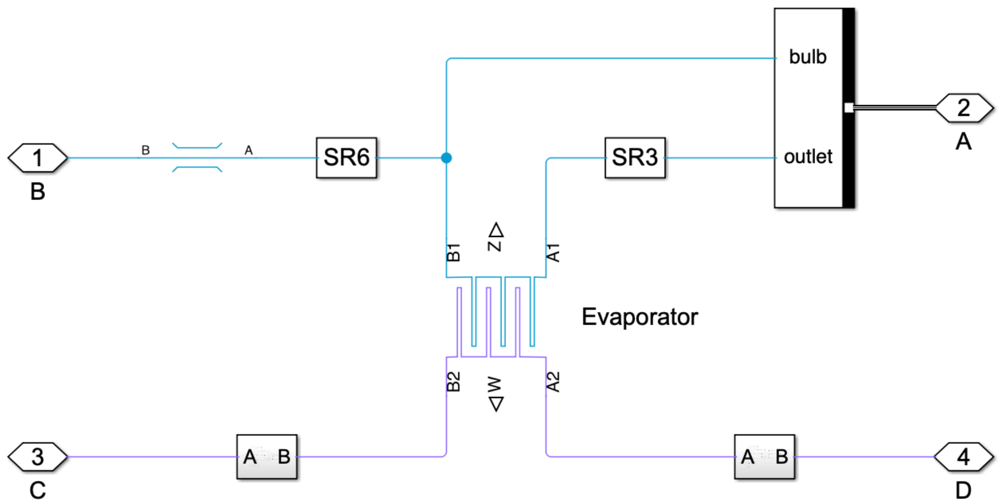

Figure 16: Evaporator

1.4. Battery cooling loop components

When the system is in parallel mode, the battery loop is composed by: battery, battery heater, DC-DC converter and battery pump.
This cooling loop has the purpose to keep the battery between 25 and 35 degrees, heating or cooling depending on the temperature.

1.4.1 Battery pump

The battery pump is responsible for the circulation of coolant in the battery loop and it is synchronized with the motor pump when the system is working in serial mode. The controlling command *cmd_battery_pump* gives the rpm that the pump has to keep; that goes as input in a ideal torque sensor block, it is used to extract the torque value necessary to compute the electrical power demand of the pump (as it was explained in the Compressor section).

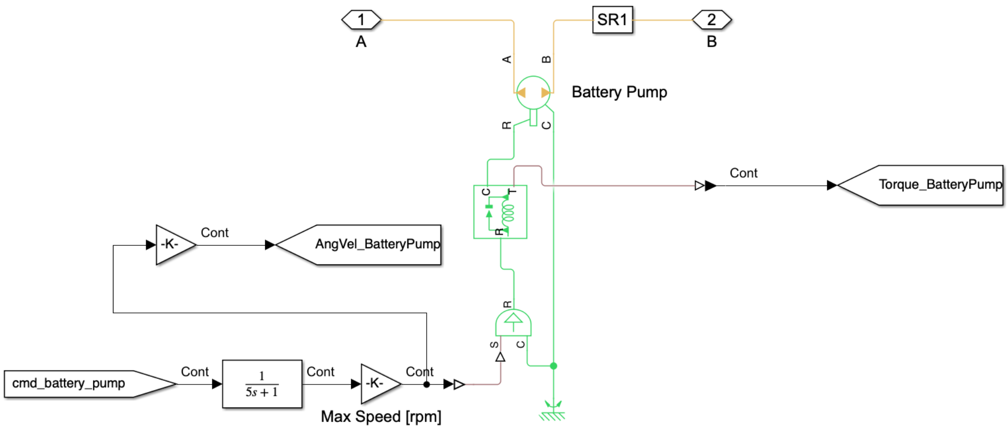

Figure 17: Battery pump

1.4.2 Battery

Battery model is design with an electric-thermal domain shown in figure 18. Starting from the left of the figure is possible to observe immediately the thermal model which represents the convective heat exchange between battery and coolant. Going to the right of the picture we enter in the electric domain of the battery, in which are present four battery packs connected in series. For each pack only one cell is modeled and its thermal and electrical behavior is replied for the others, in particular are present 20 cells for each pack. More on the right is present a voltage sensor to measure the actual voltage of the battery and a current generator block that represents both the powertrain and BTM currents, where the latter is computed dividing the total BTM power by the measured battery voltage.

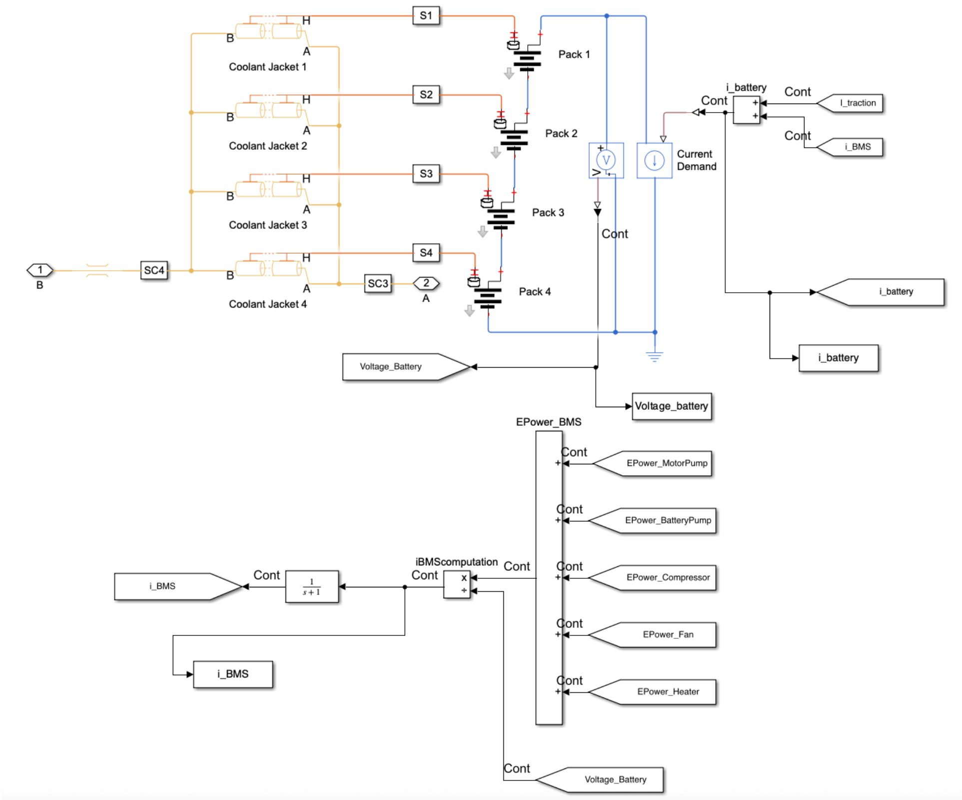

Figure 18: Battery

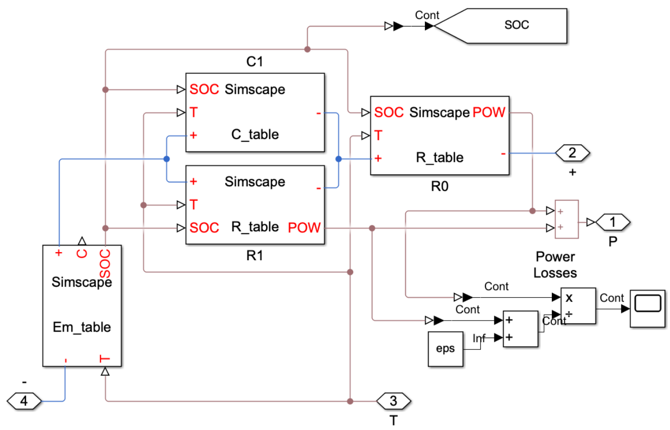

Figure 19: Battery cell

Focusing now on the battery pack, the cells modeled are lithium cells represented as a first order equivalent circuit (figure 19). First order model means that it has only one R-C group used to model the dynamic of the cell put in series with an R_0 resistance which represent the biggest part of the losses. Parameters such as the R_0, R_1 and C_1 ore provided by some look-up table as function of SOC and T_b. The single cell has a mass of 2.5 kg for a total weight of the battery of 200 kg and a capacity of 85.120 kWh.

1.5. HVAC components

The heating, ventilation and air conditioning system is responsible for maintaining the comfort in the cabin environment. It is in charge of cooling or heating the air in the cabin depending on the passengers requirements and external temperature, over maintaining the air quality filtering dust and CO2 and ensuring the right levels of humidity.
The HVAC circuit is an air circuit which main components are:

- PTC heater
- Cabin environment model
- Blower

The air is cooled passing through the evaporator; instead when is asked for an heating effect the component used is the ptc heater.

1.5.1 PTC heater

Thermal model of the ptc heater is shown in figure 20. This component is used when an heating effect is needed inside the cabin, it is shown how all the electric request from the ptc heater is converted into heat by Joule effect.

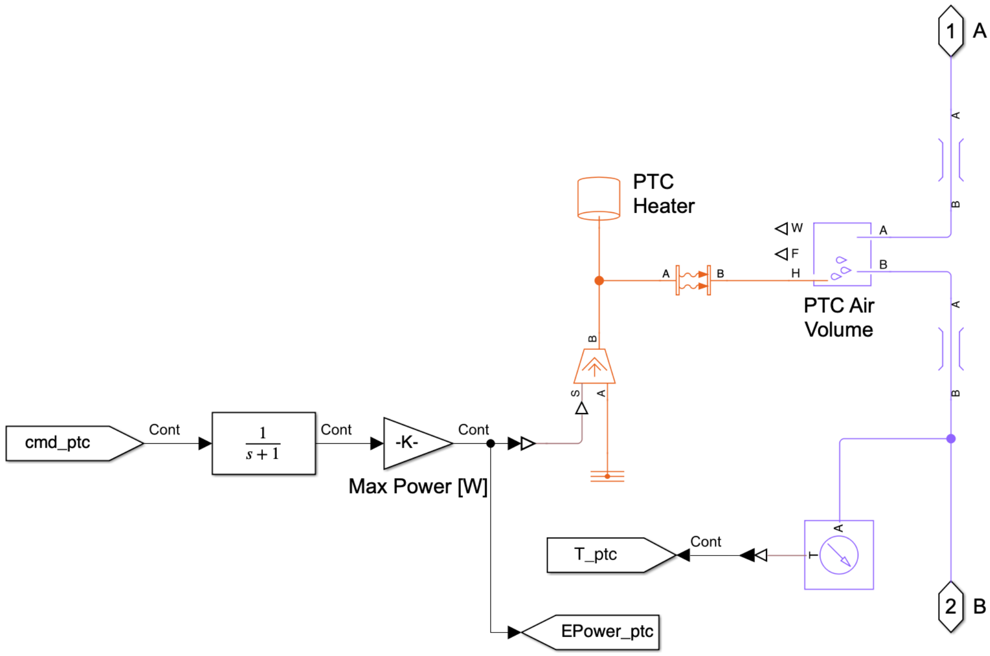

Figure 20: Ptc heater

1.5.2 Cabin

In figure 21 is described the model of the cabin. In the center is visible the block cabin air volume, in which the air coming from the ptc heater and evaporator (the one coming from the sensor block SC3) is mixed with:

- air leakages coming from external air entering the cabin air volume block in port C
- moisture and CO2 gains coming from the passengers breathing
- heat contribution coming from passengers due to the metabolic processes and heat dispersion/absorption by convection with the external environment trough glass, doors and roof

From the port B of the block it exits the mixed air that will enter the blower analyzed in the following.

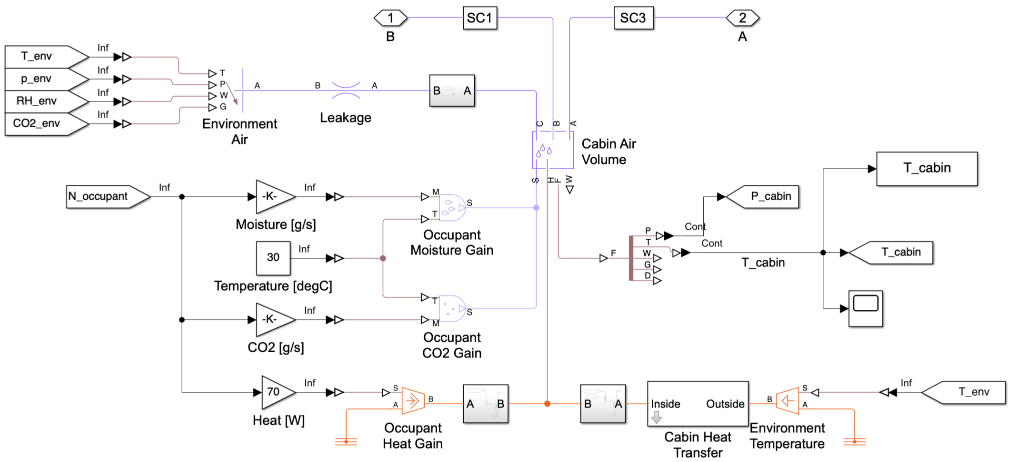

Figure 21: Cabin

1.5.3 Blower

The blower block is shown in figure 22. This component is responsible for the circulation of air in the HVAC circuit.
As observable the air coming from the cabin enter the recirculation flap, which acts as a sort of valve and decides of the air entering the blower what portion comes from the cabin air and what from the external environment. The extreme positions are 1 in case of full recirculation, which means that all the air is taken from the cabin environment, so there is no air coming from the outside of the vehicle, while in case flap position is 0 all the air pushed from the blower in the cabin comes from the external environment.
Once the air exit from the recirculation valve it is accelerated by the blower and pushed through the evaporator, ptc hater and then the cabin closes the loop.

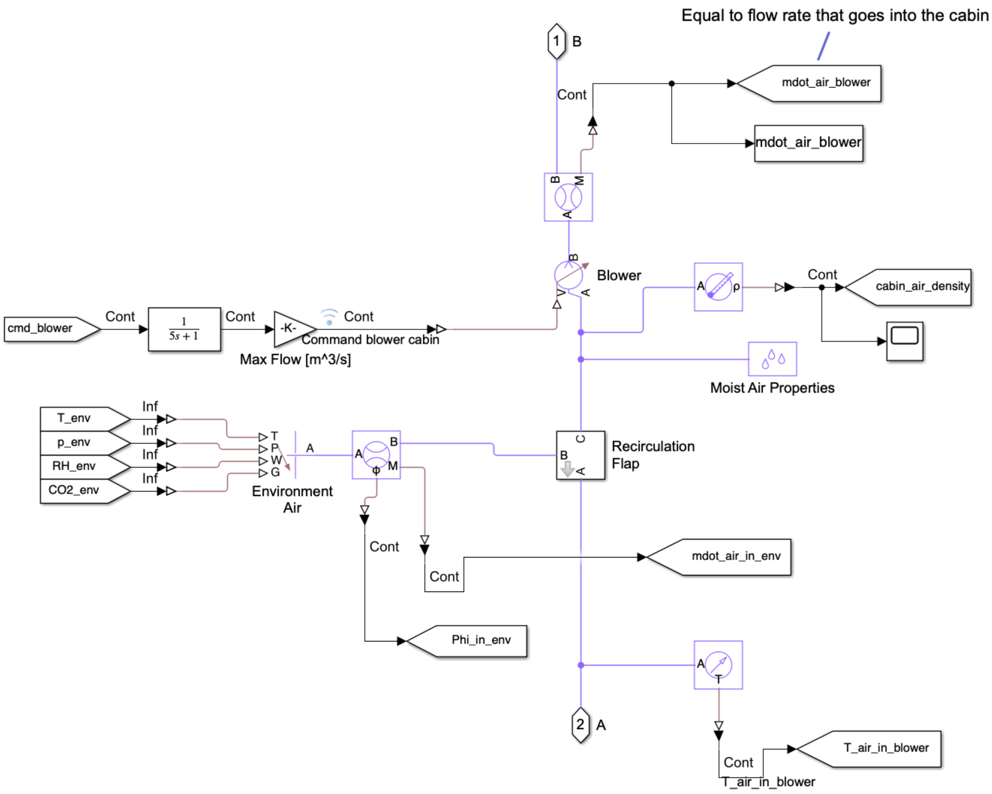

Figure 22: Blower

Chapter 2

Control strategy

In this chapter will be at first described and analyzed the original Reactive control strategy then in the second part an MPC introduction will be done.

2.1. Reactive control

A reactive control is a kind of controller which reacts to the variation of system's states or inputs relying exclusively on the real time information, without any form of prediction or past events memory. Pros of this kind of control systems are the generally simplicity in the development and implementation, light computational costs due to the absence of prediction and by consequence a fast response.
In the original model from MATLAB[17], the control system is based on this logic. Below is reported the Simulink scheme were are observable three subsystems: compressor, fan and coolant loop control. Original control will be described in detail

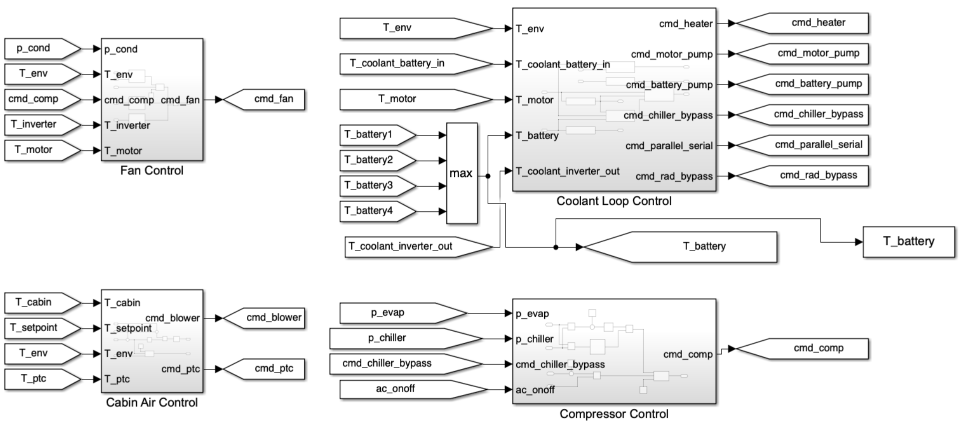

Figure 23: Original controls

because a good understanding of them will allow for a better comprehension of both the system's components interconnections and the comparison between reactive and adaptive MPC that will be done in chapter 4.
A particular focus will be done on Compressor, blower, ptc and battery pump controls which are the controls that will be managed by the MPC, all the other parts will just be briefly described.

2.1.1 Fan Control

Fan Control subsystem reported in figure 24 has the following characteristics:

- inputs: pressure at the condenser outlet, environmental temperature, compressor command, Inverter temperature and motor temperature
- output: fan command

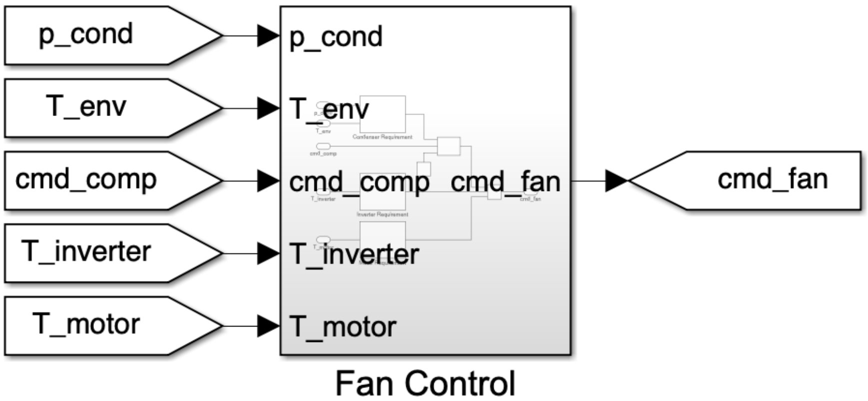

Figure 24: Fan original control

The command generated by the block is used to regulate the fan with a double purpose:

- to regulate the air flow that goes through the condenser and so regulating the condensation of two-phase liquid of the refrigerant loop
- to cool down the radiator's coolant

By going a bit more in the detail, as shown in figure 25 the command of the fan is a value between 0 and 1, it is generated by taking the maximum value between:

- Condenser Requirement: inputs of this block are the pressure at the condenser outlet and the environmental temperature. The command generated by this block is the sum of three stage relay; each relay is activated comparing the pressure of the condenser with an high and low threshold. Each relay contribute for 1/3 of the command, therefore in this case possible commands are 0,1/3,2/3 and 1.
  Command generated by the condenser requirement block is not taken into account in case the command of the compressor is equal to zero, this is because in case compressor is stopped it means that the refrigerant loop is not working.
- Inverter Requirement: also in this case we have a three relay stage signal, each relay is activated or deactivate depending on the current inverter temperature.

- Motor Requirement: the command signal generated by this block come from a look up table that generate it proportionally to the motor temperature, starting the fan at 45°C and saturating to 1 from 75°C on.

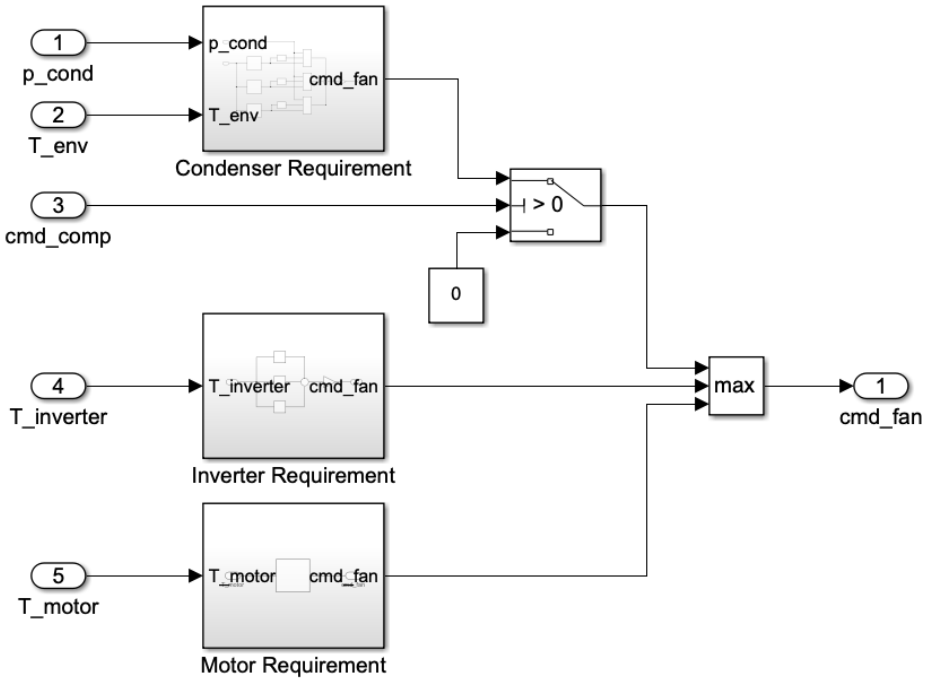

Figure 25: Fan original control detail

2.1.2 Coolant Loop Control

Coolant Loop Control reported on figure 26 has the following characteristics:

- inputs: environmental temperature, coolant temperature entering the battery coolant jacket, motor temperature, battery temperature and coolant temperature exiting the inverter coolant jacket.

- outputs: battery heater, motor pump, battery pump, chiller bypass, parallel-serial and radiator bypass valve commands.

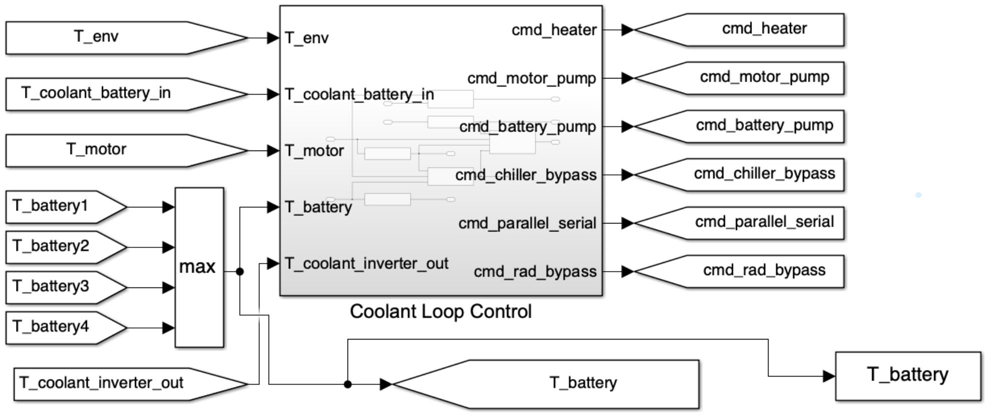

Figure 26: Coolant Loop original control

Main target of the coolant loop control is to manage the circulation of coolant in crucial components of the vehicle such as battery pack, charger, inverter, motor and DC-DC converter. Managing the coolant means also to decide the optimal configuration of the circuit depending on the conditions, therefore this block also controls the actuation of chiller and radiator bypass valve over the four way valve for parallel-serial mode decision.

