Master's Thesis

Master's Degree in Electronic Engineering

Vehicle HVAC system modeling and controlling

October 22, 2021

Author:	Sergi Esteban Bueno
UPC director:	Domingo Biel Solé
Idneo directors:	Jose Luis Blanco López
	Marc Güell Brocal

Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona

«Ce que nous connaissons est peu de chose, ce que nous ignorons est immense»
- Pierre-Simon Laplace -

Acknowledgment

I would like to thank to the thesis UPC tutor, Domingo Biel, for his continuous guidance, his advice as well as for his project supervision.

I am very grateful with the Mobility HW team at Idneo, specially with Nicolás Murguizur, Salvador Poveda, Marc Guijosa and Ramón Ramos for their assistance with any question and also with the ECAD team for the given support.

I am also grateful with my thesis co-advisor Marc Güell and my manager Manuel Márquez for making the project development possible at Idneo.

I would like to recognize my thesis advisor Jose Luís Blanco, for his always immediate availability and for being a source of technical knowledge.

Finally, but not least important, I appreciate my closest relatives for their constant support during the thesis development and entire master's degree.

Table of Contents

List of Figures

List of Tables

Acronyms

3D 3-dimensional. 26

ADC Analog-to-Digital Converter. 82
APP Application. 69
ASHRAE American Society of Heating, Refrigeration and Air-conditioning Engineers. 22, 42
ASPICE3.1 Automotive Software Performance Improvement and Capability dEtermination. 5, 15
AV Autonomous Vehicle. 13

BCM Body Control Module. 15, 16, 18, 54, 55, 79
BW Bandwidth. 29, 67

CAN Controller Area Network. 81
CPU Central Processing Unit. 81

DAC Digital-to-Analog Converter. 82
DIO Digital I/O. 82

ECU Electronic Control Unit. 15, 18, 79, 81, 82
ESD Electrostatic Discharge. 82
EV Electric Vehicle. 13, 22

FPGA Field-Programmable Gate Array. 81

HBM Heat Balance Method. 25, 26, 29, 31
HIL Hardware-in-the-Loop. 17, 37, 67, 79–82
HPF High-Pass Filter. 69
HV Hybrid Vehicle. 13
HVAC Heating, Ventilation and Air Conditioning. 18, 19, 21, 25, 27, 29, 32–34, 38, 42, 47, 52, 56, 57, 64, 68, 72, 75, 78, 79, 81
HW Hardware. 5, 14–18, 37, 56, 79, 81, 82

I.C. Initial Conditions. 58
I/O Input Output. 79, 81

LIN Local Interconnect Network. 81
LMTD Logarithmic Mean Temperature Difference. 51
LPF Low-Pass Filter. 69

MCU Micro Controller Unit. 79
Mech Mechanical. 14–16, 18

PCB Printed-Circuit Board. 82
PID Proportional, Integral and Derivative controller. 67, 69
PMV Predicted Mean Vote. 22–24, 29, 52, 54, 55, 65–67, 70, 72, 75
PPD Predicted Percentage of Dissatisfied. 22
PTC Positive Temperature Coefficient. 19, 33, 35, 47, 48, 53, 65, 67
PWM Pulse-Width Modulator. 52, 81

RH Relative Humidity. 58
RPC Rapid Control Prototyping. 79

specs Specifications. 14, 15
SPICE Simulation Program with Integrated Circuits Emphasis. 16, 17
SW Software. 14–18, 37, 56, 79

TCI Thermal Comfort Indexes. 22

WFM Weighting Factor Method. 25, 26

Abstract

HVAC systems have been developed and improved according to thermal control assessment needs. There is a wide application range in which these kind of system are used depending on the particular objective: human thermal comfort assessment, electronics cooling, chemical processes thermal control, etc. It has been thanks to the continuous improvement development as well as the new technology trends that these systems has become essential for many applications.

In order to increase the driving range of electric vehicle, while maintaining thermal comfort inside the passenger cabin, it is necessary to design a control system that simultaneously and optimally synthesizes multiple control actions of the vehicle HVAC system, while taking into account various constraints imposed by system HW and system performance requirements.

The traditional approach for vehicle thermal development relies strongly on experimentation and expertise. A virtual vehicle can be modeled to accelerate the control design phase allowing to explore virtual, but realistic, driving scenarios and the operation limits in a safer manner.

In the context of an automotive product development loop, this project is started with the aim of modeling an actual HVAC system by means of the Heat Balance Method from scratch, based on already analyzed and modeled HVAC architectures found in the literature.

This project also aims to ease the control design phase by providing some real-time simulation tools to expand the possible applications of the results obtained here.

Chapter 1

Introduction

The global automotive industry is constantly exposed to significant changes. Due to the technological innovative nature of its market, these changes must be adopted and properly managed in the design loops to maintain the market sharing and its profitability. A very visible example of this innovative nature and its rapid adaptation has been the development of new vehicle architectures such as the Electric Vehicle (EV) or Hybrid Vehicle (HV) integrated together with the Autonomous Vehicle (AV). It is expectable, then, that the development processes adapt themselves to these new trends to maintain the production capabilities and some new tools and methodologies arise.

Particularly, some analysis have been previously performed [1] and the development process can be classified according to the product complexity. In this work, there is an intrinsic internal product structure as well as its user-interface complexity, being part, then, of the so called *Complex Products* category.

There are three main imperatives in the automobile development process [2]: (1) *lead time*, (2) *productivity*, and (3) *quality*. Based on these, then, any automotive development process shall have as objectives to develop the project as quickly (1), as investment-efficiently (2) and as fitted to customer's requirements (3) as possible.

1.1 Automotive development process

Commonly, automotive development processes are defined linear in time since they have a strong sequential set of decision-making that affects to the development course. I.e., decisions made at the very beginning, when the project is not mature enough and it is still being defined, influence the available decisions later once the project is started and the resources, mostly time, are limited.

Moreover, the development process can be classified into different phases, typically six, as shown in Figure 1.1.

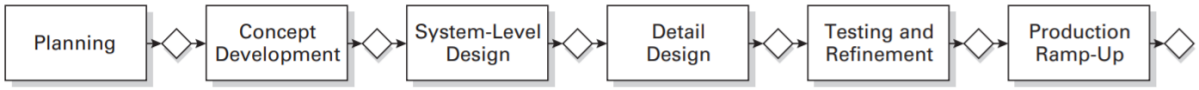


Figure 1.1: Generic automotive development phases [2]

1. **Planning:** in this phase, the processes are defined in time and the milestones, as well as the deadlines, are primarily established. The project management methodology is also established and the development process phases are defined. Project management methodologies all ultimately intend to accomplish the same objective: to complete and manage a project. Different approaches already exists:
   - Waterfall Project Management
   - Agile Project Management
   - Scrum Project Management
   - Kanban Project Management
   - V model Project Management
2. **Concept development:** it is defined and described the form, function and features of a product accordingly to its targeted specs.
3. **System-Level design:** it is defined the product architecture and the product is decomposed into functional subsystems (i.e. HW, SW and Mech). In this phase, it is also generated a list of functional specifications of each subsystem.
4. **Detail design:** it is specified the complete functional requirement and specification of every single unique part of the product fulfilling the corresponding standards. It is commonly generated a *control documentation* for the product.
5. **Testing and refinement:** in this phase, a product prototype (*alpha*) is built with no *production-intent* parts and it is tested to check the customer requirements fulfillment. Later (*beta*) prototypes are built with intended production processes parts. These prototypes are extensively tested to identify necessary engineering changes for the final product.
6. **Production:** at this stage, the product is manufactured using the intended production system and it is gradually transited to ongoing production evaluating possible development process improvements for future projects.

Note that the Figure 1.1 represents the previous defined project phases. They are sequential and need for a confirmation or an evaluation to pass (or not) to the following phase. In fact, this scheme is quite ideal since complex projects require a continuous revision of the planning, the concept might not be completely defined at the very beginning, the subsystems features might not be completely defined as well as the detailed design of each and so on. It is very common that the system review at each phase is done properly to contain any structural misalignment with the project specs avoiding cost overrun by rework appliance as pointed in [3], for instance. Thus, the different development process implementation types shall take this into account.

1.1.1 The V model project management

The *V model* project management methodology is based on an automotive standard procedure, the ASPICE3.1, defined in [21].

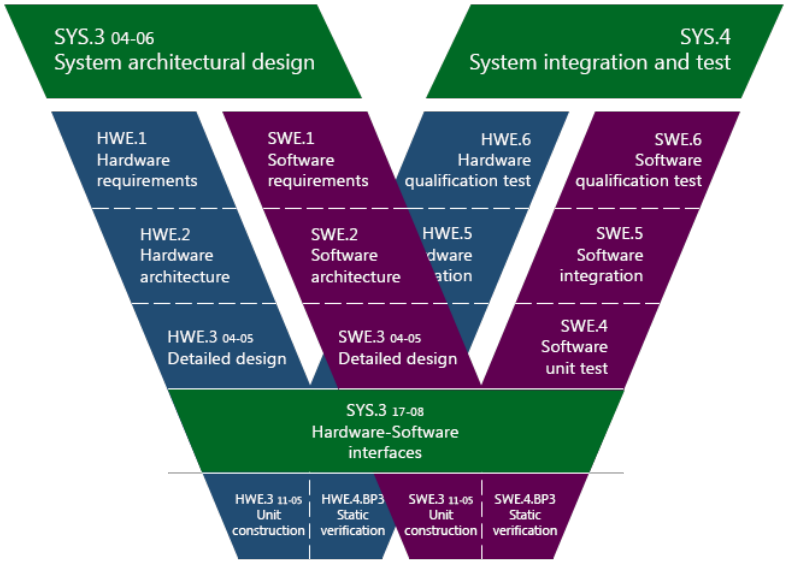


Figure 1.2: Idneo's HW process model. Based on ASPICE3.1 [21]

This management model contains every single phase already exposed in Section 1.1 with the particularity that the architectural design is aligned with the system integration tests so that each phase is well defined and well evaluated according to the standard. Note that the management complexity meets the product complexity. Since the working structure and the path are standardized, the documentation of the project can also be predefined and templates are used allowing a great tracking and reporting of the project.

Within the context of a more complex automotive project based on this project management methodology, this work is carried out. This mentioned project consists on the development of an Electronic Control Unit (ECU) responsible for the monitoring and controlling of various electronic accessories in a particular vehicle body, a so called Body Control Module (BCM). Note that the project is complex itself from every engineering point of view but also notice that the proper HW, SW and Mech design must be aligned in each development phase.

1.2 Systems modeling

As already mentioned, new design challenges arise with new technologies at the same time the used tools are constantly updating. It is proved that a given technology is mature when there are enough production capabilities as well as proper designing tools. A simple proof of that is the detail level of some simulation tools that involve general engineering knowledge (from physics to computation theory) that allow to ease the prediction of the product performance, such as SPICE based or electromagnetic simulators. In the system engineering field, it is possible to take really advantage of these tools to model, for instance, a particular system.

Taking our scenario as example, the proper modeling of the different BCM subsystems eases the testing phase but also allows to design some higher level architectures. These are, commonly, difficult to mount since require a functional prototype with all HW, SW and, most important, Mech parts. Thus, some anticipation is gained since it is not required a functional prototype of some subsystem but its model instead and there is, in general, a major reaction capability. The prototype fine-tuning spent time is also reduced. Moreover, a representative model of a particular subsystem allows to design it with an extended flexibility since the models can often be parameterized increasing the abstraction and allowing the combination of running these models with actual HW.

In spite of having good simulation tools, system modeling can eventually slow down the development phase due to the system particularities. Therefore, a proper pre-evaluation and planning is required. This is the main drawback of system modeling. In some cases, there is not enough detailed know-how about the subsystem performance or even is not worth it to model in deep detail. Then, the decision of implementing or not a representative model of a particular system must be taken carefully according to the design needs and requirements.

1.3 In-the-Loop testing

It has been already exposed a clear methodology to develop automotive projects with a clear structure. However, in the later years it has arisen different tools that have permitted to design and test in a more efficient way than in the past. The importance of the SPICE simulators and the appearing of the so called Big Data, have brought a new era in which the simulation capabilities and the processing of these simulations take and important place in the design loop and every design phase. Particularly, in-the-loop design breaks the independent phases methodology since it is allowed to emulate some structural subsystem of the product to test an already implemented one. These techniques are suitable to develop complex multi-system projects in a short period of time within a very restrictive deadline. It is also a competitive advantage since the product quality can be notably increased.

Due to the automotive industry characteristics, in-the-loop techniques allow to validate and fine-tune subsystems as well as behavioral simulations can be perform in the SW/HW integration phase. With a proper interfacing model, each functional block in the system can be tested and simulated before the different blocks are physically connected and this degree of abstraction allows to pre-validate some system parts before the project goes on. The impact of undetected issues in higher stages of the V model produces a notorious cost and time impact. Moreover, while the project is going on, the decision-making capabilities decrease and the defect fixing in later project phases shall be carried out with a more restrictive set of possible solutions. Thus, the project risks can be reduced using these kind of techniques thanks to their strong validation and design defect detection features.

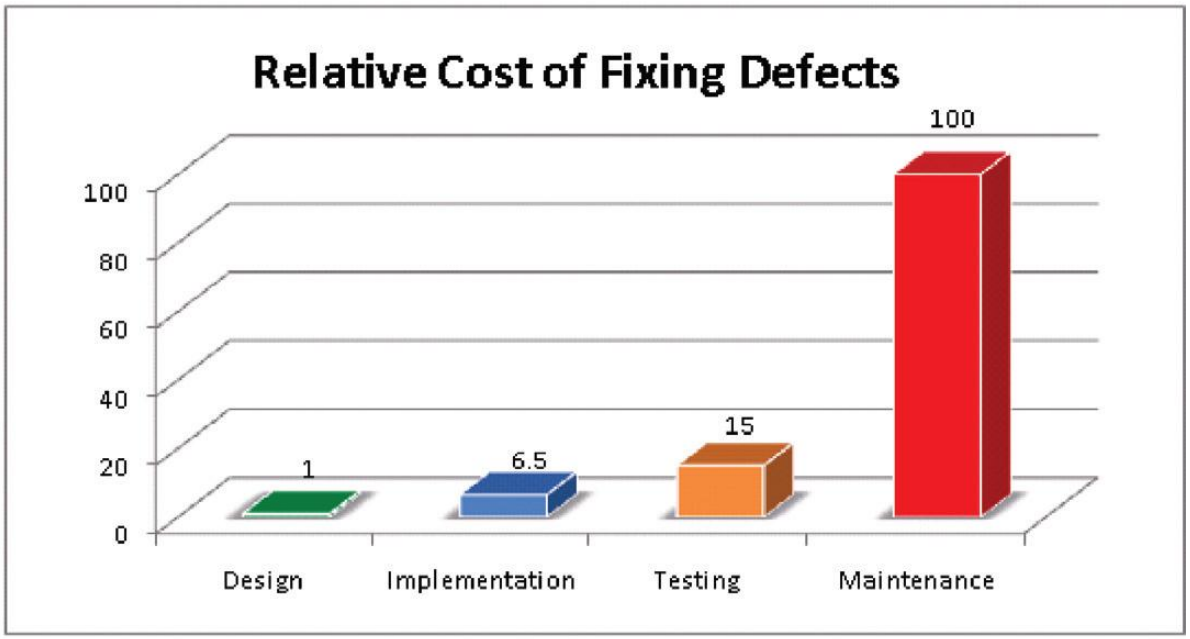


Figure 1.3: Relative Cost of Fixing Defects according to [4]

Is because of the relevance and importance of these testing techniques based on Hardware-in-the-Loop (HIL) that the project development is driven such that the results can be furtherly used for HIL testing.

1.4 Project scope and objectives

The presented work is developed within a BCM design product loop at Idneo's. It is aimed to implement the control architecture for one of the most known vehicle accessories, the HVAC system. This subsystem consists of the electronic HW for proper signal conditioning of the Mech actuators as well as the required SW control architecture managed by the ECU microcontroller.

It is also intended to design the control architecture to ease the later parameter tuning when the prototype is built with all mechanical structure together with the BCM. Is here, then, when this project starts whose main objective is to develop a representative and functional system model able to emulate the Heating, Ventilation and Air Conditioning (HVAC) elements driving its development to a real control design and testing scenario.

To sum up, this project tries to cover the following topics:

- Thermal processes research
- HVAC system conception
- HVAC model exploration
- Vehicle HVAC models research
- Modeling tools definition
- Real-time simulation tools research
- Feasible control strategies and different approaches
- Real-time simulation testbench definition

Chapter 2

State of the Art

The HVAC is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and a good enough indoor air quality. The design of an HVAC system is a sub-discipline of mechanical engineering based on thermodynamics, fluid mechanics, and heat transfer principles. This project has as one of its main objectives to model the vehicular cabin in thermal terms, thus, the heat transfer and thermodynamics principles are explored.

The HVAC system is mainly composed by three elements:

- **Heater:** is an appliance which purpose is heat generation. The heat is generated in the heating stage (mechanical room, furnace room, boiler, high power resistor, etc.) and it is transferred by convection, conduction, or radiation. Once the heat is generated within the system, it must be distributed so as the dict drives the heat and reaches the zones of interest. In vehicular HVAC systems, the heat transfer is carried out by convection employing either a high power resistive element (such as a PTC resistor) or a water system heated by high temperatures generated within the engine structure.

- **Ventilation:** it is the process of changing/replacing air in any space to control the temperature and to `clean` the air. Thus, in HVAC system in vehicles, ventilation is used to translate the heat exchange withing the HVAC cycle into cabin thermal comfort. The expectable variables involved are: the airflow temperature, the airflow rate and the air quality in terms of humidity and oxygen concentration. Therefore, the inlet air has the properties so that the thermal comfort is achieved in the cabin at the same time the outlet air is in charge of reentering in the HVAC cycle as well as exiting the system to let the system add a `fresh` air rate (i.e. oxygenizing)

- **Air Conditioning:** it provides cooling (removal of heat though a heat exchanger using refrigerant) and humidity control. By a reversing valve, the refrigerant cycle is reversed, and it can be used as a heat sink or source. The most common cooling technique, also used in vehicular systems, employs a compressor, some valves to assure different refrigerant pressures within the cycle and two heat exchangers (an evaporator for cooling and a condenser for heating). The process is as follows

    1. The liquid refrigerant at low pressure and temperature enters to the evaporator. Due to the temperature difference between the refrigerant and the outside air, there is a

heat transfer from outside air to the liquid refrigerant. The inlet air of the air circuit delivers heat to the evaporator in the form of convection to cool down the air. The refrigerant boils and reaches the compressor completely in gas state.

2. The gas refrigerant is compressed to increase its temperature increasing also its pressure. The compressor is the system element that performs a work to move heat from the cold and low pressure side (evaporator) to the hot and high pressure side (condenser).
3. The compressed refrigerant enters to the condenser and the heat transfer is backwards, from refrigerant to outside air. The condenser must be placed away the evaporator since air temperature can reach up to 100 ºC and must be thrown outside the vehicle or used as the air heating process. In this process, the refrigerant completely changes to liquid state and starts the cycle again.

2.1 Actual vehicular HVAC systems

First cooling and heating systems appeared in the third century B.C. [5]. It has been tkanks to the better understanding of the physics involved and the technology improvements over the time that we can employ the sophisticated HVAC systems nowadays. Actual HVAC systems know-how involve several physics and engineering disciplines. In fact, even though the systems became more complex over the time, they are all based in well-known principles.

Modern system were developed with a deep understanding about how they physically are and how to control them in the past century, when building HVAC systems appeared. Even nowadays more complex HVAC systems are available able to assess thermal comfort, they have mostly inherited building air conditioning knowledge and often, system modeling is consequently referred to them. Actual HVAC systems are found composed of different elements. An example of this is that the system elements are strongly dependent on the application itself and on the operating conditions, i.e. Mars rover electronics temperature control systems. Heat exchanging might be challenging in some extreme contexts, just focusing on human comfort assessment systems, the implementation of building or vehicular HVAC differs although they have the same main objective. In automotive environments, these kind of systems have been developed and well-designed in detail since they face several conditions that limit, partially, the overall performance.

One of the main constraints in vehicular HVAC systems is the available space for the system, this mechanical constraint forces to maximize the system efficiency for the given actuators since system elements bigger than required are completely unnecessary. This requires a good mechanical design of the system. For this reason, it can be widely found innovative control solutions. Under heating conditions, the heater element, commonly a high power resistor, heats up and an air flow is driven to the cabin thanks to the blower. Otherwise, under cooling conditions, the refrigerant circuit is enabled and it assures the evaporator absorbs heat driven also by the blower, as shown in Figure 2.1.

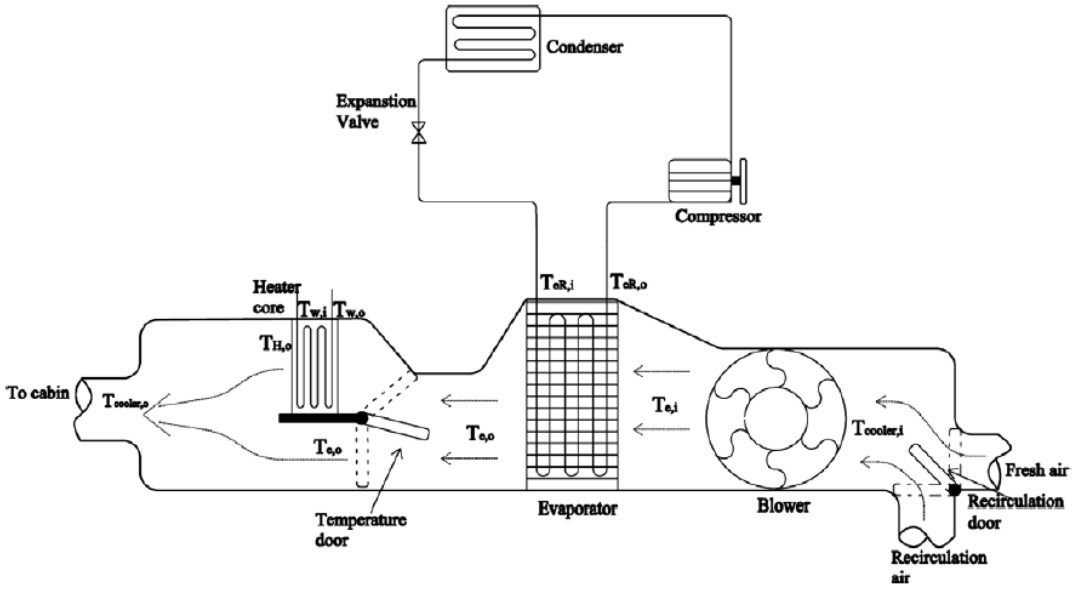


Figure 2.1: Vehicle cabin and air conditioning system schematic [8]

Another constraint is related to the overall power consumption and, accordingly, the battery usage. With the appearance of new technologies that have allowed the real implementation of EV the battery management and consumption has became important in the design loop. In fact, some analysis and simulations [6] report the direct impact on vehicle driving performance. It is the second most power consuming vehicle system, after the powertrain, and it can impact the driving range up to 60% in some given weather conditions. The designed control, thus, shall be capable of optimizing the interesting comfort indexes as well as the system efficiency. The trade-off between thermal comfort performance and autonomy shall be addressed and some techniques appear [10] [11]. The building application knowledge is inherited and control is oriented according to standard Thermal Comfort Indexes (TCI). In fact, an advantage of these TCI-based controls is their less energy and/or fuel consumption at the expenses of complex computation algorithms and a more complex control scheme.

2.1.1 Thermal comfort assessment

Thermal comfort can be defined as `that condition of mind that expresses satisfaction with thermal environment` [1]. This leads to that comfort cannot be defined in absolute terms and there will always be person-to-person discrepancies. The state of thermal comfort felt is strongly related to how physically and psychologically a person is. Thus, a challenge has been faced in the latest times to overcome this, to derive a representative thermal comfort index. The most extended and used one was introduced by Fanger, the Predicted Mean Vote (PMV). Other thermal comfort indexes have been developed to overcome new challenges such as the computation time but PMV still is the most used.

PMV is an index that expresses the quality of thermal environment as a mean value of the votes of a large group of people on the American Society of Heating, Refrigeration and Air-conditioning Engineers (ASHRAE) seven-point thermal sensation scale (+3 hot, +2 warm, +1 slightly warm, 0 neutral, -1 slightly cool, -2 cool, -3 cold). Since this index is based on the mean vote of some significant group of people, some people's vote might differ. Predicted Percentage of Dissatisfied (PPD) is an index expressing the thermal comfort level as a percentage of thermally dissatisfied people, and is directly determined from PMV. Thus, PMV expression defines the mean thermal comfort state under the given conditions and for that PMV it can be estimated the percentage of dissatisfied people by means of the PPD computation. The ANSI/ASHRAE Standard 55 states that the comfort range is $-0.5 < PMV < 0.5$, this is the PMV range in which the percentage of dissatisfied people is below 10% ($PPD(|PMV| < 0.5) < 10\%$, equation 2.2), as depicted in Figure 2.2.

[1] According to ANSI/ASHRAE Standard 55: Thermal Environmental Conditions for Human Occupacy

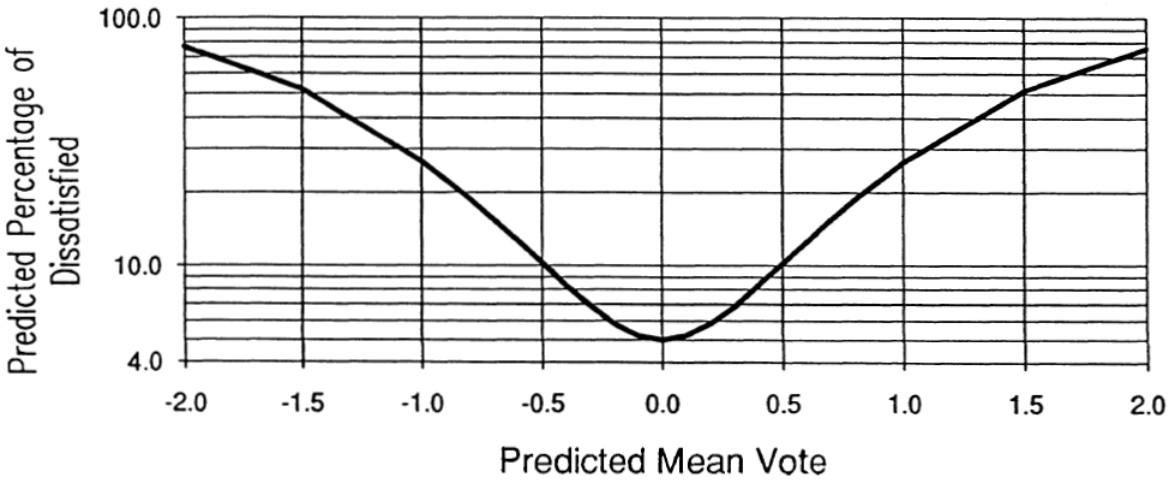


Figure 2.2: Predicted Percentage of Dissatisfied as a function of the Predicted Mean Vote

PMV is inherited from thermoregulation and heat transfer of the human bodies analysis. There are some factors that take into account according to equation 2.1.

External PMV factors:

- Air temperature
- Radiant mean temperature
- Partial pressure of water vapor
- Relative air velocity

Individual PMV factors:

- Clothing insulation
- Activity level
- Average skin temperature

$$
\begin{aligned}
PMV = (0.303e^{0.303} + 0.028)\{(M-W) - 3.05[5.73 - 0.007(M-W) - P_a] - \\
-0.42[(M-W) - 58.15] - 0.0173M(5.87 - P_a) - 0.0014M(34 - T_a) - \\
-3.9610^{-8} f_{cl}[(T_{cl}+273)^4 - (T_{mr}+273)^4] - f_{cl}h_c(T_{cl} - T_a)\}
\end{aligned}
$$

$$
h_c = \begin{cases} 2.38|T_{cl} - T_a|^{0.25} & \text{for } 2.38|T_{cl} - T_a|^{0.25} > 12.1\sqrt{v_{air}} \\ 12.1\sqrt{v_{air}} & \text{for } 2.38|T_{cl} - T_a|^{0.25} < 12.1\sqrt{v_{air}} \end{cases} \tag{2.1}
$$

$$
f_{cl} = \begin{cases} 1.00 + 1290I_{cl} & \text{for } I_{cl} <= 0.078 m^2K/W \\ 1.05 + 0.645I_{cl} & \text{for } I_{cl} > 0.078 m^2K/W \end{cases}
$$

$$
\begin{aligned}
T_{cl} = 35.7 - 0.028(M-W) - I_{cl}\{3.9610^{-8} f_{cl}[(T_{cl}+273)^4 - \\
-(T_{mr}+273)^4] + f_{cl}h_c(T_{cl}.T_a)\}
\end{aligned}
$$

$$
PPD = 100 - 95e^{-0.03353PMV^4 - 0.2179PMV^2} \tag{2.2}
$$

Where:

M: passenger metabolic rate [met, W/m^2]

W: passenger effective mechanical power [W/m^2]

P_a: water vapor partial pressure [Pa]

T_a: cabin temperature [ºC]

T_{mr}: mean radiant temperature [ºC]

T_{cl}: clothing temperature [ºC]

f_{cl}: clothing factor

I_{cl}: clothing insulation [m^2K/W]

h_c: convective heat transfer coefficient [W/m^2K]

v_{air}: cabin air velocity [m/s]

The assumptions taken are: the air in the cabin is ideal and incompressible, the concentration of organic compounds of CO_2 that affect cabin air quality are neglected, the intensity of turbulent flows in the cabin is low and only sensible heat dissipated by the human body with even distribution is considered.

Although the PMV is computed according to six variables, some of them can often be assumed constant, i.e. the clothing insulation is not expected to vary along a driving cycle. Besides, the comfort range can be achieved by a combination range of some variables, such air velocity, cabin temperature and cabin relative humidity (related to water vapor partial pressure).

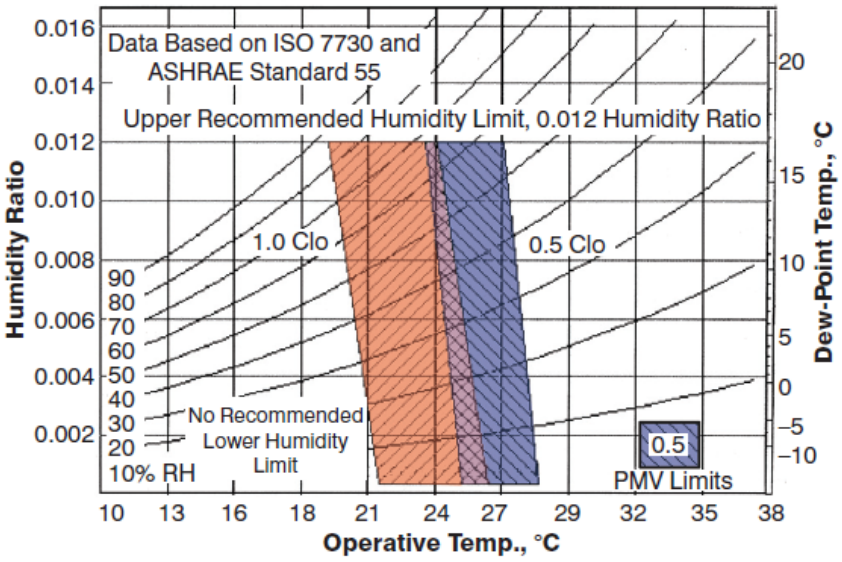


Figure 2.3: Acceptable range of operative temperature and humidity (|PMV| < 0.5)

2.2 Actual vehicular HVAC models

Since the system modeling has become a need for the proper system design loop, some modeling techniques have been developed according to these needs to fit the automotive design workflow. As already mentioned, HVAC system models have been deeply developed when their main applications were building thermal comfort. Thus, there is a deep knowledge available in the literature of the very basic physics involved and white-box models arose. However, these models might not fit with actual automotive design projects since the effort required to implement a representative model based on the physic principles sometimes become time consuming. In contrary, black-box models have been used thanks to system identification techniques together with sophisticated simulation tools, i.e. Matlab/Simulink system identification toolbox. The clear advantage of these techniques is the behavioral understanding of the system but, in some cases, this abstraction level derives to lack of understanding on the system processes and, eventually, a mixed modeling technique is required. The grey-box models fit these modeling requirements since they allow to implement a representative model based on already known principles and it is let to later project stages the fine tuning of some model parameters once the prototype is implemented. By doing so, a first guess of system dynamics is developed and the system is modeled letting some freedom degree avoiding the necessity of the prototype implementation accelerating the modeling process (see [9] for a detailed review on HVAC system modeling types). Models can also be classified as linear or nonlinear, static or dynamics, explicit or implicit, discrete or continuous, deterministic or probabilistic, and deductive, inductive or floating models. In some cases, some assumptions can be taken and the implemented model might consist of several subsystems with their own particularities, i.e. it is used a dynamic model for the slow moving temperature process inside the cabin at the same time that it is assumed static processes those which dynamics are fast enough compared to cabin dynamics, such as refrigeration cycle regulation. Thermal loads are commonly modeled with lumped-model schemes and the rigorousness of each model depends directly on the modeling intention. For example, in [13] it is discussed the impact of the air flow pattern to assess cabin thermal comfort, which lead to a more complex overall model. Another example is the cabin air leakage model, it is addressed in some works [14] and it has a huge impact when it is missed in a particular system. However, these works conclude that is not feasible in some cases to model every single detail in spite they can compromise the overall model behavior. Instead, it is tried to minimize these effects by assuming some acceptable constant leakage and avoiding some cabin zones with poor air circulation.

ASHRAE Handbook of Fundamentals [15], provides two main thermal load calculation strategies: Heat Balance Method (HBM) and Weighting Factor Method (WFM). HBM provides a more accurate and scientific approach and fit with several fundamental models, based on Newton's law of cooling, that can be incorporated in the thermal calculations, as depicted in Figure 2.4. By employing this method, heat sources can be modeled according to the cabin interactions considering the vehicle geometry as well as its environment. Under this method, it is stated that all thermal loads in the system are passive with their respective gain contribution to the overall thermal equilibrium. Thus, heat balance procedures calculate the heat exchange between surfaces based on their surface temperatures and emissivities and they typically rely on estimated splits to determine the contribution of internal loads. The point of using this method is that each heat source or sink interfaces the cabin thermal state proportionally to temperature differences and this simplifies the model construction.

On the other hand, WFM are based on statistical weighting factors that shall be deter-

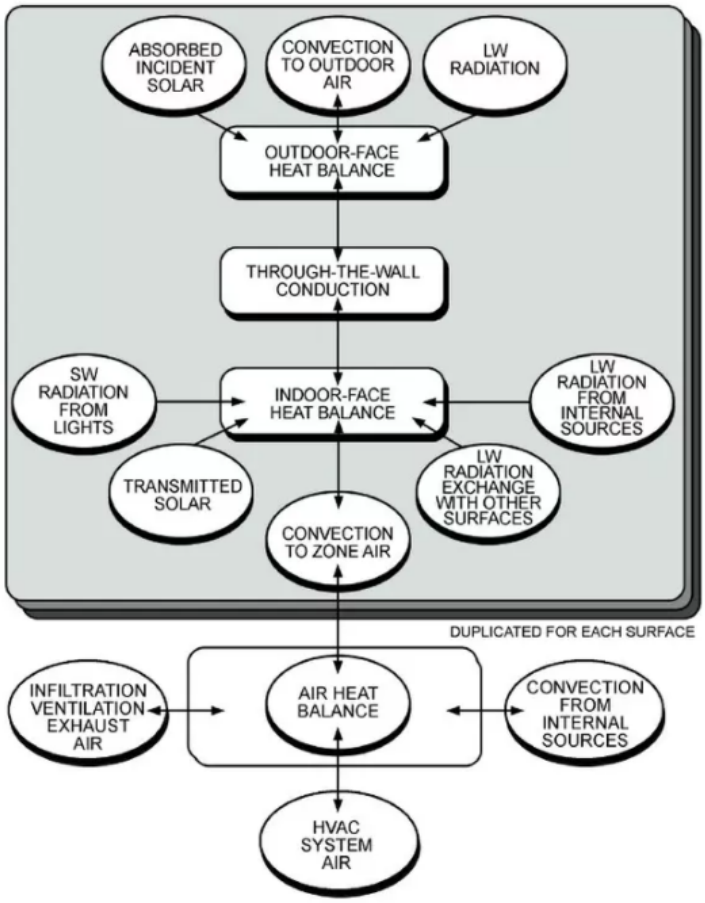


Figure 2.4: Schematic of Heat Balance Processes in Zone

mined according to thermal spatial distribution within the cabin. This method often require 3-dimensional (3D) data. The thermal load computation becomes easier in passenger vehicles. This method simplifies the heat transfer expressions at expenses of adding model abstraction. In fact, it is not commonly used in cases in which the vehicle geometry is well defined since the HBM is able to model the system more accurately and the WFM is difficult to adapt in the cases in which there are several independent block models without updating the already obtained weights. An example of these weighting factor can be found in [15] - Table 24, where the weighting factors are different for each considered time slot.

Regarding the feasibility of expressing the system dynamics in a closed form, lumped, it shall be fulfilled some requirements. In order to avoid modeling boundary conditions and different thermal zones, it shall be demonstrated that the heat flow within a body is much faster than the interaction with the surroundings. The parameter that embeds this information is the so called Biot's number.

$$B_i = \frac{\text{Thermal resistance offered by the interior of the cabin}}{\text{Thermal resistance at the cabin interface}} = \frac{hL_c}{k_b}$$

Where:

- B_i: Biot's number
- h: heat transfer coefficient between body element to exterior
- L_c: body characteristic length, commonly defined as ratio between the cabin volume and surface $L_c = \frac{V_{body}}{A_{surface}}$
- k_b: body thermal conductivity

For Biot's numbers greater than 1, the interaction with environment occurs easier than within the cabin. In those cases, the temperature inside the cabin cannot be assumed uniformly constant. This is not the case since the intended air flow rate managed by the ventilation subsystem or even the air leakage ensures the rapid heat propagation and eases the uniformity temperature within the cabin. Even the body elements present a Biot's number smaller than 1 since the chassis and windows are thermal conductors and the convection heat exchange with the environment occurs slower than the thermal conduction within the body elements. Thus, the overall system consists in two well-defined thermal masses, cabin air and body elements.

