Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

MECCANICA E SCIENZE AVANZATE DELL'INGEGNERIA

Ciclo 35

Settore Concorsuale: 09/C2 - FISICA TECNICA E INGEGNERIA NUCLEARE

Settore Scientifico Disciplinare: ING-IND/10 - FISICA TECNICA INDUSTRIALE

THERMAL AND AIR QUALITY MODELING OF AN ELECTRIC VEHICLE CABIN WITH LOW-COST SENSORS AND REINFORCEMENT LEARNING

Presentata da: Luigi Russi

Coordinatore Dottorato

Lorenzo Donati

Supervisore

Giovanni Semprini

Co-supervisore

Beatrice Pulvirenti

Esame finale anno 2023

Acknowledgement

I would like to express my heartfelt gratitude to Giovanni Semprini and Beatrice Pulvirenti, my supervisors, for their invaluable guidance, support, and encouragement throughout my PhD journey. Their advice and constructive criticism have helped me to improve my thesis in a number of ways.

I would like to thank Paolo Guidorzi for his contribution in creating the experimental setup, for teaching me how to manipulate and improve it.

I am also grateful to James Brusey for his guidance and expertise throughout the external stay and for showing me a new way to look at the same problem using Reinforcement Learning. Not to mention his help with steep learning curve tools like Emacs and OpenAI.

I would also like to thank my friends and family for their support and understanding throughout this challenging yet rewarding experience. A special mention in this category goes to Giulia, who has litteraly been there every time I nedded to keep up.

I am also thankful to the entire Industrial Engineering department and its staff for the resources and guidance they provided during the course of this research.

Finally, I would like to thank everyone who has shared with me this writing time and has provided me with feedback and encouragement, even if for a short period: the Centre for Computational Science and Mathematical Modelling of the Coventry University staff and PGRs.

Thank you all.

Abstract

The design process of any Electric Vehicle (EV) system has to be oriented towards the best energy efficiency, together with the constraint of maintaining comfort in the vehicle cabin. Main aim of this study is to research the best thermal management solution in terms of HVAC efficiency without compromising occupant's comfort and internal air quality. The work started with a literature review of the studies focused on maintaining thermal comfort in dynamic, asymmetrical condition and studies regarding the minimisation of energy use from the Heating, Ventilation and Air Conditioning (HVAC).

In terms of hardware, the use of Low Cost Sensor (LCS) in the automotive landscape was thoroughly explored. An Arduino controlled Low Cost System of Sensors (LCSoS) was developed and compared against reference instrumentation, against which has an average R^2 of 0.92. then used to characterise the vehicle cabin of a production electric vehicle in terms of thermal comfort and air quality in real parking and driving conditions trials. At same time, data about the energy use of the HVAC was retrieved from the car On-Board Diagnostic (OBD) port. This part of the work showed that energy savings using recirculation can reach 30 %, but pollutants concentration in the cabin builds up in this operating mode. Moreover, the temperature profile appeared strongly nonuniform with air temperature differences up to 10 °C.

Optimisation methods often require a high number of runs to find the optimal configuration of the system. Fast models proved to be beneficial for these task, while CFD-1D model are usually slower despite the higher level of detail provided. In this work, the collected dataset was used to train a fast Machine Learning (ML) model of both cabin and HVAC using linear regression. Average scaled RMSE over all trials is 0.4 %, while computation time is 0.0077 ms for each second of simulated time on a laptop computer.

Finally, a Reinforcement Learning (RL) environment was built in OpenAI and Stable-Baselines3 using the built-in Proximal Policy Optimisation (PPO) algorithm to update the policy and seek for the best compromise between comfort, air quality and energy reward terms. The learning curves show an oscillating behaviour overall, with only 2

experiments behaving as expected even if too slow (minor improvements over 24 days of simulated time). This result leaves large room for improvement, ranging from the reward function engineering to the expansion of the ML model.

Contents

List of Figures

List of Tables

Glossary

EV Electric Vehicle

BEV Battery Electric Vehicle

HVAC Heating, Ventilation and Air Conditioning

PMV Predicted Mean Vote

PPD Predicted Percentace of Dissatisfied

SET Standard Effective Temperature

UTCI Universal Thermal Climate Index

DR Draft Rate

EDT Effective Draft Temperature

EHT Equivalent Homogeneous Temperature

MST Mean Skin Temperature

TSV Thermal Sensation Vote

TCV Thermal Comfort Vote

IAQ Internal Air Quality

OBD On-Board Diagnostic

PM Particulate Matter

VOC Volatile Organic Compounds

LCS Low Cost Sensor

LCSoS Low Cost System of Sensors

CWT Climatic Wind Tunnel

ML Machine Learning

RL Reinforcement Learning

PPO Proximal Policy Optimisation

FTL Front Top Left

FTC Front Top Center

FTR Front Top Right

FML Front Middle Left

FMC Front Middle Center

FMR Front Middle Right

FBL Front Bottom Left

FBC Front Bottom Center

FBR Front Bottom Right

BTL Back Top Left

BTC Back Top Center

BTR Back Top Right

BML Back Middle Left

BMC Back Middle Center

BMR Back Middle Right

BBL Back Bottom Left

BBC Back Bottom Center

BBR Back Bottom Right

Chapter 1

Introduction

One of the major barriers to electric vehicle adoption is due to the limited amount of energy stored in the batteries which is needed for traction and auxiliary systems operation. Focusing on passenger's thermal requirements, ICE vehicles can rely on waste heat, at least for heating operations. Conversely, an optimised thermal management of heat loads and gains is crucial for battery electric vehicles (BEVs). This work sits in line with the studies regarding this unprecedented shift from heat disposal to heat management in vehicle cabins [Opt21; 23].

Reducing the energy consumption of the HVAC system, as well as other auxiliary systems, is of paramount importance in the era of migration to electric powered transportation; the major challenge being to achieve this while maintaining high levels of comfort inside the cabin. On one hand, the best thermal management of the car's cabin is obtained by maximising the comfort level along with minimising electrical power demand [140; Nil04; Zha+19; PBW19]. On the other hand, Internal Air Quality (IAQ) related quantities inside the vehicle's enclosure are affected by outside weather conditions as well as by the HVAC settings. From this perspective, every improvement of the cabin thermal management must be IAQ aware [Org14; Org10; HRO09; SMA20].

In terms of modeling and optimisation, different choices can be made depending on the focus being on passenger's comfort or on the HVAC system and its control, with the cabin model representing the link between the two extremes. Comfort-driven studies usually require a great level of detail on the passenger and its surroundings, together with complex and expensive experimental work involving manikins, computational fluid dynamics (CFD) and access to Climatic Wind Tunnel (CWT) facilities [War+20; Oze+19; Oi+19]. HVAC-driven studies instead, make often use of simplified cabin environments, often relying on 1D physics based simulators to save computational resources for the exploitation of the dynamic evolution of the system [Cho+18; KTM21].

Similarly to HVAC and control studies, also optimisation techniques seek for fast

models, given the frequently high number of runs needed to find the optimal configuration and operation of the system. A state of the art approach to solve this problem relies on the use of ML systems with policy gradient reinforcement learning (PGRL) methods to find the optimal compromise between comfort and energy use [Che21]. Where RL is a discipline halfway between ML and control theory consisting in interaction between an agent and the environment happening by subsequent observation of the state of the environment and taking actions according to a certain policy with the objective of maximising a reward [BK19]. A convenient, stepped model of the cabin, HVAC and control system is required in this case, with some authors using machine learnt CWT and 1D model simulation data to build it [Jes+22].

1.1 Motivation

State of the art 1D cabin models can be very accurate. They consist of several thermal masses corresponding to the main inertial masses that interact between them and the HVAC components by conduction and radiation and with the air by convection. ML-based simulators however, can be accurate and fast enough to provide a solid alternative, opening the way for novel optimisation approaches. To achieve this, a lot of data from measurements and simulations are needed, as more information could be added if enough data is available to train and test the model.

In order to train the model, we opted for the experimental characterization of the vehicle cabin using a system of low-cost sensors based on Arduino. On one hand, this required the construction of the system, its performance evaluation against a reliable measuring station, and the execution of several experiments in summer and winter real driving conditions. On the other hand, it provided a portable and customisable setup which relies on open hardware and software thus improving the reproducibility of research. On top of this, the use of RL in systems for EVs thermal management optimisation is still an under-explored field, particularly if combined with real-driving data from low-cost sensors.

1.2 Research questions

The aim of this study is to demonstrate the capabilities of an optimisation framework that uses low-cost sensor data, ML and RL to solve a thermal management problem in electric vehicle cabins.

The overall aim can be broken down into the following sub-aims, each associated with a research question:

- First aim is to build a low-cost system of sensors capable of providing good enough data to charachterise the cabin environment in terms of comfort and air quality. This is also to demonstrate the metrological capabilities of a low-cost sensors in the automotive field, thus enabling its operation in transient, non-uniform and moving environments such as vehicle cabins.

  Research Question 1 - Can a low-cost sensor system accurately characterize the cabin environment in terms of comfort and air quality?

- Second aim is to verify if it is possible to build an accurate model without the use of a detailed CFD-1D simulation of the cabin environment and CWT data but using a dataset obtained from low-cost sensors in real driving conditions instead.

  Research Question 2 - Is it possible to build an accurate cabin and HVAC model without detailed simulation using real driving data from low-cost sensors?

- Third aim is to perform the training of the RL agent in Stable Baselines3 with PPO to find the optimal thermal management policy.

  Research Question 3 - Can a reinforcement learning agent be trained to find the optimal thermal management policy that maximizes comfort and air quality while minimizing energy use?

By answering these questions, the study aims to demonstrate the feasibility and effectiveness of using low-cost sensors and machine learning techniques for thermal management in vehicle cabins.

1.3 Publications

This research has lead to the following publications:
Journal

- Luigi Russi et al. "Air Quality and Comfort Characterisation within an Electric Vehicle Cabin in Heating and Cooling Operations". In: Sensors 22.2 (2 Jan. 2022), p. 543. ISSN: 1424-8220. DOI: 10.3390/s22020543. URL: https://www.mdpi.com/1424-8220/22/2/543

Conference Proceedings

- Luigi Russi et al. "Air Quality and Comfort Characterisation within an Electric Vehicle Cabin". In: 2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive). 2021 IEEE International Workshop on Metrology for

Automotive (MetroAutomotive). July 2021, pp. 169–174. DOI : 10.1109/MetroAutomotive50197.2021.9502853.

1.4 Thesis structure

The remaining part of the thesis has the following structure:

Chapter 2 starts with a review of the comfort models used in vehicle cabins depending on the scope, main research directions are outlined in the subfields of comfort, air quality and HVAC operation modeling.

Chapter 3 contains the description of the experimental setup, from design principles to its performance evaluation against reference instrumentation.

Chapter 4 reports a series of experiments on air quality and energy efficiency inside the passenger compartment of an electric vehicle. The relation between consumed energy, HVAC system settings and pollutant concentrations is obtained, together with the thermal profile of the vehicle cabin.

Chapter 5 describes the construction of the machine learnt cabin-HVAC model, of the simulation environment. A progressive detach of the model from the data that generated it to end up in completely virtualised optimisation framework based on RL in shown.

Chapter 6 summarises the main results, it also contains some limitations of the work as well as future research directions.

Chapter 2

Background of comfort, HVAC, and control-driven modeling approaches in vehicle cabins

In revieving the state of the art, we did not define an hypothesis to test in advance, but rather we worked in the cabin HVAC context and the important questions emerged during the review. Main aim of the chapter is to review state of the art comfort models in order to locate the most suitable one to be coupled with the HVAC system model of an EV. Then to investigate the effectiveness of available range improvement techniques in terms of driving performance and range. In other words, we first aim to review most popular thermal comfort models and methods for vehicle cabins. Second we aim to define which one is the best in terms of performance if coupled with an HVAC model of an electric vehicle. Third we aim to give a contribution in consistently organising the knowledge base of thermal comfort in vehicles. Lastly we aim to define major research directions in cabins thermal management and enter one or more of them.

The chapter has the following structure. Once defined the boundaries of the study and underlined its importance, we report the extent of existing knowledge of the matter together with some limitations. In Section section 2.1 we present the approach adopted for the review. Starting from the tools used for documents collection, then we introduce the thematic organisation of papers through a 5 terms taxonomy. At the end of the section we focus on key publications presenting them in a classification table. Section section 2.2 is about the main findings in the field of comfort-driven studies, which is the first one of the two terms of the taxonomy first layer. The most widely used comfort models depending on the application are compiled, together with techniques used in state of the art studies. In Section section 2.3 the focus is on the second term of the first layer: HVAC-driven studies. First, techniques regarding heat load reduction are

compiled, then main HVAC efficiency techniques are given. An overview on internal air quality (IAQ) in cabins, an often considered secondary but always more important topic, is given in Section section 2.4. Some motivations on the choice of low-cost sensors in terms of sensing equipment for this research are presented. Section section 2.5 explores the wide field of modeling introducing a few key concepts about Machine Learning (ML) and Reinforcement Learning (RL), using physics based model as a reference. Finally, main research directions, as well as the approaches suitable for electric vehicles, in terms of both comfort model and HVAC system optimisation, as well as air quality studies are then summarised in section 2.6, proposing some research questions to be addressed in this work.

2.1 Literature review methodology

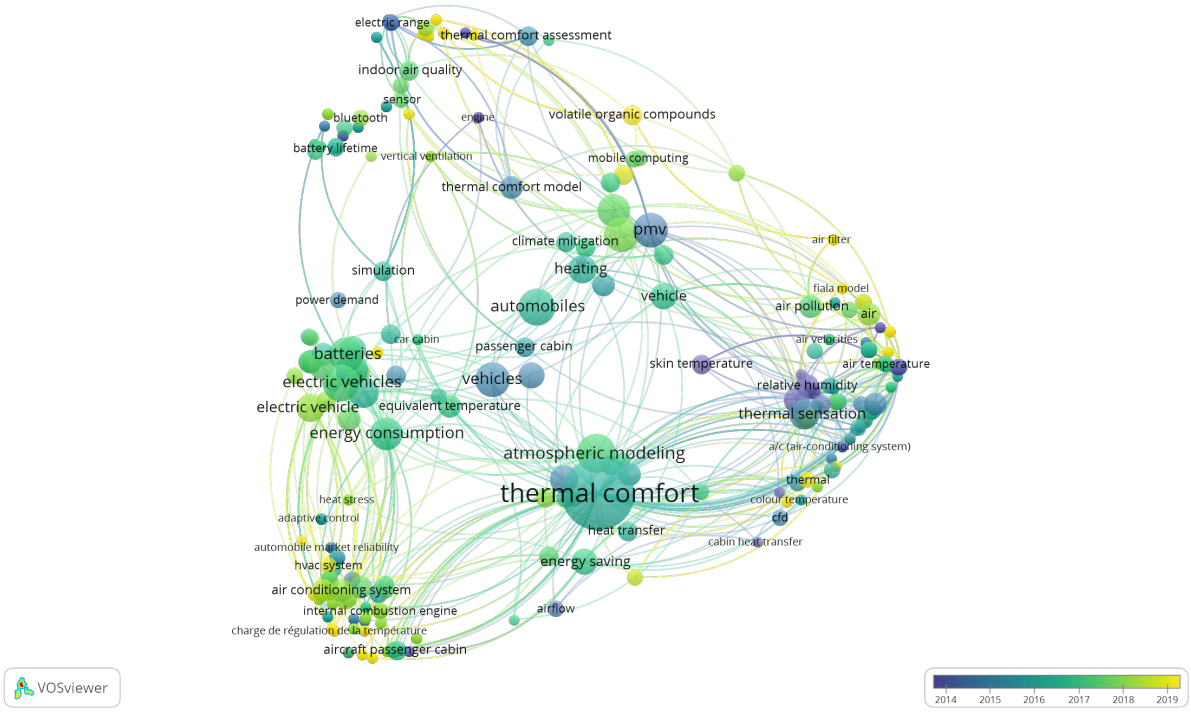


Figure 2.1: *State of the art bibliometric map by keywords made with VOSviewer, with the aim of discovering thematic clusters in the cabin thermal management landscape.*

In order to perform a consistent and well organised state of the art exploration (review), we followed a qualitative, semi-structured approach [JJ19; Sny19]. In other words, we explored the state of the art to find major research directions, in a qualitative tree structured approach, but without the conceit of providing quantitative results. In fact, given the wide scope of the matter, ranging from pure comfort studies to pure

HVAC-system and optimisation studies, it is not reccomended to quantitatively compare the results. The very first step was the definition of as relevant and unbiased as possible search terms, a limited time interval and a result selection criterion to trigger and filter the output from the following databases:

- ScienceDirect – Physical Sciences and Engineering
- IEEE Xplore - Electrical and Electronic Engineering
- SAE MOBILUS – Automotive and Mobility Engineering
- Google Scholar – Scholarly Research and Patents

The search terms used were "thermal comfort", "vehicle", "cabin", "automotive", "EVs", "method", "assessment", "air quality" and meaningful combination of them. We decided to limit the search to documents not older than 5 years, exception made for a few milestones. Moreover, the search output was limited to the first 20-25 results.

The whole search output has been collected in an open source reference manager called Zotero [Zot20], which enabled us to keep track of a collection of almost 400 documents, organise them in the same place, perform targeted searches within the collection and also to manage the bibliography in a simple and effective way.

A visual representation of the state of the art resulting from the bibliographic search is provided in Figure 2.1. The database compiled in Zotero has been exported and processed with VOSviewer, a weel estabilished tool for bibliometric mapping [vEW20]. The construction of the map is based on a co-occurrence matrix with a three step process described in detail in [vEW10]. It is interesting to note that a map created with keywords from the collection, using link strength as weight and publication year as score, provides clusters representative of the main research directions in this field. Among the main clusters, the most recent ones are focused on electric vehicles of course, but also hvac system operation and air quality. A cluster related to hardware and sensig technologies is also well connected with theme of thermal comfort.

Table 2.1: *Documents classification*

Reference	Main focus
[War+20]	Data-driven prediction of thermal comfort
Thermal model	Vehicle cabin CFD model, with virtual thermal manikin and machine learning used to represent a wide range of environmental conditions, HVAC settings and glazing properties
Numerical approach	Cabin and manikin CFD model in STAR CCM+, coupled with three machine learning algorithms, a linear model, an ensemble predictor and an artificial neural network

Continuation of Table 2.1

Reference	**Main focus**
Experimental	Climatic wind tunnel for CFD model, 10 fold cross validation for machine learning algorithms
Indexes	EHT, T_{eq}, PMV, PPD
[ZLC19]	Thermal comfort analysis under outdoor driving conditions
Thermal model	Four terms heat balance, measured quantities outside and inside cabin, surveys
Numerical approach	Experimental data processing and manipulation
Experimental	Comparison of measured and calculated quantities (MST) against surveys data (TSV)
Indexes	MST, TSV
[Xie+19]	Comfort based HVAC control strategy
Thermal model	Component based for HVAC, thermal network method and five heat loads for cabin
Numerical approach	CFD for condenser and evaporator, lumped parameters learner-regulator
Experimental	Measures in environmental chamber
Indexes	PMV, $T_{comfort}$
[SS19]	Cabin thermal management strategy, HVAC efficiency
Thermal model	Cabin and HVAC models with recirculation, blower, evaporator, heat exchanger
Numerical approach	Dynamical model with four states for cabin, quasi-static model for HVAC components. Linear quadratic approximation
Experimental	Comparison against nonlinear and baseline strategy
Indexes	PMV (extended with T_{amb} and Q_{sun})
[PBW19]	Impact of occupant-HVAC interaction on comfort
Thermal model	Nilsson equivalent temperature model
Numerical approach	Traditional sense-and send system, MySQL database
Experimental	Real driving conditions trials on production vehicles, measures of temperature, humidity, air speed, mean radiant temperature, CO2, solar loading.
Indexes	TSV, T_{eq}
[Pau+19]	Evaluation of thermal comfort according to ISO 14505
Thermal model	Comfort evaluation in transient, non-uniform environment using Fanger, Nilsson model and direct methods
Numerical approach	Dedicated automatic PMV apparatus, neuro-fuzzy control thermal manikin, data processing and manipulation
Experimental	Comfort sense measurement tool, measure of air temperature and velocity in the cabin, thermal manikin, surveys
Indexes	PMV, TSV, T_{eq}
[Och+19]	Ceiling circulator optimisation
Thermal model	Separate outlet (high induction) and blower (centrifugal) to make the circulator thinner, thermal comfort verification
Numerical approach	air outlet CFD based on lattice-Boltzmann method
Experimental	CFD comparison against conventional outlet, cooling down test using thermal manikin with vehicle placed in a wind tunnel
Indexes	SET, TSV
[Hep+18]	Evaluate a vehicle HVAC with a passive sensor manikin coupled with a thermal comfort model
Thermal model	Passive sensor manikin coupled with TAITherm Human Comfort Model

Continuation of Table 2.1

Reference	**Main focus**
Numerical approach	multi-physics solver for thermal conduction, radiation, and convection under both steady-state and transient conditions
Experimental	Human subject thermal sensation and comfort in the same test conditions
Indexes	EHT, T_{eq}
[YFL18]	Novel multi-objective HVAC control strategy considering comfort and efficiency
Thermal model	Fanger for comfort, Zhang energy based HVAC model, thermal dynamic model of cabin and powertrain
Numerical approach	Matlab/simulink like 1D model
Experimental	Comparison against two widely used control strategies, thermostat and PI
Indexes	PMV, PPD
[Thi+18]	investigation of Zhang model in a controlled thermal environment
Thermal model	Compare thermal sensation and comfort from Zhang and Fiala with experimental votes
Numerical approach	standalone implementation of both Zhang thermal sensation and comfort algorithm, Fiala DTS thermal comfort algorithm
Experimental	thermal test bench on a b-segment car with human subjects
Indexes	Sensation index, comfort index
[Pau+18]	Effect of human presence in vehicle cabin
Thermal model	Numerically evaluate the cabin thermal environment with and without a virtual manikin
Numerical approach	Cabin and manikin CFD models solved in Ansys 18
Experimental	Comparison of two study cases
Indexes	PMV, PPD
[Mor+18b]	Develop an occupant comfort prediction under non uniform an unsteady conditions
Thermal model	Quantitative evaluation of comfort using equivant temperature of a thermal manikin
Numerical approach	Experimental data processing and manipulation
Experimental	Thermal manikin in wind tunnel using air-enthalpy and refrigerant-enthalpy methods
Indexes	T_{eq}
[Man+18]	Assessment of auxiliary energy needs under WLTP cycle
Thermal model	Monozonal modelling approach with wall, cabin interior and HVAC models
Numerical approach	1D Dymola thermal vehichle model, medium class hybrid vehicle in dynamic programming
Experimental	Comparison against baseline scenario simulations
Indexes	SET
[Lah+18b]	Dynamic programming to solve the optimal control problem of balancing comfort and range
Thermal model	EV model composed of three sub-systems: the HVAC system, the powertrain and the battery, quadratic discomfort criterion based on temperature and humidity ranges
Numerical approach	Dinamic programming (DP) 1D model of EV with a comfort criterion
Experimental	Comparison of two study cases

Continuation of Table 2.1

Reference	**Main focus**
Indexes	T_{ref}, H_{ref}
[ISM18]	Develop a method to evaluate thermal sensation of a human body when using various thermal control devices like: seat and stearing wheel heater, ventilated seats
Thermal model	Extention of human thermal model constructed by Imai with conduction, 16 to 26 sbody segments,
Numerical approach	1D human thermal model, experimental data processing
Experimental	Human subject thermal sensation in 26 locations and whole body, environmental measures with thermal manikin
Indexes	local SET*, TSV, WSV
[Bru+18]	Show that Reinforcement Learning (RL) reliably produces a controller that uses less energy while delivering better comfort than existing hand-coded approaches
Thermal model	Comfort control as a Markov Decision Process (MDP), learning in simulation, 1D cabin and HVAC model
Numerical approach	"Monte Carlo Exploring Starts (MCES) and Monte Carlo -soft, continuous state MDPs, Sarsa(λ)"
Experimental	Data from real car in a climatic wind tunnel, comparison with other control techniques
Indexes	T_{air}, T_{avg}, T_{eq}
[Psi+17]	Manikin-based metodologies for thermo-physiological response
Thermal model	Thermal manikin coupled with human thermoregulation model
Numerical approach	Real time iterative exchange of data between manikin and model (feedback loop)
Experimental	Human subject data from literature, mean skin and body core temperatures
Indexes	T_{eq}, LMV
[Nea+17]	compare the global and absolute thermal comfort indexes for two vehicles with different air distribution systems
Thermal model	
Numerical approach	Numerical simulation of cabin and occupants to calculate global thermal comfort indexes
Experimental	Comparison of old comfort indexes against proposed ones
Indexes	PMV, PPD, DTS, GTCI, GATCI
[KKJ17]	Impact of cabin insulation, thermal mass and glazing on comfort
Thermal model	detailed numerical model of cabin surfaces, 1D GT_SUITE comfort and HVAC model, Fanger for comfort
Numerical approach	Complete cabin model in STAR-CCM+ of both baseline and insulated ceiling, N-S for compressible 1D flow
Experimental	Comparison with baseline result in terms of insulation and glazing
Indexes	PMV
[Hep+15]	Comparison of adaptive and passive manikin for thermal comfort assessment
Thermal model	Adaptive manikin with thermoregulation model and passive manikin with a thermo-physiological model
Numerical approach	Real time coupling with RadTherm Human Thermal Model plus virtual manikin for the passive case
Experimental	Human subject in the same test conditions

Continuation of Table 2.1

Reference	Main focus
Indexes	T_{eq}, Mean Skin Temperature, Overall Sensation, Overall Comfort
[Cro+15]	Review of thermal comfort models for buildings and vehicles
Thermal model	Comfort definition, main comfort models, practical assessment with manikins and CFD
Numerical approach	CFD, virtual thermal manikin with thermo physiological model
Experimental	Ranging from sensors to human subjects
Indexes	PMV,PPD, DR, EDT, ADPI, UTCI, T_{eq}
[DMS15]	Cabin heating of EVs with solar load
Thermal model	6 segments cabin, ASHRAE Standard 55, thermal time constant of the vehicle
Numerical approach	Comparison between measured and computed temperature
Experimental	Experiments on a Renault ZOE, 6 cabin and one outdoor temperature. Each repeated at head, torso and foot level
Indexes	T_{com}, T_{indoor}

2.1.1 Taxonomy and key publications

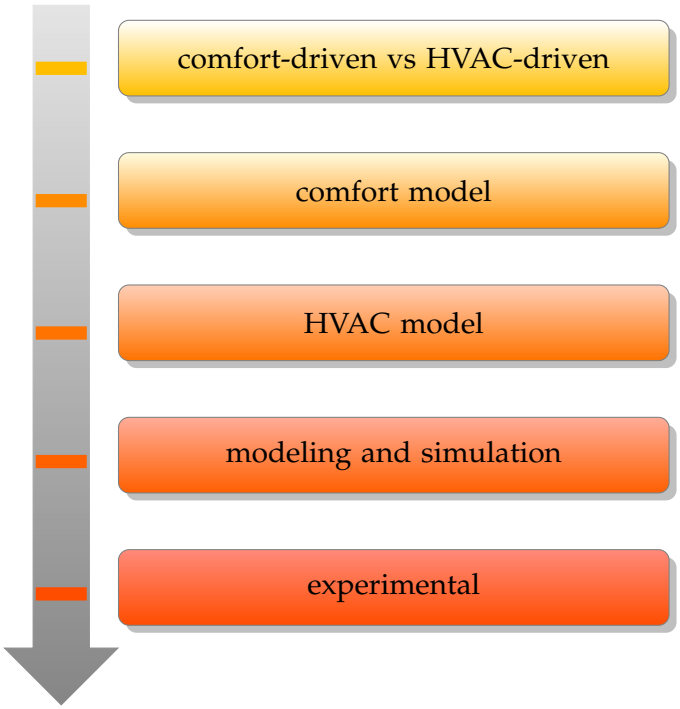


Figure 2.2: *Taxonomy developed to categorize articles*

Making a scan of abstracts and summaries in the collection is possible to have a general idea of the state of the art regarding thermal comfort in vehicles, this process is also useful to reject articles that match the search terms but are related to other sectors, such as buildings, aircraft and submersibles for example. Analysing the scheme allowed

us to develop a taxonomy (a systematic set of types and entities, figure 2.2) which guided us toward the goal of answering a first attempt research question: "Identify the major research directions for comfort aware HVAC operation in electric vehicles". The taxonomy has been used as a sieve to select and categorize the articles in a structured way.

The first distinction to make regards two main approaches that arose during the review. In fact, while the common research direction goes toward finding a balance between minimising energy use of auxiliaries and maintaining thermal comfort in the cabin, there are some studies which focus on the HVAC system, others on the comfort evaluation. However, both use a thermal model to quantitatively evaluate the performance of a certain design. For this reason we decided to consider also studies regarding HVAC performance using comfort models, together with studies on pure thermal comfort in vehicle cabins. In the following sections we will refer at the former as "HVAC-driven" approaches, the latter being "comfort-driven" approaches.The first layer of the taxonomy is about this two different approaches.

Second layer regards the comfort model chosen among different studies, with this type we want to gain knowledge about most widely used and effective comfort models in each one of the previous categories.

Third layer categorises HVAC model in terms of system components and vehicle cabin. It should be noted that cabin model is the link between thermal comfort and HVAC models, being the environment in which both approaches want to evaluate and control relevant parameters such as temperature, humidity, TSV, and pollutants concentration just to name some.

Having addressed the theoretical approach in the previous types, we continue with the fourth layer. This type assesses the HVAC modeling and simulation schemes used in the studies, which may vary a lot ranging from simple calculations to CFD.

In the fifth layer we categorise the validation techniques adopted.

After processing the articles database using the developed taxonomy as a sieve, we were able to identify major research directions for comfort-driven and HVAC-driven approaches. A summary of selected studies is given in Table 2.1, where the whole landescape of vehicle thermal management is organised. For each study a summary of the comfort, cabin and HVAC model is given, together with experimental methods and main indexes used. Main comfort indexes are also organised in systematically in Table 2.2, which contains input parameters required for their calculation and a brief description. Further considerations could be made with the aim of proposing comfort and HVAC techniques suitable for improving efficiency of battery electric vehicles (BEVs). In fact, an accurate evaluation of thermal comfort along with an energy efficient

HVAC system is crucial for BEVs, given that range is still an issue because of the limited amount of energy stored on board. Moreover, BEVs market share is expected to grow consistently in the next years, with companies building electric cars being more inclined to invest in research and development of their products, if compared with a technology that is becoming obsolete like internal combustion engine (ICE) vehicles [21].

Table 2.2: *Comfort Indexes*

Index	Input Parameters	Description
Predicted Mean Vote (PMV)	t_a-dry bulb air temperature (°C), t_{mr}-mean radiant temperature (°C), v_r-relative air velocity (m/s), RH-relative humidity (%), M-metabolic rate (met), I_{cl}-clothing insulation (clo)	Predicts the mean value of the thermal sensation votes (self-reported perceptions) of a large group of people on a sensation scale expressed from –3 to +3 corresponding to the categories "cold," "cool," "slightly cool," "neutral," "slightly warm," "warm," and "hot".
Predicted Percentage of Dissatisfied (PPD)	PMV	The PPD is an index that establishes a quantitative prediction of the percentage of thermally dissatisfied people determined from PMV.
Standard Effective Temperature (SET)	t_a-dry bulb air temperature in (°C) t_{mr}-mean radiant temperature (°C) v_a-air velocity (m/s) RH-relative humidity (%) M-metabolic rate (met) I_{cl}-clothing insulation (clo) S_b-body surface area (m^2) p_{atm}-atmospheric pressure (bar)	The temperature of an imaginary environment at 50% RH, 0.1 m/s average air speed (v_a), and tr = tdb , in which the total heat loss from the skin of an imaginary occupant with an activity level of 1.0 met and a clothing level of 0.6 clo is the same as that from a person in the actual environment with actual clothing and activity level.
Universal Thermal Climate Index (UTCI)	t_a-dry bulb air temperature in (°C) t_{mr}-mean radiant temperature in (°C) v_a-air velocity in (m/s) RH-relative humidity (%)	Defined as the air temperature of the reference environment which produces the same strain index value in comparison with the reference individual's response to the real environment. It is regarded as one of the most comprehensive indices for calculating heat stress in outdoor spaces.
Draft Rate (DR)	t_a-dry bulb air temperature in (°C) v_a-air velocity (m/s) T_u-air turbulence intensity in (%)	Correlation between local turbulence and thermal sensation.
Effective Draft Temperature (EDT)	t_ai-local air temperature (K) t_m-mean air temperature (K) v_ai-air velocity in (m/s)	Predicts comfort in terms of any sensation of coolness or warmth due to air temperature and velocity.
Equivalent Temperature (T_{eq})	t_a-dry bulb air temperature in (°C) t_{mr}-mean radiant temperature in (°C) v_a-air velocity in (m/s) I_{cl}-clothing insulation (clo) h_c-convective heat transfer coefficient ($W\,m^2/°C$) h_r-radiation heat transfer coefficient ($W\,m^2/°C$)	The uniform temperature of an imaginary black enclosure in which an occupant would exchange the same amount of heat by radiation and convection as in the actual non uniform environment.
Equivalent Homogeneous Temperature (EHT)	t_a-dry bulb air temperature in (°C) t_{mr}-mean radiant temperature in (°C) v_a-air velocity in (m/s) Q_s-solar load (W/m^2)	The uniform temperature of an imaginary enclosure with air velocity equal to zero in which a person will exchange the same dry heat by radiation and convection as in the actual non-uniform environment
Mean Skin Temperature (MST)	t_{sk}-local skin temperature in (°C) A_{sk}-local surface area (m^2) S_b-body surface area (m^2)	Area weighted temperature value of the local skin portion
Thermal Sensation Vote (TSV)	Survey	The overall or local thermal sensation responce of a human subject in an certain environment
Thermal Comfort Vote (TCV)	Survey	The overall or local thermal comfort responce of a human subject in an certain environment

2.2 Comfort-driven methods

Thermal comfort evaluation in vehicle cabins is gaining always more attention from both academy and industry, this is mainly for two reasons:

1. Every improvement on HVAC systems needs to be tested with a comfort model to verify that a certain component or technique has a positive (or at least neutral) effect on passengers thermal sensation.

2. Thermal comfort investigation in transient and strongly non-uniform environments like vehicle cabins is a challenging task and the knowledge on the topic is still fragmented.

According to some authors, the available standards are obsolete and there is not a clear consensus on which comfort model should be used in these environments [DVD16]. Control system studies for example, often use simple comfort models based on strong simplifying assumptions suitable for buildings like Fanger's Predicted Mean Vote (PMV) [YFL18]. Studies with thermal manikins instead, usually adopt sophisticated comfort evaluation techniques with detailed comfort zones segmentation and realistic physics. For the accurate evaluation of human thermal comfort in vehicle cabins it is necessary to use a detailed heat transfer model of the human body with its surroundings.

Before going deeper into comfort-driven methods, it is worth underlining the difference between thermal sensation and thermal comfort. If thermal sensation probably reflects the pure response of thermoreceptors, thermal comfort describes the synthesized human feeling about the body's thermal state as claimed by [Zha+10].

2.2.1 Thermal sensation and comfort models

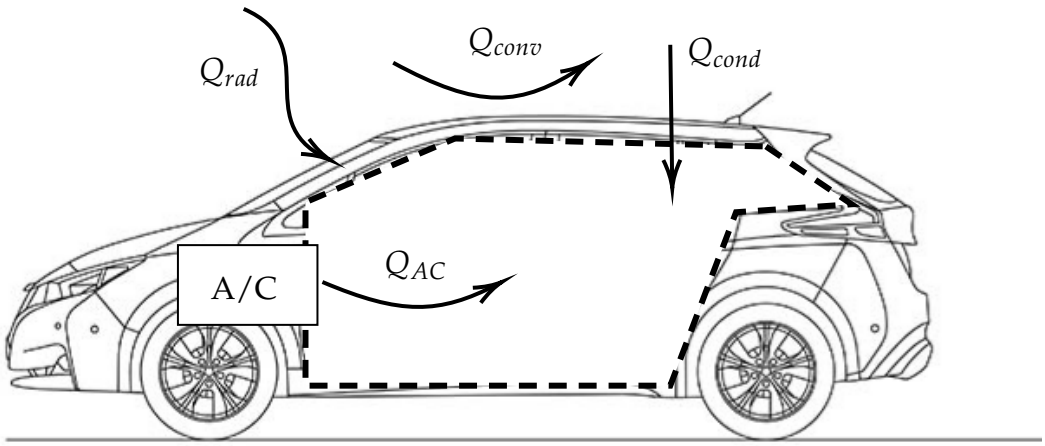


Figure 2.3: *Typical external heat loads in a vehicle cabin that affect passenger's comfort.*

Vehicle cabins are a difficult type of thermal environments because of their low thermal mass and their exposure to rapidly changing external conditions [ZLC19]. Figure 2.3 shows the main external loads that affect thermal sensation and comfort

inside the cabin in terms of the three heat exchange mechanism (convection, conduction, radiation), as well as the contribution of the HVAC system.

Thermal sensation alone is not sufficient to evaluate a subject's comfort, this is because the relationship between sensation and comfort is very different in transient and asymmetrical environments than in uniform, stable conditions [Zha03]. Thermal sensation models predict subjective perception in transient conditions linking the sensation with the physiological parameters (skin, core temperatures, rate of change of skin temperature). Some authors found that an averaged approach between Fiala DTS and Berkeley OS performed better than the models alone and the reason behind this behaviour is still unclear [Hep+18]. Available thermal sensation model to date:

- Stable asymmetrical
  - Weighting factor approach:
  - Ingersoll, Kalman et al. 1992
  - Matsunaga, Sudo et al. 1993
  - Kohri, Kataoka et al. 1995
  - Jones and Ogawa 1992
  - Brown and Jones 1997
  - Hagino and Hara 1992
  - EHT and piste approach:
  - Wyon, Larsson et al. 1989
  - Nilsson 2003
- Dynamic symmetrical
  - Hensel 1982
  - Ring and de Dear (DTS) 1993
  - Taniguchi, Aoki et al. 1992
  - Wang and Peterson 1992
  - Fiala DTS 1998
  - Guan, Hosni et al. 2003
  - Frank, Raja et al. 1999
  - Berkeley OS

The models summarised above each have a piece of the whole picture; they either address dynamic conditions without addressing asymmetrical conditions (e.g., Fiala, Wang, de Dear and Ring) or they address asymmetrical conditions (e.g., EHT) but without addressing dynamic conditions.

Despite having been developed for steady-state, uniform thermal environments, comfort models based on a fictive representation of the thermal environment with fixed parameters derived from empirical models are still widely used for vehicle cabins [Cro+15]. This is particularly true for studies which need a simple and straightforward model of the cabin, like control purposes models, as we will later explain in Section section 2.3.

Thermal comfort models known to date:

- Fanger
- Standard Effective Temperature (SET)
- Equivalent Temperature (T_{eq})
- Nilsson
- Bedford
- Madsen
- Zhang (Berkeley)

Fanger for example, proposed a single equation approach which leads to the evaluation of PMV and Predicted Percentace of Dissatisfied (PPD). Two indexes that are still used for predicting thermal sensation of cabin occupants. Several attempts have been made to improve thermal comfort prediction in transient and non-uniform environments. Among these, methods based on equivalent temperature T_{eq} are considered to provide the best results [PBW19]. Equivalent temperature or Nilsson's comfort model is indeed the method proposed by the only available standards [ANS17; ISO07]. That is still one of the most widely used methods based on equivalent temperature is, which provides two T_{eq} (one for hot and one for cold sensation) used to define an acceptable comfort range. With T_{eq} being the temperature of the stationary environment at which the human body exchanges the same amount of heat as in the transient one. Despite its reliability, ISO 14505 standard requires complex calculations on many body parts as well as the knowledge of radiant temperature. The latter is a quantity not easy do measure, if not with lab scale equipment [dAlf+21]. For this reason, some authors to simplify its calculation using linearisation techniques [Rom+21]. A step forward was made by Zhang with his PhD thesis, in which he deepened two main concepts:

1. Thermally asymmetrical environments lead to local body parts comfort being dependent on the conditions that the whole body is experiencing.

2. Transient thermal conditions lead to perception of comfort dependent on the changing conditions in time.

Zhang proposed to develop a local sensation model in the form of a function of skin and core (or mean skin) temperatures and their rates of change (Equation 2.1).

$$Local\ sensation = f(T_{skin}\frac{dT_{skin}}{dt}, T_{core}, \frac{dT_{core}}{dt}) \tag{2.1}$$

It should be noted that a sensation model needs physiological quantities that can be measured or calculated with a thermo-physiological model. Then the local thermal sensation can be calculated as a function of the physiological parameters. Having the local thermal sensation available, local thermal comfort depends on local sensation and overall sensation. Finally, overall thermal comfort can be calculated from the local one, but paying attention that it is not an additive quantity [Zha03].

The former approach was proved to perform well in asymmetrical and transient conditions and the associated results suggest that in these environments thermal neutrality becomes less relevant if compared with partial relief of discomfort, which becomes the main driver of climate control systems design process. A schematic representation of Zhang's approach is reported in figure 2.4, being one of the most complete and exhaustive view of the thermal sensation and comfort evaluation problem. Despite this, it should be noted that some authors criticised its application to practical comfort evaluation studies for control purposes [Hin+14].

Thermo-physiological models

When going forward with the understanding of local comfort concepts, one has to deal with thermo-physiological models of human body [Foj+20]. According to [Psi+17], a thermo-physiological model consists in a virtual representation of the human body made by a combination of a passive system reproducing the human body tissues properties and of an active system simulating the human thermo-physiological response. The active part of the system aims at reproducing the thermoregulatory function of the human body. They can cover different levels of detail depending on the thermoregulatory functions considered: heat storage, blood flow, sweating.

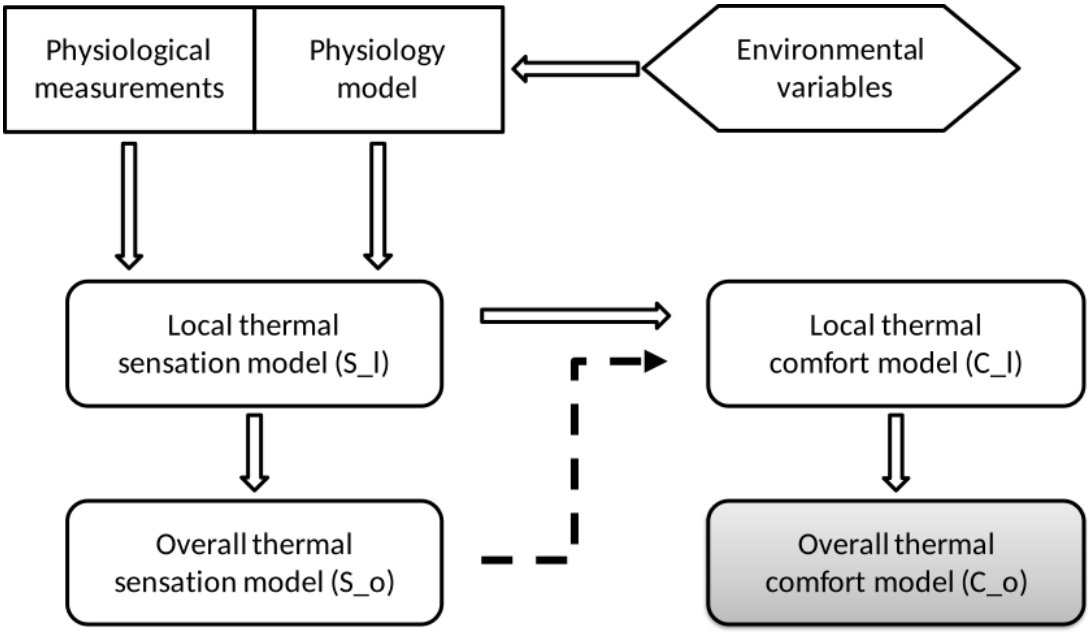


Figure 2.4: *A complete thermal sensation and comfort model proposed by Zhang [Zha03]*

2.2.2 Manikins

Latest available standard [ISO08] contemplates the use of thermal manikins as measuring instruments of local and whole body equivalent temperature (T_{eq}). A thermal manikin is defined as a "human sized and shaped sensor with its surface covered with numerous, individually controlled, heated zones". It is noticeable however, that with the development of this particular sector of comfort prediction different kinds of thermal manikins have been developed, they can be divided in three main categories: active (adaptive), passive (sensor) and virtual (numerical).

Available from many manufacturers, adaptive manikins are generally accurate, providing an estimation of human thermoregulatory functions. Passive manikins can measure the conditions at the boundary of the body that can then be used in a virtual human model to predict comfort [Hep+15; The20]. The use of numerical manikins is gaining always more attention from the scientific community and having even a related standard under development [ISO20]. One of the main drawbacks of thermal manikins is related to their cost and complexity. However, examples of low-cost systems has been successfully used by [Lan+18] as heat loads and obstruction to determine T_{eq} with an appropriate prior calibration. It should be noted though, that a thermal manikin alone is not sufficient for thermal comfort evaluation and it needs to be coupled with a thermal sensation and comfort model to translate the quantities measured by the manikin in an information about comfort.

A step forward in manikins accuracy comprises the concept of thermo-physiological human simulator. An extensive overview of the opportunities and constraints of adaptive manikins together with the related manikin based methodologies is provided by [Psi+17]. According to the authors there are two emerging trends in coupling thermal manikins with thermo-physiological models in feedback loop (i.e. thermo-physiological human simulators):

1. Set a heat production in the manikin to obtain surface temperature as feedback parameter for the thermo-physiological model (type 1 or Dirichlet approach)

2. Set surface temperature in the manikin to obtain heat production as feedback parameter for the thermo-physiological model.