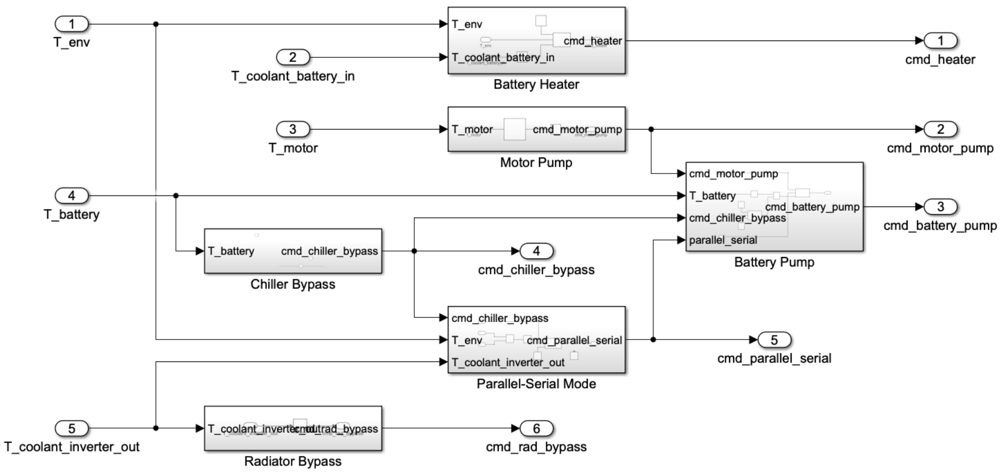

Figure 27: Coolant loop control original detail

Now, by going a bit more in the details (see Figure 27), we can describe the different blocks of the coolant loop control subsystem:

- Battery heater: this component has the role of heating the battery when the environmental temperature is below 5 degrees, avoiding the increase of impedance in the cells of the battery and by consequence the reduction of energy that can be extracted from it. Furthermore, if the environmental temperature is below 5 degree the heater is switched on and subsequently switched off only when the coolant entering the battery reach 15°C.

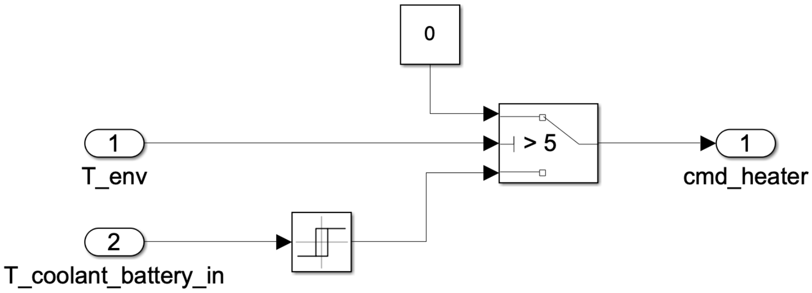

Figure 28: Battery heater original control detail

- Motor pump: this block produce an output from 0.3 to 1 that refers to the maximum angular velocity of the motor pump. This value is generated by a look up table depending on the current motor temperature.
- Chiller bypass: this control the displacement of the chiller by pass valve. The output command in this case can only be 1 or 0, the first case means that the valve is opened so the chiller is by-passed, while the latter case means that the full flow from the battery pump goes through the chiller. With this kind of controller the valve is opened when the battery temperature drops below the target set (30°C in our case) and closed if temperature reach 35°C.
- Radiator bypass: the concept here is the same described for the chiller by pass valve. Command for the valve is 1 (radiator by-passed) if the temperature of the coolant in the hottest part of the motor cooling circuit, which is the exit of the inverter's coolant jacket, is below 20°C and equal to 0 (no bypass) if temperature of the coolant exceed 25°C.
- Parallel-Serial mode: in figure 29 is reported the logic which decide if the system has to work in parallel or serial mode acting on the four way valve. When chiller is not by-passed the mode is parallel to guarantee maximum cooling capacity. if chiller is by-passed the mode can be both serial or parallel depending on the maximum between environmental and coolant exiting the inverter temperature; in particular mode will be parallel if the maximum between mentioned temperatures is above 35°C and serial if below 30°C.

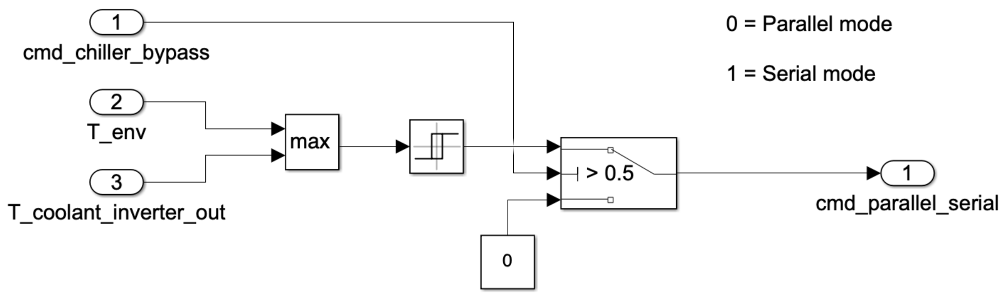

Figure 29: Parallel-Serial mode original control detail

- Battery pump: finally we come to the last block which control the battery pump following the scheme reported in figure 30 . As for the motor pump, the battery pump command is referred to the maximum angular velocity. First of all we can see that if the mode is serial the battery pump is synchronized with the motor's one; while in parallel mode the command is selected as the maximum values between the output of a relay which is set to 1 if the battery temperature goes above 50°C and 0 when below 45°C and 0.3 or 0.5 depending if the chiller is by-passed or not. So we can see how in parallel mode the command will always be between 0.3 and 1.

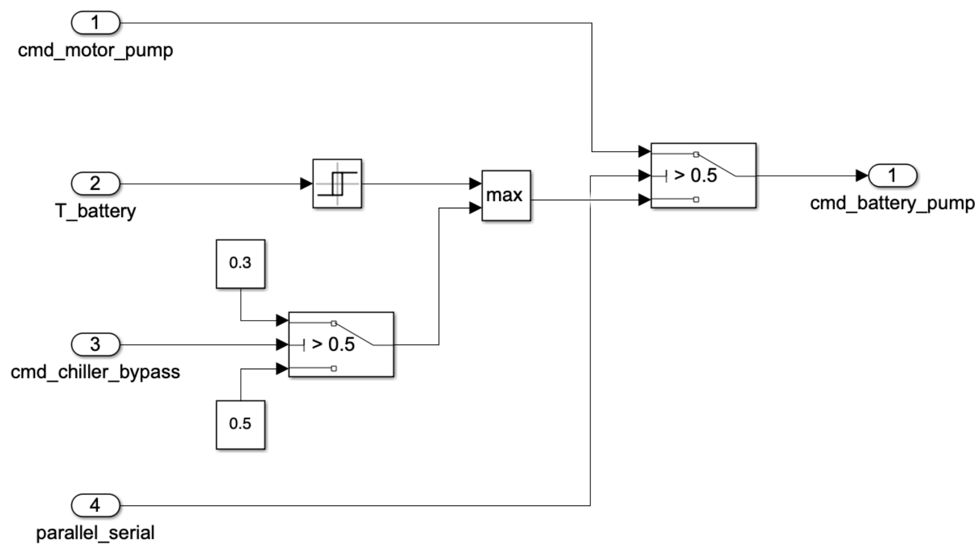

Figure 30: Battery pump original control detail

2.1.3 Cabin Air Control

Cabin Air Control reported in figure 31 has the following characteristics:

- inputs: cabin temperature, set cabin temperature, environmental temperature and ptc heater temperature
- outputs: blower and ptc heater commands.

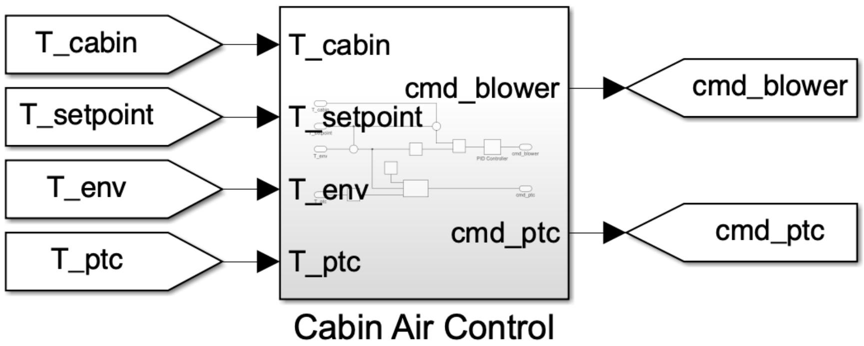

Figure 31: Cabin Air Control original

This control block plays a crucial role in maintaining and guarantee cabin comfort. Is important to underline that as shown in figure 32 the command of the blower is given by a PI controller. Due to the presence of the integrative part this controller is not considerable a pure reactive controller like the ones described previously since it takes into account the error in the previous time instances.

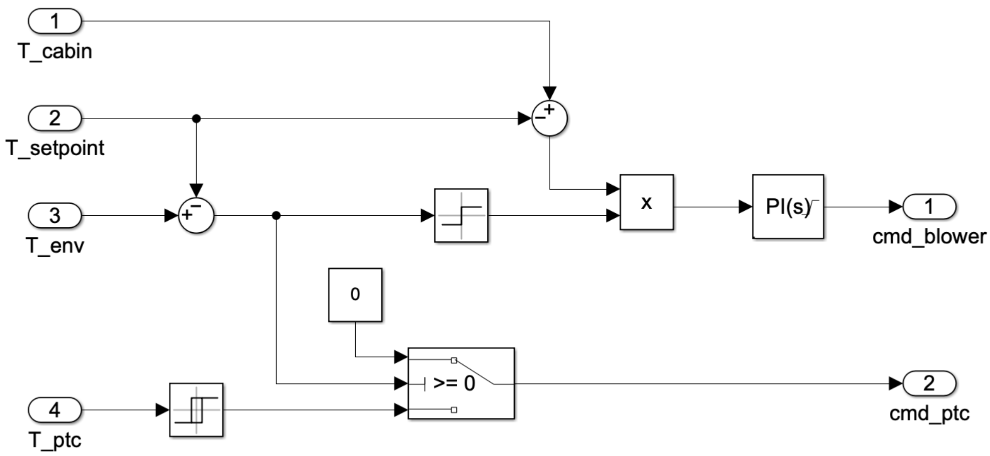

Figure 32: Cabin Air Control original detail

The error on the cabin temperature is computed and then multiplied by 1, -1 or 0 depending if the environmental temperature is higher, lower or equal to the temperature target set; then the PI control computes the input for the blower able to minimize the error. Cabin heater is activated only when the set temperature is higher than the environment's one and its command value will be 1 when the heater's temperature is lower than 35°C and 0 when higher than 45°C.

2.1.4 Compressor Control

The compressor control reported in figure 33 has the following characteristics:

- inputs: pressure a the exit of chiller and evaporator on the two phase thermal liquid side, chiller bypass command and ac on/off command.
- output: compressor command

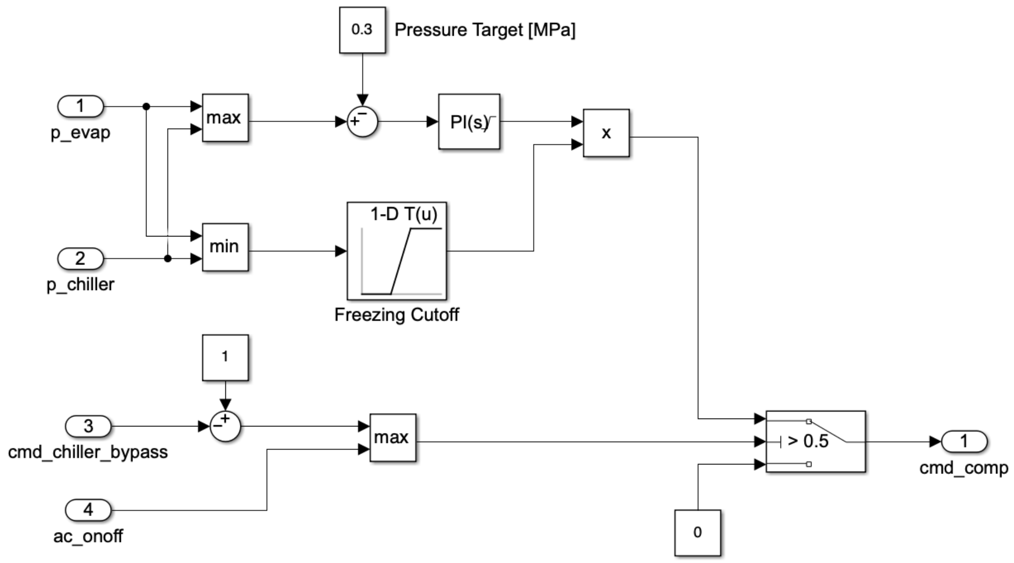

Figure 33: Compressor control original

This control has the purpose of properly activating the compressor when necessary to cool down the battery through the chiller or the cabin through the evaporator. As it was for the blower, also in this case we have a PI controller with proportional part equal to 1 and integrative equal to 0.2, so this is not a pure reactive control. The PI controller essentially has the aim of minimizing the error from the target pressure of 0.3 MPa, taking as measure the maximum between chiller and evaporator pressure. The output of the PI controller is then corrected by a cut-off coefficient. The latter is extracted from a look up table (figure 34) which take as input the minimum between evaporator and chiller pressure. The cut-off coefficient has the purpose to reduce or stop the compressor in case the pressure at the exit of chiller and evaporator is too low, this is done for safety reasons since an excessive low pressure can cause:

- condensation and freezing of moisture in the air, forming a layer of ice on the evaporator and chiller fins, reducing the heat exchange and consequently reducing the overall system efficiency, causing also an useless and excessive energy consumption [21]

- damaging of components such as the compressor since excessive low temperatures can cause a phenomenon note as liquid slagging due to which a part of refrigerant can reach the compressor, which is designed to work with gas, in liquid form consequently damaging it

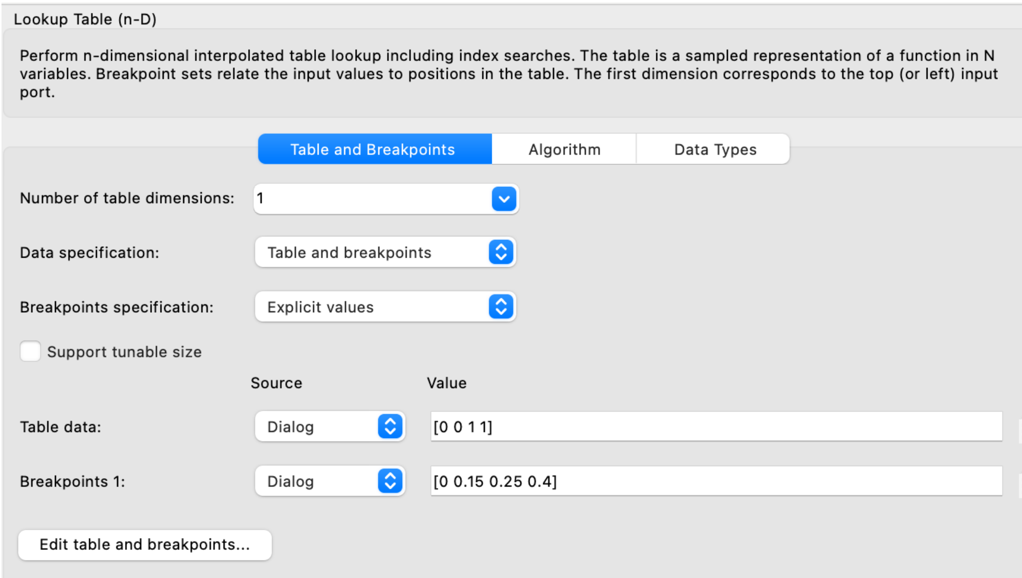

Figure 34: Compressor freezing cutoff table

And to conclude compressor control description, is also important to observe that compressor is deactivated when chiller is bypassed and at the same time cabin recirculation is off (0).

2.2. Model Predictive Control

In this section, Model Predictive Control (MPC) is described explaining the idea and working principle behind this kind of controller.

2.2.1 MPC working principle

Unlike reactive controls, MPC is an optimal controller; this means that it computes the sequence of input for the system in order to minimize the cost described by the cost function over the prediction horizon (p). In practice, at each time step the MPC calculates the optimal sequence of control action for the setted prediction horizon and applies only the first action of the sequence, discarding the following ones. The process is then repeated[22].

In figure 1 is exposed the MPC control loop.

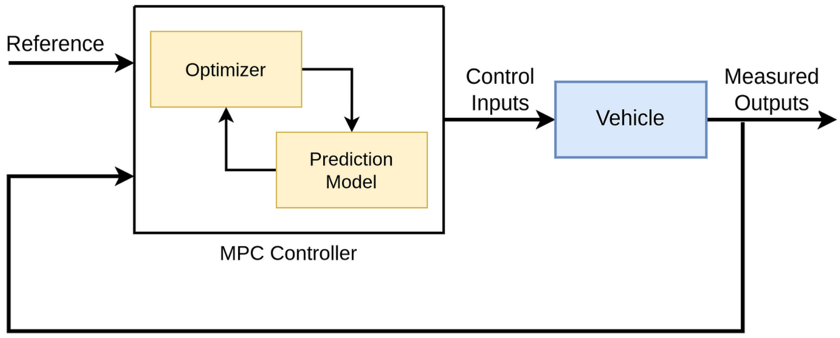

Figure 35: MPC functioning scheme

MPC controller is composed by two elements: the optimizer and the prediction model, both will be discussed in the following sections.
The controller receives as feedback outputs and measured disturbances from the system's plant, comparing them with the references. Then by using the internal prediction model of the plant, the optimizer calculates the optimal control sequence of control that minimize the cost function over the prediction horizon.
This kind of controller over the fact of having the capabilities to maximize performances and consumptions, are also much more suitable for systems that are complex and have a slow dynamic.

2.2.2 Prediction model

The prediction model is defined by a set of equations that describe the evolution in time of the states we want to control.
The purpose of this model is to be reliable and accurate in predicting the evolution of the system, but at the same time it does not have to be too heavy from the computational point of view allowing for a real-time optimization.

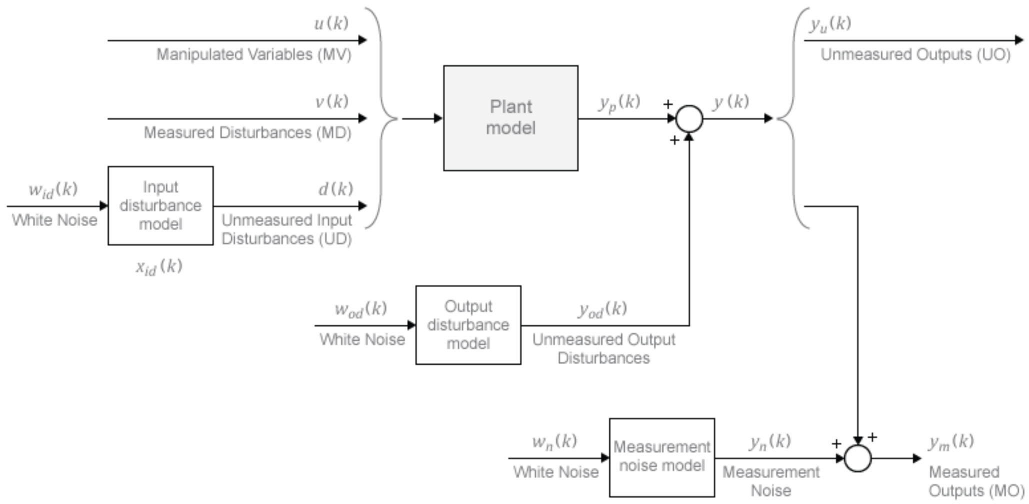

Figure 36: MPC plant (prediction) model functioning scheme[5]

From above we can observe how the prediction model have as inputs:

- **Manipulated Variables (MV)**: which are the input values on which the controller can acts modifying them to control the plant behavior. Their optimal control sequence is calculated across all the prediction horizon and then, as mentioned before, only the first action of the sequence is used as input of the plant for each time step.

- **Measured Disturbances (MD)**: which are internal or external plant's perturbations that can be measured in real time with a good accuracy, such as data coming from sensors.

- **Unmeasured Disturbances (UD)**: which are disturbances that can not be directly measured, their presence can induce uncertainties in the predictions, so it is important to design a controller which is tolerant to then or that is able to compensate them.

2.2.3 Cost function

A said before the MPC solves an optimization problem, in particular it solves a quadratic programming (QP) type problem.
This kind of problems are defined by three elements: Cost function, Constraints, decision variables.
Cost function is the sum of four therms:

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_\varepsilon(z_k) \tag{2.2.1}$$

Where the first term refers to the output reference tracking, the second to the manipulated variables tracking, the third to manipulated variables move suppression

and the latter to constraint violation.
The developed cost function is reported below:

$$J_y(z_k) = \sum_{j=1}^{n_y}\sum_{i=1}^{p}\{\frac{w_{i,j}^y}{s_j^y}[r_j(k+i|k) - y_j(k+i|k)]\}^2 \tag{2.2.2}$$

$$J_u(z_k) = \sum_{j=1}^{n_u}\sum_{i=0}^{p-1}\{\frac{w_{i,j}^u}{s_j^u}[u_j(k+i|k) - u_j(k+i|k)]\}^2 \tag{2.2.3}$$

$$J_{\Delta u}(z_k) = \sum_{j=1}^{n_u}\sum_{i=0}^{p-1}\{\frac{w_{i,j}^{\Delta u}}{s_j^{\Delta u}}[u_j(k+i|k) - u_j(k+i-1|k)]\}^2 \tag{2.2.4}$$

$$J_\varepsilon(z_k) = \rho_\varepsilon \varepsilon_k^2 \tag{2.2.5}$$

In the first term of the cost function ($J_y(z_k)$), $r_j(k+i|k)$, which represent the reference values received for the entire prediction horizon, is compared to $y_j(k+i|k)$ that are the predicted plant output which depend from measured disturbances (MDs), state estimates and manipulated variables adjustments (z_k), the latter is a vector explicited below:

$$z_k^T = [u(k|k)^T \quad u(k+1|k)^T ... u(k+p-1|k)^T \quad \varepsilon_k] \tag{2.2.6}$$

Second term of the equation ($J_u(z_k)$) do the same as the first but for the MVs instead that the outputs.
$J_{\Delta u}(z_k)$ penalize the changing in value of the MVs from an instant to the next, and $J_\varepsilon(z_k)$ as said before deal with the constraint violation, softening or hardening it depending if the value of ρ_ε is low or high.

Now speaking about the constraints they are the conditions that the solution of the QP problem must satisfy, such as physical bounds on the manipulated manipulated and plant's output variable.
Below are reported the typical constraints for an MPC:

$$\frac{y_{j,min}(i)}{s_j^y} - \varepsilon_k V_{j,min}^y(i) \leq \frac{y_j(k+i|k)}{s_j^y} \leq \frac{y_{j,max}(i)}{s_j^y} + \varepsilon_k V_{j,max}^y(i) \tag{2.2.7}$$

$$\frac{u_{j,min}(i)}{s_j^u} - \varepsilon_k V_{j,min}^u(i) \leq \frac{u_j(k+i-1|k)}{s_j^u} \leq \frac{u_{j,max}(i)}{s_j^u} + \varepsilon_k V_{j,max}^u(i) \tag{2.2.8}$$

$$\frac{\Delta u_{j,min}(i)}{s_j^{\Delta u}} - \varepsilon_k V_{j,min}^{\Delta u}(i) \leq \frac{\Delta u_j(k+i-1|k)}{s_j^{\Delta u}} \leq \frac{\Delta u_{j,max}(i)}{s_j^{\Delta u}} + \varepsilon_k V_{j,max}^{\Delta u}(i) \tag{2.2.9}$$

With:

$$i = 1 : p, \quad j = 1 : n_u \tag{2.2.10}$$

First refers to the limits about outputs, second about manipulated variables (MVs) and the latter to manipulated variables rate. The terms $V_{j,min}^*(i)/V_{j,max}^*$ are used to define if the constraint is an hard (0) or soft(>0) constraint, while ε_k is a scalar QP slack variable, that is to say a dimensionless number used to soften the constraints. Decision variables' values, also called MVs, are determined by the solution of the minimization problem[23].