The most challenging model block of actual vehicle HVAC systems is the refrigeration cycle. Vapor compression cycle models are already available in literature but usually these models deal with up to 12 state variables and lumped-parameter nonlinear processes. Thus, often it is applied a model-order reduction [17] by introducing certain assumptions and replacing state variables associated with fast dynamics with static expressions. Besides, the four-stage thermodynamic process is quite performance varying since it depend on the actual operating conditions, such as outside temperature, relative humidity, etc. Thus, it is often found internal control loops within these actuators that ensures the proper working conditions. Sometimes, it can be assumed that the HVAC actuators are able to auto-regulate themselves so that they optimize the heat exchange efficiency. In other cases, it is required to model carefully the compression cycle since the intention of the model is to optimize the control to fulfill some requirements.

To sum up, it can be found several cabin models in literature depending on the intention they have. The definition of how representative a model is became confusing, thus, all available works in literature refers to the direct intention of the model. For control optimization it can be found behavioral lumped models that try to emulate the temperature dynamics within a vehicle cabin as well as the system performance parameters are extracted and used in the control law [11] [16]. Some models have the intention of studying how flow rate affects to passenger thermal comfort [14] and others try to reduce energy consumption by adding energy management systems [10] so the model implemented computes the required elements.

Chapter 3

Methodology

As already exposed, the HVAC system model shall be coherent with its usage. In this study case, it is intended to provide a representative model for close-loop control design oriented to real-time testing, i.e. able to run in-the-loop. Thus, under this context, the model is implemented taking these considerations.

A parameterized lumped model is chosen as system model method given that a functional system prototype is not yet available. It is used the HBM method to model the system dynamics. At the same time, it is let some freedom degree in the model so that not only the system parameters can be later tuned but also the control variable can be externally selected. This eventually fulfills the project model requirements.

The model consists of four main blocks: (1) the plant, (2) the thermal loads, (3) the comfort estimate, and (4) the controller. The plant (1) can be split into the cabin dynamics and the HVAC actuators. According to the heat balance in the cabin, both the cabin and the body elements temperature dynamics can be modeled and a functional HVAC actuator's model is defined. To simplify its model they are assumed linear within their saturation limits and with self-controlling subsystem with enough Bandwidth (BW) so they are assumed static. The considered thermal loads (2) follow the scheme of Figure 2.4. Where the main heat sources are: the solar radiation, the environmental conditions, any vehicle element able to transfer heat and vehicle occupancy. The comfort estimate (3) has been already introduced. Particularly it is used the PMV algorithm to estimate the thermal state of the cabin and also it is included the cabin temperature as feedback variable to explore the system performance in both cases. Finally, the controller (5) block is proposed according to the system control variables available with the given HVAC actuators.

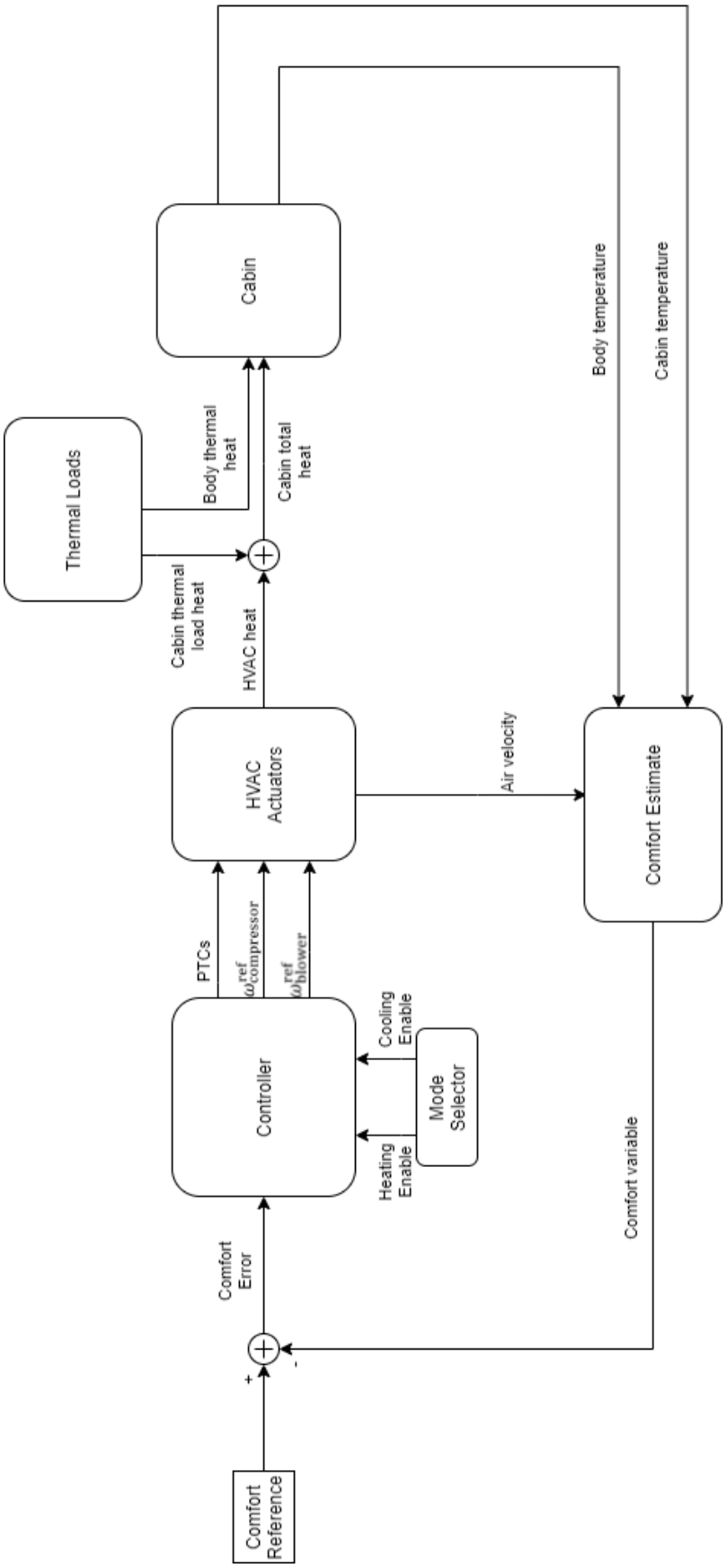


Figure 3.1: Proposed HVAC system block diagram

3.1 Cabin thermal loads

The HBM is used to model the system thermal dynamics, thus, the heat balance is achieved once all heating elements are considered and the state variables evolve to reach that equilibrium point. It can be understood the cabin thermal process as two independent processes with its own state variable each. The cabin temperature will rise or fall to reach the thermal equilibrium when all cabin thermal loads stabilize at the same time the body elements temperature does. It is important to take these second dynamics into account since it cannot be assumed that the cabin is directly in thermal contact with the thermal loads, the body elements are the thermal interface between them with a notably thermal mass with enough thermal capacity to alter the air cabin thermal dynamics. This leads to a second-order system in spite of each thermal process is expected to have an exponential, thus just first derivative involved, behavior, the cabin thermal process dependence on the body thermal process increases the system order.

Some authors have developed different thermal load models considering different heat sinks or sources [11] [8] [10]. The ones used in this work are:

- **Solar Radiation Load**: the sun is the main heat source considered. The entire vehicle is exposed to sun radiation within sunlight hours. There is a heat gain in the body elements that rises up the body temperature at the same time that some vehicle surfaces transmit almost completely the sun heat due to the poor thermal isolation capabilities of some surfaces such as glasses. Therefore, cabin temperature is affected due to sun radiation by these two mechanisms, direct sun radiation is heating the cabin air and at the same time body thermal mass is storing heat that might eventually interface the air cabin.

- **Metabolic Load**: human body is a heat source for environments under the body temperature but also it is assumed in hotter environments. It can be modeled how much heat can dissipate a person by knowing his/her metabolic rate, depending on the activity level. This thermal load is directly interfacing the air cabin so that the body elements might not be affected.

- **Ambient Load**: since it is expected a temperature difference between cabin and ambient air, a heat transfer is expected to occur. Hot environments will heat up the cabin air through the body elements that, in fact, they will also gain heat. In cold environments, the process is backwards. Therefore, it is expected that outside temperature affects air cabin equilibrium temperature but also body elements might gain or loss heat in this process.

- **Internal High Temperature Elements Load**: some high temperature elements in the vehicle might interface thermally the cabin state. The considered ones are the engine and the exhaust temperatures. Depending on the geometry and location of those, they will contribute accordingly to cabin state. In this case, although there is thermal conduction in some surfaces that thermally interface the heating elements and the air cabin, it is not considered an increase of body temperature in this process since geometrically it is only involved one single body surface and, as it will be discussed later, the contribution of these loads can be neglected.

- **Ventilation Load**: due to there is some fresh air reentering the cabin due to leakage and this air has the outside properties, it might exists a heat transfer according to this phenomena. In this case, the heat transfer involves just the cabin thermal process and not the body elements.

- **Body Elements Heat Transfer Load**: this thermal load is the model interaction due to the temperature difference between the body surfaces and the cabin air. Since both thermal masses are different and different thermal loads are involved, they would exchange heat. Is in this process in which the cabin state presents second-order dynamics since the derivative of the cabin temperature will depend on the cabin temperature itself but also on the body temperature dynamics. It can be modeled as the heat gain of the cabin thermal mass and the heat loss of the body elements thermal mass at the same time. Thus, the same heat quantity but with different sign can be modeled for both processes.

- **HVAC Load**: the HVAC actuators' action contributes to the overall cabin thermal process providing a control freedom. To cool down the cabin air, heat is absorbed by the cooling system and to warm up the cabin air, heat is supplied by the heating system. In other words, giving some driving and environmental conditions, this load is in charge of assuring a desired cabin thermal comfort state by either supplying or sinking heat from cabin thermal process.

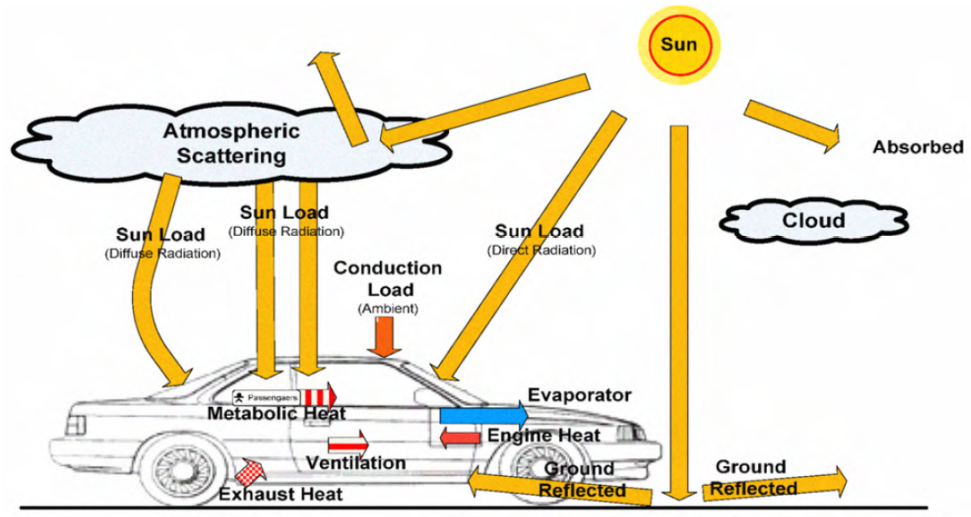


Figure 3.2: Vehicle cabin energy flow diagram [10]

The total thermal loads heat, thus, can be expressed as:

$$\dot{Q}_{cabin} = \dot{Q}_{rad} + \dot{Q}_{met} + \dot{Q}_{amb} + \dot{Q}_{exh/eng} + \dot{Q}_{vent} + \dot{Q}_{body} + \dot{Q}_{HVAC} \tag{3.1}$$

$$\dot{Q}_{surf} = \dot{Q}_{rad,body} + \dot{Q}_{amb,body} - \dot{Q}_{body} \tag{3.2}$$

3.2 HVAC actuators

The HVAC actuators' model are intended to represent the contribution of each to the total thermal cabin state, by the proper control variables. In order to simplify the overall model and reusing previous results in the field of vapor cycles modeling and system order-reduction [17], it is assumed that the cooling and heating processes are performed under self-controlled scenarios and the HVAC actuators work properly under the conditions exposed here. The self-regulation parameters, i.e. overtemperature protections or superheating range, are supposed to be either already controlled so that the overall system assumes they operate statically or always operating under the operation range. The HVAC actuators involved are:

- Heating actuators:
  - **PTC Heaters**: a resistive element that dissipates heat when they are powered up. Since they have a positive temperature coefficient, the resistance will decrease as electric power is consumed by these elements until an equilibrium point is reached. It is used the expressions empirically tested in [18].
  - **Blower**: the heating process occurs mainly by convection, the heater element provides a high temperature element that, by natural convection can transfer heat from it to the surrounding air. By employing a blower, forced convection occurs and the heat rate transfer increases according to the supplied air speed by the blower. This element is also in charge of assuring air circulation in the HVAC heating cycle.
- Cooling actuators:
  - **Compressor**: the compression cycle has been already introduced. With the assumed simplification, the compressor can be understood as the cooling element that produces certain refrigerant mass flow rate able to sink heat from surrounding air. From the heat exchange point of view, refrigerant absorbs heat from the air and the compressor, among other neglected tasks, is in charge of generate the refrigerant mass flow rate available at the evaporator. Thus, the compression action determines how much heat per unit time can be sank from surrounding air. By natural convection, this process cools down inlet air.
  - **Blower**: the same blower is used for both cooling and heating scenarios with the same objective, to help by forced convection the heat loss or gain and drive it to the cabin.
- Other actuators:
  - **Expansion Valve**: even though it really belongs to the compression cycle it is not included in its model. However, it is worth it to mention its usage: assure proper pressure at high and low pressure sides within the compression cycle. It also blocks the pass of refrigerant when the cooling system is off so that the compression cycle can be started as quick as possible.
  - **Temperature Door**: it is a mechanical door able to select the air flow path that will reach the cabin. Under heating conditions, the air moved by the blower is allowed to pass through the heater core. Otherwise, the cooled air passes directly to cabin. It is

assumed that its state is static.

- **Recirculation Door**: it is also a door able to select the inlet air, either from the cabin, so that it has the actual cabin air properties, or from the outside, with the outside air properties. It is also assumed to have static state.

The HVAC model is based on the exposed elements and the configuration shown in Figure 2.1.

3.3 HVAC operating modes

It can be defined several operating modes according to the system requirements. Due to the already commented mechanical limitations, the system typically is design to provide not only cabin thermal comfort but is also used for other purposes, i.e. electronics cooling or windshield defrizzing. Instead, in this work is only considered cabin thermal comfort applications. The considered system states are:

- **Heating state**: in this state the cooling system is switched off by means of stopping the compression cycle. The heating system is switched on letting the PTCs warm up. The blower controller manage the blower speed accordingly. The temperature door drives the blower air to the cabin through the heater core.

- **Cooling state**: in this state the cooling system is switched on, thus, the compression cycle starts. the compressor controller manage the compressor speed to control the compression cycle as well as the blower controller manage the blower speed to cool down the cabin. The heating system is switched off by means of powering off the PTCs. The temperature door drives the air to the cabin directly.

Once exposed the system states, three operating modes are defined:

- **Cooling Mode**: in this mode, only cooling system is enabled. Under cooling conditions, the system is in cooling state and off otherwise.

- **Heating Mode**: in this mode, only heating system is enabled. Under heating conditions, the system is in heating state and off otherwise.

- **Auto-climate Mode**: in this mode, both cooling and heating systems are enabled. Under cooling conditions, the system is in cooling state and under heating conditions, system is in heating state.

In order to assure more robustness, cooling and heating conditions are also defined. Thus, the cooling and heating conditions are met with some hysteresis to avoid continuous state changes near the comfort state.

Chapter 4

HVAC system model

Up to here, the methodology followed, as well as the considerations and assumptions, have been exposed. The implemented model considers some thermal loads commonly present in actual driving scenarios and at the same time, it uses simplified, but functional system representation for proper control design. The system is mathematically modeled and it is implemented in Matlab/Simulink. These tools are able to simulate the implemented model and some tools are developed accordingly to interface the system model with some real thermal perturbations. For instance, thermal loads do not remain constant all over the time but some of them are time varying modeled to analyze a more realistic situation. Outside temperature and solar radiation might vary depending on location, time, etc. The model is also exposed to a real driving scenario so that the vehicle velocity changes following some patterns.

As already commented, the intention of the model is to design the proper actuator controllers. Besides, the model is also oriented to be used in real-time simulation tools, such as HIL environments. Thus, the model implementation effort is driven so that the model is not only able to run in real-time machines but also the required interface, both SW and HW, is eased.

Two equivalent models are derived. One for qualitative understanding on system dynamics under real driving conditions with highly nonlinear dynamics and a second based on the first but linearized. This second model will be used for control design purposes to, finally, simulate the overall nonlinear model with the designed control.

4.1 Cabin dynamics

The HVAC system consists of two different thermal masses: cabin air and surface elements (or body). The heat balance equation for those is as follows:

$$\dot{Q}_{cabin} = V_{air}\rho_{air}C_{p_{air}}\hat{T}_{cabin} \tag{4.1}$$

$$\dot{Q}_{surf} = m_{body}C_{p_{body}}\hat{T}_{body} \tag{4.2}$$

Where any $\dot{Q}$ is the total heat per time unit, V_{air} is the cabin air volume in m^3, ρ_{air} is the air density in kg/m^3, m_{body} is the equivalent body mass composed by each surface element with heat capacity, any $\dot{T}$ is the temperature variation in $°C$ and any C_p is the specific heat in $W/kg°C$. Equations 4.2 and 4.1 are expressed in continuous time. Load calculations are performed at time steps during the given simulation period t_s and, after every time step, all loads are summed up and the new cabin air and body element temperature can also be expressed as:

$$\dot{Q}_{cabin} = V_{air}\rho_{air}C_{p_{air}}\frac{\Delta T_{cabin}}{t_s}$$

$$\text{or alternatively} \tag{4.3}$$

$$\Delta T_{cabin} = \frac{\dot{Q}_{cabin}}{V_{air}\rho_{air}C_{p_{air}}}t_s$$

$$\dot{Q}_{surf} = m_{body}C_{p_{body}}\frac{\Delta T_{body}}{t_s}$$

$$\text{or alternatively} \tag{4.4}$$

$$\Delta T_{body} = \frac{\dot{Q}_{surf}}{m_{body}C_{p_{body}}}t_s$$

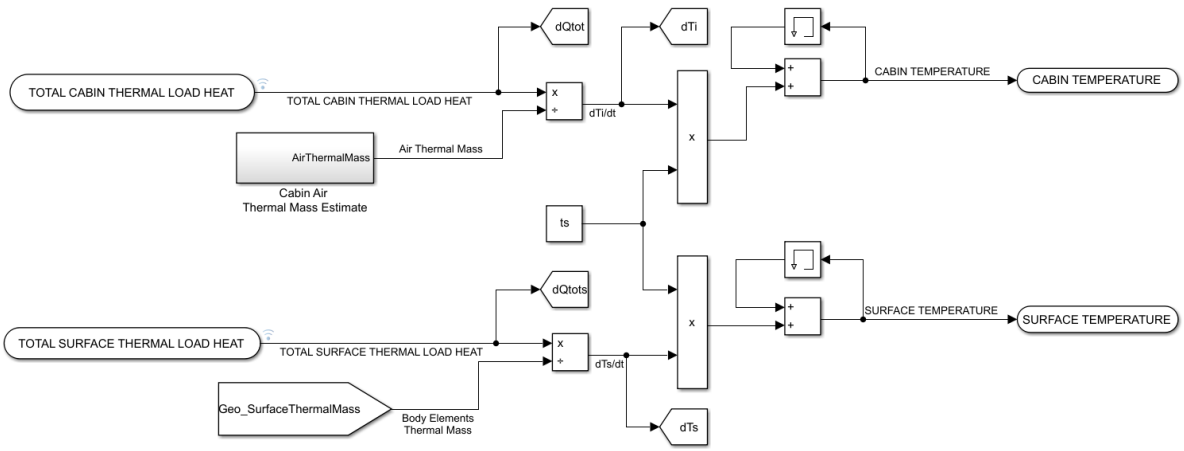


Figure 4.1: Air cabin and body elements discrete thermal dynamics

In Figure 4.1 is depicted the implemented Simulink logic. Equations 4.3 and 4.4 are implemented accordingly. The initial thermal state is defined in the memory block and states the initial thermal conditions, then, it stores the previous time step thermal state and adds up the temperature variations according to the exposed expressions.

Note that the former equations suggest that the system order is 2 as long as surface and cabin heats are linear with respect to cabin and body temperature. It can be also seen that cabin temperature dynamics are much slower than body temperature dynamics since the cabin thermal capacity ($V_{air}\rho_{air}C_{p_{air}}$) can be 10 times smaller than the body surfaces thermal capacity ($m_{body}C_{p_{body}}$). Depending on the thermal capacity ratio, the second order system can be order-reducted compensating with tuning factors the remaining simplification error. In this work, it is completely implemented the second order system.

4.2 Thermal loads

As commented in Chapter 3 the considered thermal loads are in charge of balancing the thermal system so that both cabin and body elements temperatures stabilize at certain point. These loads are expressed in equations 3.1 and 3.2. These thermal loads are cabin geometry dependent but also the driving conditions, as well as time and location, affect the thermal processes. Thus, a realistic and functional driving conditions are also modeled accordingly.

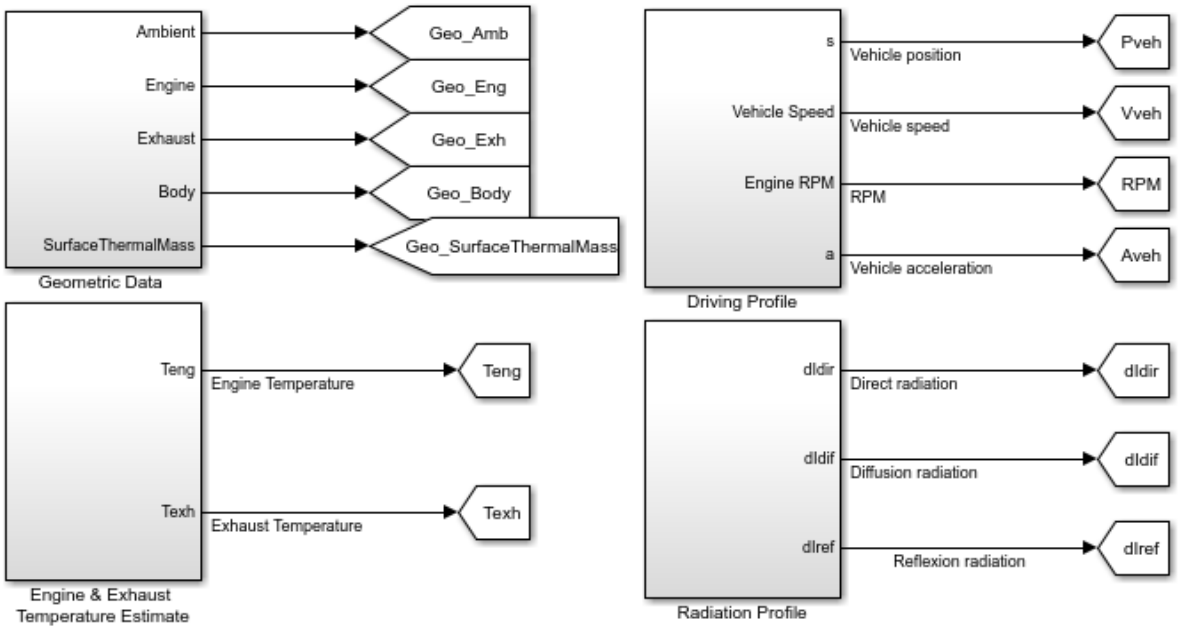


Figure 4.2: Simulink driving conditions and geometric data management blocks

Therefore, it is implemented Simulink blocks that produce these driving conditions and the geometric data is processed to generate the geometric constants, as depicted in Figure 4.2. The outputs of these blocks are the inputs used for thermal load calculations together with actual thermal states. Some of them are constants and some of them are time varying Simulink signals that emulates vehicle velocity or instantaneous solar radiations. Detailed information about these block implementations can be found in Appendix A. Driving conditions.

The Simulink implementation of the thermal loads is depicted in Figure 4.3.

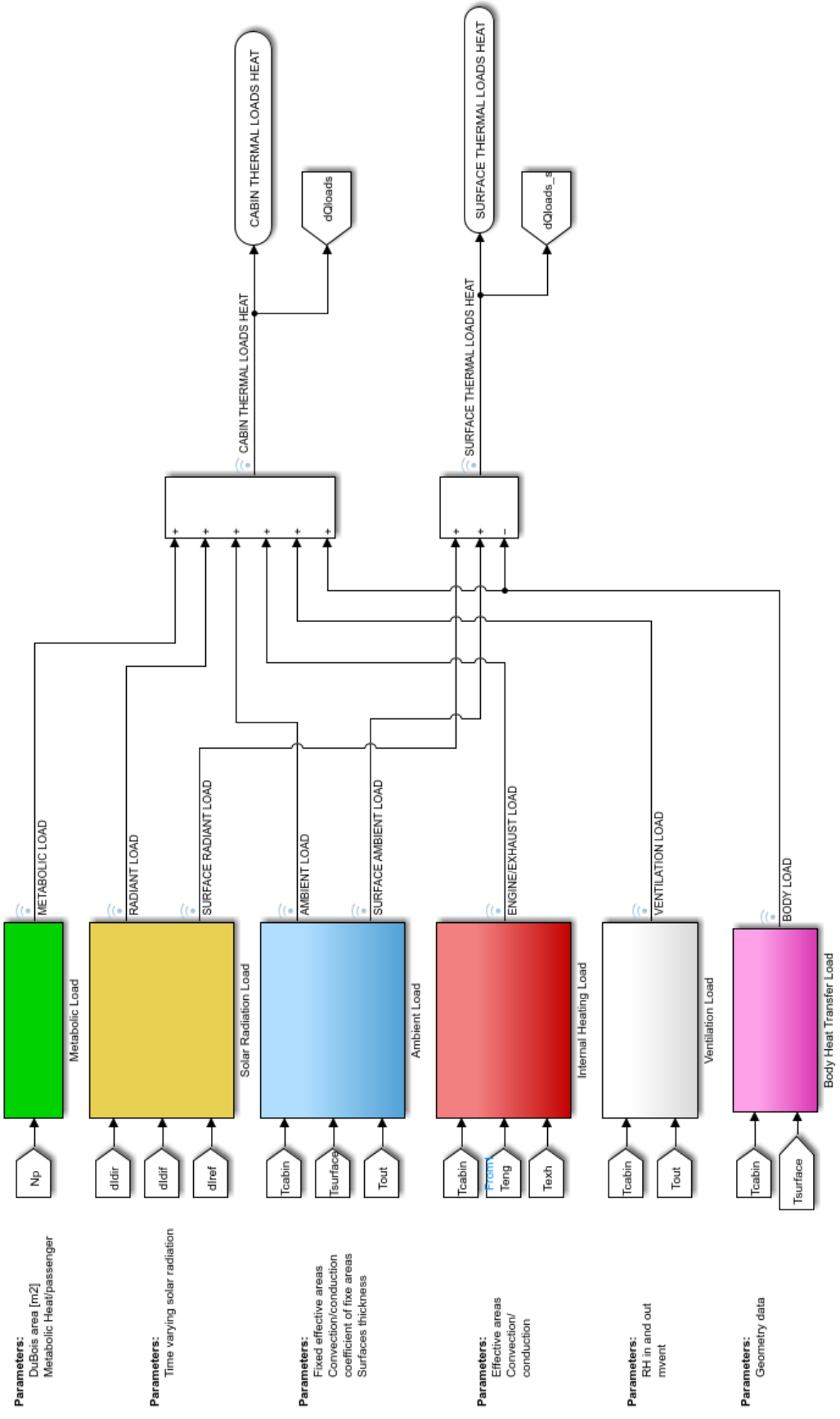


Figure 4.3: Thermal loads Simulink implementation

4.2.1 Solar radiation load

According to ASHRAE, solar radiation heat load can be split into direct, diffuse and reflected radiation loads.

$$\dot{Q}_{rad} = \dot{Q}_{rad,dir} + \dot{Q}_{rad,dif} + \dot{Q}_{rad,ref} \tag{4.5}$$

Any radiant load is expressed as the incident radiation per each surface element. Thus, it is considered the incident radiation angle, the surface element area and its emissivity properties and then, it is summed up each surface element contribution.

$$\dot{Q} = \sum_{surf} S\tau\dot{I}\cos\theta \tag{4.6}$$

Radiance expressions are extracted from [15]:

$$\dot{Q}_{rad,dir} = \sum_{surf} S\tau\dot{I}_{dir}\cos\theta \text{ where } \dot{I}_{dir} = \frac{A}{\exp(\frac{B}{\sin\beta})} \tag{4.7}$$

$$\dot{Q}_{rad,dif} = \sum_{surf} S\tau\dot{I}_{dif} \text{ where } \dot{I}_{dif} = C\dot{I}_{dir}\frac{1+\cos\Sigma}{2} \tag{4.8}$$

$$\dot{Q}_{rad,ref} = \sum_{surf} S\tau\dot{I}_{ref} \text{ where } \dot{I}_{ref} = (\dot{I}_{dir} + \dot{I}_{dif})\rho_g\frac{1-\cos\Sigma}{2} \tag{4.9}$$

Where:

A, B and C are tabulated constants in [15] for each month

β is the altitude angle calculated according to time zone and location

Σ is the surface tilt angle

ρ_g is the ground reflectivity coefficient

τ is the surface transmissivity

α is the surface absoptivity

S is the surface element area

Similarly, it can be obtained the surface heat thermal load as the incident radiation that is absorbed and neither transmitted nor reflected:

$$\dot{Q}_{body_{rad}} = \dot{Q}_{body_{rad,dir}} + \dot{Q}_{bodyrad,dif} + \dot{Q}_{bodyrad,ref} \tag{4.10}$$

$$\dot{Q}_{body_{rad,dir}} = \sum_{surf} S\alpha\dot{I}_{dir}\cos\theta \tag{4.11}$$

$$\dot{Q}_{body_{rad,dif}} = \sum_{surf} S\alpha\dot{I}_{dif} \tag{4.12}$$

$$\dot{Q}_{body_{rad,ref}} = \sum_{surf} S\alpha\dot{I}_{ref} \tag{4.13}$$

Since HVAC system is expected to operate large time periods, it is worth it to model the radiant loads time varying according to the simulation time. In this case, it is assumed constant the

location and the seasonal constants as well as the cabin disposition with respect the Sun. The load variation is achieved by the position of the Sun at each simulation time, thus, a time varying load is plugged into the model to simulate more realistic conditions. Further developments and explanations about the time expressions are discussed in Appendix A. Equation of time.

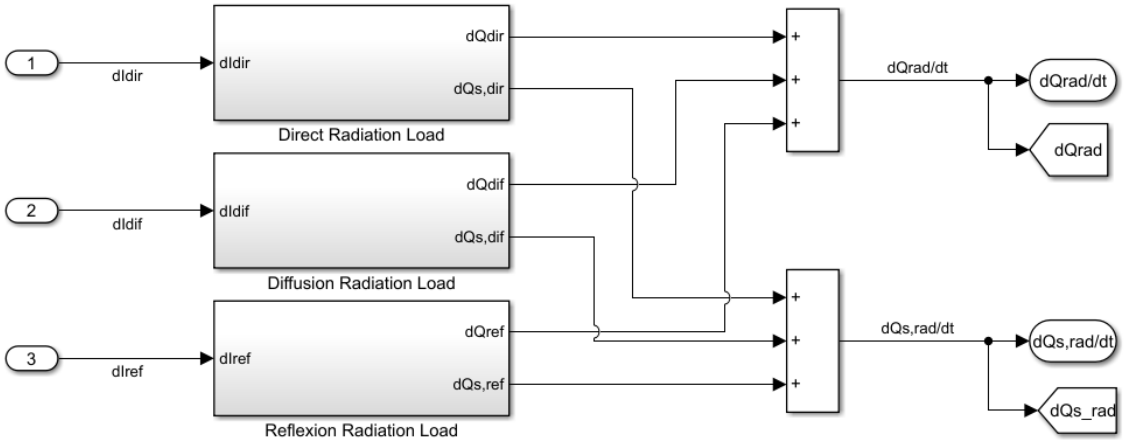
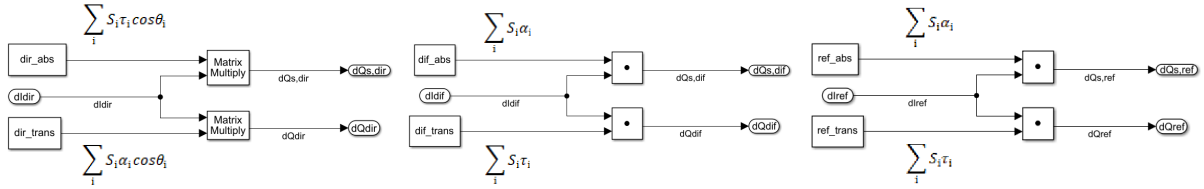


Figure 4.4: Simulink solar radiation thermal loads



(a) Direct (b) Diffusion (c) Reflected

Figure 4.5: Simulink (a) direct, (b) diffusion, and (c) reflected solar radiation loads

4.2.2 Metabolic load

Due to constant human metabolic activity, heat is generated within the human body and transferred to the cabin air. Metabolic heat rates are already tabulated in literature according to some predefined activities so that it can be determined the driving and resting metabolic rates for driver and other passengers, respectively. More accurate models might require a time varying metabolic rate according to driving conditions, i.e. heavy traffic accelerates metabolism and the metabolic rate differs to that for light driving conditions. Each passenger contributes individually to the overall metabolic thermal load.

$$\begin{aligned} \dot{Q}_{met} &= \sum_{passengers} M A_{Du} \\ A_{Du} &= 0.202 W^{0.425} H^{0.725} \end{aligned} \tag{4.14}$$

Where A_{Du} is the DuBois area, an estimator of the body surface area as a function of height, H, and weight, W.

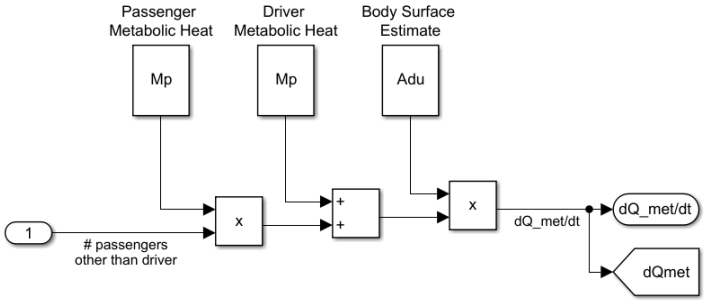


Figure 4.6: Simulink metabolic thermal load

4.2.3 Ambient load

Since there is a temperature difference between inside cabin air and outside air, heat is expected to be transferred from the hotter thermal mass to the colder thermal mass. In this process, heat is transferred by exterior convection, surfaces conduction, thus, contributing to the surface thermal loads, and interior convection. The cabin ambient thermal load, then, is produced due to the thermal interface between the surface elements and the cabin air:

$$\dot{Q}_{amb} = \sum_{surf} SU(T_{body} - T_{cabin}) \tag{4.15}$$

In eq. 4.15, U is the overall heat transfer coefficient. Within this coefficient, it is embedded the heating process.

$$U = \frac{1}{R} \text{ where } R = \frac{1}{h_{out}} + \frac{\lambda}{k} + \frac{1}{h_{in}}$$

R is the equivalent thermal resistance consisting of the interior convection, surface conduction, and exterior convection thermal resistances. It is assumed that a natural interior convection, thus, tabulated values are used. In the case of the exterior convection, it is assumed a forced convection dependent on the vehicle velocity as: $h_{out} = 0.6 + 6.64\sqrt{V_{veh}}$ [12]. The thermal conduction resistance is modeled as the ratio between the surface thickness, λ, and the surface thermal conductivity, k. As commented, a portion of the heat transfer due to ambient load across the body surface is absorbed. That is, the difference between the outside to surface and surface to cabin heat transfer:

$$\begin{aligned} \dot{Q}_{body_{amb}} &= \sum_{surf} SU(T_{out} - T_{body}) - \sum_{surf} SU(T_{body} - T_{cabin}) \\ \dot{Q}_{body_{amb}} &= \sum_{surf} SU(T_{out} - 2T_{body} + T_{cabin}) \end{aligned} \tag{4.16}$$

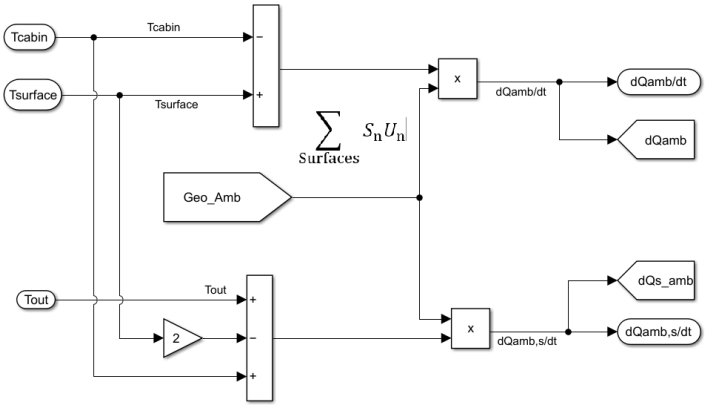


Figure 4.7: Simulink ambient thermal load

4.2.4 Internal high temperature elements load

This model is based on a combustion-type car. Thus, it is expected that the engine temperature and the exhaust gases temperatures heat the cabin air. As mentioned, it is only considered that this heat is transferred directly to the cabin air and no heat gain is produced in the surface elements. The heat transfer is derived similarly to eq. 4.15 with the difference that in this case, the hot temperature element is either the engine or the exhaust tubes and the transfer is done just by surface conduction and interior convection.

$$\begin{aligned} \dot{Q}_{eng} &= S_{eng} U (T_{eng} - T_{cabin}) \\ T_{eng} &= -2 \cdot 10^{-8} RPM^2 + 0.0355 RPM + 77.5 \end{aligned} \tag{4.17}$$

$$\begin{aligned} \dot{Q}_{exh} &= S_{exh} U (T_{exh} - T_{cabin}) \\ T_{exh} &= 0.138 RPM_{eng} - 17 \end{aligned} \tag{4.18}$$

The engine and exhaust temperatures can be estimated by means of the engine speed [12]. Thus, knowing the gear ratios and the vehicle speed, it can be determined the engine speed. Note that it is implicitly assumed a single surface element given that these heat sources are very located and just interface a single surface to reach the cabin air.