Both have pros and cons, the main being the lack of an active cooling system to measure also heat gains not only heat loss of the human body. Another main issue has to do with the impossibility to reproduce some body reactions like skin perspiration and an accurate prediction of skin wetness. Moreover, fast changing environments are still challenging in terms of responsiveness of the system [Psi+17].

Going towards a complete virtualization of the comfort evaluation framework for design purposes, the works of [Oi+19; Lor+14] are among the most representative. A combined virtual cabin and manikin model was built and validated against the real system in order to evaluate the equivalent temperature or the Fiala Physiological Comfort (FPM) model numerically, a good agreement is shown to be achieveable for practical use. It should be noted though, that if the virtual manikin can be built with heat exchange models only, the use of CFD is almost mandatory for the cabin model, being air velocity field one of the most influencing factors of thermal comfort especially in summer.

2.2.3 CFD

Several effort have been made in the development of CFD codes for the evaluation of thermal comfort in recent years. Starting with [Fia98], which suggested the use of CFD for detailed analyses of complex environments, most of the following research used CFD for thermal comfort evaluation in cabins [Lor+14; Psi+17; Pau+18; Oi+19; War+20]. Main advantages of CFD are related to the avoidance of costly and problematic human subject tests, at least in the design phase. Moreover, if correctly validated it provides a very good spatial resolution of the quantities used to calculate different comfort indexes and possibly compare them. It should be noted though, that complex CFD models of human-cabin interaction in transient environments is still a challenging task from the

computational point of view. This said it is still possible to use CFD to obtain boundary condition and quantities needed to feed a simpler dynamical model.

2.3 HVAC-driven methods

Designing an optimised HVAC system is not an easy task because of the many aspects that need to be balanced at the same time. Among the methods to improve the efficiency of cabin thermal management, a first distiction can be made in terms of *passive* or *active* solutions [Dim].

A possible workflow for a modern HVAC system design is given in figure 2.5. Where the main techniques and research directions are summarised, together with their relation with HVAC and comfort models for design purposes.

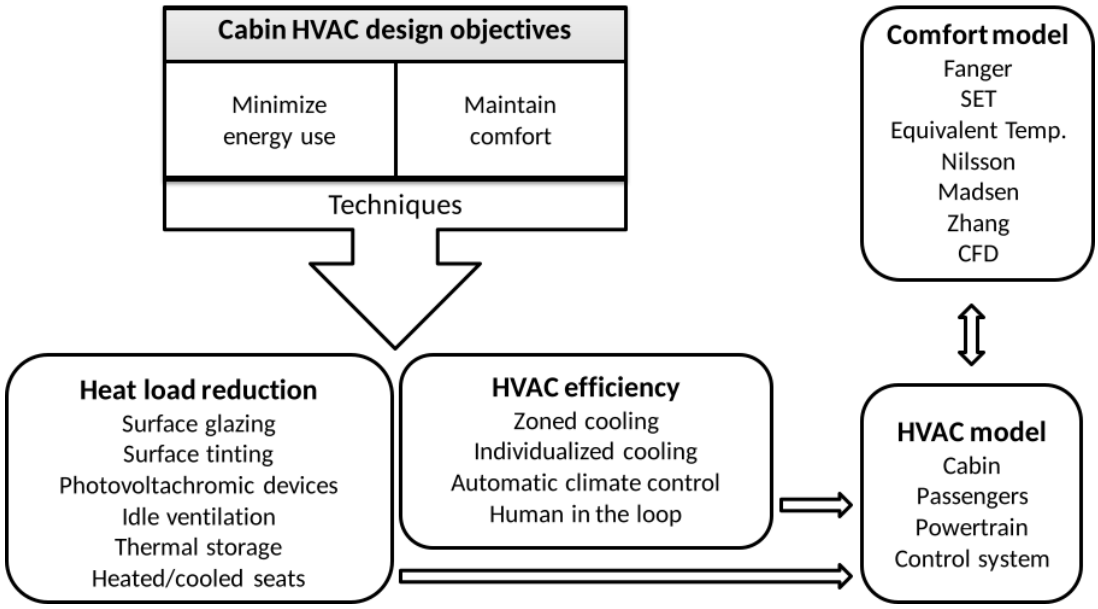


Figure 2.5: *Comfort aware HVAC design*

2.3.1 Heat load reduction

Passive solutions are focused on reducing the heat load that comes from the environment through the vehicle envelope by conduction and radiation. This can be done mainly reducing the thermophysical properties of the shell or changing the amount of solar radiation transmitted from the windows using different methods, often combined, each with advantages and drawbacks: [Kak15; GRP15; KKJ17; Lah+18a; Soc+16]:

- surface glazing

- surface tinting
- photovoltachromic devices
- idle ventilation
- thermal storage
- conditioned seats

2.3.2 HVAC efficiency

In recent years, several techniques are emerging over the classic centralised cabin thermal management, the most promising are focused on providing a more passenger centric thermal comfort or a reduced energy use in terms of finer control [LYE20; Mar+19; Kub+18; KLZ16]:

- zoned cooling
- individualised cooling
- automatic climate control (ACC)
- human in the loop control (HITL)

2.4 Air quality inside cabins and its sensing

Interest in air quality monitoring is gaining more and more attention from public authorities, companies and citizens for both outdoor and indoor environments. Nowadays, more than 80% of the world urban population exposed to pollution levels that exceed WHO limits, with estimated 6 million deaths each year related to it [Com+19; SMA20]. Climate change, pollution and COVID-19 issues are well known air quality related topics, but also automotive sector is pushing the development of novel air quality standards. Several studies have underlined the relevance of particle counters in the determination of adverse health effect air pollution, thus suggesting that both particle number and mass concentrations should be measured [Tit+08].

Research in the field of vehicle air quality is leading to new methods of testing and best practices, but still a dedicated standard on performance indicators does not exist. Several efforts have been made to define a standardised test method for interior air quality in the automotive field [Pha+19], but this is still an open question. A possible approach to address this issue relies on fractional air recirculation, demonstrating

that a compromise is achievable between the benefits of full recirculation and its side effects [Gra+13]. Other authors propose approaches based on the use of a signal from environmental prediction services [Alg+19], and/or on-board sensors [NEW19] to trigger automatic climate control, even though there is still no clear consensus on how to implement these techniques in the HVAC system's control strategy. Such an approach would require a trade-off between real time and integral I/O techniques, the former leading to large uncertainties but faster, while the latter provide slower but more stable results [Li+18a].

The Joint Research Centre of the European Commission and DG Service Environment are pushing for advancement in this field by stimulating research improvements achieved through the use of low-cost sensors. Although the data measured with these type of sensors are less accurate than laboratory reference equipment, their use has grown greatly in recent years, in applications concerning indoor air quality [MLA17]. Their ease of use, coupled with current scientific advancements [RHB18; YYT12; Try+20], makes them suitable for real-time monitoring applications. The road has therefore been opened for the employment of low-cost sensors in the automotive sector, such as monitoring air quality inside the passenger compartment.

In this context, the thorough diffusion of medium-cost [Ada+19] and low-cost sensors is considered a promising technology for the improvement of spatial and temporal resolution of these measurements. Though, it must be noted that accuracy and data quality of the devices output is often questionable if compared with reference measurements techniques and still matter of research[MLA17; Rai+17; Cas+17]. Recent efforts in the development[Alf+20] and performance assessment[Zau+21] of these devices have been made, both at sensor level as well as system of sensors implementation level, thus underlining the importance of calibration for a reliable output. Belosi et al. evaluated the performance of four optical particle counters (OPCs) with a standardised particle generator; they found good results for total particle number concentration, while aerosol size distribution and average particle density must be improved as they relate closely with particle mass concentration [BSP13]. As probably clear at this point, conversion from particle to mass concentration represents a crucial task if a good performance of OPCs is desired. A study from Franken et al. compared different conversion methods with a focus on PM2.5 mass concentrations [Fra+19]. They found that while good correlations (Pearson) are possible with the developed method, other methods resulted in an underestimation of particle mass concentration if compared with gravimetric data.

However, many manufacturers perform only calibration in environmental chambers with controlled process parameters. The latter being a necessary but not sufficient procedure, thus indicating field calibration as a mandatory task, particularly if the acqui-

sition system needs to be relocated. Recent studies have investigated the performance of different calibration approaches, ranging from simple linear regression to machine learning (ML) techniques [Lia21]. The selection of the best method is strongly case specific and must be performed considering experiment setup, sensor selection, site location and measurement duration. An example of field calibration is given by Dinoi et al. [Din+17]. The work reports a comparison between three OPCs against a urban background reference station; showing that effect of relative humidity (RH) is considerable and should be compensated, especially for mass concentration evaluation. On the contrary, Zou et al. investigated the relationship between environmental variables such as air temperature and RH on eight low-cost particle sensor output [ZCM21], but using a controlled chamber and common particle sources. On one hand, they found no significant effect of temperature, on the other hand RH has an impact on magnitude of particle readings but still may be compensated with a simple RH-based calibration as the output correlates well with reference instrumentation.

More generally, If a low cost system of sensors (LCSoS) deployed in a vehicle cabin is considered as mobile, therefore continuously relocated system, the robustness of its field calibration method becomes of paramount importance. Existing assessment techniques rely on relevant factors probability distribution changes that allow performance prediction of field calibration models, but still co-located reference measurements involving several months are required to address site specific issues [De +20].

2.5 Cabin modeling and control optimisation

A superior cabin thermal management is needed to ensure safety, efficiency and performance without compromising passenger's comfort. Performance of HVAC systems are strongly dependent on their control strategy, hence many researchers are making efforts to improve this subsystem. Several control strategies are available today while most advanced techniques are being developed. There are three main hand coded controllers that work with a similar logic, namely: bang-bang, proportional and commercial. Their general control strategy is based on a fan speed v_f control using a function of cabin temperature T_c as feedback, even though more advanced feedback controls are available [Foj+17]. A different approach comes from fuzzy logic controllers, they are still hand coded but use a set of rules to go from multiple inputs (e.g. sensor temperature, interior mass temperature) to multiple outputs (e.g. vent temperature and fan speed). Fuzzy controllers have been shown to perform better than traditional ones because of their ability to represent some unpredictable aspects of thermal comfort, unfortunately they are also computationally expensive and sometimes difficult to design [Bru+18].

State of the art control strategies however are almost all based on machine learning techniques, they are shown to provide the best results if compared with conventional control strategies. One of the reasons for this performance is due to accelerated algorithm training enabled by coupling them with high fidelity CFD-simulation of the vehicle cabin [War+20]. This has to be done from single components design to system operation and control, with the aid of advanced simulation tools [LYE20].

Multi-physics multi-scale simulation software combining detailed computational fluid dynamics (CFD) of the cabin with 1D models of the HVAC system are considered the gold standard nowadays. However, they come with a high degree of complexity and long computation times. Moreover, to build these models, many parameters often owned only by the manufacturer are needed, thus limiting the reproducibility of research in this field. A path of potential improvement could rely on machine learning (ML) techniques to accelerate scientific computing and develop enhanced reduced order models. Such approach is already widely used in numerical simulation of fluid flows [VB21], as well as in a great number of sectors related with the Sustainable Development Goals (SDGs) [Vin+20]. Cabin environments, provided their transient and non uniform nature, come with severe the challenges of understanding nonlinearity and unknown dynamical behaviour, even though the physical laws governing the processes at component level are often well known. This makes difficult to perform tasks as future state prediction, estimation and control [BK19]. For EVs in particular, design and optimisation goals become more demanding as well.

State of the art 1D cabin models consists of several thermal masses corresponding to the main inertial masses that interact between them and the HVAC components by conduction and radiation and with the air by convection. ML-based simulators however, can be accurate and fast enough to provide a solid alternative, opening the way for novel optimisation approaches. To achieve this, a lot of data from measurements and simulations are needed, as more information could be added if enough data is available to train and test the model [Jes+22]. Several research teams are already using combinations of experimental and synthetic data to increase the simulation speed, with applications ranging from weather forecasting to electronics cooling. In physics informed neural networks (PINNs), physical equation are part of the loss function, making them more tolerant to prior assumptions. This improved performance comes at a cost: PINNs are considerably harder to train [Edw22]. Among ML methods, reinforcement learning (RL), a ML approach to artificial intelligence (AI), has recently gained promising results in the solution of complex problems that require intelligence. Extraordinary examples are the capabilities of tools like ChatGPT, DALL-E and DeepLoco which profoundly revolutionised the way we think at AI. Research on thermal management for electric

vehicles is ongoing, with a focus on finding ways to improve efficiency and reduce energy consumption. One area of study is the use of RL to optimise the operation of the vehicle's cooling system. This approach has the potential to improve the overall efficiency of the vehicle, as well as reducing the energy consumption of the cooling system itself. Infact, if described as the task of learning an optimal policy through recursive interactions with a model of an environment, RL has been successfully used in the thermal management field, either on powertrain [Kum20], on the HVAC system [Eng+19] and on control systems [Bru+18]. All using a different approach for cabin modeling depending on the design objective. Another implementation relies on considering human domain rules in a reward function for RL, which are found to be enough to predict optimal setpoint temperature [Rao19] or to perform optimal thermal management in some pioneristic work [PU19; Che21]. As for the comfort models applied to cabin environments, the Nilsson's model seems to provide the best result together with fewer inputs required [Hin+14]. Despite these efforts, very few works include air quality in the ML model as this is still a poorly explored research path.

2.6 Chapter summary

With reflexivity and reproducibility in mind, this chapter starts with a description of the methods used for the literature review. A bibliometric map of the existing knowledge underlines connections between a few key research areas for the thermal management of electric vehicle cabins: thermal comfort, HVAC efficiency and air quality. Powertrain and battery thermal requirements is also a big area of research, but it will be excluded from this study. Key studies are compiled in a comparison table in terms of thermal model, numerical and experimental approach. The review shows that the three clusters have their own methods and modeling approaches, so that a distillation always comes with a compromise. From pure comfort to pure control and optimisation studies the level of detail of comfort model decreases, with detailed local thermal sensation and comfort models making room to lumped parameters and ML models.

The best compromise between complexity and accuracy is still considered to be the equivalent temperature model, mainly because of it flexibility. In fact, it can be as detailed as considering more than 40 body parts, but also requires less parameters than PMV model to predict thermal comfort. As for thermal characterisation, compact low-cost radiant temperature sensors are still far from reality, but some previous studies show that is possible to characterise a vehicle cabin using temperature sensors only. In this study an approach halfway between this and equivalent temperature is taken.

In terms of sensing and IAQ, low-cost sensors constitute a new promising path

of research, where new methods and practices are currently under development. A common approach relies on inside-outside correlation that can be either instantaneous or integral. Calibration of the experimental setup is also an issue and a whole chapter will be devoted to this issue.

In terms of cabin modeling, multi-physics multi-scale simulation software combining detailed computational fluid dynamics (CFD) of the cabin with 1D models of the HVAC system are considered the gold standard nowadays. However, they often require a high number of parameters and long computation times that make them often unsuitable for control and optimisation purposes. A promising and highly regarded solution to this relies on the use of ML and PPO-RL for achieving fast models and frame the thermal management of electric vehicle cabins as an unsupervised optimisation problem.

Chapter 3

The low-cost system of sensor and its performance evaluation

In the context thermal and air quality sensing in vehicles, the thorough diffusion of medium-cost [Ada+19] and low-cost sensors is considered a promising technology for the improvement of spatial and temporal resolution of these measurements. Though, it must be noted that accuracy and data quality of the devices output is often questionable if compared with reference measurements techniques and still matter of research[MLA17; Rai+17; Cas+17].

In this chapter, the costruction and the fast field calibration with reference instrumentation of a low-cost SoS controlled by Arduino hardware is shown in Sections sections 3.1 to 3.3. First, effects of environmental variables on particle-related output variables are discussed. Secondly, the assessment of the performance of the SPS30 particle matter sensor itself, a quite new and powerful device that has still poor literature support is given in Section section 3.4. An overall demonstration of the metrological capabilities of a LCSoS in the automotive field, thus enabling its operation in transient, non-uniform and moving environments such as vehicle cabins is summarised in Section section 3.5.

3.1 Arduino-based sensor system

Two independent measurement systems (IS,ES) controlled by an Arduino Mega 2560 were built for the measurement of environmental parameters [Kon+21; KMP19]. The systems have an on-board Real Time Clock (RTC), a data logger on flash memory, a fan and a TFT display. The RTC clocks of the two systems are constantly synchronised thanks to time data received from the GPS module.

Both systems can measure Particulate Matter (PM) concentration (Sensirion SPS30 sensor), air TVOC concentration (Sensirion SGP30 sensor), air CO_2 concentration (Win-

sen MH-Z19B non-dispersive infrared sensor), concentration formaldehyde (Winsen ZE08 sensor), air temperature, relative humidity and pressure (Bosch Sensortec BME280 sensor), air flow velocity (hot wire analog sensor) and GPS position. The systems are equipped with a fan that conveys air inside the device enclosure, where CO_2 and formaldehyde sensors are mounted, while the SPS30 sensor is equipped with its built-in fan. Both systems independently sampled data at 10-second intervals. All digital sensors used in the measurement device include a microcontroller that implements optimization and self-calibration algorithms. Among all the quantities measured by the low-cost SoS, only those overlapping with the reference instrumentation are considered in this study and reported in table 3.1 together with available daily values from the closest ARPA reference station. The quantities subset can be related to two of the sensors installed in the LCSoS, namely the BME280 and SPS30.

Table 3.1: *Measured quantities and daily summary.*

Variable	**Specifications**		**Summary (Day2,Day3)**			
(Unit)	***Description***	***Sensor***	IS	ES	AS	MSR
$t_a(°C)$	Air temp.	BME280	(21.0,20.8)	(19.5,19.6)	(19.1,18.9)	(18.7,15.8)
$RH(\%)$	Air rel. hum.	BME280	(51,43)	(55,46)	(71,62)	(42,57)
$PM_1(\mu g/m^3)$	PM1 conc.	SPS30	(1,1)	(1,1)	(0,0)	n.d.
$PM_{2.5}(\mu g/m^3)$	PM2.5 conc.	SPS30	(1.5,1.4)	(1.4,1.3)	(1.1,1.1)	(5.2,2.4)
$PM_{10}(\mu g/m^3)$	PM10 conc.	SPS30	(2,1)	(1,1)	(2,3)	(15,6)
$NC1(\#/cm^3)$	Number conc.	SPS30	(11,11)	(10,10)	(6,8)	n.d.
$NC2.5(\#/cm^3)$	Number conc.	SPS30	(11,11)	(10,10)	(6,8)	n.d.
$NC10(\#/cm^3)$	Number conc.	SPS30	(11,11)	(10,10)	(6,8)	n.d.

[a] mv = measured value.

The BME280 is a high linearity and high accuracy air temperature, humidity and pressure sensor. Respectively, its operating range is $-45\,°C$ to $85\,°C$ for temperature, and 0 % to 100 % for humidity. It features an extremely fast response time $\tau_{63\%}$ of 1 s, thus enabling a consistent oversampling if compared with the current application time granularity [Bos21].

The laser scattering-based SPS30 PM sensor allows mass concentration and number concentration sensing. The sensor encapsulates a miniaturised fan and a High Efficiency Particulate Air (HEPA) filter to reduce the optical particle contamination; it also runs its fan at full speed for 10 seconds every 7 days and at startup as an automatic cleaning procedure. Mass concentration measurement range: 0 to 1000 $\mu g/m^3$. As discussed in [Li+18b] and [Try+20], the SPS30 is an optical particle counter (OPC) optimised for

PM2.5 and smaller particle analysis. In fact, Sensirion PM sensors are calibrated using regularly maintained and aligned with high-end reference instruments (e.g., the TSI Optical Particle Sizer Model 3330 or the TSI DustTrak™ DRX 8533) only for 2.5 µm particles size. Moreover, as reported in the sensor specification statement from the producer, PM4 and PM10 outputs are not directly measured but estimated from smaller particle counts using typical aerosol profiles. This behaviour is also confirmed by the SPS30 detection range being identical for 1 µm and above, thus warning about the use of the sensor for bigger particles sensing [Kuu+20].

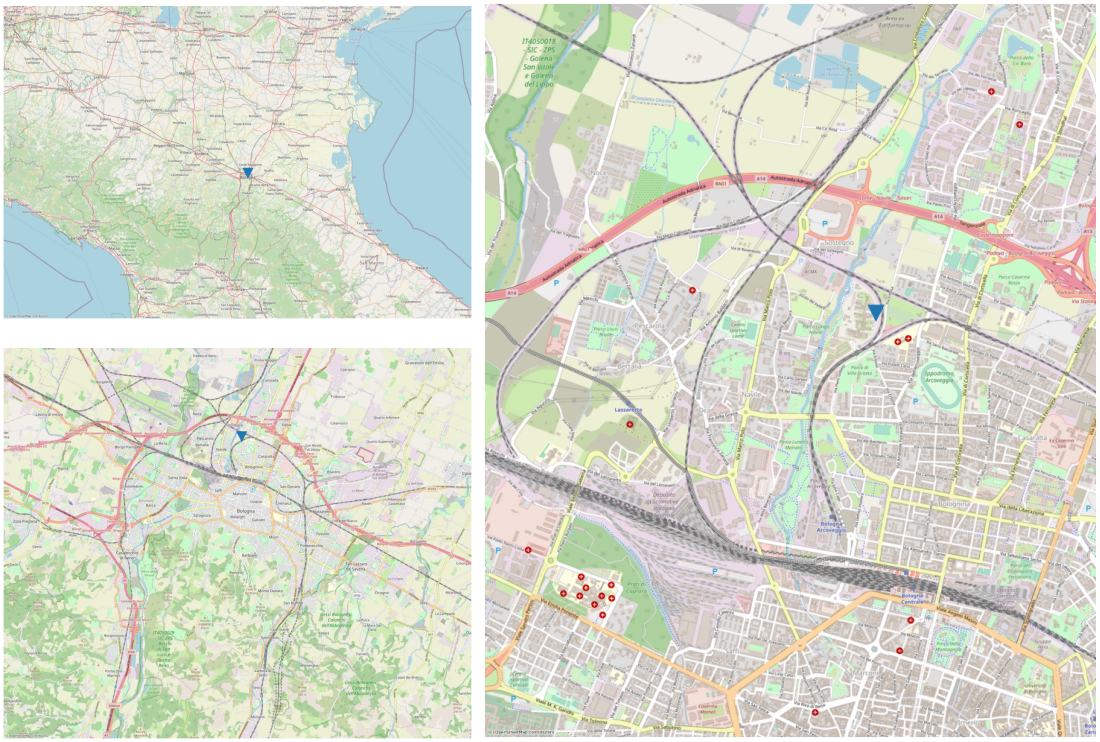


Figure 3.1: *Location of the urban background reference station site in Bologna, northern Italy, decimal degrees (DD): (44.523698,11.340034).*

3.2 Measurement site and reference instrumentation

The reference site is located in northern Italy, inside the research area of CNR in Bologna (figure 3.1). The site is classified as urban background station. Measurements from this instrumentation will be referenced as ARPAE System (AS) in the following sections.

The reference measurements chosen were performed by an OPC FAI (Multichannel Monitor, FAI Instrument - Rome, Italy) which classifies particles in 8 size intervals from 0.28 µg to 10 µg and is equipped with a 10 µg inlet head and operates with a 1 l/min

flow rate.

The measurement principle is the laser scattering: the sensor uses a 35 mW laser diode as the light source and a mirror collecting system elliptical. The light diffused by the particles and collected by the elliptical mirror is concentrated in a photodiode that converts light energy into electric current. The air sample is transferred to the mixing chamber where it is diluted with clean and dehumidified air (free of particles and with low humidity level relative): a smart heater placed in the diluter along the mixing chamber that is automatically operated only when needed. OPC Multichannel Monitor is therefore equipped with a temperature and relative humidity sensor in the external environment, protected from direct solar radiation and from rain, and one inside the instrument to detect temperature and humidity relative to the diluted sampling air that passes through the Laser Sensor. Furthermore, the FAI instrument implements a tool (Zero Test) to verify that the sensor provides "zero" counts in the presence of particle-free air. This test also makes it possible to verify that there are no infiltrations of external air in the dilution circuit (in the first hour 15 test zero minutes are dropped every day).

OPC FAI also provides measurement of estimated PM (PM1, PM2.5, PM10) and is integrated with a Swam Dual Channel (SWAM 5a DC, FAI Rome, Italy) that determines PM2.5 and PM10 daily mass concentration with β-ray attenuation method using a low volume (2.3 m^3/h). The integration gives an automatic correction with real mass values supplied by SWAM DC with self-learning procedure. Conversion from number to PM mass concentration is performed using the algorithms provided by the FAI manufacturer.

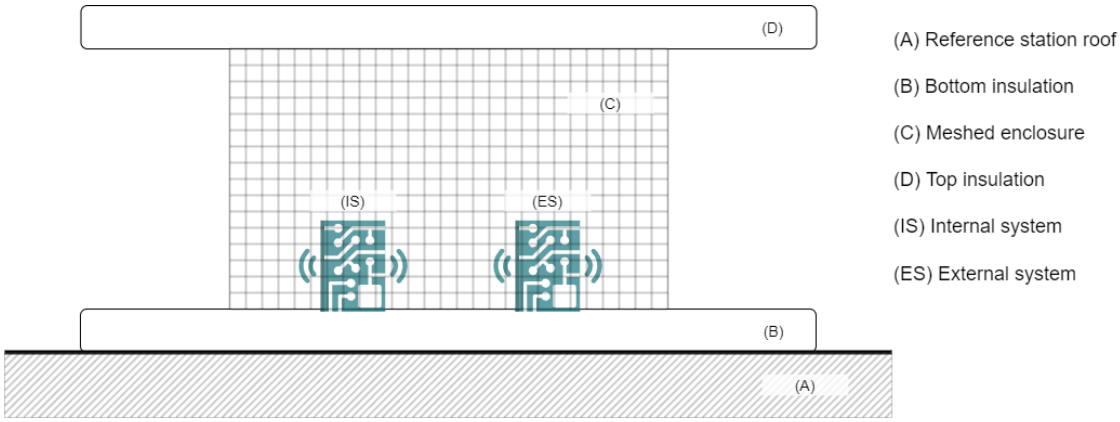


Figure 3.2: *Experimental setup for the LCSoS on the reference station roof, with insulation and meshed enclosure.*

3.3 Measurement campaign and experimental setup

Experimental campaign started 2021-05-18 at 12:00:00 and finished 2021-05-21 at 12:00:00. Total test duration was 3 full days, thus providing enough points for the chosen temporal resolution of 1 h. The latter was defined according to the typical time resolution of the two classes of instruments. Reference instrumentation provides data with a sample rate of 1 min, but verified measures are usually provided as a daily average. On the other hand, low-cost SoS can output data at 10 s sample rate. Given the different time granularity of the two systems, we opted for a 1 h time averaging, which gives a total of 72 samples calculated averaging 4320 minutely samples available (26 000 for the LCSoS). Nevertheless, a 24-h average of data measured during the parallel measurement with the reference method shall always be calculated and reported [ISO19]. Only two central days of the measurement period are suitable for such a comparison given in table 3.1, also because a daily average should be considered valid only if 75 % of the hours are covered; this is not the case for first and last day.

The experimental setup has been built balancing the needs of the experimental campaign with those of the sensors. To ensure the best performance of the SoS we extended the design and assembly guidelines provided by Sensirion for the SPS30 to the whole system. The most important being:

- A good coupling with ambient air and a proper exposition to external conditions
- Avoidance of exposure to direct sunlight or external heat sources

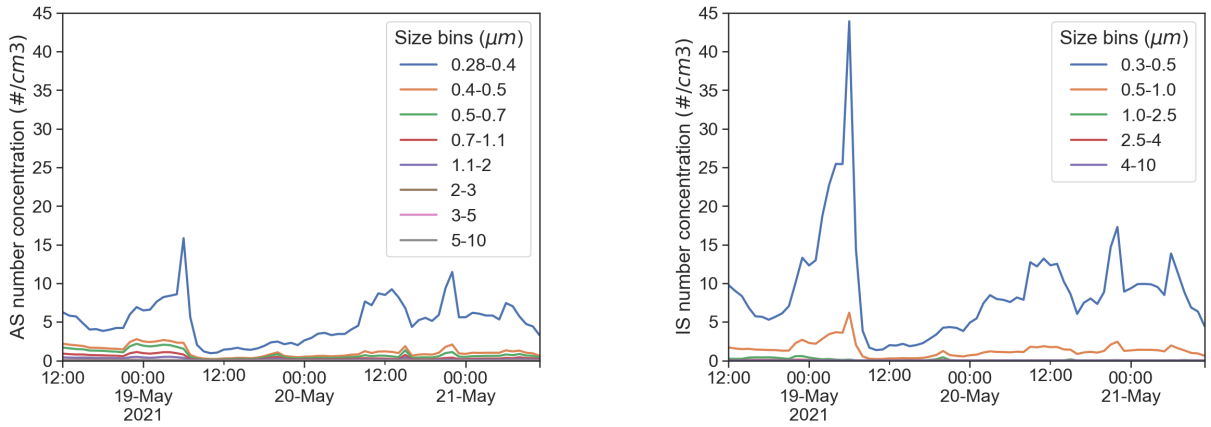
The low-cost SoS was placed near the sampling inlet of the reference station. A 4 cm thick thermal insulating layer has been used at the bottom and top to provide shading and thermal insulation, a pierced plastic shell was used to connect the two planes thus providing support for the top plane but allowing a good air exchange in the sampling volume (figure 4.1).

3.3.1 Methods used to acquire and manipulate the data

The two Arduino Mega 2560-based acquisition systems were programmed using the standard Arduino IDE. The following specific libraries were used for the sensors mentioned in the previous section: sps30.h, DallasTemperature.h, Adafruit SGP30.h, Adafruit BME280.h, DFRobotHCHOSensor.h and TinyGPS++.h. Each acquired data is saved on an SD memory card together with a time reference, synchronised in both systems by the reference clock signal received from GPS by both systems.

Table 3.2: *Particle count buckets grouping.*

AS		IS, ES	
Channel	**size range (µm)**	**Channel**	**size range (µm)**
1	0.28-0.4	1	0.3-0.5
2	0.4-0.5		
3	0.5-0.7	2	0.5-1.0
4	0.7-1.1		
5	1.1-2.0	3	1.0-2.5
6	2.0-3.0		
7	3.0-5.0	4	2.5-4
8	5.0-10	5	4.0-10



((a)) **((b))**

Figure 3.3: *AS row buckets in 8 size channels 3.3(a) and IS row buckets in 5 size channels 3.3(b).*

An Exploratory Data Analysis (EDA) approach was applied to the data-set, thus providing insights to the problem definition and model imposition only after analysing the data [Gut20]. The open source tools chosen for the EDA were python 3.8 and several scientific computing libraries (pandas, matplotlib, numpy and scipy above all). Procedures used are more graphical than quantitative as required by EDA approach.

When comparing multiple particle sensing devices, their builtin size buckets must be considered. Reference instrumentation often comes with a higher size resolution than LCSs, and an accurate grouping of size bins becomes inevitable [Kel+17]. Table 3.2 shows the grouping adopted in this study for EDA and confirmatory data analysis, while figure 3.3 exhibits channell splitting for reference and low-cost instrumentation respectively.

3.4 LCSoSs preformace evaluation

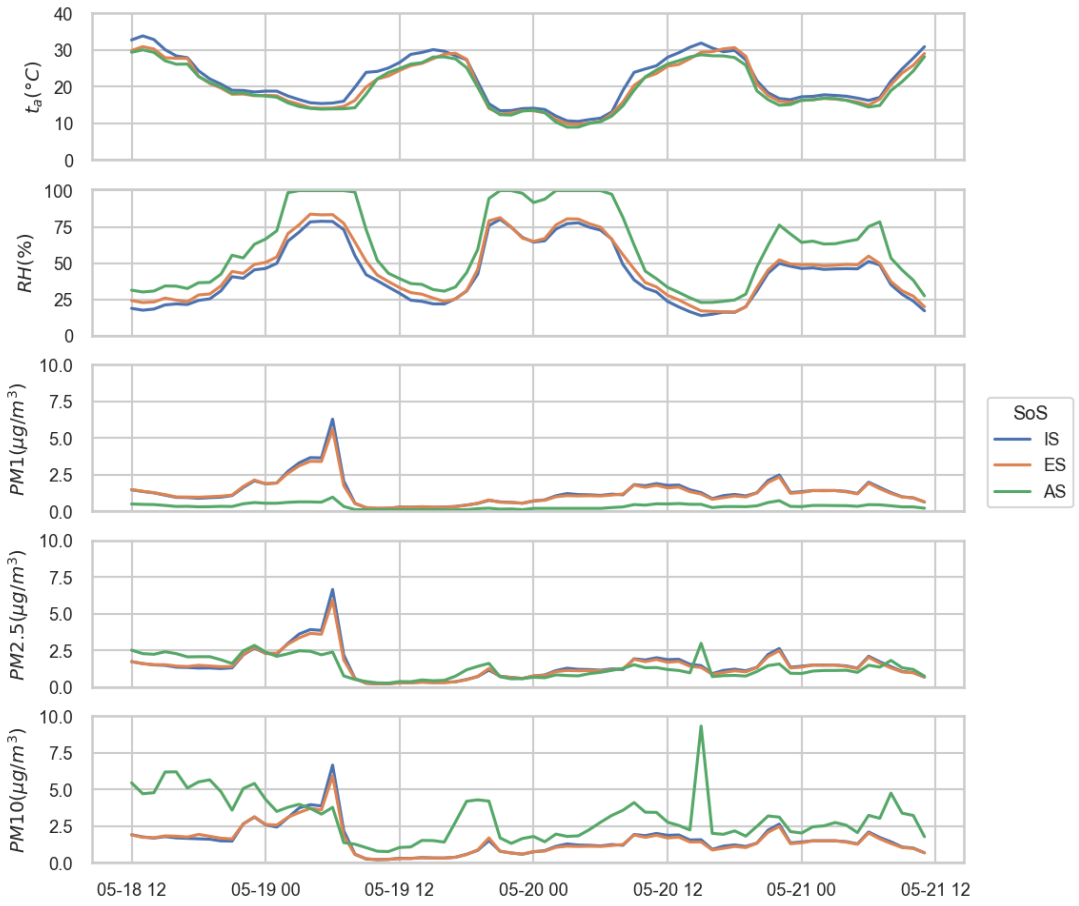


Figure 3.4: *Comparison between hourly data acquired from the two LCS (IS,ES) and from reference instrumentation (AS). Variables of interest are air temperature, relative humidity, PM1, PM2.5, PM10.*

A qualitative overview of the SoS performance is shown in figures 3.4 and 3.6 for the whole test duration. Five variables of interest have been selected considering their overall correlation with particulate matter mass concentration as shown in figure 3.5. Conversely, variables that showed a poor Pearson's correlation coefficient R have been disregarded. The two LCS show an excellent consistency between themselves as already stated in previous studies [Kuu+20]. A good agreement on temperature data is appreciable. Particular attention must be paid to RH as it has a direct impact on sensing mechanism of OPCs, and poor performance at full-scale raises an alert. Regarding PM, correlation seems to decrease ranging from smaller particle size to bigger ones. A quantitative analysis of the SoS performance is given in the following sections, with a focus on effects of time-granularity, relative humidity, mass conversion from particle counts and size detection response. Prior to that, an accent must be put on the small

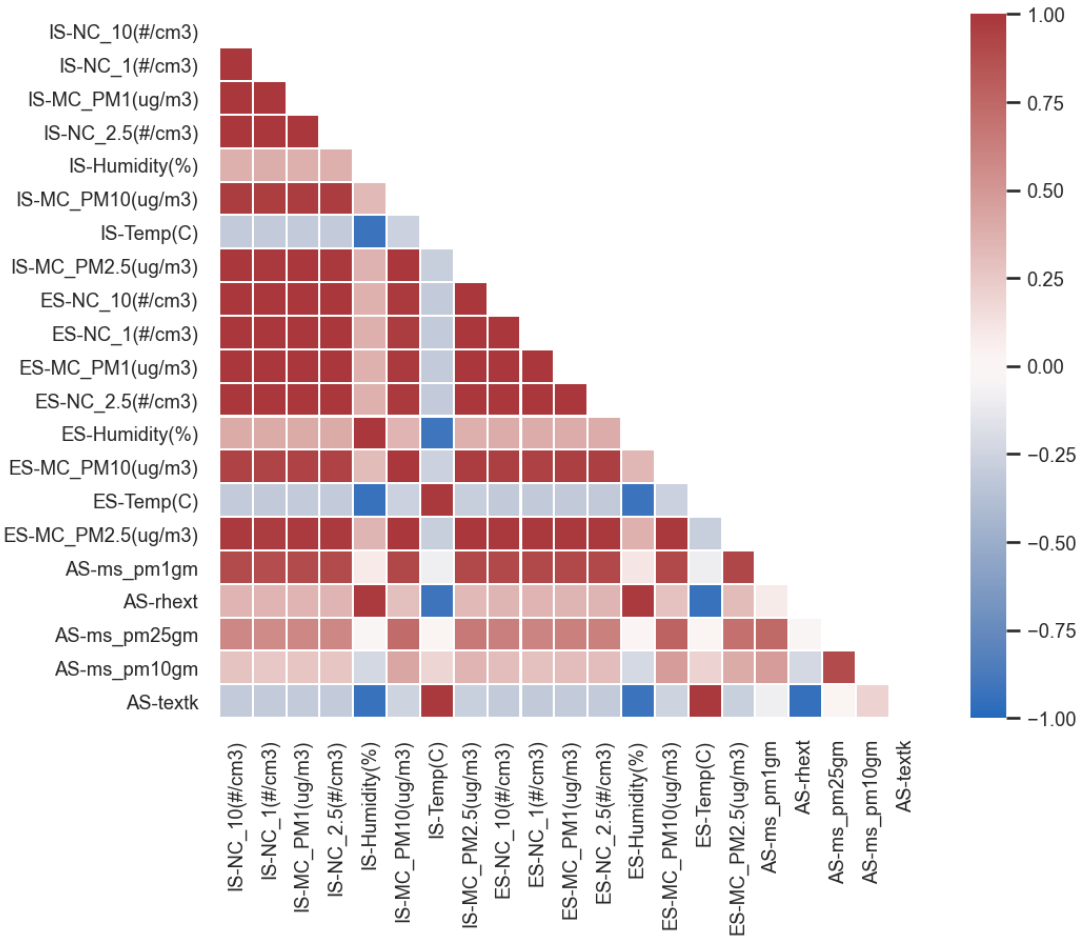


Figure 3.5: *Global correlation heatmap built with Pearson's correlation coefficient, ranging from -1 for negatively correlated variables to +1 for positively correlated ones.*

PM values observed during the test, thus making the instrumentation operate on the low-end of the measurement range.

3.4.1 Time granularity

Choosing the best time granularity for data analysis is not a simple task. Some authors suggest to define it in the EDA phase and before the data analysis is performed; moreover they warn against the use of statistical significance as a selection metric for time granularity as it can lead to misinterpretation of results [Wak+20; Ma+17]. Averaging is often used to decrease autocorrelation in timeseries data, but a coarser time granularity can lead to unacceptable reduction of significance [Col+20]. In this study, the following factors are considered: total test duration, different time scales between SoS and official data, field of application. A compromise is needed between the time scale which is best for automotive application (minutely or less) and the time scale of validated data from ARPAE (daily values with a 6 months validation process).

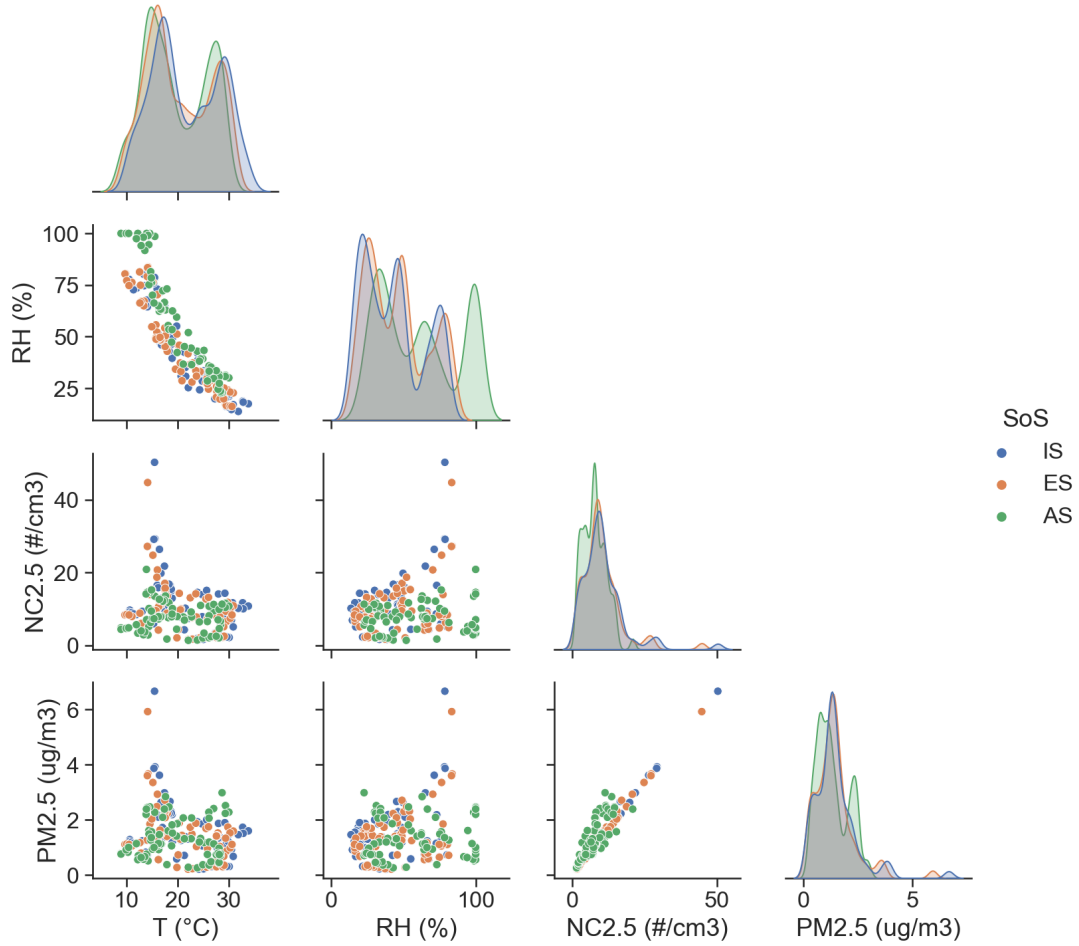


Figure 3.6: *Univariate kernel density estimation plots (diagonal) and bivariate scatter plots (off diagonal) of t_a, RH, NC2.5 and PM2.5 distributions.*

A common approach, which we also adopted, for ambient validation of PM data is to perform the analysis on hourly averaged results [Kel+17].

3.4.2 RH sensitivity analysis

It is well known that among the factors influencing PM measurements, RH plays a major role [ZCM21]. This is true to the point that reference gravimetric methods rely on a controlled sampling process in terms of temperature an relative humidity of the air sample across the filter. These methods are not applicable for LCS and the definition of a RH correction factor is quite common [Lia21]. In order to evaluate the influence of RH on the sensor response, we can look at figures 3.7 and 3.8, which show scatter and box plots of number and mass concentration for 2.5 µm particle size. RH values have been aggregated in four uniformly spaced bins, thus providing insights on statistical dispersion in each RH bin. Despite being in agreement among themselves, LCS, IS and

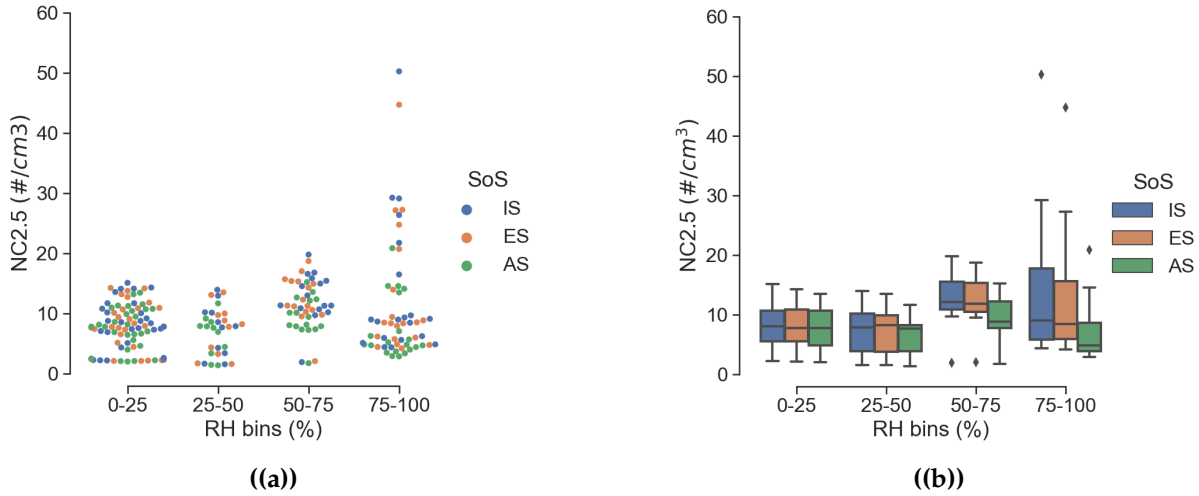


Figure 3.7: *Effect of RH on* 2.5 µm *particle count values with data aggregated in 4 uniformly spaced RH bins.*

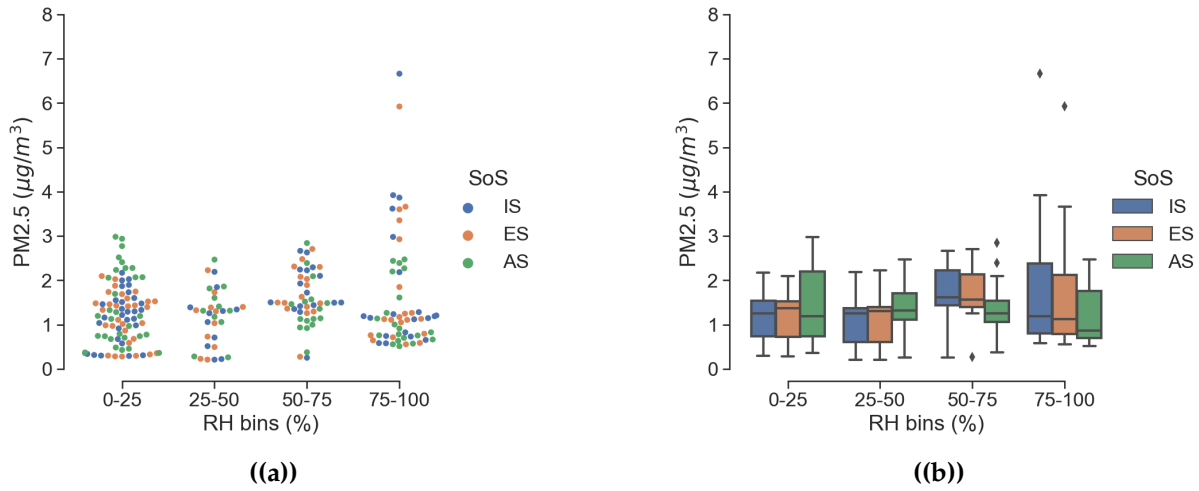


Figure 3.8: *Effect of RH on PM2.5 values with data aggregated in 4 uniformly spaced RH bins.*

ES have a wider inter quartile range (IQR) toghether with a greater number of outliers; the worst case scenario corresponds to values of RH above 75 %. Similar consideration can be drawn from figure 3.9, which reports the same quantities but without binning. It is noticeable an increased dispersion of the data with growing RH values; also in this case higher particle counts associated high RH values do not fit well with the reference system AS. It should be noted that the dispersion of data points increases in the transition from particle count to particle mass, probably due to the theoretical mass conversion adopted in OPCs. Further results dealing with mass conversion will be presented in the following section.