2.2.4 Adaptive MPC

Before describing the difference between and Adaptive MPC and a conventional one is important to say that the system to be controlled by the MPC must be modeled as a linear time invariant (LTI), state space (SS).
An LTI system can be expressed both in continuous[24]:

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) + D_c u(t) \end{aligned} \tag{2.2.11}$$

or discrete time:

$$\begin{aligned} \dot{x}(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{2.2.12}$$

The MPC's plant model will be as follow, having as inputs manipulated variables (u), measured disturbances (v) and unmeasured disturbances (d).

$$\begin{aligned} \dot{x}(k+1) &= Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \\ y(k) &= Cx(k) + D_v v(k) + D_d d(k) \end{aligned} \tag{2.2.13}$$

Now we can say Adaptive Model Predictive Control (AMPC) is preferred over traditional MPC when the system's dynamics are time-varying, uncertain, or nonlinear. While traditional MPC relies on a fixed model of the system, AMPC continuously updates the model based on real-time data, making it more effective in handling changes such as parameter variations, unmodeled dynamics, or external disturbances. It is particularly useful in environments where system behavior cannot be accurately predicted in advance, such as in the presence of aging components, wear, or unpredictable conditions. Adaptive MPC provides enhanced performance and robustness by adjusting the control strategy as the system evolves, obtaining the matrices $A, B_u, B_v, B_d, C, D_v, D_d$ through the linearization of the model at each time step on the current state of the system.

Chapter 3

Cabin/battery priority control

In this chapter will be analyzed step by step the developed AMPC controller for BTM and HVAC.
Starting from the work of Domenico Altavilla[25], in which the HVAC circuit was removed for simplification and an AMPC controller able to manage the compressor was developed, in this thesis the HVAC circuit was reintegrated as in the original model described in Chapter 1, furthermore the AMPC was developed to control both the battery cooling (through the battery pump and the compressor) and the cabin cooling/heating (through the blower and the ptc heater).
What was done consists in two different AMPC controller, the first giving priority to the regulation of the battery temperature, tracking a set temperature target. The second one gives priority to the set cabin temperature, maintaining the desired comfort for the passenger. The controller used by default is the cabin priority one to ensure cabin comfort, the switch to the other is done only when the battery temperature reaches critical values, in this scenario battery cooling becomes the main goal, slightly sacrificing cabin's comfort (still maintaining a good target tracking as shown in Chapter 4), until the battery comes back to a safe temperature to avoid excessive thermal inefficiencies and internal chemical reactions acceleration that may cause a premature aging of the batteries[8].
In figure 37 is reported the block scheme of the controller designed in Simulink:

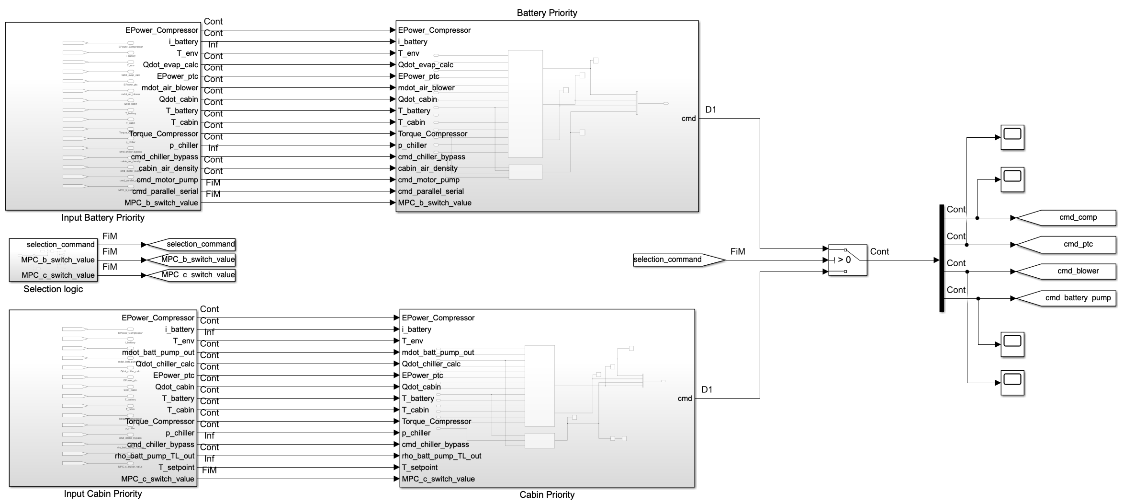

Figure 37: Cabin/Battery control

The two blocks on the left are just use to feed battery and cabin priority blocks.

As observable this control designed is responsible for the control of compressor, blower, motor pump and ptc heater, while radiator's fan, battery heater, motor pump, bypass valves and four way valve remains governed by the the reactive logic (figure 38) as described in Chapter 2.

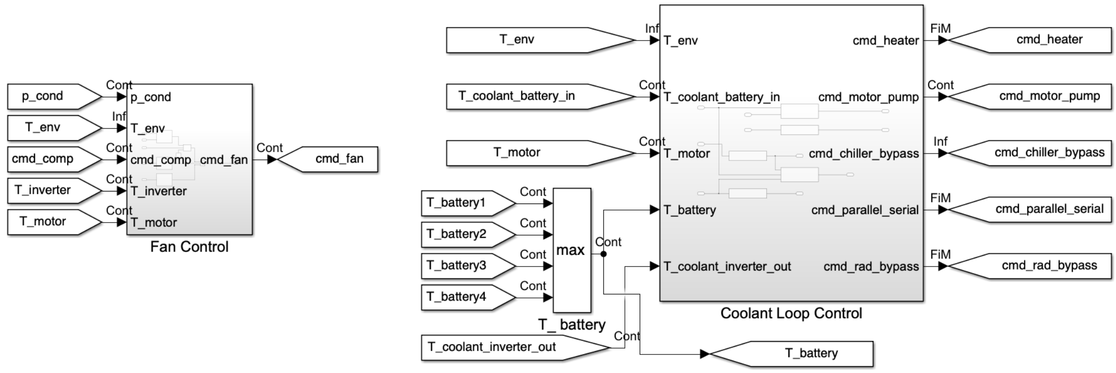

Figure 38: Reactive controls

In the following sections it will be described both the battery and cabin priority MPC with related prediction models, cost functions and weight tuning, beyond the selection logic, the simulation initialization through MATLAB and the reasons for why it was necessary to develop two different MPC.

3.1. Battery priority

Battery priority control is organized in two sub-blocks:

- Compressor, ptc and blower control: responsible for the control of the cited components through an AMPC.
- Battery Pump: responsible for the control of the latter using the same reactive logic described in Chapter 2.

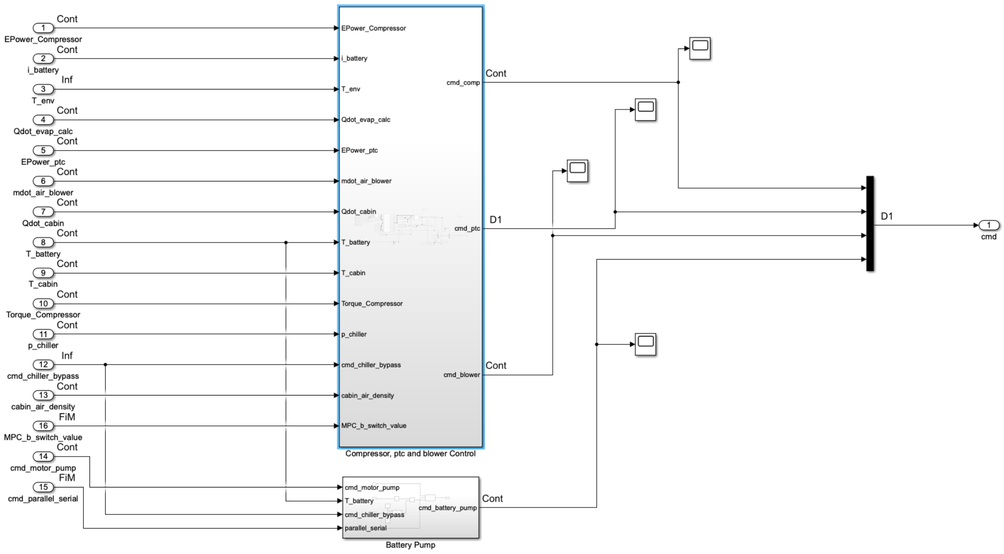

Figure 39: Battery priority control detail

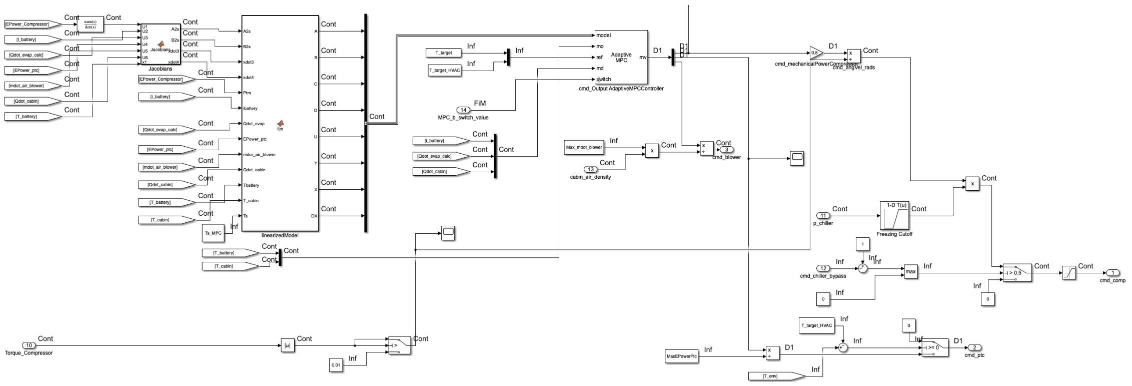

Figure 40: Battery priority AMPC overview

The detail of the first cited block is reported in figure 40, where main sections are:

- **Jacobian**: this block is generated through the command matlabFunctionBlock in MATLAB that will be see in section 3.5. This command has the purpose

to convert symbolic expression to MATLAB function block that can be used in Simulink. More in detail, the goal of the block is the linearization of the system around the current working point, therefore MDs and MVs are give as input to the block.
Linearization is made because, as said in section 2.2.4, the MPC needs that the system to be controlled is modeled as an LTI system; as will be analyzed in section 3.1.1, the state equations of the prediction model represent a non-linear time-invariant system which can be expressed in general form as follow:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.1.1}$$
$$y(t) = h(x(t), u(t)) \tag{3.1.2}$$

Let consider $\bar{u}$ as the constant input (represented by the current values of MDs and MVs) and $\bar{x}$ the equilibrium states of the state equations corresponding to $\bar{u}$. After a generic small displacement from the linearization point is defined:

$$\delta x(t) = x(t) - \bar{x} \tag{3.1.3}$$
$$\delta u(t) = u(t) - \bar{u} \tag{3.1.4}$$
$$\delta y(t) = y(t) - h(\bar{x}, \bar{u}) \tag{3.1.5}$$

By doing the Taylor expansions of f and h around $(\bar{x}, \bar{u})$ truncated at first order the system can be rewritten as follow:

$$\delta\dot{x}(t) = A\delta x(t) + B\delta u(t) \tag{3.1.6}$$
$$\delta y(t) = C\delta x(t) + D\delta u(t) \tag{3.1.7}$$

Where A,B,C,D are the Jacobian matrices computed from the partial derivatives of f and h:

$$A = \left[\frac{\partial f}{\partial x}\right]_{(\bar{x},\bar{u})}, \quad B = \left[\frac{\partial f}{\partial u}\right]_{(\bar{x},\bar{u})} \tag{3.1.8}$$
$$C = \left[\frac{\partial h}{\partial x}\right]_{(\bar{x},\bar{u})}, \quad D = \left[\frac{\partial h}{\partial u}\right]_{(\bar{x},\bar{u})} \tag{3.1.9}$$

Is always important to consider that the linearized system is an approximation of the nonlinear system holding in a neighborhood of the point $(\bar{x}, \bar{u})$, therefore when the current operating point detaches from the linearized one the error of the approximation may rapidly increase.

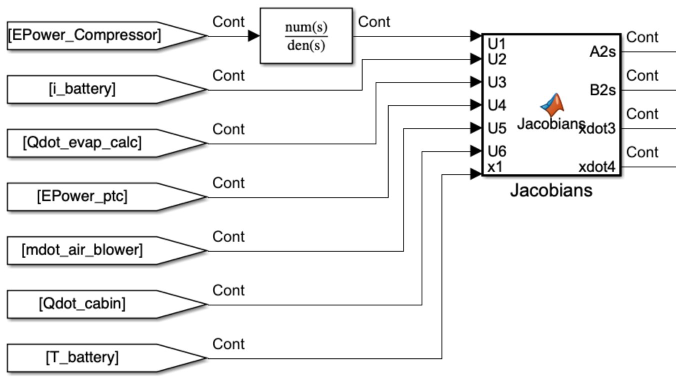

Figure 41: Jacobian block

- **Linearized model**: this block is responsible for the building of linearized matrix in the current operating point. It transmits with a bus all this data entering the model port of the AMPC block described in the following. As visible the bus contains matrices of the LTI system A, B, C, D, the current input which are the the MVs plus MDs in the matrix U; Y and X contains the current states and DX the discrete state gains.

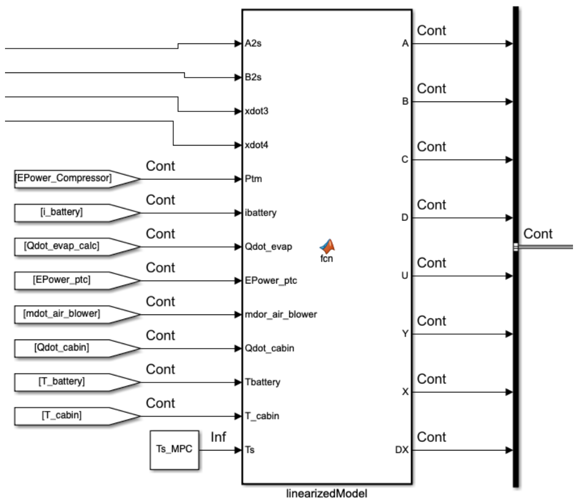

Figure 42: Linearized model

- **Adaptive MPC block**: this block receive from the linearized model MATLAB function the matrices of the LTI model used by the MPC for the prediction through the solving of the QP optimization problem. It also receives measured output (MO) from sensors (which correspond to the states T_b and T_c), target values (ref) used as reference state to track, the actual values of the MDs and an external switch signal used to enable or disenable the QP optimization, this is done to lighten the computational effort when the battery priority MPC is not the one chosen to be used by the selection logic described in section 3.3.

This choice was done with the purpose of permitting the MPC block to receive as feedback the current values of MDs also when the QP optimization is not required, this allow to have a smoother transition between one controller to the other when required by the selection command block compared to completely disenable the Battery priority block when its usage is not required, as will be described in Chapter 4.
The AMPC block contain the MPC object which is used to set all the parameters and properties of the controller's prediction model plant, it will be described in section 3.5.

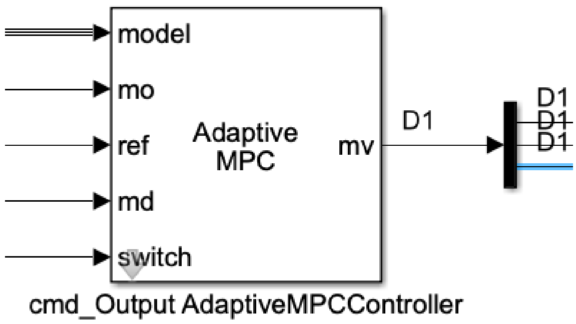

Figure 43: AMPC block

- **MPC's MVs**:
  - Compressor command: as shown in figure 44 the command exiting the MPC is related to the electrical power, but since the input of the compressor must be angular velocity, the MPC's output is multiplied by 0.8 so the mechanical power is obtained, then divided by the torque coming from the ideal torque sensor visible in figure 12 obtaining the angular velocity which is multiplied by the freezing cutoff coefficient which has the purpose to reduce compressor command in case of excessive low pressure for the reasons described in section 2.1.4, as also he fact that the compressor is deactivate when both ac recirculation is off and chiller is bypassed.

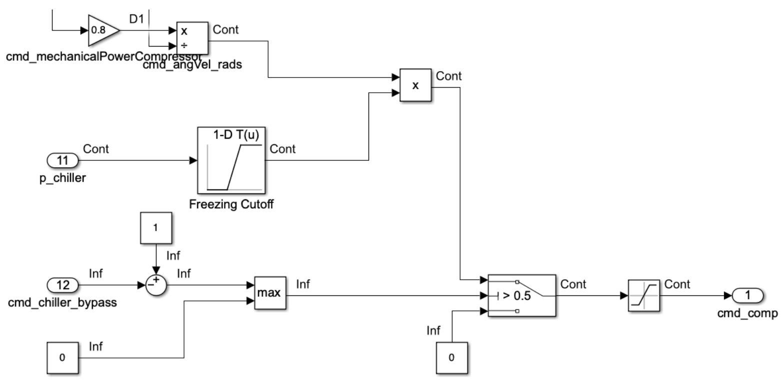

Figure 44: Battery priority control, compressor command

– Ptc command: this command exit from the MPC as a heat exchanged power, it is divided by the maximum ptc heater power, obtaining the amount of power we want it to work (from 0 to 1) related to its maximum one. If the value is major than zero it is used as ptc command just if the environmental temperature is lower than the target temperature for the cabin.

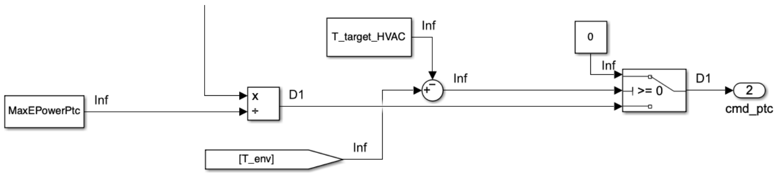

Figure 45: Battery priority control, ptc command

– blower command: the MPC return as output the mass flow rate $[kg/s]$for the blower, but it needs as command a value from 0 to 1 related to the volumetric flow rate $[m^3/s]$, therefore the values from the MPC is divided by the product between the maximum blower mass flow rate by the actual cabin air density.

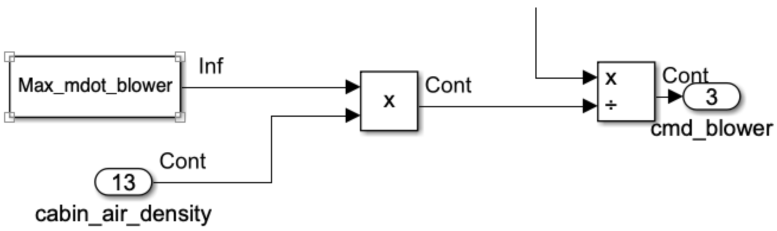

Figure 46: Battery priority control, blower command

3.1.1 Prediction model

Prediction model is used from the MPC to calculate the future states of the system to estimate the best control sequence that minimize the cost function. It is necessary

to find a compromise between accuracy of the prediction and computational cost.
As purpose of this thesis, the MPC controller has to manage both the BTM than the HVAC, therefore the state to estimate are battery and cabin temperature. The approach used is to start from the first law of thermodynamic, neglecting convective heat dispersion with external environment for simplicity.
Since state to predict are two, same will be the number of state equations:

- **Battery Temperature**: for the estimation of the battery temperature is considered the battery cooling loop, from this circuit as reported in the starting equation 3.1.10, we can say that the variation of internal energy is represented by multiplying mass of the battery cell with the specific heat of the cell with the temperature variation. Heat losses produced by the battery are modeled as Joule effect losses with a corrective factor, while heat extracted by the chiller is the total cooling capacity minus the cooling extraction from the evaporator.

$$m_b \cdot c_{p,b\_cell} \cdot \dot{T}_b = a \cdot R_0 \cdot i_b^2 - \dot{Q}_{chiller} \tag{3.1.10}$$

$$m_b = n_{cell} \cdot m_{cell} \tag{3.1.11}$$

  Where i_b take into account both the current used for traction then the one used by the BTM itself:

$$i_b = i_{b\_traction} + i_{b\_BTM} = i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}} \tag{3.1.12}$$

  Battery internal resistance is calculated as function of battery temperature and SOC interpolating (see section 3.5) the values of a look-up table already present in the original MATLAB model.
  The total cooling power of the VCR circuit is modeled as the product between the actual compressor power and a sort of fictitious coefficient of performance (COP) described in section 3.4:

$$COP \cdot P_{comp} = \dot{Q}_{chiller} + \dot{Q}_{evap} \tag{3.1.13}$$

  Substituting the equation 3.1.13 in 3.1.10 we obtain the first state equation;

$$\dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}} \right)^2 - COP \cdot P_{comp} + \dot{Q}_{evap} \right) \tag{3.1.14}$$

  with:

$$\dot{Q}_{evap} = \dot{m}_{blower} \cdot c_{p,air\_out\_evap} \cdot (T_{cabin} - T_{air\_out\_evap}) \tag{3.1.15}$$

  T_{cabin} has been assumed to be equal to $T_{air\_in\_evap}$ because for the simulations reported in Chapter 4 it was assumed that cabin recirculation is always active, in case air is also taken from the outside this assumption don't held anymore.

- **Cabin temperature**: for the estimation of the cabin temperature is taken into consideration the HVAC loop. Here the procedure is similar to the one described for the battery temperature prediction, starting from first law of thermodynamic, the variation of internal energy is defined as the product between mass of air in the cabin, specific heat of the cabin air and cabin temperature variation (equation 3.1.16). While the heat exchanges here are due to ptc heater, evaporator, cabin convection heat exchange with external environment and generated heat from the metabolic functions of passenger inside the vehicle. Also in this case thermal losses with the external environment are neglected for simplification.

$$m_{air,cabin} \cdot c_{p,air\_cabin} \cdot \dot{T}_c = \dot{Q}_{ptc} - \dot{Q}_{evap} + \dot{Q}_{cabin} \tag{3.1.16}$$

$$\dot{Q}_{cabin} = \dot{Q}_{cab,conv} + \dot{Q}_{met} \tag{3.1.17}$$

$$\dot{Q}_{ptc} = cmd\_ptc \cdot P_{max\_ptc} \tag{3.1.18}$$

Prediction model inside the MPC can be finally written as:

$$\begin{cases} \dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}} \right)^2 - COP \cdot P_{comp} + \dot{Q}_{evap} \right) \\ \dot{T}_c = \frac{1}{m_{air,cabin} \cdot c_{p,air\_cabin}} \cdot (\dot{Q}_{ptc} - \dot{Q}_{evap} + \dot{Q}_{cabin}) \end{cases} \tag{3.1.19}$$

From this system is observable how the variables on which the controller can act are the compressor power, ptc heater and the blower mass flow rate, therefore then will be the MVs, while the quantities directly measured are the battery current used for traction, the evaporator heat exchange divided by the blower mass flow rate and the cabin heat exchange, those are the MDs. COP is function of P_{comp}, R_0 is function of the battery temperature. All the other quantities present in the prediction model are considered constant and so don't need to be refreshed at each time step, but are just set in the beginning.

	variable	**new name**
MVs	P_{comp}	U1
	$\dot{Q}_{ptc}$	U4
	$\dot{m}_{blower}$	U5
MDs	$i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}}$	U2
	$c_{p,air\_out\_evap} \cdot (T_{cabin} - T_{air\_out\_evap})$	U3
	$\dot{Q}_{cabin}$	U6

Table 1: MVs and MDs battery priority model

The system can then be rewritten as in equation 3.1.20 that is then the same we will observe in section 3.5, where the variables from U1 to U6 are the feedback variables of the controller.

$$\begin{cases} \dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(U2 + \frac{U1}{V_{b\_nom}} \right)^2 - COP \cdot U1 + U5 \cdot U3 \right) \\ \dot{T}_c = \frac{1}{m_{air,cabin} \cdot c_{p,air\_cabin}} \cdot (U4 - U5 \cdot U3 + U6) \end{cases} \tag{3.1.20}$$

From the prediction model obtained is observable how the P_{comp}, which is the manipulated variable able to control the cooling power of the system, is present only in the T_b state equation. This means that the MPC will regulate the system cooling power depending only on the temperature of the battery, acting on the T_c equation only through the the blower mass flow rate which can not cool the cabin itself if the compressor is not cooling the refrigerant. Therefore, the cabin cooling will depends on the T_b, if the battery is below the target temperature of 30 °C compressor will be kept inactive and cabin will not be cooled. This is the reason for why it was necessary to build two different MPCs; as will be seen in 3.2 the prediction model of the Cabin priority MPC will have the exact opposite feature, guaranteeing an optimal cooling of the passenger compartment.