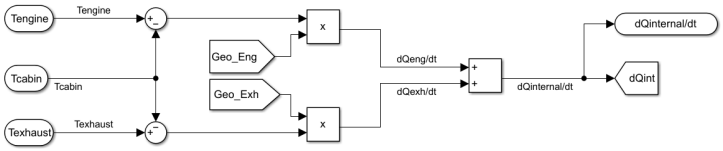


Figure 4.8: Simulink engine/exhaust thermal load

4.2.5 Ventilation load

It is assumed that the same air mass flow rate, $\dot{m}_{vent}$ that enters the cabin due to leakage exits, as well. This air has the outside air properties in terms of temperature and relative humidity. Ventilation heat gain consists of both sensible and latent loads, according to psychrometric

calculations. Both, then, are estimated by the inside and outside air enthalpies, e.

$$\begin{aligned} \dot{Q}_{vent} &= \dot{m}_{vent}(e_{out} - e_{cabin}) \\ e &= 1006T + (2.501 \cdot 10^6 + 1770T)X \\ X &= 0.62198\frac{\phi P_s}{100P - \phi P_s} \end{aligned} \tag{4.19}$$

Where

ϕ is the relative humidity

P is the air pressure

P_s is the water saturation pressure at temperature T

In the developed model, this expression is simplified by assuming the relative humidities, pressures and water saturation pressures remain constant along the simulation. The leakage air mass flow rate, $\dot{m}_{vent}$, is assumed to be $\rho_{air} \cdot 0.02m^3/s$, as suggested in [12] and its references.

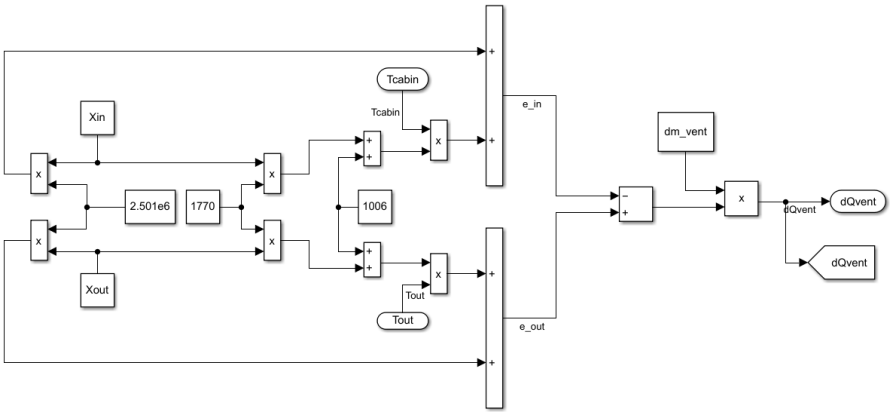


Figure 4.9: Simulink ventilation thermal load

4.2.6 Body elements heat transfer load

The interaction between the two modeled thermal masses can be accounted as the others. In this case, each surface element is accounted and it is assumed interior natural convection as heat transfer mechanism:

$$\dot{Q}_{body,cabin} = \sum_{surf} Sh_{in}(T_{body} - T_{cabin}) \tag{4.20}$$

$$\dot{Q}_{body,body} = \sum_{surf} Sh_{in}(T_{cabin} - T_{body}) \tag{4.21}$$

As already commented, this heat transfer is present in both thermal processes but with contrary sign.

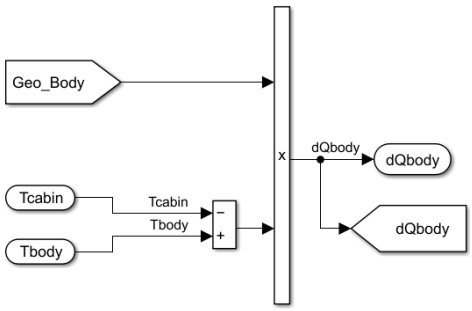


Figure 4.10: Simulink body to cabin heat transfer thermal load

4.2.7 HVAC load

The HVAC consists of the heat contribution of both modes, the heater and cooling system. Thus, it can be expressed as:

$$\dot{Q}_{HVAC} = \dot{Q}_{heating} + \dot{Q}_{cooling} \tag{4.22}$$

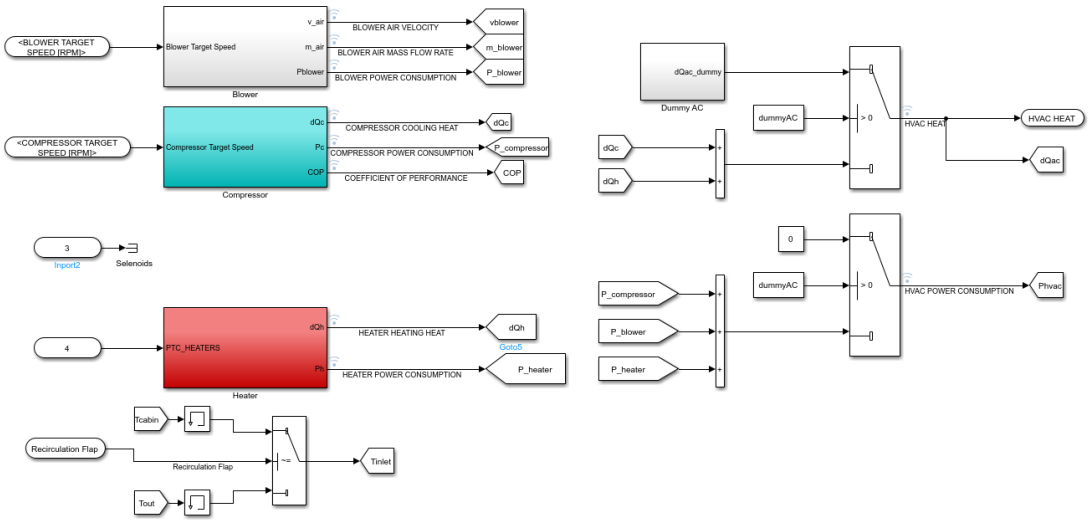


Figure 4.11: Simulink HVAC model

As described in Chapter 3, when heating system is switched on, the PTCs are powered on and the blower drives the forced convection heat from PTCs surroundings to cabin. When the PTCs are on, there is power consumption and they are self-heated up to a maximum stable temperature. Some time constant is also added in this process to ease the simulation, but the dynamics of this process are so fast that can be completely neglected in comparison with any other system element dynamic. It is used the resistance vs temperature curves experimented in [18] and the heater temperature is estimated as the inlet air temperature plus the self-heating, SHI, temperature at the actual resistance of the PTC. This actuator is power supplied by the vehicle battery, thus, it can be estimated its power consumption as:

$$P_{PTC} = \frac{V_{bat}^2}{R_{PTC}}$$

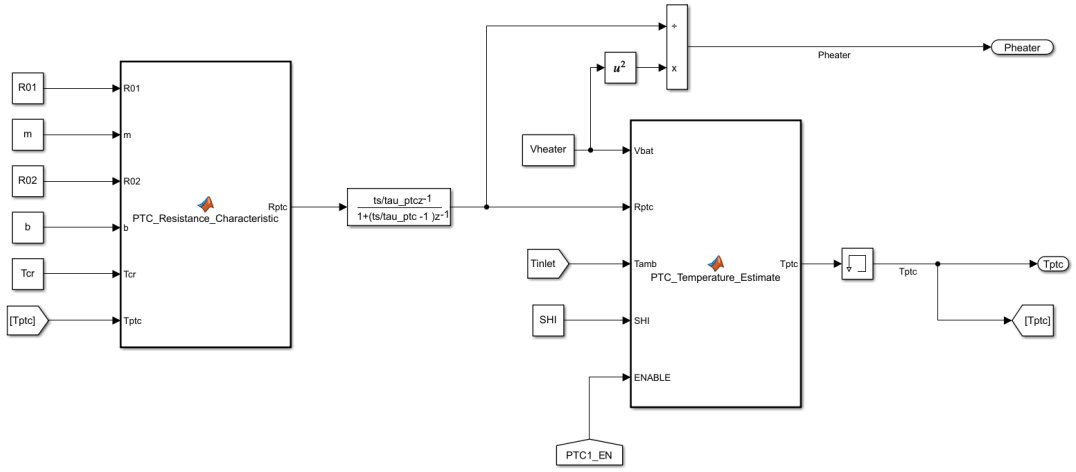


Figure 4.12: Simulink PTC model

The heating supplied to the cabin, thus, is the forced convection produced at the PTC and inlet air temperature difference governed by the blower supplied air velocity.

$$\dot{Q}_{heater} = S_{PTC} h(V_{air,blower})(T_{PTC} - T_{inlet}) \tag{4.23}$$

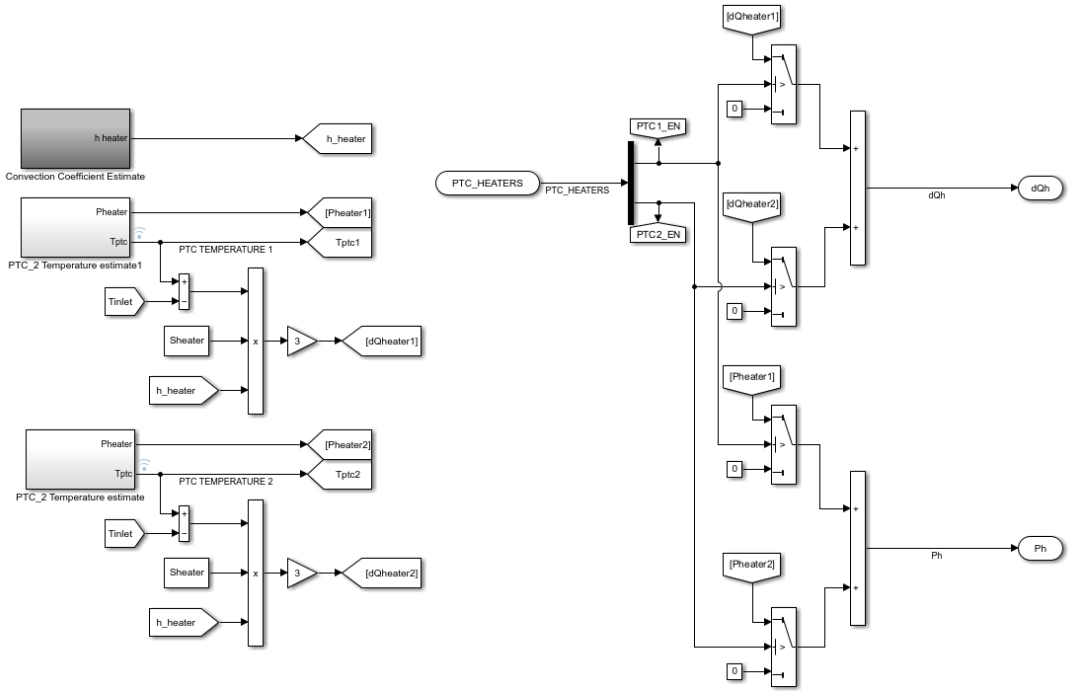


Figure 4.13: Simulink heater model

The convection coefficient is air velocity dependent.

Regarding the cooling system, the heat transfer is the same, by forced convection. However, in this case a compression cycle using refrigerant is required. The model of the compression

cycle can be heavy and much more complex than required as pointed in Chapter 3. It is only considered the heat capacity of the cycle by means of the compressor action. Thus, the work provided by this actuator will regulate the cycle so that a cold reservoir is generated at the evaporator side able to sink inlet air heat to cool down the air. By assuming an ideal compression cycle, the refrigerant temperature and pressure is given at any cycle stage. The heat capacity of the compression cycle can be expressed as:

$$\dot{Q}_{compression} = \dot{m}_{refrigerant} C_{p_{refrigerant}} \Delta T_{compressor} \tag{4.24}$$

Where $\dot{m}_{refrigerant}$ is the refrigerant mass per cycle and it is function of the compressor speed, $C_{p_{refrigerant}}$ is the heat capacity of the used refrigerant, and $\Delta T_{compressor}$ is the refrigerant temperature increase due to the compression itself. The inlet refrigerant temperature is the one at the boiling point plus some extra super-heat for safety reasons. The outlet refrigerant temperature is given by the compression ratio according to the compressor specifications. The actual refrigerant volume, considering that it is a two-phase process, can be expressed as:

$$V_{refrigerant} = V_{swept} + V_{compressor}(1 - (\frac{P_2}{P_1})^{\frac{1}{n}})$$

Where

V_{swept} is the refrigerant swept volume by the compressor action

$V_{compressor}$ is the compressor volume

$\frac{P_2}{P_1}$ is the compression ratio

n is a process constant

The refrigerant mass flow per cycle can be deduced, then, according to ideal gas law.

$$\dot{m}_{refrigerant} = \frac{P_1 V_{refrigerant}}{T_{refrigerant,in} R_{specific}} \omega_{compressor} \tag{4.25}$$

The considered refrigerant temperature, $T_{refrigerant,in}$, and refrigerant pressure, P_1, refers to the completely vapor-saturated refrigerant temperature and pressure after the evaporator (or before the compressor). The compressor speed, $\omega_{compressor}$, units are cycles per second. Note that the heat capacity of the system is assumed linear with respect to the compressor speed. The compressor shall be regulated so that the compression cycle is able to sink the demanded heat, thus, it is controlled only to assure that the refrigerant mass flow rate is able to sink enough heat. In addition, the real cooling heat provided by the cooling system is managed by the air velocity generated by the blower. By forced convection, heat sink is eased. In this system there is a saturation condition: the blower speed can be as high as desired, but if the compressor does not generate enough cooling heat capacity, the cooling performance is limited to the maximum available heat.

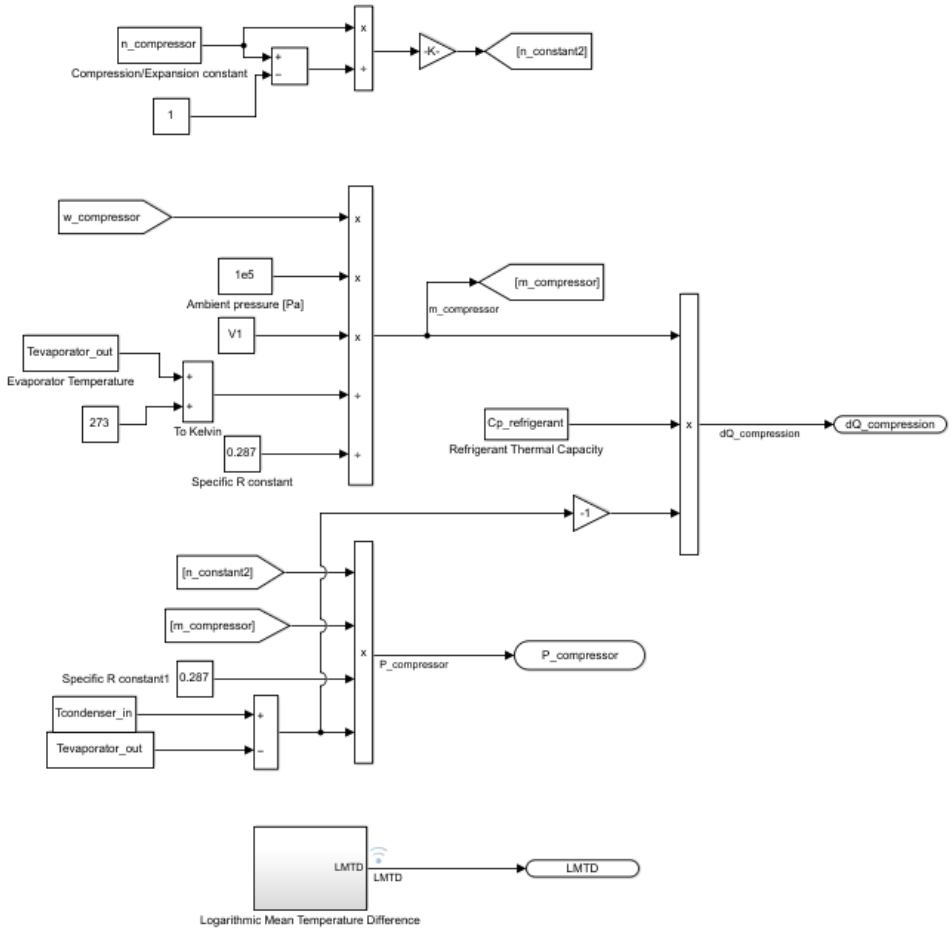


Figure 4.14: Simulink compression cycle model

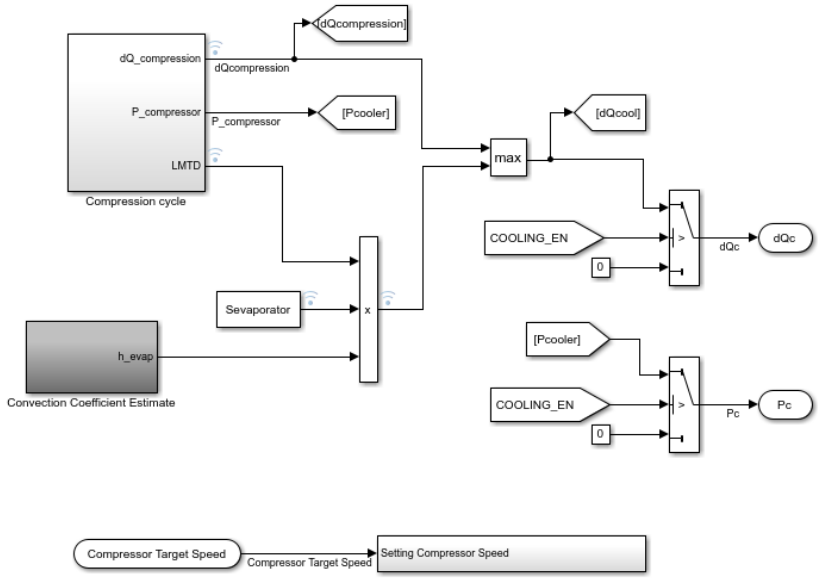


Figure 4.15: Simulink cooling system model

Since the evaporator experiments a two-phase process, it is not trivial to determine a representative cold element temperature in this system. However, to simplify and according to the thermodynamics involved have exponential behavior, it is taken the Logarithmic Mean Temperature Difference (LMTD). It is obtained an estimate of the temperature difference in the cooling process by considering a logarithmic average of the inlet and outlet air, and the inlet and outlet refrigerant temperatures.

It is also possible to determine the power consumed by the compressor as a function of the refrigerant mass flow per cycle as:

$$P_{cooling} = \frac{n}{n-1}\dot{m}_{refrigerant}R_{specific}\Delta T_{compressor} \tag{4.26}$$

Finally, the actual cooling heat supplied to the cabin follows the expression:

$$\dot{Q}_{cooling} = S_{evaporator}h(V_{air,blower})LMTD \tag{4.27}$$

4.3 HVAC dynamics

The blower is the HVAC actuator present in both systems. It is simply modeled with exponential behavior according to the actuator specifications. Thus, when some reference blade speed is required, the actual blower speed will trend exponentially to that reference with certain time constant. The interesting feature of the blower is its ventilation capabilities, thus, assuming a blade angle, α_{blade}, and radius, R_{blade}, it is estimated the moved air velocity as:

$$V_{air,blower}(s) = \omega_{ref} \frac{1}{\tau_{blower} s + 1} R_{blade} \cos \alpha_{blade}$$

Where ω_{ref} is the blower speed setpoint in rad/s. From the actuator specification is also obtained the relation between applied PWM and blower speed. Thus, it can be estimated its power consumption as:

$$P_{blower} = V_{bat} < I_{blower} > PWM \tag{4.28}$$

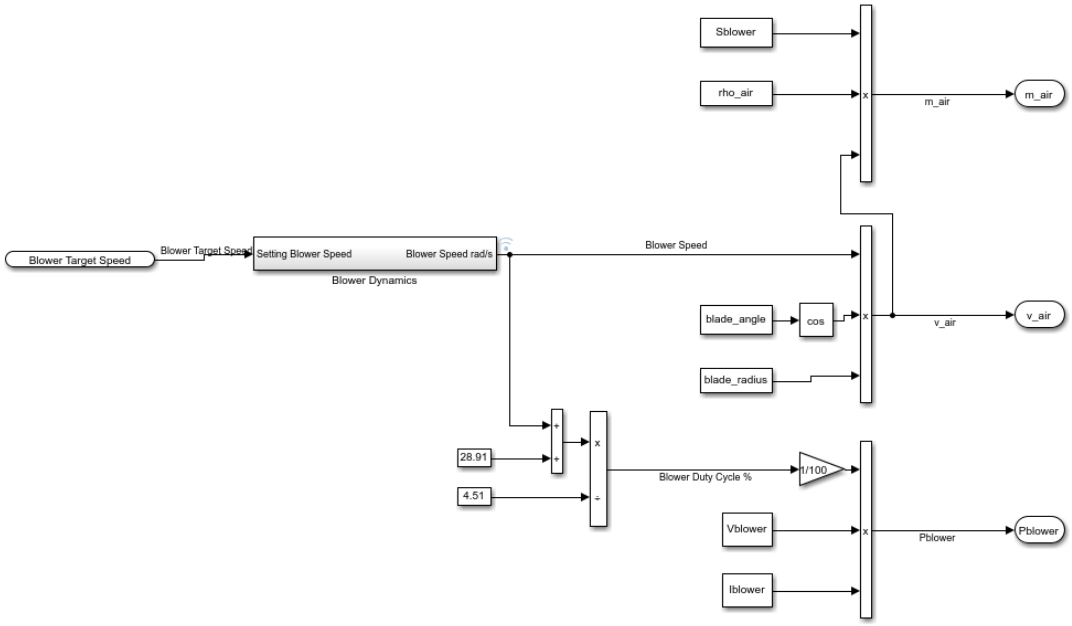


Figure 4.16: Simulink blower model

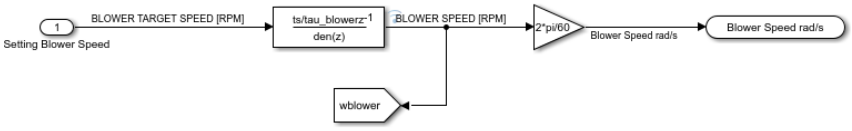
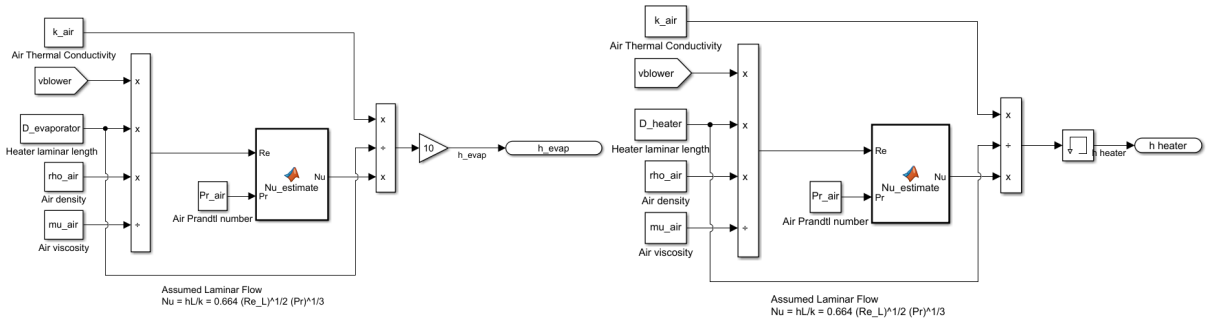


Figure 4.17: Blower dynamics model. 1st-order behavior

The blower air velocity is directly used in the PMV estimation but also, as already commented in the previous subsection, it is used to estimate the forced convection coefficient for both cooling and heating systems.



(a) Evaporator

(b) Heater

Figure 4.18: Simulink (a) evaporator, and (b) heater forced convection coefficient estimate

For the convection coefficient estimate it is assumed turbulent air flow at the heat exchanger surroundings according to actual system geometry, number of PTCs and heat exchanger effective surface.

4.3.1 Heating mode dynamics

Heating dynamics are simplified as just the blower dynamics. PTCs can be heated up so fast that the only remaining dynamics are the blower regulation itself.

4.3.2 Cooling mode dynamics

The whole compression cycle is assumed to be internally regulated by the compressor integrated controller. Thus, cycle dynamics can be assumed much faster and smooth than compression reference speed tracking. The compressor is modeled as a 1st-order subsystem and it has a time constant associated according to the actuator specifications sheet, similar to the blower.

$$\omega_{compressor}(s) = \omega_{compressor,ref}\frac{1}{\tau_{compressor}s + 1} \tag{4.29}$$

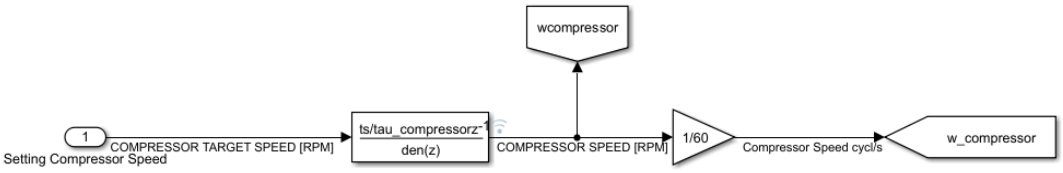


Figure 4.19: Compressor dynamics model. 1st-order behavior

4.4 Thermal comfort estimate

The implemented thermal comfort index is the PMV according to its standardized expression, eq 2.1. Thus, it is implemented accordingly. It is required to define some passenger conditions, such as, their clothing insulation and activity level. These are considered static and the time varying variables will be the air velocity perceived by passengers, mean radiant temperature and cabin air temperature.

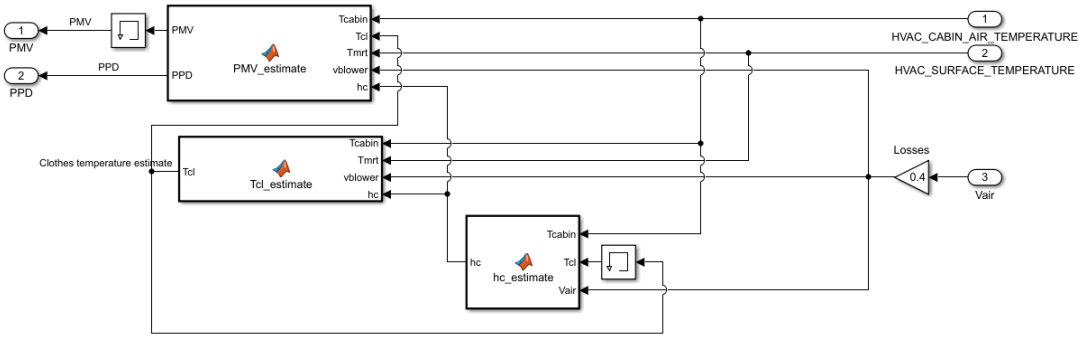
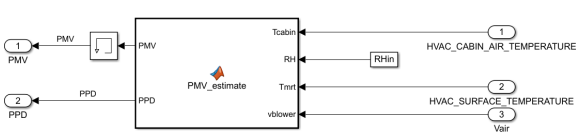


Figure 4.20: Simulink PMV Implementation

The PMV estimation also requires the clothing temperature and convection coefficient estimate. The clothing temperature is obtained recursively after N iterations. If coverage is not achieved, it is reported a warning to notify of this fact. The air provided by the blower, estimated in subsection 4.3 is scaled accounting for vents distribution and other losses.

Note that this estimate require high computational resources and it can alter the system computation performance of the BCM. Thus, to get rid of this issue, it is proposed a linearized version for the real-time simulations. As this model is addressed to be used in real-time machines and its purpose is to be used in a real-time BCM microcontroller, the estimate is simplified as depicted in Fig 4.21. The linearization is obtained using the *regress* Matlab function, from the *Statistics and Machine Learning* Matlab toolbox, where it is defined some realistic variable ranges and then, it is computed the multi-variable linear regression.



(a) Simulink implementation

```
function [PMV, PPD] = PMV_estimate(Tcabin, RH, Tmrt, vblower)

PMV     = -14.1026 + 0.5021*Tcabin +0.2035*RH +...
                              0.1407*Tmrt -2.3339*sqrt(vblower);

PPD = 100 - 95.*exp(-(0.03353.*PMV.^4+0.2179.*PMV.^2));
```

(b) Matlab code

Figure 4.21: Simulink linearized (a) PMV, and (b) PMV code

4.5 Model implementation

The overall system has been split into several building blocks that model each system element or thermal load. Besides, some blocks have been implemented to contextualize the model by means of realistic time varying elements such as driving or radiation profiles. The model, then, is ready to be implemented all together in the same Simulink file. The modular modeling allows to integrate each system element and provides a visual and qualitative representation about the thermal processes. The top level system model overview can be read as (1) the plant, (2) the controller, and (3) the feedback variable estimate. The high-level block diagram, thus, can be implemented as exposed in Chapter 3, Figure 3.

4.5.1 Simulink model implementation

Two Simulink models are implemented. The first one (referred as the original model) is the modular model built with the building blocks exposed in sections above. This allows to know each sub-block performance and provides a qualitative understanding on what is going on within the model. However, high nonlinearities appear and this is an issue for the control design and real-time simulations. The second implemented model (referred as the simplified model), then, is a linearization of the nonlinear system elements, and a rewrite of the system equations so that transfer functions can be defined. Note that this eases the control design since linear controllers can be employed as long as the linearized model behaves similar to the real one. A first approach to this simplified model is the linearization of the PMV.

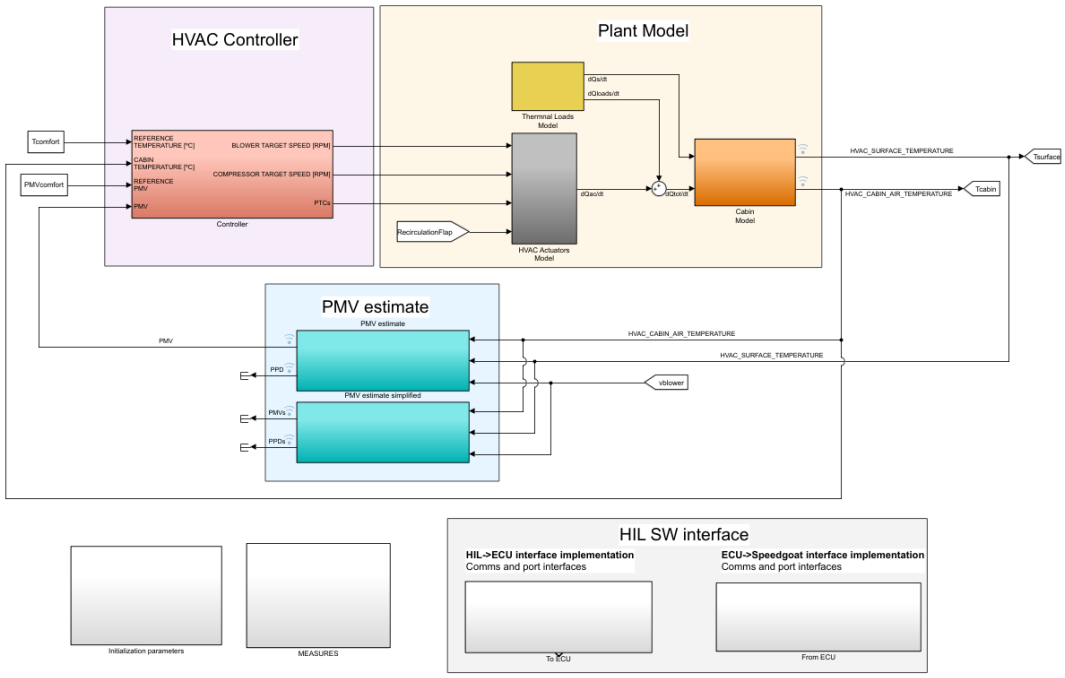


Figure 4.22: Simulink high-level model. Original system

Some extra blocks are added in the original model, as depicted in Figure 4.22. These are:

- Initialization parameters: some initial conditions are defined, such as the initial temperatures. In addition, this blocks provide some non used parameters in this model but needed for the proper BCM interface such as Controller Area Network (CAN)/Local Interconnect

Network (LIN) communication blocks or other compressor returning parameters.

- MEASURES: this block contains all signals to be monitored and plotted both in real-time and offline. A Matlab script process the gathered data during simulation based on some measures.

- HIL SW interface: the real-time machine requires some port and comms SW interface provided by this block. Thus, data is sent and received to real HW according to the contents of this blocks.

- HVAC Controller: it contains the control strategy and logic described in Chapter 5.

Regarding the simplified model, it is rewritten the thermal loads expressions as a function of the system variables; outside and inside temperatures, and metabolic and radiant loads. The HVAC actuators are linearized so as the heat expression becomes linear. This implies that the control must assure that, for instance, only available heat is transferred to the cabin. Note that Simulink already provides linearization toolboxes. The effort done here tries to ease the Simulink linearization process and also tries to provide an equivalent system model, easier to manage according to control theory, i.e. with well-defined system transfer functions. The aim of this model is just to design easier the control architecture, thus, the already defined extra blocks are not required at all.

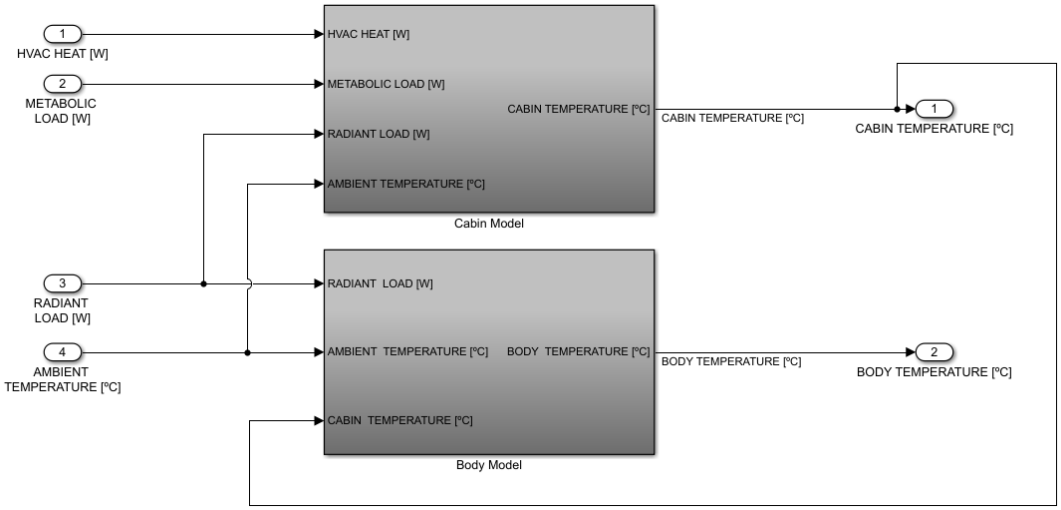
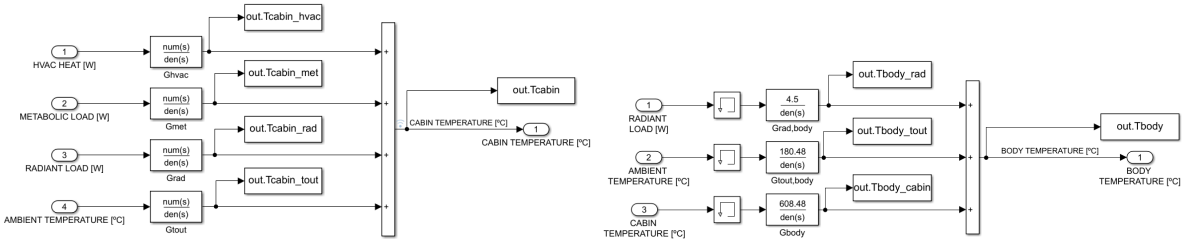


Figure 4.23: Simulink linear plant model. Simplified system



(a) Cabin (b) Body

Figure 4.24: Simulink linearized (a) cabin, and (b) body dynamics

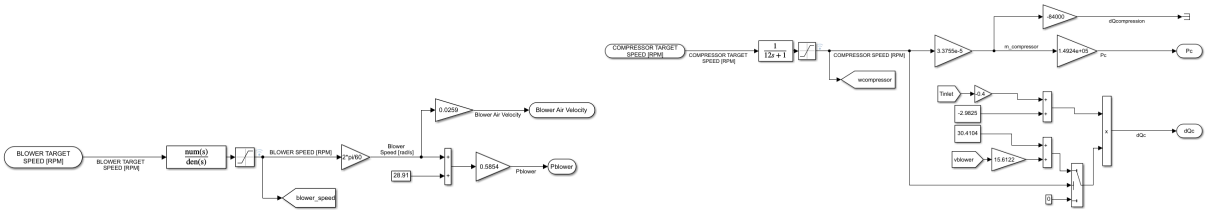
Just ambient and radiant load are assumed inputs to the simplified plant as they do not depend on system itself. Ambient and ventilation loads are rewritten together with cabin and body dynamics so that transfer functions embed all system geometric parameters of the original model. In addition, it is found a high nonlinearity in the ambient load model. There are poles and zeros that are vehicle velocity dependent with very changing dynamics. In order to avoid the model of a time-varying system, which became hard to analyze, it is fitted the curve of this load vs vehicle velocity and, before simulation starts, it is configured accordingly the ambient load parameters. This generates intrinsically a modeling error, but, comparisons will be later performed to determine model representation level for control design. It also has not been considered engine and exhaust thermal loads since they can be mitigated by proper cabin thermal insulation and the linear model would became hard to manipulate, as well.