In order to quantify the impact of the predictor variable RH on the response variable

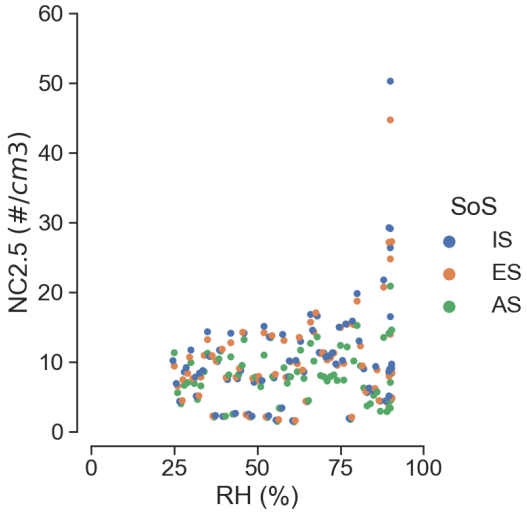


Figure 3.9: *Scatter plot of* 2.5 µm *particle count values without RH binning.*

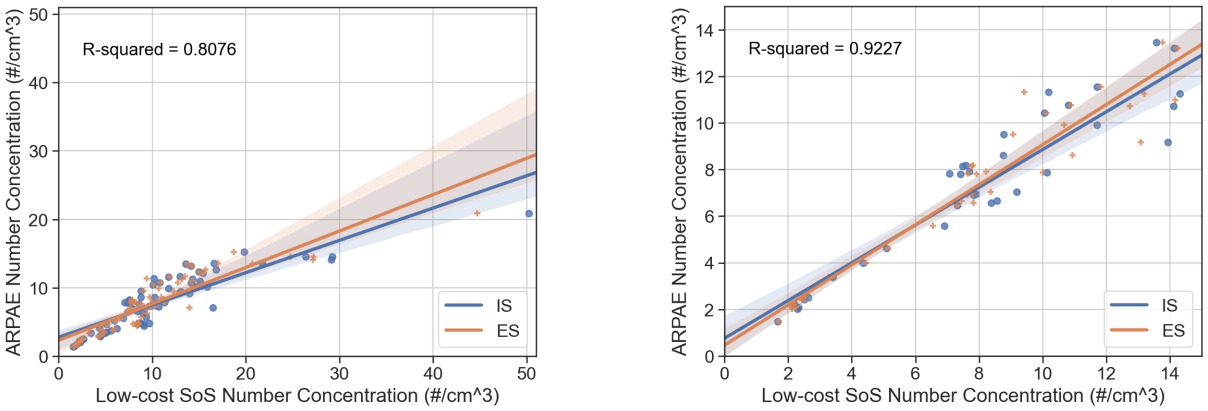


Figure 3.10: *Linear regression for* 2.5 µm *particle size using the full data-set in terms of RH 0-100% or a reduced data-set with RH 0-50% (less data points).*

NC2.5, an univariate linear regression has been adopted. Results from the ordinary least squares (OLS) analysis are reported in figure 3.10, showing again that the sensor performance degrades at high RH values. The improvement in R^2 approaches 0.11 when values of RH above 50 % are dropped from the data-set.

3.4.3 Mass conversion correction

Before delving into mass conversion results, it is worth recalling how mass concentration can be calculated from particle counts, together with associated challenges. Mass conversion adopted in OPC is based on the assumption that particle are spherical and with common density in each size bin [Fra+19]; in practice this is made according to the following equation 3.1:

$$PM_i = 10^{-9} \cdot \rho_{p,i} \cdot \frac{\pi}{6} \cdot \tilde{d}_i^3 \cdot NC_i \tag{3.1}$$

where i is the particle size bin, PM is the mass concentration in $\mu g/m^3$, $\rho_{p,i}$ is the particle average density, $\tilde{d}$ is the median particle diameter in nm and NC is number concentration for a given size bin. It is immediately clear from equation 3.1, that a good estimation of $\rho_{p,i}$ as input parameter for the conversion algorithm is crucial. A widely adopted value is $1.65\,g/cm^3$ [Wei+04], but its variability in time and space must be considered carefully after relocation. Mass conversion based on equation 3.1 requires $\tilde{d}$ to be known in each size bin; an information not always available for LCSs. Sensirion SPS30 provides a typical particle size output d_{typ} that is correlated with weighted average of the number concentration bins measured with a reference particle sizer. Substituting d_{typ} in equation 3.1 leads to the calculation of a typical particle mass, which can be used to perform a manual conversion whose results are displayed in Figure 3.11.

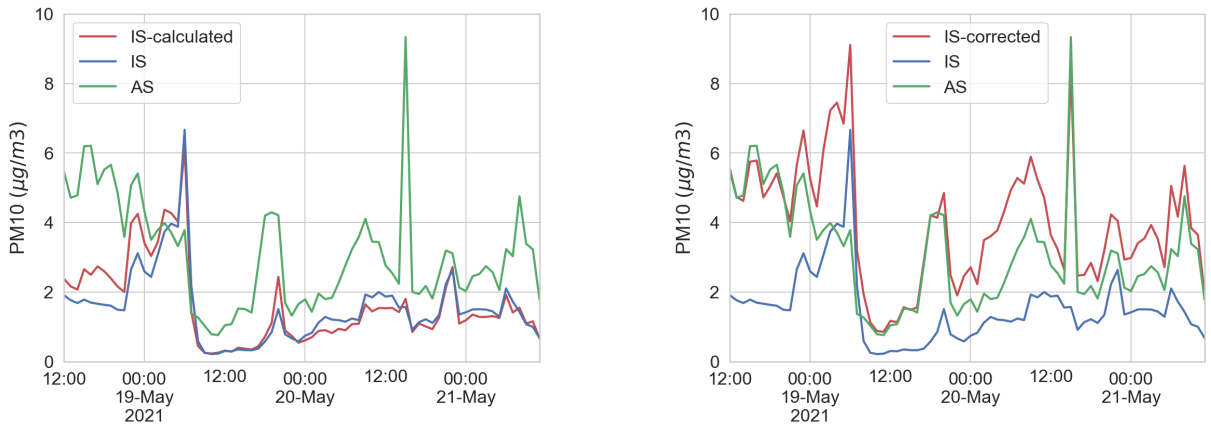


Figure 3.11: *Particle count to mass concentration conversion. 3.11(a) Calculated from particle count according to equation 3.1; 3.11(b) corrected according to reference station average particle mass*

Pearson correlation between LCS and AS increases from 0.43 to 0.60 in this case. An alternative approach proposed in this study relies instead on the calculation of a constant ratio $\frac{PM_i}{NC_i} = m_{p,i,AS}$ between mass and particle counts in the chosen bin from the reference instrumentation data. The defined quantity can be seen as an average particle mass used to calculate a corrected mass concentration from the number concentration of the LCS. The latter being a direct and more reliable measurement, as shown in the previous section. Figure 3.11, contains two plots that clarify the correction procedure. With this method, the Pearson correlation of PM values between LCS and AS increases from 0.43 to 0.81, a value again comparable with the correlation between particle number

concentration. This approach, if repeated periodically and automated, could lead to an increase in the accuracy of LCS using data from fixed stations[Ren+21].

3.5 Chapter summary

In this chapter, we showed the development of a low-cost system of sensors capable of a high spatiotemporal resolution and evaluated its performance in terms of particle concentration measurement through a field comparison against reference instrumentation operated by ARPAE. Main factors influencing particle sensors output were first measured and then identified with an EDA approach; their impact in terms of data quality and correlation with reference instrumentation was also analyzed. The LCSoS is constituted by an Arduino open-hardware platform and commercially available LCSs. Open-source software tools were used for acquisition and post-processing algorithms, thus further supporting the reproducibility of the experiment and spreadability of the device. Results show that the system has a relatively good performance with respect to air temperature, relative humidity and smaller particle number concentration ($d_p \leq 2.5\,\mu m$) with coefficient of determination approaching $R^2 = 0.92$ for the reduced data set, as also confirmed in previous studies [Kel+17]; RH has a strong impact on performance indicators and correction techniques are strongly recommended [De +20; ZCM21]. If few works were found to deal with Sensirion SPS30, even fewer were dealing with mass conversion [Fra+19; ZHF17]. In the last part of this work we compared three conversion approaches: the sensor builtin output, a mass concentration calculated from particle concentration and a mass concentration corrected according to reference instrumentation data. Both techniques provide an improvement of the overall Pearson R ranging from 0.2 to 0.4 on the PM10 value. It is also important to underline that PM correlation with daily values from the verified ARPAE network are quite poor; on the other hand, a good quality and consistency of the data among LCSoS was observed. An IAQ monitoring system like the one proposed in this research can be subject to frequent relocation (moving measurements), this can lead to calibration issues that must be addressed. More sophisticated calibration models (ML) are supposed to provide more effective calibration procedures, but require longer training [Zau+21; Lia21] and might result in unpractical methods for automotive and high spatiotemporal resolution applications. For this reason we opted for a short measurement period of three days and well established yet simple performance assessment procedures. Thus making a step towards addressing the lack of quantitative specification, formulate requirements for mobile applications and conditions of intended use of these devices and their output [Mor+18a]. Further research on these aspects must be carried on, as useful insight are

possible from LCSs, but their reliability is still far from reference instrumentation. In the following chapters, measurements from the LCSoSs are used first to characterise the vehicle cabin [Rus+21; Rus+22], and then to build a model suitable for RL.

Chapter 4

Characterisation of the vehicle cabin

This chapter describes a series of experiments on air quality and energy efficiency inside the passenger compartment of an electric vehicle. The measurements were carried out using the two portable LCSoS described in previous chapter and reading the car's On-Board Diagnostic bus (OBD) [Rus+21]. The correlation between experimentally measured air quality data and the energy spent by the HVAC system inside the vehicle cabin is investigated. Concentrations of some pollutants in the vehicle cabin are measured by means of a low-cost Arduino sensor-based system. The use of an open-source electronic platform like Arduino allowed fast prototyping and simplified design of the system. In addition, it helped to relax the constraint involved in the construction of hardware and software platforms for data acquisition, following a path that has been outlined by many authors in literature [Kon+21; KMW18; KMP19]. The first hypothesis being tested in this chapter is whether LCSoSs have the required spatiotemporal resolution, accuracy and responsiveness to characterise the cabin when deployed in a vehicle rather that in open-air. The second hypothesis deals with the HVAC settings that affect the most comfort, {IAQ and energy use parameters; together with their improvement potential.

A brief description of the experimental setup, the positioning and the details of the low-cost system of sensors is given in Sections 4.1 and 4.2. Section 4.3 contains results where HVAC system configuration fresh-air and recirculation mode of the intake air are varied, while PM2.5 and Volatile Organic Compounds (VOC) concentrations are measured. Gaps and limitation of the vehicle HVAC control system are underlined. HVAC air filter performance is also evaluated by making cabin air quality measurements with and without the filter installed. The relation between consumed energy, HVAC system settings and pollutant concentrations is obtained, with the aim of provinding a data basis to the concurrent optimisation of comfort, air quality and energy use in the next chapter. The methodology, summarised in Section 4.4 is applied to a Nissan Leaf Acenta 40 kWh MY2018 in this case, but it is generally applicable to other EVs

with minor modifications. To show that the use of these sensors for the control of the cabin can enable energy savings together with air quality and comfort gains, specially if appropriately optimised.

4.1 Description of the experimental setup

To characterise the thermal profile and air quality inside the vehicle cabin, two tests have been performed in outdoor parking conditions in the faculty parking lot with the front of the car oriented south.

The temperature distribution within the vehicle cabin is obtained with a grid of 18 DS18B20 temperature sensors following the approach used in [DM19; DMS15] to develop a thermal model of a BEV cabin for energy consumption predictions [Kor+20]. The cabin has been ideally divided into three slices horizontally: namely the top, middle and bottom levels, as shown in Figure 4.1 from a lateral view.

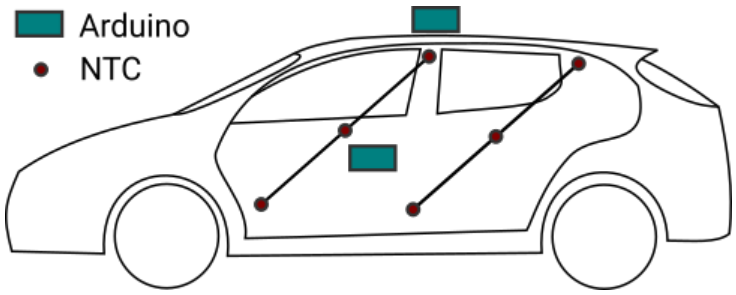


Figure 4.1: *Lateral view of the experimental setup.*

On the grid, six sensors are placed for each plane: three in the front side and three in the back side of the cabin. In addition, air quality related quantities have been measured with low-cost sensors on a unique location in the cabin; near the gear shift knob together with the acquisition system.

As external conditions can strongly affect the internal micro-climate [ISO19], a second acquisition system has been placed on the car roof. This is identical to the internal one, except for the presence of a single temperature sensor only. The presence of the second acquisition system is needed to characterise the environment outside the vehicle and to facilitate inside/outside comparisons with data having the same structure and same metrologic fingerprint.

The approach used in the study, conversely from the one used in ISO standards regarding the interior air of road vehicles [ISO12], does not rely on a vehicle test chamber. The latter is well-documented and reliable, but not suitable for real-time operation and low-cost equipment.

To have clearer insights on HVAC capabilities, an on-board diagnostic (OBD) Linux platform was cleverly installed inside the car to directly retrieve and collect different variables from the electronic control units [Agu21]. Specifically, it was the iWave OBD-II: a little device with an ARM Cortex-A7 processor embedded that runs a light Yocto Poky Linux distribution. The iWave OBD-II can upload data via a 4G/LTE CAT4/CAT1 sim modem, geolocate the device with a GPS receiver and it can transmit messages with the Bluetooth Low Energy 4.2 module. Communicating via the OBD-II interface, the board reads the HVAC power consumption, the power used by the auxiliary equipment (e.g., lights, infotainment, rear defroster etc.), and the power used by the heater. A fine-time granularity monitoring of those parameters was necessary to correctly interpret how the cabin air changes throughout the experiment.

The overall measured quantities are: air temperature t_a, relative humidity RH, air pressure p_a, TVOC concentration C_{TVOC} and $PM_{2.5}$ concentration C_{PM}. The temperature is measured in 18 points as described above, while the other measurements are taken in one point. The same quantities are measured also outside the cabin. Finally, the power usage of the HVAC system is also logged. Table 4.1 lists the measured quantities and the correspondent accuracy.

Table 4.1: *Measured quantities.*

Variable		**Sensors Specifications**		
(Unit)	**Description**	**Manufacturer**	**Model**	**Accuracy (offset + gain)**
$t_a(°C)$	Air temp.	Maxim Integrated	DS18B20	$\pm$(0.5 °C + 1% mv [a])
$RH(\%)$	Air rel. hum.	Bosch	BME280	$\pm(3\%RH + 1\%RH)$
p_a(hPa)	Air pres.	Bosch	BME280	$\pm(1.5\,\text{hPa} + 0.12\,\text{hPa})$
$TVOC$(ppb)	TVOC conc.	Sensirion	SGP30	$\pm$(15% mv)
PM ($\mu g/m^3$)	PM2.5 conc.	Sensirion	SPS30	$\pm$(10 $\mu g/m^3$ + 10% mv)
P_i(kW)	Subsystem i power usage	iWave	OBD/Linux	$\pm(250\,\text{W})$

[a] mv = measured value.

Sensor performance is a device dependent issue that can be measured with various qualifiers [CW14] and ideally addressed individually. In this study, the same approach for all the measured quantities was used. A sampling time of $T_s = 10$ s has been adopted. The raw data from the acquisition system has been filtered with a moving mean over a one minute period, this leads to a six-point moving mean. Subsequently, the filtered data has been converted into time-stamped data in tabular form, and eventually re-sampled and synchronised among the three acquisition systems. The data analysis process has been performed using open-source tools, including Python 3.8 and several scientific

computing libraries (pandas, matplotlib, numpy and scipy above all) following the exploratory data analysis (EDA) approach provided in [Gut20].

4.2 Description of the Arduino-based system of sensors

Two independent measurement systems based on Arduino Mega 2560 were built for the measurement of environmental parameters. The systems have an on-board real time clock (RTC), a data logger on flash memory, a fan and a TFT display. The RTC clocks of the two systems are constantly adjusted thanks to time data received from the GPS module. This operation is implemented to facilitate the synchronization of signals from the three acquisition systems.

The internal system is capable of measuring temperatures at 18 locations in the cabin (Maxim Integrated DS18B20 probes). Moreover, the internal system can measure particulate air matter (PM) concentration (Sensirion SPS30 sensor), air TVOC concentration (Sensirion SGP30 sensor), air CO_2 concentration (Winsen MH-Z19B non-dispersive infrared sensor), concentration formaldehyde (Winsen ZE08 sensor), air temperature, relative humidity and pressure (Bosch BME280 sensor), air flow velocity (hot wire analog sensor) and GPS position, at a unique location. The external system, albeit sharing the same characteristics and using the same sensors, it lacks the 18-spots temperature measurement, the GPS receiver and the air flow velocity sensor.

Both systems are equipped with a fan that conveys air inside the device enclosure, where CO_2 and formaldehyde sensors are mounted, while the SPS30 sensor is equipped with its built-in fan. Both systems independently sampled data at 10-s intervals. All digital sensors used in the measurement device include a microcontroller that implements optimisation and self-calibration algorithms.

High-precision, easy-to-use DS18B20 sensors were used to measure temperatures inside the cabin in 18 distinct positions; they have a typical accuracy of $\pm 0.5\,^{\circ}\mathrm{C}$ from $-10\,^{\circ}\mathrm{C}$ to $85\,^{\circ}\mathrm{C}$ and digitally transmit temperature data on a 1-Wire® bus. The use of 1-Wire protocol [Max], together with the unique 64-bit serial code allows many sensors on the same bus, thus reducing the cable length and allowing to uniquely associate a sensor output with its position in the network through a serial-position coupling. Specifically, the DS18B20 actual temperature is provided by a 12-bit analog to digital converter built-in in the digital sensor, with a fine temperature resolution up to $0.0625\,^{\circ}\mathrm{C}$. Its operating range is between $-55\,^{\circ}\mathrm{C}$ to $125\,^{\circ}\mathrm{C}$.

The BME280 is a high linearity and high accuracy combined temperature, humidity and pressure digital sensor. Its pressure sensing mechanism is resistive, with an operation range of 300 hPa to 1100 hPa, the temperature sensing principle is of the

type diode-voltage with a measurement range of $-45\,^{\circ}\mathrm{C}$ to $85\,^{\circ}\mathrm{C}$, the measurement principle behind humidity is capacitive and its range is 0% to 100% [Bos21]. It features an extremely fast response time $\tau_{63\%}$ of 1 s, thus enabling a consistent oversampling if compared with the current application time granularity of 1 min.

The sensing principle of SPS30 PM sensor is based on laser-scattering, and allows mass concentration and number concentration sensing for particle sizes ranging from 1 µm to 10 µm. As discussed in [Li+18b; Try+20], the SPS30 is an optical particle counter (OPC) optimised for PM2.5 and smaller particle analysis. Sensirion PM sensors are indeed calibrated using regularly maintained and aligned high-end reference instruments (e.g., the TSI Optical Particle Sizer Model 3330 or the TSI DustTrak™ DRX 8533) only for 2.5 µm particles size. Moreover, as stated in the sensor specification sheet from the manufacturer, PM4 and PM10 outputs are not directly measured but estimated from smaller particle counts using typical aerosol profiles. A miniaturized fan and a high efficiency particulate air (HEPA) filter are included to reduce the optical part contamination; it also runs its fan at full speed for 10 s every seven days as an automatic cleaning procedure. The mass concentration measurement range is $0\,\mu g/m^3$ to $1000\,\mu g/m^3$.

The SGP30 TVOC sensor is a digital "multi-pixel" gas sensor. It uses multiple sensors, housed on a single metal-oxide gas sensor chip, placed on a thermally controlled hotplate. Digital data output from the sensor includes raw measurements of ethanol and H_2, and calculated values of TVOC and equivalent CO_2 via internal algorithm, such as automatic baseline compensation of the measurement [RHB18]. The TVOC data range from this sensor is between 0 to 60,000 ppb. This sensor's equivalent CO_2 were disregarded due to its low sensitivity to external pollutants and due to the absence of passengers in parking conditions.

The measurement system has been characterised both in winter and in summer conditions. In the following sections, two typical conditions for winter and summer have been chosen in order to characterise the HVAC system performance in heating and cooling operations, respectively. Winter tests have been performed on 29 January 2021, while summer test have been carried out from 14 July to 15 July 2021.

4.3 Measurements in heating and cooling operation

In this section, the combined measurements of air quality and comfort parameters, together with the energy consumption by the electric car are shown and discussed in different seasons, in order to show the differences in relation to the operational mode for the air conditioning. Moreover, an estimation of the filtration performances of the HVAC system is given, by comparing the results corresponding to new and used filters.

Ultimately, a detailed treatment of high spatial resolution cabin air temperature profiles is provided in the supplementary materials.

4.3.1 Heating operation

Two different test conditions have been investigated for the winter, starting from a state of equilibrium with the external environment, obtained maintaining all systems off and all doors opened for 15 min. Once the equilibrium was reached, the proper test was performed while maintaining the heater on for one hour, and the set-point temperature at its maximum of 30 °C, the fan speed was at its maximum (position 7), and all of the windows and all the doors were closed. During the first test, the recirculation system was off (meaning that the air ventilation system was in the fresh-air configuration), while during the second test the recirculation system was on instead.

Fresh-air mode

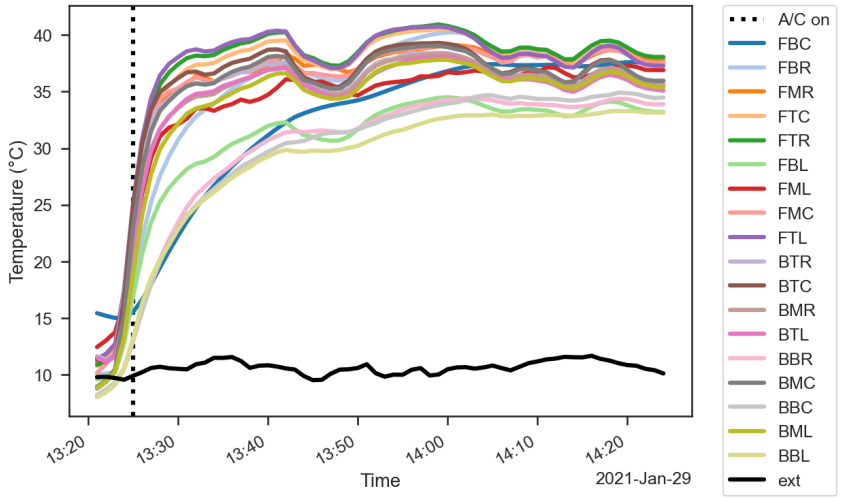


Figure 4.2: *Heating temperature profiles, fresh-air mode.*

All the experiments confirm that the heating system is capable of reaching a quasi-steady state condition in about 20 min, even though the temperature reached by the air inside the cabin is well over the set-point temperature. Figure 4.2 shows the readings from all the 18 temperature sensors in the cabin for the case with fresh-air mode with a 3 letters naming scheme [Rus+21]:

1. The first letter tells us if we are in the front (F) or in the back (B) of the cabin.
2. The second letter refers to the location in vertical direction, namely bottom (B), middle (M) and top (T) plane.

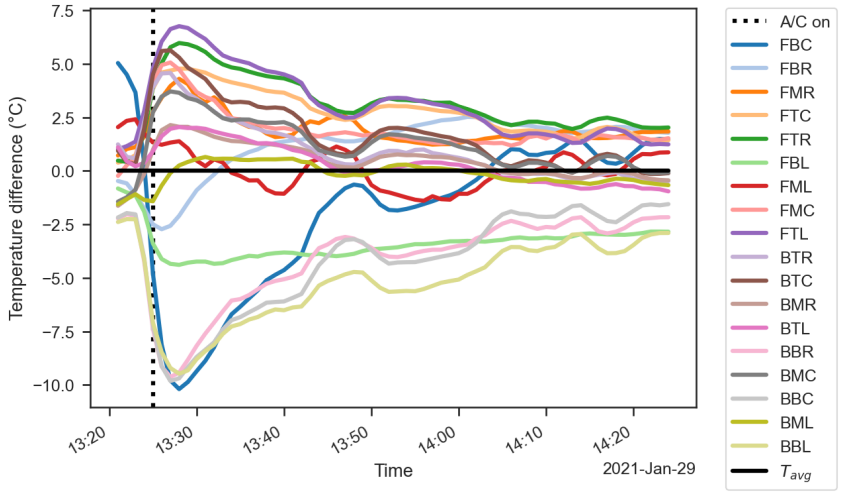


Figure 4.3: *Heating temperature difference profiles, fresh-air mode.*

3. The third letter refers to the location in the horizontal direction, namely left (L) or driver, center (C) or right (R) side.

Colder spots can be found at feet level of the back seats, moreover this is in disagreement with the studies of Nilsson on equivalent temperature of body segments [Nil04]. In other words, analysing the results of the work that led to the development of the only available standard for comfort evaluation in vehicles (ISO-14505), we can infer that occupants will be likely to accept colder temperatures in upper body parts during winter, while here we are in the opposite situation. Data from position FBC have a different behaviour because it refers to the sensor installed inside the acquisition system box, thus suffering from thermal inertia issues.

Defining a temperature difference $\Delta t = t_i - t_{avg}$; where t_i is the temperature in one generic position of the grid, and t_{avg} the mean value of the 18 temperature readings for each timestamp; is possible to obtain Figure 4.3. It is worth noting that air temperature inside the cabin can reach discrepancies of more than 15 °C in the first minutes of operation, while the Δt values at the end of test is lower and the temperature distribution much more uniform.

The temperature measured inside and outside the cabin keeping the fresh-air mode on is shown in Figure 4.4a. The air temperature inside the cabin is obtained by the average of the air temperature measurements on the sensors placed on the grid shown in Figure 4.1, i.e., $t_{int} = t_{avg}$. The TVOC concentration measured inside and outside the cabin in fresh-air mode are shown in Figure 4.4b. It is shown that the TVOC concentration increases while the HVAC system is working, even in the fresh-air mode. This effect is related to the presence of sources of VOC inside the vehicle cabin and the build up phenomena during the HVAC operation, as expected from the literature [ISO12].

Figure 4.4c displays the PM2.5 concentration measured inside and outside the cabin in fresh-air mode. The PM concentrations decrease while the HVAC system is working due to the filtering activity of the HVAC filter. This result ties with what is found in the literature [Heo+19]. For the case analysed, a filtration efficiency of about $\eta = 0.5$ for PM2.5 is ascertained. The power usage of the HVAC system, with the contributions of power used by auxiliary equipment, A/C system and PTC heater recorded by the OBD system is shown in the stacked line plot in Figure 4.4d.

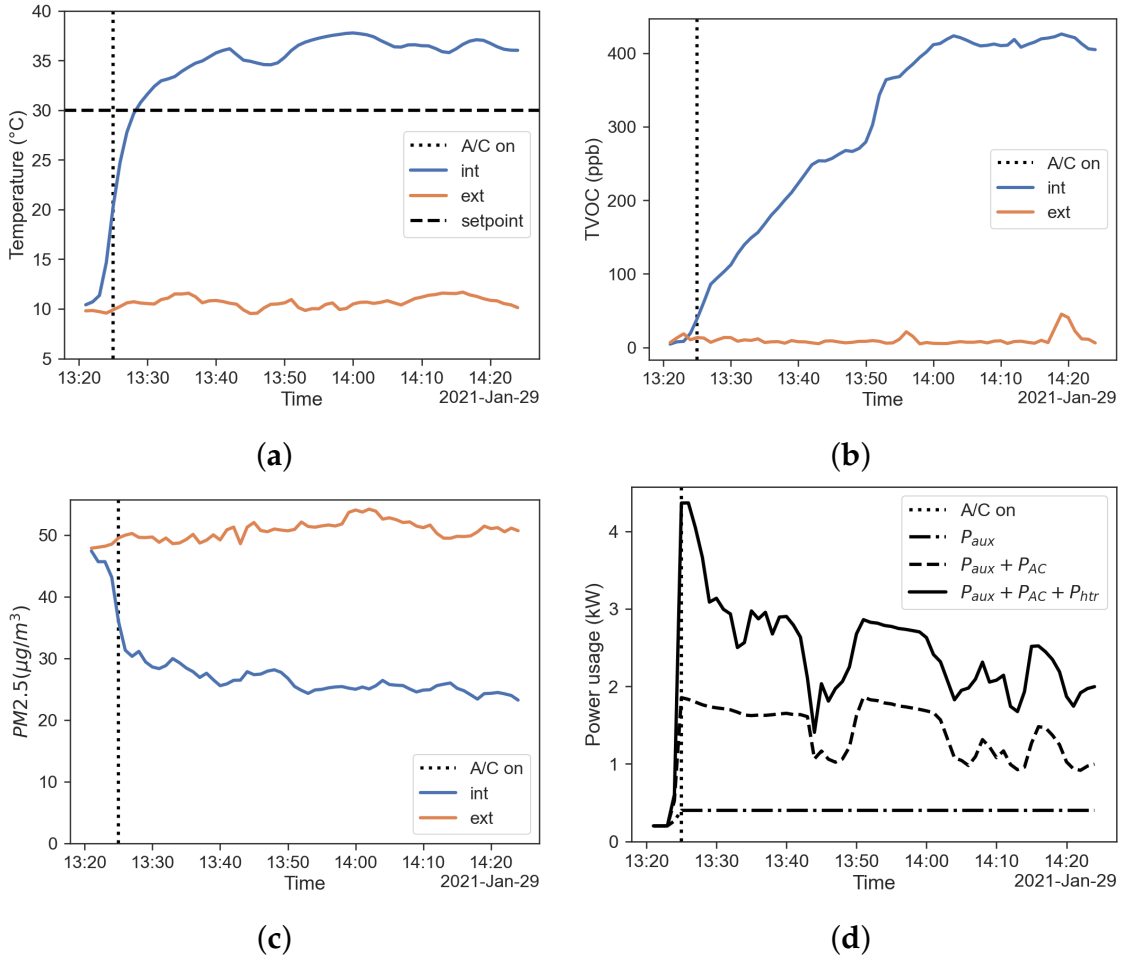


Figure 4.4: *Results regarding the fresh-air mode. (**a**) Temperature inside (blue) and outside (red) the cabin; (**b**) TVOC concentration inside (blue) and outside (red) the cabin; (**c**) PM2.5 concentration inside (blue) and outside (red) the cabin; (**d**) power usage of the HVAC system.*

Recirculation mode

Figure 4.5 shows the readings from all the 18 temperature sensors in the cabin for the case with recirculation mode. Air temperature inside the cabin shows slightly higher discrepancies with respect to the previous case, colder spots can be found again at feet

level of the back seats. Signal from the FBC sensor shows similar responsiveness issues as for the case without recirculation. Higher temperature gradients in the startup phase are evident from Figure 4.6, while a smoother profile is reached at the end of test.

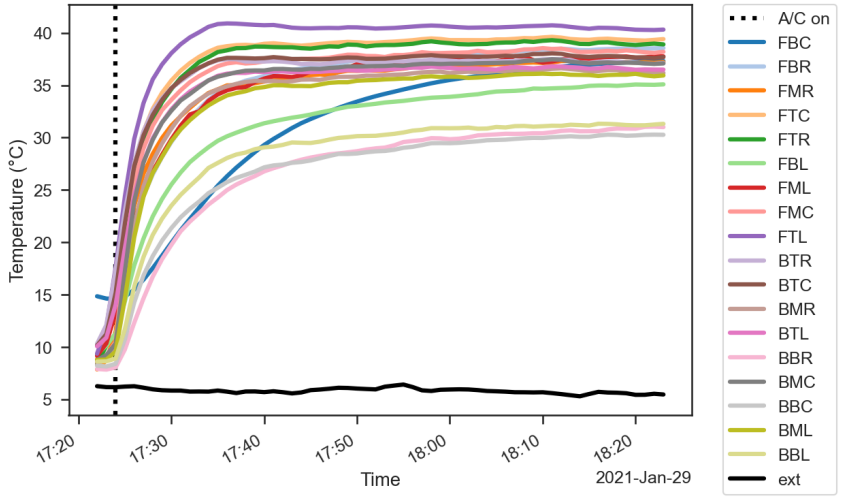


Figure 4.5: *Heating temperature profiles, recirculation mode.*

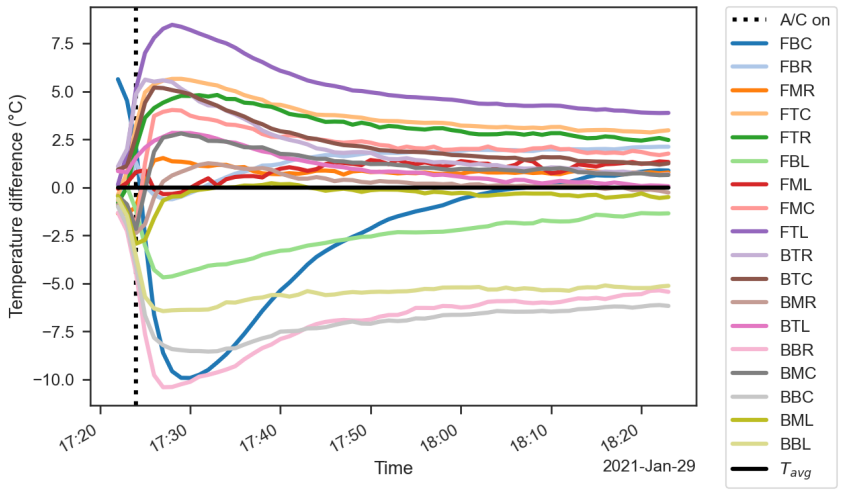


Figure 4.6: *Heating temperature difference profiles, recirculation mode.*

Similar considerations can be made for the case with recirculation activated. From the results shown in Figure 4.7 it is clear that the time to steady state is close to 20 min; again the over-temperature issue remains significant. TVOC concentration reached a value similar to the case without recirculation, but in a longer time with respect to the fresh-air mode. A possible explanation for this behaviour could rely on the fact the source of VOCs inside the cabin is compensated by an improved adsorption performance, as observed by [Heo+19]. The PM concentrations decrease to lower values with respect to the fresh-air mode, as shown in Figure 4.7c. This result shows that the

filtration performance is improved by the recirculation mode. Figure 4.7d shows the power usage of the HVAC system. The figure shows the contributions of auxiliary equipment, the A/C system and the PTC heater to the overall power usage.

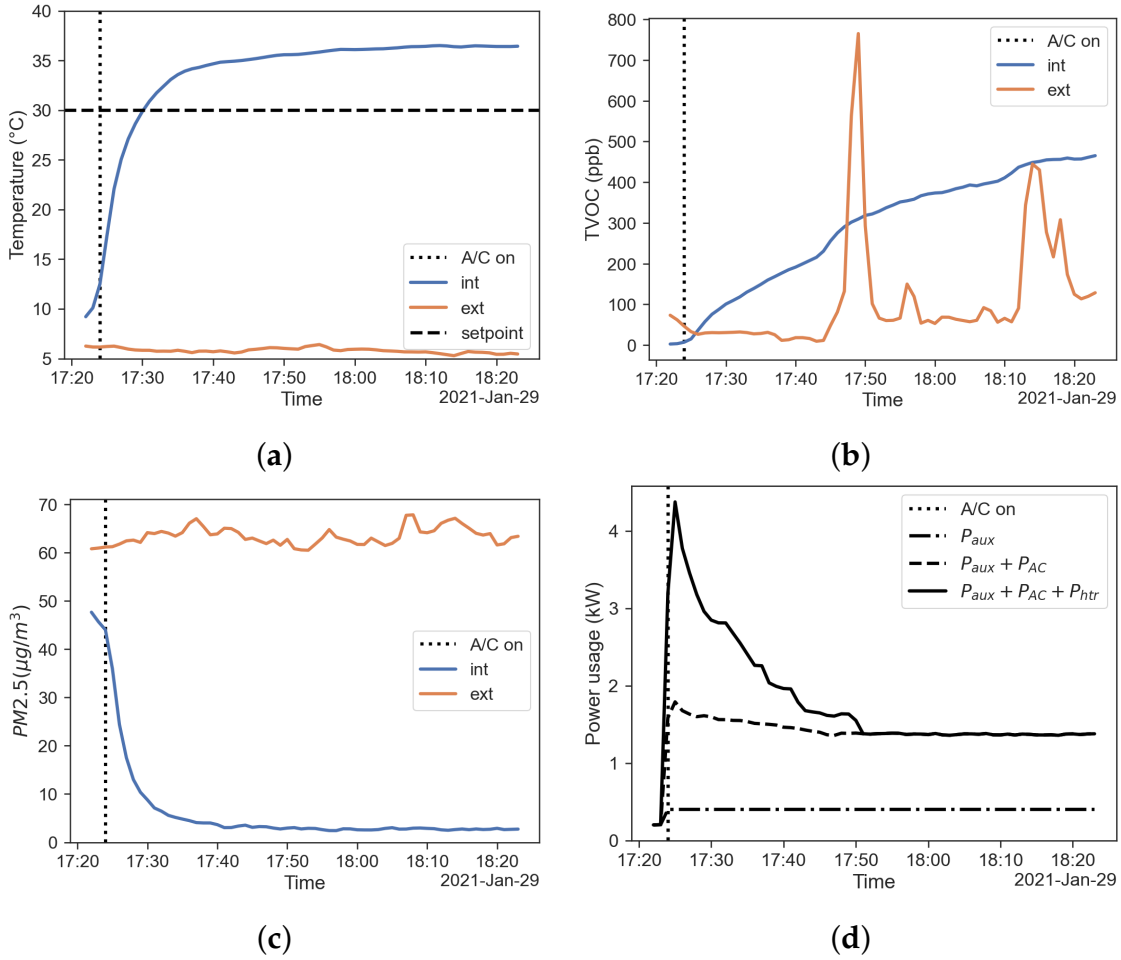


Figure 4.7: *Results regarding the recirculation mode. (**a**) Temperature inside (blue) and outside (red) the cabin; (**b**) TVOC concentration inside (blue) and outside (red) the cabin; (**c**) PM2.5 concentration inside (blue) and outside (red) the cabin; (**d**) power usage of the HVAC system.*

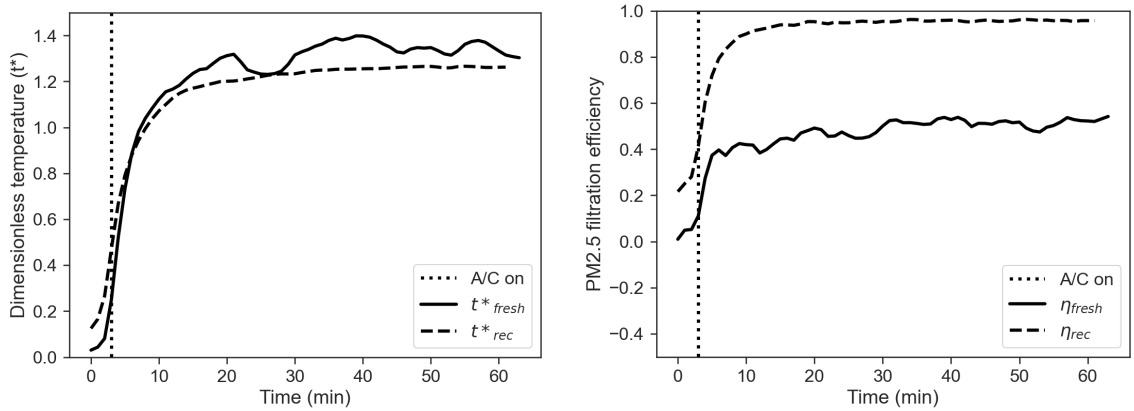
Comparison between fresh-air and recirculation mode in winter

The open-field tests conducted in this work have been chosen because representative of the real operating conditions of the vehicle. On the other hand, the experiments have been performed with no control on the environment outside the cabin, with repeatability issues. In order to compare the experiments, the following dimensionless temperature is defined:

$$t^* = \frac{t_{int} - t_{ext}}{t_{set} - t_{ext}} \tag{4.1}$$

where t^* is the dimensionless temperature, t_{int} is the air temperature measured inside the cabin, t_{ext} is the air temperature measured outside the cabin and t_{set} is the set-point temperature. It is worth to underline that when $t_{int} = t_{ext}$, dimensionless temperature t^* is equal to 0, while when $t_{int} = t_{set}$, then t^* is equal to 1. These two key points represent two relevant physical states, equilibrium with the external environment and fulfilment of the set-point request, respectively. Figure 4.8a shows a comparison between the dimensionless temperatures obtained for the two experiments.

It is noticeable that the dimensionless temperature obtained without recirculation is always higher than the one obtained in the case of recirculation mode, thus suggesting that the over-temperature issue is more significant in this case. In addition, the set-point is reached faster during the fresh-air mode than during the recirculation mode.



(**a**) (**b**)

Figure 4.8: *(**a**) Dimensionless temperature profiles, comparison between fresh-air (solid line) and recirculation (dashed line) mode. (**b**) Filtration efficiency, comparison between fresh-air (solid line) and recirculation mode (dashed line).*

The filtration efficiency of the vehicle can be defined using a black box approach, where the vehicle cabin is considered as a system with an unknown filtration capacity, while inlet (external) and outlet (internal) concentrations are known. The filtration efficiency is then defined by

$$\eta = 1 - \frac{C_{int}}{C_{ext}} \tag{4.2}$$

where C_{int} and C_{ext} are the internal and external concentrations, respectively. Figure 4.8b shows a comparison between the PM filtration efficiency obtained in the two regimes. The figure shows that PM filtration efficiency with recirculation mode is almost double than the one obtained with the fresh-air mode. It is also noticeable that the filtration efficiency does never reach the ideal value of $\eta = 1$, suggesting that infiltration rate not equal to zero occur even if the vehicle is parked.

An alternative method to get insights about IAQ of a vehicle cabin relies on a time integrated inside/outside approach proposed in [Pha+19]. The associated index, the cabin air quality index (CAQI), is defined as follows:

$$CAQI = \frac{\int_{t_i}^{t_f} C_{int}(t)\,dt}{\int_{t_i}^{t_f} C_{ext}(t)\,dt} \tag{4.3}$$

where C_{int} is the internal concentration , C_{ext} is the external concentration, t_i is the start time and t_f is the stop time. Results based on this index for PM2.5 and TVOC are given in Figure 4.9.

The figure shows that the CAQI indexes for PM2.5 and VOC obtained for the fresh-air mode are much greater than the one obtained for the recirculation mode. Figure 4.10 shows the comparison between the cumulative energy consumption in the two cases of recirculation on and off, calculated as the approximate cumulative integral of $P_{tot} = P_{aux} + P_{AC} + P_{Htr}$ via the trapezoidal method, in order to integrate numeric data rather than a functional expression:

$$E = \int_{t_i}^{t_f} P_{tot}(t)\,dt \approx \frac{t_f - t_i}{2N} \sum_{n=1}^{N} (P_{tot}(t_n) + P_{tot}(t_{n+1})) \tag{4.4}$$

where t_i is the start time, t_f the final time and N+1 the number of samples available (equally spaced). The total energy consumption obtained in the recirculation mode is about 3/4 of the value obtained for in fresh-air mode. This result can explain what is shown in Figures 4.4d and 4.7d. These figures show that the power usage from the HVAC system is similar for the two modes in the first minutes of operation. However, when the effects of recirculation become prevalent, the values of the HVAC power usage related to the two modes differ considerably. In fact, while P_{tot} peaks at more than 4 kW in the first minutes of operation in both modes, it varies significantly towards the end of the test.