3.1.2 Cost function

The cost function, as described in section 2.2.3, is used to estimate the total cost of a certain combination of MVs and the aim of the QP optimization is to minimize it, that is to say finding the MVs sequence which produce the minimal cost.
For this controller, with priority on battery temperature, the quantities which will impact on the cost are: the states deviation from targets, MVs (compressor power, ptc heater power and blower mass flow rate) and MVs rate, therefore cost function is formulated as follow:

$$
\begin{aligned}
J(z_k) = {} & \sum_{i=1}^{p} \{ \{ \frac{w_i^{T_b}}{s^{T_b}} [T_{\text{b,target}}(k+i|k) - T_b(k+i|k)] \}^2 + \{ \frac{w_i^{T_c}}{s^{T_c}} [T_{\text{target, HVAC}}(k+i|k) - \\
& + T_c(k+i|k)] \}^2 \} + \sum_{i=0}^{p-1} \{ \{ \frac{w_i^{P_{\text{comp}}}}{s^{P_{\text{comp}}}} [P_{\text{comp}}(k+i|k) - P_{\text{comp-target}}(k+i|k)] \}^2 + \\
& + \{ \frac{w_i^{P_{\text{ptc}}}}{s^{P_{\text{ptc}}}} [P_{\text{ptc}}(k+i|k) - P_{\text{ptc-target}}(k+i|k)] \}^2 + \{ \frac{w_i^{\dot{m}_{\text{blower}}}}{s^{\dot{m}_{\text{blower}}}} [\dot{m}_{\text{blower}}(k+i|k) - \\
& - \dot{m}_{\text{blower- target}}(k+i|k)] \}^2 \} + \sum_{i=0}^{p-1} \{ \{ \frac{w_i^{\Delta P_{\text{comp}}}}{s^{\Delta P_{\text{comp}}}} [P_{\text{comp}}(k+i|k) - \\
& - P_{\text{comp}}(k+i-1|k)] \}^2 + \{ \frac{w_i^{\Delta P_{\text{ptc}}}}{s^{\Delta P_{\text{ptc}}}} [P_{\text{ptc}}(k+i|k) - P_{\text{ptc}}(k+i-1|k)] \}^2 + \\
& + \{ \frac{w_i^{\Delta \dot{m}_{\text{blower}}}}{s^{\Delta \dot{m}_{\text{blower}}}} [\dot{m}_{\text{blower}}(k+i|k) - \dot{m}_{\text{blower}}(k+i-1|k)] \}^2 \} + \rho_\varepsilon \varepsilon_k^2
\end{aligned}
\tag{3.1.21}
$$

Scale factor values reported in table 2 are non-dimensional, set taking into account the range of values a variable can assume, so by doing the difference between it's maximum and minimum possible value. As an example we can take the cabin temperature, the HVAC is set to work with set temperatures from 16 to 30 Celsius degrees in the cabin, therefore scale factor for it will be 14.

Scale factor	**value**
s^{T_b}	10
s^{T_c}	14
$s^{P_{\text{comp}}}$	6000
$s^{P_{\text{ptc}}}$	2500
$s^{\dot{m}_{\text{blower}}}$	0.15

Table 2: Scale factors battery priority model

Constraints of the optimization problem are reported in equation 3.1.22, they are meant to ensure that the solution of the QP problem produce both feasible and realistic results. About the states, we want them to stay in the operating range, but even when they go out of the range the situation is still feasible; for this reason the constraints are made soft to penalize the cost function when the temperature range is exceeded but still allowing one to produce a feasible result. On the contrary for the MVs the constraints are made hard, this because is not possible to have feasible solution out of the physical operating range of compressor, ptc heater and blower.

Constraint relaxation factors	**value**
$V_{min}^{T_b}/V_{max}^{T_b}$	1
$V_{min}^{T_c}/V_{max}^{T_c}$	1
$V_{min}^{P_{comp}}/V_{max}^{P_{comp}}$	0
$V_{min}^{P_{ptc}}/V_{max}^{P_{ptc}}$	0
$V_{min}^{\dot{m}_{blower}}/V_{max}^{\dot{m}_{blower}}$	0
$V_{min}^{\Delta P_{comp}}/V_{max}^{\Delta P_{comp}}$	0
$V_{min}^{\Delta P_{ptc}}/V_{max}^{\Delta P_{ptc}}$	0
$V_{min}^{\Delta \dot{m}_{blower}}/V_{max}^{\Delta \dot{m}_{blower}}$	0

Table 3: Constraint relaxation battery priority model

$$
\begin{aligned}
&\frac{T_{b,min}(i)}{s^{T_b}} - \varepsilon_k V_{min}^{T_b}(i) \leq \frac{T_b(k+i|k)}{s^{T_b}} \leq \frac{T_{b,max}(i)}{s^{T_b}} + \varepsilon_k V_{max}^{T_b}(i)\\
&\frac{T_{c,min}(i)}{s^{T_c}} - \varepsilon_k V_{min}^{T_c}(i) \leq \frac{T_c(k+i|k)}{s^{T_c}} \leq \frac{T_{c,max}(i)}{s^{T_c}} + \varepsilon_k V_{max}^{T_c}(i)\\
&\frac{P_{comp,min}(i)}{s^{P_{comp}}} - \varepsilon_k V_{min}^{P_{comp}}(i) \leq \frac{P_{comp}(k+i-1|k)}{s^{P_{comp}}} \leq \frac{P_{comp,max}(i)}{s^{P_{comp}}} + \varepsilon_k V_{max}^{P_{comp}}(i)\\
&\frac{P_{ptc,min}(i)}{s^{P_{ptc}}} - \varepsilon_k V_{min}^{P_{ptc}}(i) \leq \frac{P_{ptc}(k+i-1|k)}{s^{P_{ptc}}} \leq \frac{P_{ptc,max}(i)}{s^{P_{ptc}}} + \varepsilon_k V_{max}^{P_{ptc}}(i)\\
&\frac{\dot{m}_{blower,min}(i)}{s^{\dot{m}_{blower}}} - \varepsilon_k V_{min}^{\dot{m}_{blower}}(i) \leq \frac{\dot{m}_{blower}(k+i-1|k)}{s^{\dot{m}_{blower}}} \leq \frac{\dot{m}_{blower,max}(i)}{s^{\dot{m}_{blower}}} + \varepsilon_k V_{max}^{\dot{m}_{blower}}(i)\\
&\frac{\Delta P_{comp,min}(i)}{s^{\Delta P_{comp}}} - \varepsilon_k V_{min}^{\Delta P_{comp}}(i) \leq \frac{\Delta P_{comp}(k+i-1|k)}{s^{\Delta P_{comp}}} \leq \frac{\Delta P_{comp,max}(i)}{s^{\Delta P_{comp}}} + \varepsilon_k V_{max}^{\Delta P_{comp}}(i)\\
&\frac{\Delta P_{ptc,min}(i)}{s^{\Delta P_{ptc}}} - \varepsilon_k V_{min}^{\Delta P_{ptc}}(i) \leq \frac{\Delta P_{ptc}(k+i-1|k)}{s^{\Delta P_{ptc}}} \leq \frac{\Delta P_{ptc,max}(i)}{s^{\Delta P_{ptc}}} + \varepsilon_k V_{max}^{\Delta P_{ptc}}(i)\\
&\frac{\Delta \dot{m}_{blower,min}(i)}{s^{\Delta \dot{m}_{blower}}} - \varepsilon_k V_{min}^{\Delta \dot{m}_{blower}}(i) \leq \frac{\Delta \dot{m}_{blower}(k+i-1|k)}{s^{\Delta \dot{m}_{blower}}} \leq \frac{\Delta \dot{m}_{blower,max}(i)}{s^{\Delta \dot{m}_{blower}}} + \varepsilon_k V_{max}^{\Delta \dot{m}_{blower}}(i)
\end{aligned}
\tag{3.1.22}
$$

with:

$$i = 1 : p$$

3.1.3 Tuning weights

Weights in the cost function are used with the purpose to decide how much a certain deviation of MVs or MDs from the target values impact on the cost of the function. The tuning of this weights is necessary to ensure a correct functioning of the MPC, it can be done prioritizing the target states achievement or the MVs usage so as consequence the energy consumption. In this case it was done taking into consideration both, therefore a trade-off between the two was done.

Weight	**value**
$w_i^{T_b}$	1
$w_i^{T_c}$	3
$w_i^{P_{\text{comp}}}$	0.1
$w_i^{P_{\text{ptc}}}$	0.1
$w_i^{\dot{m}_{\text{blower}}}$	0.3
$w_i^{\Delta P_{\text{comp}}}$	0.1
$w_i^{\Delta P_{\text{ptc}}}$	0.1
$w_i^{\Delta \dot{m}_{\text{blower}}}$	0.1

Table 4: Tuning weight battery priority model

Suggested values for the weights by MATLAB documentation [26], are 1 for the state variables and 0.1 for the MVs and MVs rate. Here since the MPC doesn't have the control of the cabin temperature trough the compressor, which is the command that allow to regulate the cooling power,the weight on cabin temperature deviation was increased to 3 to incentivize the MPC to use more blower flow rate which is the only variable through which it can regulate the cabin temperature. Hence an acceptable manage of the cabin temperature can be achieved in this way.
Over this also the weight of the blower usage was increased from 0.1 to 0.3, this was done for comfort reasons, because since now the MPC is much more incentivize to use the blower flow rate, it can happens that the blower goes up and down at very high rpm in the cabin, and this can be a boring and a discomfortable feature for passengers. So increasing that values blower is managed in a much more comfortable way.

3.2. Cabin priority

Cabin priority control is organized in two sub-blocks:

- Compressor, battery pump and ptc control: governed by the AMPC
- Blower control: managed by the original PI control described in Chapter 2.

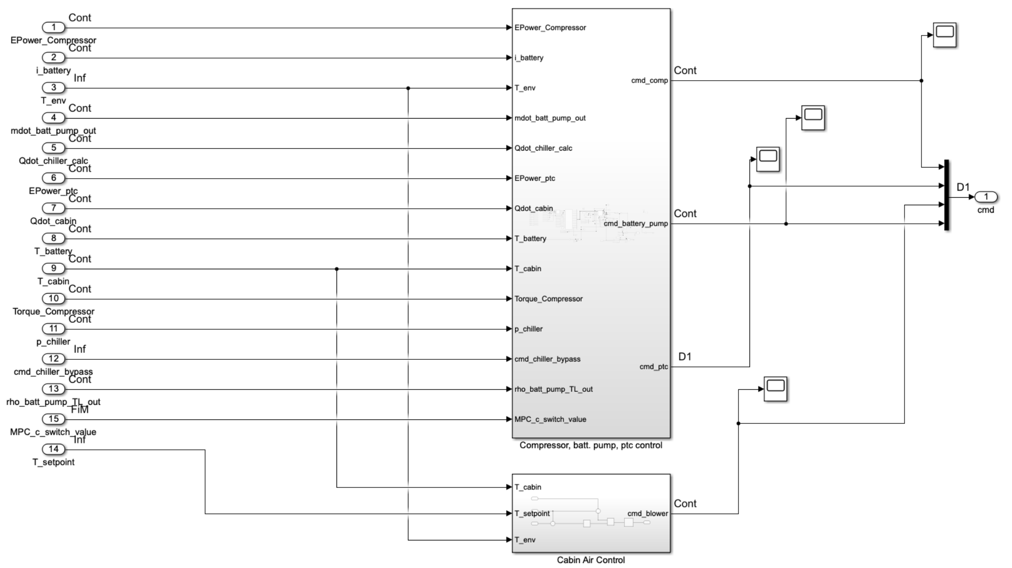

Figure 47: Cabin priority control detail

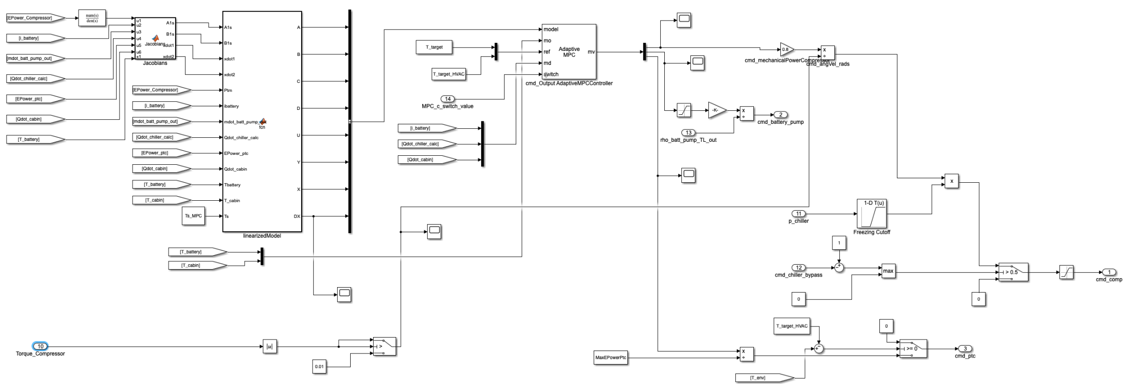

Figure 48: Cabin priority control AMPC overview

Detail of the first cited block are shown in figure 48, where main sections are:

- **Jacobian**, **Linearized model**, **Adaptive MPC block**: these blocks are the same described before, take reference to section 3.1.

- **MPC's MV's**:
  - Compressor control: this command is used by the MPC to control the compressor, it exit from the MPC as the electrical power desired and is converted to rpm to be the input of the compressor as dscribed in section 3.1.
  - Battery pump control: as it was for the blower command in the battery priority control, also here the MPC turn as output the mass flow rate $[kg/s]$. The battery pump need as input the reference fraction (from 0 to 1) related to the maximum rpm speed, therefore the MPC output need to be converted. Considering that:

$$\dot{m}_{b,pump}\left[\frac{kg}{s}\right] = pump\_volumetric\_dispacement\left[\frac{l}{rev}\right]\cdot\frac{rpm}{60}\cdot\rho\left[\frac{kg}{m^3}\right] \tag{3.2.1}$$

is then possible to obtain the reference rpm speed:

$$rpm = \frac{\dot{m}_{b,pump}\left[\frac{kg}{s}\right]\cdot 60}{pump\_volumetric\_dispacement\left[\frac{l}{rev}\right]\cdot\rho\left[\frac{kg}{m^3}\right]} \tag{3.2.2}$$

finally dividing it by the the maximum pump rpm we can obtain the desired command:

$$cmd\_battery\_pump = \frac{rpm}{pump\_max\_rpm} \tag{3.2.3}$$

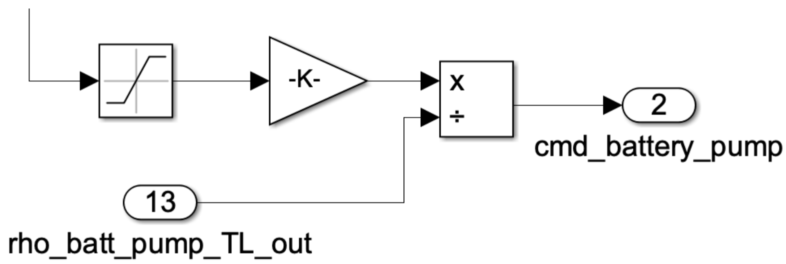

Figure 49: Cabin priority control, battery pump command

  - Ptc command: this command is managed in the smae way it was done for the battery priority controller described in section 3.1.

3.2.1 Prediction model

As it was for the battery priority model also here the MPC has to manage both the BTM and the HVAC, the approach used is to start from the first thermodynamics' law neglecting the convective heat dispersion/absorption from the external environment for simplicity. State to predict are the same two also in this case:

- **Battery temperature**: estimation of the battery temperature is always made taking into account the battery cooling loop, starting equation in the same as before. Hence from equation 3.1.10 is possible to obtain the first state equation:

$$\dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}} \right)^2 - \dot{Q}_{chiller} \right) \tag{3.2.4}$$

with:

$$\dot{Q}_{chiller} = \dot{m}_{b,pump} \cdot c_{p,coolant} \cdot (T_{batt} - T_{coolant\_chiller\_out}) \tag{3.2.5}$$

- **Cabin temperature**: with this controller we want to give priority to the cabin, therefore the target is to have the MV of P_{comp} in the cabin temperature prediction equation, because this is the variable governing the refrigerant power of the system, so having it in the equation allow to properly control the related state (T_c in this case).
Taking into account the HVAC loop and substituting equation 3.1.13 in equation 3.1.16 is possible to obtain the cabin state equation:

$$\dot{T}_c = \frac{1}{m_{air,cabin} \cdot c_{p,air\_cabin}} \cdot (\dot{Q}_{ptc} + \dot{Q}_{cabin} + \dot{Q}_{chiller} - COP \cdot P_{comp}) \tag{3.2.6}$$

Prediction model inside cabin priority MPC is:

$$\begin{cases} \dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}} \right)^2 - \dot{Q}_{chiller} \right) \\ \dot{T}_c = \frac{1}{m_{air,cabin} \cdot c_{p,air\_cabin}} \cdot (\dot{Q}_{ptc} + \dot{Q}_{cabin} + \dot{Q}_{chiller} - COP \cdot P_{comp}) \end{cases} \tag{3.2.7}$$

The variables on which the controller can act (MVs) are the compressor power, battery pump mass flow rate and ptc heater, while quantities measured by sensors (MDs) are battery current used for traction, the chiller heat exchange divided by the battery pump mass flow rate and the cabin heat exchange. COP is function of P_{comp}, while R_0 depends on the battery temperature. All other quantities are considered constant and set when the MPC is initialized.

	variable	**new name**
MVs	P_{comp}	u1
	$\dot{m}_{b,pump}$	u3
	$\dot{Q}_{ptc}$	u5
MDs	$i_{b\_traction} + \frac{P_{comp}}{V_{b\_nom}}$	u2
	$c_{p,coolant} \cdot (T_{batt} - T_{coolant\_chiller\_out})$	u4
	$\dot{Q}_{cabin}$	u6

Table 5: MVs and MDs cabin priority model

Finally prediction model 3.2.7 can be rewritten as:

$$\begin{cases} \dot{T}_b = \frac{1}{m_b \cdot c_{p,b\_cell}} \cdot \left(a \cdot R_0 \cdot \left(u2 + \frac{u1}{V_{b\_nom}} \right)^2 - u3 \cdot u4 \right) \\ \dot{T}_c = \frac{1}{m_{air,cabin} \cdot c_{p,air\_cabin}} \cdot (u5 + u6 + u3 \cdot u4 - COP \cdot u1) \end{cases} \tag{3.2.8}$$

3.2.2 Cost function

Used to estimate the total cost of the optimal combination of MVs calculated by the QP optimization problem, the cost function is defined in the following according to section 2.2.3:

$$\begin{aligned} J(z_k) = \sum_{i=1}^{p} \{ \{ \frac{w_i^{T_b}}{s^{T_b}} [T_{\text{b,target}}(k+i|k) - T_b(k+i|k)] \}^2 + \{ \frac{w_i^{T_c}}{s^{T_c}} [T_{\text{target, HVAC}}(k+i|k) - \\ + T_c(k+i|k)] \}^2 \} + \sum_{i=0}^{p-1} \{ \{ \frac{w_i^{P_{\text{comp}}}}{s^{P_{\text{comp}}}} [P_{\text{comp}}(k+i|k) - P_{\text{comp-target}}(k+i|k)] \}^2 + \\ + \{ \frac{w_i^{\dot{m}_{\text{b,pump}}}}{s^{\dot{m}_{\text{b,pump}}}} [\dot{m}_{\text{b,pump}}(k+i|k) - \dot{m}_{\text{b,pump-target}}(k+i|k)] \}^2 + \{ \frac{w_i^{P_{\text{ptc}}}}{s^{P_{\text{ptc}}}} [P_{\text{ptc}}(k+i|k) - \\ - P_{\text{ptc-target}}(k+i|k)] \}^2 \} + \sum_{i=0}^{p-1} \{ \{ \frac{w_i^{\Delta P_{\text{comp}}}}{s^{\Delta P_{\text{comp}}}} [P_{\text{comp}}(k+i|k) - P_{\text{comp}}(k+i-1|k)] \}^2 + \\ + \{ \frac{w_i^{\Delta \dot{m}_{\text{b,pump}}}}{s^{\Delta \dot{m}_{\text{b,pump}}}} [\dot{m}_{\text{b,pump}}(k+i|k) - \dot{m}_{\text{b,pump}}(k+i-1|k)] \}^2 + \{ \frac{w_i^{\Delta P_{\text{ptc}}}}{s^{\Delta P_{\text{ptc}}}} [P_{\text{ptc}}(k+i|k) - \\ - P_{\text{ptc}}(k+i-1|k)] \}^2 \} + \rho_\varepsilon \varepsilon_k^2 \end{aligned} \tag{3.2.9}$$

Scale factor values for the cabin priority model are reported in the following table:

Scale factor	**value**
s^{T_b}	10
s^{T_c}	14
$s^{P_{\text{comp}}}$	6000
$s^{\dot{m}_{\text{b,pump}}}$	0.355
$s^{P_{\text{ptc}}}$	2500

Table 6: Scale factors cabin priority model

Constraint of the optimization problem are reported in equation 3.2.10, to ensure that the QP optimization problem will produce both feasible and realistic results. Relaxation of the constraint is treated in the same way as it was done for battery priority model, therefore make reference to table 3.

$$
\begin{aligned}
&\frac{T_{b,min}(i)}{s^{T_b}} - \varepsilon_k V_{min}^{T_b}(i) \le \frac{T_b(k+i|k)}{s^{T_b}} \le \frac{T_{b,max}(i)}{s^{T_b}} + \varepsilon_k V_{max}^{T_b}(i) \\
&\frac{T_{c,min}(i)}{s^{T_c}} - \varepsilon_k V_{min}^{T_c}(i) \le \frac{T_c(k+i|k)}{s^{T_c}} \le \frac{T_{c,max}(i)}{s^{T_c}} + \varepsilon_k V_{max}^{T_c}(i) \\
&\frac{P_{comp,min}(i)}{s^{P_{comp}}} - \varepsilon_k V_{min}^{P_{comp}}(i) \le \frac{P_{comp}(k+i-1|k)}{s^{P_{comp}}} \le \frac{P_{comp,max}(i)}{s^{P_{comp}}} + \varepsilon_k V_{max}^{P_{comp}}(i) \\
&\frac{\dot{m}_{b,pump,min}(i)}{s^{\dot{m}_{b,pump}}} - \varepsilon_k V_{min}^{\dot{m}_{b,pump}}(i) \le \frac{\dot{m}_{b,pump}(k+i-1|k)}{s^{\dot{m}_{b,pump}}} \le \frac{\dot{m}_{b,pump,max}(i)}{s^{\dot{m}_{b,pump}}} + \varepsilon_k V_{max}^{\dot{m}_{b,pump}}(i) \\
&\frac{P_{ptc,min}(i)}{s^{P_{ptc}}} - \varepsilon_k V_{min}^{P_{ptc}}(i) \le \frac{P_{ptc}(k+i-1|k)}{s^{P_{ptc}}} \le \frac{P_{ptc,max}(i)}{s^{P_{ptc}}} + \varepsilon_k V_{max}^{P_{ptc}}(i) \\
&\frac{\Delta P_{comp,min}(i)}{s^{\Delta P_{comp}}} - \varepsilon_k V_{min}^{\Delta P_{comp}}(i) \le \frac{\Delta P_{comp}(k+i-1|k)}{s^{\Delta P_{comp}}} \le \frac{\Delta P_{comp,max}(i)}{s^{\Delta P_{comp}}} + \varepsilon_k V_{max}^{\Delta P_{comp}}(i) \\
&\frac{\Delta \dot{m}_{b,pump,min}(i)}{s^{\Delta \dot{m}_{b,pump}}} - \varepsilon_k V_{min}^{\Delta \dot{m}_{b,pump}}(i) \le \frac{\Delta \dot{m}_{b,pump}(k+i-1|k)}{s^{\Delta \dot{m}_{b,pump}}} \le \frac{\Delta \dot{m}_{b,pump,max}(i)}{s^{\Delta \dot{m}_{b,pump}}} + \varepsilon_k V_{max}^{\Delta \dot{m}_{b,pump}}(i) \\
&\frac{\Delta P_{ptc,min}(i)}{s^{\Delta P_{ptc}}} - \varepsilon_k V_{min}^{\Delta P_{ptc}}(i) \le \frac{\Delta P_{ptc}(k+i-1|k)}{s^{\Delta P_{ptc}}} \le \frac{\Delta P_{ptc,max}(i)}{s^{\Delta P_{ptc}}} + \varepsilon_k V_{max}^{\Delta P_{ptc}}(i)
\end{aligned}
\tag{3.2.10}
$$

with:

$$i = 1 : p$$

3.2.3 Tuning weights

Also in this case as it was done for the battery priority model the weights were tuned by doing a trade-off between target precision tracking and energy consumption.