The cabin and body dynamics, then, can be expressed as:

$$T_{cabin}(s) = G_{met}(s)\dot{Q}_{met} + G_{rad}(s)\dot{Q}_{rad}(s) + G_{HVAC}(s)\dot{Q}_{HVAC}(s) + G_{T_{out}}T_{out}(s) \tag{4.30}$$

$$T_{body}(s) = G_{rad,body}(s)\dot{Q}_{rad}(s) + G_{T_{out}}(s)T_{out}(s) + G_{body}(s)T_{cabin}(s) \tag{4.31}$$

Note that $T_{body}(s)$ is expressed in terms of the cabin temperature. During the linearization process, cabin transfer functions are obtained according to their dependence, also, on the body temperatures, thus, there are two poles in each. Also note that the HVAC heat can be furtherly expressed in terms of actual actuators set point, thus, control variables are embedded within, eq 4.22, linearly. A linearization is required for the cooling and heating expressions since the forced convection coefficient estimation and LMTD are quite nonlinear. However, a linear regression can be performed with neglectable error.



(a) Linear blower (b) Linear compressor

Figure 4.25: Simulink linearized (a) blower, and (b) compressor dynamics

Regarding the heating model, it is left as it is since it only consists of am *if/else* condition that Simulink can easily linearize.

4.5.2 Matlab script implementation

In order to validate the models implementations, it is created a Matlab script that runs the desired model. By doing so, the model is parameterized to adjust the values, as originally discussed. These scripts allow to simulate with some flexibility at the same time that it is let to next project phases to reuse the model or even fine tune it. Scripts can be found in Appendix B.

Particularly, these scripts set the simulation variables, such as the simulation duration, the initial time, the transfer functions but also uploads the geometric data and the radiation profile dataset used. Additionally, some figures are created to process the simulation offline from the measured signals in the models.

4.6 Cabin dynamics simulation results

Four different scenarios are considered to compare the system behavior as different environmental conditions might be present while driving:

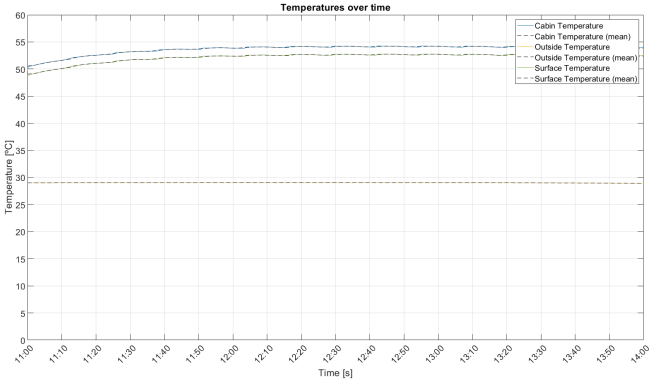
- **Scenario 1**: summer, day
- **Scenario 2**: summer, night
- **Scenario 3**: winter, day
- **Scenario 4**: winter, night

It is chosen July as the summer month and February as the winter month. The simulation starts at 11:00h, Barcelona local time, for the day simulation and at 21:00h for the night simulation.

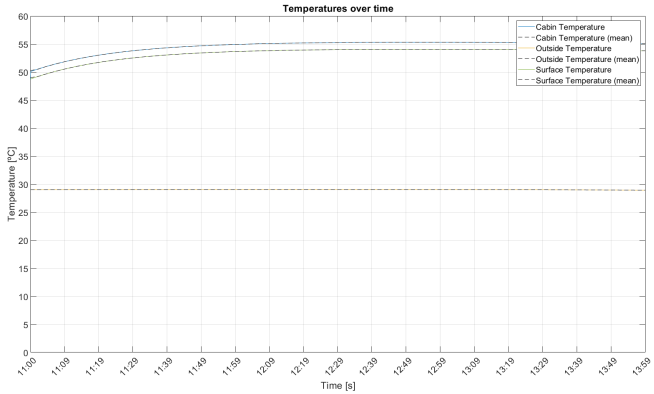
Table 4.1: Model specifications for initial validation

Spec	Original model	Simplified model
Driving profile	HWFET[1]	$V_{veh} = 50km/h$
Simulation time	3 hours	same
Location	Barcelona	same
Number of passengers	Driver + Passenger	same
Cabin RH	60 %	same
Outside RH	50 %	same
Initial Temperatures	Close to steady-state	same I.C.

With the model specifications defined in Table 4.1, the geometric data provided by the customer, and the remaining parameters taken as generic values (i.e. ground reflectivity or material properties), the initial simulations can be obtained to validate both models.

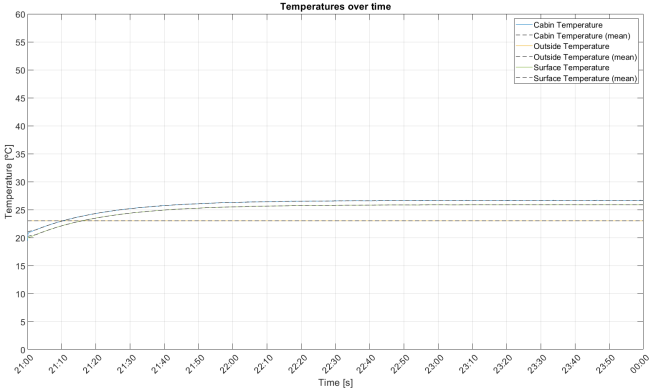


(a) Original system

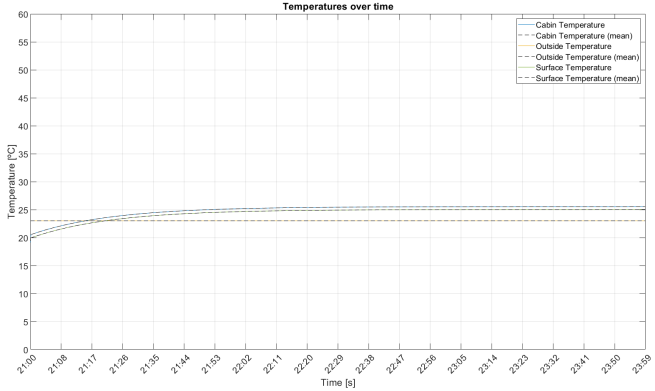


(b) Simplified system

Figure 4.26: Temperature evolution. Scenario 1

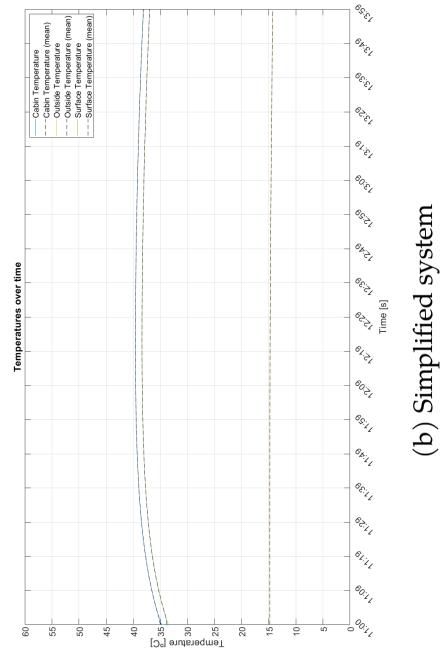


(a) Original system

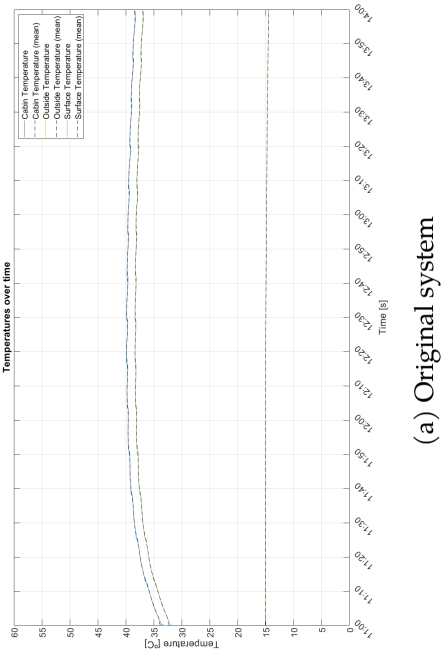


(b) Simplified system

Figure 4.27: Temperature evolution. Scenario 2

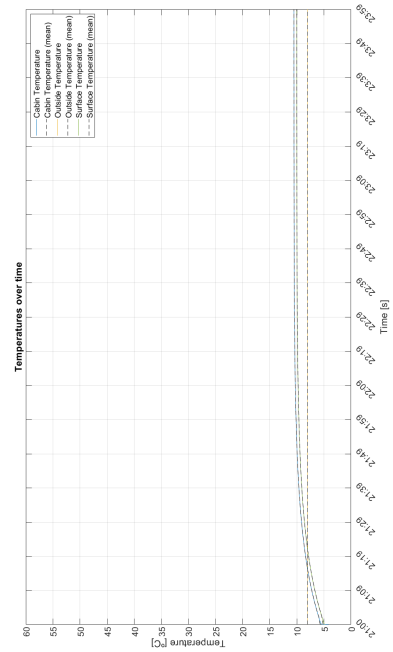


(a) Original system

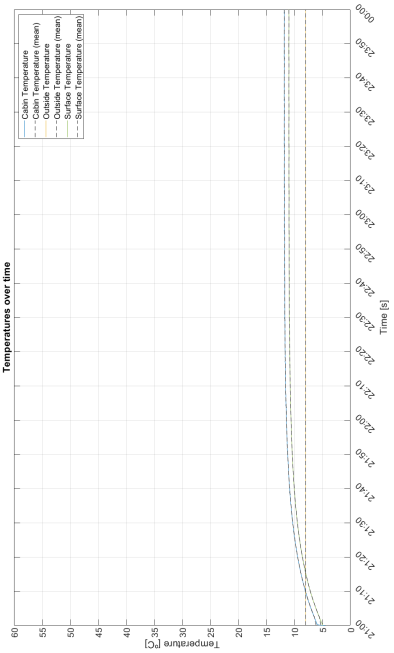


(b) Simplified system

Figure 4.28: Temperature evolution. Scenario 3



(a) Original system



(b) Simplified system

Figure 4.29: Temperature evolution. Scenario 4

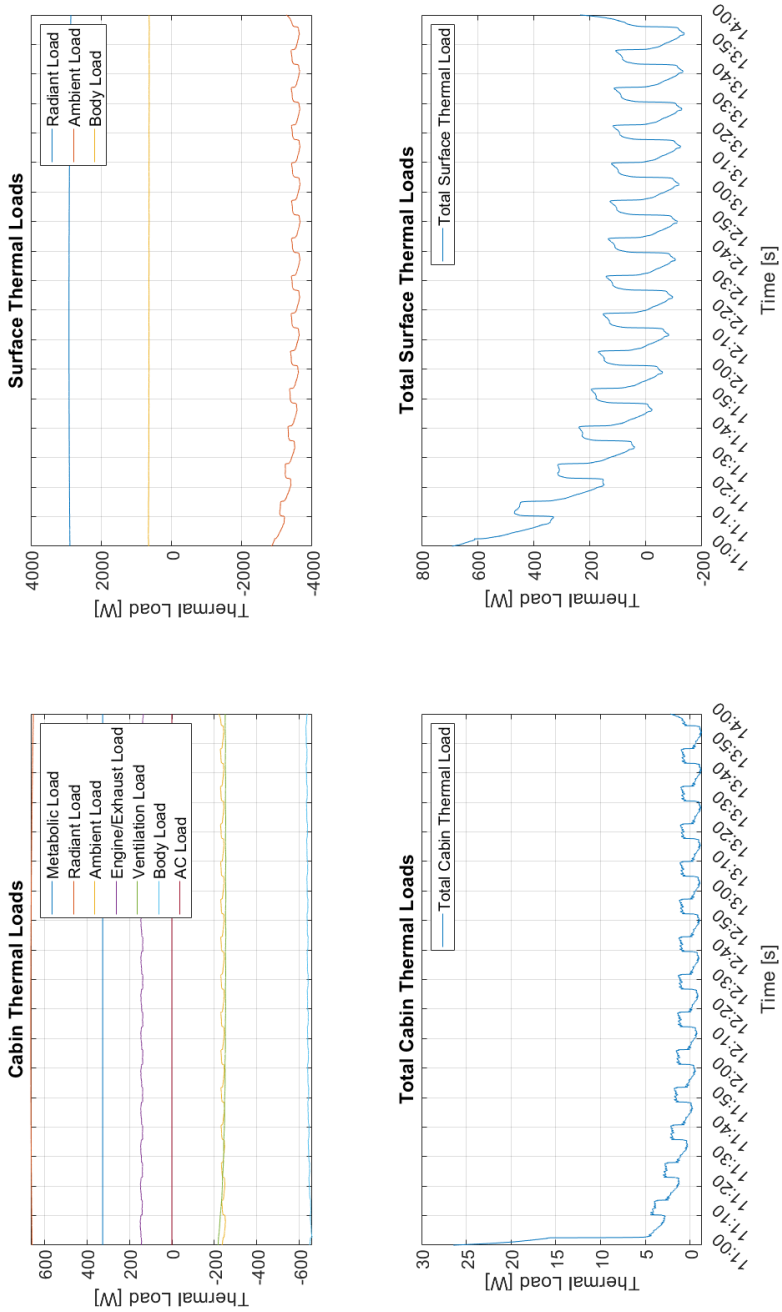


Figure 4.30: Thermal loads evolution of the original system. Scenario 1

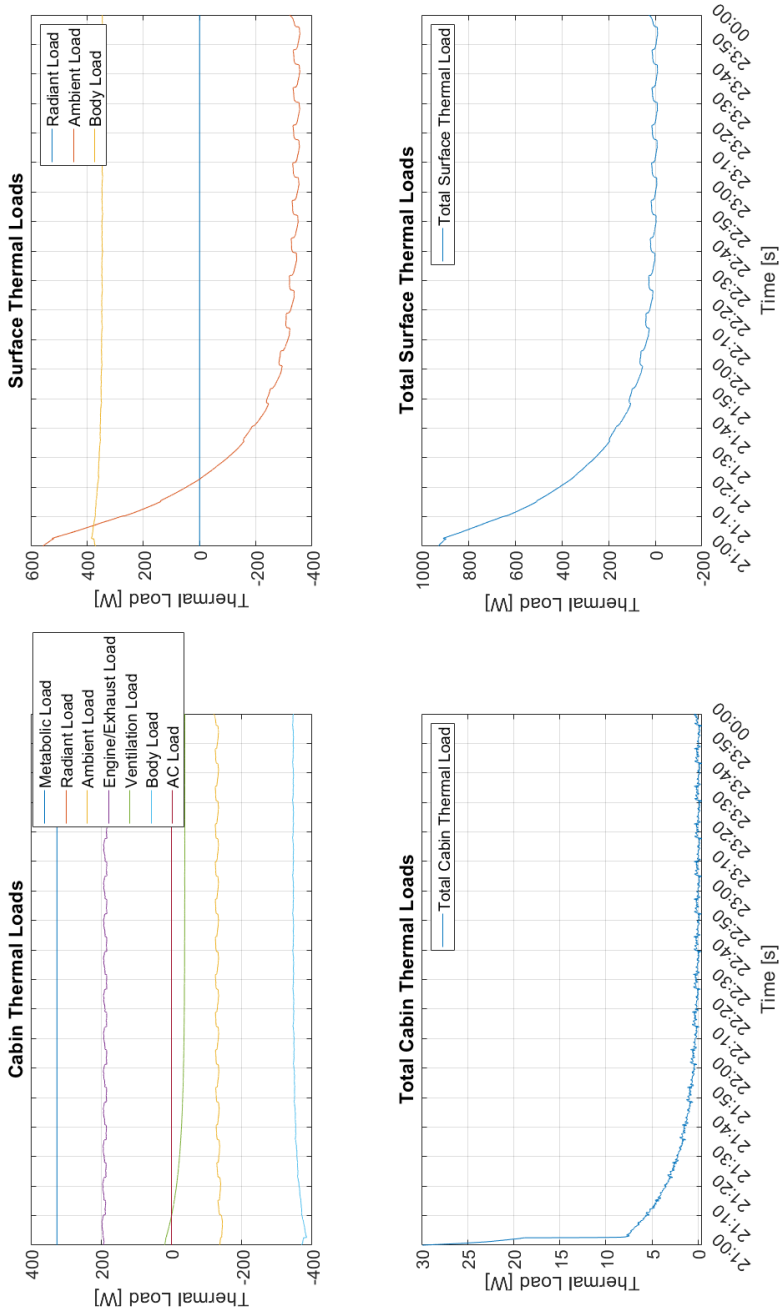


Figure 4.31: Thermal loads evolution of the original system. Scenario 4

Table 4.2: HVAC thermal load to cabin temperature poles and zeros

Pole/Zero	Location [rad/s]	Location [Hz]
ω_{z_1}	0.0033	$f_{z_1} = 0.000521$
ω_{p_1}	0.3026	$f_{p_1} = 0.0482$
ω_{p_2}	0.0007	$f_{p_2} = 0.0001$

As expected, there are small discrepancies between both proposed models due to the approximations and the linearization taken. However, it is designed the control with the simplified model and then it is tested with the original one to, again, validate it. Both models are estimating both surface and cabin temperature dynamics quite similar. On one hand, the original model is more representative. For instance, the loaded driving profile can be plugged into the model and it can be seen how the temperatures response to that. In addition, thermal loads are well defined and can be furtherly analyzed. On the other hand, the simplified model avoids the computational load by providing less system information and accuracy, i.e. engine and exhaust thermal loads are not modeled, but as representative as required for its purpose.

Regarding the temperatures evolution, Figures 4.26-4.29, under day conditions, cabin temperature rises over the outside temperature since solar radiation and metabolic activity heats up the cabin up to the steady state. In contrary, under night conditions, cabin temperature is not able to increase as much since the metabolic heat gain is partially lost though the surfaces to the outside (Ambient Load).

Regarding the thermal loads evolution, Figures 4.30-4.31, they respond as expected. In the winter night case, there is not solar radiation, passengers heats up the cabin, thus, cabin transfers heat to the body elements (Body Load) and to the exterior (Ambient Load) since the outside temperature is lower. In the summer day case, solar radiation exists and, added to the passengers metabolic heat, increases cabin temperature. This, produced greater temperature difference between cabin and outside temperature, thus, ventilation load is also greater.

Finally, from the simplified model, it can be obtained from the open-loop transfer functions the corresponding bode plots. The HVAC thermal load to cabin temperature bode is depicted in Figure 4.32.

Note that this transfer function is defined as a function of the HVAC heat, thus, it is not still a typical `control to output` transfer function. Since the actuators are linearized, they just add an extra pole at the actuator pole frequency (around $f_{p_{blower}} \approx 0.0265$ Hz) and some gain (up to ≈ 65 dB). It is not computed since there are two operating modes and the system dynamics together with the actuators is analyzed later in Chapter 5. Regarding the poles and zeros location, they depend on the vehicle velocity, assumed constant in the linear model. However, the maximum frequency shift obtained is $f_{p,z_{max}} \approx 2 f_{p,z_{min}}$.

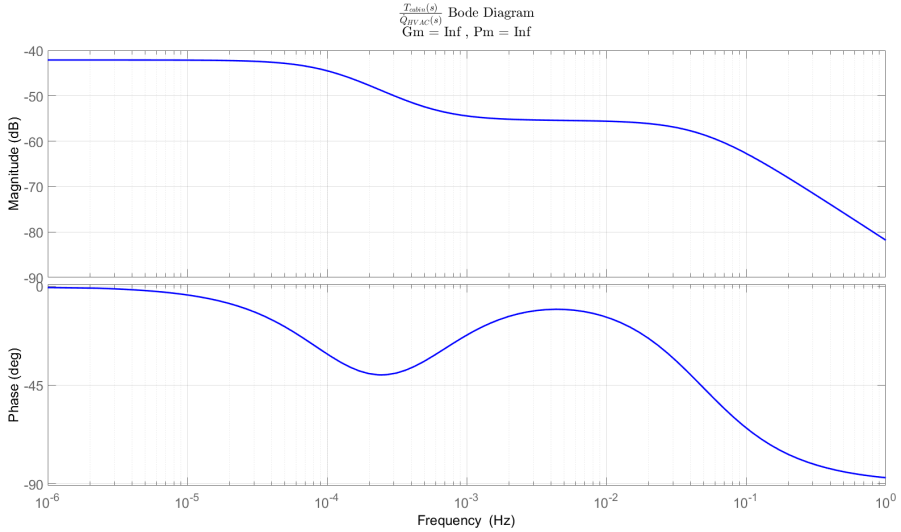


Figure 4.32: HVAC thermal load to cabin temperature transfer function bode plot ($V_{veh} = 50km/h$)

Chapter 5

HVAC control design

Control design shall take into account the system purpose. The control architecture, then, is accordingly designed under the system considerations taken in Section 3.3. HVAC operating modes. The controller is implemented so that either the user temperature set point, temperature-based, or thermal comfort, PMV-based, is reached. Thus, the actuators are controlled to fulfill this system control requirement. Other more sophisticated systems could be temperature-based with a PMV minimization error approach. This lead to a more complex control but more user-managed system since the user temperature requirement is always fulfilled. However, the control approach presented here just considers one variable to control.

Under heating conditions, PTCs are enabled and the blower speed shall be adjusted to regulate cabin temperature or thermal comfort. In this case, the compression cycle is disabled by means of setting the compressor speed to zero. The model does not consider other compression cycle variables such as the valves closing, for simplicity. In this case, the PTC dynamics are not expected to modify the closed-loop performance since they are much faster than blower and cabin dynamics.

Under cooling conditions, compressor is enabled and its speed is regulated so that the compression cycle generates enough heat capacity. Then, the blower speed is adjusted to control cabin comfort. Note that in this case, it is required both regulations: compressor and blower speed. Thus, a multi-variable controller architecture shall be implemented since the compressor and blower dynamics are comparable and a proper control strategy shall be addressed to satisfy both, system performance, and stability.

Note that the operating conditions range is quite wide. Therefore, it is prioritized the controller robustness to assure proper system performance. This can be interpreted as, depending on the environmental conditions or the driving scenario, the controller effort will vary. In addition, there exists two operating states with different purposes, one is heating and the other is cooling. They share a common actuator, the blower, so the implemented controller shall provide proper performance for both system states, as well.

5.1 Control strategy

The considered control strategy considers the system operating modes as well as the system states. Thus, heating and/or cooling are enabled as simulation initial conditions and it is let to the system to determine whether there are cooling or heating conditions. The considered system states conditions are:

- Heating conditions: the variable to control (temperature or PMV) is above the setpoint with certain hysteresis defined at compilation time.
- Cooling conditions: the variable to control is below the setpoint plus the hysteresis.

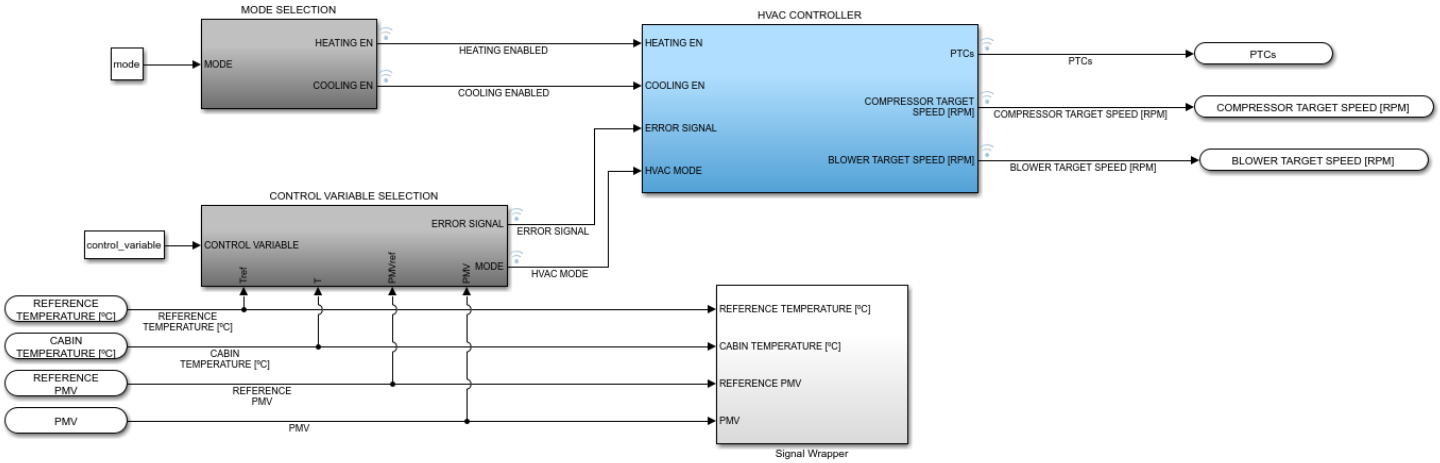
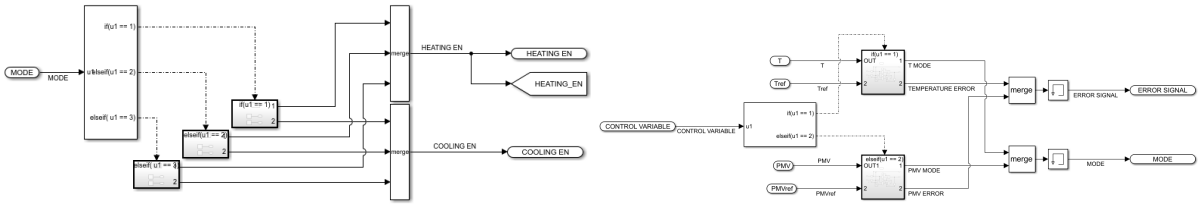


Figure 5.1: Simulink controller model

The hysteresis is implemented to avoid constant system state changes in steady-state conditions since the system dynamics are slow and there is a direct impact in control performance if the systems states are constantly oscillating from cooling to heating. In Figure 5.1 it is depicted the controller scheme. The *MODE SELECTION* block is in charge of determining if heating and/or cooling states are enabled. Thus, when they are not, the corresponding subsystem shall be disabled. The *CONTROL VARIABLE SELECTION* block has two different objectives. It determines which variable is addressed to be controlled, either the temperature or the PMV. In addition it generates the error signal taking into account the defined hysteresis. Thus, it determines all time which is the error signal at the same time it defines the system states, either cooling or heating. The *HVAC CONTROLLER* block is in charge of setting the proper actuators driving depending on the operating modes. Finally, the *Signal Wrapper* embeds interesting signals to be plotted to monitor system dynamics and controller performance.



(a) MODE SELECTION (b) CONTROL VARIABLE SELECTION

Figure 5.2: Simulink controller blocks

Since the system has been simplified and linearized, the system controller topology is chosen to be linear, as well. Different architectures can be implemented according to open-loop bode plot. However, this system is very slow and the controller BW will be mostly greater than system dynamics. Thus, simple architectures can be used. Particularly, from Figure 4.32, it can be seen that the system is *Type-I*, since there is not any pole at the origin (at very low frequencies there is constant magnitude). Thus, controller shall contain an integrator path to remove steady-state error when a step is applied. A proportional action is also required to speed up the control performance. However, it must be carefully chosen since the system dynamics are very slow and it is interesting to avoid actuators saturation to extend their lifespan. Finally, these kind of processes often require overshoot minimization since oscillations around the setpoint can, eventually, lead easily to thermal discomfort. This phenomena is not modeled with the PMV thermal comfort estimate, but, it shall be considered stable and small variations comfort states.

The linear controller proposed topology is the so called Proportional, Integral and Derivative controller (PID) since this topology offers, potentially, enough flexibility to fulfill the control requirements. In addition, the selection of this topology also offers flexibility in the final prototype tuning together with real-time simulation tools, such as HIL machines. The three independent actions of the controller (proportional scaling, integration and derivative action) can bee independently tuned in real-time with these simulation tools as they manage different control performances.

$$\begin{aligned} y(t) &= g_{PID}(e(t)) = Pe(t) + I\int_{-\infty}^{t} e(\tau)d\tau + D\frac{d}{dt}e(t) \\ Y(s) &= G_{PID}(E(s)) = (P + I\frac{1}{s} + Ds)E(s) \end{aligned} \tag{5.1}$$

In Figure 5.3 is depicted the controller strategy and scheme. According with the actual operating conditions, PTCs are enabled/disabled and blower and compressor speed are set. The conditions of enabling/disabling the heating and/or cooling systems are provided by the *MODE SELECTION* and *CONTROL VARIABLE SELECTION* blocks. Thus, two different control schemes are implemented accordingly. In the case of the cooling state operation, the compressor speed is controlled in an inner loop to assure that the cooling system has always available heat capacity. Therefore, the compression cycle is rapidly regulated according to the blower request. The outer loop sees the system heat requirement and assumes that there is always enough heat capacity, thus, it is adjusted to extract the desired heat from the cabin accordingly. Since the expression for the cooling capacity and the cooling heat have already been referred to their control variables, compressor and blower speed respectively, it is found the relationship

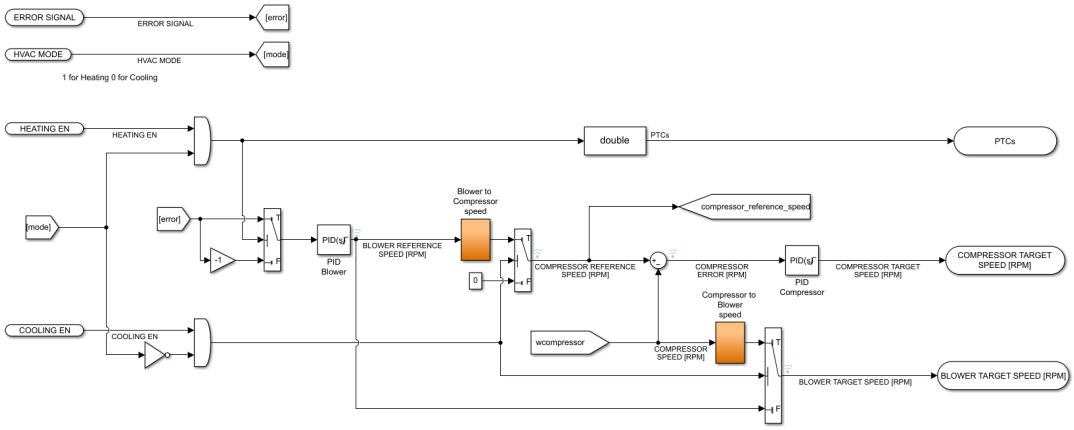


Figure 5.3: Simulink controller scheme

between them. Therefore, it can be written the required compressor speed according to the cooling requirement by means of scaling the required blower speed. By doing so, the blower is regulated to fulfill the cooling requirement and, the requested blower speed can be scaled to be translated into compressor speed. This, then, would be the reference setpoint for the compressor inner loop. Note that the inner loop dynamics shall be faster to be able to design both loops independently assuming that the inner loop is static regarding the outer loop. However, this is not fulfilled in this system, thus, analytical methods to design the controllers became complex. In addition, to partially mitigate this issue and to establish some safety margins, the compressor speed translation includes a *Safety Factor* that prevents the cooling system to work in the limit conditions. By doing so, the compressor dynamics, that are quite slower than blower ones, will have less impact in the overall performance.

5.1.1 Control specifications

The proposed model is partially linked to actual realistic system data. However, it cannot be said that it represents a given HVAC system completely. Thus, it is pointless to define several specific control specifications, such as setting time. Instead, general objectives are defined to orientate the control design and to validate accordingly the control performance. The system, then, shall be:

- capable of regulating any realistic condition
- stable and robust
- accurate (i.e. small step response steady-state error)
- able to self control the system states
- dumped (i.e. small step response overshoot)
- balanced

  power consumption vs settling time

  power consumption vs perturbation rejection

  settling time vs overshoot

In order to analyze the designed control performance, same scenarios are simulated to compare different solutions. These scenarios have already been described in Section 4.6. Although it has been simplified the original system model and the temperature simulation results show that both models are quite equivalent, the proposed controller will be later validated also with the original system to observe how perturbations and driving conditions are managed by the designed controller.

5.1.2 Control design methodology

The linearized model is used for the control design since its implementation has less computational load and Simulink tools can be used for the control design. In spite of having linearized the thermal processes of the system and simplified the actuators response, nonlinear elements still remain in the overall model. For instance, HVAC heat delivered to or sank from the cabin expressions contains time varying variable products. However, further simplifications could have been developed and a small signal analysis would have provide a completely representative and equivalent linear model. Instead, it is assumed that the dynamics of these products are very different and the models are left as they are, letting Simulink to linearize them. The control design, then, can be done by means of Simulink *Control System Designer* APP. This tool linearizes the system and allows to design the controller by means of different techniques (root-locus, Bode, Nyquist, etc).

It is used the Bode-based tuning method to fine tune both controllers, the compressor and the blower ones. The controllers to be tuned has a PID topology. The simplified model is run in continuous time, thus, high frequency components present in the real system, such as sensors noise, can affect system performance since the derivative action can be understood as a High-Pass Filter (HPF), weighting excessively these high-frequency components and, thus, control performance worsen. It is added then, a Low-Pass Filter (LPF) in the derivative path to limit the operating frequency range. In addition, this system is likely to saturate the actuators in some extreme conditions, when system setpoint is far away of actual cabin state. This has been taken into account in the integration path of the controller since if the actuators are saturated, the remaining error signal will be integrated constantly. Therefore, an anti-windup method [19] must be used to disable the integrator action when these conditions are met.

It can be obtained the closed-loop transfer function of the complete system with the control architecture shown in Figure 5.3.

The compressor cycle crossover frequency, $\omega_{c_{compressor}}$, is 0.00141 Hz and the closed-loop crossover frequency, $\omega_{c_{cl}}$, is not defined since the closed-loop gain is below 0 dB in all the frequency spectrum. The followed procedure to design the controllers has been as follows:

1. It is designed the cooling controller first. The compressor cycle is firstly designed increasing its crossover frequency to speed up heat sink dynamics. Then, the blower controller is designed together with the closed-loop scheme.
2. The overall system performance validation is done by means of injecting steps to the reference variable to control.
3. When the qualitative specifications defined in Section 5.1.1 are met, the designed controller is ready to be tested in other conditions
4. Controller is tuned in case there is any Scenario in which the specifications are not met

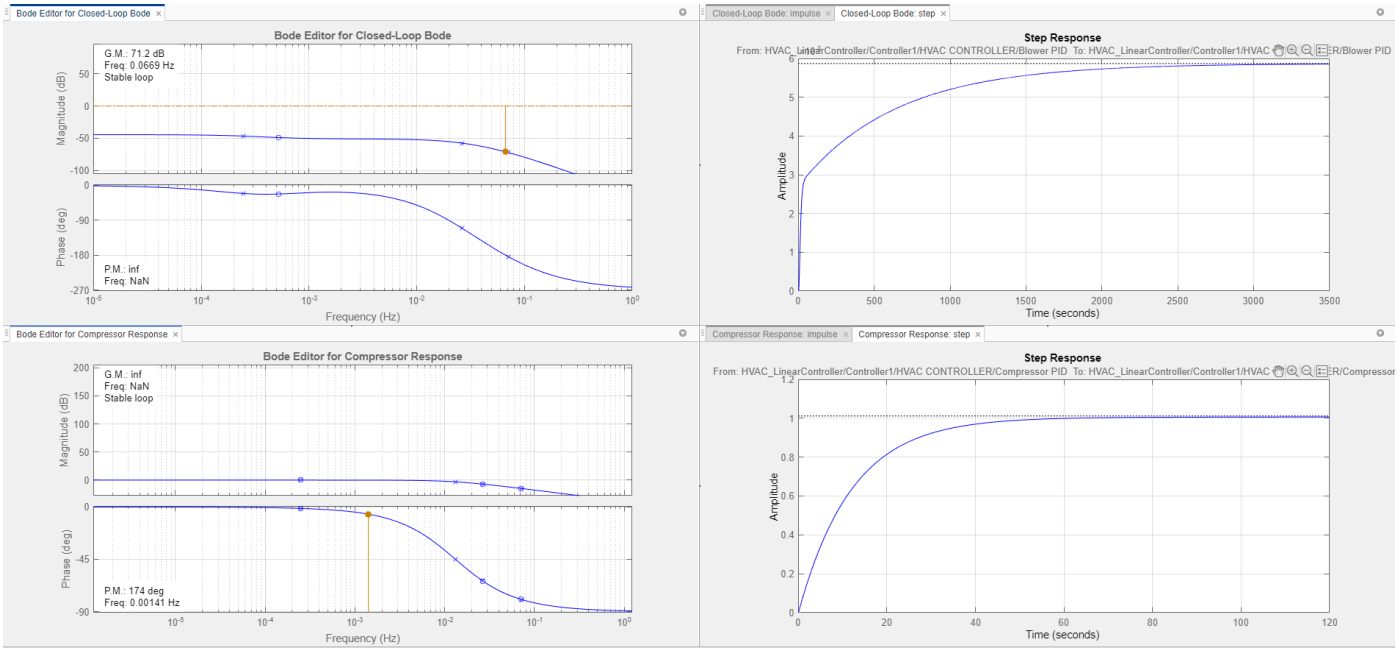


Figure 5.4: Closed-loop and compressor Bode, closed-loop and compressor step responses (from top to bottom and left to right) with $G_{control}(s) = 1$

The controllers are graphically designed by loop-shaping using the *Control System Designer* Matlab tool.