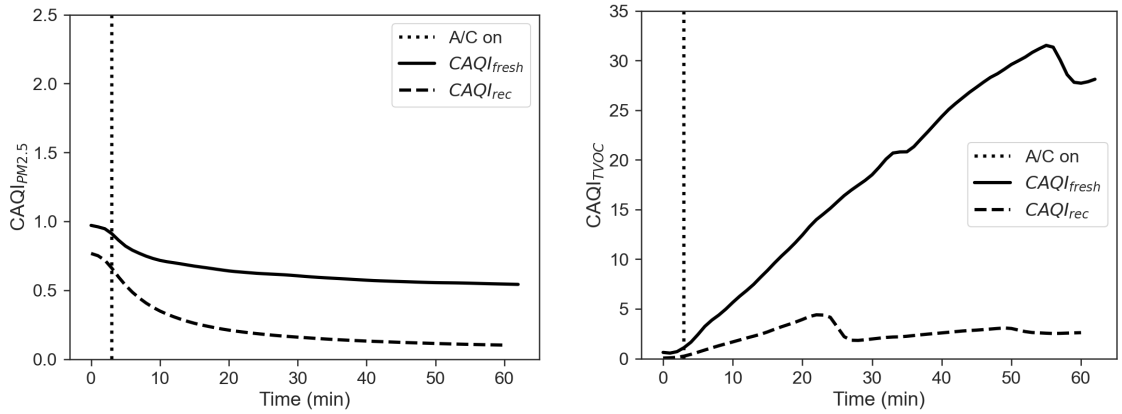


Figure 4.9: *(**a**) CAQI for PM2.5 and (**b**) CAQI for TVOC.*

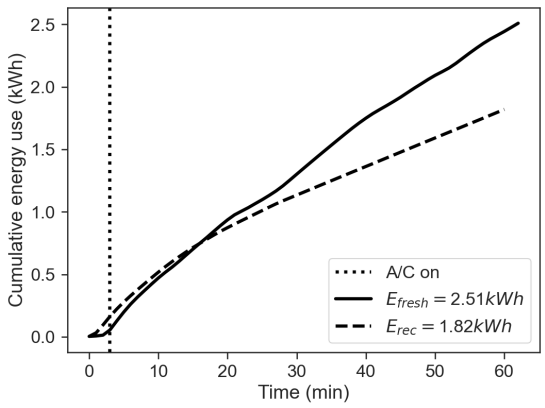


Figure 4.10: *Cumulative energy use, comparison between fresh-air (solid line) and recirculation (dashed line) mode.*

4.3.2 Cooling operation

Two different test conditions have been investigated starting from a state of equilibrium with the external environment, obtained by maintaining all systems off and all doors opened for 15 min. Once the equilibrium was reached, the proper test was performed while maintaining the A/C on for one hour, the set-point temperature at its minimum of 16 °C, the fan speed at its maximum (position 7), and all windows and all doors closed. During the first test, the recirculation system was off (that means that the air ventilation system was in fresh-air configuration), while during the second test the recirculation system was on.

Fresh-air mode

All the experiments confirm that the cooling system is not capable of reaching a quasi-steady state condition in about 60 min, i.e., the temperature reached by the air inside the cabin is far from the set-point temperature value.

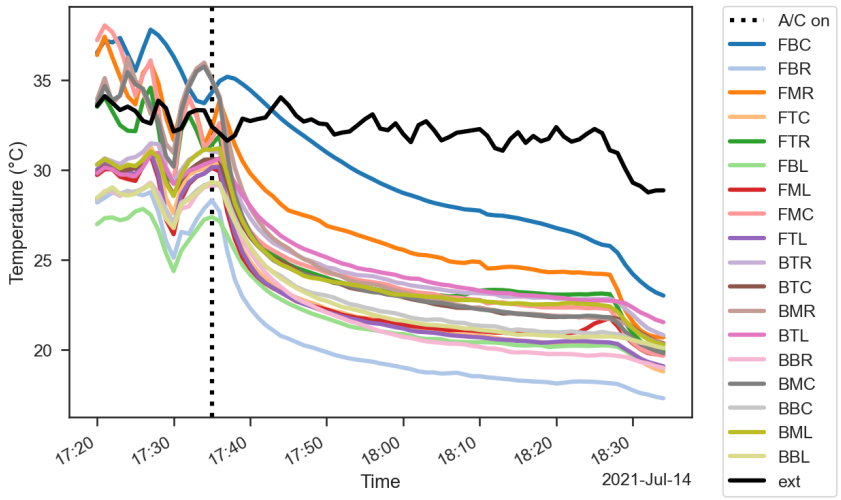


Figure 4.11: *Cooling temperature profiles, fresh-air mode.*

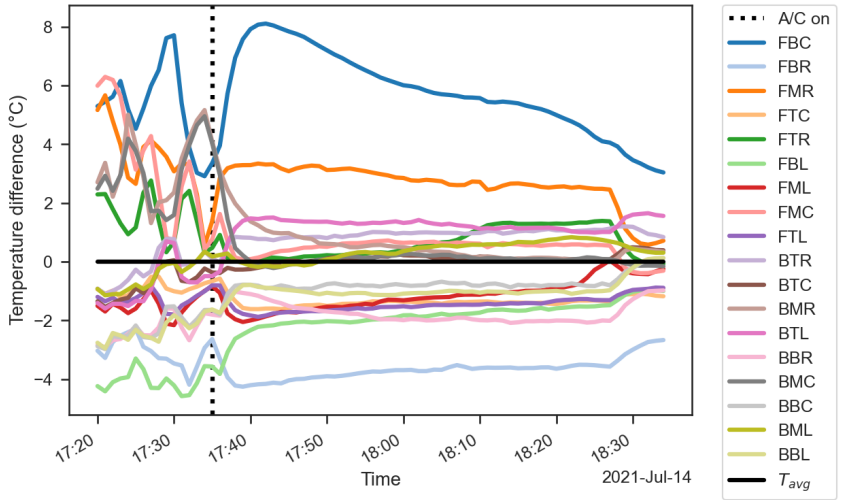


Figure 4.12: *Cooling temperature difference profiles, fresh-air mode.*

All the experiments confirm that the cooling system is not capable of reaching a quasi-steady state condition in about 60 min, textiti.e. the temperature reached by the air inside the cabin is far from the set-point temperature value. Figure 4.11 shows the readings from all the 18 temperature sensors in the cabin for the case with fresh-air mode with the same naming scheme introduced in the previous section.

Temperature discrepancies in the cabin can approach 12 °C; warmer spots can be found at top and middle locations, while bottom zone is generally colder (Fig. 4.12). To a certain extent, this is in agreement with a stratification of cabin air during operation, despite the vents working at full power. Looking at "Light summer" Comfort Zones defined by [Nil04], we can infer that occupants will be likely to accept warmer temperatures in lower body parts during summer, while this result goes in the opposite direction. Self heating and thermal inertia issue of the sensor positioned at FBC position are still evident.

Figure 4.13a shows the temperature measured inside and outside the cabin keeping the fresh-air mode. Like for the winter operation, the air temperature inside the cabin is calculated as the mean value of the 18 temperature readings for each timestamp.

Figure 4.13b shows TVOC measured inside and outside the cabin in fresh-air mode. The figure reports that the TVOC concentration is higher than that of the external air at the beginning, but decreases while the HVAC system operates. The lowest concentration is reached, despite the fluctuation on the outside. This behaviour can be explained with a drop of temperature inside the cabin combined with fresh-air mixing, thus reducing the emission from the internal sources.

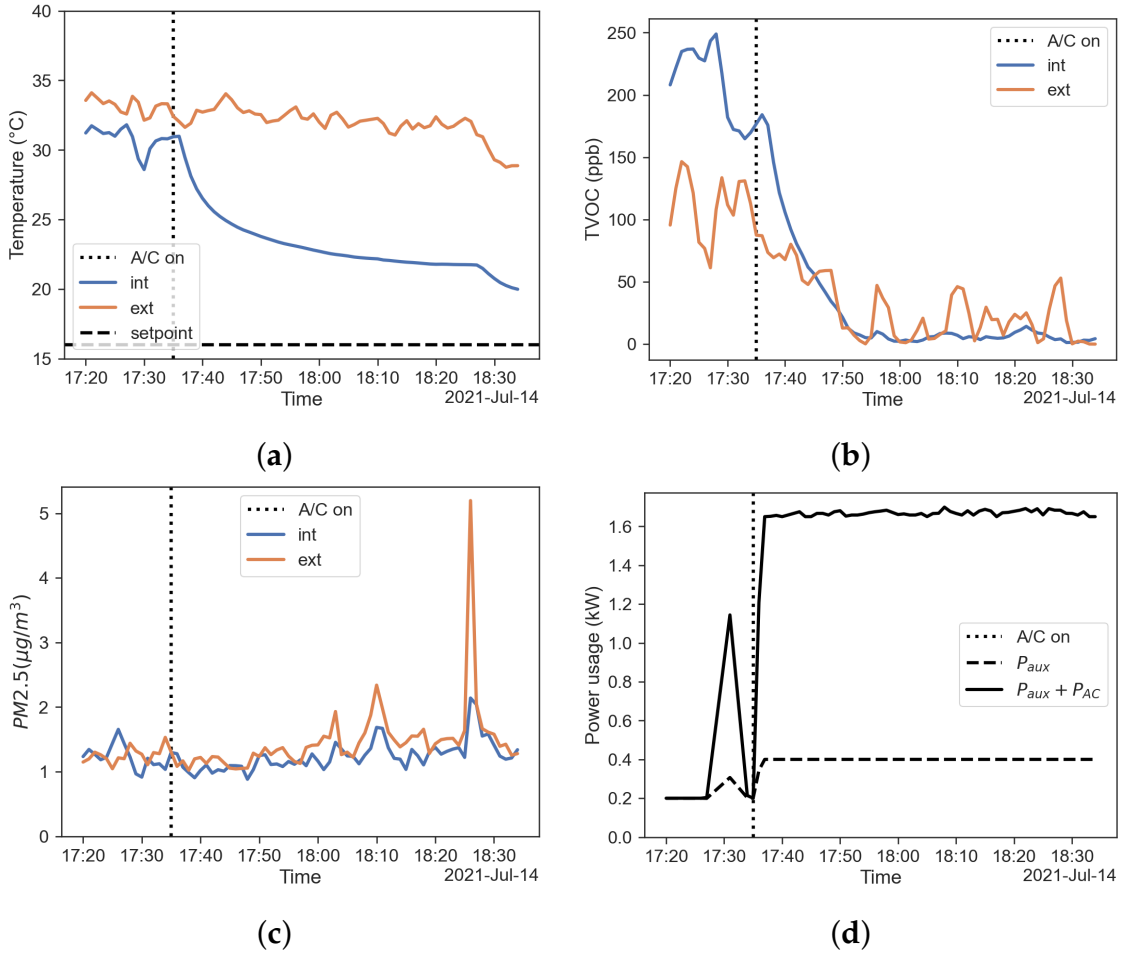


Figure 4.13: *Results regarding the fresh-air mode. (**a**) Temperature inside (blue) and outside (red) the cabin; (**b**) TVOC concentration inside (blue) and outside (red) the cabin; (**c**) PM2.5 concentration inside (blue) and outside (red) the cabin; (**d**) power usage of the HVAC system.*

Figure 4.13c reports plots of the PM2.5 concentration measured inside and outside the cabin in fresh-air configuration. This result appears to be in contrast with the results shown by [Heo+19], i.e., the cabin filter is not able to lower the PM concentration inside the car with a steady state filtration efficiency $\eta = 0$ to 0.2. Here we need to consider that the value of PM2.5 concentrations measured were extremely low and under the sensor precision for that particle size range ($\pm 10\,\mu g/m^3$).

Figure 4.13d shows the power usage of the HVAC system. The figure shows the contributions of power used by auxiliary equipment and A/C system; with PTC heater power being indeed equal to zero in cooling operation.

Recirculation mode

As for the previous case with recirculation activated, time to steady state is close to 20 min of operation, while the over-temperature issue is still significant.

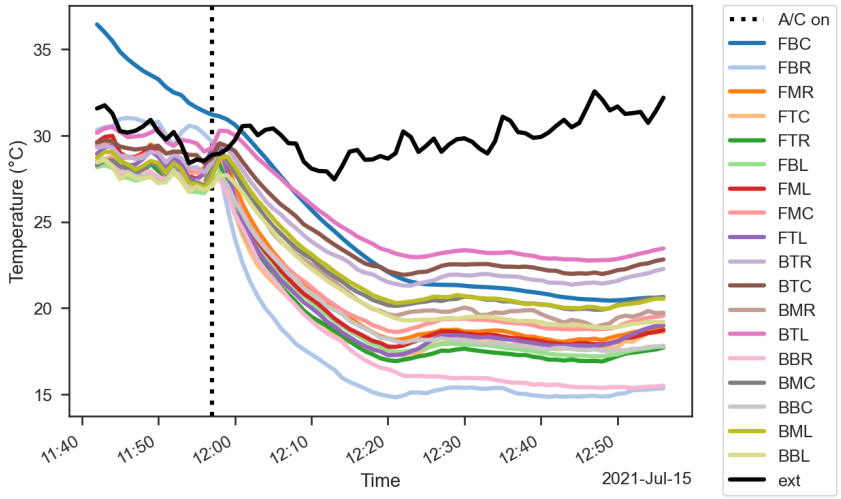


Figure 4.14: *Cooling temperature profiles, recirculation mode.*

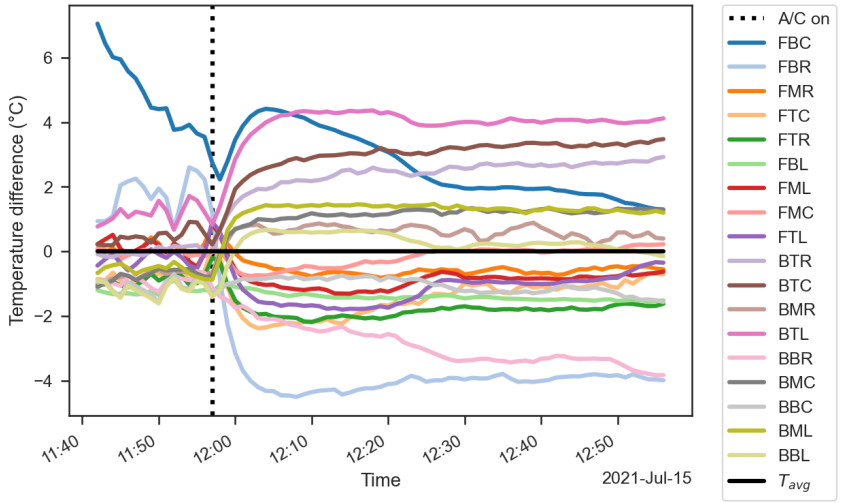


Figure 4.15: *Cooling temperature difference profiles, recirculation mode.*

Figures 4.14 and 4.15 show the readings from all the 18 temperature sensors in the cabin for the case with recirculation mode. Air temperature stratification with respect to the vertical axis is again considerable, colder spots can be found at feet level of the back seats. The main difference with the fresh-air case is that bottom right location are well capable of reaching the set-point temperature.

TVOC concentration shows a different trend. Despite external concentration peaks at the end of the test, the internal one remains quite low. As for PM values, they

follow a completely different trend; it is quite clear from Figure 4.16c how the filtration performance is improved by the recirculation mode, even with absolute values well within the precision range as is the previous test. Figure 4.16d shows the power usage of the HVAC system, with the contributions of power used by auxiliary equipment and A/C system recorded by the OBD system.

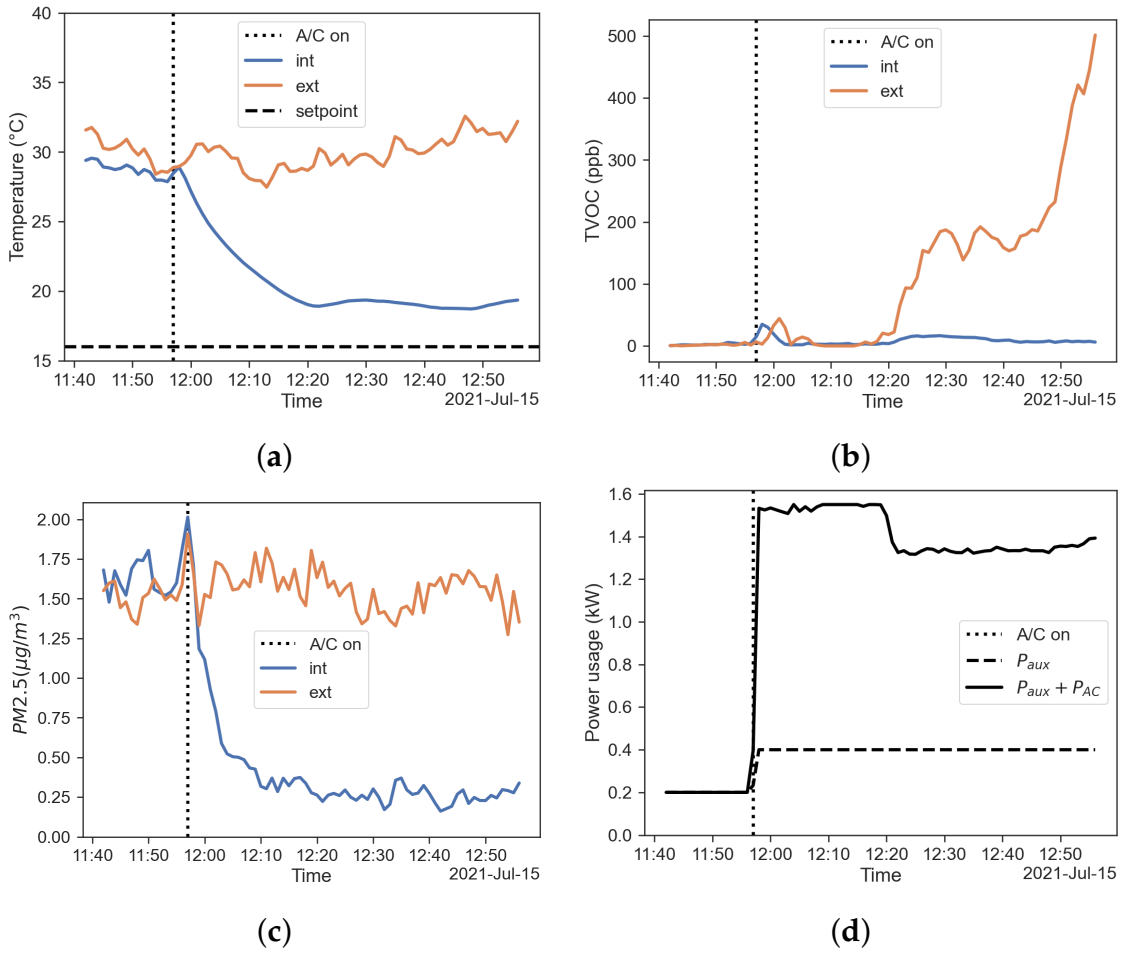


Figure 4.16: *Results regarding the recirculation mode. (**a**) Temperature inside (blue) and outside (red) the cabin; (**d**) TVOC concentration inside (blue) and outside (red) the cabin; (**c**), PM2.5 concentration inside (blue) and outside (red) the cabin; (**d**) power usage of the HVAC system.*

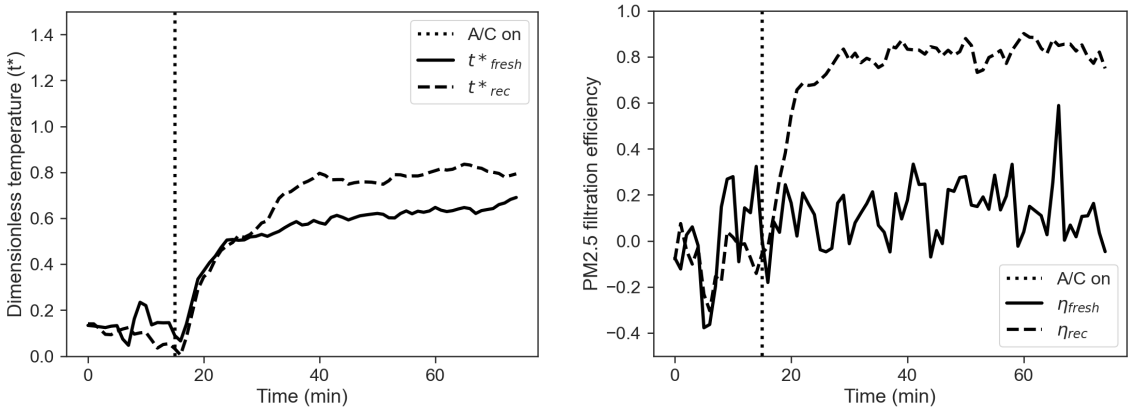
Comparison between fresh-air and recirculation mode in summer

Experiments in cooling as well as heating mode have been performed in real parking conditions with no control on the environment outside the cabin, repeatability issues are worsen by the increased contribution of solar load in summer. In order to compare the experiments, a temperature adimensionalisation is performed according to Equation (4.1). Figure 4.17a shows a comparison between the dimensionless temperatures

obtained for the two experiments.

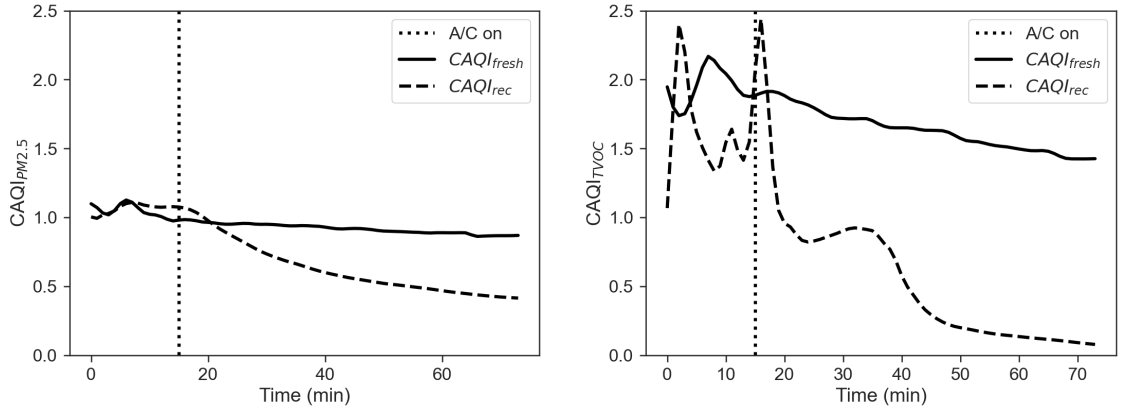
It is noticeable that the value of dimensionless temperature obtained with recirculation is always higher than the one obtained in the fresh-air case, thus suggesting that the cabin approaches better the set-pot in the first case.

The filtration efficiency of the vehicle can be defined again using a black box approach, where the vehicle cabin is considered as a system with an unknown filtration capacity, while inlet (external) and outlet (internal) concentrations are known, according to Equation (4.2). Figure 4.17b shows a comparison between the two cases. The PM filtration efficiency with recirculation mode is well over the one with the fresh-air mode. It is also noticeable that as for the winter case the filtration efficiency never does reach the ideal value of $\eta = 1$, but it is even lower indeed. This trend can be explained with η being a function of particle size [Heo+19], but also of particle concentration itself. As shown for the winter case, another way to investigate cabin performance on airborne pollutants is provided by Equation (4.3). Figure 4.18 reports the CAQI trend for PM2.5 and TVOC in fresh-air and recirculation, the latter being less prone to build up of pollutants during operation.



(**a**) (**b**)

Figure 4.17: *(**a**) Dimensionless temperature profiles, comparison between fresh-air (solid line) and recirculation (dashed line) mode. (**b**) Filtration efficiency, comparison between fresh-air (solid line) and recirculation mode (dashed line).*



(**a**) (**b**)

Figure 4.18: *(**a**) CAQI for PM2.5 and (**b**) CAQI for TVOC.*

Figure 4.19 shows the comparison between the cumulative energy consumption in the two cases of recirculation on and off, calculated as the approximate cumulative integral of $P_{tot} = P_{aux} + P_{AC}$ via the trapezoidal method as done in the winter case with Equation (4.4). The total energy consumption obtained in the recirculation mode on is about 4/5 of the value obtained for the fresh-air mode. This result can explain what is shown in Figures 4.13d and 4.16d. These figures show that the power usage from the HVAC system is similar for the two modes in the first minutes of operation. However, when the effects of recirculation become prevalent, the HVAC power usage related to this mode decreases after about 20 min, even though the difference is less prominent than in winter operation.

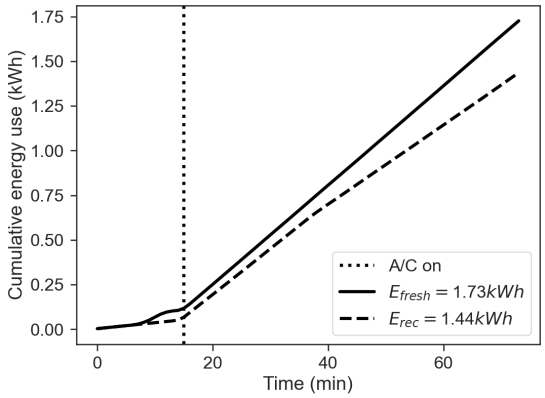


Figure 4.19: *Cumulative energy use, comparison between fresh-air mode (solid line) and recirculation (dashed line) mode.*

4.3.3 Effect of filter condition on filtration performance

In this section we report some results regarding the filtration performances of the Leaf cabin filter (Figure 4.20). In detail we performed two tests in summer operating conditions manipulating the filter. First, we performed the experiment without the filter. Then, a second experiment was performed after installing a brand-new filter. Figure 4.20a shows that the presence of the filter has an effect of filtration efficiency, lowering its value from 80% to 60%. The figure shows that even without a filter the recirculation mode provides a sort of filtration. This result suggests that part of the filtration is made by the filter and part is given by other devices in the HVAC system, i.e., a fraction of the pollutants is captured by the evaporator fins, or by the ducts between the cabin and the evaporator. Curves for fresh-air mode show filtration efficiency around zero for both the cases (no filter and with a filter), with more fluctuations for the case without a filter. Then, the presence of the filter does not improve the air quality within the cabin both by using a filter and by not using it.

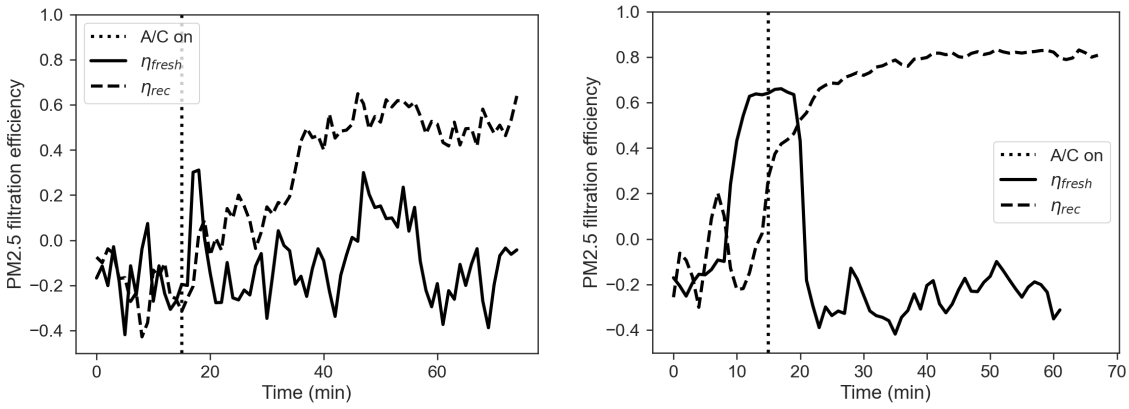


Figure 4.20: *Filtration efficiency, comparison between fresh-air mode (solid line) and recirculation (dashed line) mode, with (**a**) no cabin filter installed and (**b**) with brand new cabin filter installed.*

4.4 Chapter summary

In this chapter, a characterisation of the thermal profiles and air quality within the cabin of a Battery Electric Vehicle (BEV), together with real time measurements of HVAC system energy consumption has been presented. The temperature, PM, and VOC concentrations have been measured by means of a low-cost Arduino-based system of sensors. Comparisons between the air quality obtained in the cabin during in two different configurations of the HVAC system have been carried out.

The data collected shows that, while PMs are filtered, VOCs concentration increases during operation in recirculation mode. At the same time, the HVAC energy consumption in recirculation mode is about 70% of the energy consumption measured in fresh-air mode during heating operation. In the cooling operation, the HVAC energy consumption in recirculation mode is about 80% of the energy consumption measured in fresh-air mode.

Recirculation mode is found to be the best choice for BEVs, both for reducing some pollutants concentrations and for saving energy. The use of a new filter can improve the filtration efficiency in recirculation mode.

The air quality is strongly related to air-circulation modes, such as the fresh-air or recirculation. The recirculation mode should be chosen for energy saving in order to extend the BEV driving range, but a fresh-air mode is needed in some cases to ensure low concentrations of pollutants within the cabin. Control systems should consider these results in order optimally manage the HVAC system operation in BEVs.

In the next chapter, the same experimental setup is used to generate real driving data for a ML model of the cabin and HVAC system for optimal control purposes.

Chapter 5

Machine learnt model and Reinforcement Learning control

The previous chapter describes an application of the low-cost system of sensors to charachterise the cabin of a production electric vehicle in terms of thermal and air quality profiles. In a wider perpective, the charachterisation stage can be seen as the precursor to building and evaluation of the model stage. It also provides knowledge of context, intuition as well as development of the computational toolbox from pre-processing to visualisation.

In this chapter we aim to use the output of the LCSoS to build a ML model of the cabin and HVAC system, formulate its transient behaviour as a linear state space control problem and immerse it in a RL environment to investigate the possibility of finding the best iaq-aware, energy efficient thermal management policy for the vehicle cabin using the proximal policy optimisation (PPO) algorithm.

The chapter starts with a description of the physics based modeling approaches in Section 5.1, ranging from their typical structure to the most used software tools, together with some limitations for our application. Section 5.2 introduces the ML approach to thermal modeling in vehicle cabins, the trials specifications, the requirements of data preparation and cross-validation of the linear regression prediction on the test datasetset during the construction of the model, state and actuation inputs are also introduced. In Section 5.3 the ML model is benchmarked progressively detaching it from the data, first using actuation imput from data and then connecting the model with a mathematical model of a bang-bang controller. The algorithmic loop that updates successively state and control inputs is given.

Section 5.4 introduces the basic concepts behind RL. Starting from the definition of an agent learning the best policy from interaction with an environment (which comprises a cabin, HVAC model), the Q-learning and PPO algorithms as well as the

reward function are described. Section 5.5 contains results on the learning process in terms of the evolution in time of the mean reward for each episode. A total number of 8 trials has been executed, the values used to inizialise the environment toghether with a discussion of the results is also provided.

5.1 Physics based models

Despite the complexity involved in the modelig of transient and non uniform thermal environments like vehicle cabins, physics based models are extensively used in their design process with exceptional results [LYE20]. CFD simulation tools in particular, are unrivaled when a fine spatial resolution is required, often combined in co-simulation with a lumped parameters model of the HVAC and control system.

The main reason behind the wide adoption of this modeling scheme both in academia and industry, is the inevitable performance tradeoff between spatial and temporal resolution. Among the commercial tools used in combination with CFD we can find AMESim, Modelica, MATLAB & Simulink and more recently Simscape. Most commercial 1D modeling tools offer a robust equation based platform, capable of native interaction well established control software. They also provide a wide variety of advanced thermal-hydraulic and two-phase flow libraries for modeling evaporation and condensation. The main drawback is that they often require many parameters to be set up and are usually slower than custom algorithms built on purpose. Moreover, most of the state of the art numerical methods for ML are written in other languages like Python.

The typical structure of lumped models is based on building blocks that represent a particular physical phenomenon using the mass and energy balance equations that best describe it. Thermal blocks for example can be used for conduction, convection and radiation, they then can linked to form a thermal network together with a solver block [MAT18]. Different physics like fluid flow or electrical components can be linked together through dedicated ports, in some cases is possible to modify the equation used and even build custom blocks. For example, a conduction block would solve the well known Fourier equation between two layers of the same material:

$$Q = k \cdot \frac{A}{d}(t_1 - t_2) \tag{5.1}$$

where Q is the heat flow, k is the thermal conductivity of the material, A is the area normal to the heat flow direction, d is the distance between the two layers and t_1, t_2 are the temperatures of the layers.

State of the art studies make use of 0-dimensional (0D) volumes connected with 1-dimensional (1D) for the fluid phase, an extensive thermal network for heat exchange and custom Simulink blocks for the control network, even though no mention in made to the computing time per computed time step [Cho+18; Tit+16; CZ19]. Since its first releases, Simscape has been successfully used by some authors for vehicle cabin modeling tasks, with some interesting studies proposing a recovery of the spatial resolution using multiple volumes linked to represent the different section of the cabin [MS11]. Further developments of this scheme present a detailed description of the envelope layers and of the heat exchange mechanism involving the cabin, with a particular focus on the external side convection which uses Nusselt number to include a speed dependent convection heat transfer coefficient. The importance and the lack of prior art on moisture, CO_2 and pollutants build up is also underlined [KTM21]. As imaginable, solving heat transfer equations in each layer of each component of the vehicle envelope can be accurate but model speed will be reduced.

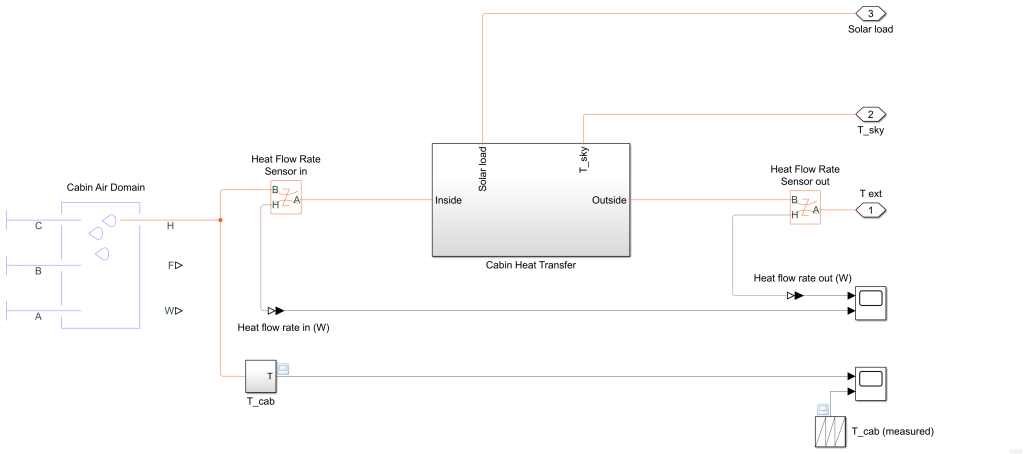


Figure 5.1: *An example of the cabin model that was initially built in Simscape to fit data from the LCSoS. The model required about* 40 min *to fit data from a single* 20 min *trial.*

In this work, a simple thermal and fluid cabin model shown in Figure 5.1 was initially built in Simscape with the aim of fitting it to the experimental data from a Nissan Leaf as described in Chapter 4, in an effort to develop a complementary tool to the one already available on the same vehicle that focus on powertrain modeling [IT19]. It was soon discovered that the process of tuning the model was quite slow using the builtin optimiser, taking up to 2 seconds to process 1 second of experimental data in the learning stage. Moreover, the structure of the model, the difficulties in retrieving thermophysical properties for each block, on a production car without support from the manufacturer and its poor interoperability with state of the art RL Python libraries made us opt for an alternative approach.

A slightly different approach was taken by Doyle et al., which developed a temperature prediction algorithm based on global heat energy supplied to the cabin environment as a function of setpoint and cabin temperature [DM19; DMS15]. Similarly, a global heat balance like the one in Equation (5.2) comprising heat exchange with environment and heating/cooling energy from the HVAC system can be used to predict internal cabin temperature if all other quantities are know. Which, for a given time step yields:

$$\dot{Q} = UA \cdot (t_i - t_e) \qquad \text{or} \qquad t_i = t_e + \frac{\dot{Q}}{UA} \tag{5.2}$$

where $\dot{Q}(t_i)$ is the sum of the rate of heat exchanged with the environment and with HVAC system, UA is the heat transfer rate and t_i, t_e are internal and external temperatures. With such formulation the internal-external relation to estimate future cabin states becomes explicit.

In the same way, also IAQ related pollutant concentrations can be described in terms of finding the correspondence between concentrations inside and outside the vehicle cabin, pollutant sources/sinks and internal-external air mass flow rate [Kor20]. Which for a given time step can be written as:

$$V\dot{C} = G + H \cdot (C_e - C_i) \qquad \text{or} \qquad C_i = C_e + \frac{V\dot{C}_i - G}{H} \tag{5.3}$$

where V is the volume, $\dot{C}_i$ is the rate of change of internal concentration, G is the generation/removal term, H is the internal-external exchange rate while C_i, C_e are the internal and external concentrations. It should be noted though, that despite this implicit formulation a composite functional relationship remains between left and right hand side of Equations (5.2) and (5.3).

5.2 Machine learning models

Another interesting result in this field, which poses the ground for machine learning approaches, relies on the implementation of the Newton's Law for the analysis and dynamical simulation of physical systems. In fact, starting from the assumption that many systems can be described using current state and control inputs, with most complex and highly non-linear systems requiring also information on state and control inputs at previous time steps [Jes+22], it is possible to show that the behaviour of these systems can be approximated with good accuracy with a linear relationship. For a

general Newton's model, considering a small time interval this can be written as:

$$\frac{dy}{dt} = -k \cdot (y - y_0) \qquad \text{with} \qquad \frac{\Delta y}{\Delta t} \approx \frac{dy}{dt} \tag{5.4}$$

$$\frac{y(t + \Delta t) - y(t)}{\Delta t} \approx \frac{\Delta y}{\Delta t} \tag{5.5}$$

$$y(t + \Delta t) \approx k\Delta t y_0 + (1 - k\Delta t) \cdot y(t) \tag{5.6}$$

which shows how the prediction on the next time step has a linear correlation with the previous one if $y_0, k, \Delta t$ are constants.

Another way to look at the problem is provided by the definition of dynamic mode decomposition with control (DMDc) and its formulation in terms of linear regression. The latter does not rely on physics equations but aims at finding the relationship between state and control variables at time t (remember that complex and dynamic environments often require also state at time $t - 1$) on one side, and state at time $t + 1$ which represents a prediction on the evolution of the system [BK19]. The generalised evolution equation can be written in terms of a dynamic mode decomposition with control (DMDc) problem:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \tag{5.7}$$

where A and B are the best fit linear operators that satisfy the dynamics of measurement data, $\mathbf{x}_t$ is the snapshot of the system at time $t + 1$, $\mathbf{x}_t$ is the snapshot of the system at time t, and $\mathbf{u}_t$ is the actuation input at time t.

In the next sections, it will be shown how to use this formulation to build a ML model of the Leaf cabin and HVAC directly from the data collected with the equipment shown in Chapter 4.

5.2.1 Real driving condition trials

Every ML task starts with the collection of data, in this study we personally collected all the data from a real driving experiment using the equipment as well as the same pre-processing described in the previous chapters. Test specification for heating and cooling operation are given in Tables 5.1 and 5.2.

For the cooling trials, a fourth sensing system (Seward Solar Survey 200r) was also installed to measure direct solar radiation I_{sol}. The device is a silicon based solar Irradiance meter with a measurement range of $0\,\mathrm{W/m^2}$ to $1500\,\mathrm{W/m^2}$, a resolution of $1\,\mathrm{W/m^2}$ and minimum sampling time of 60 s, it is widely used in the solar industry even though some alternatives relying on solar illuminance have recently been proven viable [MJM20]. The resulting dataset was appended to the previous ones, using a secondly

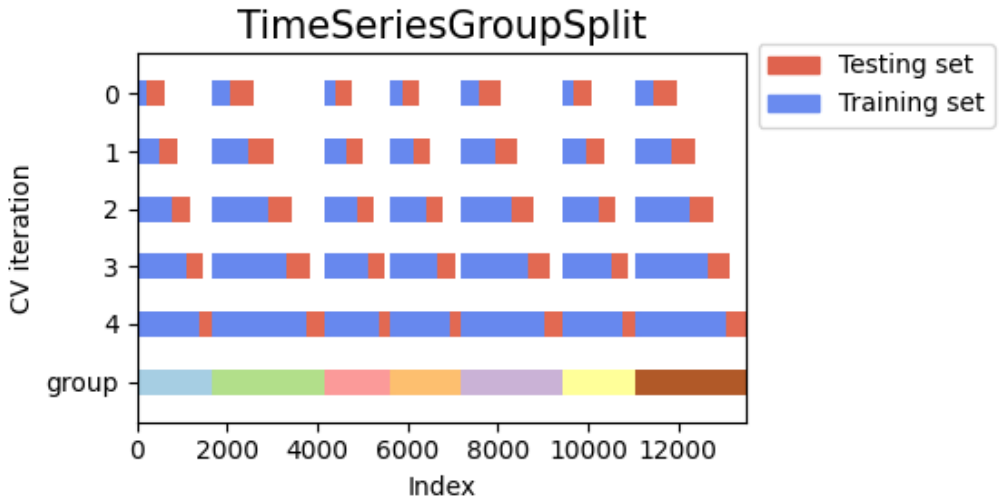


Figure 5.2: *Successive training and testing sets are super-sets of those that come before them,thus adding surplus data to the previous training partition, which is then used to train the model using cross-validation.*

upsampling to match the other three LCSoS.

The main difference with prior data preparation is that in this case there is a need to feed a ML task using the scikit-learn python package. Two widely used techniques for this scope are scaling and cross-validation. Often performed together in the same algorithm [KJ13]. All inputs and outputs are scaled to the range 0 p to 1 prior to learning with sklearn minmax scaler, so that the minimum value on the dataset can be represented with 0 and the maximum value with 1 for each variable. The scaling is performed at the algorithmic step right before cross-validation and is thus a small influence on validity.

The advantage of incorporating scaling within cross-validation sets is that it better reflects unseen data that may have a range outside that of seen data and this has an even more significant impact when used with neural networks that produce outputs strictly in the range 0 to 1

In the case of linear regression scaling is not really required, but for an improved interpretability of the results requires that model output regarding physical quantities are obtained scaling back the results using the same minmax scaler object. Another advantage of the use of scaling is that it allows to compute basic in-training performance metrics of several different quantities with different units, which would otherwise be impossible to compare.

Table 5.1: *Real driving test specification in heating operation*

Trial	Date	Start Time (CET)	Duration (mins)	Setpoint (°C)	Distribution	Fan
R6	2021-11-23	17:49	42	20	fresh	4
R7	2021-11-23	18:34	39	20	rec	4
R8	2021-11-24	9:36	44	16	fresh	4
N5	2021-12-14	17:00	32	16	rec	4
N6	2021-12-14	17:33	26	16	rec	1
N7	2021-12-14	18:00	29	20	rec	1
N8	2021-12-14	18:30	30	25	rec	4

Table 5.2: *Real driving test specification in cooling operation*

Trial	Date	Start Time (CEST)	Duration (mins)	Setpoint (°C)	Distribution	Fan
RC1	2022-10-07	12:56	27	16	fresh	1
RC2	2022-10-07	13:28	25	16	rec	1
RC3	2022-10-07	13:58	25	20	fresh	1
RC4	2022-10-07	14:31	26	20	rec	1
RC5	2022-10-07	15:02	25	16	fresh	4
RC6	2022-10-07	15:37	23	16	rec	4
RC7	2022-10-07	16:06	28	20	fresh	4
RC8	2022-10-07	16:45	25	20	rec	4

Linear regression model learning

A special type of k-fold cross-validation technique, suitable for timeseries data, called time series group splitting is used to produce a more appropriate estimate of the model on the training set. While a subsection of the dataset is stored to be used as test set. To make it representative of the time correlation between two consecutive time steps in the model, successive training and testing sets are super-sets of those that come before them,thus adding surplus data to the previous training partition, which is then used to train the model [Ped+11].

A graphical representation of the cross validation process is given in Figure 5.2, where the dataset is split in terms of the different trials which form the groups. Within each group, five successive training-testing cross validation are performed. The index correspond to a single sample of data or 1 s for the considered database. A time series split k-fold cross-validation is performed within each group (trial), while the training happens across all groups at each fold.

A summary of the cross-validation accuracy for each trial is given in Table 5.3, which tells how well the model performs in predicting the next state of the system when the previous two states and the actual control variables are available, if compared with unseen experimental data from the test set. Average scaled RMSE on the test set is ± 0.004 or 0.4%. The resulting linear regression model is then ready to be used in the RL environment, being able to predict 1300 s of simulated time in 0.01 s of computational time.

State and action variables definition

The definition of state and action variables, as well as which one and how many of them should be used is not straightforward and it is linked with interpretability issues. On one hand, if a model is built using too many features, they could be redundant, cross-correlated and both the ML and RL model could end up being unnecessarily slow. On the other hand, using too few of them could lead to a poor prediction accuracy. Ultimately, using a control variable as a state variable and vice versa may produce conflicts in the simulation pipeline, with more than one actor trying to control the same quantity.

A general framing of the problem relies on the assumption that all the quantities that are sensed and can not be modified by an agent are state variables, while all external thus non controllable inputs, as well as the quantities associated with the HVAC and control system are considered action variables, although some exceptions may apply in special cases.