Weight	**value**
$w_i^{T_b}$	1.8
$w_i^{T_c}$	0.9
$w_i^{P_\text{comp}}$	2
$w_i^{\dot{m}_\text{b,pump}}$	0.01
$w_i^{P_\text{ptc}}$	0.1
$w_i^{\Delta P_\text{comp}}$	0.1
$w_i^{\Delta \dot{m}_\text{b,pump}}$	0.01
$w_i^{\Delta P_\text{ptc}}$	0.1

Table 7: Tuning weight cabin priority model

In this case the weight on the battery temperature tracking was increased from 1 (default value) to 1.8, because in this model the cooling power is managed from the cabin temperature equation, therefore the MPC in its prediction can not use the compressor power to cool the battery, so by increasing $w_i^{T_b}$ is incentivized the usage of the battery pump, which is the only variable MPC can use to manage battery temperature. $w_i^{T_c}$ was decreased from 1 to 0.9 to reduce a bit the usage of the compressor which is the most power demanding component of the BTM, but still guaranteeing a good tracking of the cabin temperature target. To reduce the compressor usage was also increased $w_i^{P_{\text{comp}}}$ from 0.1 to 2, this allowed a much better energy optimization. The reason for why it was increased the weight on the compressor power instead of further reduce the weight on cabin temperature deviation was to avoid excessive temperature oscillations in the cabin, promoting the passenger comfort (as will be shown in chapter 4). To conclude also the weight on battery pump power and power rate were decreased from 0.1 to 0.01 to use more the latter component which allows to better exploit the cooling power of the refrigerant circuit to cool the battery and at the same time has a very low impact on power consumption and no impact on passengers comfort.

3.3. Selection logic

Selection logic block showed in figure 50, has the purpose to manage the selection between the two controller producing the following three outputs:

- **selection command**: this command is responsible for the selection of the controller to use depending on the situation. First of all is observable that through the command $force\_sim$, which is set in the MATLAB file explained in section 3.5, the system can be forced to run with a single controller. In case the latter is set to zero, the switching will occur according to the following logic: the current battery temperature is given as input to a relay block which generate as output one when temperature overlap 35 Celsius degree and 0 when it return below 33 Celsius degree. The margin of two degrees is taken to avoid continue switching in certain operating conditions. To conclude, when output is one battery priority outputs are used while relay output is 0 cabin priority ones are used.

- **MPC b switch value**: this command is used to activate (0) or deactivate (1) the QP optimization on the battery priority AMPC block described in section 3.1. It depends from *selection_command*, in particular it is always the opposite of it therefore a boolean NOT block is used.

- **MPC c switch value**: this command is used to activate (0) or deactivate (1) the QP optimization on the battery priority AMPC block, and has the same value of the *selection_command*.

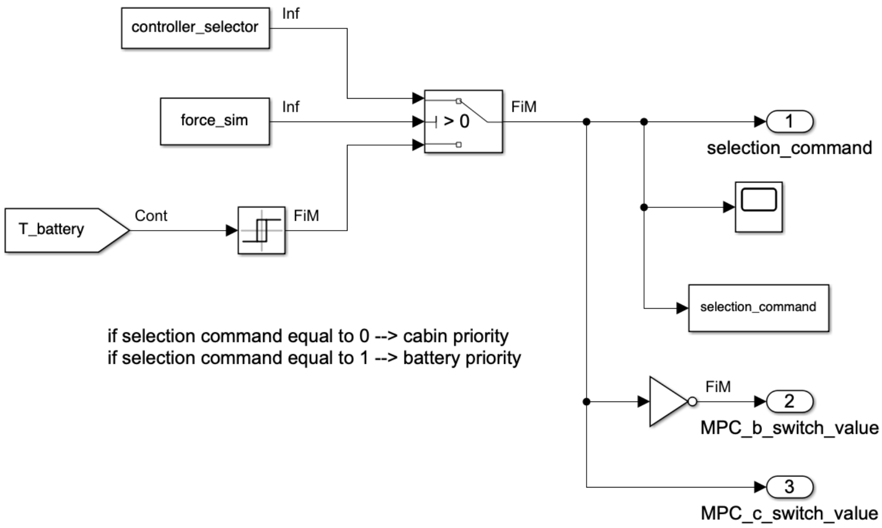

Figure 50: Selection logic block

3.4. COP estimation

COP is used in the prediction models 3.1.1 and 3.2.7. The coefficient of performance of a refrigerant cycle is a key parameter able to show if the refrigerant cycle is working efficiently or not.
Main parameters which influence the COP, considering that the refrigerant remains always the r134a, are the environmental temperature and the compressor power. The environmental temperature directly influences the condensation temperature in the condenser. For example if T_{env} increase, condensation temperature will also increase, this means that more work is needed to dissipate the heat and by consequence more work from the compressor is needed, decreasing the COP. In the initial model [25] the COP was calculated as function of the temperature as with the following analytical expression:

$$COP = -0.0699 \cdot T_{env} + 4.5754 \tag{3.4.1}$$

But considering that the simulation where done with a constant environmental temperature of 26 °C it was decided to model the COP as function of P_{comp}, this also allowed the have one less MD in the prediction model, simplifying the calculations. The process used to find an analytical expression of the COP as function of P_{comp} consists in the following steps:

1. calculating the COP current value:

$$COP = \frac{\dot{Q}_{evap} + \dot{Q}_{chiller}}{P_{comp}} \tag{3.4.2}$$

2. this values are saved in the MATLAB workspace

3. since the COP values were notice to be quite dependent on the P_{comp} rate so by the accelerations present on the test cycle; several simulation on UDDS, WLTC3, AUDC and ARDC test cycles where done to have a bigger data cluster and make the interpolation more reliable

4. by using the MATLAB command $polyfit$ it was find the best polynomial which approximate the data cluster

It was find that the best approximation were achieved by the following second degree polynomial:

$$COP = 5.3607 \cdot 10^{-7} \cdot P_{comp}^2 - 0.0031 \cdot P_{comp} + 6.1878 \tag{3.4.3}$$

But after several simulations was noticed that the product $COP \cdot P_{comp}$ which represent the cooling power of the system in the prediction models 3.1.1 and 3.2.7 gives as a result a third degree polynomial with a low steepness between 1000 and 2500 W (figure 51), for this reason this range of power was never used by the MPC because an increase in P_{comp} didn't produce a considerable gain in the cooling power to justify the cost.
It was then decided to use a first degree polynomial (equation 3.4.4) which allows to have a gradual gain in the cooling power along all the P_{comp} interval (figure 51) allowing the MPC to use all that interval having a gain in the cooling power for each gain of cost due to the increment of P_{comp}.

$$COP = -0.00004 \cdot P_{comp} + 3.1797 \tag{3.4.4}$$

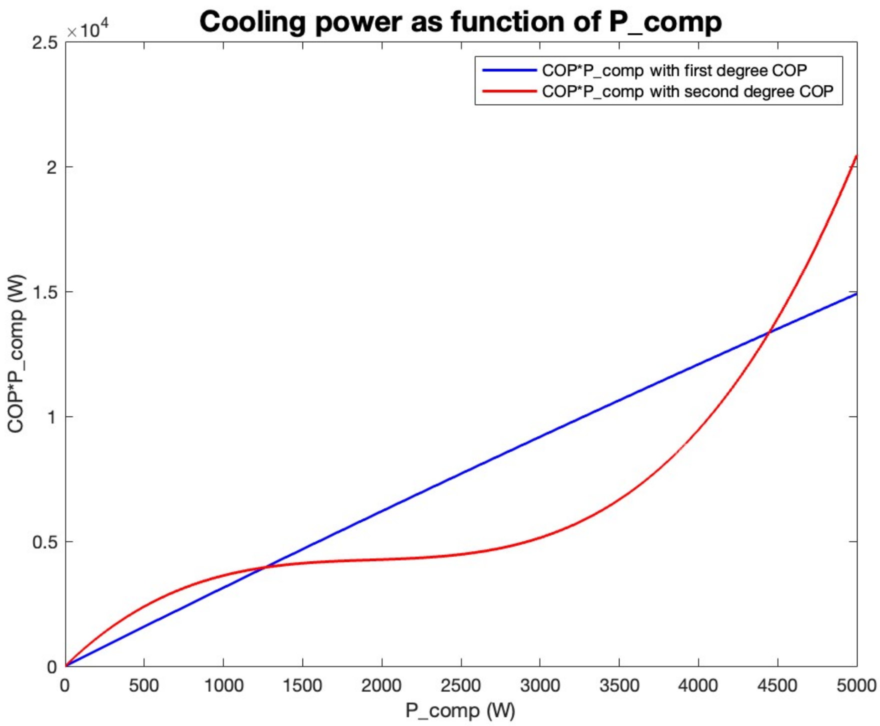

Figure 51: Cooling power as function of P_{comp}

3.5. MPC settings

The simulations and the MPC are set from a MATLAB file used to define initial conditions, driving cycle to simulate, type of control to use and the MPC object. In the following is shown the code to set the user inputs:

Listing 3.1: User input configuration

```
%% User input
% Environment temperature
T_init = 26;
% Driving cycle
cycle = 'WLTC3';

% Target battery temperature
T_target = 30; % C
% Target cabin temperature
T_target_HVAC = 23; % C

% Strategy (AMPC or Reactive)
strategy = 'AMPC';
model_name = append('ElectricVehicleThermalManagement_',
    strategy);
open(model_name)

% Sample time for "To Workspace"
Tsample = 0.1; % set -1 to not define it
```

First of all initial environmental temperature for the simulation is set, then is selected the test cycle to be used (in the example WLTC3). Depending on the cycle to be used the battery current are calculated as described in 1.1. After this the target temperature to be tracked by the controller are defined, both for battery and cabin temperature which are the two states of the prediction models.
Control strategy is then defined choosing between the developed AMPC or original Reactive logic. The $Tsample$ is used to select sample time to be used to collect data from Simscape sensors and report it in MATLAB workspace for the post processing.

3.5.1 Cabin priority control settings

As seen in section 3.3, if the control strategy is AMPC, it can be chosen to use the default switching logic between cabin and battery priority controls or it can be forced one of the two controller with the two variables defined:

Listing 3.2: AMPC forcing logic

```
if strcmp(strategy,'AMPC')
    % force cabin or battery priority mode
    force_sim = 0; % if 1 you force the simulation if 0 it
        switch automatically between cabin/battery priority
        mode
```

```
    controller_selector = 1; % if 0 cabin priority if 1
       battery priority
```

Is now used the symbolic toolbox of MATLAB that allow to create scalar symbolic variables, in our case used to create the state, output and input variables. Remember from 3.1.20 and 3.2.7, that states are two for each prediction model, output are equal to the states and input variables are 6 (three MVs and three MDs). The state defined are four, this because each prediction model have the same two states but described by two different equations, therefore are necessary two symbolic variables for battery temperature and two for cabin's one.

Listing 3.3: Symbolic variables definition and Jacobian matrices for battery priority control

```
    %% Write symbolic jacobian matrices, which means write the
        initial linearized model for Cabin Priority
    % Define states, outputs and inputs
    syms x [4 1] % States (x1 = battery temperature cabin_p,
       x2 = cabin temperature cabin_p, x3 = battery
       temperature batt_p, x4 = cabin temperature batt_p)
    syms y [2 1] % Output (y1 = battery temperature cabin_p,
       y2 = cabin temperature cabin_p)
    syms u [6 1] % Input (u1 = compressor power, u2 = current,
        u3 = mdot_batt_pump, u4 = Qdot_chiller, u5 = ptc_power
       , u6 = Qdot_cabin)

    %syms
    syms R0 c_1 % Define internal resistance R0 and
       coefficient of performance COP
    R0 = poly2sym(r_pol_coef,x1); % Cell R0 = f(battery
       temperature);
    c_1 = -0.00004*u1+3.1797;

    % States derivative
    syms xdot [4 1] % States derivatives

    % Specific heat and cabin air mass
    m = 2.7673*10^(-4); % 1/(m_air_cab*cp)
    COP_C = c_1; % COP value

    % Cabin priority prediction model
    xdot1 = (Joule_incr_factor*battery_N_cells_per_module*
       battery_N_modules*R0*(u2+u1/battery_V_nom)^2 ...
    - u3*u4) / (battery_mass*battery_cell_cp);
    xdot2 = (m)*(u6+u5+u3*u4-COP_C*u1);

    xdot_C = [xdot1; xdot2];

    % Output
```

```
    y1 = x1;
    y2 = x2;

    % Jacobians Cabin
    A1s = [diff(xdot1,x1) diff(xdot1,x2);
           diff(xdot2,x1) diff(xdot2,x2)];
    B1s = [diff(xdot1,u1) diff(xdot1,u2) diff(xdot1,u3) diff(xdot1,u4) diff(xdot1,u5) diff(xdot1,u6);
           diff(xdot2,u1) diff(xdot2,u2) diff(xdot2,u3) diff(xdot2,u4) diff(xdot2,u6) diff(xdot2,u6)];
    C1s = [diff(y1,x1) diff(y1,x2);
           diff(y2,x1) diff(y2,x2)];
    D1s = [diff(y1,u1) diff(y1,u2) diff(y1,u3) diff(y1,u4) diff(y1,u5) diff(y1,u6);
           diff(y2,u1) diff(y2,u2) diff(y2,u3) diff(y2,u4) diff(y2,u5) diff(y2,u6)];
```

Internal battery cell resistance is find fitting datas from a look up table as function of battery temperature. COP is defined as in section 3.4 and the prediction model is written. Symbolic Jacobian matrices are defined with the initial conditions shown in the next listing.
As said in section 3.1 the function *matlabFunctionBlock* is used to create a usable Simulink MATLAB function to create the Jacobians linearized matrices, battery cell internal resistance and COP as function of P_{comp} at each time instant.

Listing 3.4: Symbolic MATLAB function generation

```
    %% Symbolic function generation
    % Creation of Jacobian, R0, and COP functions in Simulink ( block "Jacobians" (Compressor control subsystem), "R0" (interal model subsystem), "c" (interal model subsystem))
    matlabFunctionBlock(append(model_name,'/Controls/Cabin Priority/Compressor, batt. pump, ptc control/Jacobians'),A1s,B1s,xdot1,xdot2)
    matlabFunctionBlock(append(model_name,'/Internal Model AMPC/R0'),R0)
    matlabFunctionBlock(append(model_name,'/Internal Model AMPC/c'),c_1)
```

Next step is to create the MPC object based on the discrete-time prediction model, which is inserted in the Simulink AMPC block to set the controller. First of all are defined nominal initial linearization point:

Listing 3.5: Nominal linearization point cabin priority

```
    %% Creation of mpcobj
    % Write the initial condition x0 u0 y0
    x0 = [T_battery_init; cabin_T_init]; % look up file "ElectricVehicleManagementParamenters" initial condition block
```

```
    u1_0 = [EPowerCompressor_init; i_battery_init;
       mdot_pump_init; Qdot_chiller_init; EPower_ptc_init;
       Qdot_cabin_init];
    y0 = [T_battery_init; cabin_T_init];

    % Define the nominal initial point
    x1 = sym(x0(1));
    x2 = sym(x0(2));
    u1 = sym(u1_0(1));
    u2 = sym(u1_0(2));
    u3 = sym(u1_0(3));
    u4 = sym(u1_0(4));
    u5 = sym(u1_0(5));
    u6 = sym(u1_0(6));

    % Matrixes to create the plant model Cabin Priority
    A1 = double(subs(A1s));
    B1 = double(subs(B1s));
    C1 = double(subs(C1s));
    D1 = double(subs(D1s));
```

MPC object structure in now defined, declaring which input is a MV and which a MD, with related names, then are also set the initial condition previously defined:

Listing 3.6: MPC object structure cabin priority

```
    % Continuous and discrete time plant models Cabin Priority
    plantCT1 = ss(A1,B1,C1,D1);
    plantDT1 = c2d(plantCT1,Ts_MPC);
    % Set disturbances, manipulated variables, output
    plantDT1.InputGroup.MeasuredDisturbances = [2 4 6];
    plantDT1.InputGroup.ManipulatedVariables = [1 3 5];
    plantDT1.OutputGroup.Measured = [1 2];
    plantDT1.StateName = {'Tbattery','Tcabin'};
    plantDT1.InputName = {'Pbtm','iB','mdot_pump','
       Qdot_chiller','Pptc','Qdot_cabin'};
    plantDT1.OutputName = {'Tbattery','Tcabin'};

    % Create MPC object
    mpcobj1 = mpc(plantDT1,Ts_MPC,p,pc);

    % Nominal starting point
    mpcobj1.Model.Nominal = struct('X', x0, 'U', u1_0, 'Y', y0
       , 'DX', [0;0]);
```

Now are defined the constraint for the QP optimization problem. Taking as example the battery state constraint and compressor the correspondences with equation 3.2.10 are:

Variable	Correspondence	Value
$mpcobj1.OV(1).Min$	$T_{b,min}(i)$	25 °C
$mpcobj1.OV(1).Max$	$T_{b,max}(i)$	35 °C
$mpcobj1.OV(1).MinECR$	$V_{min}^{T_b}(i)$	1
$mpcobj1.OV(1).MaxECR$	$V_{max}^{T_b}(i)$	1

Table 8: QP constrain cabin priority Battery state

Variable	Correspondence	Value [W]
$mpcobj1.MV(1).Max$	$P_{comp,min}(i)$	0
$mpcobj1.MV(1).Min$	$P_{comp,max}(i)$	6000
$mpcobj1.MV(1).RateMin$	$\Delta P_{comp,min}(i)$	-500
$mpcobj1.MV(1).RateMax$	$\Delta P_{comp,max}(i)$	500

Table 9: QP constrain cabin priority Compressor

Same logic is followed by the Cabin temperature state and the other MDs, furthermore as said in section 3.1.2 the constraints are made hard for MVs because the defined range is the physical possible operating range for the components so it can not be overlapped, therefore $mpcobj1.MV.MinECR$ is not modified since MATLAB puts 0 as default value.

Listing 3.7: QP constraint definition for cabin priority control

```
    % Output variable
    % Batt temperature
    mpcobj1.OV(1).Min = min_batt_T;
    mpcobj1.OV(1).MinECR = 1; % 0 is hard constraint, the
       higher the softer the constraint
    mpcobj1.OV(1).Max = max_batt_T;
    mpcobj1.OV(1).MaxECR = 1; % 0 is hard constraint, the
       higher the softer the constraint
    mpcobj1.OV(1).ScaleFactor = T_batt_scale;

    % Cabin temperature
    mpcobj1.OV(2).Min = min_cabin_T;
    mpcobj1.OV(2).MinECR = 1; % 0 is hard constraint, the
       higher the softer the constraint
    mpcobj1.OV(2).Max = max_cabin_T;
    mpcobj1.OV(2).MaxECR = 1; % 0 is hard constraint, the
       higher the softer the constraint
    mpcobj1.OV(2).ScaleFactor = T_cabin_scale;

    % Control variable
    % Compressor power
```

```
    mpcobj1.MV(1).Max = MaxEPowerCompressor;
    mpcobj1.MV(1).Min = 0;
    mpcobj1.MV(1).ScaleFactor = MaxEPowerCompressor;
    mpcobj1.MV(1).Target = 0; % We want that the compressor
       power is kept minimum
    mpcobj1.MV(1).RateMin = Pcomp_RateMin;
    mpcobj1.MV(1).RateMax = Pcomp_RateMax;

    % mdot_pump
    mpcobj1.MV(2).Max = Max_mdot_pump;
    mpcobj1.MV(2).Min = (Max_mdot_pump/5); % 0.2 of the
       maximum flow rate to guarantee circulation and avoid
       formation of temperature gradient in the coolant
    mpcobj1.MV(2).ScaleFactor = mdot_pump_scale;
    mpcobj1.MV(2).Target = (Max_mdot_pump/5);
    mpcobj1.MV(2).RateMin = mdot_pump_RateMin;
    mpcobj1.MV(2).RateMax = mdot_pump_RateMax;

    % Ptc power
    mpcobj1.MV(3).Max = MaxEPowerPtc;
    mpcobj1.MV(3).Min = 0;
    mpcobj1.MV(3).ScaleFactor = MaxEPowerPtc;
    mpcobj1.MV(3).Target = 0; % We want that the ptc power is
       kept minimum
    mpcobj1.MV(3).RateMin = Pptc_RateMin;
    mpcobj1.MV(3).RateMax = Pptc_RateMax;
```

The scale factors of the MVs have already been set in the code above, and now are also set the scale factors of the MDs.

Listing 3.8: MDs scale factors

```
    % Disturbance variable
    mpcobj1.DV(1).ScaleFactor = current_scale; % current
    mpcobj1.DV(2).ScaleFactor = Qdot_chiller_scale; %
       Qdot_chiller/mdot_pump
    mpcobj1.DV(3).ScaleFactor = Qdot_cabin_scale; % cabin
       convection + heat generated by the passengers
```

The last thing that needs to be set are the weights, they are tuned with the values showed and explained in section 7.

Listing 3.9: Weights setting for Cabin priority control

```
    % Cost function weights
    mpcobj1.Weights.OutputVariables = [WeightTbattery_C*ones(p
       ,1), WeightTcabin_C*ones(p,1)];
    mpcobj1.Weights.ManipulatedVariables = [WeightEPowerComp_C
       *ones(p,1), Weight_mdot_pump_C*ones(p,1),
       WeightEPowerPtc_C*ones(p,1)];
```

```
    mpcobj1.Weights.ManipulatedVariablesRate = [
       WeightEPowerRateComp_C*ones(p,1),
       Weight_mdot_pumpRate_C*ones(p,1), WeightEPowerRatePtc_C
       *ones(p,1)];
    mpcobj1.Weights.ECR = WeightECR_C;
```

3.5.2 Battery priority control settings

For battery priority MPC was done the same as for cabin priority one, therefore, it will not be further commented on, but just the code will be reported. The symbolic variable about the states and the states derivative were already created in the cabin priority part, and this is the reason why they are not declared in the following code. Obviously MVs and DVs are the ones described for the battery priority control in section 3.1.