The proposed controllers, finally, are shown in Table 5.1.

Table 5.1: Controllers designed PID parameters

	Compressor	Blower
(**P**) Proportional	5.8738	284.77
(**I**) Integral	0.06247	48.969
(**D**) Derivative	0.21915	19.275
(**N**) Derivative filter	1000	1000

Table 5.2: Controllers poles/zeros location

	Compressor Units		Blower Units	
	rad/s	Hz	rad/s	Hz
ω_{p_1}	Origin			
ω_{p_2}	1000	159.15	1000	159.15
ω_{z_1}	0.0106	0.00169	0.174	0.0277
ω_{z_2}	26.1	4.154	14.4	2.29

During the control design by the bode shaping method, two different scenarios with dif-

ferent performance are found. Although the compressor loop shall be designed independently on the close-loop to assure assumptions fulfillment, two performances can be obtained from these scenarios. Initially, the cabin temperature and the initial conditions of the thermal loads set the cabin thermal state. Thus, a huge step is entering the system and the controllers set the actuators control variables accordingly. Under this situation, the actuators, eventually, saturates to provide a rapid response to these conditions. However, in near steady-state conditions, any thermal perturbation, i.e. a change request of the desired temperature or PMV, has a response according to the static controller already designed parameters. Thus, there is an important trade-off between control performance out of the steady-state conditions and when the desired thermal state is achieved. Two controllers can be designed depending on the desired performance. However, in this work is simply used a mix of both. It is required, then, that the comfort thermal state is reached as fast as possible without degrading steady-state performance defined above.

The control parameters and the location of the introduced poles and zeros is expressed in Tables 5.1 and 5.2, respectively.

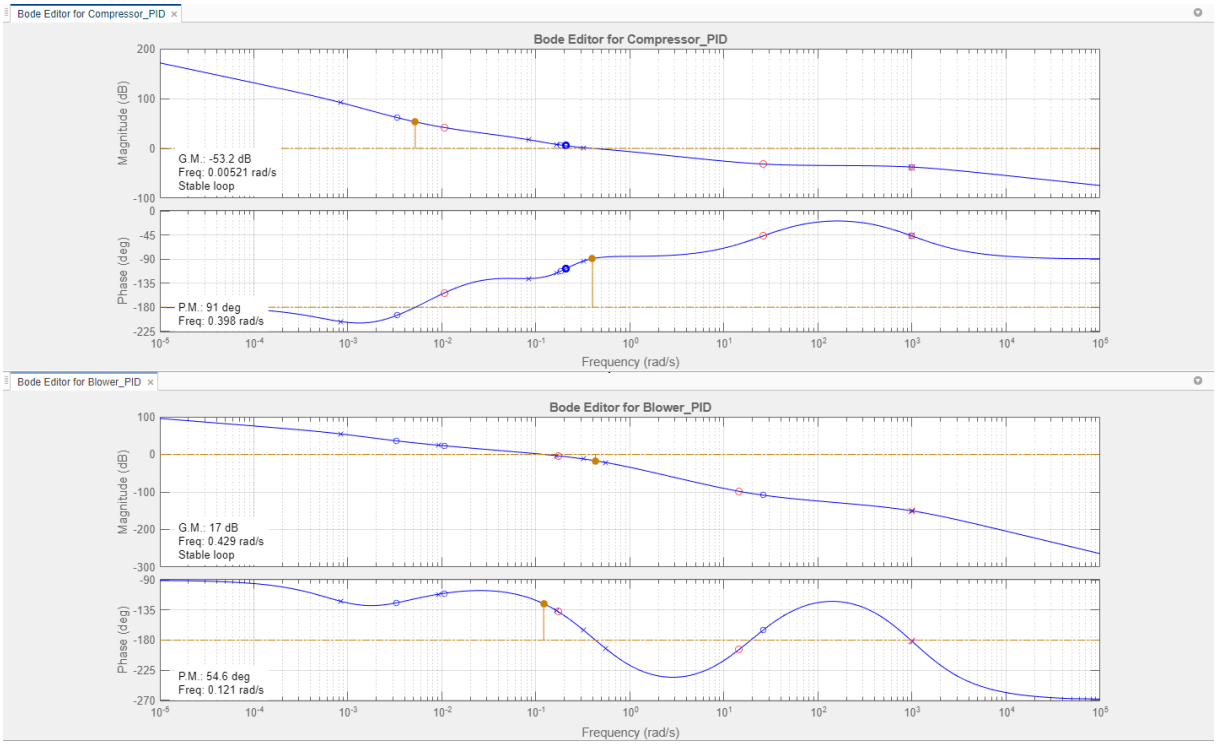


Figure 5.5: Closed-loop and compressor Bode with the proposed controllers (Table 5.1)

Table 5.3: Controllers close-loop performance

	Compressor	Blower
ω_{cross} [rad/s | Hz]	0.398 | 0.0633	0.121 | 0.0193
Phase Margin [º]	91	54.6
Gain Margin [dB]	∞	17

The proposed controllers give frequency responses summarized in Table 5.3. Note that the Matlab tool states that the gain margin for the inner close-loop is negative, thus unstable. This is not true in reality. The compressor bode plot is referred to the signal error that, in this case, is sign inverted. The outer loop has a phase shift of 180°since negative temperature error (reference below cabin temperature) shall increase the blower speed. Compressor speed error do not follow this logic, when the speed reference is below the actual one, the actuator speed shall decrease. The bodes are referred, then, to the temperature error that in one case is 180°phase shifted. Thus, the compressor gain margin is defined when the phase crosses 0 or -360°, at very high frequencies.

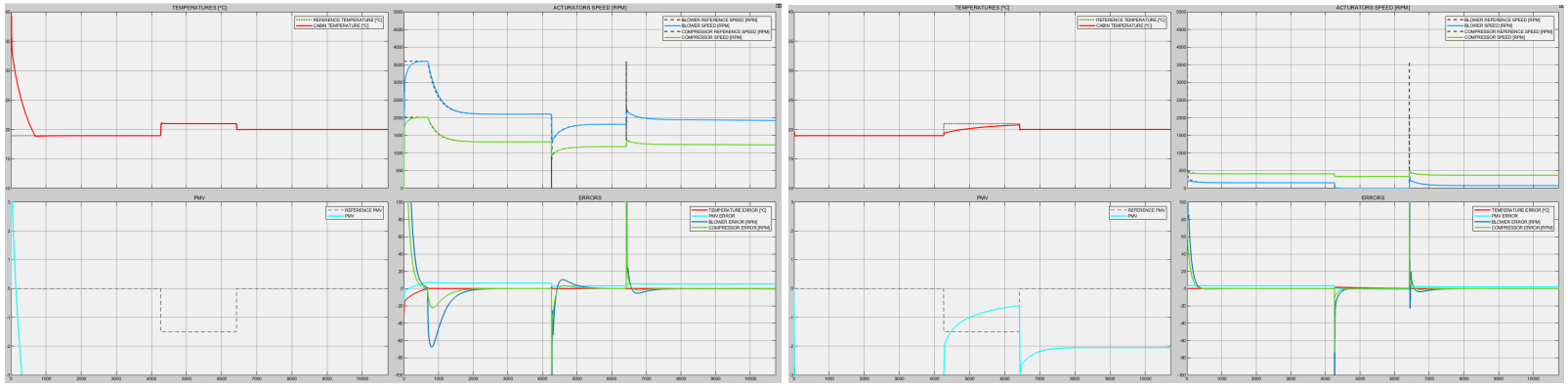
5.2 Temperature-based controller

Taking the cabin temperature as the feedback signal of the system, the system fulfills its general purpose (see Figure 5.6 and 5.7). In fact, it has been tuned the controllers such that the power consumption is minimized (see Figure 5.8 and 5.9). The control tuning for assuring thermal control state has been quite eased since the derived model in Chapter 4. HVAC system model has been simplified and analyzed with the described Matlab tools. The location of the control poles and zeros in the system could still vary without affecting cabin control state too much, so, the controllers can be furtherly optimized according to more specific system requirements.

Note that the hysteresis can also be optimized. In these results, the temperature hysteresis has been 1.5 °C. Thus, the system operates in cooling mode since thermal state under the simulation conditions trends to be hotter than desired all time. In fact, it has been forced the initial conditions of the scenario 4 so as the heating mode is enabled. The heating system is also able to regulate the temperature and is also robust against setpoint changes.

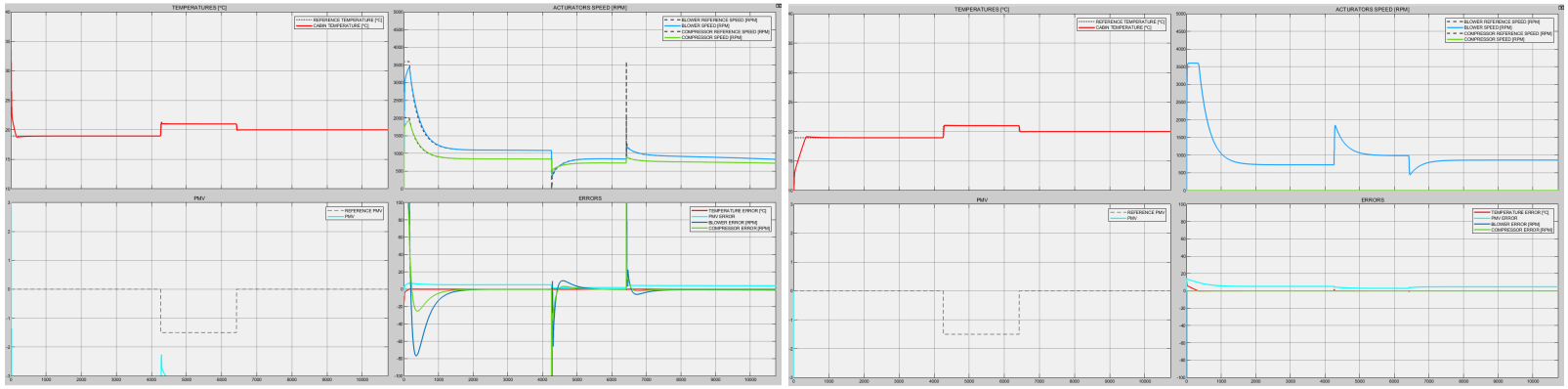
Also note that the PMV estimate is suggesting that there would be a cold thermal discomfort. This is expectable since the PMV has dependence on several variables that are not controlled in this case. Indeed, the chosen control strategy limits the control actuation to assure both thermal comfort and user temperature setpoint tracking.

Regarding the HVAC power consumption, it is obtained that the compressor is the most consuming element of the system. Even compared to the blower consumption. The controllers tuning has considered the total energy consumed as index to optimize. Thus, it is desired small overshoot but fast compressor dynamics.



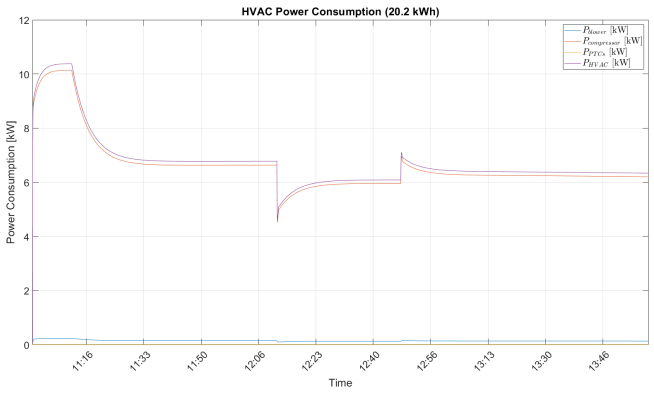
(a) Scenario 1 (b) Scenario 2

Figure 5.6: Temperature, actuators speeds and errors in summer conditions. Temperature-based

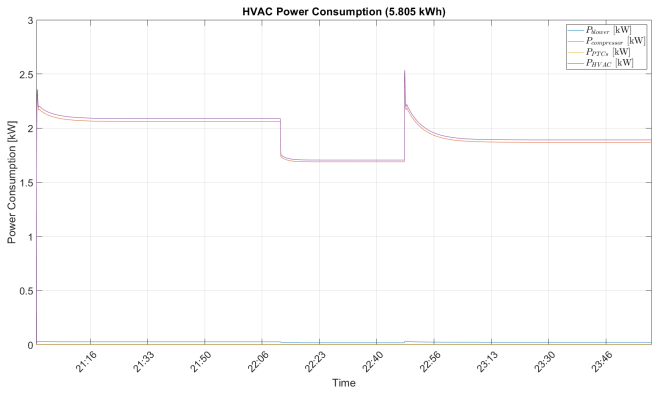


(a) Scenario 3 (b) Scenario 4

Figure 5.7: Temperature, actuators speeds and errors in winter conditions. Temperature-based.



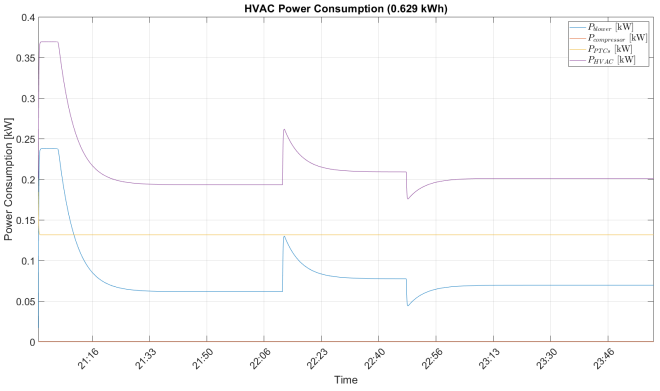
(a) Scenario 1



(b) Scenario 2

Figure 5.8: HVAC power consumption in summer conditions. Temperature-based

(a) Scenario 3



(b) Scenario 4

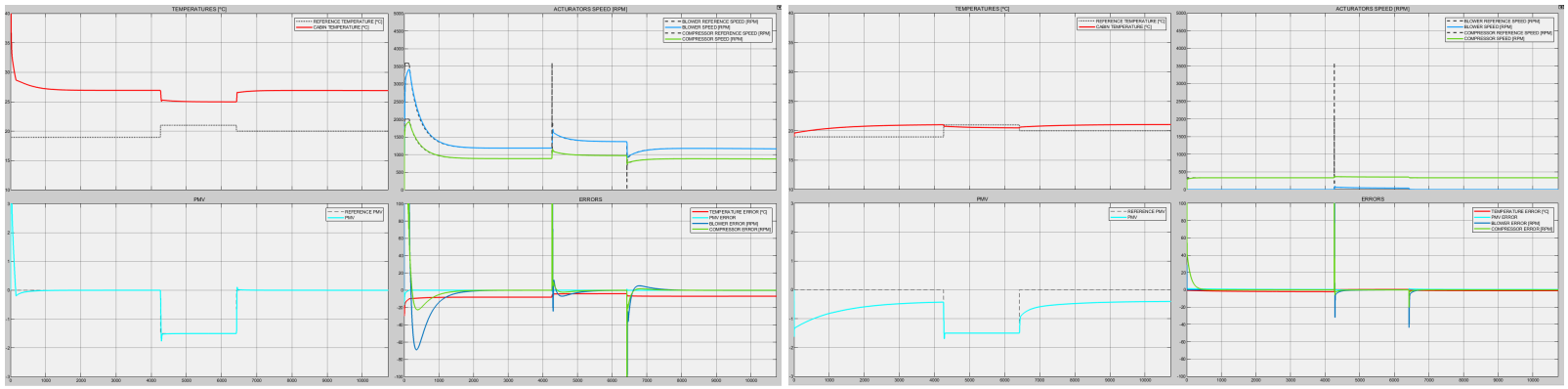
Figure 5.9: HVAC power consumption in winter conditions. Temperature-based

5.3 Thermal comfort-based controller

The same controller parameters are used for the PMV-based control and the same scenarios are simulated (see from Figure 5.10 to 5.13).

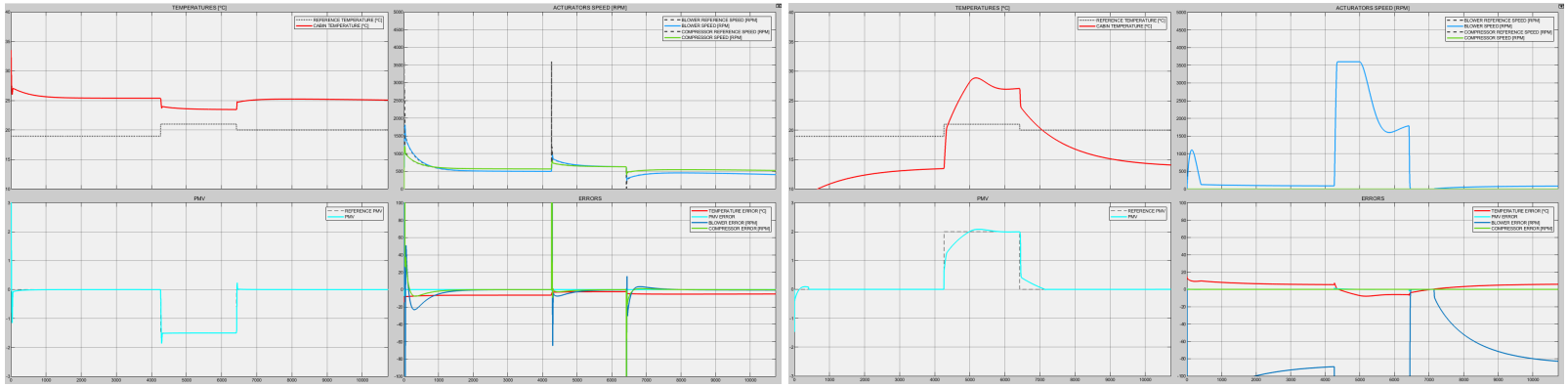
From the hot scenarios, it is seen that cabin temperature is over the reference one since the variable to control, in this case, is the PMV. This is consistent with the fact that even in hot environments, the air velocity that the passengers feel help metabolic heat dissipation, thus, gaining thermal satisfaction. In fact, control variables present smoother transitions since the PMV estimate changes slower than temperature cabin itself. The negative effect of the multivariable thermal estimate is shown in Scenario 4. Where it can be seen that the perturbation is not properly followed. This is more likely to happen in heating scenarios since the PMV increases as cabin temperature increases and as air velocity decreases sharing the same mechanism, the provided blower speed. Note that it could be possible to consider both, air velocity and cabin temperature at the same time to properly control the thermal comfort but, as commented, this leads to more complex controllers that hardly are expected to be linear.

The main benefit of this controller is the power consumption reduction. The total energy consumed by the HVAC system is always smaller than in the temperature-based topology. In fact, the presented results suggest the power consumption minimization can be optimized assuring thermal comfort, for instance, weighting the contribution to the total thermal comfort index differently according to the exposed trade-off between thermal comfort dynamics and power consumption.



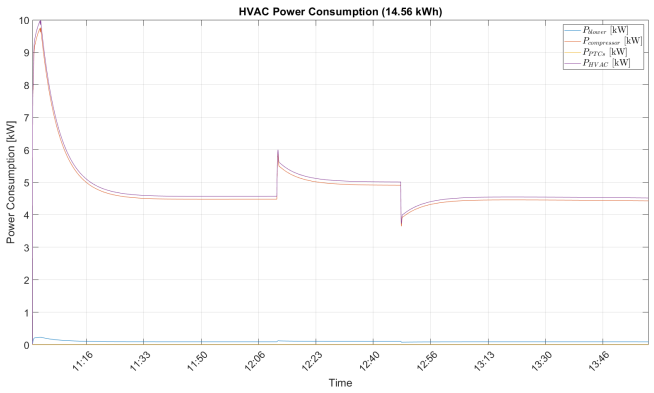
(a) Scenario 1 (b) Scenario 2

Figure 5.10: Temperature, actuators speeds and errors in summer conditions. PMV-based

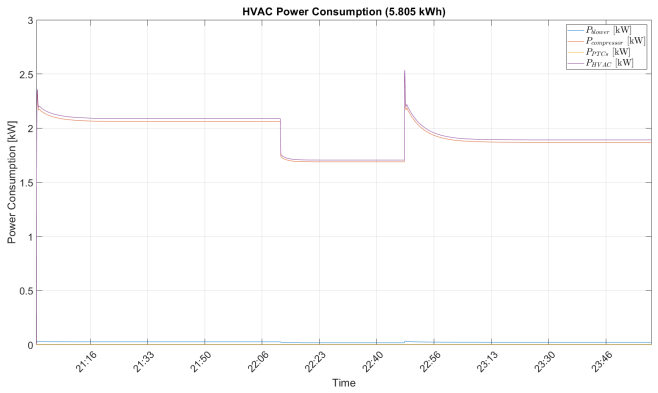


(a) Scenario 3 (b) Scenario 4

Figure 5.11: Temperature, actuators speeds and errors in winter conditions. PMV-based

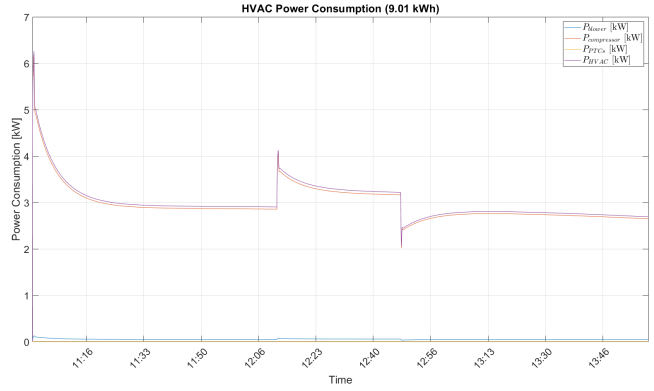


(a) Scenario 1

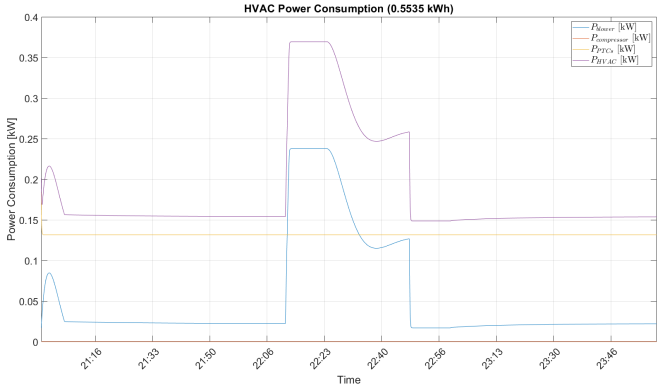


(b) Scenario 2

Figure 5.12: HVAC power consumption in summer conditions. PMV-based



(a) Scenario 3



(b) Scenario 4

Figure 5.13: HVAC power consumption in winter conditions

5.4 Control performance review

The controller performance has been simulated with the simplified model since it has been the first validation step. Now, the same designed controlled is simulated in the nonlinear model to observe how does it behaves in a more realistic driving scenario.

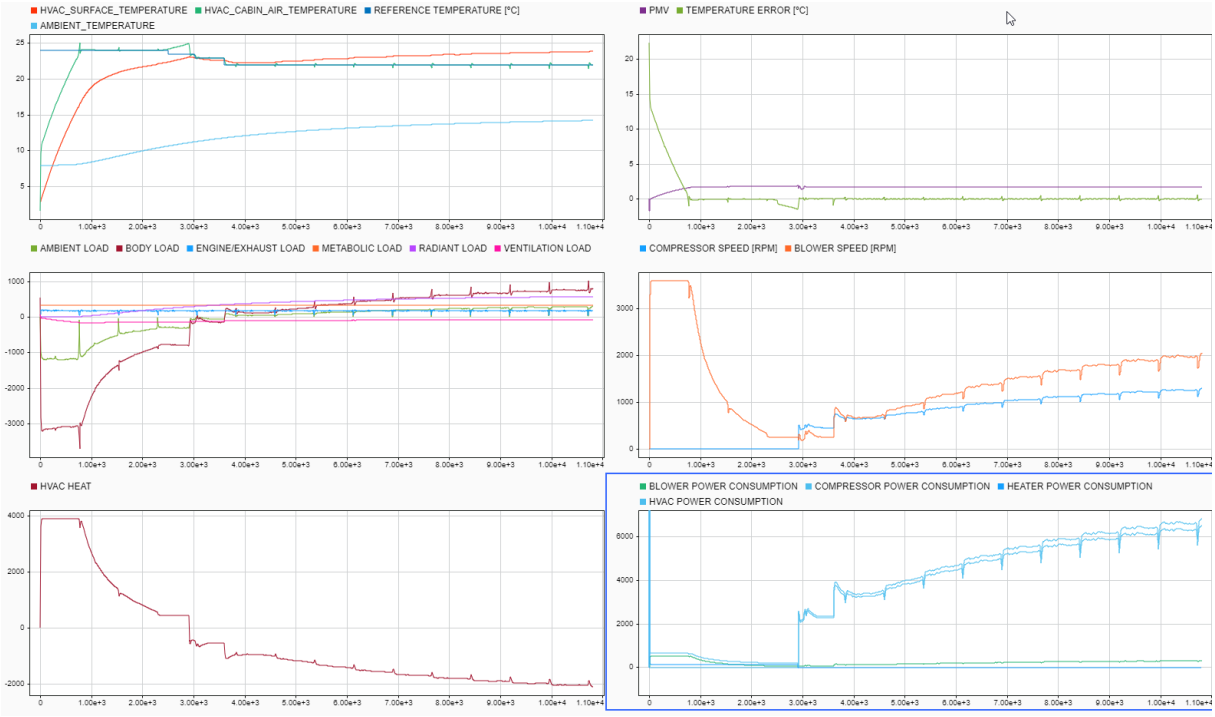


Figure 5.14: HVAC system overall performance temperature-based controller in winter morning

It can be observed in Figure 5.14 how the mode management performs. The initial state cabin is cold, without solar radiation thermal load, thus, the control system enters in heating mode to reach the cabin temperature setpoint. The blower is saturated, so the thermal dynamics only depends on the actual thermal loads. When the reference temperature is reached, the blower is controlled so that the temperature can be stabilized. Note that due to the driving conditions, mainly the vehicle speed, some thermal loads vary, thus, the controller must adapt itself to reject this perturbations. As it does. Also note that the main contributor to the cabin total heat is the heat transfer to the body elements. This is expected since there is an important temperature difference between the cabin and the outside.

Radiant load starts increasing as sun rises. Thus, the required HVAC heat is smaller down to switch off the blower in some time interval. This occurs until the temperature hysteresis is passed. Then, the system enters in cooling mode and both blower and compressor switch on to continue the regulation.

From the compressor and blower speed plot, it can be seen how they react to thermal perturbations and how the system state changes.

To sum up, it can be concluded that the control performance works similar in both models, linear and non linear, thanks to the meticulous simplifications and the assumptions made. It is

also worth it to say that the control can furtherly designed according to more specific requirements using any model.

Chapter 6

Project applications. Hardware-in-the-Loop

As already commented, modeling systems provide wide usefulness range depending on the direct application the model is built for. In this case, the main application of the HVAC modeling allows to understand how this system behaves before the prototype is built so that electronic HW can be designed accordingly. In an automotive HW and SW product design context, the implementation of this model allows to explore the performance of the entire system while the product is still in the design phase. To do so, the model and the designed controller are designed so that they can be furtherly tested even without the actual mechanical prototype built yet. Particularly, the model is built in a SW environment able to interface a real-time machine so that it can be tested both, control performance and low level HW and SW blocks of the BCM. Rapid Control Prototyping (RPC) can be done. It is a workfow that aims to expedite the development process of control strategies. Experimental iterations can be rapidly done to identify and resolve potential problems. It can be expanded safer the control tuning since all cases can be simulated in the model, even the ones in the operation limits.

Therefore, the main application of the developed model has been the first control scheme design, since the model is oriented in that way. In addition, the built model is able to interface HIL machine so that the BCM product is able to operate without the actual external HW mounted. This allows to test functional blocks independently at a higher abstraction level, i.e. it can be tested how MCU computational load is affected when the system is run and how signals are driven to the modeled actuators, to name few.

In the automotive industry is commonly used these modeling approaches since ECU testing can be automated. HIL machines integrate I/O connectivity so that it can be accessed to relevant ECU interfaces, including the communications protocols used in the automotive industry.

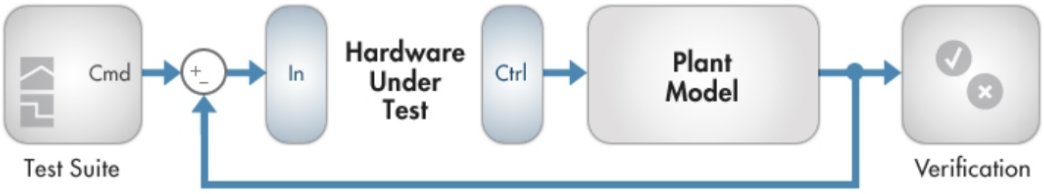


Figure 6.1: Real-time simulation with HIL scheme

During the project development, then, some analysis has been done to properly interface a particular HIL machine to finally provide a useful tool for real-time testing and control tuning according to the product requirements. Note that the provided model and control do not follow any particular product requirement other than real vehicle data, such as geometric data. This has been the intention, since, the project scope is to built a parameterized model to provide a general tool to fine tune a more specific product.

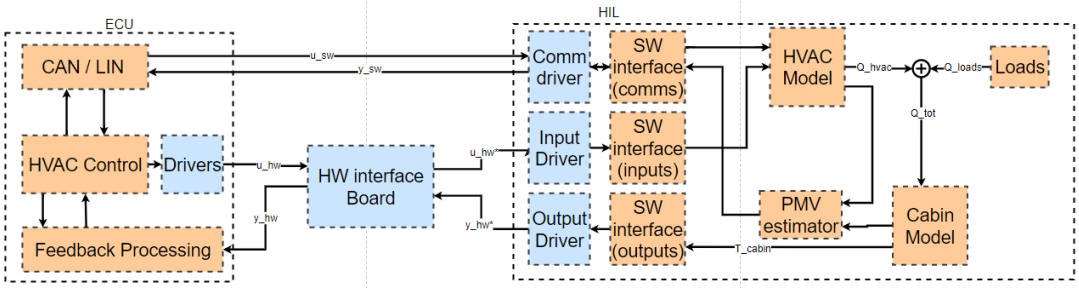


Figure 6.2: Proposed real-time simulation with HIL testbench diagram

The integration of this project model together with HIL machines can be done following the diagram depicted in Figure 6.2.

6.1 Speedgoat

The proposed HIL environment is based on the *Performance real-time target machine*, from *speedgoat* [20]. This target machine offers high computational load capability using high-speed CPU and FPGAs. Due to its high-speed performance, it can be used in control loop design operating up to hundreds of MHz. It is also modular, so several HW interfaces can be plugged-in depending on the application. In this case, it is required to establish CAN and LIN communication, digital, i.e. PWMs, and analog, i.e. HVAC sensors response, signal monitoring.

In particular, this machine offers HW interface flexibility due to its general interface with some proper capabilities. Following the proposed diagram in Figure 6.2, three main drivers are required:

- Analog I/O: the ECU monitors temperature, pressure, humidity and solar radiation from the HVAC sensors. Thus, the model run in the target-machine shall translate the emulated corresponding physical variable to voltage so that the ECU is able to response consequently. Eventually, the ECU would produce some analog signal expecting some certain response. Thus, the analog interface shall also be capable of monitoring analog signals.

- Digital I/O: some PWM signals are used so, the target-machine shall be able to read them. There are also some digital signals that the target-machine shall generate emulating switches, for instance.

- Communications: vehicle elements usually are operated by means of automotive communication protocols. Thus, the target-machine shall be able to emulate the controlled communication interface so that the target machine is connected in-the-loop with the system network.

The used I/O modules are: IO135, for analog and digital interface, and IO611, for communication interface.

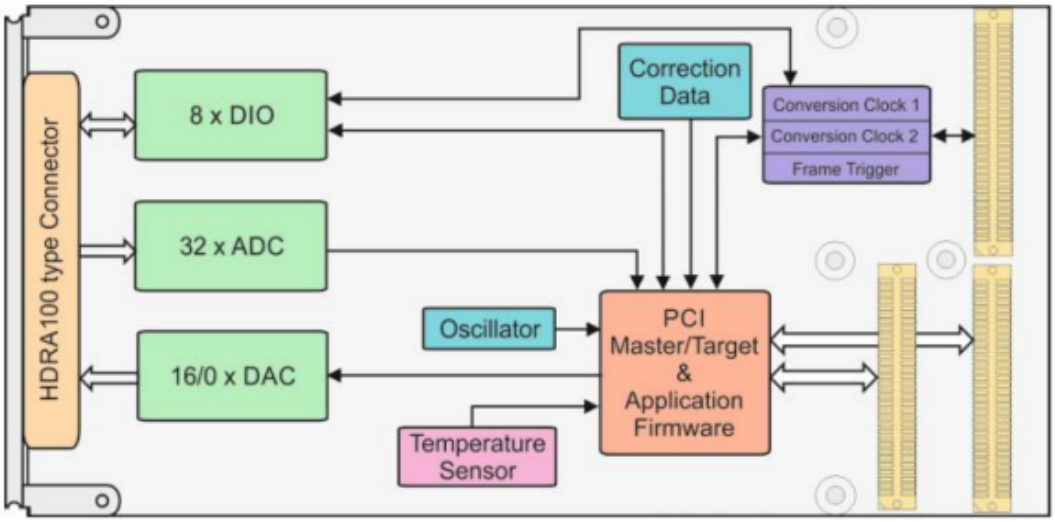


Figure 6.3: IO135 module block diagram

6.2 Hardware interface

The IO135 module offers 8 Digital I/O (DIO), 32 Analog-to-Digital Converter (ADC), and 16 Digital-to-Analog Converter (DAC) ports. Particularly, there are 32 analog ports, 16 are fixed ADC and the remaining can be configured either with ADCs or DACs. These analog interface is powered supply by either 5 or 3.3 volts and are ESD protected. They also have their own buffer in order to drive or read analog signals properly. Note that both analog and digital signals can be referred to battery voltage levels, typically 12V. Thus, analog and digital signals shall be step-down to prevent module damaging. In addition, the analog ports have some configuration flexibility. Thus, a HW interface shall be implemented accordingly.

According to the project philosophy, a generic PCB is designed to interface any given ECU and this HIL machine based on generic automotive electronics requirements and the module specifications.

The designed board schematic can be found in Appendix C. HW interface. Since the modules do not offer access to the supply rails, except for the reference ground, a power supply block shall be implemented.

6.2.1 Digital interface

The digital interface consist of a NMOSFET with the gate terminal connected to the higher voltage supply rail. A digital signal can be propagated as follows: when a '0' is forced in the source terminal, the threshold voltage increases and the transistor conducts, thus, connecting both sides with a low impedance connection, the $R_{ds_{on}}$ of the transistor. When there is a '0' in the drain terminal, given that the pull-up resistor in the source terminal produces a greater voltage with respect to the drain terminal, the transistor conducts though the body diode. In the case of transmitting '1's, they are just obtained by the corresponding pull-ups configuration.

The chosen transistor and the resistance value are determined according to the HIL HW interface. Note that it is required an open-drain topology from the ECU side, as usually are.

6.2.2 Analog interface

The analog interface consists of two different topologies since some analog ports in the HIL side are fixed to be ADCs. Thus, the analog interface has two sub-blocks: the bidirectional and the unidirectional signal conditioning circuits. The conditioning circuit consists of clamping diodes to protect signals against overvoltages but also to limit the maximum voltage level to the supply rail. Then, the signal is filtered by an RC filter and is buffered.

The bidirectional functionality is given by an analog multiplexer, that, is user configurable by means of a mechanical switch. When the analog ports of the HIL are configured as ADC, the signal shall be propagated from ECU to HIL and the mechanical switch must be in its proper position to allow the signal to pass the multiplexer. Is as simple as changing the HIL analog port configuration to enable the DACs together with the switch position. In this case, it is allowed to propagate a signal to the ECU generated by the HIL.

Note that there are some cases in which the ECU can be the signal emitter and the HW interface is set inversely. In these cases, the protection circuit disables the multiplexer since an abnormal situation is detected by means of a comparator. A red LED shines under this

conditions to alert the user.

6.2.3 Communication interface

Regarding the communications required, both LIN and CAN, the IO611 module provides two DB9 standard connectors so that each of them can manage one LIN and one CAN communication since there is enough room in the connector pinout to hold both communications. In fact, each DB9 connector has the capability to either use Flexible Data CAN (CAN-FD) and classical CAN protocols.

In this case, it is just implemented a harness with the proper connectors to be able to connect the HIL to any DB9 connector available in the ECU.