Table 5.3: *Summary of cross-validation accuracy for predicted values against the test set, train R-squared: 0.99, train time:* 0.02 s.

Trial	simulate MSE
N5	0.00139
N6	0.00577
N7	0.00865
N8	0.00214
R6	0.00914
R7	0.00540
R8	0.00244
RC1	0.00169
RC2	0.00234
RC3	0.00079
RC4	0.01335
RC5	0.00649
RC6	0.00854
RC7	0.00086
RC8	0.00123
average	0.00468
+/-	0.00373

Table 5.4: *Variables that form the elements of the state* $\boldsymbol{x}$ *and control* $\boldsymbol{u}$ *vectors.*

State variable	Name	Description	Unit
x_1	FBC	Front Bottom Center air temperature	°C
x_2	BTR	Back Top Right air temperature	°C
x_3	BTC	Back Top Center air temperature	°C
x_4	BMR	Back Middle Right air temperature	°C
x_5	BTL	Back Top Left air temperature	°C
x_6	BBR	Back Bottom Right air temperature	°C
x_7	BMC	Back Middle Center air temperature	°C
x_8	BBC	Back Bottom Center air temperature	°C
x_9	BML	Back Middle Left air temperature	°C
x_{10}	BBL	Back Bottom Left air temperature	°C
x_{11}	RH_i	Internal relative humidity	%
x_{12}	$PM25_i$	Internal PM.2.5 concentration	$\mu g/m^3$
Control variable	**Name**	**Description**	**Unit**
u_1	T_e	External air temperature	°C
u_2	RH_e	External relative humidity	%
u_3	$PM25_e$	External PM2.5 concentration	$\mu g/m^3$
u_4	I_{sol}	Direct solar radiation	W/m2
u_5	S	Vehicle speed	Km/h
u_6	P_{aux}	Auxiliary power usage	W
u_7	P_{comp}	Compressor power usage	W
u_8	P_{htr}	Heater power	W
u_9	P_{AC}	Total AC power usage	W
u_{10}	T_{set}	Setpoint temperature	°C
u_{11}	M_{dist}	Air distribution mode	-
u_{12}	L_{fan}	Fan level	-

Another consideration about the availability and definition of state and action variables deal with correspondence between the variables available in the model and those available in the real application, specially for controlled systems. From this point of view, data-driven models have a clear advantage as they are built directly on the same quantities that will possibly be available to the real controller rather that on fictitious quantities that are only available in a Physics based model for example. On the contrary, A ML model will be constrained by the limited data available, and will only have a partial but hopefully good enough sensing of the physics phenomena involving the cabin.

In the current model, air temperature in 10 locations (T_i), relative humidity (RH) and $PM2.5$ concentration in the cabin are considered as state variables. While external air temperature (T_e) and RH, $PM2.5$ concentration, direct solar radiation (I_{sol}), together with vehicle speed (S), HVAC power consumption P_{hvac}, temperature setpoint T_{set}, air distribution mode M_{dist} (0 for fresh air and 1 for recirculation) and fan level L_{fan} (0–7) are considered the forcing variables that lead to the system snapshot at the following time step. A detailed overview of the structure of vectors **x** and **u** is available in Table 5.4.

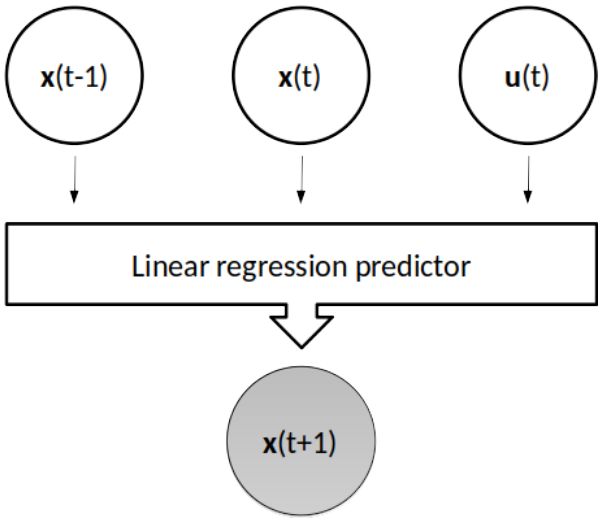


Figure 5.3: *Linear ML scheme for the simulator with* $x_{lag} = 1$ *and* $u_{lag} = 0$.

One last consideration about state and control vectors definition, that is directly related with the simulator structure and RL environment, deals with the lag choice. Following the prescriptions of Equation (5.4), before calling the `learn()` function the dataset is expanded according to two variables: x_{lag} and u_{lag}. They define how many timesteps backwards we want to inform the model.

In other words, they establish how the prediction should rely on past data, again also here a compromise is needed. Using high lag values could improve the accuracy of

the prediction but reduce its responsiveness to the dynamical evolution of the system. In this work we opted for $x_{lag} = 2$ and $u_{lag} = 1$, meaning that at the given timestep t, the model is considering also state variables at time $t-1, t-2$ and action variables at time $t-1$ to make its prediction on state at time $t+1$. This can be easily achieved expanding the data creating new columns, as is they were effectively extra variables. The same structure has to be conserved in the simulator and in every call that uses the `predict()` method on the saved model. A visual insight on the ML process workflow is given in Figure 5.3.

5.3 Simulator structure

5.3.1 Results with control actions from data

The LR model is able to predict 1300 s of simulated time in 0.01 s of computational time with a coefficient of determination above 0.99 on a commercial 8-core 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz laptop machine. The results shown in Figures 5.4 to 5.7 are based on the possibility to predict the next state when only the true initial value is given, toghether with the true control variable. That means at each time step the control value is the same used in the training process but the state variable at future timesteps are calculated on the previous prediction and not on the test data.

It is worth observing that the model is able to represent the general trend in the data, with some divergence occuring towards the end of the trial. A good corerspondence is observed for the first 500 s, which in a real world deployment scenario may be well providing enough time to train and run the model before the next simulation is needed. Unfortunately, a consistent divergence is observed after that time, even if the absolute values are near the sensitivity of the LCSoS, which leads to the consideration that a the model is not suitable for long term accurate predictions. It must be underlined that despite the relatively high number of trials (similar studies used 5, this study uses 15 trials), there was a small variability between them, specially within the same thermal operating mode. This may overestimate the model predictive capabilities as well as forcing a high degree of extrapolation when used with simulated control inputs.

For the sake of completeness, results comparing simulated and measured value of all 12 state variables in all 15 trials are given in the Appendix A.

It should be noted though, that the quality of the prediction is related to the fact that the model runs on the same control variable used for training, so it could very well present lack of generalisation as well as over-fitting the data. For this reason, furher results will be presented in the next section where the control action is not related with

the data. Another aspect that deserves further comment is the ambiguity between system dynamics and actuation for closed-loop systems. One way to address this issue is to add perturbation to the actuation signal in order to be able to distinguish between the two [BK19].

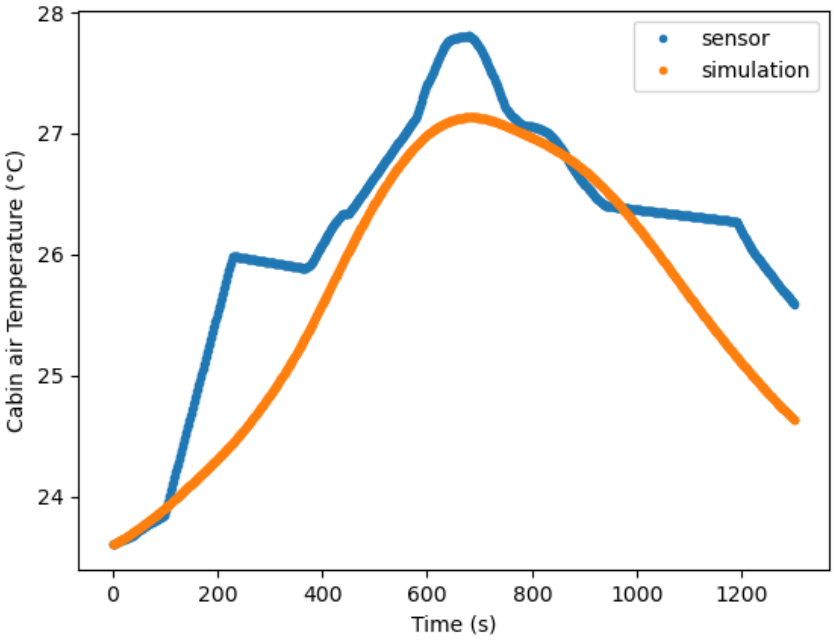


Figure 5.4: *Cabin air T heating.*

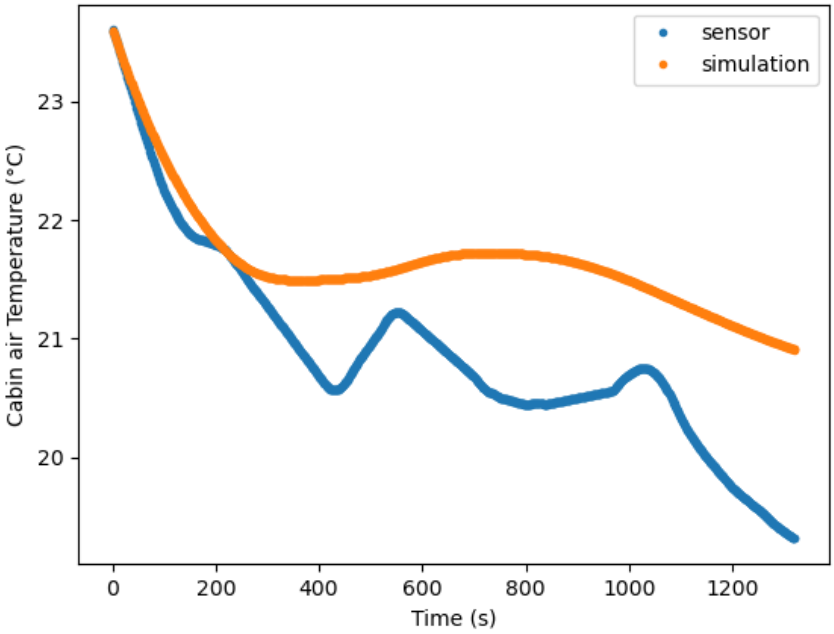


Figure 5.5: *Cabin air T cooling.*

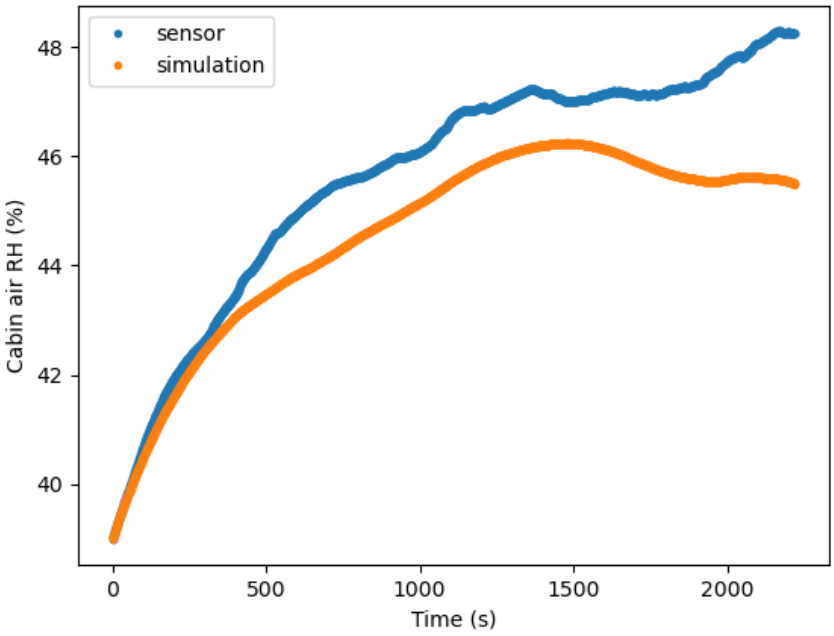


Figure 5.6: *Cabin air RH.*

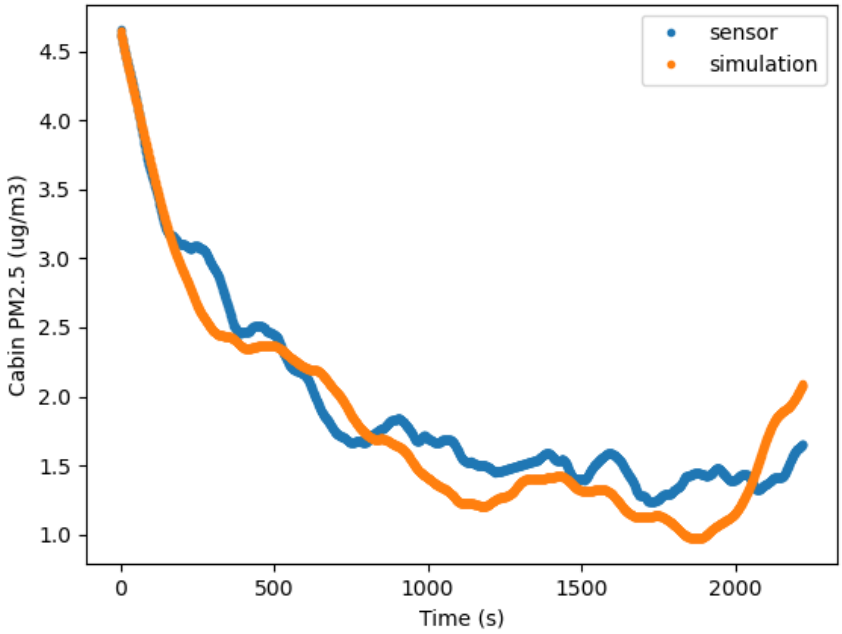


Figure 5.7: *Cabin air PM2.5.*

Using MLSim class

At the core of the simulation workflow is the MLSim class from the Domus project. From MLSim documentation[1]:

> The idea of this class is to wrap up the ML simulator and deal with scaling, lagged state and actions, and differing intervals to make the simulator easier to call. The initial state and prior actions are provided to the constructor and then the simulator can be run by successive calls to `step`.

The concept of stepped objects is crucial in this work, it starts from requirements of the RL environment (as will be shown in 5.4) and goes back over the LR model construction, borrowing the same formulation of linear state space control systems [WL07]. A sample code that demonstrates how the simulator can be put together with the learnt model to produce an output that predicts the state of the cabin using the action variables from data is the following:

```
1 import pandas as pd
2 import numpy as np
3 import joblib
4 from domus_mlsim import MLSim
```

[1]For further details about this class see the GitHub repository with all the code listings [Bru21b].

```
5
6  xlag=2
7  ulag=1
8  uc = 'R7'
9  scaler_and_model = joblib.load('lr.joblib')
10 leaf = pd.read_pickle("trials.pickle.gz")
11 initial_state = leaf.loc[uc].loc[start_t-1:start_t][xt].to_numpy().
       reshape(xlag, -1)
12 actions = leaf.loc[uc].loc[start_t::interval][ut]
13 mlsim = MLSim(scaler_and_model, initial_state=initial_state, xlag=xlag,
       ulag=ulag, xlen=len(xt), ulen=len(ut))
14 res = np.array([mlsim.step(actions.to_numpy()[i, :])[1] for i in range(
       len(actions)) ])
15 resdf = pd.DataFrame(np.array(res).reshape(-1, len(xt)), columns=xt,
       index=actions.index)
```

Listing 5.1: Sample code that calls the MLSim class with the LR model, initial state from data, control actions from data and stores the result in a pandas data frame for comparison with trial R7.

5.3.2 Results with control action from bang-bang controller

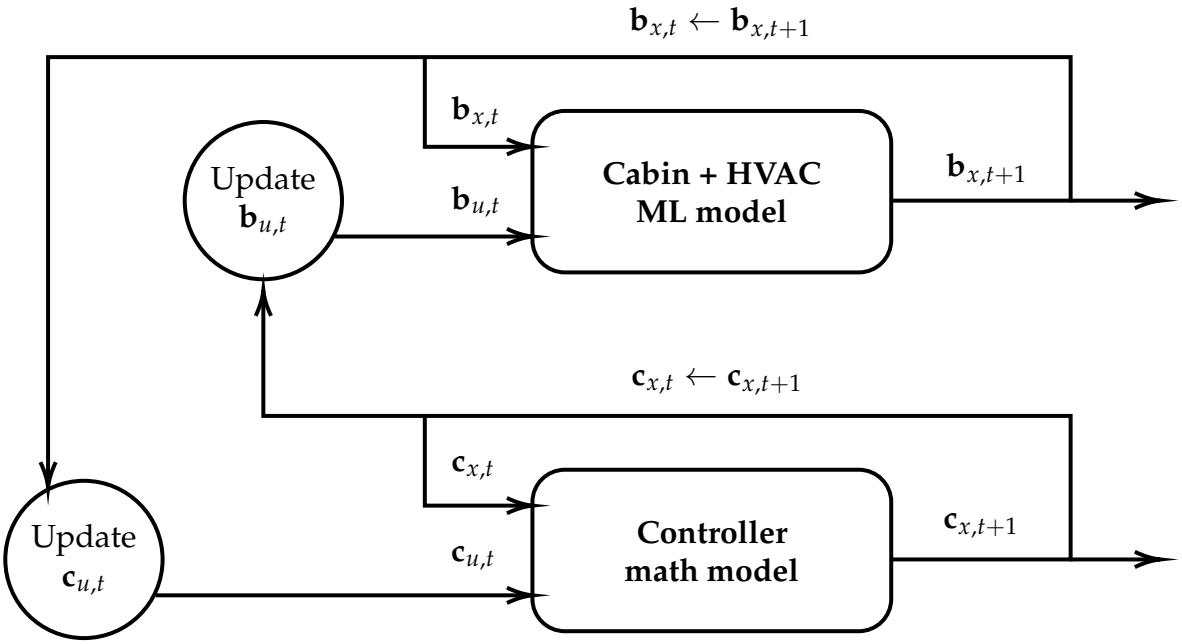


Figure 5.8: *Schematic diagram of the interaction between the controller and the ML model.*

After testing the model with the true action values from data, the idea behind this section is to add another layer of virtualisation using simulated control actions. To do

this, we use the simple-pid python package. A basic PID controller which given the proportional, integral and differential constants, and sensing the cabin state is able to set the next control action to reach a desired setpoint.

Conversely from the previous section, where the actuation vector was known in advance, to end up in a condition where the ML model of the cabin and the HVAC system can interact with a controller, the latter will need to provide the actuation input at each step, and to be stepped itself. The addition of a second model even though it is a simple mathematical one, leads to a slightly more complicated simulation structure that is shown in Figure 5.8.

It should be noted that learning cabin and HVAC system together simplifies the simulator structure. In fact, learning them separately would require a third model for the HVAC system. Such approach was taken in the Domus project, where an HVAC model was built on top of 1D simulation data and then broadcast with the cabin and controller ones. Other studies on buildings call directly the 1D model istead of building a ML model from it, which results in higher computation times but also enables the opportunity to try differend design solutions, while ML model are difficult to extrapolate to different designs [Bra+20; Han+19; WH20].

For a clearer formulation of the problem, it is helpful to define state and actuation input vectors for each of the two models, so that they contain the information on x_{lag} and u_{lag} in a more compact notation. In particular we will call $\mathbf{b}_{x,t}$ the set of state and lagged state variables for the cabin, $\mathbf{b}_{u,t}$ the set of actuation and lagged actuation variables for the cabin, $\mathbf{c}_{x,t}$ the set of state and lagged state variables for the controller, $\mathbf{c}_{u,t}$ the set of actuation and lagged actuation variables for the controller. A complete step cycle can be summarised as follows:

1. Initialise the state and actuation vectors.
2. Step the cabin and HVAC model to compute the new cabin state with MLSim
3. Step the controller to compute the next controller state with simple-pid
4. Update all old states with the newer values
5. Use the cabin and HVAC state to update the controller actuation input
6. Use the controller state to update cabin and HVAC actuation input
7. Continue with the next step

Figures 5.9 to 5.11 contain plots of internal air temperature, relative humidity and particulate matter concentration for a heating case where the control actions are set from a bang-bang controller with constants $P = 600$, $I = 0.3$, $D = 0.01$. Setpoint is fixed at 20 °C and $PM2.5$ concentration treshold that triggers recirculation on is set to 15 µg/m^3. External and initial temperatures are both set to 1 °C, external and initial RH are both set to 50 %, external and initial $PM2.5$ are set to 100 and 10 respectively. The intention in this setup is to model a cold startup

A big overshooting issue is evident on temperature but it then stabilised correctly around the setpoit, Some unrealistic negative values for the relative humidity are predicted corresponding to the overshoot, showing room for improovement, but in general it follows an inverse relationship with temperature which suggests an explainable result. Also Further tuning of PID constants may improve the results, but some overshoot as well as poor responsiveness issues may probably remain, this may be due to the filtering of data during pre-processing or lack of variability in the trials data.

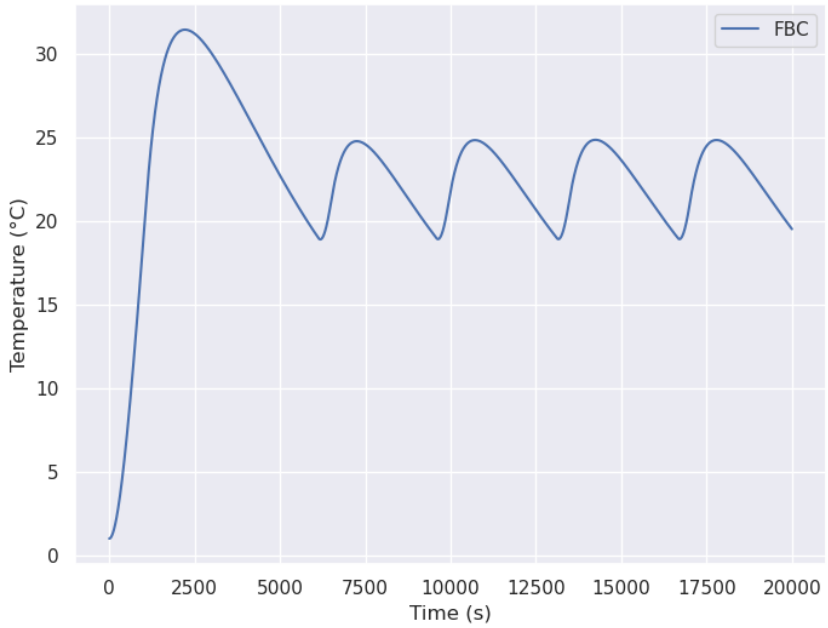


Figure 5.9: *Predicted temperature profile where the model is controlled with a simple PID controller.*

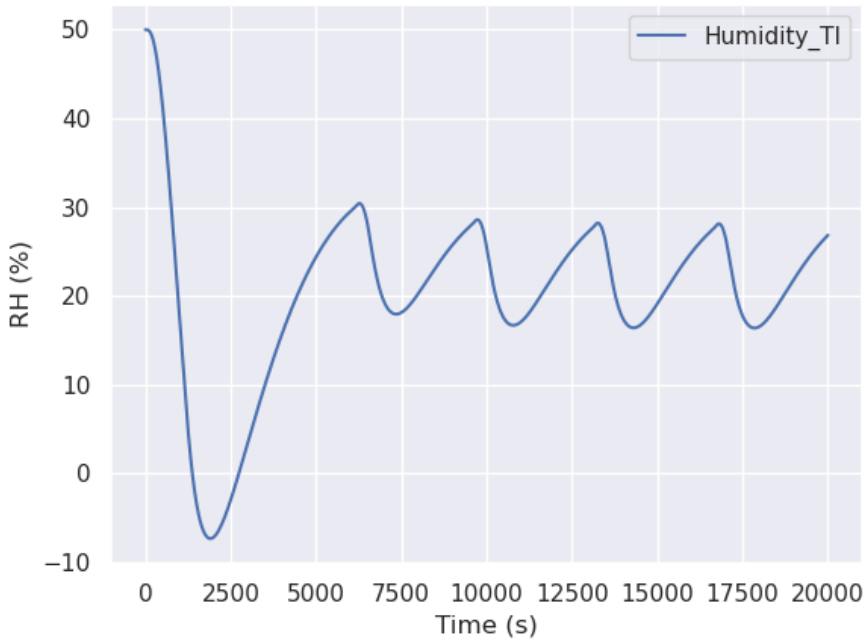


Figure 5.10: *Predicted humidity profile where the model is controlled with a simple PID controller.*

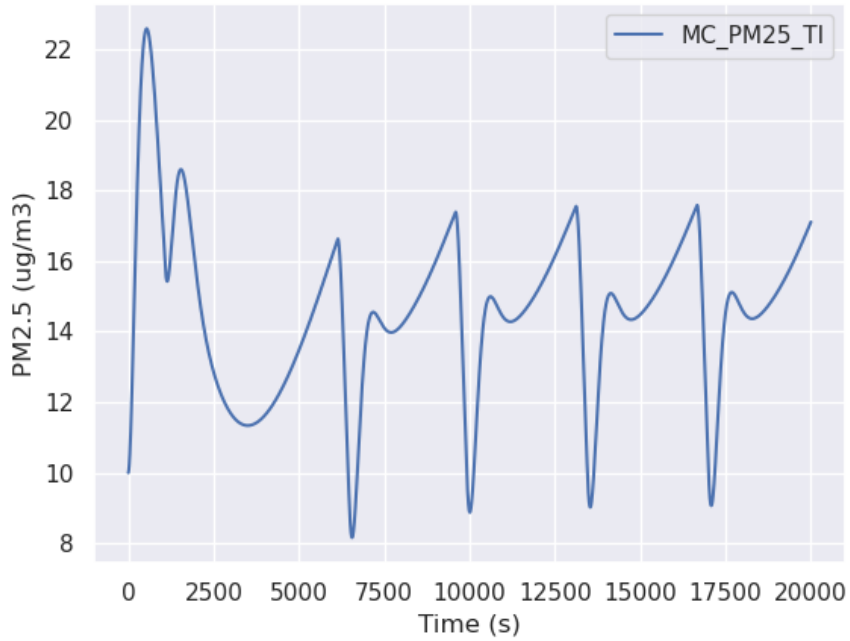


Figure 5.11: *Predicted PM2.5 concentration where the model is controlled with a simple PID controller.*

5.4 Reinforcement learning scheme

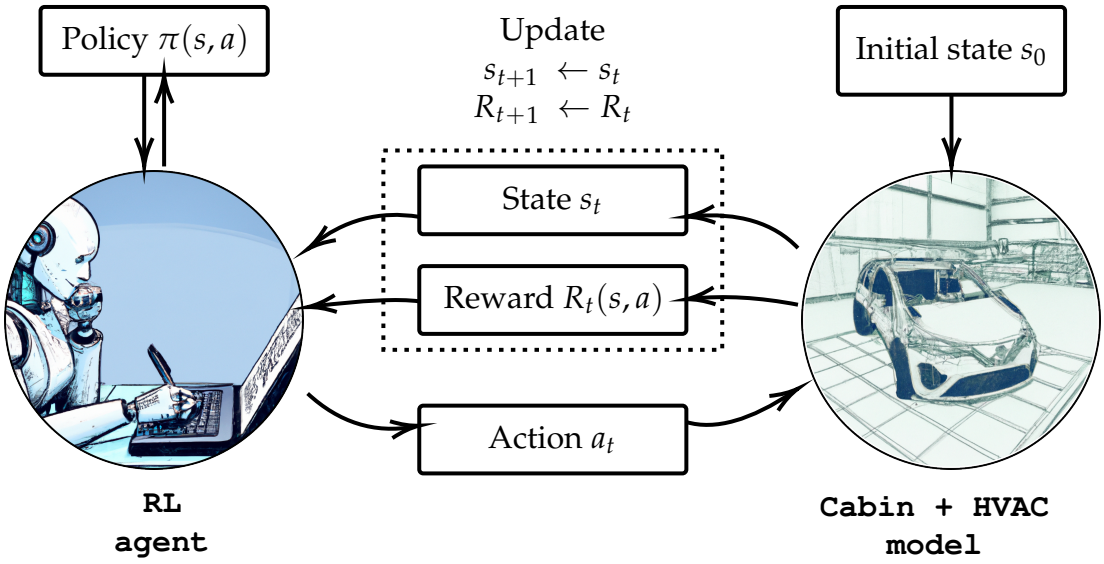


Figure 5.12: *The agent-environment interaction with reinforcement learning feedback used in this work.*

In this section we will frame cabin thermal management as an optimisation problem in terms of an agent (the controller) interacting with a ML model of an environment (cabin and HVAC), in each interaction the agent will receive a reward and update its policy for future interaction to maximise this reward (Figure 5.12). In terms of tools, OpenAI as environment structure and Stable-Baselines3 as experiment manager were used like in the Domus project. The fundamental building block of OpenAI Gym is the Env class. It is a Python class that basically implements a simulator that runs the environment in which the training of the agent happens. Main functions of the Env class that enable the agent to interact with the environment are:

- reset: This function resets the environment to its initial state, and returns the observation of the environment corresponding to the initial state. The reset function returns four things:
- observation: The observation of the state of the environment.
- reward: The reward that you can get from the environment after executing the action that was given as the input to the step function.
- done: Whether the episode has been terminated. If true, you may need to end the simulation or reset the environment to restart the episode.

- info: This provides additional information depending on the environment, such as value of hyperparameters, or general information that may be conducive in debugging.
- step: This function takes an action as an input and applies it to the environment, which leads to the environment transitioning to a new state.

Q-learning

In reinforcement learning, we want our agent to take actions that will maximise the possible rewards it receives from its environment. Q-learning is a reinforcement learning algorithm that seeks to find the best possible next action given its current state, in order to maximise the reward it receives.

The Q-learning algorithm will help our agent update the current Q-value $Q(S_t, A_t)$ with its observations after taking an action. Thus increasing Q if it encountered a positive reward, or decreasing Q if it encountered a negative one:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{5.8}$$

where a refers to all the possible actions available in a state. There are two hyperparameters which we can specify:

1. Learning rate (α): how easily the agent should accept new information over previously learnt information
2. Discount factor (γ): how much the agent should take into consideration the rewards it could receive in the future versus its immediate reward

As the agent learns more about the environment (exploration), it uses this knowledge to take more optimal actions and converge faster (exploitation).

Proximal Policy Optimisation and reward function

To make use of the ML model in the defined environment and run the RL we opted for a tool called Stable-Baselines3, an open-source framework implementing commonly used RL algorithms, with PPO being one of them. Moreover, it comes with an experimental framework, namely RL Baselines Zoo that provides scripts to train and evaluate agents, tune hyperparameters, record videos, store experiment setup and visualize results. Is also comes with a test suite that covers 95 % of the code in order to minimise implementation errors [Raf+21]. All the test available from Stable-Baselines3

as well as those available from the Domus project code base were passed successfully. Which leaves out essentially the data import and preprocessing scripts used in this work.

The PPO method performs multiple epochs of policy gradient update per data sample, namely batches of states, actions and rewards being used to estimate policy. Is has been used extensively in traning for locomotion, image generation and NLP tasks with a good success rate and data efficiency, such that is has become the default RL algorithm in organisations like OpenAI [Bro+16].

A detailed algorithmic analysis of PPO and its variants is out of the scope of this work, rather we focus on exploiting its well proven performance in benchmark tasks (Atari, MuJoCo) or other research areas, namely: good balance between ease of implementation, sample efficiency and ease of tuning. Another peculiarity of PPO is that after an update in the policy, the new one should not differ too much from the old policy [Sch+17; Sch16]. This can be done through clipping, so that given the ratio between the old stochastic policy (or the probability of taking a certain action in the current state), and the current policy (or an estimate of the advantage of the new policy using a gradient estimator):

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{5.9}$$

the surrogate objective to be maximised will be:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}\left[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \tag{5.10}$$

where ϵ is a hyperparameter, θ is the policy parameter, $\hat{A}_t$ is the estimated advantage at time t and $\mathbb{E}_t$ is the expectation or the average over a batch of samples in an alternation between exploration and exploitation.

More generally, we want the agent to take actions that will maximise the rewards it receives from the environment. The reward R(s,a) is used to evaluate how good the current state-action pair is for the current cabin plus HVAC plus RL agent jointly. The design process of the reward function is extremely difficult, if the RL agent is not given the right rewards or it is not receiving enough or not meaningful enough feedback from the environment it will almost certainly fail to reach the global optimum, or simply develop a policy that gives a reward for a reason that was not considered in the design process [Yu+21].

One way to increase the reliability of the model is to carefully define shaped rewards, the downside is that in doing so the agent may become too constrained in its exploration of the environment and follow a biased learning path. In this work, the optimisation

trade off between comfort, energy use and air quality is implemented through the following reward function, shaped on cabin temperature between two consecutive steps with a discount factor $\gamma = 0.99$ and a scaling factor of 0.1, to guide the learning process of the RL agent:

$$R(s,a) = w_c R_c(s) + w_q R_q(s) - w_e R_e(s,a) \tag{5.11}$$

where R represents the reward, s is state, a is action, w is the weight while the c, q, e subscripts stand for comfort, air quality and energy.

Looking at Equation (5.11), comfort and air quality terms give a positive reward, while the energy term gives a negative reward. Overall it acts as a penalty, so that the agent will seek for values that bring $R(s,a)$ close to 0. Each term at right hand side is normalised, cabin temperature is already scaled in the model. Energy is normalised between 0 i to 1 in terms of the maximum and minimum values set as control action, calculated as the sum of heating and cooling energy use [Val+19]. IAQ is instead normalised using a threshold value, 0 if the $PM2.5$ value is below the WHO 24h maximum exposure, -1 otherwise. Once the optimal control policy is available, meaning enough information to do the inference (i.e. convert observations into an sction prediction) is available, the trained agent be called in the same environment (without performing the training but using it as a playground) with the `step()` method. Similarly to what is shown in Section section 5.3, using the same model and scaler but changing the initial as well as the boundary conditions if needed.

5.5 Training results

The RL environment in the current configuration completes a traning task in about 10 h on a 32-cores 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz machine. A good way to benchmark the agent performance in terms of evaluating its ability to learn from interaction with the cabin environment is to monitor the evolution in time of the reward value also referred as learning curve. What drives the PPO agent in the policy update is the maximisation of the received reward, calculated as the sum of the state-action reward that the agent receives from the environment in a epoch.

We can look at each epoch as a virtual experiment similar to those performed during the experiments with the real system, but it this case instead of a human operator following a schedule there is a RL agents setting actuation inputs and measuring the outcome in terms of comfort, energy use and IAQ in that experiment. The process is then repeated over a large number of epochs until the agent learns the best set of actions to take when a certain state of the cabin and HVAC environment is sensed.

The following learning curves refer to a set of virtual experiments where, after initialising the state vector and the external conditions, the RL agent explores the control actions space for 1 epoch, then the environment is re-initialised. Each epoch has a duration randomly sampled from an exponential distribution with a mean value of 25 minutes. This is meant to reduce the dependency of the results from the duration of the experiment. Different sets of reward weights were also fixed for the whole duration of each training experiment. These, together with initial and fixed values are given in Table 5.5.

Figures 5.13 to 5.20 show the the learning curves for all the set of trials. The general appearance is not smooth as they all seem to be affected by some kind of disturbance or noise with an highly unsteady convergence if any. The most promising result are from trial 12 and 22 where an overall decreasing trend of the reward is observable. Both trials use a reduced comfort weight suggesting that reducing the constraint on cabin temperature leads to a still too slow (total simulated training time caps over 2×10^6 s or 23 days in most trials) and oscillating but converging reward. Further experiments that use double the amount of total timesteps show no significant changes in the reward profile, thus suggesting that doubling the simulated training time does not help the learning process in this case. In other words seems that the agent is able to do minor or no improvements to the policy even over a very long period of time. The only other work in the field shows a coverging learning curve after 100 min of global simulated time, but it is using data from CWT for the cabin and from a 1D AMESim model for the HVAC [Che21].

As discussed in section 5.4, designing the right reward function can be an extremely difficult task, requiring a lot of tuning on time consuming simulations. Moreover, in ML pipelines there are two additional dimensions along which bugs are common: the actual model and the data. Those two add up to the classic implementation error dimension to prune the search space of possible bugs becomes harder [MSA22; Mak+21; MSA20]. With this in mind, there are a number of reasons that could be proposed to explain this behaviour. Among them we can find the quality of the data collected. If compared with lab-scale equipment or simulated data, this is still an open issue for LCSs. On top of that, dealing with variable and not controllable test environment may play a crucial role in the quality of the model. In fact, real driving data may be more representative of the vehicle final operating condition but also add an unaccepptable amount of noise and bias to the model. Another factor that should be taken into account is the amount of data used to train the model, adding more data may be beneficial for the accuracy of the prediction. Finally, adding more features to the model is considered beneficial, given that they are not cross correlated [KJ13].

Table 5.5: *Initial values and reward function weights used in the RL environment in different trials.*

Variable	Value							
Trial	4	5	11	12	13	14	17	22
T_e	1	1	12	17	35	35	35	35
RH_e	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$PM25_e$	100	20	20	20	100	20	20	20
T_i	1	1	1	1	35	35	35	35
RH_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$PM25_i$	10	10	10	10	10	10	10	10
T_{set}	22	22	22	22	24	24	24	24
weights (c,e,q)	(0.5,-0.5,0.5)	(0.5,-0.5,0.5)	(0.5,-0.5,0.2)	(0.3,-0.5,0.2)	(0.5,-0.5,0.5)	(0.5,-0.5,0.5)	(0.3,-0.5,0.2)	(0.3,-0.5,-0.2)