Listing 3.10: Battery priority control settings

```
%% Write symbolic jacobian matrices, which means write the
    initial linearized model for Battery Priority
% Define states, outputs and inputs
syms Y [2 1] % Output (Y1 = battery temperature batt_p, Y2
    = cabin temperature batt_p)
syms U [6 1] % Input (U1 = compressor power, U2 = current,
    U3 = Qdot_evap, U4 = ptc_power, U5 = mdot_blower, U6 =
    Qdot_cabin)

% COP
c_2 = -0.00004*U1+3.1797;
COP_B = c_2; % COP value

% Battery priority
xdot3 = (Joule_incr_factor*battery_N_cells_per_module*
   battery_N_modules*R0*(U2+U1/battery_V_nom)^2 ...
- COP_B*U1+U5*U3) / (battery_mass*battery_cell_cp);
xdot4 = (m)*(U4-U5*U3+U6);

xdot_B = [xdot3; xdot4];

% Output
Y1 = x3;
Y2 = x4;

% Jacobians Battery
A2s = [diff(xdot3,x3) diff(xdot3,x4);
      diff(xdot4,x3) diff(xdot4,x4)];
B2s = [diff(xdot3,U1) diff(xdot3,U2) diff(xdot3,U3) diff(
   xdot3,U4) diff(xdot3,U5) diff(xdot3,U6);
```

```
          diff(xdot4,U1) diff(xdot4,U2) diff(xdot4,U3) diff(xdot4,U4) diff(xdot4,U5) diff(xdot4,U6)];
    C2s = [diff(Y1,x3) diff(Y1,x4);
          diff(Y2,x3) diff(Y2,x4)];
    D2s = [diff(Y1,U1) diff(Y1,U2) diff(Y1,U3) diff(Y1,U4) diff(Y1,U5) diff(Y1,U6);
          diff(Y2,U1) diff(Y2,U2) diff(Y2,U3) diff(Y2,U4) diff(Y2,U5) diff(Y2,U6)];

    %% Symbolic function generation
    % Creation of Jacobian, R0, and COP functions in Simulink ( block "Jacobians" (Compressor control subsystem),)
    matlabFunctionBlock(append(model_name,'/Controls/Battery Priority/Compressor, ptc and blower Control/Jacobians'),A2s,B2s,xdot3,xdot4)

    %% Creation of mpcobj
    % Write the initial condition x0 u0 y0
    u2_0 = [EPowerCompressor_init; i_battery_init; Qdot_evap_init; EPower_ptc_init; mdot_blower_init; Qdot_cabin_init];

    % Define the nominal initial point
    x3 = sym(x0(1));
    x4 = sym(x0(2));
    U1 = sym(u2_0(1));
    U2 = sym(u2_0(2));
    U3 = sym(u2_0(3));
    U4 = sym(u2_0(4));
    U5 = sym(u2_0(5));
    U6 = sym(u2_0(6));

    % Matrixes to create the plant model Battery Priority
    A2 = double(subs(A2s));
    B2 = double(subs(B2s));
    C2 = double(subs(C2s));
    D2 = double(subs(D2s));

    % Continuous and discrete time plant models Battery Priority
    plantCT2 = ss(A2,B2,C2,D2);
    plantDT2 = c2d(plantCT2,Ts_MPC);
    % Set disturbances, manipulated variables, output
    plantDT2.InputGroup.MeasuredDisturbances = [2 3 6];
    plantDT2.InputGroup.ManipulatedVariables = [1 4 5];
    plantDT2.OutputGroup.Measured = [1 2];
    plantDT2.StateName = {'Tbattery','Tcabin'};
    plantDT2.InputName = {'Pbtm','iB','Qdot_evap','Pptc','
```

```
    mdot_blower','Qdot_cabin'};
plantDT2.OutputName = {'Tbattery','Tcabin'};

% Create MPC object
mpcobj2 = mpc(plantDT2,Ts_MPC,p,pc);

% Nominal starting point
mpcobj2.Model.Nominal = struct('X', x0 , 'U', u2_0, 'Y',
    y0, 'DX', [0;0]);

% Output variable
% Batt temperature
mpcobj2.OV(1).Min = min_batt_T;
mpcobj2.OV(1).MinECR = 1; % 0 is hard constraint, the
    higher the softer the constraint
mpcobj2.OV(1).Max = max_batt_T;
mpcobj2.OV(1).MaxECR = 1; % 0 is hard constraint, the
    higher the softer the constraint
mpcobj2.OV(1).ScaleFactor = T_batt_scale;
% Cabin temperature
mpcobj2.OV(2).Min = min_cabin_T;
mpcobj2.OV(2).MinECR = 1; % 0 is hard constraint, the
    higher the softer the constraint
mpcobj2.OV(2).Max = max_cabin_T;
mpcobj2.OV(2).MaxECR = 1; % 0 is hard constraint, the
    higher the softer the constraint
mpcobj2.OV(2).ScaleFactor = T_cabin_scale;

% Control variable
% Compressor power
mpcobj2.MV(1).Max = MaxEPowerCompressor;
mpcobj2.MV(1).Min = 0;
mpcobj2.MV(1).ScaleFactor = MaxEPowerCompressor;
mpcobj2.MV(1).Target = 0; % We want that the compressor
    power is kept minimum
mpcobj2.MV(1).RateMin = Pcomp_RateMin;
mpcobj2.MV(1).RateMax = Pcomp_RateMax;

% Ptc power
mpcobj2.MV(2).Max = MaxEPowerPtc;
mpcobj2.MV(2).Min = 0;
mpcobj2.MV(2).ScaleFactor = MaxEPowerPtc;
mpcobj2.MV(2).Target = 0; % We want that the ptc power is
    kept minimum
mpcobj2.MV(2).RateMin = Pptc_RateMin;
mpcobj2.MV(2).RateMax = Pptc_RateMax;

% mdot_blower
```

```
    mpcobj2.MV(3).Max = Max_mdot_blower;
    mpcobj2.MV(3).Min = 0;
    mpcobj2.MV(3).ScaleFactor = mdot_blower_scale;
    mpcobj2.MV(3).Target = 0.015;
    mpcobj2.MV(3).RateMin = mdot_blower_RateMin;
    mpcobj2.MV(3).RateMax = mdot_blower_RateMax;

    % Disturbance variable
    mpcobj2.DV(1).ScaleFactor = current_scale; % current
    mpcobj2.DV(2).ScaleFactor = Qdot_evap_scale; % Qdot_evap/
       mdot_blower
    mpcobj2.DV(3).ScaleFactor = Qdot_cabin_scale; % cabin
       convection + heat generated by the passengers

    % Cost function weights
    mpcobj2.Weights.OutputVariables = [WeightTbattery_B*ones(p
       ,1), WeightTcabin_B*ones(p,1)];
    mpcobj2.Weights.ManipulatedVariables = [WeightEPowerComp_B
       *ones(p,1), WeightEPowerPtc_B*ones(p,1),
       Weight_mdot_blower_B*ones(p,1)];
    mpcobj2.Weights.ManipulatedVariablesRate = [
       WeightEPowerRateComp_B*ones(p,1), WeightEPowerRatePtc_B
       *ones(p,1), Weight_mdot_blowerRate_B*ones(p,1)];
    mpcobj2.Weights.ECR = WeightECR_B;
```

3.6. Prediction and control horizon

Prediction and control horizon are key parameters to ensure a correct functioning of the MPC controller. They are related to the number of future time step on which the MPC is going to evaluate the QP problem cost to optimize the MVs.
Usual practice is to set initially the prediction horizon of a length similar to the closed loop system response time. Increasing the prediction horizon weight down the computational effort, but usually increase control accuracy and optimization of the MVs since the controller has a longer vision in the future behavior of the system allowing it to act in advance in the optimal way.
In this case it was necessary to investigate the controller behavior changing the prediction and control horizon, because the used controller is an adaptive MPC controlling a nonlinear system, and the optimization is done around the linearized model at each time step, therefore a longer prediction horizon may produce inaccuracies instead of better performance.
First of all, it was tested if a shorter time step produces advantages in the controller performances, initial T_s was set to 1 second, several tests were performed with 0.1, 0.2, and 0.5 maintaining the same length in time prediction, therefore decreasing T_s p and pc where proportionally increased. What was found is that computational time increased rapidly while the advantage in energy optimization was not that rel-

evant to justify that big spike in computational cost, furthermore the temperature rms error remained very similar changing T_s, so it was decided to keep it at 1 second.

Ts [s]	1	0.5	0.2	0.1
E_comp [kWh]	0.1501	0.1485	0.1478	0.1476
E_BTM [kWh]	0.1605	0.1589	0.1582	0.1580
max Tb [°C]	31.13	31.16	29.78	29.79
Tb end [°C]	29.70	29.76	31.16	31.16
Tb rmse	0.6025	0.6316	0.6415	0.6449
Tcabin rmse	0.0242	0.0239	0.0238	0.0237
comp. time	173	271	595	1070

Table 10: Sample time influence on MPC performance

Then several combinations of p and pc were tested:

p|pc	4|2	7|3	9|4	12|6	20|10	30|15
E_comp [kWh]	0.0652	0.1111	0.1501	0.2131	0.2976	0.3198
E_BTM [kWh]	0.0756	0.1215	0.1605	0.2235	0.3081	0.3302
max Tb [°C]	33.16	31.81	31.13	30.45	29.83	29.76
Tb end [°C]	33.04	31.19	29.70	27.18	24.72	24.21
Tb rmse	2.7528	1.5394	0.6025	1.5018	3.0558	3.3852
Tcabin rmse	0.3604	0.0241	0.0242	0.0364	0.0241	0.0203
comp. time	169	171	173	177	179	187

Table 11: p and pc influence on MPC performance

As observable from the table above is clear how a too small values of p and pc is not able to produce a satisfactory tracking of both the cabin and battery temperatures, even though the rms errors do not seem big we have to consider that the used cycle for this test is the UDDS which is not too demanding on the cooling system; therefore, using a more challenging driving cycle would produce higher rms errors. On the other hand since the temperatures are kept higher the energy demand from the BTM is lower.
Increasing p and pc does not produce always better results, indeed is observable how, after a certain threshold, as much their values are increased as much the MPC work inefficiently, maintaining a good manage of the cabin temperature but cooling too much the battery one, going well below the target of 30 °C, consequently increasing a lot the electric energy consumptions from the BTM. This excessive cooling is due to the over usage of the compressor, the cabin temperature is still well regulated managing the blower, while the battery is over cooled because the minimum battery

battery pump command can not go below 0.2 to ensure fluid circulation, so if the fluid is cooled to much by consequence also the battery will be cooled even tough battery pump is kept at minimum flow rate. This behavior is produced by a limit of the linearization, in fact when the prediction is done for a too long time period it becomes unreliable because the predicted states will be considerably different from the real ones because of the non-linear behavior of the system. The cooling effect coming from the MPC MVs will not be predict if p is too long, since the MDs are not updated along the prediction, therefore this force the MPC to give as output higher values of MVs compromising the control efficiency.
The best trade-off was then find with:

$$p = 9; \quad pc = 4; \quad T_s = 1; \tag{3.6.1}$$

this values allow to maintain a good energy efficiency of the system without sacrifice the target temperatures tracking, furthermore this time horizon is similar to the system's response time, therefore the effect of the linearization does not influence the reliability of the control.
To conclude, it was also observed how p and pc does not influence the computational effort as T_s does.

Chapter 4

Results

In this chapter will be deeply analyzed the results obtained from the simulations. First of all will be clarified the test cycles used and assumptions done, after which will be compared AMPC with Reactive control logic focusing on energy savings and the difference in the usage of the control variables.

4.1. Test cycle used

To ensure the reliability of the results it was necessary to test the cooling system of the vehicle on usage cycles which represent real life driving conditions with the best possible accuracy, therefore were used the following emission/consumption test cycles which were deigned with that scope:

- **US FTP-72 or UDDS**: United States Federal Test Procedure or Urban Dynamometer Driving Schedule, which simulates an urban route of 12,7 km with frequent start and stops, the maximum speed reached in the cycle is 91,25 km/h [27].

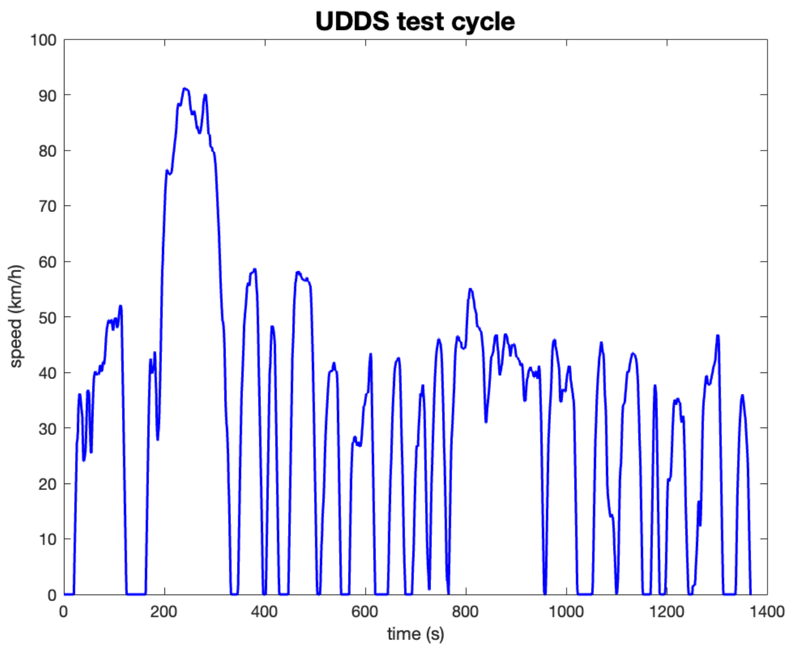

Figure 52: UDDS test cycle

- **WLTC3**: Worldwide harmonized Light vehicles Test Cycles are chassis dynamometer tests for the determination of emissions and fuel consumption from light-duty vehicles; in particular they are part of the WLTP. The 3 refers to the category of vehicle; in fact, the WLTC has different test cycles depending on the power-to-weight ratio, in our case the category is 3b, this means power to mass ratio (PMR) > 34 $[W/kg]$ and a maximum vehicle speed over 120 km/h. The WLTC3 is divided in low, medium, high and extra-high speed sections, for a total length of 23,266 km and a maximum speed of 131,3 km/h [28].

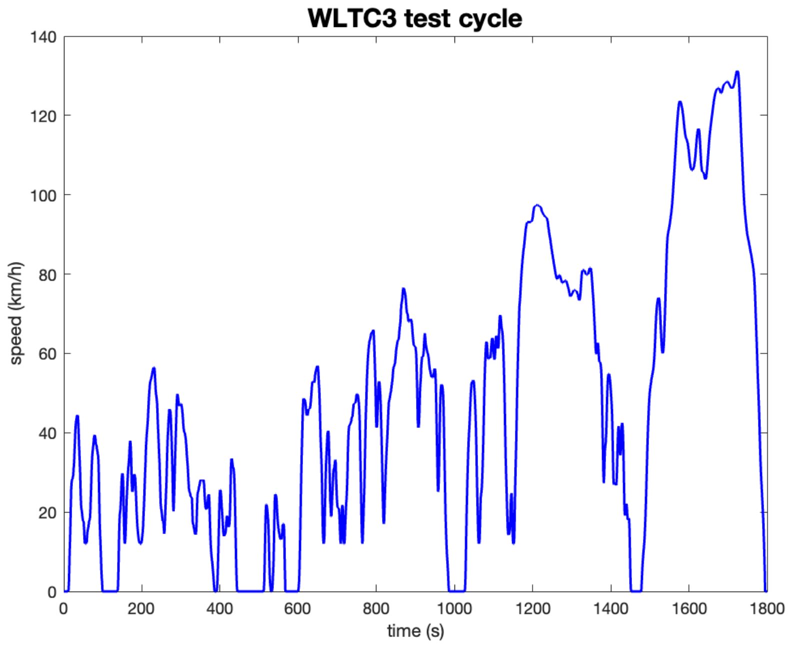

Figure 53: WLTC3 test cycle

It was decided to simulate both cycles to increase the reliability of the results. The cooling system is tested in urban and minor severe conditions with the UDDS cycle and in high stress conditions with the WLTC3, which presents higher speed and more severe accelerations and decelerations.

4.2. Simulation assumptions

For the simulation several assumptions were done:

- Parallel mode: the cooling system was always tested in parallel mode that is the condition in which is available the maximum cooling power. This was done to test the system with severe cooling requirement and to ensure its compliance in any conditions.

- Chiller by-pass valve closed: for the AMPC controller the chiller by-pass valve is kept closed any time, therefore $\dot{m}_{coolant\_chiller\_in} = \dot{m}_{b,pump}$. The cooling power in the chiller is regulated through a continuous regulation of the battery pump mass flow rate, while for the Reactive logic the battery pump mass flow rate has only three possible regulations related to its maximum flow rate which are 0.3,0.5 and 1. In the latter case the chiller cooling power is therefore regulated through the chiller by-pass valve.

- Cabin air recirculation always active: for both the AMPC and the Reactive control strategy recirculation was kept active, therefore the full mass flow rate of the air entering the blower comes from the cabin. This allowed to make in the Battery priority prediction model (3.1.1) the assumption that $T_{in\_evap} = T_{cabin}$.

- $T_{coolant\_chiller\_in} = T_{batt}$: this assumption was verified in the Simscape model putting a temperature sensor at the inlet of the chiller in the coolant side, ad what was verified was a maximum difference between the two temperatures below 1 degree, therefore this assumption allowed to simplify the prediction model without compromising its precision.

Results presented will come from the usage of:

- **Cabin/Battery MPC**: the switching between the two controllers is described in section 3.3.

- **Battery Forced MPC**: the usage of the only battery MPC is forced to see the behavior of that single controller.

- **Cabin Forced MPC**: the usage of the only cabin MPC is forced to see the behavior of that single controller.

- **Reactive control**: usage of the original controls.

It is important to remember that battery temperature target was set to 30°C the cabin one to 23°C, furthermore battery temperature is allowed to oscillate between 25 and 35 °C while cabin one can be adjusted from 16°C to 30°C.

4.3. AMPC vs Reactive on UDDS test cycle

In this section, the results of the simulations done, comparing AMPC and reactive control strategy, will be analyzed.
All the parameters used and corresponding values have been declared in previous sections.
Before commenting the obtained results, it is interesting to observe that with the UDDS test cycle the battery never overlapped the temperature of 35°C, therefore following the selection logic (3.3) the Cabin/Battery priority MPC will never switch to Battery priority, so it will have the exact same results of the Cabin Forced MPC.

4.3.1 Temperatures management

The first result commented is the compressor usage in figure 54. Starting from the Cabin Forced MPC we can observe how it manage to use the compressor power in a very constant and smooth way, the power along the cycle goes from 300 to 500 W. It start to use the compressor from the beginning of the cycle because the initial temperature is set at 26 °C while the cabin target as said before is 23 °C.

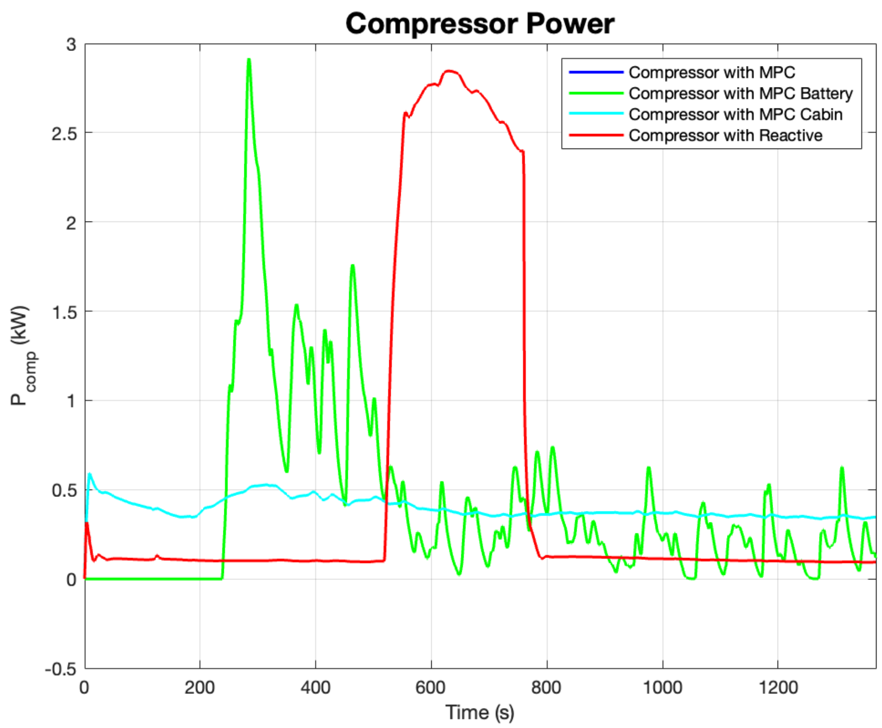

Figure 54: Compressor power UDDS

Battery Forced control present a curve of compressor usage with very sharp peaks in time correspondence with the accelerations in the driving cycle, because these are the moments in which the heat produced by the battery due to the Joule effect is maximum, therefore is required a proportional cooling power to keep the battery around the target temperature. In this case is observable that the compressor starts to work just around 250 s of simulation, this is because before that time the battery is below 30 °C (figure 55) which is the target to maintain; in the battery control prediction model 3.1.1 the P_{comp} MV is located in the battery temperature state equation, therefore, as said when it was presented the prediction model, the compressor starts to be used by the MPC only when is detected a T_b over the tracking value.

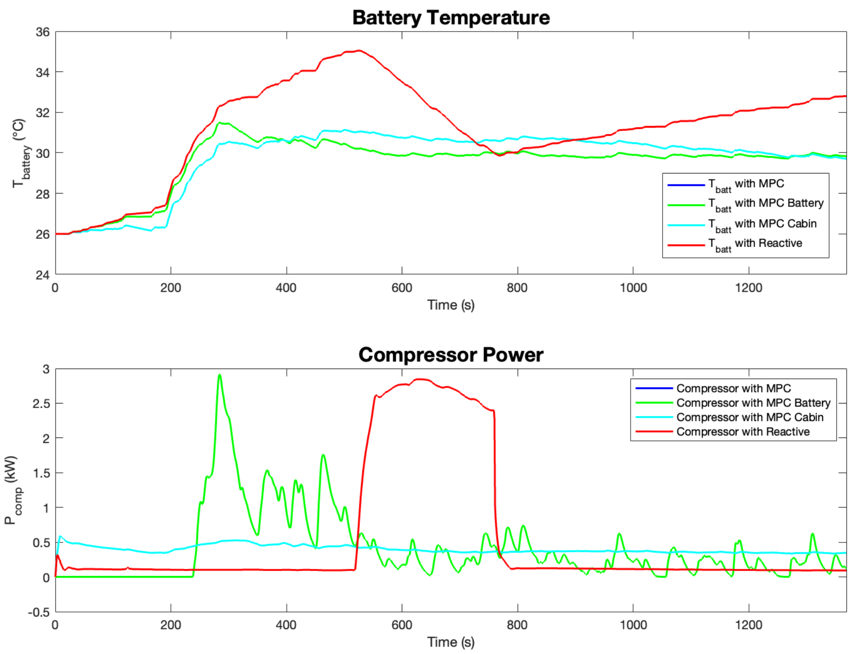

Figure 55: Battery Temperature UDDS

About the Reactive logic we can see from figure 55 that compressor is kept at very low power , to cool the cabin, until the maximum battery temperature is reached (35 °C), at that point the compressor is heavily activated to bring back the battery to the target value temperature. Is clearly observable how both the MPC manage to maintain very well the temperature target:

	RMSE [°C]
Reactive control	2.5708
Cabin Forced MPC	0.6025
Battery Forced MPC	0.3802

Table 12: RMSE battery temperature on UDDS

Root mean square error result to be the lowest with battery priority control as expected, ensuring the correct functioning of the latter.
Is also interesting to underline how at the beginning of the driving cycle the MPC

needs to maintain an higher power to cool the cabin, this is because the Reactive control, as said in 4.2, is able to regulate the chiller cooling power trough the by-pass valve, opening it when no cooling of the battery is needed. While for the MPC model the latter valve is assumed to be always closed, forcing the cooling of the battery even when only the cooling of the cabin would be necessary, requesting an higher power from the compressor; this open the way for a possible future development of the MPC control model.

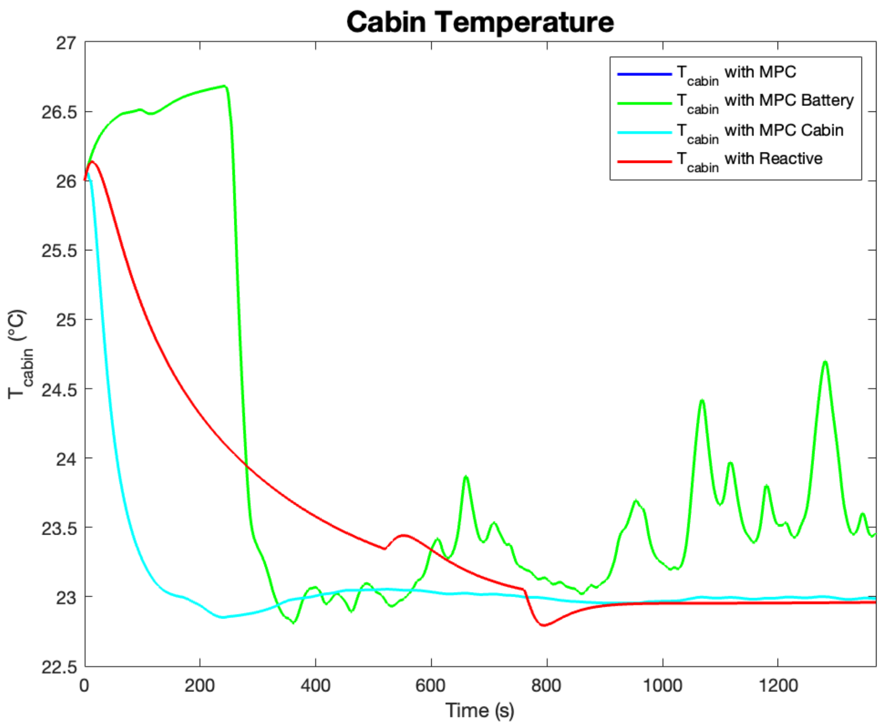

Figure 56: Cabin temperature UDDS

Commenting now the cabin temperature management in figure 56, starting this time from the Reactive control is observable how it starts to cool the cabin from the beginning of the cycle reaching and maintaining the target of 23 °C after 750 s of simulation, some small oscillations are present in the instants in which the compressor is turned up or turned down rapidly, because the blower has to readjust to avoid excessive or insufficient cooling of the cabin. The Battery priority MPC is not able to well manage the cabin comfort, it start to cool just after around 300 s od simulation, this because the battery priority prediction model use the P_{comp} only to explicitly control the battery temperature, therefore until the battery overlap the target temperature the compressor is deactivated and consequently also the cabin can not be cooled. Also in the later part of the simulation is visible how the cabin temperature is very dependent from the battery one presenting big oscillations which traduce in lack of comfort for the passengers. This is the reason for why it was developed the Cabin priority MPC which is shown to be able to track very efficiently the target, reaching it after 150 s of simulation so much faster than the reactive one. Below are exposed the rms errors from 750 s of simulation, which is the moment where all the controllers reach the target temperature:

	RMSE [°C]
Reactive control	0.0754
Cabin Forced MPC	0.0242
Battery Forced MPC	0.6453

Table 13: RMSE cabin temperature on UDDS

4.3.2 Commands usage

Now are compared the commands produced by the different controls type. The focus is made on *cmd_comp*, *cmd_ptc*, *cmd_blower* and *cmd_battery_pump*, which are the MVs of the MPC, remembering that for cabin priority one *cmd_blower* is generated by a PI controller and for battery priority one *cmd_battery_pump*.