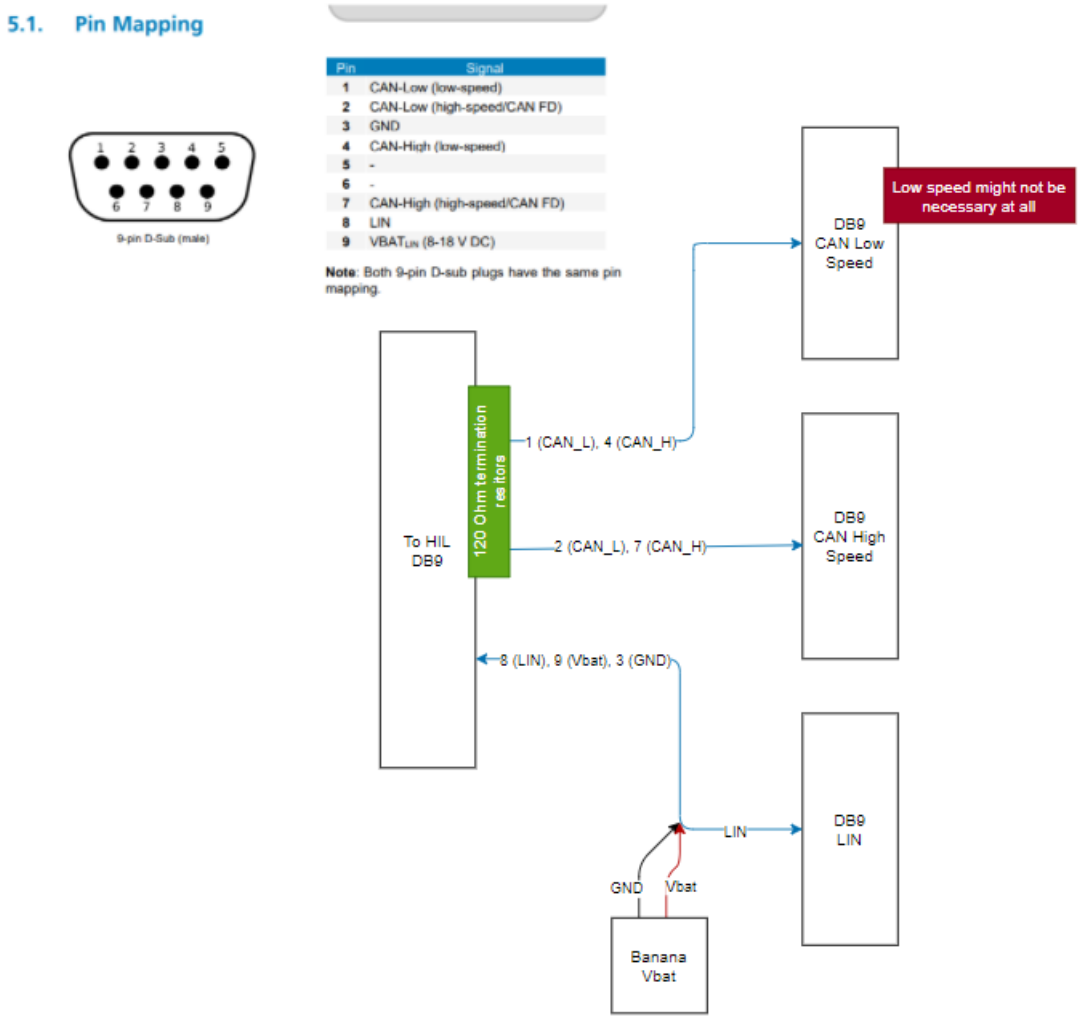


Figure 6.4: Communication interface harness definition

6.3 Software interface

SW drivers are provided together with the HW modules. Speedgoat, then, offers their HIL solution with embedded SW blocksets that directly drive the module ports. Since they are already implemented, only it has been tested in a loop-back configuration. To do so, the ADC and DAC ports are interconnected so that the signals generated by the DAC are read by the ADC. Similarly, the digital ports are interconnected and properly configured being well-defined which port sends and which receives. The communication ports are also interconnected being one the sender and the other the receiver. It is prepared a simulation setup that generates an arbitrary signal in the analog ports, a 50% duty cycle PWM in the digital ports and some data frames are transmitted via LIN and CAN.

This testbench allows to validate the proper port configuration and access by the real-time simulation machine. By receiving the analog arbitrary signal, both ADC and DAC drivers are validated. Similarly, the digital drivers can also be validated. Regarding the communications, it is additionally prepared a data packing and unpacking blocks to graphically perceive what is received.

Chapter 7

Conclusions

In an automotive product design context, an HVAC system is modeled. Some generalizations have been made to allow to explore system behavior under different driving conditions. Thus, a parameterized HVAC system model results from this work. The modeling has been control-oriented so that a second system model has been developed in order to ease the control design by means of making assumptions and linearizations. With this simplified model, a control strategy is addressed and designed to, later, simulate the overall representative model performance under realistic driving conditions. The simulation results show that both, the system model and its designed control are valid under these realistic conditions. Thus, the model is able to be particularized for the automotive product specific requirements and it is provided as a tool for testing. In addition, a HW interface board is designed to properly interface an actual real-time simulation machine that allows fast control prototyping by simulating the particularized model in real-time and in-the-loop.

Objectives completion

As initially stated, the aim of this project has been the model development for an actual HVAC system. Thus, a parameterized model has been developed and validated via simulations. The model starts with the HVAC system conception and by exploring the thermal processes involved in the system, a representative model is proposed from scratch based on the already analyzed and modeled systems in the literature.

A linear controller has been designed and validated as a first control approach. Two controllers are analyzed according to the control objective, either temperature or thermal comfort state regulation. It can be concluded from the obtained results that different objectives can be addressed when designing the controller. The proposed controller allows some flexibility so that it can be furtherly optimized accounting for other system requirements, such as overall power consumption. In this sense, PMV-based controller is promising better overall performance despite the fact that a more complex controller than the presented here might be required.

Moreover, from the implemented model and after its validation with a particular designed controller, it is developed a testbench tool based on HIL real-time simulation machine.

Thus, it can be stated that the project objectives have been successfully accomplished. The project results can be furtherly used in the commented application context: real-time testing and fast control prototyping.

Next steps

Project results can be furtherly optimized. The given model is limited to the considerations taken for simplicity and some lack of information of the system to model. Even they try to be realistic, more complex driving scenarios can be implemented and thermal load variations can be analyzed to obtain more information about the system behavior. Next modeling loop could be the computation optimization of the actual model. Depending on the modeling purpose, it might not be necessary to tune the model more accurately since it is representative enough. However, it can be analyzed how accurate the actual model is by performing some tests with the mechanical prototype already built and comparing the results with the model.

Regarding the controller, it has been developed linear controller by means of a designed control strategy. It has been exposed the thermal comfort regulation issues in cold environments and, the control strategy shall be reviewed in this case. As next control design steps, the control strategy can be optimized according to actual product requirements. For instance, a multivariable controller can be addressed by means of controlling both, temperature and thermal comfort state at the same time. This requires a change in the control topology. Besides, even the control performance has resulted in specification fulfillment, the controller can be redesigned in order to also account for the power consumption optimization as this system constitutes one of the most consuming vehicle elements. In particular, it has been seen that there are two different control scenarios, thus, static controllers might not be properly optimized to provide the same performance in all cases. Some more complex control topologies can be used: non-linear, Fuzzy logic based, adaptative-PIDs, etc. Only with well-defined system requirement, a proper controller and model can be implemented. Otherwise, the modeling and controlling strategy has to be addressed in general terms with the model parameterized and ready to be put into numbers with real data.

Finally, the implemented PCB for HIL interfacing any ECU with the HIL machine, opens a door for continuing the project since the results obtained can be entirely used as a testing tool or even a real-time simulation tool.

Lessons learned

Since the very beginning, this project has been challenging. From the personal point of view, collaborating with such a professional team has made me better engineer by means of sharing results, commenting them, and also by listening to the supervisors advice. Some sections of the project has been possible thanks to the teamwork and the good communication between project team.

From the technical point of view, neither me nor the project team had previous knowledge and experience regarding this topic. In fact, my thermodynamics background was quite limited. Thanks to the research done and the analysis of the state of the art, I have learned both, system modeling techniques and thermal process dynamics. I never have not faced any modeling project like this but, in fact, the development of this project together with the gained knowledge encourage me for future projects, since I have found motivation in the research field and the study of different applications regarding my past background. What it seemed, initially, to be a very limited project has become an implementation reality and the project has been orientated according to real product design needs. In addition, some new simulation tools have been used during the modeling and it also has been thanks to the utilization and learning of them that the project has flourished. There is a general satisfaction regarding the obtained results.

To conclude, then, the project carried out at Idneo opens a door for the HW team to continue with the implementation of new interesting testing and design techniques and, some of them, can be resulted from this project. I am grateful of being part of the first physical model development of the team and of starting a new technical branch not used yet before, physical system modeling before prototyping to accelerate both HW and SW automotive product designs.

References

[1] Daniel Sörensen. *The Automotive Development Process*. [*A Real Option Analysis*]. Dissertation Universität Stuttgart, 2006

[2] Kim B. Clark, W. Bruce Chew, Takahiro Fujimoto, John Meyer and F. M. Scherer. *Development in the World Auto Industry*. Brookings Papers on Economic Activity, 1987.

[3] Tan, J.; Otto, K.; Wood, K. *A Comparison Of Design Decisions Made Early And Late In Development* Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 2: Design Processes | Design Organisation and Management, Vancouver, Canada, 21.-25.08.2017

[4] M. Dawson, D.N. Burrel, E. Rahim, S. Brewster. *Integrated Software Assurance into the Software Development Life Cycle* (*SDLC*). Journal of information systems technology & planning, 2010.

[5] R. Bean, B.W. Olesen, K. Woo. *History of Radiant Heating & Cooling Sysyems*. ASHRAE Jpurnal, 2010.

[6] X. Yan, J. Fleming, R. Lot. *A/C Energy Management and Vehicle Cabin Thermal Comfort Control*. Control Engineering Practice, Volume 8, Issue 6, 2000.

[7] John G. Ingersoll, Thomas G. Kalman, Linda M. Maxwell. *Automobile Passenger Compartment Thermal Comfort Model - Part II: Human Thermal Comfort Calculation*. 2018, University of Leeds, SAE International.

[8] Y. Farzaneh, Ali A. Tootoonchi. *Controlling Automobile Thermal Comfort Using Optimized Fuzzy Controller*. Applied Thermal Engineering. Deparment of Mechanical Engineering, Ferdowsi University of Mashlad, Iran.

[9] Raad Z. Homod. *Review on the HVAC System Modeling Types and the Shortcomings of Their Application*. Department of Mechanical Engineering, University of Basrah, Iraq

[10] H. Khayyam, Abbas Z. Kouzani, Eric J. Hu. *Reducing Energy Consumption of Vehicle Air Conditioning System by an Energy Management System*. 2009 IEEE Intelligent Vehicles Symposium.

[11] I. Cvok, B. Škugor, J. Deur. *Control Trajectory Optimisation and Optimised Control Strategy for an Electric Vehicle HVAC System and Favourable Thermal Comfort*. Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia.

[12] M. Ali Fayazbakhsh, M. Bahrami. *Comprehensive Modeling of Vehicle Air Conditioning Loads Using Heat Balance Method*. 2013, SAE International, Simon Fraser University.

[13] A. Alexandrov, V. Kudriavtsev, M. Reggio. *Analysis of Flow Patterns and Heat Transfer in Generic Passenger Car Mini-Environment*. 2001, 9th Annual Conference of the CFD Society of Canada, Kitchener, Ontario.

[14] W. Ott, N. Klepeis, P. Switzer. *Air Change Rates of Motor Vehicles and In-Vehicle Pollutant Concentrations from Secondhand Smoke*. 2007, Stanford University, California, USA. Journal of Exposure Science and Environmental Epidemiology.

[15] David E. Claridge, Frederick H. Kohloss, Brian A. Rock, T. David Underwood, Michael W. Woodford. *ASHRAE Handbook of Fundamentals*. ASHRAE.

[16] Öner Arici, Song-Lin Yang. *Computer Model for Automobile Climate Control System Simulation and Application*. Mechanical Engineering and Engineering Mechanics, Michigan Technological University, USA.

[17] I. Ratkovic, I.Cvok, V. Soldo, J. Deur. *Control-oriented Modelling of Vapour-Compression Cycle Including Model-order Reduction and Analysis Tools*. 2020, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia.

[18] R. Musat, E. Helerea. *Characteristics of the PTC Heater Used in Automotive HVAC Systems*. 2010, Department of Electrical Engineering, Faculty of Electrical Engineering and Computer Science, Transilvania University, Romania.

[19] C. Bohn, D.P. Atherton. *An Analysis Package Comparing PID Anti-Windup Strategies*. 1995, IEEE Control Systems.

[20] *https://www.speedgoat.com*.

[21] VDA QMC Working Group 13 / Automotive SIG. *Automotive SICE Process Assessment / Reference Model* (Version 3.1).

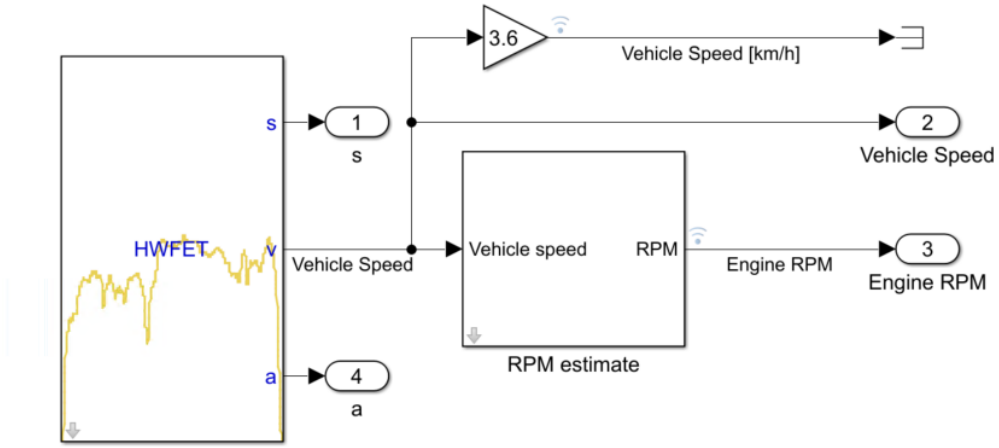
Appendix A

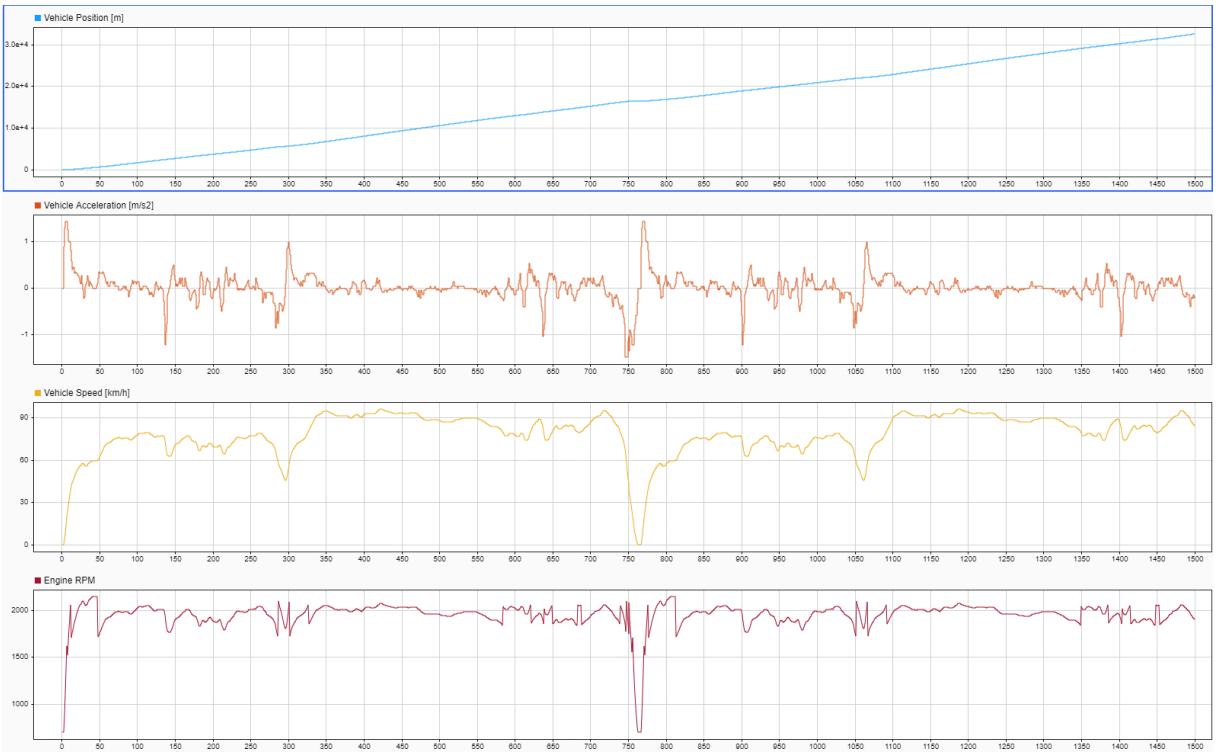
HVAC modeling. Details

A.1 Driving conditions

A.1.1 Driving profile

It is used the driving cycle Simulink block provided by Daniel J. Auger (`https://es.mathworks.com/matlabcentral/fileexchange/46777-driving-cycle-simulink-block`). This block can produce several different standard driving cycles used for automotive emissions, range and energy consumption testing. It outputs the vehicle speed, position, and acceleration according to the uploaded driving cycle. The used driving cycle in this work is the HWFET one:





Additionally, it is implemented a block that estimates the engine revolutions according to a gear ratio setting. The wheel speed estimator together with the gear ratio estimate provide the engine speed for the engine and exhaust temperature estimate.

A.2 Equation of time

In order to compute the solar radiation, typically solar data is referred to the absolute position of the sun in the sky. Thus, is possible knowing the location location and the time to calculate the corresponding radiation. The location is set by the latitude and longitude of the location itself and it can be used to refer every sun position to the vehicle Earth location.

The equation of time is used for the relative position of the sun to the vehicle according to the day and solar time. Usual clocks define the day duration equitatively with 24 hours duration. In reality, this is not true, there are longer summer days than winter ones, i.e.. In addition, solar declination also depends on the season and the location of the sun in the sky varies significantly depending on the considered season.

All angles are in degrees. The solar azimuth ϕ and the surface azimuth Φ are measured in degrees form south; angles to the east of south are negative, and angles to the west of south are positive. Calculate solar altitude, azimuth, and surface incident angles as follows: Apparent solar time AST, in decimal hours:

$$AST = LST + \frac{ET}{60} + \frac{LSM - LON}{15}$$

Hour angle H, degrees: ‘

$$H = 15 \cdot (AST - 12)$$

Solar altitude β;

$$\sin\beta = \cos L \cos\delta \cos H + \sin L \sin\delta$$

Solar azimuth ϕ:

$$\cos\phi = \frac{\sin\beta \sin L - \sin\delta}{\cos\beta \cos L}$$

Surface-solar azimuth γ:

$$\gamma = \phi - \Phi$$

Incident angle Θ:

$$\cos\Theta = \cos\beta \cos\gamma \sin\sigma + \sin\beta\Sigma$$

Where:

ET = equation of time, decimal minutes

L = latitude

LON = local longitude, decimal degrees of arc

LSM = local standard time meridian, decimal degrees of arc

LST = local standard time, decimal hours

δ = solar declination, °

Φ = surface azimuth, °

Σ = surface tilt from horizontal

If $\beta > 0$, then there is solar radiation. It occurs at sunlight hours.

Table 7 Extraterrestrial Solar Irradiance and Related Data

	E_o, W/m^2	Equation of Time, min	Declination, degrees	A W/m^2	B (Dimensionless Ratios)	C
Jan	1416	−11.2	−20.0	1230	0.142	0.058
Feb	1401	−13.9	−10.8	1215	0.144	0.060
Mar	1381	−7.5	0.0	1186	0.156	0.071
Apr	1356	1.1	11.6	1136	0.180	0.097
May	1336	3.3	20.0	1104	0.196	0.121
June	1336	−1.4	23.45	1088	0.205	0.134
July	1336	−6.2	20.6	1085	0.207	0.136
Aug	1338	−2.4	12.3	1107	0.201	0.122
Sep	1359	7.5	0.0	1151	0.177	0.092
Oct	1380	15.4	−10.5	1192	0.160	0.073
Nov	1405	13.8	−19.8	1221	0.149	0.063
Dec	1417	1.6	−23.45	1233	0.142	0.057

Appendix B

Matlab scripts

B.1 Original system script

Listing B.1: Original model Matlab script

```
%%%%% HVAC Simulation data loading and simulation %%%%%
close all; clc; clear all; warning('off','all');
%{
 - This .m file sets the input vectors used for HVAC system simulation
 - Code lines that are commented with double % refers to variables that
 might be inteoutting to set as vectors with different values in time.
 This way, variable dependance on system dynamics can be accordingly
 analyzed.
 - Geometric data is loaded from
 .\GeometricProfileData\GeometricData.mat, this file contains all
 reported data from RedDot
 %}

plot_temps          =   0;
plot_heat           =   0;
plot_both           =   0;
plot_ACload         =   0;
plot_PMV            =   0;
plot_consumption    =   0;

%%
%%%%%%%%%% SIMULATION SETTINGS %%%%%%%%%%
Tend        =   3 * 60 * 60;         % Simulation duration [min * s/min]
ts          =   0.001;                  % Sampling time [s]
AirVolume   =   1.63;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%%%%%%%%%% INITIAL CONDITIONS %%%%%%%%%%

% Time data is referred to 21st of each month in 1964
Months      =   {'Jan', 'Feb', 'Mar', 'Apr', 'May', ...
 'June', 'July', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'}; % Month
month       =   'Aug';  %%%% SELECT THE MONTH
n_month     =   find(contains(Months, month));

hour        =   6;

InitialTime =   datetime(1964, n_month, ...
21, hour , 0, 0); %%%% SELECT THE TIME HH,MIN,SS
LST0        =   InitialTime.Hour+...
InitialTime.Minute/60+ InitialTime.Second/3600;

% Location Data: BARCELONA
Latitude    =   41.3879;
Longitude   =   2.16992;
```

```
UTC         =   1;
if and(n_month >= 4, n_month <= 10), ...
UTC = 0; end   % Spain is UTC+2h from April to October (correction needed)
LSTM        =   15*UTC;

EoT_tab     =   [-11.2, -13.9, -7.5, 1.1,...
3.3, -1.4, -6.2, -2.4, 7.5, 15.4, 13.8, 1.6]; % Equation of Time [min]
Declination_tab=   [-20, -10.8, 0, 11.6, 20, 23.45, 20.6, 12.3, 0,...
-10.5, -19.8, -23.45];      % Declination [  ]
A_tab       =   [1230, 1215, 1186, 1136, 1104, 1088, 1085, 1107, 1151, ...
1192, 1221, 1233];       % A [W/m2]
B_tab       =   [0.142, 0.144, 0.156, 0.18, 0.196, 0.205, 0.207, ...
0.201, 0.177, 0.16, 0.149, 0.142];     % B [/]
C_tab       =   [0.058, 0.06, 0.071, 0.097, 0.121, 0.134, 0.136, ...
0.122, 0.092, 0.073, 0.063, 0.057];    % C [/]

AST         =   LST0 + EoT_tab(n_month)/60 + (LSTM-Longitude)/15;
H           =   15*(AST-12);
sinBeta_tab =   cosd(Latitude).*cosd(Declination_tab).*cosd(H)' ...
+ sind(Latitude).*sind(Declination_tab);
Beta        =   asin(sinBeta_tab);

Tout_min_tab=   [9, 8, 10, 13, 16, 20, 23, 23, 20, ...
17, 12, 9];     % Barcelona data
Tout_max_tab=   [15, 15, 17, 20, 23, 27, 29, ...
29, 26, 23, 18, 15];     % Barcelona data
Rad_max_tab =   [ 615.6311,  645.3447,  655.4160,  ...
656.1268,  659.6520,  660.2711,  658.3445,  653.7843,  645.8083, ...
635.2982,  607.2597,  593.5552];    % Barcelona data (from simulation)

Tout_min    =   Tout_min_tab(n_month);
Tout_max    =   Tout_max_tab(n_month);
Rad_max     =   Rad_max_tab(n_month);
%%
%%%%%%%%%%%%%% INPUT VECTORS %%%%%%%%%%%%%%

%%%%% LOADS %%%%%
% Metabolic Load
Np          =   1;              % Number of passengers other than driver
Mp          =   94.6;           %% Metabolic rate [W/m2] (passengers)
Md          =   144.5;          %% Metabolic rate [W/m2] (driver)
Adu         =   1.72;           % DuBois area estimate [m2]
% Solar Load
A           =   A_tab(n_month);
B           =   B_tab(n_month);
C           =   C_tab(n_month);
sinB        =   sinBeta_tab(n_month);
tilt        =   Beta(n_month); % Sun declination angle [  ]
% Load GMD geometric data
```

```
load('.\..\GeometricProfileData\GeometricData.mat')

dir_abs     = sum(Surface.*Absorptivity.* abs(cos((Tilt+tilt)/180*pi)));
dir_trans   = sum(Surface.*Transmissivity.* abs(cos((Tilt+tilt)/180*pi)));

dif_abs     = Surface.*Absorptivity;  %  dif_abs = dif_abs(:)';
dif_trans   = Surface.*Transmissivity; % dif_trans = dif_trans(:)';

ref_abs     = dif_abs;
ref_trans   = dif_trans;

% Ambient Load

% Ventilation Load
Pout        = 10132500;             % Pressure outside [Pa]
Pin         = Pout - 10;            % Pressure inside [Pa]
Ps          = 2.065e3;              % Water saturation pressure [Pa]
RHout       = 0.3;                 % Relative Humidity outside [%]
RHin        = 0.6;                 % Relative Humidity inside [%]

% Humidity ratio [H20[g] / Dry air[g]]
Xin         = 0.62198 * RHin*Ps/(100*Pin-RHin*Ps);
% Humidity ratio [H20[g] / Dry air[g]]
Xout        = 0.62198 * RHout*Ps/(100*Pout-RHout*Ps);

dm_vent     = 0.01;                 % Airflow leakage [kg/s]

% IC
dIdir = 0;
if sinB >= 0,
dIdir = A/exp(B/sinB);
end
dQdir = dir_trans * dIdir;
dIdif = C.*dIdir.*(1+cosd(Tilt))./2;
dQdif = sum(dif_trans .* dIdif);
dIref = (dIdir+dIdif) * 0.2 .*(1-cosd(Tilt))/2;
dQref = sum(ref_trans.* dIref);

Qhvac_ini           = -818.039;
hvac_ini            =  Qhvac_ini            * 0.0078;
Qmet_ini            = 0;(Np * Mp + Mp)*Adu;
met_ini             = Qmet_ini              * 0.0078;
Qrad_ini            = 0;(dQdir+dQdif+dQref);
rad_ini             = Qrad_ini              * 0.0359;
Tout_ini            = (Tout_min+(Tout_max-Tout_min)/Rad_max*Qrad_ini);
tout_ini            = Tout_ini              * 1;
Tcabin_ini          = hvac_ini + met_ini + rad_ini + tout_ini

Qrad_body_ini       = 0;Qrad_ini;
```

```
rad_body_ini        = Qrad_body_ini         * 0.0062;
Tout_body_ini       = Tout_ini;
tout_body_ini       = Tout_body_ini         * 0.2044;
body_ini            = Tcabin_ini            * 0.7956;
Tbody_ini           = rad_body_ini + tout_body_ini + body_ini

%% HVAC LOAD

RecirculationFlap = 0;
% DUMMY
dummyAC     =   -1;
tc          =   2 * 60;

% BLOWER
blade_angle =   75*pi/180;
blade_radius=   0.1;
Sblower     =   pi*blade_radius^2;
Vblower     =   12;
Iblower     =   22;

tau_blower  =   30/5;
% HEATER
Sheater     = 0.3142*0.184;     % [m2]

Iheater     = 32.5;
Vheater     = 48;

R01         = 1.457;
m           = 0.0064;
R02         = 0.3678*1e-3;
b           = 0.0494;
Rmax        = 31.399;
Tmin        = -50;
Tcr         = 120;
Tmax        = 250;

SHI         = 0.1;              % [W/ C ]

D_heater    = 92e-3;
rho_air     = 1.14;
mu_air      = 1.895e-5;
k_air       = 25.72e-3;
Pr_air      = 0.69;

tau_ptc     = 10;
% COMPRESSOR
Sevaporator     = Sheater   *12;
D_evaporator    = D_heater  *2.5;
k_compressor    = 5/100;
```

```
%r_compressor    = 4.85;
n_compressor    = 1.3;
Vs              = 1.8e-6    *1;
Tevaporator_out = 0;
Tevaporator_in  = -15;
Cp_refrigerant  = 0.7*1e3;
Tcondenser_in   = 120;

r_compressor    = ((Tcondenser_in+273)/...
(Tevaporator_out+273))^(n_compressor/(n_compressor-1));

VolumetricEfficiency = 1 + k_compressor ...
- k_compressor*r_compressor^(1/n_compressor);
V1 = VolumetricEfficiency * Vs;

COPmax          = (Tevaporator_out+273)/(Tcondenser_in-Tevaporator_out);

tau_compressor  = 60/5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%%%%%%%%%%%%%%% PMV estimate %%%%%%%%%%%%%
MoverA      = 58;           % [W/m2] Light Traffic
h           = 0;
pa          = 2.5e3;        % [Pa] @20 C
pa          = pa / 133;     % [mmHg]

ncl         = 4;            % Slacks, Short
fcl         = [1,   1,    1.05, 1.1,  1.1, 1.15, 1.15];     % [-]
Icl         = [0,   0.1,  0.35, 0.5,  0.6, 1,    1.5];      % [clo]
as          = [0.5, 0.55, 0.6,  0.65, 0.7, 0.75, 0.75];     % [-]

fcl = fcl(ncl); Icl = Icl(ncl); as = as(ncl);

%%%%%%%%%%%%%%% Tcl estimate %%%%%%%%%%%%%

N           = 100; % Number of iterations
Tcl_ini     = 37;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% CONTROLLER
MODE        = { Automatic Climate Control,
 Heating Control,
 Cooling Control   };
mode        = 1;
CONTROL_VARIABLE = {    Temperature-based,
 PMV-based };
control_variable = 1;
```

```
Thyst       = 1.5;
PMVhyst     = 1;

Tcomfort    =   25;
PMVcomfort  =   0;
%%%%%%%%%%%%%%%%%% PID %%%%%%%%%%%%%%%%%%

Pcompressor     = 5.8738;
Icompressor     = 0.06247;
Dcompressor     = 0.21915;
N               = 1000;

Pblower         = 284.77;
Iblower         = 48.969;
Dblower         = 19.275;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%%%%%%%%%%%%%%% RUN MODEL %%%%%%%%%%%%%%%

mod         =   'HVAC_SimulinkModel.slx';
out         =   sim(mod);
disp('Simulation finished')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%%%%%%%%%%%%%%%%% PLOTS %%%%%%%%%%%%%%%%%
clc;

fs          = 16;

total_min   = floor(Tend/60);
min_tick    = 10;
n_delta     = floor(max(max(60*min_tick/ts), length(out.TIME)/20));
n_ticks     = 1:n_delta:length(out.TIME);

% LOADS
time        = out.TIME;
time        = time(n_ticks);
time_str    = datestr(InitialTime+seconds(out.TIME),'HH:MM');
time_str    = time_str(n_ticks, :);

% Window length for moving avg [min * sec/min / (sec/samples)]
k_temp = (5*60)/ts;
k_heat = (5*60)/ts;

Mcabin  = movmean(out.CABIN_TEMPERATURE,k_temp);
Mout    = movmean(out.OUTSIDE_TEMPERATURE,k_temp);
Msurf   = movmean(out.SURFACE_TEMPERATURE, k_temp);
```

```
292
293 Mac_load        = movmean(out.AC_LOAD, k_heat);
294 Mthermal_load   = movmean(out.dQ_TOTAL,k_heat);
295 Mmetabolic      = movmean(out.METABOLIC_LOAD,k_heat);
296 Mrad            = movmean(out.RADIANT_LOAD, k_heat);
297 Mamb            = movmean(out.AMBIENT_LOAD, k_heat);
298 Meng_exh        = movmean(out.ENG_EXH_LOAD, k_heat);
299 Mvent           = movmean(out.VENTILATION_LOAD, k_heat);
300 Mbody           = movmean(out.BODY_LOAD, k_heat);
301
302 Mrad_s          = movmean(out.RADIANT_LOAD_S, k_heat);
303 Mamb_s          = movmean(out.AMBIENT_LOAD_S, k_heat);
304 Mbody_s         = movmean(out.BODY_LOAD_S, k_heat);
305 Mthermal_load_s = movmean(out.THERMAL_LOADS_S,k_heat);
306
307 if plot_heat,
308 figure('units', 'normalized','outerposition', ...
309 [0 0 1 1], 'defaultaxesfontsize', 18)
310 subplot(2,2,1)
311 plot(out.TIME, out.METABOLIC_LOAD, out.TIME,...
312 out.RADIANT_LOAD, out.TIME, out.AMBIENT_LOAD, out.TIME,...
313 out.ENG_EXH_LOAD, out.TIME, out.VENTILATION_LOAD, out.TIME, ...
314 out.BODY_LOAD, out.TIME, out.AC_LOAD); grid on;
315 legend('Metabolic Load', 'Radiant Load', 'Ambient Load', ...
316 'Engine/Exhaust Load', 'Ventilation Load', 'Body Load', 'AC Load')
317 ylabel('Thermal Load [W]');
318 xlim([out.TIME(1) out.TIME(end)]);
319 set(gca, 'xtick',       time);
320 set(gca,'XTickLabel',[])
321 set(gca, 'fontsize', fs)
322
323 title('Cabin Thermal Loads');
324
325 subplot(2,2,3)
326 plot(out.TIME, out.THERMAL_LOADS); grid on;
327 legend('Total Cabin Thermal Load')
328 xlabel('Time [s]'); ylabel('Thermal Load [W]');
329 xlim([out.TIME(1) out.TIME(end)]);
330 time = get(gca,'xtick');
331 [~,idx] = min(abs(time-out.TIME));
332 set(gca, 'xticklabel', ...
333 datestr(InitialTime+seconds(out.TIME(idx)),'HH:MM'))
334 set(gca, 'fontsize', fs)
335 xtickangle(45);
336
337 title('Total Cabin Thermal Loads');
338
339 subplot(2,2,2)
340 plot(out.TIME, out.RADIANT_LOAD_S, out.TIME, out.AMBIENT_LOAD_S,...
```

```
out.TIME, out.BODY_LOAD_S); grid on;
legend('Radiant Load', 'Ambient Load', 'Body Load')
xlim([out.TIME(1) out.TIME(end)]);
ylabel('Thermal Load [W]');
set(gca, 'xtick',       time);
set(gca,'XTickLabel',[])
set(gca, 'fontsize', fs)

title('Surface Thermal Loads');

subplot(2,2,4)
plot(out.TIME, out.THERMAL_LOADS_S); grid on;
legend('Total Surface Thermal Load')
xlabel('Time [s]'); ylabel('Thermal Load [W]');
xlim([out.TIME(1) out.TIME(end)]);
time = get(gca,'xtick');
[~,idx] = min(abs(time-out.TIME));
set(gca, 'xticklabel', ...
datestr(InitialTime+seconds(out.TIME(idx)),'HH:MM'))
set(gca, 'fontsize', fs)
xtickangle(45);

title('Total Surface Thermal Loads');

end
% TEMPERATURES
if plot_temps,
figure('units', 'normalized','outerposition',...
[0 0 1 1], 'defaultaxesfontsize', 18)
plot(out.TIME, out.CABIN_TEMPERATURE); hold on; grid on;
plot(out.TIME, out.Tref,'--k');
plot(out.TIME, out.OUTSIDE_TEMPERATURE);
plot(out.TIME, out.SURFACE_TEMPERATURE);

legend('Cabin Temperature', 'Reference Temperature', ...
'Outside Temperature', 'Surface Temperature');
xlabel('Time [s]'); ylabel('Temperature [ C ]');
xlim([out.TIME(1) out.TIME(end)])
ylim([min(min(out.CABIN_TEMPERATURE(:),0)),...
max(max(out.CABIN_TEMPERATURE(:),60))])
time = get(gca,'xtick');
[~,idx] = min(abs(time-out.TIME));
set(gca, 'xticklabel', ...
datestr(InitialTime+seconds(out.TIME(idx)),'HH:MM'))
set(gca, 'fontsize', fs)
xtickangle(45);

set(gca, 'ytick',       ...
min(min(out.CABIN_TEMPERATURE(:),0)):5:...
```

```
max(max(out.CABIN_TEMPERATURE(:),60)));

title('Temperatures over time');
end
% TEMPERATURE && HEAT
if plot_both,
figure('units', 'normalized','outerposition', ...
[0 0 1 1], 'defaultaxesfontsize', 18)
yyaxis left
plot(out.TIME, out.METABOLIC_LOAD, '--b', 'linewidth', 2);
set(gca, 'xtick',       time);
set(gca, 'xticklabel',  time_str);
set(gca, 'fontsize', fs)
xtickangle(45);
xlabel('Time');
ylabel('Total thermal load inside the cabin [W]');
set(gca,'ycolor','b')
xlim([out.TIME(1) out.TIME(end)])

yyaxis right
plot(out.TIME, Mcabin, '--r', 'linewidth', 2);
ylabel('Cabin temperature [ C ]')
set(gca, 'ytick',       ...
min(min(out.CABIN_TEMPERATURE(:),0)):5:...
max(max(out.CABIN_TEMPERATURE(:),60)));
set(gca,'ycolor','r')
set(gca, 'fontsize', fs)
grid on
end

% AC Load
if plot_ACload,
figure('units', 'normalized','outerposition', ...
[0 0 1 1], 'defaultaxesfontsize', 18)
plot(out.TIME, out.AC_LOAD);
grid on;
time = get(gca,'xtick');
[~,idx] = min(abs(time-out.TIME));
set(gca, 'xticklabel', ...
datestr(InitialTime+seconds(out.TIME(idx)),'HH:MM'))
set(gca, 'fontsize', fs)
xtickangle(45);
xlabel('Time');
ylabel('Thermal Load [W]');
xlim([out.TIME(1) out.TIME(end)])
title('AC Load')
end

% PMV && PPD estimate
```

```
if plot_PMV,
figure('units', 'normalized','outerposition', ...
[0 0 1 1], 'defaultaxesfontsize', 18)
subplot(2,1,1)
plot(out.TIME, out.PMV, out.TIME, out.PMV);
grid on;
set(gca, 'xtick',       time);
set(gca, 'xticklabel',  []);
set(gca, 'ytick',       [-4:1:-2, -1:0.5:1, 2:1:4]);
set(gca, 'fontsize', fs)
xtickangle(45);
ylabel('Predicted Mean Vote');
xlim([out.TIME(1) out.TIME(end)])
ylim([-4 4])
title('PMV estimate')

subplot(2,1,2)
plot(out.TIME, out.PPD);
grid on;
time = get(gca,'xtick');
[~,idx] = min(abs(time-out.TIME));
set(gca, 'xtick', datestr(InitialTime+seconds(out.TIME(idx))))
set(gca, 'fontsize', fs)
xtickangle(45);
set(gca, 'ytick',       [0:5:20, 30:10:100]);
set(gca, 'fontsize', fs)
xlabel('Time');
ylabel('Predicted Percentage of Dissatisfied [%]');
xlim([out.TIME(1) out.TIME(end)])
ylim([0 100])
title('PPD estimate')
end

% HVAC Power Concumption

Phvac = out.Pcompressor + out.Pblower + out.Pheater.*(out.Pheater<500);

P_kwh = trapz(out.TIME(~isinf(Phvac)),Phvac(~isinf(Phvac)))/3600/1000;
disp(['Energy consumed by the HVAC: ', num2str(P_kwh,3),' kWh'])
if plot_consumption,
figure('units', 'normalized', 'position', [0 0 1 1])
plot(out.TIME, Phvac/1e3, out.TIME, out.Pblower/1e3, out.TIME,...
out.Pcompressor/1e3, out.TIME, out.Pheater.*(out.Pheater<500)/1e3);
grid on;
legend('Total HVAC consumption', 'Blower consumption',...
'Compressor consumption', 'Heater consumption')
time = get(gca,'xtick');
[~,idx] = min(abs(time-out.TIME));
set(gca, 'xticklabel', datestr(InitialTime+seconds(out.TIME(idx)),...
```

```
'HH:MM'))
set(gca, 'fontsize', fs)
xtickangle(45);
xlabel('Time');
ylabel('Power Consumption [kW]');
xlim([out.TIME(1) out.TIME(end)])
title(['HVAC Power Consumption (', num2str(P_kwh,4),' kWh)'])

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('END OF SCRIPT')
```