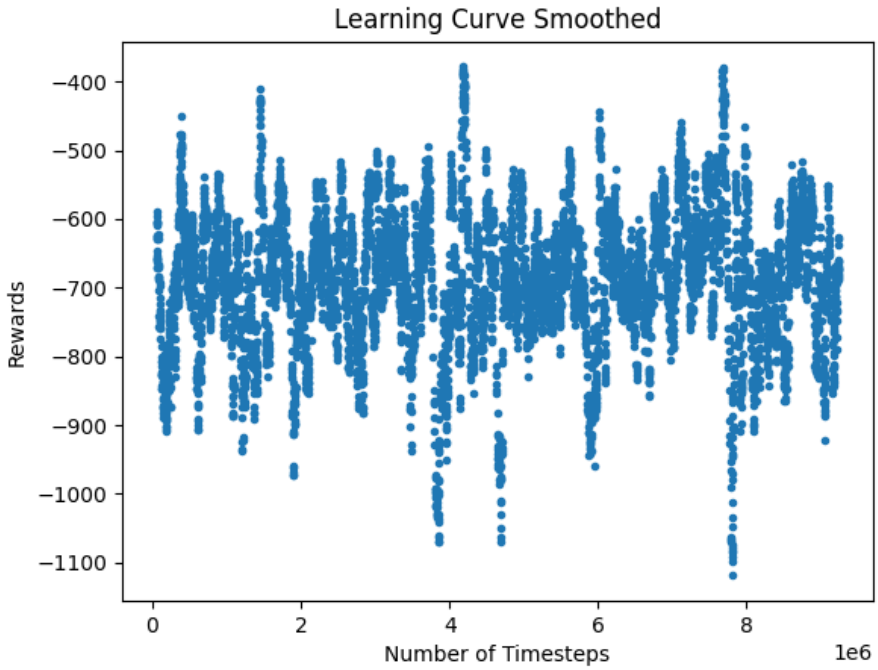


Figure 5.13: *Smoothed learning curve for trial 4, heating mode.*

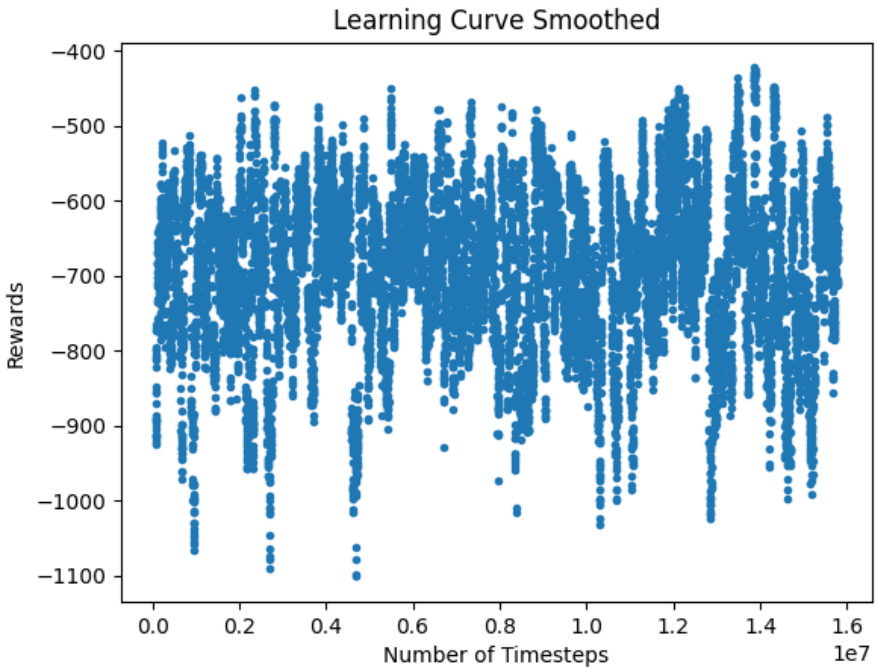


Figure 5.14: *Smoothed learning curve for trial 5, heating mode.*

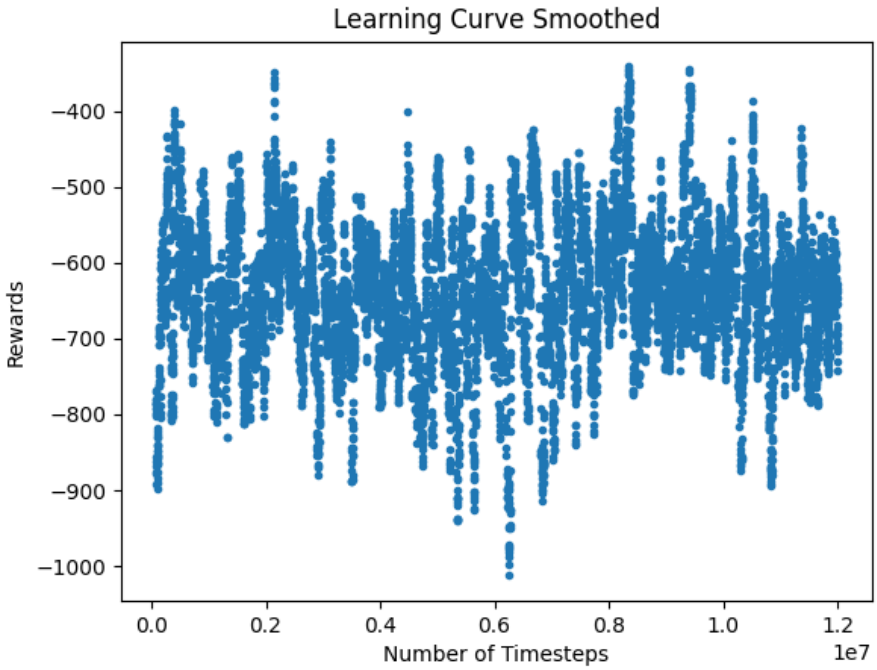


Figure 5.15: *Smoothed learning curve for trial 11, heating mode.*

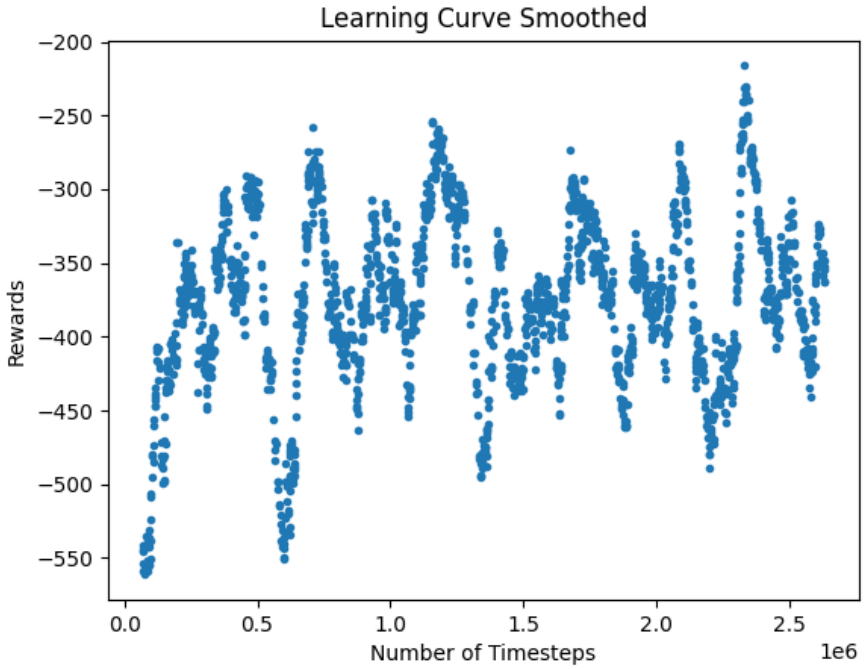


Figure 5.16: *Smoothed learning curve for trial 12, heating mode.*

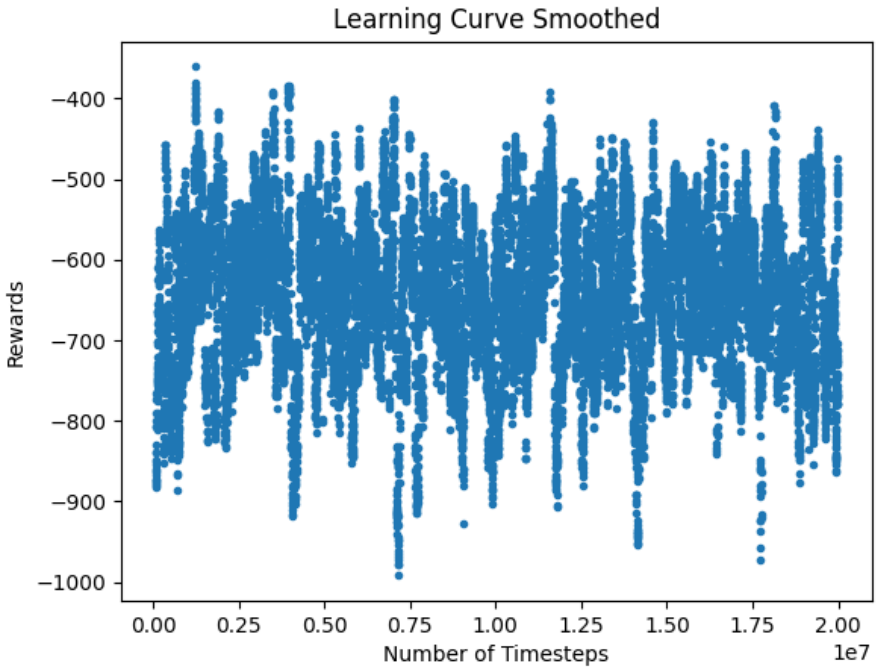


Figure 5.17: *Smoothed learning curve for trial 13, cooling mode.*

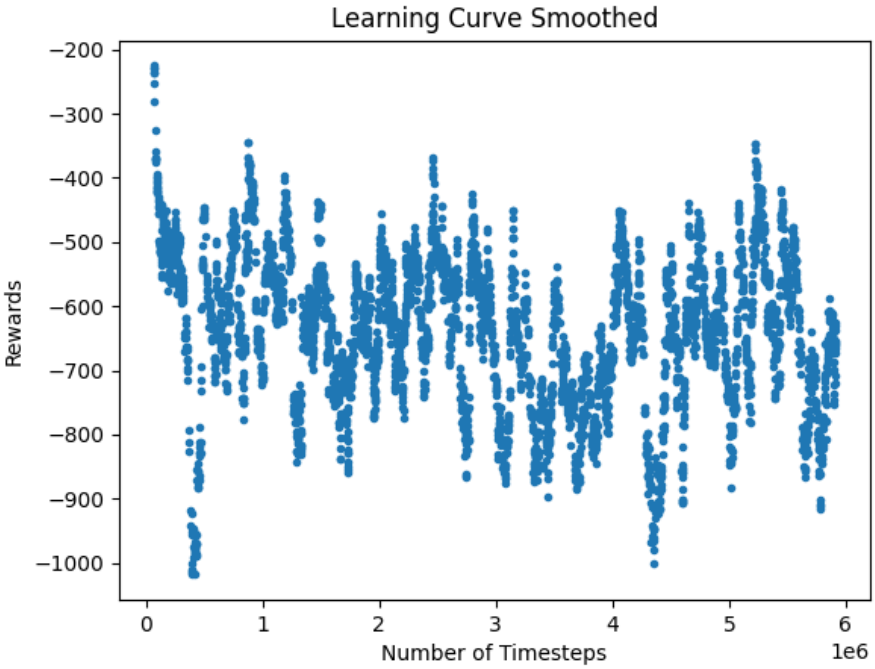


Figure 5.18: *Smoothed learning curve for trial 14, cooling mode.*

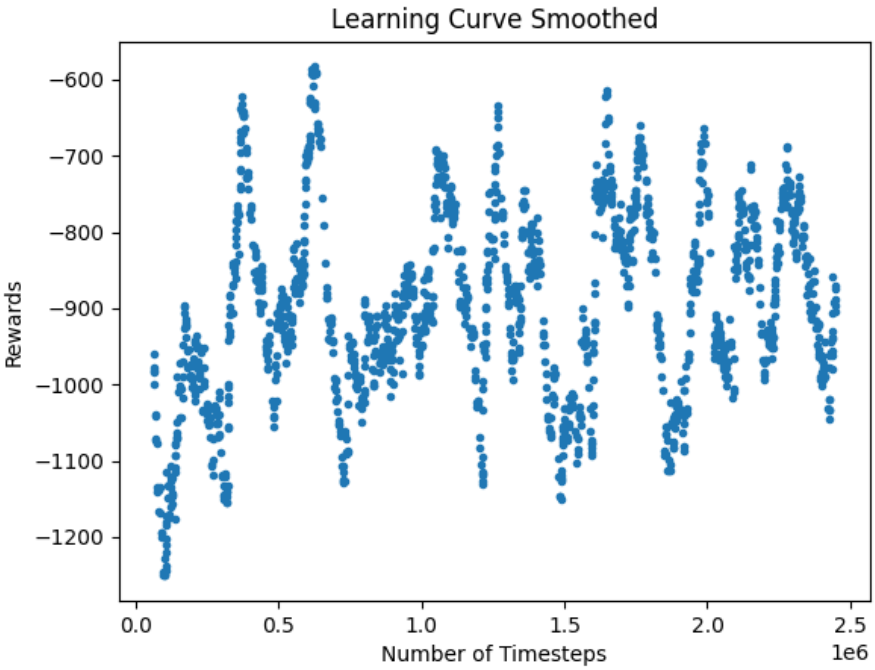


Figure 5.19: *Smoothed learning curve for trial 17, cooling mode.*

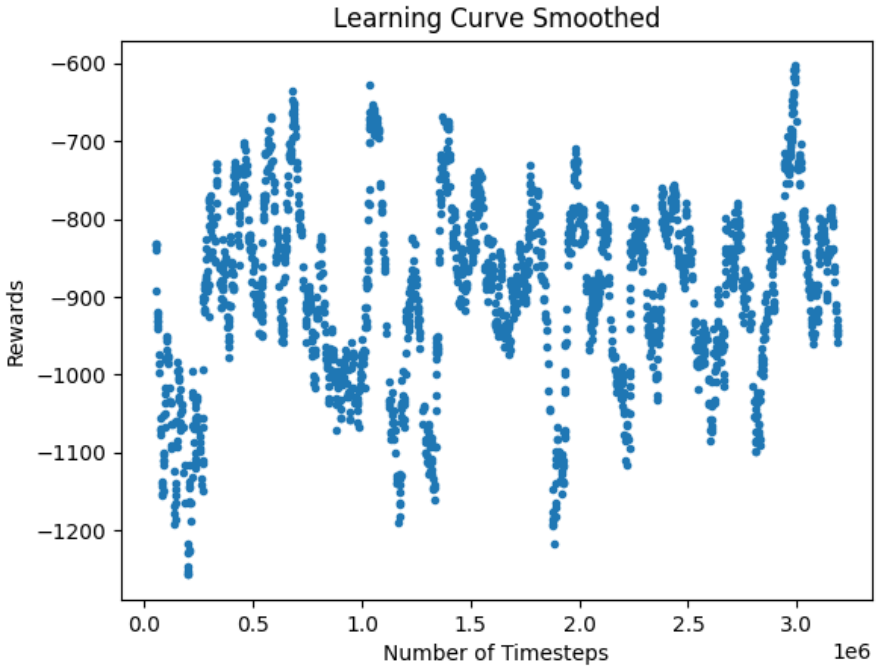


Figure 5.20: *Smoothed learning curve for trial 22, cooling mode.*

5.6 Chapter summary

In this chapter we showed how to use of the LCSoS to build a ML model of the cabin and HVAC system, formulate its transient behaviour as a linear state space control problem and immerse it in a RL environment to investigate the possibility of finding the best iaq-aware, energy efficient thermal management policy for the vehicle cabin using the proximal policy optimisation (PPO) algorithm in an OpenAI gym environment.

Conversely from a physics based models the ML model is built directly on real driving data organised in 15 trials. Meaningful variables for comfort, air quality and energy use are selected from those available in the dataset and then restructured to form the input-output variables of a linear state space control problem. After a data preparation stage, a linear regression model is built on the test set using time series group splitting. A cross-validation is performed within learning in each super-set of the train and test set, which gives an average RMSE over all trials of 0.4 %.

To benchmark the ML model, it was progressively detached from train data. The results regarding actuation imput from data show a relatively good performance in predicting the first 500 s, while a considerable divergence is observed for successive timesteps.

A further step away from the data is made coupling the ML cabin and HVAC

model with a simple bang-bang controller, so that the mathematical model of the controller sets actuation inputs to the ML model. Results show an overall overshooting issue indicating that the model is not sensible enough to the actuation or that the PID constants need further tuning. The steady state behaviour seems reasonable overall for cabin temperature and humidity, while some unexplainable fluctuation are observed for $PM2.5$.

The implementation of a RL environment for the ML model is successfully built, with a three terms reward function that contributes as a penalty to the total episodic reward. Results about the learning process for a total of 8 experiment in terms of the mean reward for each training episode leave a wide margin for improvement. In particular, only two experiments present an observable learning trend, while the remaining have an unsteady oscillating profile. Relaxing the comfort weight is shown to be helpful for a smoother learning process, which suggest further reward function engineering is needed.

Chapter 6

Conclusions

This thesis addresses the issues associated with one of the main barriers to a greater penetration of electric vehicles (EVs) in the transportation sector, namely the energy use due to the Heatin Ventilation and Air Conditioning (HVAC) system operation. In this context, despite being highly regarded as a solution to reduce dependence on fossil fuels increasing the share of renewables (SDG 7) and improve community health (SDG 3), there are still several gaps that need to be filled. In order to increase their driving range or to reduce the mass of the battery (given also the rare and costly materials used in their construction) is crucial to aim for the best possible thermal management of both cabin and powertrain, as this can cover up to 30 % of the energy use.

The research started from the human element, with a review of studies regarding thermal comfort in vehicles that evidentiated the main research direction in this field. Air quality was also included as a further element of comfort in such enclosed environments like vehicle cabins. Once defined the starting points, the focus switched to sensing and modeliing solutions, with one key refrexivity objective in mind, they needed to be as human-centric and reproducible as possible. The choice of using an open hardware platform of low-cost sensors (LCS) as well as set of free, open-source software tools did not came without compromise, but became part of the study itself to some extent. The data from the LCSs was used to charachterise the cabin together with the HVAC system and build a fast Machine Learning (ML) model of the vehicle Finally, an optimisation framework was needed, and as Reinforcement Learning (RL) recently proved to be an outstanding tool for this purpose, it was used in this work to seek the best compromise between comfort, air quality and energy use. Besides that, the primitive formulation of RL as the process of learning by trial and error while interacting with an environment was inspired by the way living beings learn, thus beautifully closing the loop with the human element.

In terms of research questions the following findings can be used to answer them:

- *Aim 1:* Is it possible to build a low-cost system of sensors (LCSoS) capable of providing good enough data to characterise the cabin environment in terms of thermal comfort and air quality, without using Climatic Wind Tunnels (CWT) and lab scale equipment?

  - *Aim 1.1:* Is it possible to demonstrate the metrological capabilities of a low-cost sensors, particularly IAQ sensing devices, in real driving conditions, thus enabling its operation in transient, non-uniform and moving environments such as vehicle cabins?

  Yes. The development of a portable LCSoS for thermal and air quality monitoring and carachterisation in vehicle cabins, was succesfully performed. The system is capable of high spatiotemporal resolution, with sampling times as low as 5 s. The LCSoS has a typycal coefficient of determination $R^2 = 0.92$. which can replace or at least reduce the use of rare and expensive facilities like climatic wind tunnels.

  Yes. The LCSoS is suitable for charachterising the cabin with real driving data as shown in this work. A step towards addressing the lack of quantitative specification, formulate requirements for mobile applications and conditions of intended use of these devices and their output was addressed. Results on real parking condition trials show that the systemis able to quantify a strong effect of recirculation on IAQ and energy use. Cabin air temperature in some locations of the cabin can differ up to 10 °C from the average in all 18 locations. While recirculation can produce energy saving of 70 % and 80 % in heating and cooling mode respectively.

- *Aim 2:* Is it possible to build a fast and accurate model of the cabin without the use of detailed CFD-1D simulations and CWT data but using a dataset obtained from LCS in real driving conditions instead?

  Yes. A fast, machine learnt thermal and air quality model of an EV cabin in an open source framework was built, ranging from data gathering, preprocessing, model training, testing and cross-validation on timeseries data. Results show that it is possible to build a fast thermal model of an EV cabin with data from an open source framework. Average scaled RMSE over all trials is ± 0.4 %, while computation time is 0.0077 ms for each second of simulated time on a laptop computer. Which makes it suitable for near realtime operation. If compared with measurment data, the model is able to predict pretty well the first 500 s of simulated time for cabin air temperature, relative humidity

and $PM2.5$ concentration. Some divergence is observed after that time even though the main trend respected. Interaction with bang-bang controller provides a stable responce of the ML model, leaving room for improvement on overshoot issue and responsiveness.

- *Aim 3:* Third aim is to perform the training of the RL agent in Stable Baselines3 with proximal policy optimisation (PPO) to find the optimal thermal management policy that finds the best trade-off between comfort, air quality and energy use.

  Yes. A training environment for the RL algorithm with PPO was built, and a three factor weighted reward funcion was defined, with the aim of finding the best compromise between comfort, air quality and energy use. The current simulation environment takes about 10 h to train the agent on an HPC node. Only 2 out of 8 experiments show a decreasing trend in the learning curve, which means the RL agent is able to improve the policy over time while interacting with the environment. A large oscillation is observed in all the learning curves, this suggest the agent in not receiving the right reward signal from the environment, or the model is not representative enough. Improvement of quantity and quality of data, together with more variables in the model could help smooting the curve.

6.1 Limitations and future work

Several further actions could be proposed to improve this work building up on its findings. Among which are:

- Implementation of T_{eq} instead of T_i using low-cost radiant temperature sensors (if available) for comfort, conserving the model structure
- Use of more structured IAQ index in reward function, for example $CAQI$ instead of threshold values
- Use more data (trials) and more variables (CO2, TVOC, powertrain, battery) to train ML model of cabin, HVAC and possibly powertrain all together
- Define a more complex RL environment with more state and action variables to train RL agent
- Further reward function engineering

In a scenario where all the improvements above are addressed, and a remote communication channel is available on-board, a practical implementation of this thesis work would enable to:

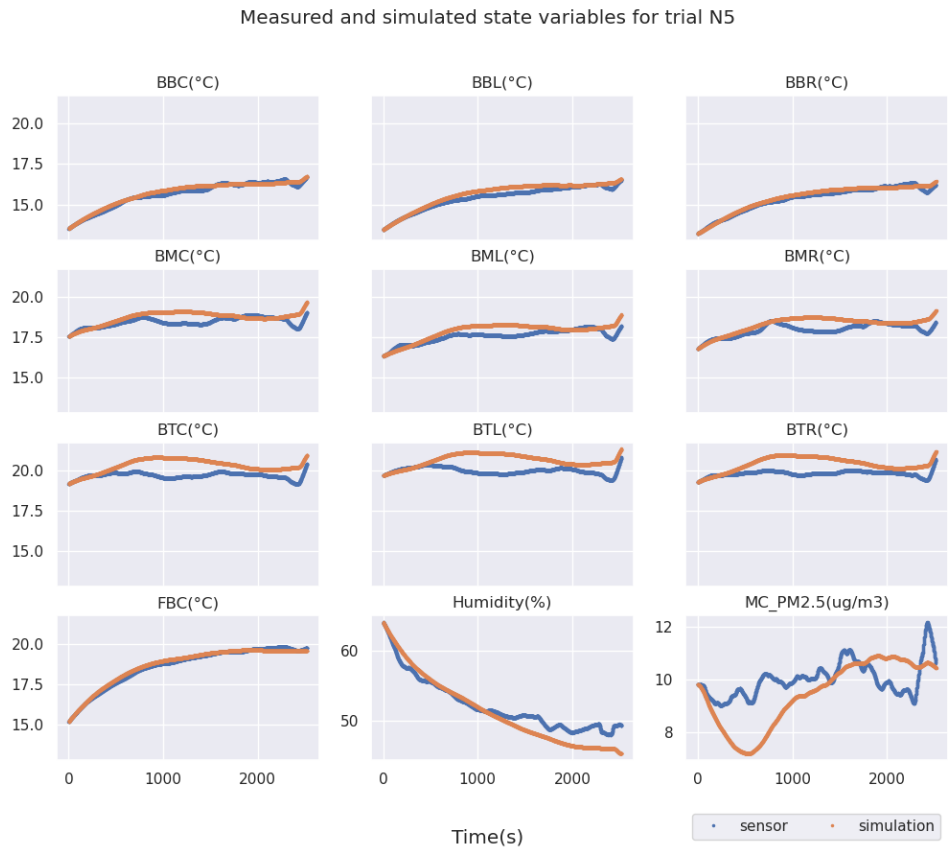
- Implement a fast realtime on vehicle ML and offline remote RL training
- Perform a field test with portable RL specific hardware capable of running the trained agent

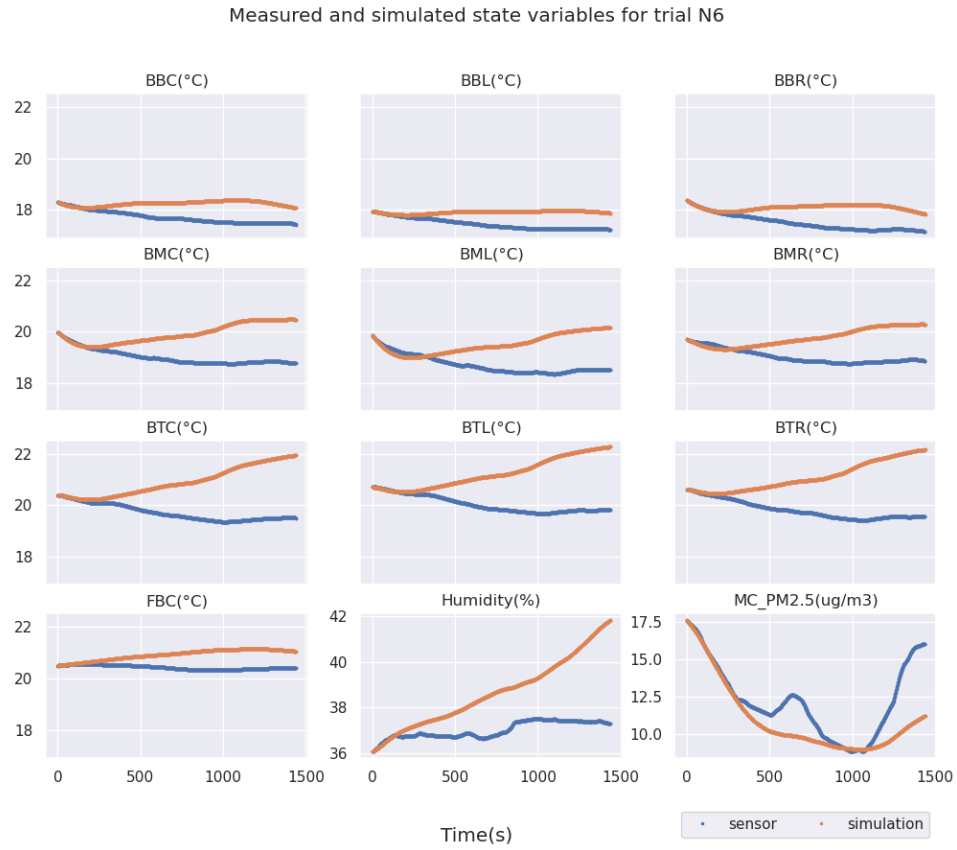
Appendices

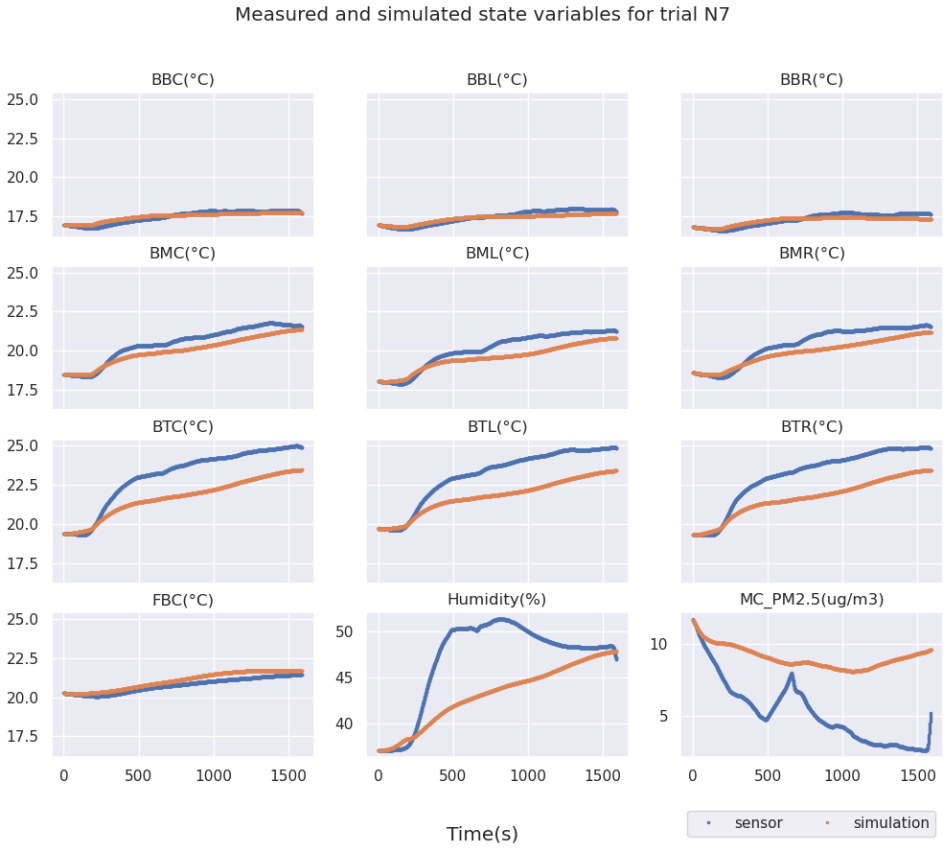
Appendix A

Appendix

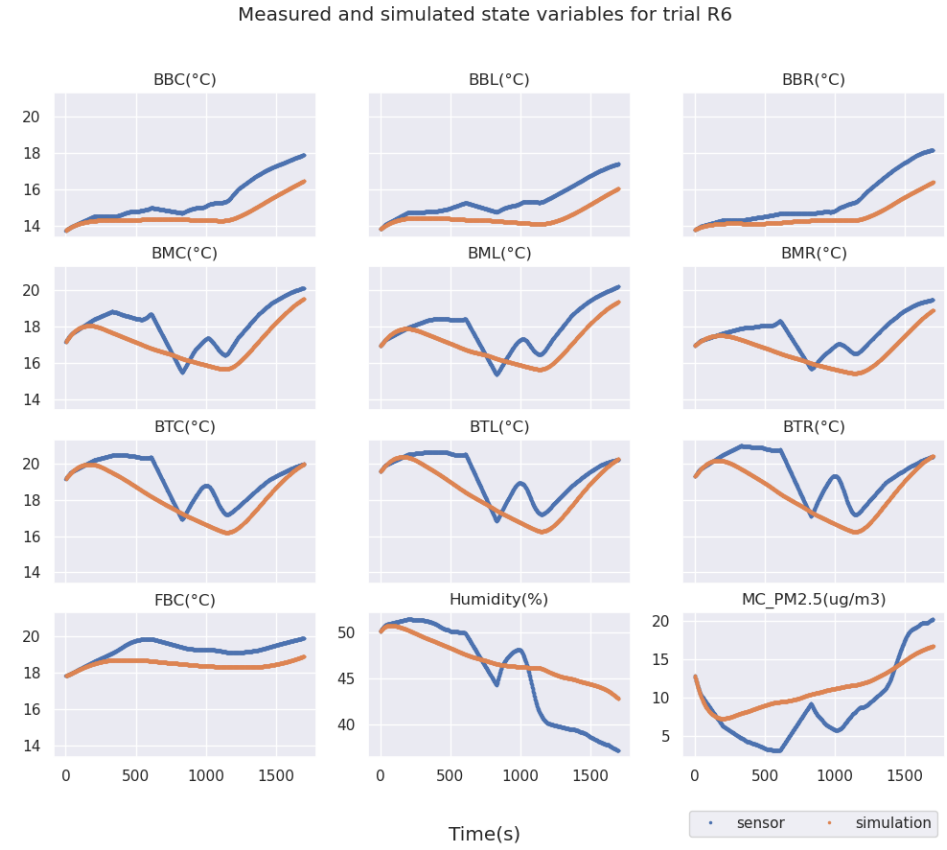
A.1 Appendix - Simulator results for all trials

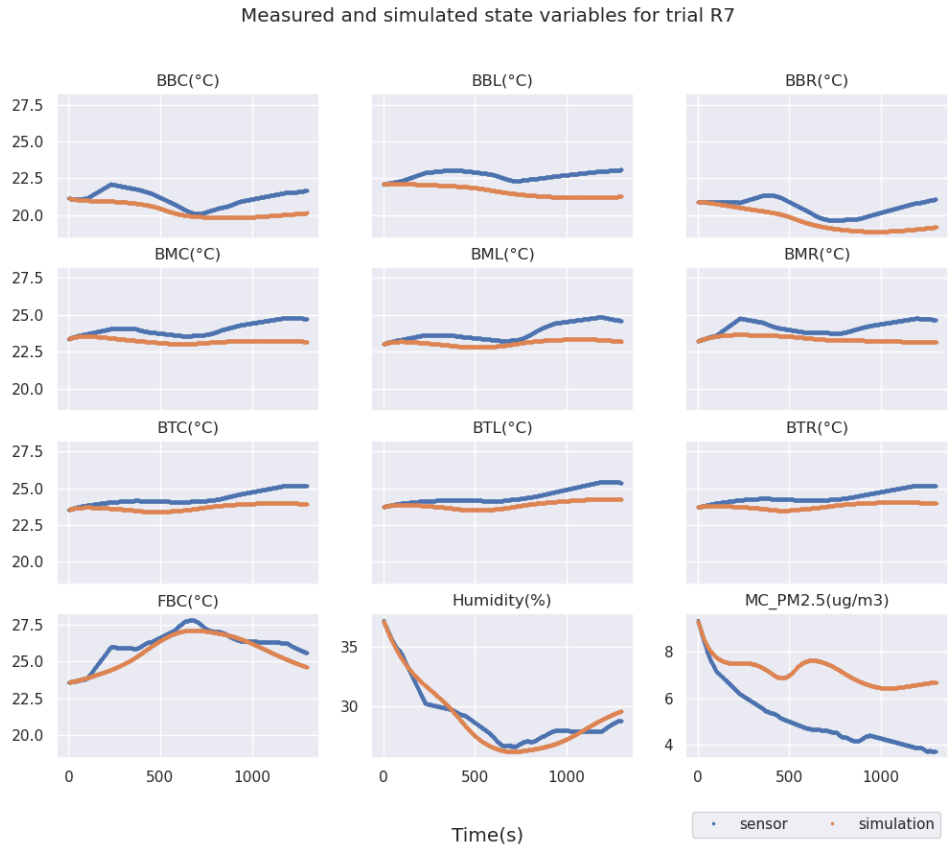


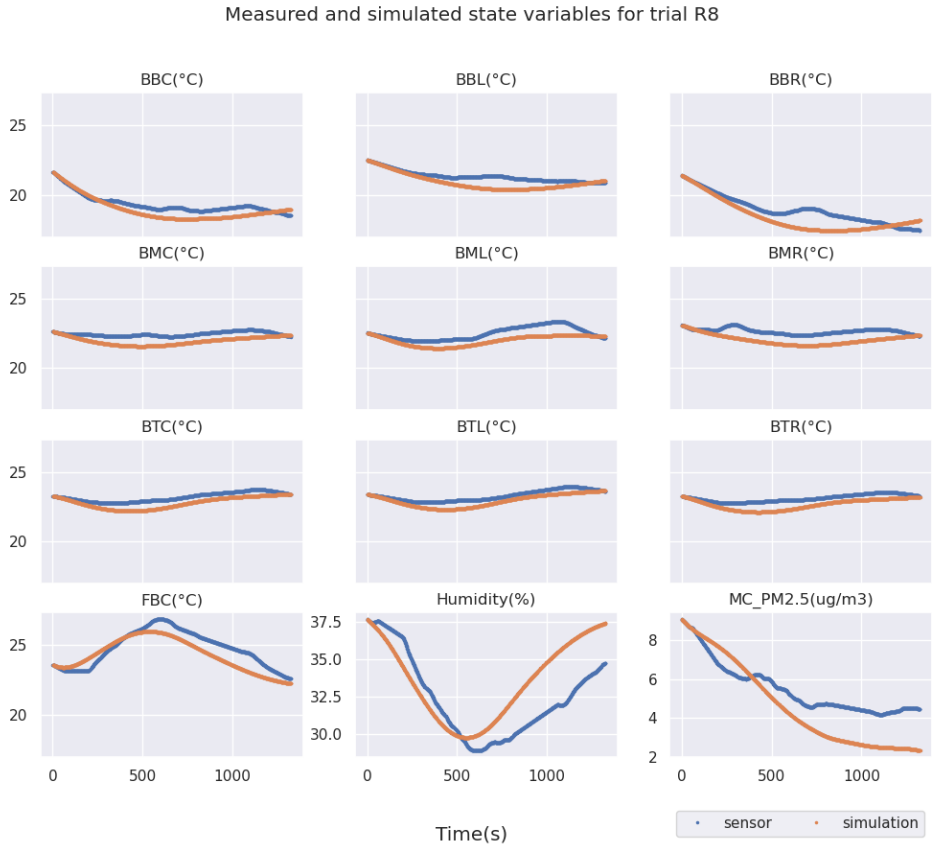

Measured and simulated state variables for trial N6
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial N7
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation

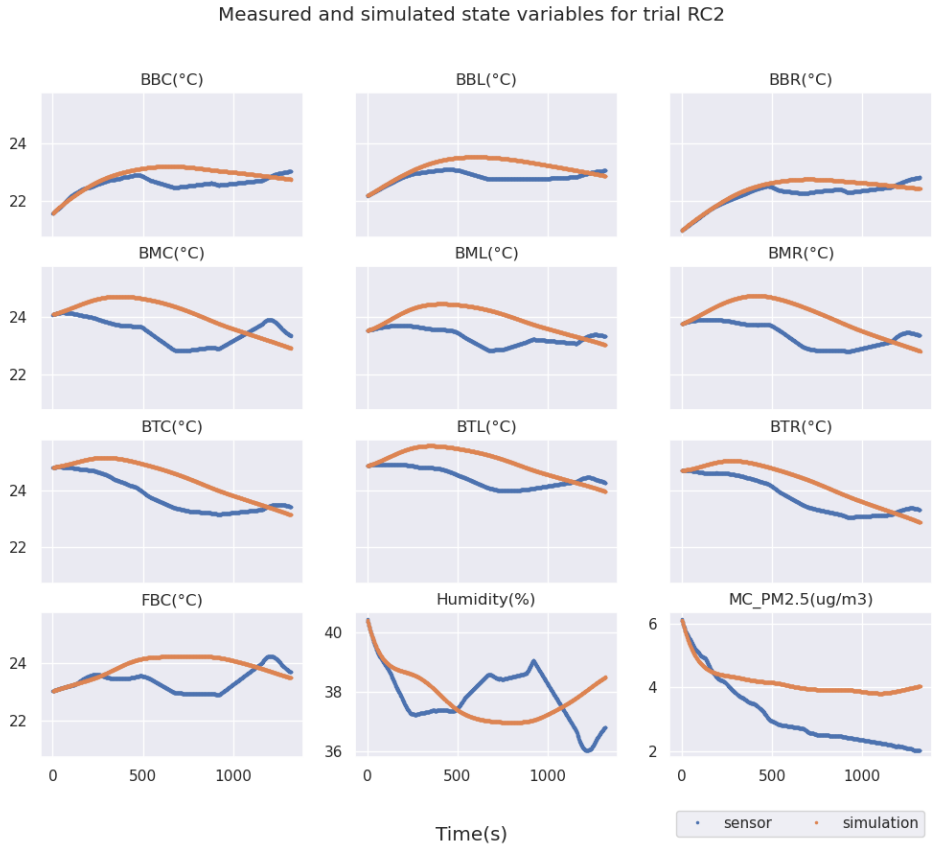

Measured and simulated state variables for trial N8
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation

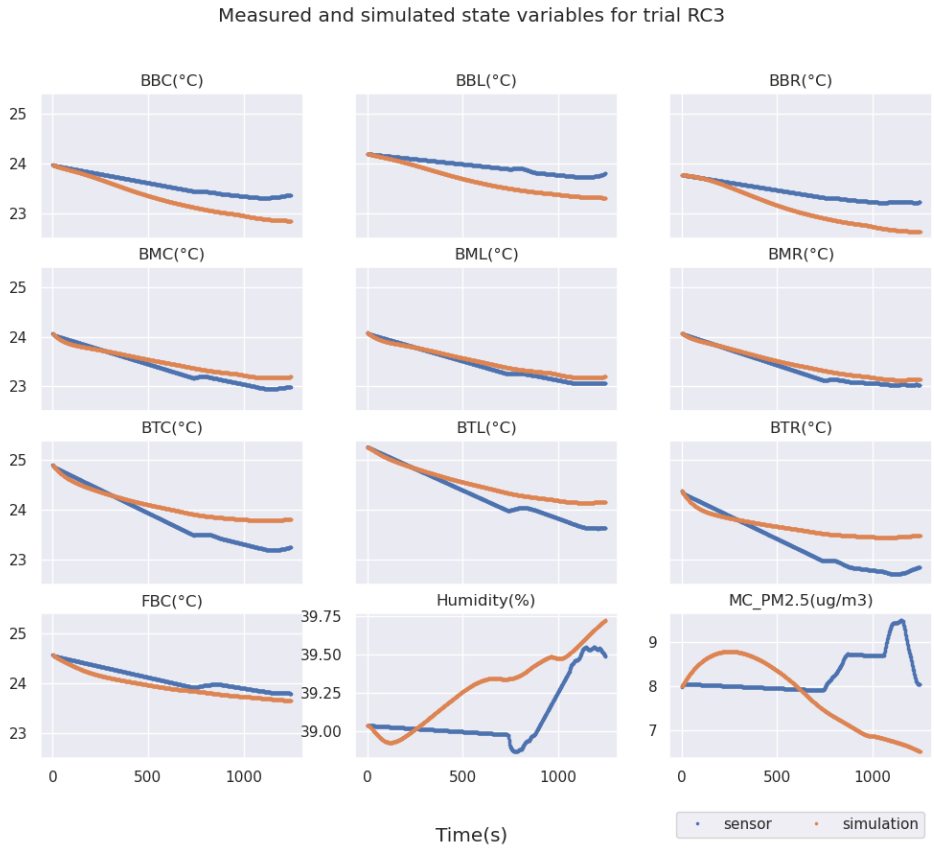

Measured and simulated state variables for trial R6
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation

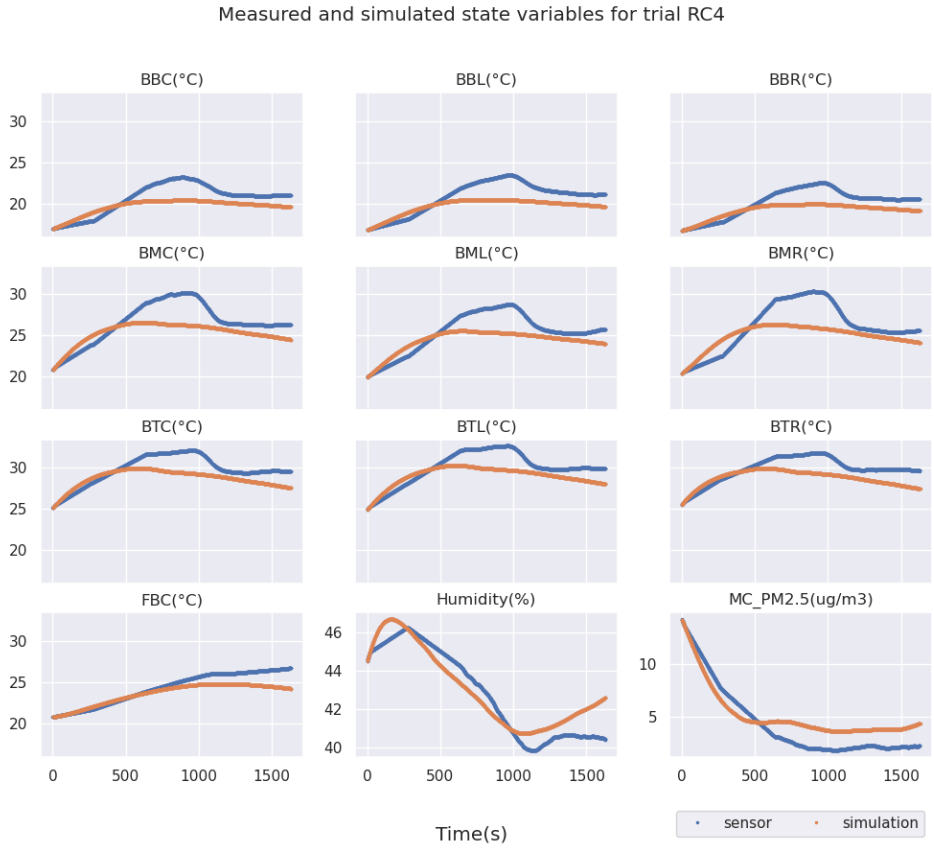

Measured and simulated state variables for trial R7
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation

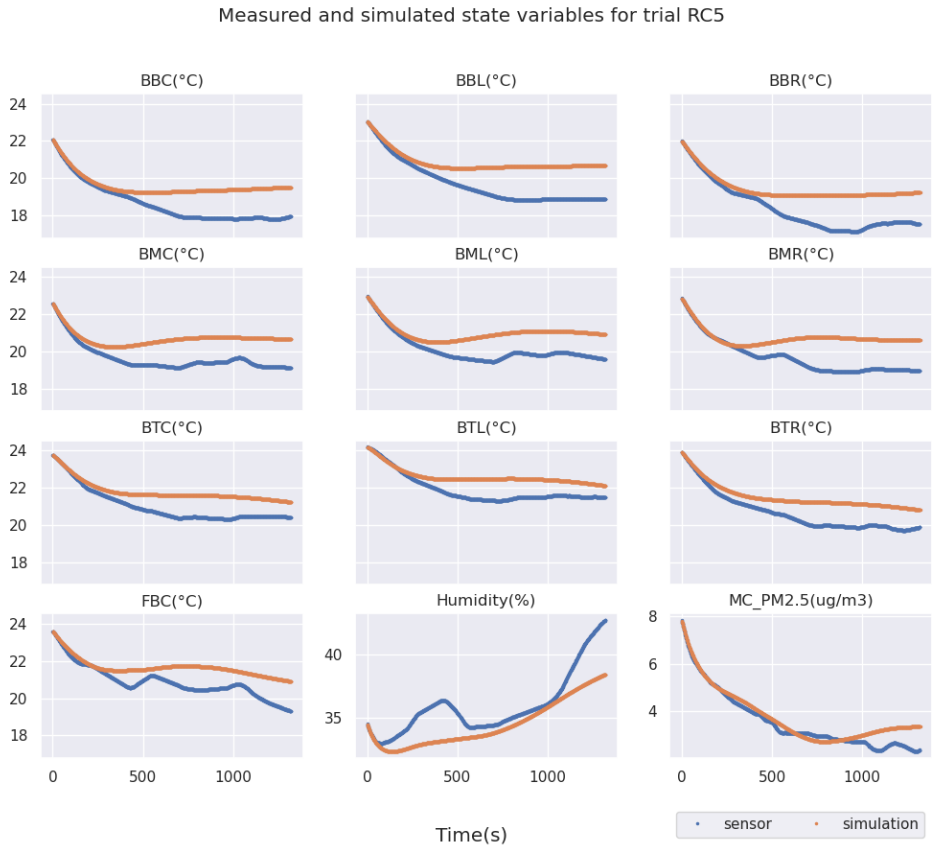

Measured and simulated state variables for trial R8
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC1
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC2
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC3
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC4
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC5
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC6
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC7
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation


Measured and simulated state variables for trial RC8
BBC(°C)
BBL(°C)
BBR(°C)
BMC(°C)
BML(°C)
BMR(°C)
BTC(°C)
BTL(°C)
BTR(°C)
FBC(°C)
Humidity(%)
MC_PM2.5(ug/m3)
Time(s)
sensor
simulation

A.2 Appendix - Code listings

```
'''
Import and prepare raw Leaf data into a form suitable for further
    processing.
Then saves the single trial dataframes in separate csv and pickle files.

Inspired from the domus project

@autor: Luigi Russi


HEATING OPERATION
'''

import os
import pickle
import numpy as np
from numpy.core.shape_base import block
import pandas as pd
from pandas.core import indexing
from pandas.core.indexes.base import Index
from pandas.tseries.offsets import Minute
from datetime import datetime, date, time, timezone, timedelta



def pp(folder, test_start, duration):
    '''
    Given a path to NasoInt, NasoExt, CANBUS logs it reads the raw data,
    aligns
    the datasets, slices according to test setup AC_on and REC. Then
    resamples
    secondly and merges the three dataframes.

    Parameters
    ----------
    path (string): path to single test raw data

    test_start (ISO datetime string YYYY-MM-DD hh:mm): time at which AC
    is turned on

    duration (int): test duration in minutes

```

```
    Returns
    -------

    TTT: pandas dataframe

    '''

    # define paths
    path_Ext = folder + "/LOG_NasoExt.CSV"
    path_Int = folder + "/LOG_NasoInt.CSV"
    path_Can = folder + "/LOG_CAN.csv"

    # define test setup
    duration = duration.tolist()  # ensure duration is of the type int
    test_start = pd.to_datetime(test_start)
    test_end = test_start + timedelta(minutes=duration)

    # import data from NasoExt SD card (datetime in CEST)
    def import_data_ext():
        raw_data_df = pd.read_csv(path_Ext, parse_dates=[
                                ['Date', 'Time']], sep=',', decimal=',',
    quotechar='"', engine='python', error_bad_lines=False)
        return raw_data_df

    TE = import_data_ext()
    TE.head()  # show the first 5 rows of the imported table

    # convert to numeric except datetime
    TE.rename({'Date_Time': 'DateTime'}, axis=1, inplace=True)
    TE[TE.columns[1:]].apply(pd.to_numeric, downcast='float', errors='
    coerce')

    # set correct datetime type
    TE.DateTime = pd.to_datetime(TE.DateTime, errors='coerce')

    TE['DateTime'] = TE['DateTime'] + timedelta(hours=-1, minutes=0)
    TE = TE.set_index(TE.DateTime)

    # for some reason humidity is not converted, also lat and long have
    wrong separator
    # let's force conversion, check with TE.info()

    TE[TE.columns[1:]] = TE[TE.columns[1:]].astype('float')

    # index the series object with datetime column
```

```
81      TE = TE.set_index(TE.DateTime)
82      TE.head()
83
84      # slice dataset
85      TE = TE.loc[(TE.index > test_start) & (TE.index < test_end)]
86
87      # use a rolling average for smoothing
88      # 5 seconds for this dataset (add automation for sample rate
    retrievement)
89      sample_rate = 5
90      roll = int(60/sample_rate)
91      TE_roll = TE.rolling(roll, center=True).mean().dropna()
92
93      # resample the smoothed dataset with secondly frequency
94      TE_mm = TE_roll.resample('1s').mean()
95      TE_mm = TE_mm.interpolate()
96
97      # import data from NasoInt SD card (datetime in CEST)
98      def import_data_int():
99          raw_data_df = pd.read_csv(path_Int, parse_dates=[
100                                  ['Date', 'Time']], sep=',', decimal=',',
    quotechar='"', engine='python', error_bad_lines=False)
101         return raw_data_df
102
103     TI = import_data_int()
104     TI.head()  # show the first 5 rows of the imported table
105
106     # import DS sensorserial
107
108     DSser = pd.read_csv('sensorserial.csv')
109     DSser.rename({0:'position',1:'serial'},inplace=True)
110     TI.rename({0:'serial'},inplace=True) # rename first line containing
    serials
111
112     # connect naming scheme to serials in TI
113     DSpos=pd.merge(DSser.T.astype(str),TI.T.astype(str),how='right',
    left_on='serial',right_on='serial')
114
115     # rename variables accordingly
116     DSpos['position'][23:43]
117     newcolname=DSpos['position'][23:43]
118     oldcolname=TI.columns[23:43]
119     raw_dict=oldcolname.to_series().reset_index().drop(columns='index').
    join(newcolname.reset_index().drop(columns='index'))
120     dict=raw_dict.set_index(0).to_dict()
```

```
121     TI.rename(columns=dict['position'],inplace=True)
122     TI.rename(columns={'Temp(C)':'FBC'},inplace=True)
123     # clean dataset
124     TI = TI.loc[:, TI.columns.notnull()] # drop nan column name
125     TI.drop('serial',inplace=True) # drop serial row
126
127     DS = ['FBC','BTR','BTC','BMR','BTL','BBR','BMC','BBC','BML','BBL']
128
129     # convert to numeric except datetime
130     TI.rename({'Date_Time': 'DateTime'}, axis=1, inplace=True)
131     TI[TI.columns[1:]].apply(pd.to_numeric, downcast='float', errors='
    coerce')
132
133     # set correct datetime type
134     TI.DateTime = pd.to_datetime(TI.DateTime, errors='coerce')
135
136     TI['DateTime'] = TI['DateTime']+timedelta(hours=-1, minutes=0)
137     TI = TI.set_index(TI.DateTime)
138
139     # for some reason humidity is not converted, also lat and long have
    wrong separator
140     # let's force conversion, check with TE.info()
141
142     TI[TI.columns[1:]] = TI[TI.columns[1:]].astype('float')
143
144     # index the series object with datetime column
145     TI = TI.set_index(TI.DateTime)
146
147     # compute mean over DS
148     TI['T_avg']=TI[DS].mean(axis=1)
149
150     TI.head()
151
152     # slice dataset
153     TI = TI.loc[(TI.index > test_start) & (TI.index < test_end)]
154
155     # use a rolling average for smoothing
156     # 5 seconds for this dataset (add automation for sample rate
    retrievement)
157     roll = int(60/sample_rate)
158     TI_roll = TI.rolling(roll, center=True).mean().dropna()
159
160     # resample the smoothed dataset with secondly frequency
161     TI_mm = TI_roll.resample('1s').mean()
162     TI_mm = TI_mm.interpolate()
```

```
163
164     # import data from CANBUS (datetime in UTC)
165     def import_data_canbus():
166         raw_data_df = pd.read_csv(path_Can, parse_dates=[
167                                 'DateTime'], sep=',', decimal='.',
    quotechar='"', engine='python', error_bad_lines=False)
168         return raw_data_df
169
170     TC = import_data_canbus()
171
172     TC['DateTime'] = TC['DateTime']+timedelta(hours=1)
173     TC = TC.set_index(TC.DateTime)
174
175     # slice dataset
176     TC = TC.loc[(TC.index > test_start) & (TC.index < test_end)]
177
178     # use a rolling average for smoothing
179     # 5 seconds for this dataset (add automation for sample rate
    retrievement)
180     roll = int(60/sample_rate)
181     TC_roll = TC.rolling(roll, center=True, min_periods=1).mean()
182
183     # resample the smoothed dataset with minutely frequency
184     TC_mm = TC_roll.resample('1s').mean()
185     TC_mm = TC_mm.interpolate()
186
187     # merge the datasets
188     TT = TE_mm.join(TI_mm, how='inner', lsuffix='_TE', rsuffix='_TI')
189     TTT = TT.join(TC_mm, how='inner', lsuffix='', rsuffix='_TA')
190
191     return TTT
192
193
194 # Define trials specs
195 trials = pd.DataFrame(
196     columns=['folder', 'trial', 'start_time', 'duration', 'setpoint', '
    distribution', 'fan'])
197
198 trials.loc[0] = ['Test_20211123', 'R6', '2021-11-23 17:49', 42, 20, '
    fresh', 4]
199 trials.loc[1] = ['Test_20211123', 'R7', '2021-11-23 18:34', 39, 20, 'rec
    ', 4]
200 trials.loc[2] = ['Test_20211124', 'R8', '2021-11-24 09:36', 44,  16,  '
    fresh', 4]
201 trials.loc[3] = ['Test_20211214', 'N5', '2021-12-14 17:00', 32,  16,  '
```

```
    rec', 4]
trials.loc[4] = ['Test_20211214', 'N6', '2021-12-14 17:33', 26,  16,  '
    rec', 1]
trials.loc[5] = ['Test_20211214', 'N7', '2021-12-14 18:00', 29, 20,  '
    rec', 1]
trials.loc[6] = ['Test_20211214', 'N8', '2021-12-14 18:30', 30,  25,  '
    rec', 4]


for i in range(len(trials)):

    # execute pp() function
    TTT = pp(trials.loc[i, 'folder'], trials.loc[i, 'start_time'],
    trials.loc[i, 'duration'])
    # save to pickle and csv
    pickle.dump(TTT, open('trials/' + trials.loc[i, 'trial'] + '-data.
    pickle', 'wb'))
    TTT.to_csv('trials/' + trials.loc[i, 'trial'] + '-data.csv',
    encoding='utf-8')


print('Done.')
```