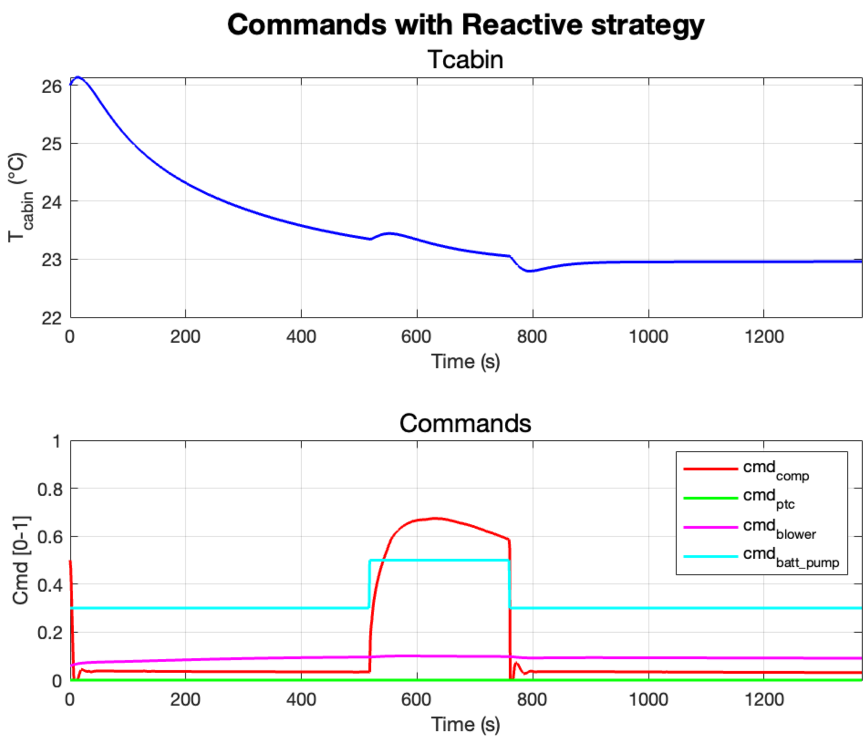

Figure 57: Reactive commands UDDS

In figure 57 are observable commands produced by Reactive control logic. The compressor usage was already described in the previous section and here are visible the oscillations of the cabin temperature mentioned before when compressor usage is rapidly increased or reduced. The battery pump command is maintained to 0.3 for most of the cycle, which means that in these regions the chiller is bypassed, while when cooling power is needed for the battery bypass valve is closed and the pump command increased to 0.5. Ptc as for the MPC is kept to zero because we are in cooling conditions, blower is maintained to a quite constant and low value promoting cabin comfort.

From figure 58, starting with the Battery priority MPC, the battery pump command is kept at a minimum value of 0.5 to prioritize battery cooling, the blower is used in a much more forceful way, because it is the only command the Battery MPC has to control the cabin temperature (see prediction model 3.1.1), therefore since the cooling power $COP \cdot P_{comp}$ is not dependent from the cabin temperature the controller try through the blower to optimize the temperature in the passengers compartment, this continuous regulation may cause some discomfort. Cabin priority MPC work in a very different way, maintaining as said before a very low but almost constant command of the compressor, utilizing also the blower at a constant regime, promoting the cabin silence and comfort; the command value used for the latter is a bit higher than the one from the Reactive control, this because the MPC try to exploit more the blower which has a very low impact on the electric energy consumption to maximize the energy efficiency. The battery pump command, since for this cycle a strong cooling is not required, is maintained to the minimum value, which is 0.2.

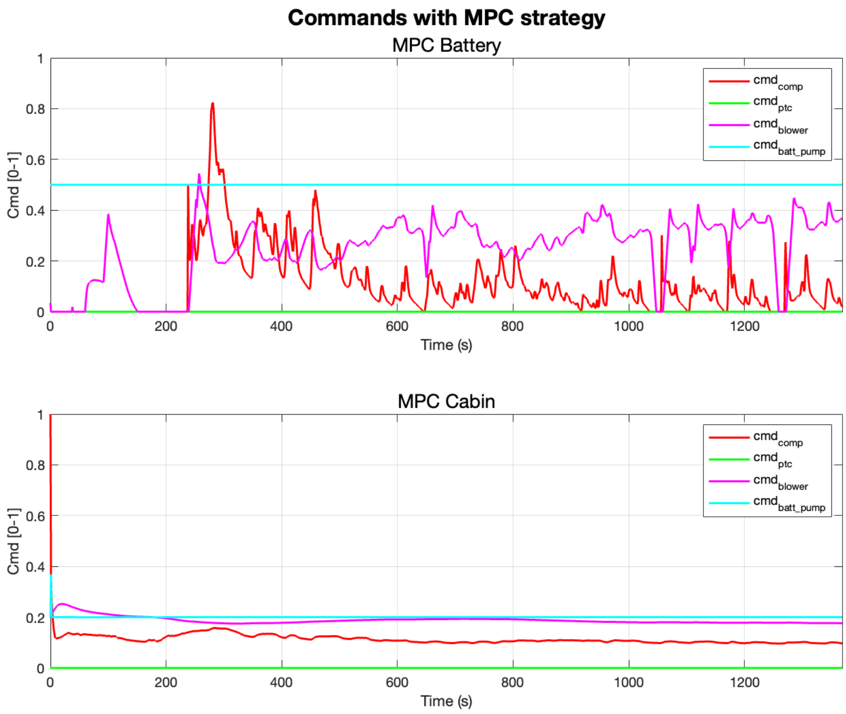

Figure 58: Cabin and Battery MPC commands UDDS

4.3.3 Energy consumption and savings

In addition to the ability to properly track the set target temperatures, the other objective of the MPC was to optimize the energy consumption of the BTM.
Consumptions of compressor and radiator's fan are shown in figure 59; this two components are analyzed because they are the ones which impact the most on the total BTM electrical requests, in particular between the two the compressor presents an order of magnitude of difference being by far the most impacting component. Is observable how both MPC use in a much optimized way the energy, thanks to the prediction model they are able to activate in advance the compressor avoiding to use it at high power for long periods as the Reactive logic does, having so an higher consumption at the beginning of the driving cycle but being able to save energy in the long term. The table below shows the respective consumption of these components for each control type:

	Compressor [kWh]	Fan [kWh]
Reactive control	0.2044	0.0180
Cabin Forced MPC	0.1501	0.0069
Battery Forced MPC	0.1561	0.0069

Table 14: Compressor and Fan energy consumptions on UDDS

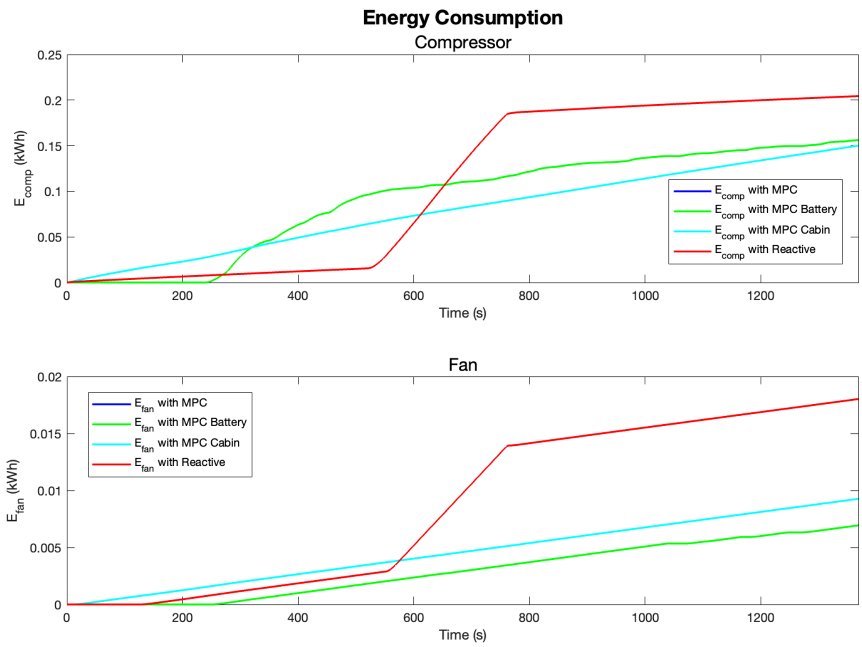

Figure 59: Compressor and Fan energy consumption UDDS

The total BTM consumptions are shown in figure 60, these values are obtained from the contribution of compressor, fan, blower, battery and motor pumps. The graph has a similar path to the compressor one, since, as said before, the latter component is the one impacting the most the total BTM consumptions.

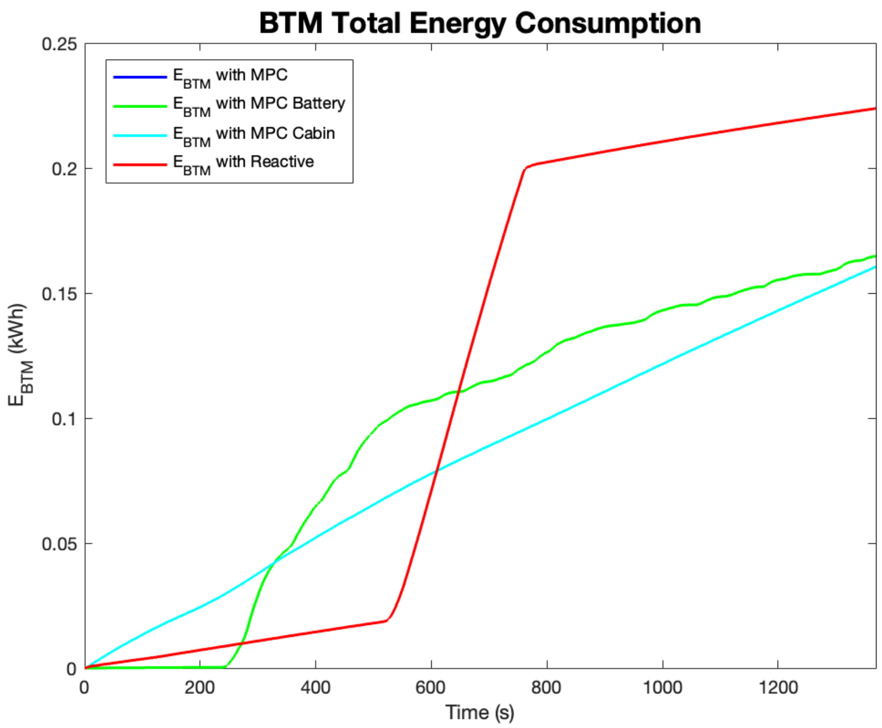

Figure 60: BTM energy consumption UDDS

In the following table are summarized the consumption, savings and tracking performances:

	BTM [kWh]	Total [kWh]	RMSE T_b [°C]	RMSE T_c [°C]
Reactive control	0.2238	2.2928	2.5708	0.0754
Cabin Forced MPC	0.1605 -28.28 %	2.2305 -2.72 %	0.6025	0.0242
Battery Forced MPC	0.1648 -26.36 %	2.2338 -2.57 %	0.3802	0.6453

Table 15: Overall savings and performance on UDDS

Considerable savings of almost one third on the total BTM are obtained with both types of MPC, producing an overall saving along the UDDS cycle over 2.5%. This percentage is obtained by testing the system with 26 ° C environmental temperature, but it will turn out to be even higher if the system is tested under more extreme environmental conditions where the BTM has a greater impact on total consumption.

Over the grater energy efficiency, MPCs resulted also better in tracking the target temperature, producing lower rms errors. For this test cycle when the selection logic (3.3) is activated, only cabin priority MPC will be used since as seen previously with this control is never reached a T_b over 35 °C, therefore there are no switch between the two controllers. This MPC results to track a little worst the T_b with an higher rms error compared to the Battery priority one, but still guarantee great performances compared to Reactive logic.

4.4. AMPC vs Reactive on WLTC test cycle

As said in the previous chapter, the UDDS cycle did not result stressful enough to require the usage of the Battery priority MPC, therefore it was decided to test the system on the WLTC3 cycle, which, as explained, presents higher accelerations, to see how the switch between the two MPCs is managed.

4.4.1 Temperatures management

As before the analysis starts from the compressor usage shown in figure 61, this because it is necessary to understand the battery and cabin temperature management. Here differently from what happened with the UDDS cycle, we have four different graphs, because the blue one corresponds to the one of the MPCs used with the selection logic. In the figure are also represented the results forcing the Cabin and Battery MPCs to see how they work individually.

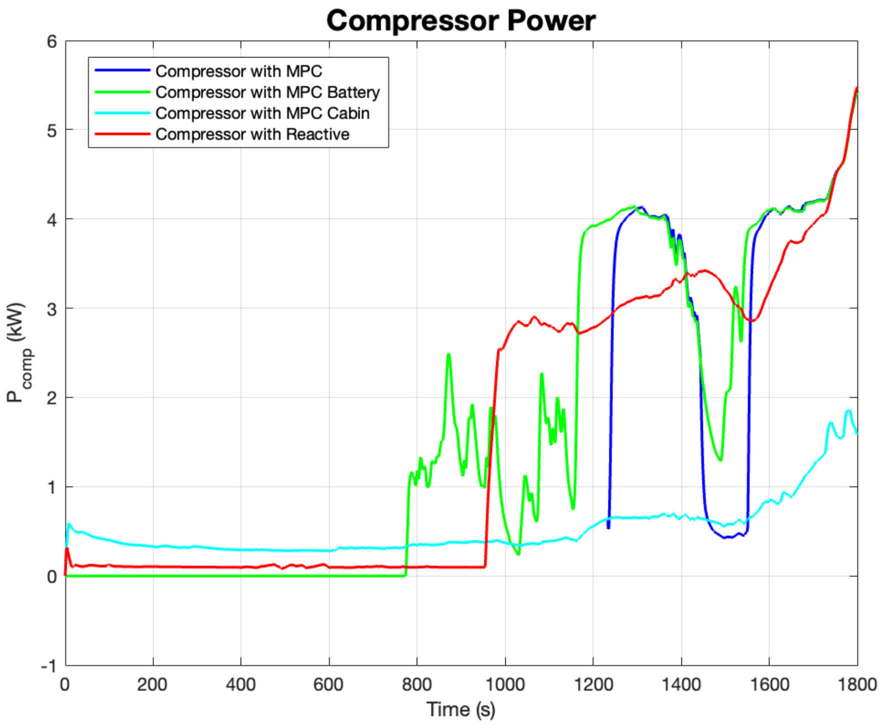

Figure 61: Compressor power WLTC

Starting for the Battery priority MPC, as it was for the UDDS cycle it start to activate in advance the compressor compared to the Reactive to optimize the temperature tracking and consumptions, but until the battery does not reach the target temperature it is maintained turned off, this as will be seen also influences the cabin cooling. Cabin priority control maintain a low power profile for the compressor, but still higher than the Reactive logic until the T_b reach the target, this because as said before the MPC can not open the chiller bypass valve, therefore to cool the cabin it

indirectly also needs to cool a bit the battery.
Very high power profiles are observable near the end of the cycle, this is because the total heat generated by Joule losses is very high in that part of the cycle due to high acceleration profile. Even the Cabin forced MPC increase the compressor usage by the end of the cycle, because the high increment in battery temperature produces high heating absorption from the chiller, which traduce to less cooling power also for the evaporator, then the Cabin MPC, even if not directly, is forced to increase the compressor power (visible at the end part of the cycle) to maintain the target temperature in the cabin.

From figure 62, is observable how none of the controllers is able to keep the battery temperature within the 35 °C maximum limit. This is because the cooling system is undersized, this means it is not able to maintain cool the battery in case of maximum electrical load request. This is shown in figure 63, where in blue is represented the maximum theoretical cooling power and in red the Joule losses along the WLTC, is clearly visible how near the end of the cycle the losses are more than double the maximum cooling power, this reflects in the battery temperature which go up to around 60 °C.

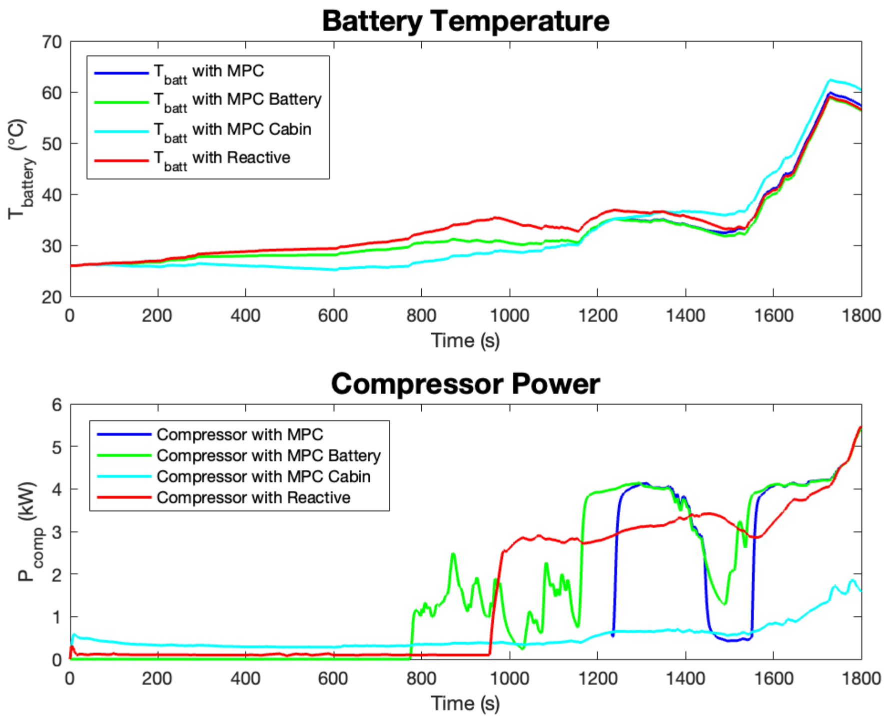

Figure 62: Battery Temperature WLTC

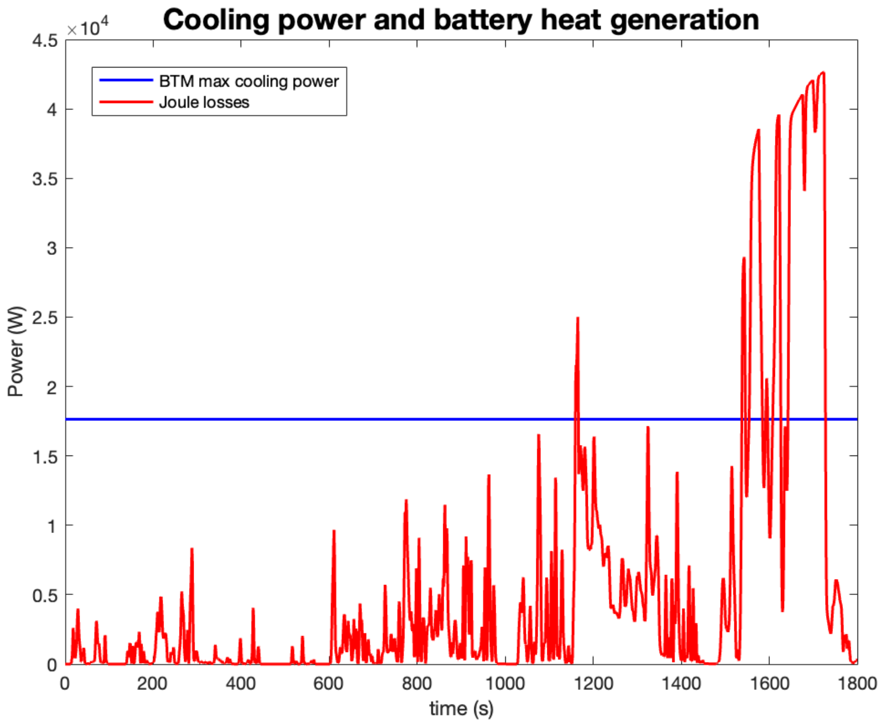

Figure 63: Cooling power and battery heat generation

A part the above mentioned effect, the MPCs resulted to manage much better the T_b specially in the first half of the cycle, while in the later part Cabin MPC is the one who reach the higher temperatures due to its inability to directly control the compressor to regulate the T_b. Overall the Battery MPC is the one able to maintain the lower peak temperature, but not far from is the Cabin/Battery MPC.
Tracking performances are shown above:

	RMSE [°C]
Reactive control	11.047
Cabin Forced MPC	12.457
Battery Forced MPC	10.363
Cabin/Battery MPC	10.875

Table 16: RMSE battery temperature on WLTC

Even if all controllers do not satisfy the cooling request due to the undersizing of the BTM, the Cabin/Battery MPC is still able to produce a lower rms error compared to Reactive control, with a considerable energy saving that will be analyzed below.
From figure 64 is observable the cabin temperature management. As observed with the UDDS cycle the Cabin priority MPC cool the cabin much faster than the Recative one, reaching the target of 23 °C after 150 s instead of 950 s promoting passenger

comfort. Battery priority MPC is not able to cool immediately the cabin as said before because is limited by the T_b, after it reach the 30 °C target (around 800 s) the compressor is activated and the cabin start to be cooled reaching the target temperature of 23 °C and oscillating around it. With this cycle the Cabin MPC is not able to track the T_c target at the end of the cycle because the compressor is not activated enough, it was tested that increasing the weight (see section 3.2.3) on the cabin temperature deviation the target is reached and keep, but this produced an higher electrical consumption, and considering that the controller that is intended to be used is the Cabin/Battery MPC, it was chosen to not increase too much the weight on battery temperature as explained in section 3.2.3 to promote the energy saving being aware that at high battery temperature the Cabin/Battery MPC would chose to use the Battery MPC that at high cooling request (as shown in figure 64) is also better to manage the T_c.
Cabin/Battery MPC presents some oscillation in the moment the switch between controllers is done, this is due to the fact that when one controller is used on the other is deactivated the QP optimization to reduce the computational effort (as explained in section 3.3). This allows to each controller to have always refreshed the state of the MDs, but when the QP optimization is deactivated, the controller interested, keep internally as MVs values the last one calculated before the deactivation; therefore when it is reactivated the state of the MVs will be different (because until the previous instant were controlled by the other MPC) compared to the one internally saved, so the MPC try to correct it but having the system components, for example compressor and blower, different inertia, their state will change in different time causing the oscillations visible in the cabin temperature figure. By the way these oscillations are at most of 0.35 °C, this makes them almost undetectable from passengers.
In the following table are shown rms errors starting from simulation time of 950 s which is the instant in which all the controller reach the cabin target temperature:

	RMSE [°C]
Reactive control	0.0745
Cabin Forced MPC	0.1456
Battery Forced MPC	0.1601
Cabin/Battery MPC	0.1603

Table 17: RMSE cabin temperature on WLTC

Rms error is higher for Cabin/Battery MPC, but we have to consider that with that controller the target is reached 800 s before, compared to Reactive one and also significant energy savings are achieved 4.4.3, justifying this slightly higher rms error due mainly to the controller switch oscillations explained just before.