B.2 Simplified system script

Listing B.2: Simplified model Matlab script

```
%%% HVAC MODEL SIMPLIFICATION SCRIPT
clear all; close all; clc;
file    =   'HVAC_LinearController.slx';

plot_temps  = 0;
plot_bode   = 0;
plot_power  = 1;

Tend    =   3 * 60 * 60;    % [s]
ts      =   1;

%% Loads setting
% Metabolic Load
Np      =   1;      % Passengers other than driver
Mp      =   94.6;   % [W/m2]
Md      =   144.5;  % [W/m2]
Adu     =   1.72;   % [m2]

% Radiant Load
% Time data is referred to 21st of each month in 1964
Months      =   {'Jan', 'Feb', 'Mar', 'Apr', 'May', 'June', ..
 'July', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'}; % Month
month       =   'Feb';  %%%% SELECT THE MONTH
n_month     =   find(contains(Months, month));

hour        =   21;

InitialTime =   datetime(1964, n_month, 21, hour , 0, 0);
LST0        =   InitialTime.Hour+InitialTime.Minute/60+ ...
InitialTime.Second/3600;

% Location Data: BARCELONA
Latitude    =   41.3879;
Longitude   =   2.16992;
UTC         =   1;
% Spain is UTC+2h from April to October (correction needed)
if and(n_month >= 4, n_month <= 10), UTC = 0; end
LSTM        =   15*UTC;

EoT_tab     =   [-11.2, -13.9, -7.5, 1.1, 3.3, -1.4, -6.2, -2.4,...
7.5, 15.4, 13.8, 1.6];          % Equation of Time [min]
Declination_tab=   [-20, -10.8, 0, 11.6, 20, 23.45, 20.6, 12.3, ...
0, -10.5, -19.8, -23.45];       % Declination [  ]
A_tab       =   [1230, 1215, 1186, 1136, 1104, 1088, 1085, 1107, ...
1151, 1192, 1221, 1233];        % A [W/m2]
B_tab       =   [0.142, 0.144, 0.156, 0.18, 0.196, 0.205, 0.207,...
```

```
0.201, 0.177, 0.16, 0.149, 0.142];     % B [/]
C_tab       =   [0.058, 0.06, 0.071, 0.097, 0.121, 0.134, 0.136, ...
0.122, 0.092, 0.073, 0.063, 0.057];    % C [/]

AST         =   LST0 + EoT_tab(n_month)/60 + (LSTM-Longitude)/15;
H           =   15*(AST-12);
sinBeta_tab =   cosd(Latitude).*cosd(Declination_tab).*cosd(H)' +...
sind(Latitude).*sind(Declination_tab);
Beta        =   asin(sinBeta_tab);

Tout_min_tab=   [9, 8, 10, 13, 16, 20, 23, 23, 20, 17, ...
12, 9]; % Barcelona data
Tout_max_tab=   [15, 15, 17, 20, 23, 27, 29, 29, 26, 23, ...
18, 15];     % Barcelona data
Rad_max_tab =   [ 615.6311,  645.3447,  655.4160,  656.1268, ...
659.6520,  660.2711,  658.3445,  653.7843,  645.8083, ...
635.2982,  607.2597,  593.5552];% Barcelona data (from simulation)

Tout_min    =   Tout_min_tab(n_month);
Tout_max    =   Tout_max_tab(n_month);
Rad_max     =   Rad_max_tab(n_month);

A           =   A_tab(n_month);
B           =   B_tab(n_month);
C           =   C_tab(n_month);
sinB        =   sinBeta_tab(n_month);
tilt        =   Beta(n_month);             %% Sun declination angle [  ]
% Load GMD geometric data
load('.\..\GeometricProfileData\GeometricData.mat')

dir_abs     = sum(Surface.*Absorptivity.* ...
abs(cos((Tilt+tilt)/180*pi)));
dir_trans   = sum(Surface.*Transmissivity.* ...
abs(cos((Tilt+tilt)/180*pi)));

dif_abs     = Surface.*Absorptivity;  %  dif_abs = dif_abs(:)';
dif_trans   = Surface.*Transmissivity; % dif_trans = dif_trans(:)';

ref_abs     = dif_abs;
ref_trans   = dif_trans;

% HVAC Load

RecirculationFlap = 1;
% DUMMY
dummyAC     =   0;
tc          =   10 * 60;


```

```
% HEATER
Sheater     = 0.3142*0.184;     % [m2]

Iheater     = 32.5;
Vheater     = 48;

R01         = 1.457;
m           = 0.0064;
R02         = 0.3678*1e-3;
b           = 0.0494;
Rmax        = 31.399;
Tmin        = -50;
Tcr         = 120;
Tmax        = 250;

SHI         = 0.1;              % [W/ C ]

D_heater    = 92e-3;
rho_air     = 1.14;
mu_air      = 1.895e-5;
k_air       = 25.72e-3;
Pr_air      = 0.69;

tau_ptc     = 5;


%% PMV Estimate
MoverA      = 58;           % [W/m2] Light Traffic
h           = 0;
pa          = 2.5e3;        % [Pa] @20 C
pa          = pa / 133;     % [mmHg]

ncl         = 4;            % Slacks, Short
fcl         = [1,   1,    1.05, 1.1,  1.1, 1.15, 1.15];     % [-]
Icl         = [0,   0.1,  0.35, 0.5,  0.6, 1,    1.5];      % [clo]
as          = [0.5, 0.55, 0.6,  0.65, 0.7, 0.75, 0.75];     % [-]

fcl = fcl(ncl); Icl = Icl(ncl); as = as(ncl);

%%%%%%%%%%%%%% Tcl estimate %%%%%%%%%%%%%

N           = 100; % Number of iterations
Tcl_ini     = 37;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Transfer Functions
%{

 Tc(s) = Ghvac * Qhvac + Gmet * Qmet + Grad * Qrad + Gtout * Tout
```

```
 All TF are vehicle speed dependent
 %}
maCa                =   1953;
mbCb                =   220956;

krad                =   4.5;

Vveh                =   50;             % [km/h]
kamb                =   exp(4.3986)*(Vveh/3.6)^0.2273;

% CABIN MODEL
Ghvac_numerator     =   1/maCa * [1, (2*kamb+428)/mbCb];
Ghvac_denominator   =   [1, ((2*kamb+428)/mbCb+(kamb+438)/maCa),...
(kamb^2+448*kamb+4280)/maCa/mbCb];
Ghvac               =   tf(Ghvac_numerator, Ghvac_denominator)

Gmet_numerator      =   Ghvac_numerator;
Gmet_denominator    =   Ghvac_denominator;
Gmet                =   tf(Gmet_numerator, Gmet_denominator)

Grad_numerator      =   1/maCa * [1, (kamb*(2+krad)+428*(1+krad))/mbCb];
Grad_denominator    =   Ghvac_denominator;
Grad                =   tf(Grad_numerator, Grad_denominator)

Gtout_numerator     =   10/maCa * [1, (0.1*kamb^2+44.8*kamb+428)/mbCb];
Gtout_denominator   =   Ghvac_denominator;
Gtout               =   tf(Gtout_numerator, Gtout_denominator)

%BODY MODEL
Grad_body_numerator =   [krad];
Grad_body_denominator=  [mbCb, 2*kamb+428];
Grad_body           =   tf(Grad_body_numerator, Grad_body_denominator)

Gtout_body_numerator=   [kamb];
Gtout_body_denominator= Grad_body_denominator;
Gtout_body          =   tf(Gtout_body_numerator, Gtout_body_denominator)

Gbody_numerator     =   [kamb+428];
Gbody_denominator   =   Grad_body_denominator;
Gbody               =   tf(Gbody_numerator, Gbody_denominator)



% IC
dIdir = 0;
if sinB >= 0,
dIdir = A/exp(B/sinB);
end
```

```
dQdir = dir_trans * dIdir;
dIdif = C.*dIdir.*(1+cosd(Tilt))./2;
dQdif = sum(dif_trans .* dIdif);
dIref = (dIdir+dIdif) * 0.2 .*(1-cosd(Tilt))/2;
dQref = sum(ref_trans.* dIref);

Qhvac_ini           = -818.039;
hvac_ini            =  Qhvac_ini            * Ghvac_numerator(end)/...
Ghvac_denominator(end);
Qmet_ini            = (Np * Mp + Mp)*Adu;
met_ini             = Qmet_ini              * Gmet_numerator(end) /...
Gmet_denominator(end);
Qrad_ini            = (dQdir+dQdif+dQref);
rad_ini             = Qrad_ini              * Grad_numerator(end) / ...
Grad_denominator(end);
Tout_ini            = (Tout_min+(Tout_max-Tout_min)/Rad_max*Qrad_ini);
tout_ini            = Tout_ini              * Gtout_numerator(end)/...
Gtout_denominator(end);
Tcabin_ini          = hvac_ini + met_ini + rad_ini + tout_ini

Qrad_body_ini       = Qrad_ini;
rad_body_ini        = Qrad_body_ini         * Grad_body_numerator(end) /...
Grad_body_denominator(end);
Tout_body_ini       = Tout_ini;
tout_body_ini       = Tout_body_ini         * Gtout_body_numerator(end)/...
Gtout_body_denominator(end);
body_ini            = Tcabin_ini            * Gbody_numerator(end)     /...
Gbody_denominator(end);
Tbody_ini           = rad_body_ini + tout_body_ini + body_ini

%% CONTROL
Tcomfort            =   21;
Thyst               =   1;

PMVconfort          =   0;
PMVhyst             =   0.5;

MODE                = { Automatic Climate Control,
 Heating Control,
 Cooling Control   };
mode                = 3;
CONTROL_VARIABLE    = {    Temperature-based,
 PMV-based };
control_variable    = 1 ;


%% Simulation
out = sim(file);

```

```
%% PLOTS
fs          = 18;

total_min   = floor(Tend/60);
min_tick    = 10;
n_delta     = floor(max(max(60*min_tick/ts), length(out.time)/20));
n_ticks     = 1:n_delta:length(out.time);

% LOADS
time        = out.time;
time        = time(n_ticks);
time_str    = datestr(InitialTime+seconds(out.time),'HH:MM');
time_str    = time_str(n_ticks, :);

k_temp = (5*60)/ts;
k_heat = (5*60)/ts;

Mcabin  = movmean(out.Tcabin,k_temp);
Mout    = movmean(out.Tout,k_temp);
Msurf   = movmean(out.Tbody, k_temp);

% TEMPERATURES
if plot_temps,
figure('units', 'normalized','outerposition', [0 0 1 1], ...
'defaultaxesfontsize', 18)
plot(out.time, out.Tcabin,                          ...
out.time, Mcabin, '--k'); hold on; grid on;
plot(out.time, out.Tout,                          ...
out.time, Mout,'--k');
plot(out.time, out.Tbody,                          ...
out.time, Msurf, '--k');

legend('Cabin Temperature', 'Cabin Temperature (mean)', ...
'Outside Temperature', 'Outside Temperature (mean)', ...
'Surface Temperature', 'Surface Temperature (mean)');
xlabel('Time [s]'); ylabel('Temperature [ C ]');
xlim([out.time(1) out.time(end)])
ylim([min(min(out.Tcabin(:),0)), max(max(out.Tcabin(:),60))])

tcks    = get(gca,'xtick');
[~,idx]  = min(abs(out.time-tcks));
set(gca, 'xticklabel',  ...
datestr(InitialTime+seconds(out.time(idx)), 'HH:MM'));
set(gca, 'fontsize', fs)

xtickangle(45);

set(gca, 'ytick',       ...
min(min(out.Tcabin(:),0)):5:max(max(out.Tcabin(:),60)));
```

```
292
293 title('Temperatures over time');
294 end
295
296 if plot_power,
297 Phvac = out.Pblower + out.Pcompressor + out.Pheater.*(out.Pheater<500);
298 figure('units', 'normalized','outerposition',...
299 [0 0 1 1], 'defaultaxesfontsize', 18)
300 plot(out.time, out.Pblower/1e3, out.time, ...
301 out.Pcompressor/1e3, out.time, out.Pheater.*(out.Pheater<500)/1e3,...
302 out.time, (Phvac)/1e3)
303 legend({'$P_{blower}$ [kW]', '$P_{compressor}$ [kW]',...
304  '$P_{PTCs}$ [kW]','$P_{HVAC}$ [kW]'}, 'interpreter','latex')
305
306 P_kwh = trapz(out.time(~isinf(Phvac)),Phvac(~isinf(Phvac)))/3600/1000;
307 disp(['Energy consumed by the HVAC: ', num2str(P_kwh,3),' kWh'])
308 title(['HVAC Power Consumption (', num2str(P_kwh,4),' kWh)'])
309
310 xtickangle(45);
311 xlabel('Time');
312 ylabel('Power Consumption [kW]');
313 xlim([out.time(1) out.time(end)])
314
315 tcks    = get(gca,'xtick');
316 [~,idx]  = min(abs(out.time-tcks));
317 set(gca, 'xticklabel',  ...
318 datestr(InitialTime+seconds(out.time(idx)), 'HH:MM'));
319 set(gca, 'fontsize', fs)
320
321 xtickangle(45);
322
323 grid on;
324
325 end
326
327 % BODE
328 if plot_bode,
329 fs = 18;
330 figure('units', 'normalized','outerposition', [0 0 1 1], ...
331 'defaultaxesfontsize', fs)
332 [hvacGM, hvacPM, ~, hvacCrossover] = margin(Ghvac)
333 margin(Ghvac)
334
335 % Set data line width and color
336 set(findall(gcf,'Type','line'),'LineWidth',2,'Color','blue')
337 % Set axes tick label font size, color, and line width
338 set(findall(gcf,'Type','axes'),'FontSize',fs)
339 grid(findall(gcf,'Type','axes'), 'on')
340
```

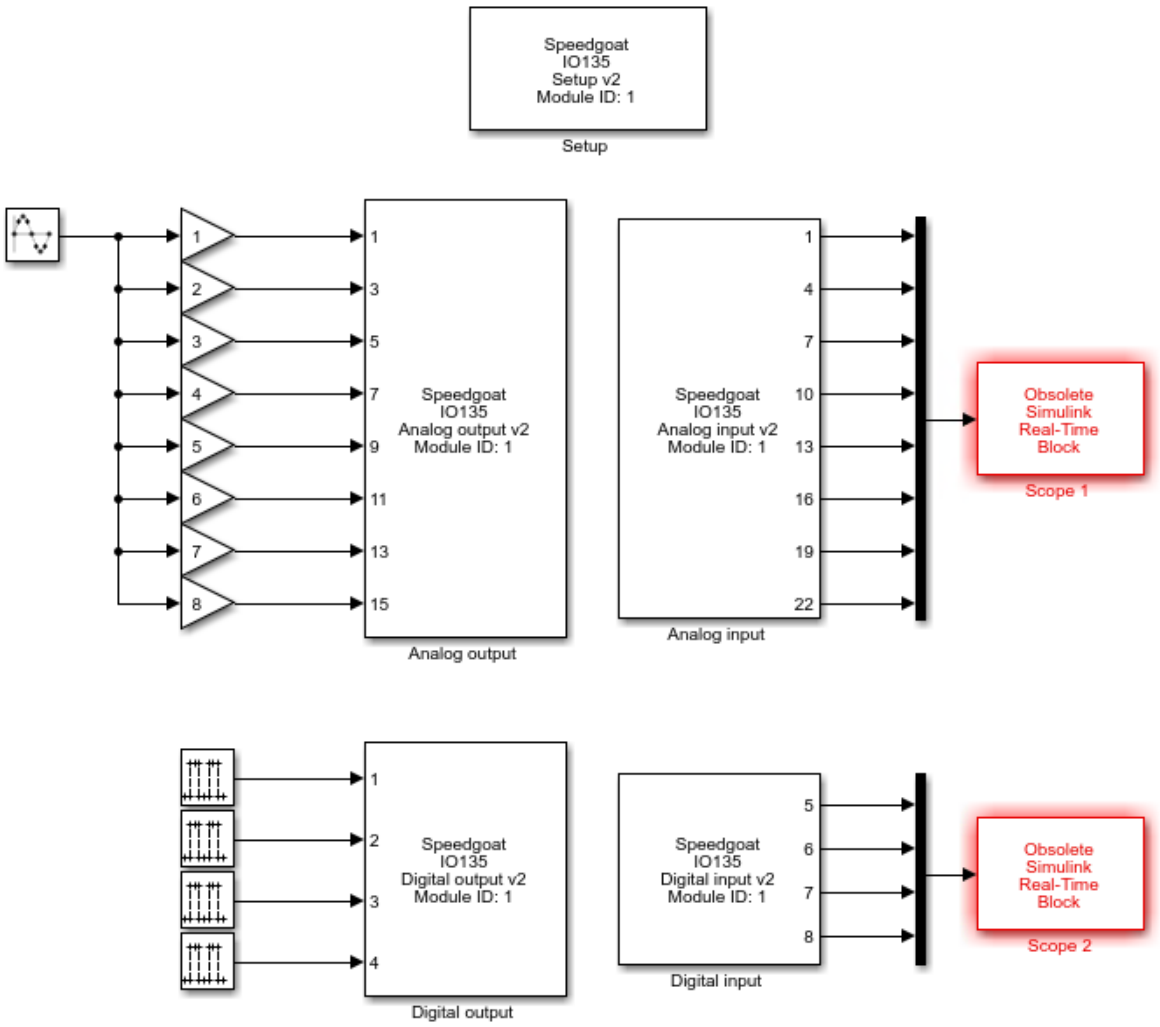
```
341 setoptions(gcr, 'FreqUnits','Hz')
342
343 % Set title, axes labels, and legend font size
344 titles  = findall(gcf, '-property', 'Title');
345 tl      = get(titles(1).Title, 'String')
346 tl{1}   = '$\frac{T_{cabin}(s)}{\dot{Q}_{HVAC}(s)}$ Bode Diagram';
347 set(titles(1).Title,'String',tl, 'Interpreter', 'latex')
348 set(findall(gcf,'-property','FontSize'),'FontSize',fs)
349
350 hvacPoles  = pole(Ghvac)
351 hvacZeros  = zero(Ghvac)
352
353
354 end
```

Appendix C

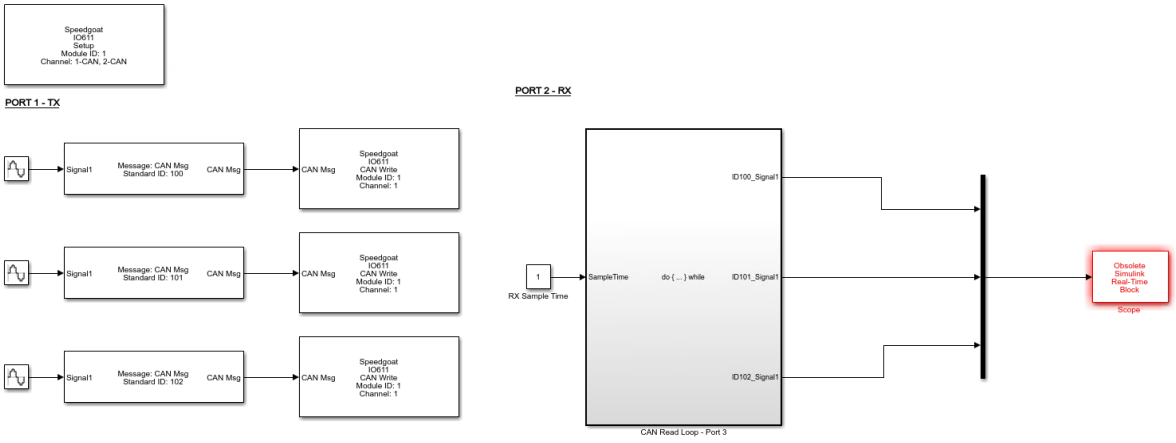
HIL interface

C.1 SW interface

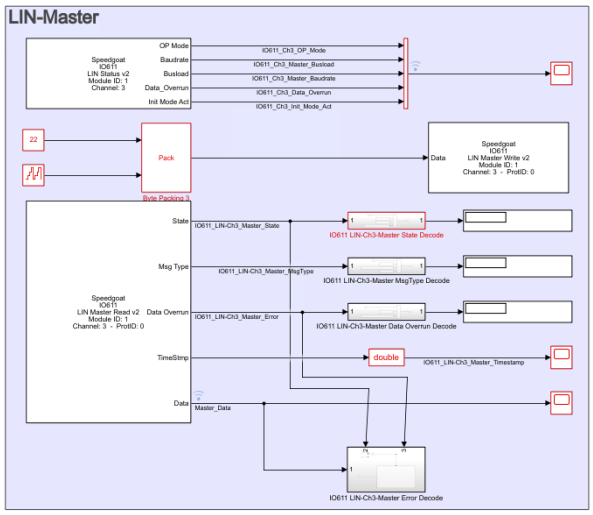
Analog and digital software testbench in a loop-back configuration:



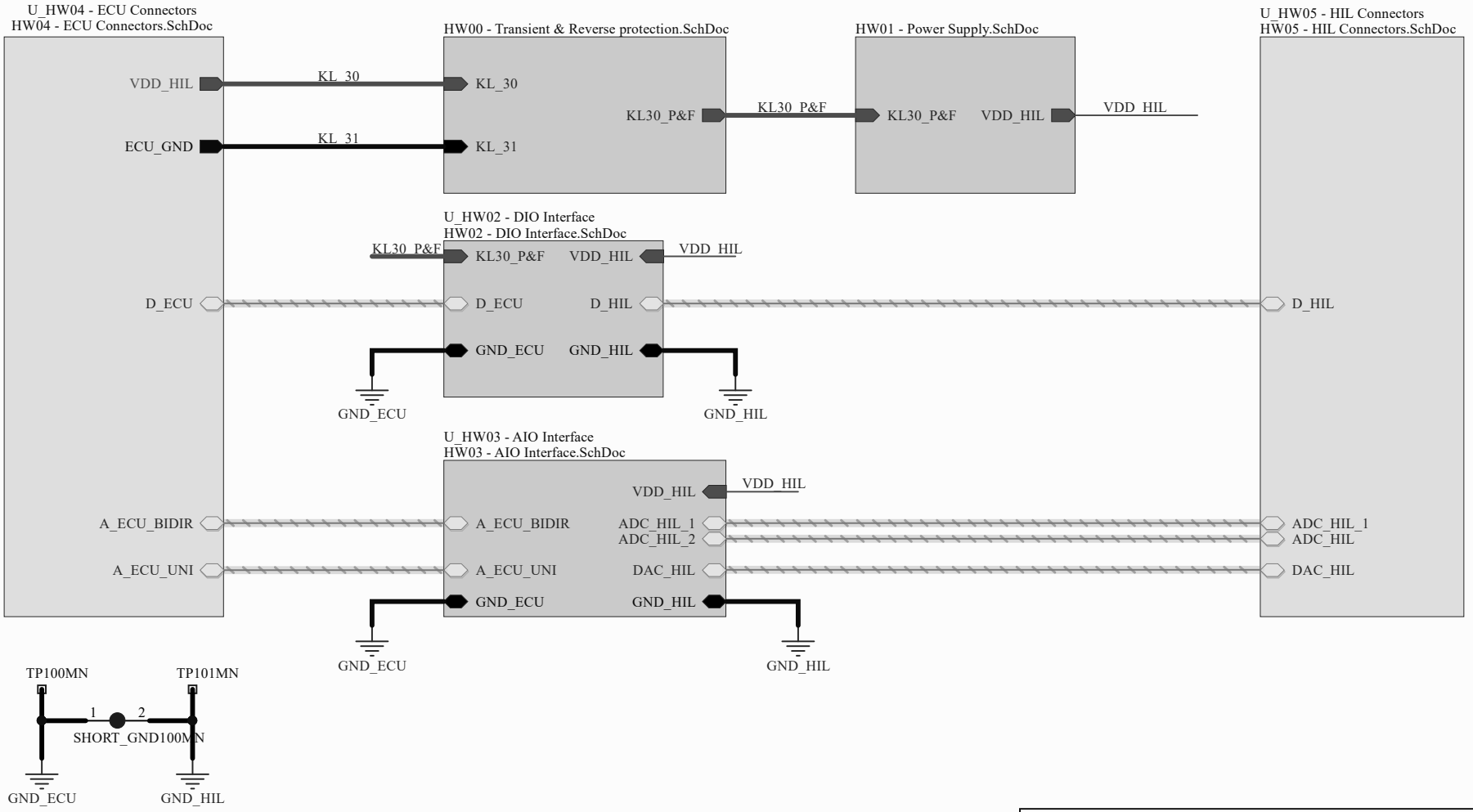
CAN interface:



LIN interface:

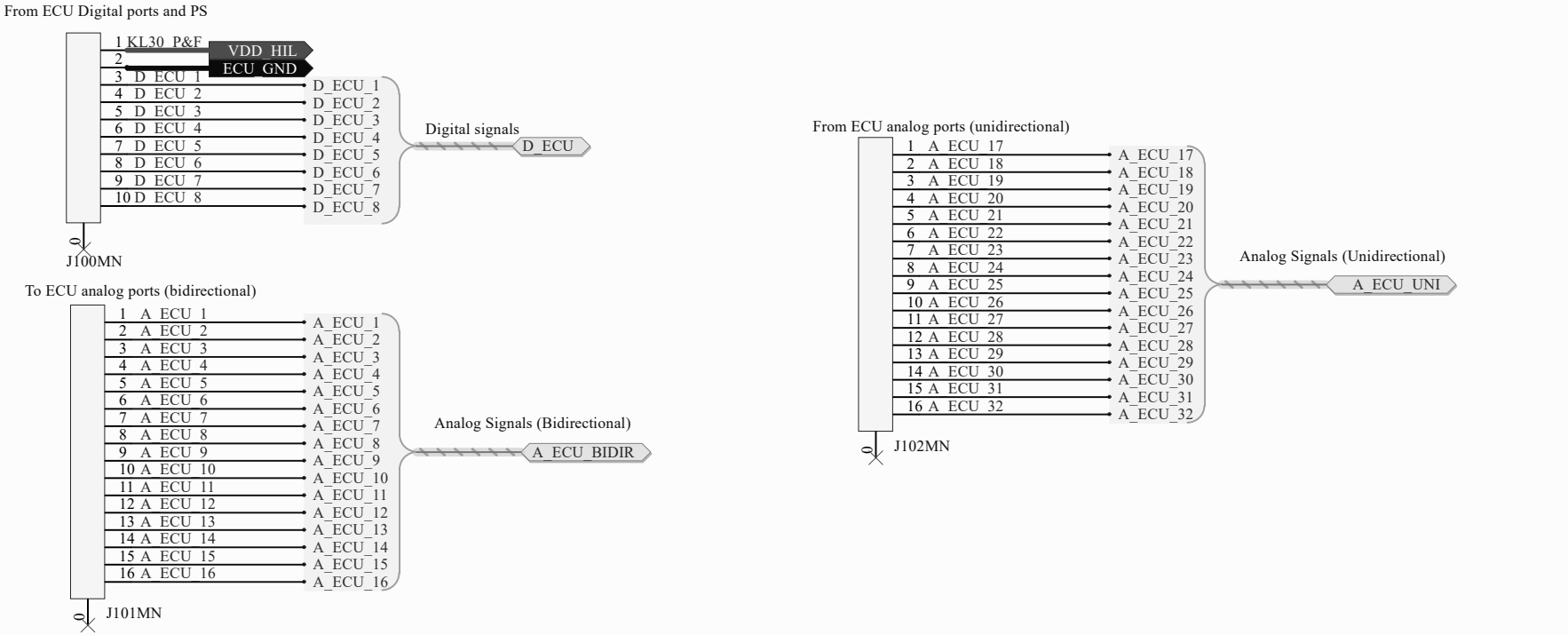


C.2 HW interface



Vehicle HVAC system modeling and controlling					
idneo			Name	Date	
		Drawn	S. Esteban	30/07/2021	
		Checked	M. Marquez	28/07/2021	
		Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	MAIN	Level	100
Doc. Name	HIL Interface.SchDoc			Sheet	1 / 10

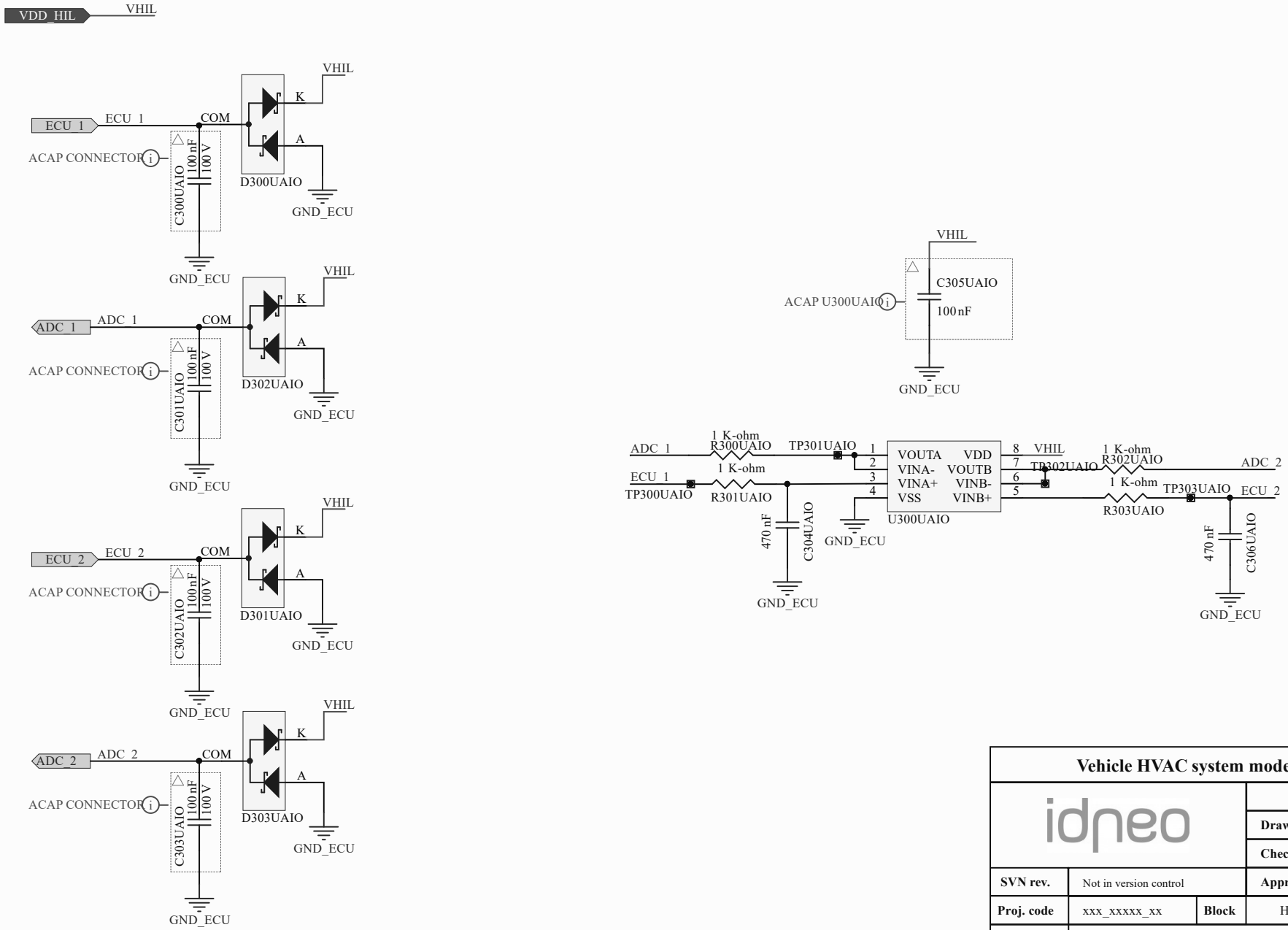
A4



Vehicle HVAC system modeling and controlling

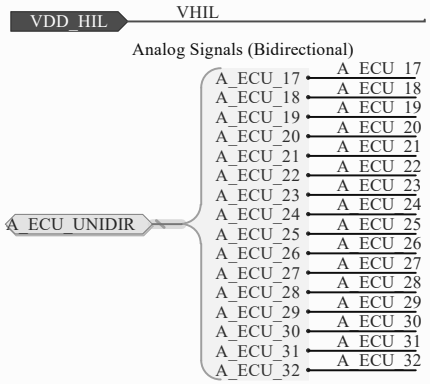
			Name	Date
idneo		Drawn	S. Esteban	30/07/2021
		Checked	M. Marquez	28/07/2021
SVN rev.	Not in version control	Approved	name	dd/mm/aaaa
Proj. code	xxx_xxxxx_xx	Block	HW00	Level 100
Doc. Name	HW04 - ECU Connectors.SchDoc			Sheet 2 10

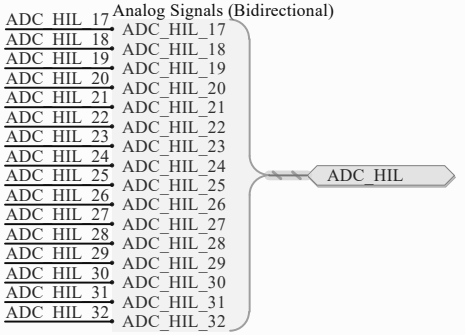
A4

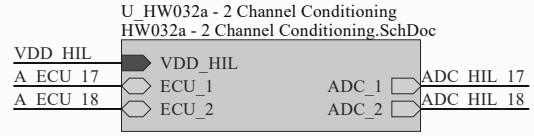


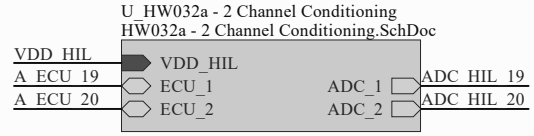
Vehicle HVAC system modeling and controlling					
idneo			Name	Date	
		Drawn	S. Esteban	30/07/2021	
		Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control	Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW032a	Level	100
Doc. Name	HW032a - 2 Channel Conditioning.SchDoc			Sheet	2 / 10