Listing A.1: Import and prepare data from the LCSoS heating trials.

```
'''
Import and prepare raw Leaf data into a form suitable for further
    processing.
Then saves the single trial dataframes in separate csv and pickle files.

Inspired from the domus project

@autor: Luigi Russi

Make sure python process runs in this directory, use run-python from
    this buffer.

COOLING OPERATION
'''

import os
import pickle
import numpy as np
from numpy.core.shape_base import block
import pandas as pd
```

```
19 from pandas.core import indexing
20 from pandas.core.indexes.base import Index
21 from pandas.tseries.offsets import Minute
22 from datetime import datetime, date, time, timezone, timedelta
23
24
25
26 def pp(folder, test_start, duration):
27     '''
28     Given a path to NasoInt, NasoExt, CANBUS logs it reads the raw data,
       aligns
29     the datasets, slices according to test setup AC_on and REC. Then
       resamples
30     secondly and merges the three dataframes.
31
32     Parameters
33     ----------
34     path (string): path to single test raw data
35
36     test_start (ISO datetime string YYYY-MM-DD hh:mm): time at which AC
       is turned on
37
38     duration (int): test duration in minutes
39
40     Returns
41     -------
42
43     TTT: pandas dataframe
44
45     '''
46
47     # define paths
48     path_Ext = folder + "/LOG_NasoExt.CSV"
49     path_Int = folder + "/LOG_NasoInt.CSV"
50     path_Can = folder + "/LOG_CAN.csv"
51     path_Sun = folder + "/LOG_Sun.csv"
52
53     # define test setup
54     duration = duration.tolist()  # ensure duration is of the type int
55     test_start = pd.to_datetime(test_start)
56     test_end = test_start + timedelta(minutes=duration)
57
58     # import data from NasoExt SD card (datetime in CEST)
59     def import_data_ext():
60         raw_data_df = pd.read_csv(path_Ext, parse_dates=[
```

```
                                ['Date', 'Time']], sep=',', decimal=',',
quotechar='"', engine='python', error_bad_lines=False)
    return raw_data_df

TE = import_data_ext()
TE.head()  # show the first 5 rows of the imported table

# convert to numeric except datetime
TE.rename({'Date_Time': 'DateTime'}, axis=1, inplace=True)
TE[TE.columns[1:]].apply(pd.to_numeric, downcast='float', errors='
coerce')

# set correct datetime type
TE.DateTime = pd.to_datetime(TE.DateTime, errors='coerce')

TE['DateTime'] = TE['DateTime'] + timedelta(hours=0, minutes=0)
TE = TE.set_index(TE.DateTime)

# for some reason humidity is not converted, also lat and long have
wrong separator
# let's force conversion, check with TE.info()

TE[TE.columns[1:]] = TE[TE.columns[1:]].astype('float')

# index the series object with datetime column
TE = TE.set_index(TE.DateTime)
TE.head()

# slice dataset
TE = TE.loc[(TE.index > test_start) & (TE.index < test_end)]

# use a rolling average for smoothing
# 5 seconds for this dataset (add automation for sample rate
retrievement)
sample_rate = 5
roll = int(60/sample_rate)
TE_roll = TE.rolling(roll, center=True).mean().dropna()

# resample the smoothed dataset with secondly frequency
TE_mm = TE_roll.resample('1s').mean()
TE_mm = TE_mm.interpolate()

# import data from NasoInt SD card (datetime in CEST)
def import_data_int():
    raw_data_df = pd.read_csv(path_Int, parse_dates=[
```

```
                              ['Date', 'Time']], sep=',', decimal=',',
quotechar='"', engine='python', error_bad_lines=False)
     return raw_data_df

TI = import_data_int()
TI.head()  # show the first 5 rows of the imported table

# import DS sensorserial

DSser = pd.read_csv('sensorserial.csv')
DSser.rename({0:'position',1:'serial'},inplace=True)
TI.rename({0:'serial'},inplace=True) # rename first line containing
serials

# connect naming scheme to serials in TI
DSpos=pd.merge(DSser.T.astype(str),TI.T.astype(str),how='right',
left_on='serial',right_on='serial')

# rename variables accordingly
DSpos['position'][23:43]
newcolname=DSpos['position'][23:43]
oldcolname=TI.columns[23:43]
raw_dict=oldcolname.to_series().reset_index().drop(columns='index').
join(newcolname.reset_index().drop(columns='index'))
dict=raw_dict.set_index(0).to_dict()
TI.rename(columns=dict['position'],inplace=True)
TI.rename(columns={'Temp(C)':'FBC'},inplace=True)
# clean dataset
TI = TI.loc[:, TI.columns.notnull()] # drop nan column name
TI.drop('serial',inplace=True) # drop serial row

DS = ['FBC','BTR','BTC','BMR','BTL','BBR','BMC','BBC','BML','BBL']

# convert to numeric except datetime
TI.rename({'Date_Time': 'DateTime'}, axis=1, inplace=True)
TI[TI.columns[1:]].apply(pd.to_numeric, downcast='float', errors='
coerce')

# set correct datetime type
TI.DateTime = pd.to_datetime(TI.DateTime, errors='coerce')

TI['DateTime'] = TI['DateTime']+timedelta(hours=0, minutes=0)
TI = TI.set_index(TI.DateTime)

# for some reason humidity is not converted, also lat and long have
```

```
wrong separator
 # let's force conversion, check with TE.info()

 TI[TI.columns[1:]] = TI[TI.columns[1:]].astype('float')

 # index the series object with datetime column
 TI = TI.set_index(TI.DateTime)

 # compute mean over DS
 TI['T_avg'] = TI[DS].mean(axis=1)

 TI.head()

 # slice dataset
 TI = TI.loc[(TI.index > test_start) & (TI.index < test_end)]

 # use a rolling average for smoothing
 # 5 seconds for this dataset (add automation for sample rate
retrievement)
 roll = int(60/sample_rate)
 TI_roll = TI.rolling(roll, center=True).mean().dropna()

 # resample the smoothed dataset with secondly frequency
 TI_mm = TI_roll.resample('1s').mean()
 TI_mm = TI_mm.interpolate()

 # import data from CANBUS (datetime in UTC)
 def import_data_canbus():
     raw_data_df = pd.read_csv(path_Can, sep=';', decimal='.',
quotechar='"', engine='python', error_bad_lines=False)
     return raw_data_df

 # drop nan columns
 TC = import_data_canbus().dropna(axis=1)

 # set corret datetime import format
 TC['DateTime'] = pd.to_datetime(TC.loc[:, 'DateTime'], format='%d-%m
-%Y %H:%M:%S.%f')

 # adjust to Bologna local time
 TC['DateTime'] = TC['DateTime']+timedelta(hours=+2)
 TC = TC.set_index(TC.DateTime).sort_index(ascending=True)

 # slice dataset
 TC = TC.loc[(TC.index > test_start) & (TC.index < test_end)]
```

```
183
184     # use a rolling average for smoothing
185     # 5 seconds for this dataset (add automation for sample rate
    retrievement)
186     roll = int(60/sample_rate)
187     TC_roll = TC.rolling(roll, center=True, min_periods=1).mean()
188
189     # resample the smoothed dataset with secondly frequency
190     TC_mm = TC_roll.resample('1s').mean()
191     TC_mm = TC_mm.interpolate()
192
193     # import data from Solar sensor (datetime in CEST)
194     def import_data_solar():
195         raw_data_df = pd.read_csv(path_Sun, parse_dates=[
196             ['Date','Time']], usecols=['Date','Time','Irr'], sep=',',
    quotechar='"', engine='c', on_bad_lines='skip')
197         return raw_data_df
198
199     TS = import_data_solar().dropna()
200
201     # set correct datetime type
202     datetime = pd.to_datetime(TS.loc[:, 'Date_Time'], format='%d.%m.%y %
    H:%M:%S')
203     TS['DateTime'] = datetime
204
205     # convert to numeric except datetime
206     Irr = TS['Irr'].str.replace('<', '').apply(pd.to_numeric, downcast='
    float', errors='coerce')
207     TS['Irr'] = Irr
208
209     TS['DateTime'] = TS['DateTime']+timedelta(hours=0, minutes=55)
210     TS = TS.set_index(TS.DateTime)
211
212     TS = TS.drop(columns=['Date_Time', 'DateTime'])
213
214     # slice dataset
215     TS = TS.loc[(TS.index > test_start) & (TS.index < test_end)]
216
217     # use a rolling average for smoothing
218     # 60 seconds for this dataset (add automation for sample rate
    retrievement)
219     roll = int(3)  # rolling average over 5 minutes
220     TS_roll = TS.rolling(roll, center=True, min_periods=1).mean()
221
222     # resample the smoothed dataset with secondly frequency
```

```
223     TS_mm = TS_roll.resample('1s').mean()
224     TS_mm = TS_mm.interpolate()
225 
226     # merge the datasets
227     TT = TE_mm.join(TI_mm, how='inner', lsuffix='_TE', rsuffix='_TI')
228     TTT = TT.join(TC_mm, how='inner', lsuffix='', rsuffix='_TC')
229     TTTT = TTT.join(TS_mm, how='inner', lsuffix='', rsuffix='TS')
230 
231     return TTTT, TTT, TT, TE, TI, TC, TS
232 
233 
234 # Define trials specs
235 trials = pd.DataFrame(
236     columns=['folder', 'trial', 'start_time', 'duration', 'setpoint', '
    distribution', 'fan'])
237 
238 trials.loc[0] = ['Test_20221007', 'RC1', '2022-10-07 12:56', 27, 16, '
    rec', 1]
239 trials.loc[1] = ['Test_20221007', 'RC2', '2022-10-07 13:28', 25, 16, '
    fresh', 1]
240 trials.loc[2] = ['Test_20221007', 'RC3', '2022-10-07 13:58', 25, 20, '
    rec', 1]
241 trials.loc[3] = ['Test_20221007', 'RC4', '2022-10-07 14:31', 26, 20, '
    fresh', 1]
242 trials.loc[4] = ['Test_20221007', 'RC5', '2022-10-07 15:02', 25, 16, '
    rec', 4]
243 trials.loc[5] = ['Test_20221007', 'RC6', '2022-10-07 15:37', 23, 16, '
    fresh', 4]
244 trials.loc[6] = ['Test_20221007', 'RC7', '2022-10-07 16:06', 28, 20, '
    rec', 4]
245 trials.loc[7] = ['Test_20221007', 'RC8', '2022-10-07 16:45', 25, 20, '
    fresh', 4]
246 
247 
248 for i in range(len(trials)):
249 
250     # execute pp() function
251     # return all the values but save only TTTT in each loop
252     TTTT, TTT, TT, TE, TI, TC, TS = pp(trials.loc[i, 'folder'], trials.
    loc[i, 'start_time'], trials.loc[i, 'duration'])
253     # save to pickle and csv
254     pickle.dump(TTTT, open('../trials/' + trials.loc[i, 'trial'] + '-
    data.pickle', 'wb'))
255     TTTT.to_csv('../trials/' + trials.loc[i, 'trial'] + '-data.csv',
    encoding='utf-8')
```

```
256
257
258 print('Done.')
```

Listing A.2: Import and prepare data from the LCSoS heating trials.

A.3 Appendix - Domus code listings

THe following codes from the Domus project were used or adapted to this work's study case [23; Bru21b; Bru21a]:

```
1  """harness.py
2
3  Author
4  ------
5  J. Brusey
6
7  Date
8  ----
9  May 27, 2021
10
11 Description
12 -----------
13
14 Connect simulators and controller together and run for a certain
15 number of timesteps under specific conditions.
16
17 """
18
19
20 import joblib
21 import numpy as np
22 import pkg_resources
23
24 from .cols import (
25     DV0_UT_COLUMNS,
26     DV0_UT_MAX,
27     DV0_UT_MIN,
28     DV0_XT_COLUMNS,
29     DV1_UT_COLUMNS,
30     DV1_UT_MAX,
31     DV1_UT_MIN,
32     DV1_XT_COLUMNS,
33     HVAC_UT_COLUMNS,
34     HVAC_UT_MAX,
35     HVAC_UT_MIN,
```

```
    HVAC_XT_COLUMNS,
    DV0Ut,
    DV0Xt,
    DV1Ut,
    DV1Xt,
    HvacUt,
    HvacXt,
)
from .mlsim import MLSim
from .simple_hvac import SimpleHvac
from .util import kw_to_array


def estimate_cabin_temperature_dv0(b_x):
    """estimate the cabin temperature based on the average front bench temperatures.
    Assumes DV0 model
    """
    assert len(b_x) == len(DV0Xt)
    return np.mean(
        b_x[
            [
                DV0Xt.t_drvr1,
                DV0Xt.t_drvr2,
                DV0Xt.t_drvr3,
                DV0Xt.t_psgr1,
                DV0Xt.t_psgr2,
                DV0Xt.t_psgr3,
                DV0Xt.m_drvr1,
                DV0Xt.m_drvr2,
                DV0Xt.m_drvr3,
                DV0Xt.m_psgr1,
                DV0Xt.m_psgr2,
                DV0Xt.m_psgr3,
            ]
        ]
    )


def estimate_cabin_temperature_dv1(b_x):
    """estimate the cabin temperature based on the average front bench temperatures.
    Assumes DV1 model
    """
    assert len(b_x) == len(DV1Xt)
```

```
79      return np.mean(
80          b_x[
81              [
82                  DV1Xt.t_drvr1,
83                  DV1Xt.t_drvr2,
84                  DV1Xt.t_drvr3,
85                  DV1Xt.t_psgr1,
86                  DV1Xt.t_psgr2,
87                  DV1Xt.t_psgr3,
88                  DV1Xt.m_drvr1,
89                  DV1Xt.m_drvr2,
90                  DV1Xt.m_drvr3,
91                  DV1Xt.m_psgr1,
92                  DV1Xt.m_psgr2,
93                  DV1Xt.m_psgr3,
94              ]
95          ]
96      )
97
98
99  def update_control_inputs_dv0(c_u, b_x, h_x, cab_t):
100     """update control vector based on cabin, hvac, and front bench
        temperatures.
101     Assumes DV0 model.
102     """
103     c_u[SimpleHvac.Ut.cabin_temperature] = cab_t
104
105     c_u[SimpleHvac.Ut.window_temperature] = b_x[DV0Xt.ws]
106
107     c_u[[SimpleHvac.Ut.cabin_humidity, SimpleHvac.Ut.vent_temperature]]
        = h_x[
108         [HvacXt.cab_RH, HvacXt.vent_T]
109     ]
110
111
112 def update_control_inputs_dv1(c_u, b_x, h_x, cab_t):
113     """update control vector based on cabin, hvac, and front bench
        temperatures.
114     Assumes DV1 model.
115     """
116     c_u[SimpleHvac.Ut.cabin_temperature] = cab_t
117
118     c_u[SimpleHvac.Ut.window_temperature] = b_x[DV1Xt.ws]
119
```

```
120     c_u[[SimpleHvac.Ut.cabin_humidity, SimpleHvac.Ut.vent_temperature]]
    = h_x[
121         [HvacXt.cab_RH, HvacXt.vent_T]
122     ]
123
124
125 def update_hvac_inputs(h_u, c_x, cab_t):
126     """update hvac input vector based on control, cabin, and front bench
    temperatures."""
127     h_u[HvacUt.cab_T] = cab_t
128
129     h_u[
130         [
131             HvacUt.blw_power,
132             HvacUt.cmp_power,
133             HvacUt.hv_heater,
134             HvacUt.fan_power,
135             HvacUt.recirc,
136         ]
137     ] = c_x[
138         [
139             SimpleHvac.Xt.blower_level,
140             SimpleHvac.Xt.compressor_power,
141             SimpleHvac.Xt.heater_power,
142             SimpleHvac.Xt.fan_power,
143             SimpleHvac.Xt.recirc,
144         ]
145     ]
146
147
148 def update_dv0_inputs(b_u, h_x, c_x):
149     """update dv0 input vector b_u based on hvac state h_x and control
    state c_x."""
150     b_u[[DV0Ut.t_HVACMain, DV0Ut.v_HVACMain,]] = h_x[
151         [
152             HvacXt.vent_T,
153             HvacXt.evp_mdot,
154         ]
155     ]
156
157     b_u[[DV0Ut.recirc, DV0Ut.dist_defrost]] = c_x[
158         [SimpleHvac.Xt.recirc, SimpleHvac.Xt.dist_defrost]
159     ]
160
161
```

```
162 def update_dv1_inputs(b_u, h_x, c_x):
163     """update dv1 input vector b_u based on hvac state h_x and control
    state c_x."""
164     b_u[
165         [
166             DV1Ut.new_air_mode_Floor_SO_Defrost,
167             DV1Ut.seat_off,
168         ]
169     ] = [1, 1]
170 
171     b_u[[DV1Ut.HvacMain,]] = h_x[
172         [
173             HvacXt.vent_T,
174         ]
175     ]
176 
177     b_u[[DV1Ut.recirc, DV1Ut.window_heating, DV1Ut.dist_defrost]] = c_x[
178         [SimpleHvac.Xt.recirc, SimpleHvac.Xt.window_heating, SimpleHvac.
    Xt.dist_defrost]
179     ]
180     # simplification to get dv1 working
181     b_u[DV1Ut.vent_flow_rate] = np.interp(
182         c_x[SimpleHvac.Xt.blower_level],
183         np.array([5, 10, 18], dtype=np.float32) * 17 + 94,
184         np.array([1, 3, 5], dtype=np.float32),
185     )
186 
187 
188 def load_dv0():
189     fname = pkg_resources.resource_filename(__name__, "model/3d_lr.
    joblib")
190     scaler_and_model = joblib.load(fname)
191 
192     return scaler_and_model
193 
194 
195 def load_dv1():
196     fname = pkg_resources.resource_filename(__name__, "model/dv1_lr.
    joblib")
197     scaler_and_model = joblib.load(fname)
198 
199     return scaler_and_model
200 
201 
202 def load_hvac():
```

```
203     fname = pkg_resources.resource_filename(__name__, "model/hvac_lr.
    joblib")
204     scaler_and_model = joblib.load(fname)
205 
206     return scaler_and_model
207 
208 
209 def make_dv0_sim(scaler_and_model, cabin_state):
210     return MLSim(
211         scaler_and_model,
212         initial_state=np.vstack([cabin_state] * 2),
213         xlag=2,
214         ulag=2,
215         xlen=len(DV0_XT_COLUMNS),
216         ulen=len(DV0_UT_COLUMNS),
217         ut_min=DV0_UT_MIN,
218         ut_max=DV0_UT_MAX,
219     )
220 
221 
222 def make_dv1_sim(scaler_and_model, cabin_state):
223     return MLSim(
224         scaler_and_model,
225         initial_state=np.vstack([cabin_state]),
226         xlag=1,
227         ulag=1,
228         xlen=len(DV1_XT_COLUMNS),
229         ulen=len(DV1_UT_COLUMNS),
230         ut_min=DV1_UT_MIN,
231         ut_max=DV1_UT_MAX,
232     )
233 
234 
235 def make_hvac_sim(scaler_and_model, hvac_state):
236     return MLSim(
237         scaler_and_model,
238         initial_state=np.vstack([hvac_state]),
239         xlag=1,
240         ulag=1,
241         xlen=len(HVAC_XT_COLUMNS),
242         ulen=len(HVAC_UT_COLUMNS),
243         ut_min=HVAC_UT_MIN,
244         ut_max=HVAC_UT_MAX,
245     )
246 
```

```
247
248 def run_dv0_sim(
249     dv0_scaler_and_model,
250     hvac_scaler_and_model,
251     controller,
252     setpoint,
253     n,
254     ambient_t,
255     ambient_rh,
256     cabin_t,
257     cabin_v,
258     cabin_rh,
259     solar1,
260     solar2,
261     car_speed,
262     log_inputs=False,
263 ):
264     b_x = kw_to_array(
265         DV0_XT_COLUMNS,
266         t_drvr1=cabin_t,
267         t_drvr2=cabin_t,
268         t_drvr3=cabin_t,
269         t_psgr1=cabin_t,
270         t_psgr2=cabin_t,
271         t_psgr3=cabin_t,
272         t_psgr21=cabin_t,
273         t_psgr22=cabin_t,
274         t_psgr23=cabin_t,
275         t_psgr31=cabin_t,
276         t_psgr32=cabin_t,
277         t_psgr33=cabin_t,
278         v_drvr1=cabin_v,
279         v_drvr2=cabin_v,
280         v_drvr3=cabin_v,
281         v_psgr1=cabin_v,
282         v_psgr2=cabin_v,
283         v_psgr3=cabin_v,
284         v_psgr21=cabin_v,
285         v_psgr22=cabin_v,
286         v_psgr23=cabin_v,
287         v_psgr31=cabin_v,
288         v_psgr32=cabin_v,
289         v_psgr33=cabin_v,
290         m_drvr1=cabin_t,
291         m_drvr2=cabin_t,
```

```
292         m_drvr3=cabin_t,
293         m_psgr1=cabin_t,
294         m_psgr2=cabin_t,
295         m_psgr3=cabin_t,
296         m_psgr21=cabin_t,
297         m_psgr22=cabin_t,
298         m_psgr23=cabin_t,
299         m_psgr31=cabin_t,
300         m_psgr32=cabin_t,
301         m_psgr33=cabin_t,
302         rhc=cabin_rh,
303         ws=cabin_t,
304     )
305     h_x = kw_to_array(
306         HVAC_XT_COLUMNS, cab_RH=cabin_rh, evp_mdot=cabin_v, vent_T=
    cabin_t
307     )
308
309     cabin_mlsim = make_dv0_sim(dv0_scaler_and_model, b_x)
310
311     hvac_mlsim = make_hvac_sim(hvac_scaler_and_model, h_x)
312
313     cabin = np.zeros((n, len(b_x)), dtype=np.float32)
314     cabin[0] = b_x
315     hvac = np.zeros((n, len(h_x)), dtype=np.float32)
316     hvac[0] = h_x
317     ctrl = np.zeros((n, len(SimpleHvac.Xt)), dtype=np.float32)
318     c_u = np.zeros((len(controller.Ut)), dtype=np.float32)
319     c_u[controller.Ut.setpoint] = setpoint
320     h_u = np.zeros((len(HvacUt)), dtype=np.float32)
321     h_u[[HvacUt.ambient, HvacUt.humidity, HvacUt.solar, HvacUt.speed]] =
    [
322         ambient_t,
323         ambient_rh,
324         solar1,
325         car_speed,
326     ]
327     b_u = np.zeros((len(DV0Ut)), dtype=np.float32)
328     b_u[
329         [
330             DV0Ut.t_a,
331             DV0Ut.rh_a,
332             DV0Ut.rad1,
333             DV0Ut.rad2,
334             DV0Ut.VehicleSpeed,
```

```
335         ]
336     ] = [ambient_t, ambient_rh, solar1, solar2, car_speed / 100 *
    27.778]
337     if log_inputs:
338         b_u_log = np.zeros((n, len(b_u)), dtype=np.float32)
339         h_u_log = np.zeros((n, len(h_u)), dtype=np.float32)
340         c_u_log = np.zeros((n, len(c_u)), dtype=np.float32)
341     for i in range(n):
342         # average temperature over front bench
343         cab_t = estimate_cabin_temperature_dv0(b_x)
344         update_control_inputs_dv0(c_u, b_x, h_x, cab_t)
345         #     print(c_u, cab_t)
346         c_x = controller.step(c_u)
347 
348         # drive HVAC
349         update_hvac_inputs(h_u, c_x, cab_t)
350         #     print(h_u)
351         _, h_x = hvac_mlsim.step(h_u)
352         #     print(h_x)
353         # if h_x[HvacXt.evp_mdot] < 0:
354         #     h_x[HvacXt.evp_mdot] = 0
355         # drive cabin
356         update_dv0_inputs(b_u, h_x, c_x)
357         _, b_x = cabin_mlsim.step(b_u)
358         cabin[i] = b_x
359         hvac[i] = h_x
360         ctrl[i] = c_x
361         if log_inputs:
362             b_u_log[i] = b_u
363             h_u_log[i] = h_u
364             c_u_log[i] = c_u
365 
366     if log_inputs:
367         return cabin, hvac, ctrl, b_u_log, h_u_log, c_u_log
368     else:
369         return cabin, hvac, ctrl
370 
371 
372 def run_dv1_sim(
373     dv1_scaler_and_model,
374     hvac_scaler_and_model,
375     controller,
376     setpoint,
377     n,
378     ambient_t,
```

```
379     ambient_rh,
380     cabin_t,
381     cabin_v,
382     cabin_rh,
383     solar1,
384     solar2,
385     car_speed,
386     log_inputs=False,
387 ):
388     b_x = kw_to_array(
389         DV1_XT_COLUMNS,
390         t_drvr1=cabin_t,
391         t_drvr2=cabin_t,
392         t_drvr3=cabin_t,
393         t_psgr1=cabin_t,
394         t_psgr2=cabin_t,
395         t_psgr3=cabin_t,
396         t_psgr21=cabin_t,
397         t_psgr22=cabin_t,
398         t_psgr23=cabin_t,
399         t_psgr31=cabin_t,
400         t_psgr32=cabin_t,
401         t_psgr33=cabin_t,
402         v_drvr1=cabin_v,
403         v_drvr2=cabin_v,
404         v_drvr3=cabin_v,
405         v_psgr1=cabin_v,
406         v_psgr2=cabin_v,
407         v_psgr3=cabin_v,
408         v_psgr21=cabin_v,
409         v_psgr22=cabin_v,
410         v_psgr23=cabin_v,
411         v_psgr31=cabin_v,
412         v_psgr32=cabin_v,
413         v_psgr33=cabin_v,
414         m_drvr1=cabin_t,
415         m_drvr2=cabin_t,
416         m_drvr3=cabin_t,
417         m_psgr1=cabin_t,
418         m_psgr2=cabin_t,
419         m_psgr3=cabin_t,
420         m_psgr21=cabin_t,
421         m_psgr22=cabin_t,
422         m_psgr23=cabin_t,
423         m_psgr31=cabin_t,
```

```
424         m_psgr32=cabin_t,
425         m_psgr33=cabin_t,
426         rhc=cabin_rh,
427         ws=cabin_t,
428     )
429     h_x = kw_to_array(
430         HVAC_XT_COLUMNS, cab_RH=cabin_rh, evp_mdot=cabin_v, vent_T=
    cabin_t
431     )
432
433     cabin_mlsim = make_dv1_sim(dv1_scaler_and_model, b_x)
434
435     hvac_mlsim = make_hvac_sim(hvac_scaler_and_model, h_x)
436
437     cabin = np.zeros((n, len(b_x)), dtype=np.float32)
438     cabin[0] = b_x
439     hvac = np.zeros((n, len(h_x)), dtype=np.float32)
440     hvac[0] = h_x
441     ctrl = np.zeros((n, len(SimpleHvac.Xt)), dtype=np.float32)
442     c_u = np.zeros((len(controller.Ut)), dtype=np.float32)
443     c_u[controller.Ut.setpoint] = setpoint
444     h_u = np.zeros((len(HvacUt)), dtype=np.float32)
445     h_u[[HvacUt.ambient, HvacUt.humidity, HvacUt.solar, HvacUt.speed]] =
    [
446         ambient_t,
447         ambient_rh,
448         solar1,
449         car_speed,
450     ]
451     b_u = np.zeros((len(DV1Ut)), dtype=np.float32)
452     b_u[
453         [
454             DV1Ut.t_a,
455             DV1Ut.rh_a,
456             DV1Ut.rad1,
457             DV1Ut.rad2,
458             DV1Ut.VehicleSpeed,
459         ]
460     ] = [ambient_t, ambient_rh, solar1, solar2, car_speed / 100 *
    27.778]
461
462     if log_inputs:
463         b_u_log = np.zeros((n, len(b_u)), dtype=np.float32)
464         h_u_log = np.zeros((n, len(h_u)), dtype=np.float32)
465         c_u_log = np.zeros((n, len(c_u)), dtype=np.float32)
```

```
466     for i in range(n):
467         # average temperature over front bench
468         cab_t = estimate_cabin_temperature_dv1(b_x)
469         update_control_inputs_dv1(c_u, b_x, h_x, cab_t)
470         c_x = controller.step(c_u)
471 
472         # drive HVAC
473         update_hvac_inputs(h_u, c_x, cab_t)
474         _, h_x = hvac_mlsim.step(h_u)
475         update_dv1_inputs(b_u, h_x, c_x)
476         _, b_x = cabin_mlsim.step(b_u)
477         cabin[i] = b_x
478         hvac[i] = h_x
479         ctrl[i] = c_x
480         if log_inputs:
481             b_u_log[i] = b_u
482             h_u_log[i] = h_u
483             c_u_log[i] = c_u
484 
485     if log_inputs:
486         return cabin, hvac, ctrl, b_u_log, h_u_log, c_u_log
487     else:
488         return cabin, hvac, ctrl
```

Listing A.3: Connector between stepped objects.

```
1  """
2 
3  mlsim
4 
5 
6  Author
7  ------
8  J. Brusey
9 
10 Date
11 ----
12 4-March-2021
13 
14 
15 MLSim class that packages up a machine learnt model and scaler.
16 
17 """
18 
19 
```

```
20 import numpy as np
21
22 from .partial_scaler import PartialScaler
23
24
25 class MLSim:
26     """Discrete time simulator derived from machine-learnt model.
27
28     The idea of this class is to wrap up the ML simulator and deal
29     with scaling, lagged state and actions, and differing intervals to
30     make the simulator easier to call. The initial state and prior
31     actions are provided to the constructor and then the simulator can
32     be run by successive calls to ~step~.
33
34     """
35
36     def __init__(
37         self,
38         scaler_and_model,
39         initial_state,
40         xlag,
41         ulag,
42         xlen,
43         ulen,
44         interval=1,
45         initial_clock=0,
46         prior_actions=None,
47         ut_min=None,
48         ut_max=None,
49     ):
50         """construct a simulator object
51
52         Parameters
53         ----------
54         scaler_and_model : tuple
55
56           tuple containing scaler and model. The scaler transforms the
57           combined x, u vector, while model provides a single step
58           simulator for scaled values.
59
60         initial_state : array-like
61
62           vector containing initial state. Can also be a 2D array of
63           xlag vectors for more complete initialisation. If only a
64           single vector is supplied, that vector is replicated xlag
```

```
65            times to make up the initial state.
66
67        xlag : integer
68
69            number of lagged state values. This must match the
70            parameters used to develop the model
71
72        ulag : integer
73
74            number of lagged control values. This must match the
75            parameters used to develop the model
76
77        xlen : integer
78
79            length of the state vector
80
81        ulen : integer
82
83            length of the control vector
84
85        interval : integer
86
87            time interval that the simulator was developed using, in
88            seconds
89
90        initial_clock : integer
91
92            initial clock value so that first step will have time
93            initial_clock + interval
94
95        prior_actions : array
96
97            ulag - 1 actions
98
99        ut_min : array
100
101           action array minimum values (used for clipping)
102
103       ut_max : array
104
105           action array maximum values (used for clipping)
106
107       """
108       scaler, self.model = scaler_and_model
109       self.xt = initial_state
```

```
110         assert initial_state.shape == (xlag, xlen)
111         self.xlag = xlag
112         self.ulag = ulag
113         self.xlen = xlen
114         self.ulen = ulen
115         self.interval = interval
116         self.clock = initial_clock
117         self.prior_actions = prior_actions
118         self.first = True
119         self.x_scaler = PartialScaler(scaler, 0, xlen, xlen + ulen)
120         self.u_scaler = PartialScaler(scaler, xlen, xlen + ulen, xlen +
    ulen)
121         self.xt = self.x_scaler.transform(self.xt)
122         self.ut_min = ut_min
123         self.ut_max = ut_max
124         assert (ut_min is None) == (ut_max is None)
125
126     def step(self, ut):
127         """simulate a single time step
128
129         Given the current state history given by ~xt~ and current
130         control history ~ut~
131
132         Parameters
133         ----------
134
135         ut : array-like
136
137           control input for current time step
138
139         Returns
140         -------
141
142         (time, state vector at end of time step)
143
144         """
145         ut = np.array(ut, dtype=np.float32).reshape(1, -1)
146         if self.ut_min is not None and self.ut_max is not None:
147             ut = np.clip(ut, self.ut_min, self.ut_max)
148         ut = self.u_scaler.transform(ut)
149         assert self.xt.shape[0] == self.xlag
150         assert self.xt.shape[1] == self.xlen
151         if self.first:
152             # for first step, assume that the same control inputs were
153             # used for all previous steps
```

```
154             if self.prior_actions is not None:
155                 self.ut = np.vstack([self.u_scaler.transform(self.
    prior_actions), ut])
156             else:
157                 self.ut = np.vstack([ut] * (self.ulag))
158             self.first = False
159         else:
160             self.ut = np.append(self.ut[1:], ut, axis=0)
161         assert self.ut.shape == (self.ulag, self.ulen)
162 
163         x = np.hstack((self.xt.reshape(1, -1), self.ut.reshape(1, -1)))
164         new_xt = self.model.predict(x)
165         self.xt = np.append(self.xt[1:], new_xt.reshape(1, -1), axis=0)
166         self.clock += self.interval
167         return (self.clock, self.x_scaler.inverse_transform(new_xt)[0])
```

Listing A.4: MLSim class.

```
1  import numpy as np
2 
3  from domus_mlsim import (
4      DV0_XT_COLUMNS,
5      KELVIN,
6      DV0Ut,
7      DV0Xt,
8      HvacUt,
9      estimate_cabin_temperature_dv0,
10     hcm_reduced,
11     kw_to_array,
12     load_dv0,
13     make_dv0_sim,
14     update_control_inputs_dv0,
15     update_dv0_inputs,
16 )
17 
18 from . import DomusContEnv
19 from .acoustics import calc_sound_level
20 
21 
22 class DomusDv0ContEnv(DomusContEnv):
23     CABIN_ENERGY = np.zeros((len(DV0Ut)))
24 
25     def __init__(
26         self,
27         **kwargs,
```

```
    ):
        """Description:
            Simulation of the DV0 thermal environment of a Fiat 500e car
            cabin.

        This modifies DomusContEnv by overriding use of DV1 with DV0
    where needed.

        """
        super().__init__(**kwargs)
        self.b_u = np.zeros((len(DV0Ut)))
        self.dv1_scaler_and_model = None
        self.dv0_scaler_and_model = load_dv0()

    def _convert_state(self):
        """given the current state, create a vector that can be used as
    input to the controller"""
        cab_t = estimate_cabin_temperature_dv0(self.b_x)
        update_control_inputs_dv0(self.c_u, self.b_x, self.h_x, cab_t)
        return self.obs_tr.transform(self.c_u)

    def _body_state(self, b_x, n):
        """return the body state matrix for passenger n where 0 is the
    driver, etc"""
        if n == 0:
            v = b_x[
                [
                    DV0Xt.t_drvr1,
                    DV0Xt.m_drvr1,
                    DV0Xt.v_drvr1,
                    DV0Xt.t_drvr2,
                    DV0Xt.m_drvr2,
                    DV0Xt.v_drvr2,
                    DV0Xt.t_drvr3,
                    DV0Xt.m_drvr3,
                    DV0Xt.v_drvr3,
                ]
            ]
        elif n == 1:
            v = b_x[
                [
                    DV0Xt.t_psgr1,
                    DV0Xt.m_psgr1,
                    DV0Xt.v_psgr1,
                    DV0Xt.t_psgr2,
```

```
70                      DV0Xt.m_psgr2,
71                      DV0Xt.v_psgr2,
72                      DV0Xt.t_psgr3,
73                      DV0Xt.m_psgr3,
74                      DV0Xt.v_psgr3,
75                  ]
76              ]
77          elif n == 2:
78              v = b_x[
79                  [
80                      DV0Xt.t_psgr21,
81                      DV0Xt.m_psgr21,
82                      DV0Xt.v_psgr21,
83                      DV0Xt.t_psgr22,
84                      DV0Xt.m_psgr22,
85                      DV0Xt.v_psgr22,
86                      DV0Xt.t_psgr23,
87                      DV0Xt.m_psgr23,
88                      DV0Xt.v_psgr23,
89                  ]
90              ]
91          elif n == 3:
92              v = b_x[
93                  [
94                      DV0Xt.t_psgr31,
95                      DV0Xt.m_psgr31,
96                      DV0Xt.v_psgr31,
97                      DV0Xt.t_psgr32,
98                      DV0Xt.m_psgr32,
99                      DV0Xt.v_psgr32,
100                     DV0Xt.t_psgr33,
101                     DV0Xt.m_psgr33,
102                     DV0Xt.v_psgr33,
103                 ]
104             ]
105         # hcm uses celsius not kelvin
106         v = v - np.array([KELVIN, KELVIN, 0, KELVIN, KELVIN, 0, KELVIN,
    KELVIN, 0])
107         return v.reshape((3, 3))
108
109     def _comfort(self, b_x, h_u):
110         # temporarily just assess driver and front passenger comfort
111
112         # assess driver comfort
113         hcm = [
```

```
114             hcm_reduced(
115                 model=self.hcm_model,
116                 pre_clo=self.pre_clo,
117                 pre_out=h_u[HvacUt.ambient] - KELVIN,
118                 body_state=self._body_state(b_x, i),
119                 rh=b_x[DV0Xt.rhc] * 100,
120                 sound=calc_sound_level(h_u[HvacUt.speed], h_u[HvacUt.
    blw_power])[0],
121             )
122             for i in self.configured_passengers
123         ]
124         return np.mean(hcm)
125
126     def _ws_and_rh(self, b_x):
127         return b_x[DV0Xt.ws], b_x[DV0Xt.rhc]
128
129     def _step_cabin(self, c_x):
130         self.b_u[
131             [
132                 DV0Ut.t_a,
133                 DV0Ut.rh_a,
134                 DV0Ut.rad1,
135                 DV0Ut.rad2,
136                 DV0Ut.VehicleSpeed,
137             ]
138         ] = [
139             self.ambient_t,
140             self.ambient_rh,
141             self.solar1,
142             self.solar2,
143             self.car_speed / 100 * 27.778,
144         ]
145         update_dv0_inputs(self.b_u, self.h_x, c_x)
146         _, self.b_x = self.dv0_sim.step(self.b_u)
147
148     def _make_cabin_state(self):
149         return kw_to_array(
150             DV0_XT_COLUMNS,
151             t_drvr1=self.cabin_t,
152             t_drvr2=self.cabin_t,
153             t_drvr3=self.cabin_t,
154             t_psgr1=self.cabin_t,
155             t_psgr2=self.cabin_t,
156             t_psgr3=self.cabin_t,
157             t_psgr21=self.cabin_t,
```

```
158             t_psgr22=self.cabin_t,
159             t_psgr23=self.cabin_t,
160             t_psgr31=self.cabin_t,
161             t_psgr32=self.cabin_t,
162             t_psgr33=self.cabin_t,
163             v_drvr1=self.cabin_v,
164             v_drvr2=self.cabin_v,
165             v_drvr3=self.cabin_v,
166             v_psgr1=self.cabin_v,
167             v_psgr2=self.cabin_v,
168             v_psgr3=self.cabin_v,
169             v_psgr21=self.cabin_v,
170             v_psgr22=self.cabin_v,
171             v_psgr23=self.cabin_v,
172             v_psgr31=self.cabin_v,
173             v_psgr32=self.cabin_v,
174             v_psgr33=self.cabin_v,
175             m_drvr1=self.cabin_t,
176             m_drvr2=self.cabin_t,
177             m_drvr3=self.cabin_t,
178             m_psgr1=self.cabin_t,
179             m_psgr2=self.cabin_t,
180             m_psgr3=self.cabin_t,
181             m_psgr21=self.cabin_t,
182             m_psgr22=self.cabin_t,
183             m_psgr23=self.cabin_t,
184             m_psgr31=self.cabin_t,
185             m_psgr32=self.cabin_t,
186             m_psgr33=self.cabin_t,
187             rhc=self.cabin_rh,
188             ws=self.cabin_t,
189         )
190
191     def _make_cabin_sim(self):
192         self.dv0_sim = make_dv0_sim(self.dv0_scaler_and_model, self.b_x)
```

Listing A.5: Reinforcement Learning Gym environment.

References

[Fia98] Dusan Fiala. *Dynamic Simulation of Human Heat Transfer and Thermal Comfort*. ResearchGate. 1998. URL: https://www.researchgate.net/publication/35402573_Dynamic_Simulation_of_Human_Heat_Transfer_and_Thermal_Comfort (visited on 01/21/2020).

[Zha03] H. Zhang. "Human Thermal Sensation and Comfort in Transient and Non-Uniform Thermal Environments". Sept. 1, 2003. URL: https://escholarship.org/uc/item/11m0n1wt (visited on 01/21/2020).

[Nil04] Håkan Nilsson. "Comfort Climate Evaluation with Thermal Manikin Methods and Computer Simulation Models". Apr. 16, 2004.

[Wei+04] E. P. Weijers et al. "Variability of Particulate Matter Concentrations along Roads and Motorways Determined by a Moving Measurement Unit". In: *Atmospheric Environment* 38.19 (June 1, 2004), pp. 2993–3002. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2004.02.045. URL: https://www.sciencedirect.com/science/article/pii/S1352231004001876 (visited on 02/16/2022).

[ISO07] ISO. *ISO/TS 14505-1:2007*. International Organisation for standardization, 2007.

[WL07] Robert L. Williams and Douglas A. Lawrence. *Linear State-Space Control Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Feb. 14, 2007. ISBN: 978-0-470-11787-3 978-0-471-73555-7. DOI: 10.1002/9780470117873. URL: http://doi.wiley.com/10.1002/9780470117873 (visited on 01/28/2023).

[ISO08] ISO. *ISO 14505-2:2006*. Under Review. 2008. ISBN: 978-0-580-60114-9.

[Tit+08] A. Tittarelli et al. "Estimation of Particle Mass Concentration in Ambient Air Using a Particle Counter". In: *Atmospheric Environment* 42.36 (Nov. 1, 2008), pp. 8543–8548. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2008.07.056. URL: https://www.sciencedirect.com/science/article/pii/S135223100800695X (visited on 12/03/2021).

[HRO09] Elisabeth Heseltine, Jerome Rosen, and World Health Organization, eds. *WHO Guidelines for Indoor Air Quality: Dampness and Mould*. Copenhagen: WHO, 2009. 228 pp. ISBN: 978-92-890-4168-3.