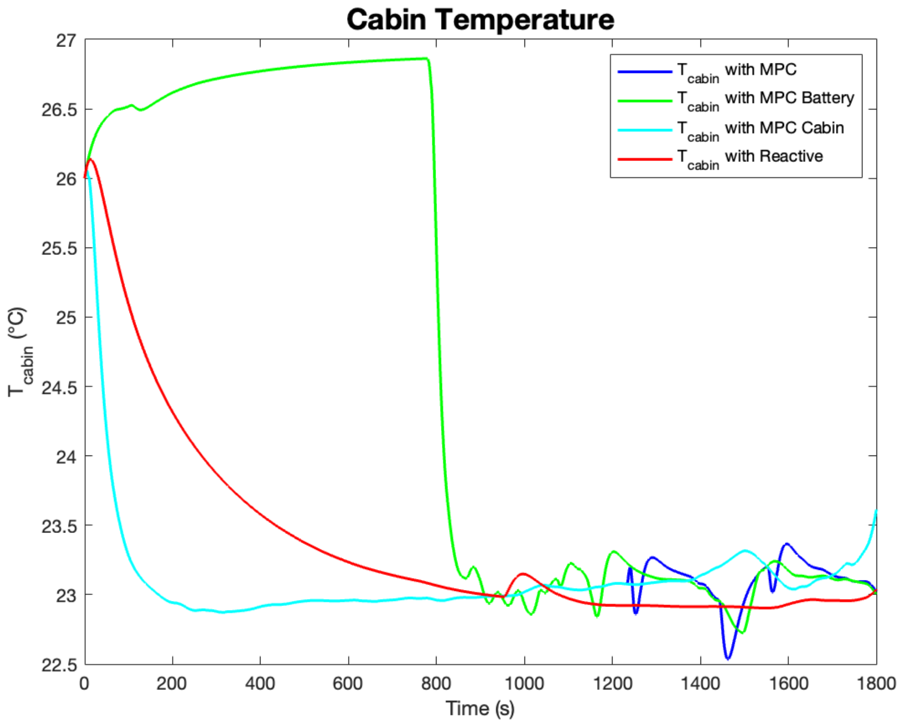

Figure 64: Cabin temperature WLTC

4.4.2 Commands usage

Here are described the commands used, they are the same mentioned on the UDDS cycle section, *cmd_comp*, *cmd_ptc*, *cmd_blower* and *cmd_battery_pump*. As before, for the Battery MPC the *cmd_battery_pump* is managed by a PI controller as the *cmd_blower* for the Cabin MPC.
Starting from Reactive control, as visible in figure 65, the compressor, as already seen before, is used at a low value, to cool the cabin, bypassing the chiller, until maximum battery temperature limit of 35 °C is reached, then it is activated at high power value. The battery pump command is maintained to 0.3, which is the minimum value for the Reactive control, when T_b goes up over the maximum target its value is increased to 0.5, then near the end of the cycle when T_b rise up to 60 °C the battery pump is kept to the maximum flow rate to maximize the heat exchange in the chiller. Blower is used at low values increasing the cabin comfort, while ptc is always at zero because no heating is needed.

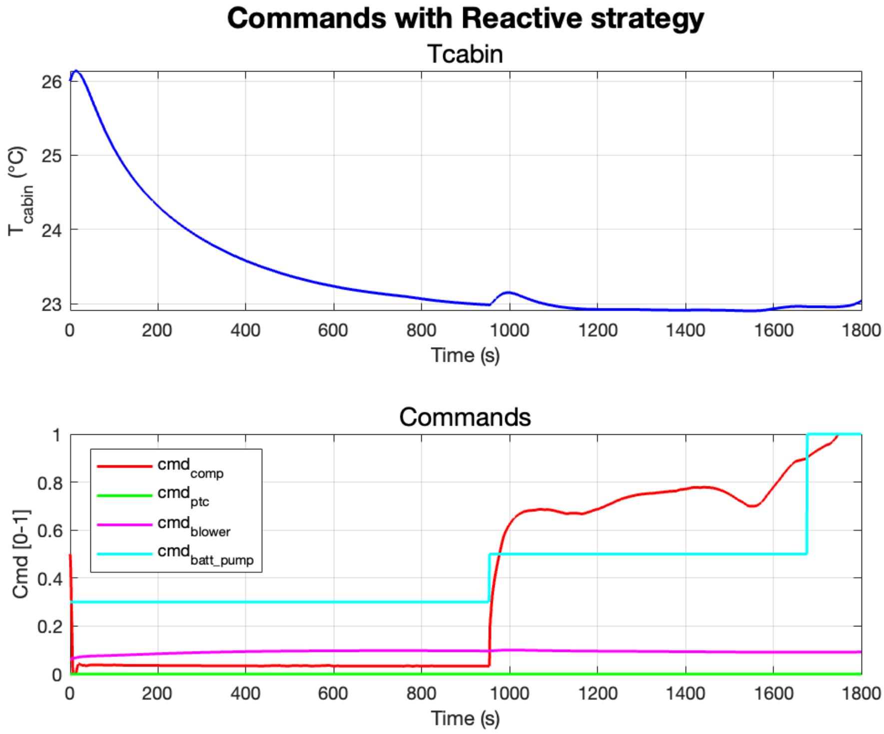

Figure 65: Reactive commands WLTC

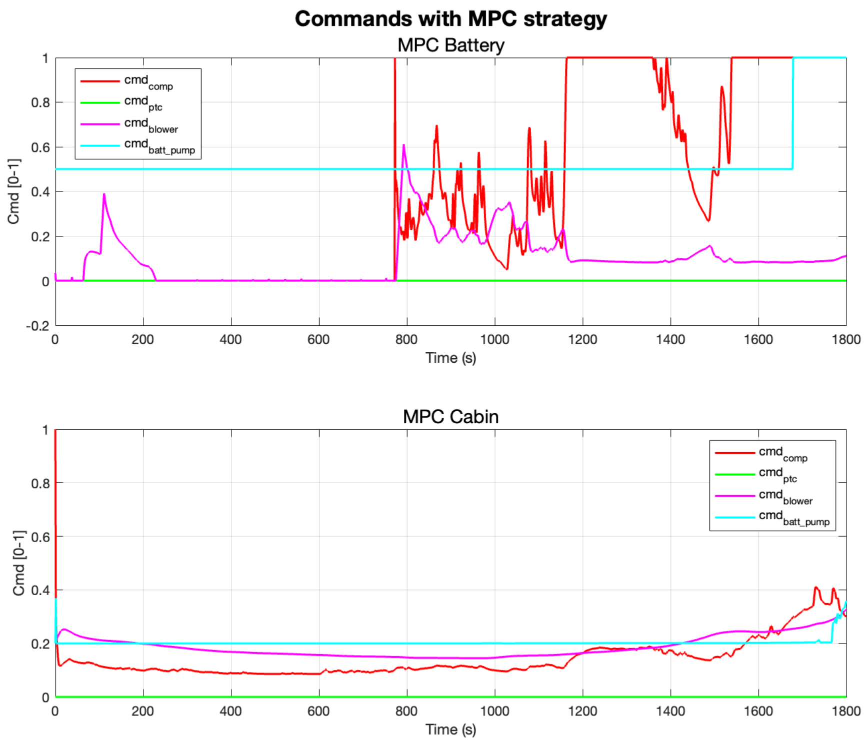

Figure 66: Cabin and Battery MPC commands WLTC

Battery MPC shown in figure 66, manage the blower to work in a more sustained way to cool the cabin to compensate the infeasibility to use the compressor for that purpose; while the pump command is maintained to 0.5 as minimum value and increased to 1 at the end of the cycle for maximum cooling. The Cabin MPC instead use the compressor at low power for most of the cycle forcing it more at the end of the cycle to compensate the excessive battery heating, as described in previous section. The blower also is used at a low-mid value to guarantee heat exchange in the evaporator but still maintaining the cabin comfort, while the pump command is maintained to 0.2 for most of the cycle increased just at the end where more heat exchange is needed.

Now the Cabin/Battery MPC is analyzed (figure 67), its behavior is an hybrid between the two different MPCs. When Cabin MPC is selected a low compressor power is used, almost constant blower and battery pump command is kept at 0.2; on the counter when the selected controller is the Battery MPC, because more cooling is needed to the battery, the compressor is used more as also the blower, while the battery pump command is maintained at 0.5 except at the end of the cycle where maximum cooling power is requested. The switch between the two MPC are visible in the zones of T_c oscillations, which corresponds also to when the battery pump command pass from 0.2 to 0.5 and vice versa.

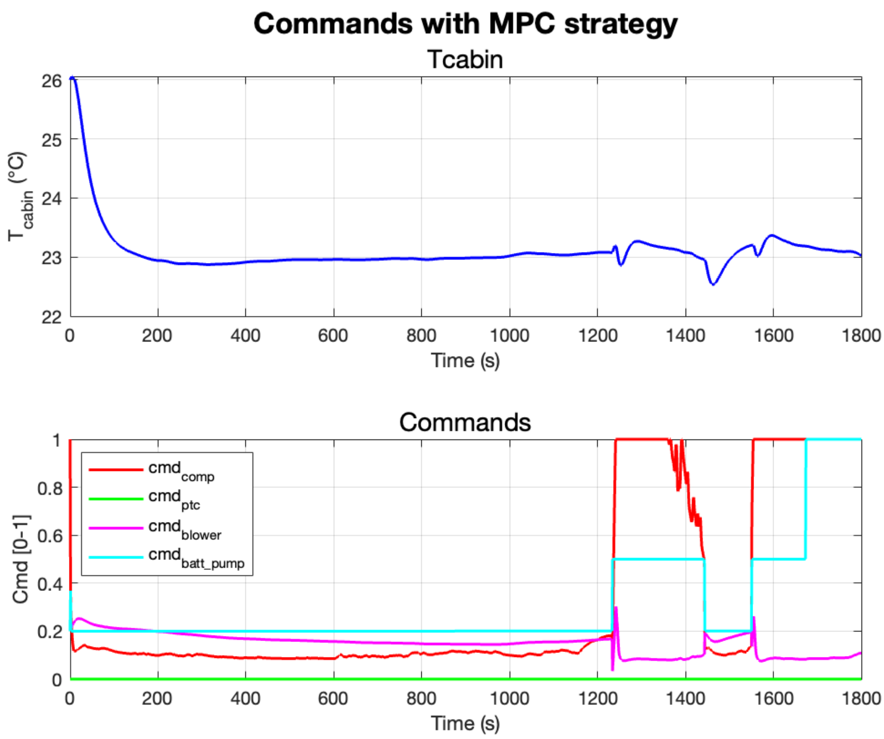

Figure 67: Cabin/Battery MPC commands WLTC

4.4.3 Energy consumption and savings

Compressor and radiator's fan consumptions are shown in figure 68; as it was on the UDDS cycle compressor confirms to be the most energy demanding component of the BTM with more that an order of magnitude of difference compared to the fan, this bigger impact, compared to the UDDS, is due to the fact that this test cycle ask for higher cooling request so the gap of consumption between compressor and fan further increase. The predictive behavior of the MPC cause an higher energy consumption at the beginning of the cycle compared to reactive control, which traduce to an energy saving in the later stages and a better temperature managing. The table below shows the consumption of the two mentioned components for each control type:

	Compressor [kWh]	**Fan [kWh]**
Reactive control	0.7801	0.0497
Cabin Forced MPC	0.2581	0.0296
Battery Forced MPC	0.7945	0.0310
Cabin/Battery MPC	0.6422	0.0343

Table 18: Compressor and Fan energy consumptions on WLTC

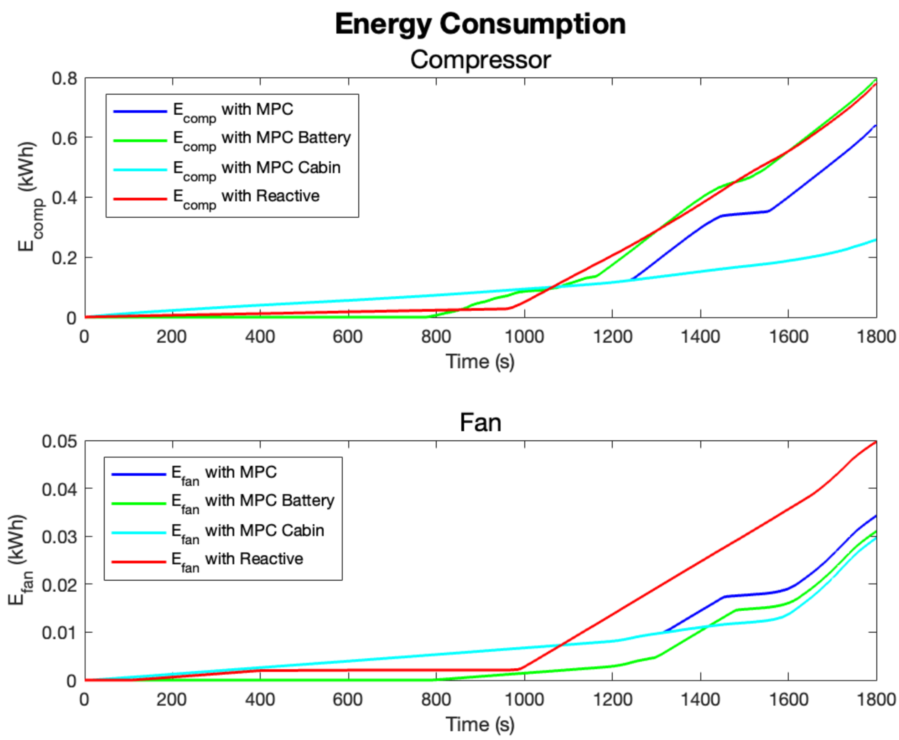

Figure 68: Compressor and Fan energy consumption WLTC

Total BTM consumptions are shown in figure 69, which represents the sum of contributions from compressor, fan, blower, battery and motor pump. As for the UDDS this graph has a similar path to the compressor one, since it is the most impacting component.

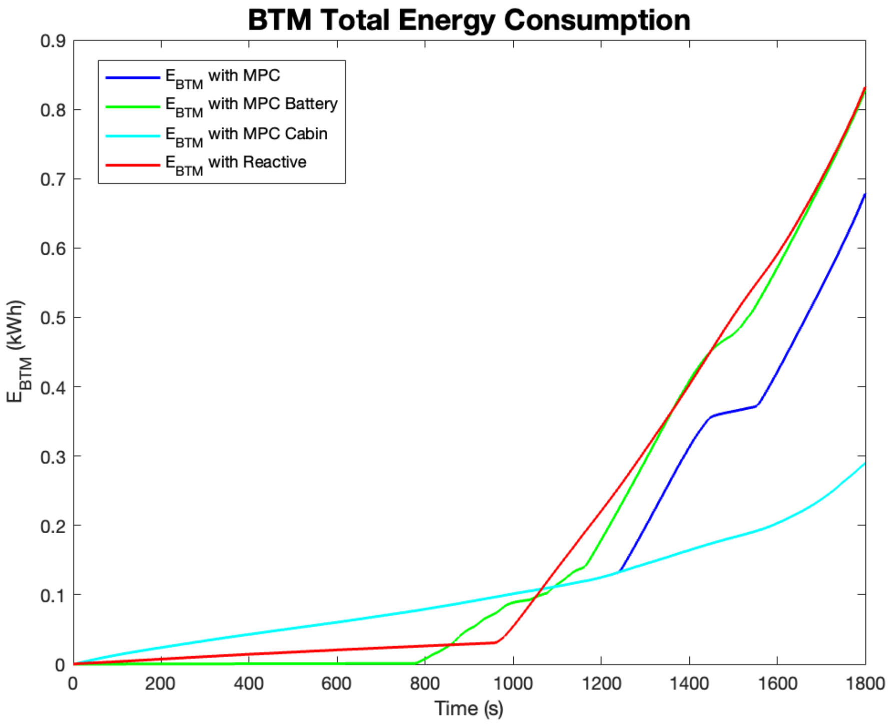

Figure 69: BTM energy consumption WLTC

In the following table are summarized the consumption, savings and tracking performances:

	BTM [kWh]	Total [kWh]	RMSE T_b [°C]	RMSE T_c [°C]
Reactive control	0.8323	5.3086	11.047	0.0745
Cabin Forced MPC	0.2897 -65.19 %	4.9470 -6.81 %	12.457	0.1456
Battery Forced MPC	0.8284 -0.47 %	5.2761 -0.61 %	10.363	0.1601
Cabin/Battery MPC	0.6787 -18.45 %	5.1841 -2.35 %	10.875	0.1603

Table 19: Overall savings and performance on WLTC

The Cabin MPC is able to achieve an energy saving of more than 65% for the BTM, this is a huge reduction in the consumption, but we have to consider that with this controller the peak T_b is the highest achieved compared to the other controllers (figure 62). This controller is shown to be unsuitable for use in all conditions, since insufficient cooling of the battery is provided at high loads, aging and damaging the battery prematurely. On the other hand, the Battery MPC is not able to achieve great energy savings along this driving cycle, but is the one who better manages the T_b throughout the cycle, preventing excessive aging of the battery. Finally, the Cabin/Battery MPC is the trade off able to ensure passenger comfort, good management of the T_b and energy savings of more than 18% for the BTM which traduce in 2.35% saving on the total electric energy consumption, a percentage that has the potential to increase in more challenging environmental conditions where the BTM impact more on total vehicle energy consumption. For this controller, the higher rms error on the T_c, as said before, is due to the oscillations given by the switching between the two MPCs, which is at the maximum point 0.35 °C far from the target of 23 °C, a low value justified by the considerable energy savings.

Chapter 5

Conclusions

In this thesis work, an advanced predictive control strategy for the control of the BTM and the HVAC systems through an adaptive MPC was developed. The main goal was to improve the energy efficiency of the aforementioned systems, guaranteeing at the same time comfort in the passenger compartment, and protecting the battery from critical temperatures to avoid premature aging.
Main results achieved comparing the Cabin/Battery MPC with the Reactive control strategy are:

- Increased energy efficiency: BTM consumption reduction of over 28% on the UDDS and 18% on the WLTC, with an overall energy saving above 2.5% and the potential to see that percentage increase in more challenging environmental conditions where the energy used for cooling has a greater impact on the total energy consumed by the vehicle.

- Better temperature tracking: for most of the simulations the tracking of the reference temperatures of 30 °C for the battery and 23 °C for the cabin was achieved with a lower rms error, showing the ability of the MPC to better maintained the desired values plus reducing the consumptions. Is also to be considered, about the passengers comfort, that the Cabin MPC is able to reach the 23 °C starting from an initial condition of 26 °C in 150 s compared to the 950 s needed by the Reactive logic.

- Some fluctuations of T_c are present when the switching between Cabin and Battery MPC is performed and vice versa, the reasons were analyzed in Chapter 4, but considering that these oscillations are in the order of the tenth of Celsius degree them are not even perceived by the passengers, furthermore them are largely compensated by the considerable energy consumption reductions and thermal management of the vehicle mentioned just above.

These results demonstrate how the adaptive MPC can represent a significant step forward in the thermal management on electric vehicles, contributing to improve both energy efficiency and durability of critical components such as the battery which represents a significant percentage of an EV cost.

These results were obtained thanks to the Cabin/Battery MPC which is composed by two different MPCs:

1. Battery priority MPC: with a prediction model designed to optimally manage the battery temperature, acting on the compressor, blower and ptc heater

commands. This strategy guarantees protection of the battery from excessive heating increasing the operating life.

2. Cabin priority MPC: with a prediction model designed to prioritize the cabin comfort acting on the compressor, battery pump and ptc heater to reach the desired temperature set by the passengers.

By default is used the Cabin priority MPC to guarantee maximum comfort for passengers, when the battery temperature increase over the limit of 35 °C, the selection logic decide to switch to Battery priority MPC to avoid over temperature in the battery, then when T_b goes below 33 °C the selection logic switch back again to Cabin priority MPC.

Future developments

The results obtained showed the potential of this control method for the thermal management on EV, paving the way for further future improvements and development aimed to optimize even more the energy used by the BTM and HVAC. Possible evolution may include:

- Experimental validation: since at the moment all the results have been obtained through simulations with the virtual model created in Simscape, a key passage will be the future testing of the thermal management system on a test bench that will be build in Politecnico of Turin.

- Study of the impact of extreme environmental conditions: test the controller with very high and very low temperatures to verify that total percentage energy savings increase in these conditions.

- Optimal tuning: find a method to optimally tune the cost function weights without the necessity to proceed with a trial-and-error procedure.

- Further optimization and usage of all the possible working modes: another step in the optimization could be made introducing the usage of the chiller bypass valve for the MPC to avoid battery cooling when is not needed, since the battery pump can not be completely switched off to avoid the formation of temperatures gradients inside the cooling loop the clever usage of this valve could bring to even higher energy savings. Plus the reintroduction of the usage of the four way valve to use the system both in parallel than serial mode could increase the overall system efficiency. The usage of this valves can not be integrated in the same way it was done for the Reactive control, but it has to be adapted with the MPC operating mode.

- Optimization of the selection command: implementation of a filter when the switching between the two MPCs is done to reduce T_c oscillations and improve even more the cabin comfort and control precision and robustness.

In conclusion, this control method represents an advanced and optimized way for the thermal management of EVs going towards an increasingly advanced optimization of BTM and HVAC systems, contributing to the vehicle energy efficiency and the sustainability of the automotive sector.

Bibliography

[1] L. Lefebvre. Smart Battery Thermal Management for PHEV Efficiency. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, 68(1):149–164, 2013.

[2] GM Volt Community. The chevrolet volt cooling & heating systems explained, 2024. Accessed: 2024-03-06.

[3] Yunus A. Çengel and Michael A. Boles. *Thermodynamics: An Engineering Approach*. McGraw-Hill Education, 9 edition, 2019.

[4] Wikipedia contributors. Thermal expansion valve, 2024. Accessed: 2024-03-06.

[5] MathWorks. Mpc modeling, 2024. Accessed: 2024-03-06.

[6] International Energy Agency. Trends in electric cars. *Global EV Outlook 2024*, 2024. Accessed: 2025-03-06.

[7] Christopher R. Knittel and Shinsuke Tanaka. Challenges to expanding ev adoption and policy responses. *CEEPR Working Paper 2024-16*, 2024. Accessed: 2025-03-06.

[8] Shuai Ma, Modi Jiang, Peng Tao, Chengyi Song, Jianbo Wu, Jun Wang, Tao Deng, and Wen Shang. Temperature effect and thermal impact in lithium-ion batteries: A review. *Progress in Natural Science: Materials International*, 28(6):653–666, 2018.

[9] The Engineering Toolbox. The engineering toolbox, 2024. Accessed: 2024-03-06.

[10] Jeremy Neubauer and Eric Wood. Thru-life impacts of driver aggression, climate, cabin thermal management, and battery thermal management on battery electric vehicle utility. *Journal of Power Sources*, 259:262–275, 2014.

[11] Zhen Zhang, Wei Wang, Jian Xu, Fan Jiang, Zhiyong Li, and Wei Zhang. Thermal management of lithium-ion batteries using phase change materials: A review. *Energy*, 234:121243, 2021.

[12] Mohammad Reza Amini, Jing Sun, and Ilya Kolmanovsky. Two-layer model predictive battery thermal and energy management optimization for connected and automated electric vehicles. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6976–6981, 2018.

[13] Shuofeng Zhao and Chunting Chris Mi. A two-stage real-time optimized ev battery cooling control based on hierarchical and iterative dynamic programming and mpc. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11677–11687, 2022.

[14] Mohammad Reza Amini, Ilya Kolmanovsky, and Jing Sun. Hierarchical mpc for robust eco-cooling of connected and automated vehicles and its application to electric vehicle battery thermal management. *IEEE Transactions on Control Systems Technology*, 29(1):316–328, 2021.

[15] Patrick Schutzeich, Stefan Pischinger, David Hemkemeyer, Kai Franke, and Paul Hamelbeck. A predictive cabin conditioning strategy for battery electric vehicles. *World Electric Vehicle Journal*, 15(6), 2024.

[16] You-Syuan Chen, Kyoung hyun Kwak, Jaewoong Kim, Youngki Kim, and Dewey Dohoy Jung. Energy-efficient cabin climate control of electric vehicles using linear time-varying model predictive control. *Optimal Control Applications and Methods*, 44:773 – 797, 2021.

[17] MathWorks. Electric vehicle thermal management, 2024. Accessed: 2024-03-06.

[18] Yaşar Demirel and Vincent Gerbaud. Chapter 4 - using the second law: Thermodynamic analysis. In Yaşar Demirel and Vincent Gerbaud, editors, *Nonequilibrium Thermodynamics (Fourth Edition)*, pages 187–265. Elsevier, fourth edition edition, 2019.

[19] M.C. Diniz, E.L.L. Pereira, and C.J. Deschamps. Simulation model to predict temperature distribution along scroll wraps. In *8th International Conference on Compressors and their Systems*, pages 301–310. Woodhead Publishing, 2013.

[20] G.F. Hundy, A.R. Trott, and T.C. Welch. Chapter 10 - component selection and balancing. In G.F. Hundy, A.R. Trott, and T.C. Welch, editors, *Refrigeration, Air Conditioning and Heat Pumps (Fifth Edition)*, pages 165–182. Butterworth-Heinemann, fifth edition edition, 2016.

[21] Yulu Chen, Akihito Ozaki, and Haksung Lee. Energy saving potential of passive dehumidification system combined with energy recovery ventilation using renewable energy. *Energy and Buildings*, 268:112170, 2022.

[22] MathWorks. Model predictive control (mpc), 2024. Accessed: 2024-03-06.

[23] MathWorks. Optimization problem in model predictive control, 2024. Accessed: 2024-03-06.

[24] Novara C. Driver assistance system design a, 2024. lecture notes.

[25] Domenico Altavilla. Model predictive control for battery thermal management. Master's thesis, Politecnico di Torino, April 2024.

[26] MathWorks. Tuning weights in model predictive control, 2024. Accessed: 2024-03-06.

[27] DieselNet. Ftp-72 test cycle, 2024. Accessed: 2024-03-06.

[28] DieselNet. Emission test cycles: Wltc, 2025. Accessed: 2025-03-06.