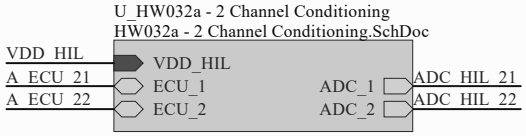
A4

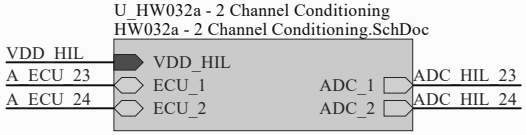


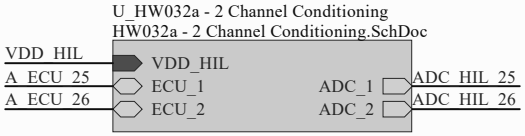


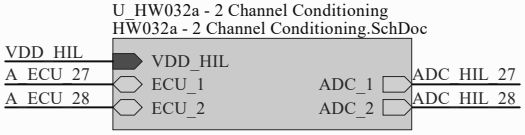


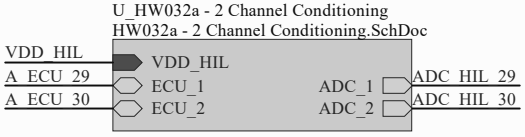


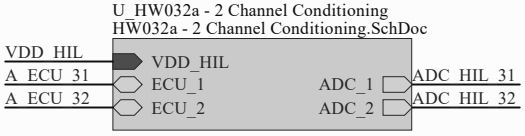








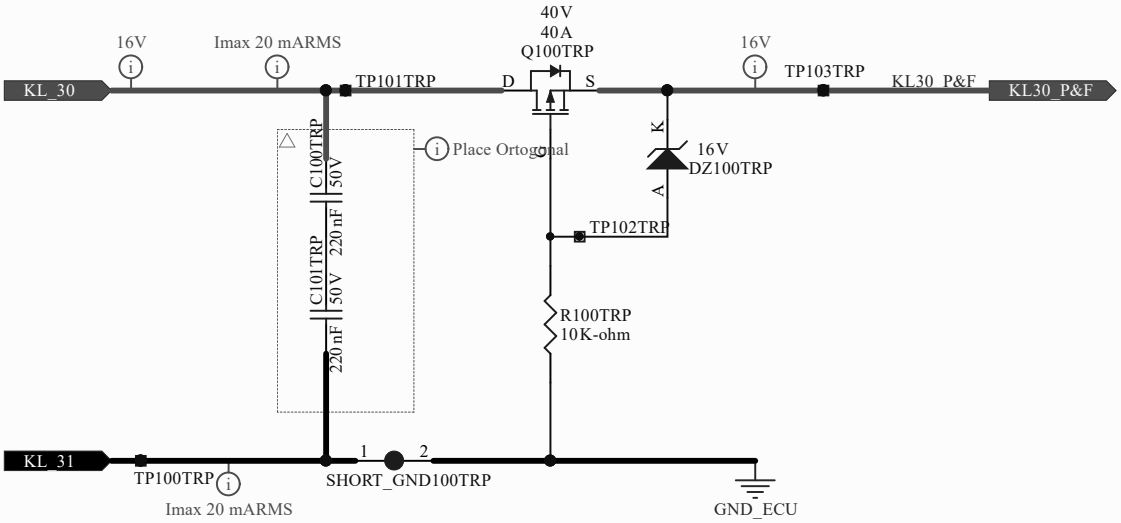




Vehicle HVAC system modeling and controlling

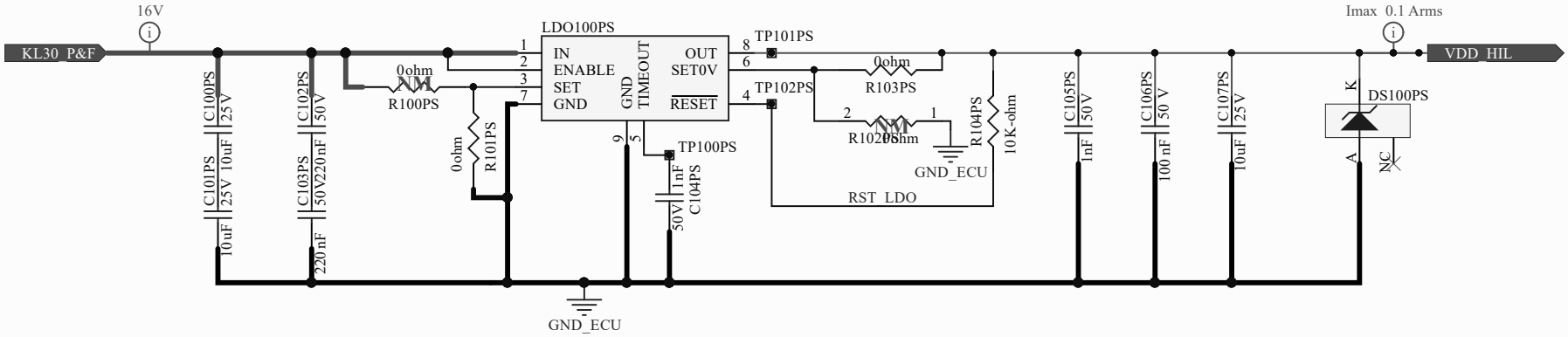
			Name	Date
idneo		Drawn	S. Esteban	30/07/2021
		Checked	M. Marquez	28/07/2021
SVN rev.	Not in version control	Approved	name	dd/mm/aaaa
Proj. code	xxx_xxxxx_xx	Block	HW032	Level 100
Doc. Name	HW032 - ADC Unidierectional Interface.SchDoc		Sheet	2 10

A4



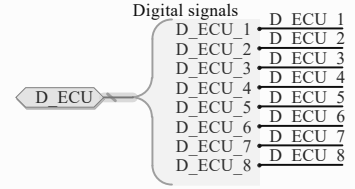
Vehicle HVAC system modeling and controlling							
idneo					Name	Date	
				Drawn	S. Esteban	30/07/2021	
				Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control			Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW00		Level	100	
Doc. Name	HW00 - Transient & Reverse protection.SchDoc				Sheet	2	10

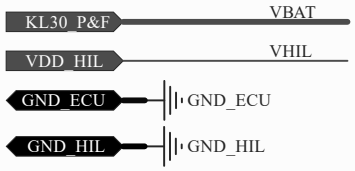
A4

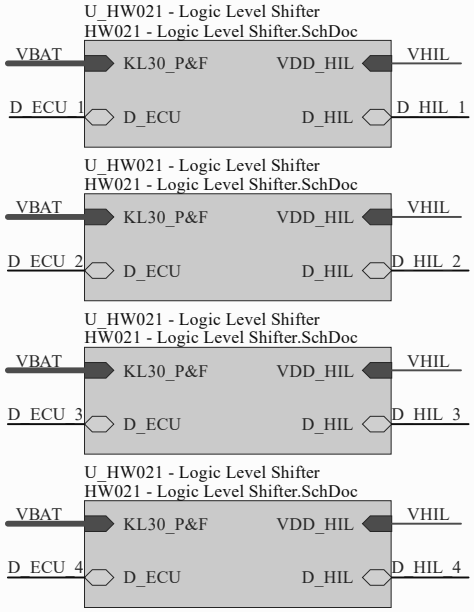


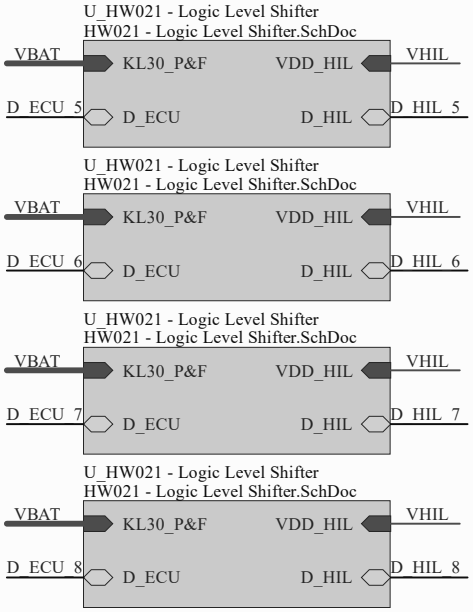
Vehicle HVAC system modeling and controlling						
idneo				Name	Date	
			Drawn	S. Esteban	30/07/2021	
			Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control		Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW01	Level	100	
Doc. Name	HW01 - Power Supply.SchDoc			Sheet	3	10

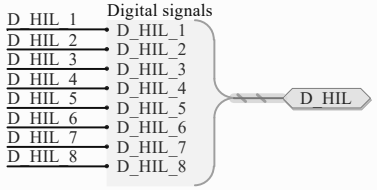
A4





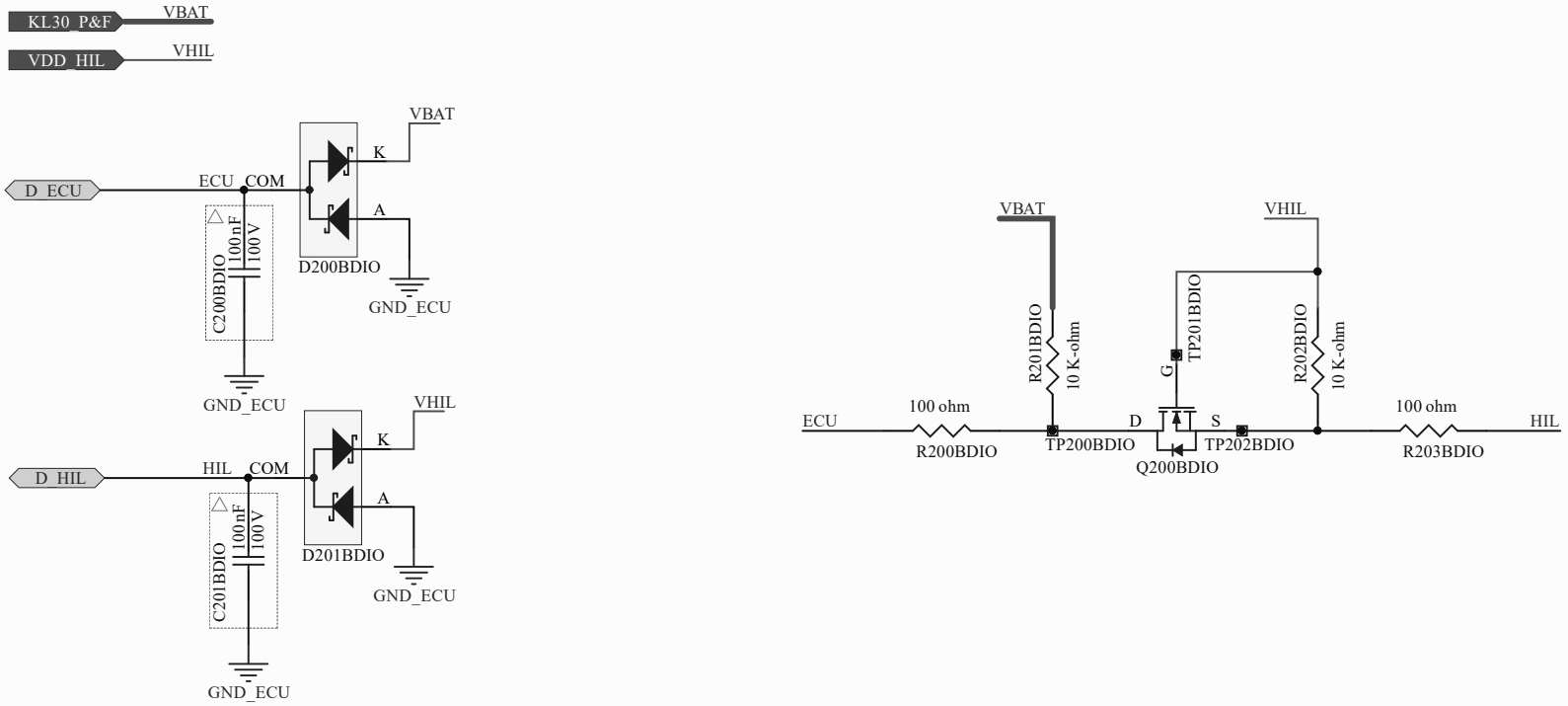






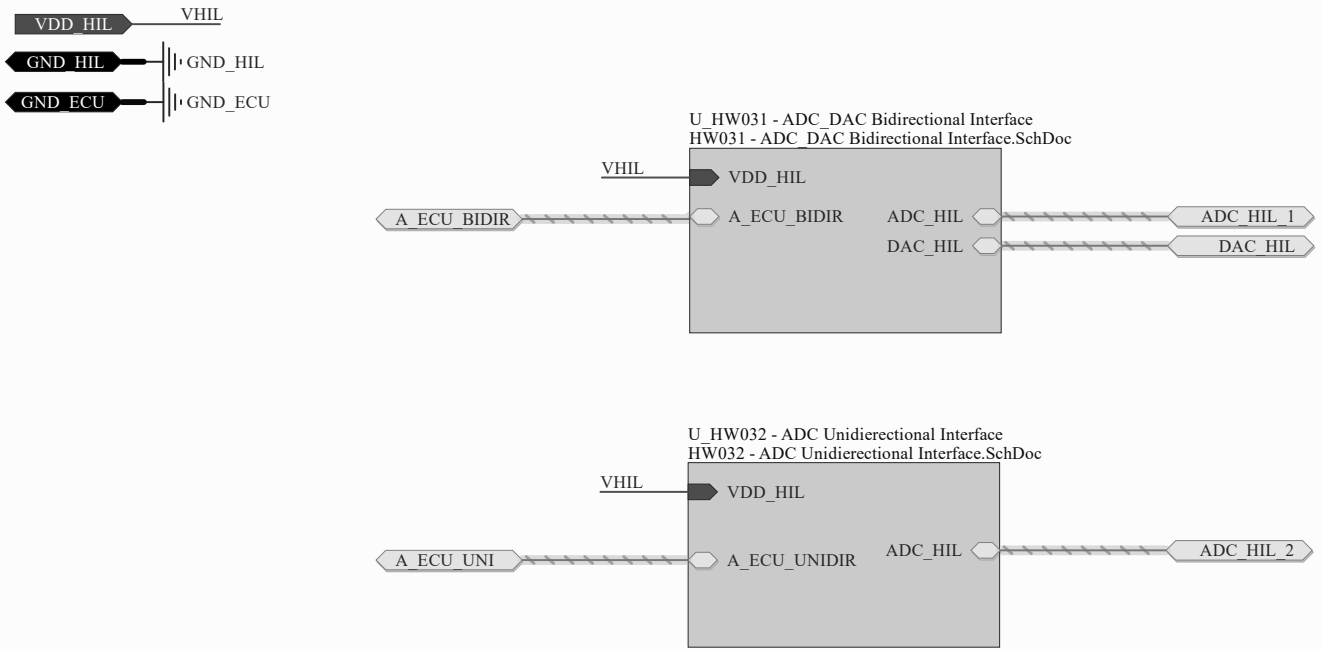
Vehicle HVAC system modeling and controlling						
idneo				Name	Date	
			Drawn	S. Esteban	30/07/2021	
			Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control		Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW02	Level	100	
Doc. Name	HW02 - DIO Interface.SchDoc			Sheet	4	10

A4



Vehicle HVAC system modeling and controlling						
idneo				Name	Date	
			Drawn	S. Esteban	30/07/2021	
			Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control		Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW021	Level	100	
Doc. Name	HW021 - Logic Level Shifter.SchDoc			Sheet	5	10

A4



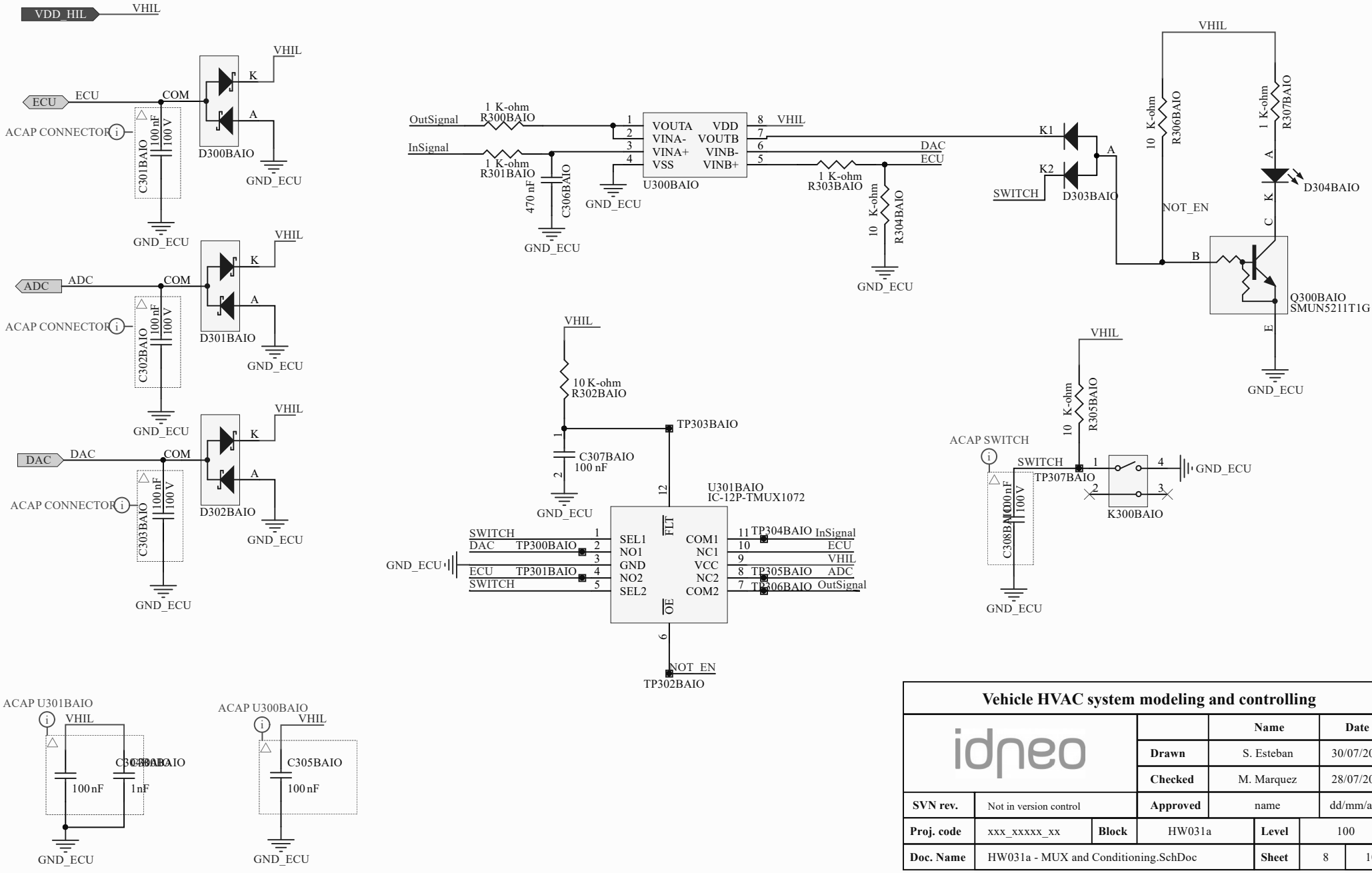
Vehicle HVAC system modeling and controlling						
idneo				**Name**	**Date**	
			Drawn	S. Esteban	30/07/2021	
			Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control		**Approved**	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	**Block**	HW03	**Level**	100	
Doc. Name	HW03 - AIO Interface.SchDoc			**Sheet**	6	10

A4

Vehicle HVAC system modeling and controlling

		Name	Date
idneo	Drawn	S. Esteban	30/07/2021
	Checked	M. Marquez	28/07/2021
SVN rev. Not in version control	Approved	name	dd/mm/aaaa
Proj. code xxx_xxxxx_xx	Block HW031	Level 100	
Doc. Name HW031 - ADC_DAC Bidirectional Interface.SchDoc		Sheet 7	10

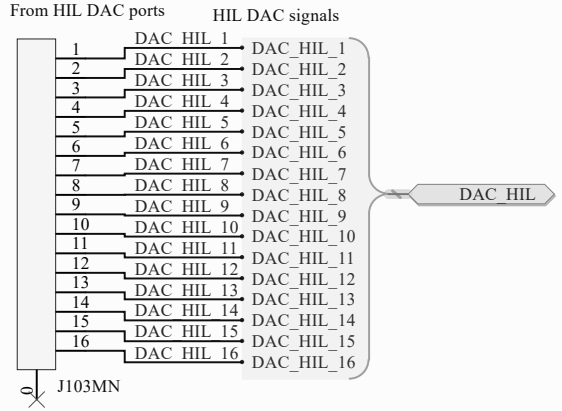
A4

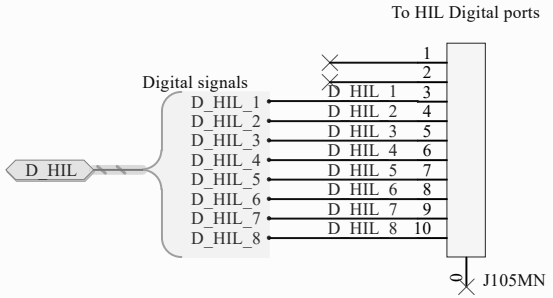


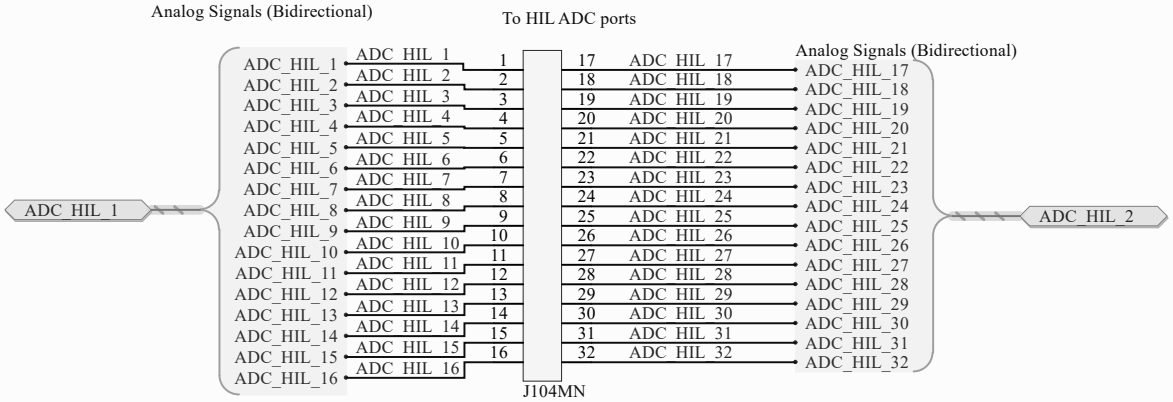
Vehicle HVAC system modeling and controlling

			Name	Date
idneo		Drawn	S. Esteban	30/07/2021
		Checked	M. Marquez	28/07/2021
SVN rev.	Not in version control	Approved	name	dd/mm/aaaa
Proj. code	xxx_xxxxx_xx	Block	HW031a	Level 100
Doc. Name	HW031a - MUX and Conditioning.SchDoc			Sheet 8 10

A4







Vehicle HVAC system modeling and controlling					
idneo			Name	Date	
		Drawn	S. Esteban	30/07/2021	
		Checked	M. Marquez	28/07/2021	
SVN rev.	Not in version control	Approved	name	dd/mm/aaaa	
Proj. code	xxx_xxxxx_xx	Block	HW00	Level	100
Doc. Name	HW05 - HIL Connectors.SchDoc			Sheet	12 / 10

A4

BILL OF MATERIALS

idneo

Project: IDN_TRMHW_N7_HILInterface.PrjPCB
Created by: S. Esteban

Level: 100

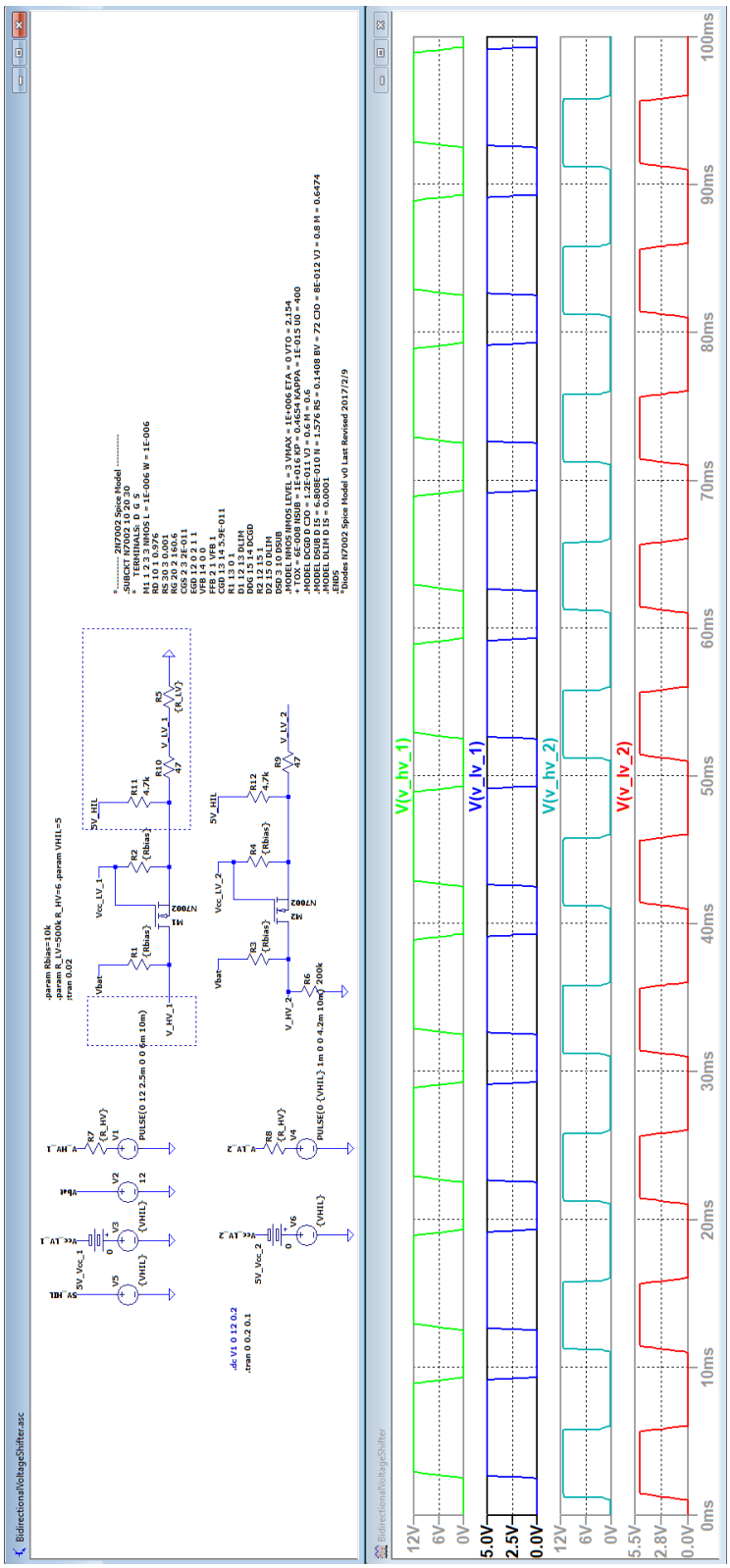
Total SCH sheets: 12
Variant: 5V_LV
Review 01

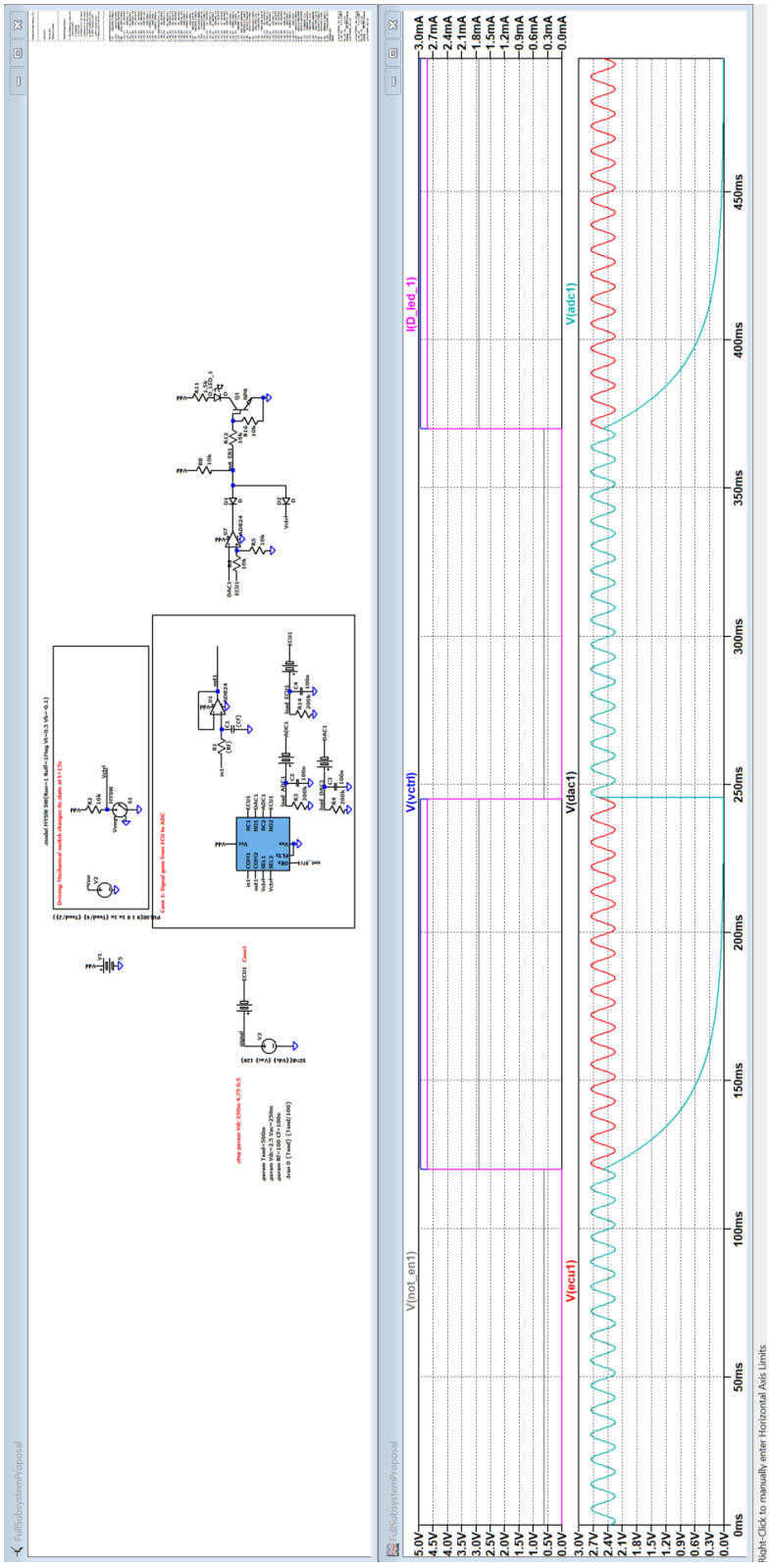
PCB INFO RFI

Layers:	x
Material:	<Parameter PCB_Material not found>
Class:	xxx
TG:	xxx
Size Single:	xxxx x xxxx
Thickness:	x,x ±10%

Report Date	20/10/2021	20:07:22
Print Date	20-Oct-21	8:12:27 PM

#	Quantity	Designator	FAMILY_BY_PE	SUBFAMILY	MANUFACTURER_NAME	ORDER_CODE	MOUNT_TYPE	SOLDER_METHOD	Footprint	VALUE	RATING	AECQ	TOLERANCE	Name_Current
1	3	C100PS, C101PS, C107PS	CAP	Chip ceramic capacitor	MURATA	GRT21BC71E106KE13L	SMD	Reflow	CAPC2012X145N-	10 uF	25 V	AEC-Q200	10%	HW01
2	2	C102PS, C103PS	CAP	Chip ceramic capacitor	KEMET	C0805C224K5NACAUTO	SMD	Reflow	CAPC2012X100N-	220 nF	50 V	AEC-Q200	10%	HW01
3	18	C104PS, C105PS,	CAP	Chip ceramic capacitor	MURATA	GCM155R71H102KA37D	SMD	Reflow	CAPC1005X55N-	1 nF	50 V	AEC-Q200	10%	HW01, HW01,
4	41	C106PS, C304BAIOA,	CAP	Chip ceramic capacitor	MURATA	GCM155R71H104KE02D	SMD	Reflow	CAPC1005X55N-	100 nF	50 V	AEC-Q200	10%	HW01, HW031a,
5	112	C200BDIOA, C200BDIOB,	CAP	Chip ceramic capacitor Fail	MURATA	GCJ21BR72A104KA01L	SMD	Reflow	CAPC2012X145N-	100 nF	100 V	AEC-Q200	10%	HW021, HW021,
6	32	C304UAIOA, C304UAIOB,	CAP	Chip ceramic capacitor	MURATA	GRM155R61A474KE15D	SMD	Reflow	CAPC1005X55N-	470 nF	10 V			HW032a, HW032a,
7	16	C307BAIOA, C307BAIOB,	CAP	Chip ceramic capacitor	TDK	CGA2B3X7R1H104K050BB	SMD	Reflow	CAPC1005X55N-	100 nF	50 V	AEC-Q200	10%	HW031a
8	96	D200BDIOA, D200BDIOB,	DI		ROHM	RAS40-04HYFH	SMD	Reflow	SOT95P240X115-	40 V	120 mA	AEC-Q101	0.1	HW021, HW021
9	16	D303BAIOA, D303BAIOB,	DI		DIODES	BAS70-06Q-7-F	SMD	Reflow	SOT91P240X110-	70 V	70 mA	AEC-Q101	N/A	HW031a
10	16	D304BAIOA, D304BAIOB,	DI		VISHAY	TLMS1000-GS08	SMD	Reflow	DIO_VISHAY_TLMS	1.8 V	12 V	AEC-Q101	N/A	HW031a
11	1	DS100PS	DI	Zener diode	ON SEMICONDUCTOR	SZBZX84C5V6LT1G	SMD	Reflow	SOT95P237X111-	5.6 V		AEC-Q101	0.05	HW01
12	2	J100MN, J105MN	CON	Board to wire Connector -	MOLEX	504194-1070	SMD	Reflow	MOLEX_504194-	10 Pin Number	50 V	NON	N/A	HW00
13	3	J101MN, J102MN	CON	Board to wire Connector -	MOLEX	504194-1670	SMD	Reflow	Molex_504194-	16 Pin Number	50 V	NON	N/A	HW00
14	1	J104MN	CON	Board to wire Connector -	TE CONNECTIVITY	966658-2	THROUGH HOLE	Selective DIP	TE_966658-2	32 Pin Number	12 V	NON	N/A	HW00
15	16	K300BAIOA, K300BAIOB,												HW031a
16	1	LDO100PS	IC	Low drop-out regulator	MAXIM	MAX16910CATA8/V+T	SMD	Reflow	SON65P300X300X8			AEC-Q101		HW01
17	8	O200BDIOA, O200BDIOB,	TRT	Small signal MOSFET - N	FAIRCHILD	2N7002	SMD	Reflow	SOT95P240X120-3	60 V	120 mA	NON	N/A	HW021
18	16	O300BAIOA, O300BAIOB,	TRT	Digital type - Bipolar	ON SEMICONDUCTOR	SMUN5211T1G	SMD	Reflow	SOT65P220X100-			AEC-Q101		HW031a
19	4	B100PS, R101PS, R102PS,	RES	Thick film chip	KOA	RK73Z1ETTP	SMD	Reflow	RESC1005X40N-	0 ohm		AEC-Q200		HW01
20	1	R104PS	RES	Thick film chip	KOA	RK73G1ETTP1002F	SMD	Reflow	RESC1005X40N-	10 K-ohm		AEC-Q200	0.01	HW01
21	16	R200BDIOA, R200BDIOB,	RES	Thick film chip	KOA	RK73G1JTTD1000F	SMD	Reflow	RESC1608X55N-	100 ohm	0.1 W	AEC-Q200	0.01	HW021
22	16	R201BDIOA, R201BDIOB,	RES	Thick film chip	KOA	RK73H1JTTD1002F	SMD	Reflow	RESC1608X55N-	10 K-ohm		AEC-Q200	0.01	HW021
23	96	R300BAIOA, R300BAIOB,	RES	Thick film chip	KOA	RK73H1ETTP1001F	SMD	Reflow	RESC1005X40N-	1 K-ohm		AEC-Q200	1% (100ppm)	HW031a, HW031a,
24	64	R302BAIOA, R302BAIOB,	RES		KOA	RN731JTTD2203B25	SMD	Reflow	RESC1608X60L-HP	10 K-ohm		NON	0.001	HW031a
25	24	U300BAIOA, U300BAIOB,	IC	Operational amplEiselfier	MICROCHIP	MCP6002T-E/MS	SMD	DIP	SOP65P489DX110-	0 IC	0 Ah	AEC-Q100	0.3	HW031a, HW031a,
26	16	U301BAIOA, U301BAIOB,	IC		TEXAS INSTRUMENTS	TMUX1072RUTR	SMD	Reflow	QFN40P200X170X5	12 Pin Number	6 V	NON	N/A	HW031a


BidirectionalVoltageShifter.asc
.param Rbias=10k
.param R_LV=500k R_HV=6 param VHIL=5
.tran 0.02
.dc V1 0 12 0.2
.tran 0 0.2 0.1
.model NMOS NMOS LEVEL = 3 VMAX = 1E+006 ETA = 0 VTO = 2.154
.model DGDO D IS = 1E-014 N = 1.04 CJO = 1E-015 M = 0.6
.model DSUB D IS = 6.830E-010 N = 1.576 RS = 0.1408 BV = 72 CJO = 8E-012 VJ = 0.8 M = 0.6474
.model DLIM D IS = 0.0001
.ENDS
*Diodes N7002 Spice Model v0 Last Revised 2017/2/9
BidirectionalVoltageShifter
V(v_hv_1)
V(v_lv_1)
V(v_hv_2)
V(v_lv_2)
12V
6V
0V
5.0V
2.5V
0.0V
5.5V
2.8V
0ms
10ms
20ms
30ms
40ms
50ms
60ms
70ms
80ms
90ms
100ms


I(D_led_1)
V(vctrl)
V(adc1)
V(not_en1)
V(dac1)
V(ecu1)
3.0mA
2.7mA
2.4mA
2.1mA
1.8mA
1.5mA
1.2mA
0.9mA
0.6mA
0.3mA
0.0mA
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V
1.5V
1.0V
0.5V
0.0V
0ms
50ms
100ms
150ms
200ms
250ms
300ms
350ms
400ms
450ms
Right-Click to manually enter Horizontal Axis Limits