[Org10] World Health Organization, ed. *Who Guidelines for Indoor Air Quality: Selected Pollutants*. Copenhagen: WHO, 2010. 454 pp. ISBN: 978-92-890-0213-4.

[vEW10] Nees Jan van Eck and Ludo Waltman. "Software Survey: VOSviewer, a Computer Program for Bibliometric Mapping". In: *Scientometrics* 84.2 (Aug. 2010), pp. 523–538. ISSN: 0138-9130, 1588-2861. DOI: 10.1007/s11192-009-0146-3. URL: http://link.springer.com/10.1007/s11192-009-0146-3 (visited on 06/22/2020).

[Zha+10] Hui Zhang et al. "Thermal Sensation and Comfort Models for Non-Uniform and Transient Environments, Part III: Whole-body Sensation and Comfort". In: *Building and Environment*. 1st International Symposium on Sustainable Healthy Buildings 45.2 (Feb. 1, 2010), pp. 399–410. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2009.06.020. URL: http://www.sciencedirect.com/science/article/pii/S0360132309001619 (visited on 12/19/2019).

[MS11] LENA MÖLLER and LAUST SÖRENSEN. "Thermodynamical Modeling of a Car Cabin". Gothenburg, Sweden, 2011: CHALMERS UNIVERSITY OF TECHNOLOGY, 2011. URL: https://odr.chalmers.se/bitstream/20.500.12380/156753/1/156753.pdf.

[Ped+11] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928. URL: http://jmlr.org/papers/v12/pedregosa11a.html (visited on 06/02/2022).

[ISO12] ISO. *BS ISO 12219-1:2012. Interior Air of Road Vehicles — Part 1: Whole Vehicle Test Chamber —Specification and Method for the Determination of Volatile Organic Compounds in Cabin Interiors*. BSI Standards Publication, 2012.

[YYT12] Tomomi Yasuda, Seiichiro Yonemura, and Akira Tani. "Comparison of the Characteristics of Small Commercial NDIR CO2 Sensor Models and Development of a Portable CO2 Measurement Device". In: *Sensors* 12.3 (3 Mar. 2012), pp. 3641–3655. DOI: 10.3390/s120303641. URL: https://www.mdpi.com/1424-8220/12/3/3641 (visited on 03/16/2021).

[BSP13] Franco Belosi, Gianni Santachiara, and Franco Prodi. "Performance Evaluation of Four Commercial Optical Particle Counters". In: *Atmospheric and Climate Sciences* 3.1 (1 Jan. 30, 2013), pp. 41–46. DOI: 10.4236/acs.2013.31006. URL: http://www.scirp.org/Journal/Paperabs.aspx?paperid=27561 (visited on 12/03/2021).

[Gra+13] Michael L. Grady et al. "Vehicle Cabin Air Quality with Fractional Air Recirculation". In: SAE 2013 World Congress & Exhibition. Apr. 8, 2013, pp. 2013-01–1494. DOI: 10.4271/2013-01-1494. URL: https://www.sae.org/content/2013-01-1494/ (visited on 10/20/2021).

[KJ13] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. New York, NY: Springer, 2013. ISBN: 978-1-4614-6848-6 978-1-4614-6849-3. DOI: 10.1007/978-1-4614-6849-3. URL: http://link.springer.com/10.1007/978-1-4614-6849-3 (visited on 05/05/2022).

[CW14] Robert D. Christ and Robert L. Wernli. "Sensor Theory". In: *The ROV Manual*. Elsevier, 2014, pp. 297–326. ISBN: 978-0-08-098288-5. DOI: 10.1016/B978-0-08-098288-5.00012-9. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780080982885000129 (visited on 03/05/2021).

[Hin+14] Diana Hintea et al. "Applicability of Thermal Comfort Models to Car Cabin Environments". In: *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. 2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO). Vol. 01. Sept. 2014, pp. 769–776. DOI: 10.5220/0005101707690776.

[Lor+14] Manuel Lorenz et al. "A Coupled Numerical Model to Predict Heat Transfer and Passenger Thermal Comfort in Vehicle Cabins". In: SAE 2014 World Congress & Exhibition. Apr. 1, 2014, pp. 2014-01–0664. DOI: 10.4271/2014-01-0664. URL: https://www.sae.org/content/2014-01-0664/ (visited on 12/16/2019).

[Org14] World Health Organization, ed. *WHO Guidelines for Indoor Air Quality: Household Fuel Combustion*. Geneva, Switzerland: World Health Organization, 2014. 160 pp. ISBN: 978-92-4-154887-8.

[Cro+15] Cristiana Croitoru et al. "Thermal Comfort Models for Indoor Spaces and Vehicles—Current Capabilities and Future Perspectives". In: *Renewable and Sustainable Energy Reviews* 44 (Apr. 1, 2015), pp. 304–318. ISSN: 1364-0321. DOI: 10.1016/j.rser.2014.10.105. URL: http://www.sciencedirect.com/science/article/pii/S1364032114009332 (visited on 12/16/2019).

[DMS15] Aisling Doyle, Tariq Muneer, and Ian Smith. "A Review of the Thermal Performance of Electric Vehicles". In: *2015 IEEE International Transportation Electrification Conference (ITEC)*. 2015 IEEE International Transportation Electrification Conference (ITEC). Aug. 2015, pp. 1–5. DOI: 10.1109/ITEC-India.2015.7386922.

[GRP15] Aled Gravelle, Simon Robinson, and Alessandro Picarelli. "Modeling the Effects of Energy Efficient Glazing on Cabin Thermal Energy and Vehicle Efficiency". In: The 11th International Modelica Conference. Sept. 18, 2015, pp. 291–300. DOI: 10.3384/ecp15118291. URL: https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=118&Article_No=31 (visited on 05/24/2022).

[Hep+15] Mark Hepokoski et al. "Simulating Physiological Response with a Passive Sensor Manikin and an Adaptive Thermal Manikin to Predict Thermal Sensation and Comfort". In: SAE 2015 World Congress & Exhibition. Apr. 14, 2015, pp. 2015-01–0329. DOI: 10.4271/2015-01-0329. URL: https://www.sae.org/content/2015-01-0329/ (visited on 12/19/2019).

[Kak15] Rupesh Sonu Kakade. "Least-Enthalpy Based Control of Cabin Air Recirculation". In: SAE 2015 World Congress & Exhibition. Apr. 14, 2015, pp. 2015-01–0372. DOI: 10.4271/2015-01-0372. URL: https://www.sae.org/content/2015-01-0372/ (visited on 12/19/2019).

[Bro+16] Greg Brockman et al. *OpenAI Gym*. June 5, 2016. arXiv: 1606.01540 [cs]. URL: http://arxiv.org/abs/1606.01540 (visited on 06/24/2022).

[DVD16] Paul Danca, Andreea Vartires, and Angel Dogeanu. "An Overview of Current Methods for Thermal Comfort Assessment in Vehicle Cabin". In: *Energy Procedia*. EENVIRO-YRC 2015 - Bucharest 85 (Jan. 1, 2016), pp. 162–169. ISSN: 1876-6102. DOI: 10.1016/j.egypro.2015.12.322. URL: http://www.sciencedirect.com/science/article/pii/S1876610215029872 (visited on 12/17/2019).

[KLZ16] Katarina Katić, Rongling Li, and Wim Zeiler. "Thermophysiological Models and Their Applications: A Review". In: *Building and Environment* 106 (Sept. 1, 2016), pp. 286–300. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2016.06.031. URL: http://www.sciencedirect.com/science/article/pii/S0360132316302384 (visited on 03/03/2020).

[Sch16] John Schulman. "Optimizing Expectations: From Deep Reinforcement Learning to Stochastic Computation Graphs". UC Berkeley, 2016. URL: https://escholarship.org/uc/item/9z908523 (visited on 07/01/2022).

[Soc+16] Lavinia Socaciu et al. "PCM Selection Using AHP Method to Maintain Thermal Comfort of the Vehicle Occupants". In: *Energy Procedia*. EENVIRO-YRC 2015 - Bucharest 85 (Jan. 1, 2016), pp. 489–497. ISSN: 1876-6102. DOI: 10.1016/j.egypro.2015.12.232. URL: http://www.sciencedirect.com/science/article/pii/S1876610215028970 (visited on 12/17/2019).

[Tit+16] Gene Titov et al. "MATLAB/Simulink Framework for Modeling Complex Coolant Flow Configurations of Advanced Automotive Thermal Management Systems". In: SAE 2016 World Congress and Exhibition. Apr. 5, 2016, pp. 2016-01–0230. DOI: 10.4271/2016-01-0230. URL: https://www.sae.org/content/2016-01-0230/ (visited on 07/14/2022).

[ANS17] ANSI/ASHRAE. *ANSI/ASHRAE Standard 55-2017*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2017.

[Cas+17] Nuria Castell et al. "Can Commercial Low-Cost Sensor Platforms Contribute to Air Quality Monitoring and Exposure Estimates?" In: *Environment International* 99 (Feb. 1, 2017), pp. 293–302. ISSN: 0160-4120. DOI: 10.1016/j.envint.2016.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0160412016309989 (visited on 11/16/2021).

[Din+17] A. Dinoi et al. "Comparison of Atmospheric Particle Concentration Measurements Using Different Optical Detectors: Potentiality and Limits for Air Quality Applications". In: *Measurement* 106 (Aug. 1, 2017), pp. 274–282. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2016.02.019. URL: https://www.sciencedirect.com/science/article/pii/S0263224116000968 (visited on 12/03/2021).

[Foj+17] Miloš Fojtlín et al. "An Innovative HVAC Control System: Implementation and Testing in a Vehicular Cabin". In: *Journal of Thermal Biology*. Assessing and Modeling of the Thermal Impact of Human-Clothing Environment 70 (Dec. 1, 2017), pp. 64–68. ISSN: 0306-4565. DOI: 10.1016/j.jtherbio.2017.04.002. URL: http://www.sciencedirect.com/science/article/pii/S0306456517300013 (visited on 12/18/2019).

[KKJ17] Neelakandan Kandasamy, Koundinya Narasimha Kota, and Prasad Joshi. "Numerical Evaluation of Vehicle Orientation and Glazing Material Impact on Cabin Climate and Occupant Thermal Comfort". In: Symposium on

International Automotive Technology 2017. Jan. 10, 2017, pp. 2017-26–0262. DOI: 10.4271/2017-26-0262. URL: https://www.sae.org/content/2017-26-0262/ (visited on 12/19/2019).

[Kel+17] K. E. Kelly et al. "Ambient and Laboratory Evaluation of a Low-Cost Particulate Matter Sensor". In: *Environmental Pollution* 221 (Feb. 1, 2017), pp. 491–500. ISSN: 0269-7491. DOI: 10.1016/j.envpol.2016.12.039. URL: https://www.sciencedirect.com/science/article/pii/S026974911632718X (visited on 11/16/2021).

[MLA17] M. Gerboles, L. Spinelle, and A. Borowiak. *Measuring Air Pollution with Low-Cost Sensors*. EU Science Hub - European Commission. Nov. 28, 2017. URL: https://ec.europa.eu/jrc/en/publication/brochures-leaflets/measuring-air-pollution-low-cost-sensors (visited on 03/05/2021).

[Ma+17] Yan Ma et al. "Time Granularity Transformation of Time Series Data for Failure Prediction of Overhead Line". In: *Journal of Physics: Conference Series* 787 (Jan. 2017), p. 012031. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/787/1/012031. URL: https://iopscience.iop.org/article/10.1088/1742-6596/787/1/012031 (visited on 02/11/2022).

[Nea+17] Catalin Neacsu et al. "The Evaluation of the Overall Thermal Comfort inside a Vehicle". In: *IOP Conference Series: Materials Science and Engineering* 252 (Oct. 2017), p. 012031. ISSN: 1757-899X. DOI: 10.1088/1757-899X/252/1/012031. URL: https://doi.org/10.1088%2F1757-899x%2F252%2F1%2F012031 (visited on 12/17/2019).

[Psi+17] Agnes Psikuta et al. "Thermal Manikins Controlled by Human Thermoregulation Models for Energy Efficiency and Thermal Comfort Research – A Review". In: *Renewable and Sustainable Energy Reviews* 78 (Oct. 1, 2017), pp. 1315–1330. ISSN: 1364-0321. DOI: 10.1016/j.rser.2017.04.115. URL: http://www.sciencedirect.com/science/article/pii/S1364032117306202 (visited on 12/17/2019).

[Rai+17] Aakash C. Rai et al. "End-User Perspective of Low-Cost Sensors for Outdoor Air Pollution Monitoring". In: *Science of The Total Environment* 607–608 (Dec. 31, 2017), pp. 691–705. ISSN: 0048-9697. DOI: 10.1016/j.scitotenv.2017.06.266. URL: https://www.sciencedirect.com/science/article/pii/S0048969717316935 (visited on 11/16/2021).

[Sch+17] John Schulman et al. *Proximal Policy Optimization Algorithms*. Aug. 28, 2017. DOI: 10.48550/arXiv.1707.06347. arXiv: 1707.06347 [cs]. URL: http://arxiv.org/abs/1707.06347 (visited on 01/26/2023).

[ZHF17] Nadezda Zikova, Philip K. Hopke, and Andrea R. Ferro. "Evaluation of New Low-Cost Particle Monitors for PM2.5 Concentrations Measurements". In: *Journal of Aerosol Science* 105 (Mar. 1, 2017), pp. 24–34. ISSN: 0021-8502. DOI: 10.1016/j.jaerosci.2016.11.010. URL: https://www.sciencedirect.com/science/article/pii/S0021850216301835 (visited on 11/16/2021).

[Bru+18] James Brusey et al. "Reinforcement Learning-Based Thermal Comfort Control for Vehicle Cabins". In: *Mechatronics* 50 (Apr. 1, 2018), pp. 413–421. ISSN: 0957-4158. DOI: 10.1016/j.mechatronics.2017.04.010. URL: http://www.sciencedirect.com/science/article/pii/S0957415817300570 (visited on 12/18/2019).

[Cho+18] Sourav Chowdhury et al. "Total Thermal Management of Battery Electric Vehicles (BEVs)". In: CO2 Reduction for Transportation Systems Conference. May 30, 2018, pp. 2018-37–0026. DOI: 10.4271/2018-37-0026. URL: https://www.sae.org/content/2018-37-0026/ (visited on 01/31/2022).

[Hep+18] Mark Hepokoski et al. "Evaluating a Vehicle Climate Control System with a Passive Sensor Manikin Coupled with a Thermal Comfort Model". In: WCX World Congress Experience. Apr. 3, 2018, pp. 2018-01–0065. DOI: 10.4271/2018-01-0065. URL: https://www.sae.org/content/2018-01-0065/ (visited on 12/19/2019).

[ISM18] Yusuke Ito, Tomonori Sakoi, and Takeshi Miyamoto. "Evaluation Method of Thermal Sensation and Comfort for Air Conditioning Performance Reduction". In: WCX World Congress Experience. Apr. 3, 2018, pp. 2018-01–0775. DOI: 10.4271/2018-01-0775. URL: https://www.sae.org/content/2018-01-0775/ (visited on 12/19/2019).

[KMW18] Majid Karami, Gabrielle Viola McMorrow, and Liping Wang. "Continuous Monitoring of Indoor Environmental Quality Using an Arduino-based Data Acquisition System". In: *Journal of Building Engineering* 19 (Sept. 1, 2018), pp. 412–419. ISSN: 2352-7102. DOI: 10.1016/j.jobe.2018.05.014. URL: https://www.sciencedirect.com/science/article/pii/S2352710218301025 (visited on 03/18/2021).

[Kub+18] Takuya Kubota et al. "Automotive Thermal Environment Model to Design Climate Control Logics Based on Thermal Sensation". In: WCX World Congress Experience. Apr. 3, 2018, pp. 2018-01–0064. DOI: 10.4271/2018-01-0064. URL: https://www.sae.org/content/2018-01-0064/ (visited on 12/19/2019).

[Lah+18a] A. A. Lahimer et al. "Potential of Solar Reflective Cover on Regulating the Car Cabin Conditions and Fuel Consumption". In: *Applied Thermal Engineering* 143 (Oct. 1, 2018), pp. 59–71. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2018.07.020. URL: http://www.sciencedirect.com/science/article/pii/S1359431117331988 (visited on 12/18/2019).

[Lah+18b] A. Lahlou et al. "A Dynamic Programming Approach for Thermal Comfort Control in Electric Vehicles". In: *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*. 2018 IEEE Vehicle Power and Propulsion Conference (VPPC). Aug. 2018, pp. 1–6. DOI: 10.1109/VPPC.2018.8604983.

[Lan+18] Pascal Lange et al. "Low-Cost Thermal Manikin – A Competitive Instrument to Simulate Thermal Loads and to Determine Thermal Passenger Comfort". Aug. 28, 2018. DOI: 10.5281/zenodo.1404483. URL: https://zenodo.org/record/1404483#.XvYf189fhhE (visited on 06/26/2020).

[Li+18a] Chengguo Li et al. "Reducing Mobile Air Conditioner (MAC) Power Consumption Using Active Cabin-Air-Recirculation in A Plug-In Hybrid Electric Vehicle (PHEV)". In: *World Electric Vehicle Journal* 9.4 (4 Dec. 2018), p. 51. DOI: 10.3390/wevj9040051. URL: https://www.mdpi.com/2032-6653/9/4/51 (visited on 03/18/2021).

[Li+18b] Jiayu Li et al. "Spatiotemporal Distribution of Indoor Particulate Matter Concentration with a Low-Cost Sensor Network". In: *Building and Environment* 127 (Jan. 1, 2018), pp. 138–147. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2017.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0360132317304997 (visited on 03/10/2021).

[Man+18] Charbel Mansour et al. "Assessing Additional Fuel Consumption from Cabin Thermal Comfort and Auxiliary Needs on the Worldwide Harmonized Light Vehicles Test Cycle". In: *Transportation Research Part D: Transport and Environment* 62 (July 1, 2018), pp. 139–151. ISSN: 1361-9209. DOI: 10.1016/j.trd.2018.02.012. URL: http://www.sciencedirect.com/science/article/pii/S1361920917307204 (visited on 12/18/2019).

[MAT18] MATLAB. *9.7.0.1190202 (R2019b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[Mor+18a] Lidia Morawska et al. "Applications of Low-Cost Sensing Technologies for Air Quality Monitoring and Exposure Assessment: How Far Have They Gone?" In: *Environment International* 116 (July 1, 2018), pp. 286–299. ISSN: 0160-4120. DOI: 10.1016/j.envint.2018.04.018. URL: https://www.sciencedirect.com/science/article/pii/S0160412018302460 (visited on 11/16/2021).

[Mor+18b] Masahiro Morishita et al. "Evaluation of Thermal Environment in Vehicles for Occupant Comfort Using Equivalent Temperature of Thermal Manikin during Start-Stop Function with Energy Storage Evaporators". In: WCX World Congress Experience. Apr. 3, 2018, pp. 2018-01–0059. DOI: 10.4271/2018-01-0059. URL: https://www.sae.org/content/2018-01-0059/ (visited on 12/19/2019).

[Pau+18] Danca Paul Alexandru et al. "CFD Simulation of a Cabin Thermal Environment with and without Human Body – Thermal Comfort Evaluation". In: *E3S Web of Conferences* 32 (Jan. 1, 2018), p. 01018. DOI: 10.1051/e3sconf/20183201018.

[RHB18] Daniel Rüffer, Felix Hoehne, and Johannes Bühler. "New Digital Metal-Oxide (MOx) Sensor Platform". In: *Sensors (Basel, Switzerland)* 18.4 (Mar. 31, 2018). ISSN: 1424-8220. DOI: 10.3390/s18041052. pmid: 29614746. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5948493/ (visited on 03/08/2021).

[Thi+18] Ilango Thiagalingam et al. "Confrontation Of Thermal Sensation And Comfort Models To Votes In A Transient Thermal Exposure". In: (Aug. 28, 2018). DOI: 10.5281/ZENODO.1404546. URL: https://zenodo.org/record/1404546 (visited on 12/19/2019).

[YFL18] Xingda Yan, James Fleming, and Roberto Lot. "A/C Energy Management and Vehicle Cabin Thermal Comfort Control". In: *IEEE Transactions on Vehicular Technology* 67.11 (Nov. 2018), pp. 11238–11242. ISSN: 1939-9359. DOI: 10.1109/TVT.2018.2869030.

[Ada+19] Ewa Adamiec et al. "Using Medium-Cost Sensors to Estimate Air Quality in Remote Locations. Case Study of Niedzica, Southern Poland". In: *Atmosphere* 10.7 (7 July 2019), p. 393. DOI: 10.3390/atmos10070393. URL: https://www.mdpi.com/2073-4433/10/7/393 (visited on 11/09/2021).

[Alg+19] Joshua A. Alger et al. "Automatic Vehicle Climate Control Based on Predicted Air Quality". U.S. pat. 10226982B2. International Business Machines Corp. Mar. 12, 2019. URL: https://patents.google.com/patent/US10226982B2/en (visited on 03/18/2021).

[BK19] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge: Cambridge University Press, 2019. ISBN: 978-1-108-42209-3. DOI: 10.1017/9781108380690. URL: https://www.cambridge.org/core/books/datadriven-science-and-engineering/77D52B171B60A496EAFE4DB662ADC36E (visited on 05/24/2022).

[CZ19] Sourav Chowdhury and Mark Zima. *Unitary Thermal Energy Management for Propulsion Range Augmentation (UTEMPRA)*. DOE-MAHLE-0006840, 1569646. Sept. 25, 2019, DOE-MAHLE–0006840, 1569646. DOI: 10.2172/1569646. URL: http://www.osti.gov/servlets/purl/1569646/ (visited on 02/02/2022).

[Com+19] Joint Research Centre (European Commission) et al. *Review of Sensors for Air Quality Monitoring*. LU: Publications Office of the European Union, 2019. ISBN: 978-92-76-09255-1. URL: https://data.europa.eu/doi/10.2760/568261 (visited on 10/21/2021).

[DM19] Aisling Doyle and Tariq Muneer. "Energy Consumption and Modelling of the Climate Control System in the Electric Vehicle". In: *Energy Exploration & Exploitation* 37.1 (Jan. 1, 2019), pp. 519–543. ISSN: 0144-5987. DOI: 10.1177/0144598718806458. URL: https://doi.org/10.1177/0144598718806458 (visited on 03/05/2021).

[Eng+19] Peter Engel et al. "Modeling of Automotive HVAC Systems Using Long Short-Term Memory Networks". In: May 5, 2019.

[Fra+19] Remy Franken et al. "Comparison of Methods for Converting Dylos Particle Number Concentrations to PM2.5 Mass Concentrations". In: *Indoor Air* 29.3 (2019), pp. 450–459. ISSN: 1600-0668. DOI: 10.1111/ina.12546. URL: http://onlinelibrary.wiley.com/doi/abs/10.1111/ina.12546 (visited on 12/03/2021).

[Han+19] Mengjie Han et al. "A Review of Reinforcement Learning Methodologies for Controlling Occupant Comfort in Buildings". In: *Sustainable Cities and Society* 51 (Nov. 1, 2019), p. 101748. ISSN: 2210-6707. DOI: 10.1016/j.scs.

2019.101748. URL: https://www.sciencedirect.com/science/article/pii/S2210670719307589 (visited on 01/14/2023).

[Heo+19] Ki Joon Heo et al. "Comparison of Filtration Performance of Commercially Available Automotive Cabin Air Filters against Various Airborne Pollutants". In: *Building and Environment* 161 (Aug. 15, 2019), p. 106272. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2019.106272. URL: https://www.sciencedirect.com/science/article/pii/S0360132319304822 (visited on 03/11/2021).

[IT19] Paolo Iora and Laura Tribioli. "Effect of Ambient Temperature on Electric Vehicles' Energy Consumption and Range: Model Definition and Sensitivity Analysis Based on Nissan Leaf Data". In: *World Electric Vehicle Journal* 10.1 (1 Mar. 2019), p. 2. ISSN: 2032-6653. DOI: 10.3390/wevj10010002. URL: https://www.mdpi.com/2032-6653/10/1/2 (visited on 03/16/2022).

[ISO19] ISO. *BS ISO 16000-37:2019. Indoor Air - Part 37: Measurement of PM_{2,5} Mass Concentration*. BSI Standards Publication, 2019.

[JJ19] Wooyoung Jung and Farrokh Jazizadeh. "Human-in-the-Loop HVAC Operations: A Quantitative Review on Occupancy, Comfort, and Energy-Efficiency Dimensions". In: *Applied Energy* 239 (Apr. 1, 2019), pp. 1471–1508. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2019.01.070. URL: http://www.sciencedirect.com/science/article/pii/S030626191930073X (visited on 12/17/2019).

[KMP19] Kinnera Bharath Kumar Sai, Subhaditya Mukherjee, and H Parveen Sultana. "Low Cost IoT Based Air Quality Monitoring Setup Using Arduino and MQ Series Sensors With Dataset Analysis". In: *Procedia Computer Science*. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019 165 (Jan. 1, 2019), pp. 322–327. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.01.043. URL: https://www.sciencedirect.com/science/article/pii/S187705092030051X (visited on 03/18/2021).

[Mar+19] Garrett J. Marshall et al. "Thermal Management of Vehicle Cabins, External Surfaces, and Onboard Electronics: An Overview". In: *Engineering* 5.5 (Oct. 1, 2019), pp. 954–969. ISSN: 2095-8099. DOI: 10.1016/j.eng.2019.02.009. URL: http://www.sciencedirect.com/science/article/pii/S2095809918312529 (visited on 12/18/2019).

[NEW19] Austin NEWMAN. "Automatic Vehicle Cabin Air Filtration System". U.S. pat. 10245924B2. NIO USA Inc. Apr. 2, 2019. URL: https://patents.google.com/patent/US10245924B2/en (visited on 03/18/2021).

[Och+19] Toshinori Ochiai et al. "Thin Ceiling Circulator to Enhance Thermal Comfort and Cabin Space". In: WCX SAE World Congress Experience. Apr. 2, 2019, pp. 2019-01–0913. DOI: 10.4271/2019-01-0913. URL: https://www.sae.org/content/2019-01-0913/ (visited on 12/19/2019).

[Oi+19] Hajime Oi et al. "Evaluation of Equivalent Temperature in a Vehicle Cabin with a Numerical Thermal Manikin (Part 1): Measurement of Equivalent Temperature in a Vehicle Cabin and Development of a Numerical Thermal Manikin". In: WCX SAE World Congress Experience. Apr. 2, 2019, pp. 2019-01–0697. DOI: 10.4271/2019-01-0697. URL: https://www.sae.org/content/2019-01-0697/ (visited on 12/19/2019).

[Oze+19] Yoshiichi Ozeki et al. "Evaluation of Equivalent Temperature in a Vehicle Cabin with a Numerical Thermal Manikin (Part 2): Evaluation of Thermal Environment and Equivalent Temperature in a Vehicle Cabin". In: WCX SAE World Congress Experience. Apr. 2, 2019, pp. 2019-01–0698. DOI: 10.4271/2019-01-0698. URL: https://www.sae.org/content/2019-01-0698/ (visited on 12/19/2019).

[Pau+19] Danca Paul Alexandru et al. "Evaluation of the Thermal Comfort for Its Occupants inside a Vehicle during Summer". In: *IOP Conference Series: Materials Science and Engineering* 595 (Sept. 20, 2019), p. 012027. DOI: 10.1088/1757-899X/595/1/012027.

[PBW19] Alexandra Galina Petre, James Brusey, and Ross Wilkins. "An Examination of Comfort and Sensation for Manual and Automatic Controls of the Vehicle HVAC System". In: Automotive Technical Papers. Jan. 15, 2019, pp. 2019-01–5005. DOI: 10.4271/2019-01-5005. URL: https://www.sae.org/content/2019-01-5005/ (visited on 12/19/2019).

[PU19] Alexandra-Galina Petre and Coventry University. "User Feedback-based Reinforcement Learning for Vehicle Comfort Control". 2019.

[Pha+19] Liem Pham et al. "Development of a Standard Testing Method for Vehicle Cabin Air Quality Index". In: *SAE International Journal of Commercial Vehicles* 12 (May 20, 2019). DOI: 10.4271/02-12-02-0012.

[Rao19] Dattaraj Rao. "Leveraging Human Domain Knowledge to Model an Empirical Reward Function for a Reinforcement Learning Problem". Sept. 16, 2019. DOI: 10.48550/arXiv.1909.07116. arXiv: 1909.07116 [cs, eess]. URL: http://arxiv.org/abs/1909.07116 (visited on 05/16/2022).

[SS19] Stefan Schaut and Oliver Sawodny. "Thermal Management for the Cabin of a Battery Electric Vehicle Considering Passengers' Comfort". In: *IEEE Transactions on Control Systems Technology* (2019), pp. 1–17. ISSN: 2374-0159. DOI: 10.1109/TCST.2019.2914888.

[Sny19] Hannah Snyder. "Literature Review as a Research Methodology: An Overview and Guidelines". In: *Journal of Business Research* 104 (Nov. 1, 2019), pp. 333–339. ISSN: 0148-2963. DOI: 10.1016/j.jbusres.2019.07.039. URL: https://www.sciencedirect.com/science/article/pii/S0148296319304564 (visited on 01/11/2023).

[Val+19] William Valladares et al. "Energy Optimization Associated with Thermal Comfort and Indoor Air Control via a Deep Reinforcement Learning Algorithm". In: *Building and Environment* 155 (May 15, 2019), pp. 105–117. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2019.03.038. URL: https://www.sciencedirect.com/science/article/pii/S0360132319302008 (visited on 07/14/2022).

[Xie+19] Yi Xie et al. "A Self-learning Intelligent Passenger Vehicle Comfort Cooling System Control Strategy". In: *Applied Thermal Engineering* (Nov. 7, 2019), p. 114646. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2019.114646. URL: http://www.sciencedirect.com/science/article/pii/S1359431119358892 (visited on 12/17/2019).

[Zha+19] Cheng Zhang et al. "Multiphysics Modeling of Energy Intensity and Energy Efficiency of Electric Vehicle Operation". In: *Procedia CIRP*. 26th CIRP Conference on Life Cycle Engineering (LCE) Purdue University, West Lafayette, IN, USA May 7-9, 2019 80 (Jan. 1, 2019), pp. 322–327. ISSN: 2212-8271. DOI: 10.1016/j.procir.2019.01.058. URL: https://www.sciencedirect.com/science/article/pii/S2212827119300605 (visited on 06/18/2021).

[ZLC19] Xiaojie Zhou, Dayi Lai, and Qingyan Chen. "Experimental Investigation of Thermal Comfort in a Passenger Car under Driving Conditions". In: *Building and Environment* 149 (Feb. 1, 2019), pp. 109–119. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.12.022. URL: http://www.sciencedirect.com/science/article/pii/S0360132318307583 (visited on 12/17/2019).

[Alf+20] Brigida Alfano et al. "A Review of Low-Cost Particulate Matter Sensors from the Developers' Perspectives". In: *Sensors* 20.23 (23 Jan. 2020), p. 6819. DOI: 10.3390/s20236819. URL: https://www.mdpi.com/1424-8220/20/23/6819 (visited on 03/30/2021).

[Bra+20] Silvio Brandi et al. "Deep Reinforcement Learning to Optimise Indoor Temperature Control and Heating Energy Consumption in Buildings". In: *Energy and Buildings* 224 (Oct. 1, 2020), p. 110225. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2020.110225. URL: https://www.sciencedirect.com/science/article/pii/S0378778820308963 (visited on 01/14/2023).

[Col+20] Martine Collaud Coen et al. "Effects of the Prewhitening Method, the Time Granularity, and the Time Segmentation on the Mann–Kendall Trend Detection and the Associated Sen's Slope". In: *Atmospheric Measurement Techniques* 13.12 (Dec. 21, 2020), pp. 6945–6964. ISSN: 1867-1381. DOI: 10.5194/amt-13-6945-2020. URL: https://amt.copernicus.org/articles/13/6945/2020/ (visited on 02/11/2022).

[De +20] Saverio De Vito et al. "On the Robustness of Field Calibration for Smart Air Quality Monitors". In: *Sensors and Actuators B: Chemical* 310 (May 1, 2020), p. 127869. ISSN: 0925-4005. DOI: 10.1016/j.snb.2020.127869. URL: https://www.sciencedirect.com/science/article/pii/S0925400520302161 (visited on 02/14/2022).

[Foj+20] Miloš Fojtlín et al. "Thermal Model of an Unconditioned, Heated and Ventilated Seat to Predict Human Thermo-Physiological Response and Local Thermal Sensation". In: *Building and Environment* 169 (Feb. 1, 2020), p. 106571. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2019.106571. URL: http://www.sciencedirect.com/science/article/pii/S0360132319307838 (visited on 12/19/2019).

[Gut20] William F. Guthrie. *NIST/SEMATECH e-Handbook of Statistical Methods (NIST Handbook 151)*. National Institute of Standards and Technology, 2020. DOI: 10.18434/M32189. URL: https://www.itl.nist.gov/div898/handbook/ (visited on 03/18/2021).

[ISO20] ISO. *ISO/DIS 14505-4:2020*. International Organisation for standardization, 2020.

[Kor+20] Kyriakos Koritsoglou et al. "Improving the Accuracy of Low-Cost Sensor Measurements for Freezer Automation". In: *Sensors* 20.21 (21 Jan. 2020),

p. 6389. DOI: 10.3390/s20216389. URL: https://www.mdpi.com/1424-8220/20/21/6389 (visited on 03/09/2021).

[Kor20] Sepideh Korsavi. "Indoor Environment Quality and Adaptive Behaviours in Primary Schools in the UK". 2020.

[Kum20] Akshyt Bimal Kumar. "Battery Thermal Management for an Urban Electric Freight Vehicle Using Reinforcement Learning". In: (2020), p. 25.

[Kuu+20] Joel Kuula et al. "Laboratory Evaluation of Particle-Size Selectivity of Optical Low-Cost Particulate Matter Sensors". In: *Atmospheric Measurement Techniques* 13.5 (May 15, 2020), pp. 2413–2423. ISSN: 1867-1381. DOI: 10.5194/amt-13-2413-2020. URL: https://amt.copernicus.org/articles/13/2413/2020/ (visited on 02/10/2022).

[LYE20] Antti Lajunen, Yinye Yang, and Ali Emadi. "Review of Cabin Thermal Management for Electrified Passenger Vehicles". In: *IEEE Transactions on Vehicular Technology* 69.6 (June 2020), pp. 6025–6040. ISSN: 1939-9359. DOI: 10.1109/TVT.2020.2988468.

[MSA20] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: 100,000 Time Series and 61 Forecasting Methods". In: *International Journal of Forecasting*. M4 Competition 36.1 (Jan. 1, 2020), pp. 54–74. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2019.04.014. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301128 (visited on 06/30/2022).

[MJM20] Peter R. Michael, Danvers E. Johnston, and Wilfrido Moreno. "A Conversion Guide: Solar Irradiance and Lux Illuminance". In: *Journal of Measurements in Engineering* 8.4 (4 2020), pp. 153–166. ISSN: 24244635. DOI: 10.21595/jme.2020.21667. URL: https://www.mendeley.com/catalogue/59b60244-f03f-3d08-9650-757f50111b1b/ (visited on 01/27/2023).

[SMA20] Gaetano Settimo, Maurizio Manigrasso, and Pasquale Avino. "Indoor Air Quality: A Focus on the European Legislation and State-of-the-Art Research in Italy". In: *Atmosphere* 11.4 (4 Apr. 2020), p. 370. DOI: 10.3390/atmos11040370. URL: https://www.mdpi.com/2073-4433/11/4/370 (visited on 03/16/2021).

[The20] Thermetrics. *Full Body Manikins | Thermetrics*. 2020. URL: https://thermetrics.com/products/full-body-manikins (visited on 06/29/2020).

[Try+20] Jessica Tryner et al. "Effects of Aerosol Type and Simulated Aging on Performance of Low-Cost PM Sensors". In: *Journal of Aerosol Science* 150 (Dec. 1, 2020), p. 105654. ISSN: 0021-8502. DOI: 10.1016/j.jaerosci.2020.105654. URL: https://www.sciencedirect.com/science/article/pii/S0021850220301415 (visited on 03/05/2021).

[vEW20] Nees Jan van Eck and Ludo Waltman. "VOSviewer Manual". In: (Apr. 2020), p. 53.

[Vin+20] Ricardo Vinuesa et al. "The Role of Artificial Intelligence in Achieving the Sustainable Development Goals". In: *Nature Communications* 11.1 (1 Jan. 13, 2020), p. 233. ISSN: 2041-1723. DOI: 10.1038/s41467-019-14108-y. URL: https://www.nature.com/articles/s41467-019-14108-y (visited on 06/01/2022).

[Wak+20] Nicole I. Wakim et al. "Choosing the Right Time Granularity for Analysis of Digital Biomarker Trajectories". In: *Alzheimer's & Dementia: Translational Research & Clinical Interventions* 6.1 (2020), e12094. ISSN: 2352-8737. DOI: 10.1002/trc2.12094. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/trc2.12094 (visited on 02/11/2022).

[WH20] Zhe Wang and Tianzhen Hong. "Reinforcement Learning for Building Controls: The Opportunities and Challenges". In: *Applied Energy* 269 (July 1, 2020), p. 115036. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2020.115036. URL: https://www.sciencedirect.com/science/article/pii/S0306261920305481 (visited on 01/14/2023).

[War+20] Alok Warey et al. "Data-Driven Prediction of Vehicle Cabin Thermal Comfort: Using Machine Learning and High-Fidelity Simulation Results". In: *International Journal of Heat and Mass Transfer* 148 (Feb. 1, 2020), p. 119083. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2019.119083. URL: https://www.sciencedirect.com/science/article/pii/S0017931019336348 (visited on 03/09/2022).

[Zot20] Zotero. *Zotero | Your Personal Research Assistant*. 2020. URL: https://www.zotero.org/ (visited on 01/28/2020).

[Agu21] Davide Aguiari. "Exploring Computing Continuum in IoT Systems: Sensing, Communicating and Processing at the Network Edge." PhD thesis. Cagliari, Italy: University of Cagliari–Sorbonne Université, 2021.

[Bos21] Bosch. *BME280*. Bosch Sensortec. 2021. URL: https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/ (visited on 11/08/2021).

[Bru21a] James Brusey. *DOMUS Gym Environment for Ml-Sim*. Dec. 8, 2021. URL: https://github.com/jbrusey/domus_gym (visited on 01/26/2023).

[Bru21b] James Brusey. *DOMUS ML-Simulator*. Nov. 11, 2021. URL: https://github.com/jbrusey/domus_mlsim (visited on 01/26/2023).

[Che21] Gaobo Chen. "Policy Gradient Reinforcement Learning-Based Vehicle Thermal Comfort Control". PhD thesis. Coventry University, 2021. URL: https://pureportal.coventry.ac.uk/en/studentthesis/policy-gradient-reinforcement-learningbased-vehicle-thermal-comfort-control(39f19f65-4ee2-4571-b6bc-3de26cbf9fda).html (visited on 05/16/2022).

[dAlf+21] Francesca Romana d'Ambrosio Alfano et al. "Mean Radiant Temperature Measurements through Small Black Globes under Forced Convection Conditions". In: *Atmosphere* 12.5 (5 May 2021), p. 621. DOI: 10.3390/atmos12050621. URL: https://www.mdpi.com/2073-4433/12/5/621 (visited on 11/03/2021).

[21] *Global EV Outlook 2021*. 2021, p. 101.

[KTM21] Jaedeok Ko, Kyaw Thu, and Takahiko Miyazaki. "Transient Analysis of an Electric Vehicle Air-Conditioning System Using CO2 for Start-up and Cabin Pull-down Operations". In: *Applied Thermal Engineering* 190 (May 25, 2021), p. 116825. ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2021.116825. URL: https://www.sciencedirect.com/science/article/pii/S135943112100274X (visited on 02/25/2022).

[Kon+21] Hari Kishan Kondaveeti et al. "A Systematic Literature Review on Prototyping with Arduino: Applications, Challenges, Advantages, and Limitations". In: *Computer Science Review* 40 (May 1, 2021), p. 100364. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2021.100364. URL: https://www.sciencedirect.com/science/article/pii/S1574013721000046 (visited on 03/18/2021).

[Lia21] Lu Liang. "Calibrating Low-Cost Sensors for Ambient Air Monitoring: Techniques, Trends, and Challenges". In: *Environmental Research* 197 (June 1, 2021), p. 111163. ISSN: 0013-9351. DOI: 10.1016/j.envres.2021.111163. URL: https://www.sciencedirect.com/science/article/pii/S0013935121004576 (visited on 12/03/2021).

[Mak+21] Spyros Makridakis et al. "The M5 Uncertainty Competition: Results, Findings and Conclusions". In: *International Journal of Forecasting* (Dec. 9, 2021). ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2021.10.009. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001722 (visited on 06/30/2022).

[Opt21] Optemus. *Optemus - Optimised and Systematic Energy Management in Electric Vehicles*. Optemus. 2021. URL: http://www.optemus.eu/ (visited on 12/24/2021).

[Raf+21] Antonin Raffin et al. "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. ISSN: 1533-7928. URL: http://jmlr.org/papers/v22/20-1364.html (visited on 06/30/2022).

[Ren+21] Jinjing Ren et al. "Fine-Granularity Urban Microclimate Monitoring Using Wearable Multi-Source Sensors". In: *Sustainability* 13.24 (24 Jan. 2021), p. 14062. ISSN: 2071-1050. DOI: 10.3390/su132414062. URL: https://www.mdpi.com/2071-1050/13/24/14062 (visited on 02/11/2022).

[Rom+21] Christian Rommelfanger et al. "Linearization of Thermal Equivalent Temperature Calculation for Fast Thermal Comfort Prediction". In: *Energies* 14.18 (18 Jan. 2021), p. 5922. DOI: 10.3390/en14185922. URL: https://www.mdpi.com/1996-1073/14/18/5922 (visited on 10/24/2021).

[Rus+21] Luigi Russi et al. "Air Quality and Comfort Characterisation within an Electric Vehicle Cabin". In: *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*. 2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive). July 2021, pp. 169–174. DOI: 10.1109/MetroAutomotive50197.2021.9502853.

[VB21] Ricardo Vinuesa and Steven L. Brunton. "The Potential of Machine Learning to Enhance Computational Fluid Dynamics". Oct. 5, 2021. DOI: 10.48550/arXiv.2110.02085. arXiv: 2110.02085 [physics]. URL: http://arxiv.org/abs/2110.02085 (visited on 06/01/2022).

[Yu+21] Liang Yu et al. "A Review of Deep Reinforcement Learning for Smart Building Energy Management". In: *IEEE Internet of Things Journal* 8.15 (Aug. 2021), pp. 12046–12063. ISSN: 2327-4662. DOI: 10.1109/JIOT.2021.3078462.

[Zau+21] Stefano Zauli-Sajani et al. "Assessment of Air Quality Sensor System Performance after Relocation". In: *Atmospheric Pollution Research* 12.2 (Feb. 1, 2021), pp. 282–291. ISSN: 1309-1042. DOI: 10.1016/j.apr.2020.11.010. URL: https://www.sciencedirect.com/science/article/pii/S1309104220303330 (visited on 10/21/2021).

[ZCM21] Yangyang Zou, Jordan D. Clark, and Andrew A. May. "A Systematic Investigation on the Effects of Temperature and Relative Humidity on the Performance of Eight Low-Cost Particle Sensors and Devices". In: *Journal of Aerosol Science* 152 (Feb. 1, 2021), p. 105715. ISSN: 0021-8502. DOI: 10.1016/j.jaerosci.2020.105715. URL: https://www.sciencedirect.com/science/article/pii/S0021850220302007 (visited on 12/03/2021).

[Edw22] Chris Edwards. "Neural Networks Learn to Speed up Simulations". In: *Communications of the ACM* 65.5 (Apr. 25, 2022), pp. 27–29. ISSN: 0001-0782. DOI: 10.1145/3524015. URL: https://doi.org/10.1145/3524015 (visited on 05/26/2022).

[Jes+22] Brandi Jess et al. "Fast, Detailed, Accurate Simulation of a Thermal Car-Cabin Using Machine-Learning". In: *Frontiers in Mechanical Engineering* 8 (2022). ISSN: 2297-3079. URL: https://www.frontiersin.org/article/10.3389/fmech.2022.753169 (visited on 03/11/2022).

[MSA22] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "M5 Accuracy Competition: Results, Findings, and Conclusions". In: *International Journal of Forecasting* (Jan. 11, 2022). ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2021.11.013. URL: https://www.sciencedirect.com/science/article/pii/S0169207021001874 (visited on 06/30/2022).

[Rus+22] Luigi Russi et al. "Air Quality and Comfort Characterisation within an Electric Vehicle Cabin in Heating and Cooling Operations". In: *Sensors* 22.2 (2 Jan. 2022), p. 543. ISSN: 1424-8220. DOI: 10.3390/s22020543. URL: https://www.mdpi.com/1424-8220/22/2/543 (visited on 05/24/2022).

[23] *DOMUS Project, a H2020 Funded Research to Increase the Electric Range of EV*. DOMUS PROJECT. 2023. URL: https://www.domus-project.eu/ (visited on 01/08/2023).

[140] 14:00-17:00. *ISO/TS 14505-1:2007*. ISO. URL: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/82/38265.html (visited on 02/17/2020).

[Dim] Manol Dimovski. "Alternative Solutions in Automotive HVAC Systems. Comfort, Efficiency and Sustainability in Car Cabin Temperature Control". URL: https://www.politesi.polimi.it/bitstream/10589/138779/5/2017_12_Dimovski_Manol.pdf (visited on 06/26/2020).

[Max] Maxim. *Guidelines for Reliable Long Line 1-Wire Networks*. URL: https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/148.html (visited on 10/14/2021).