U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Human physiological simulator: Towards a personalized digital-twin human simulator for different scenarios

Lucas Teixeira Souza
Master in Engineering Physics
Department of Physics and Astronomy
Faculty of Sciences and Faculty of Engineering of the University of Porto
2026

Human physiological simulator: Towards a personalized digital-twin human simulator for different scenarios

Lucas Teixeira Souza

Dissertation carried out as part of the Master in Engineering Physics

Department of Physics and Astronomy

2026

Supervisor

Prof. Dr. João Paulo Cunha, Full Professor at the Faculty of Engineering of the University of Porto (FEUP), senior researcher at the INESC TEC and coordinator of the Center for Biomedical Engineering Research (C-BER)

Co-supervisor

Duarte Dias, Research at INESC TEC and assistant coordinator of the Center for Biomedical Engineering Research (C-BER)

Acknowledgments

Esta dedicação que faço é para todos aqueles que estiveram presentes em minha vida, e que, de alguma forma, caminharam ao meu lado nesta jornada.

Primeiramente, agradeço e dedico esta tese aos meus pais, que sempre me apoiaram em minhas decisões. À minha mãe, Vilma Teixeira Silva e Souza, em quem sempre pude confiar e encontrar o meu porto seguro, e cujo amor sempre pude sentir comigo. Ao meu pai, Silvio Cássio de Souza, que foi o primeiro a me incentivar e a me encorajar a atravessar o Atlântico em busca do conhecimento, sempre com parcimónia e sabedoria. Aos meus pais, que sempre estão em meus pensamentos, deixo a minha mais profunda gratidão, pois são a minha maior inspiração de vida, tanto pela trajetória que construíram quanto pelos desafios que enfrentaram individualmente. Portanto, digo, com todo o meu carinho; muito obrigado.

Aos meus irmãos, Guilherme Henrique Teixeira Souza e Leonardo Teixeira Souza, que estiveram presentes comigo durante todos os meus desafios e sempre me apoiaram. A todo o restante da minha família, meus tios, primos e avós agradeço pelos incentivos, que mesmo à distância sempre soube que torciam e estavam ao meu lado.

Aos meus orientadores, agradeço pelo suporte que me proporcionaram durante toda a construção e desenvolvimento desta tese. Ao Professor Doutor João Paulo Cunha, que, com o seu conhecimento, a sua paciência e a sua sabedoria, contribuiu significativamente para a concretização deste trabalho. Ao Duarte Filipe Dias, que, com os seus conselhos, a sua paciência e a sua empatia, contribuiu para o meu desenvolvimento académico e pessoal, auxiliando-me ao longo de todo este processo. Agradeço a ambos pela oportunidade, pela orientação e pelo apoio prestado.

Aos amigos que fiz durante esta jornada tão desafiadora e com quem partilhei angústias e dificuldades que, por vezes, julgava serem apenas minhas, expresso o meu agradecimento por me ouvirem e apoiarem. De todos os amigos e colegas que fizeram parte do meu percurso, agradeço em particular àqueles que estiveram comigo desde o início e que sempre me incentivaram, em particular ao Lorenzo Santini e a Joyce Aquino que estiveram comigo do princípio ao fim.

Finalmente, e com a grande felicidade, agradeço a Deus, pois sem Ele jamais teria forças e coragem para enfrentar os desafios diários, sempre na certeza de que “O

fardo é proporcional às forças, assim como a recompensa será proporcional à resignação e à coragem”.

Resumo

O surgimento dos *Virtual Human Twins* representou um avanço significativo para a sociedade, ao possibilitar a integração de modelos computacionais capazes de representar, de forma sistemática, diferentes aspetos do funcionamento do corpo humano. Com o avanço desses modelos e a incorporação de dados reais provenientes de indivíduos específicos, surgem os *Human Digital Twins*, definidos como representações digitais individualizadas que descrevem, de maneira detalhada e dinâmica sistemas humanos individuais.

Visto que a maioria das plataformas e estudos relacionados aos *Human Digital Twins* tem sido predominantemente direcionada a aplicações clínicas, como suporte a decisões médicas, intervenções terapêuticas ou cenários de urgência, a investigação de casos não clínicos permanece relativamente pouco explorada na literatura científica. Assim, o desenvolvimento e o aprofundamento de *Human Digital Twins* voltados para aplicações não clínicas configuram-se como uma oportunidade relevante de investigação, permitindo a expansão do estado da arte e a exploração de novos domínios de aplicação.

Este estudo tem como objetivo estudar a lacuna existente nas aplicações não clínicas dos *Human Digital Twins*, propondo uma metodologia baseada na aquisição e análise de dados reais relevantes. Esses dados são posteriormente utilizados no desenvolvimento de modelos computacionais que podem ser integrados em plataformas de software existentes, melhorando, assim, a representatividade e a aplicabilidade dos *Human Digital Twins* em cenários não clínicos. Por fim, foi desenvolvido o PhysTwin, uma plataforma de software para a personalização de modelos de pacientes destinados a aplicações não clínicas.

Palavras-chave: Gémeo Humano Virtual, Gémeo Humano Digital, Casos não clínicos, Fisiologia Computacional, Modelos Matemáticos do Corpo Humano, PhysTwin

Abstract

The emergence of *Virtual Human Twins* has represented a significant advancement for society, as it has enabled the integration of computational models capable of systematically representing different aspects of human body functioning. With the evolution of these models and the incorporation of real-world data derived from specific individuals, *Human Digital Twins* have emerged, defined as individualized digital representations that describe human systems in a detailed and dynamic manner.

Given that most platforms and studies related to *Human Digital Twins* have been predominantly directed toward clinical applications, such as medical decision support, therapeutic interventions, or emergency scenarios, the investigation of non-clinical use cases remains relatively underexplored in the scientific literature. Consequently, the development and further advancement of *Human Digital Twins* oriented toward non-clinical applications constitute a relevant research opportunity, enabling the expansion of the state of the art and the exploration of new application domains.

This study aims to address the gap in non-clinical *Human Digital Twin* applications by proposing a methodology based on the acquisition and analysis of relevant real-world data. These data are then used for the design of computational models that can be integrated into existing software platforms, thereby improving the representation and applicability of *Human Digital Twins* in non-clinical scenarios. Finally, the PhysTwin was developed as the final software platform, enabling patient personalization for non-clinical use cases.

Keywords: Virtual Human Twin, Human Digital Twin, Non clinical cases, Computational physiology, Human Body Mathematical models, PhysTwin

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BRAIN Biomedical Research and Innovation. 1

CFT Cold Face Test. 53

DT Digital Twin. 1

ECG Electrocardiogram. 13

HDT Human Digital Twin. 1

HR Heart Rate. 5, 18

INESCTEC Institute for Systems and Computer Engineering, Technology and Science. 1

MAE Mean Absolute Error. 68

MET Metabolic Equivalent Task. 78

ODEs Ordinary Differential Equations. 9

PCr Phosphocreatine. 48

PPE Pulse Physiology Engine. 15

UI User Interface. 35

UX User Experience. 35

VHT Virtual Human Twin. 1

VPH Virtual Physiological Human. 2

1. Introduction

In this section, the theme of the study will be introduced, including the motivation, objectives, and contributions to the field. It aims to clearly present the research problem and highlight the significance of addressing it. This dissertation was carried out at the Biomedical Research and Innovation (BRAIN) group at the Institute for Systems and Computer Engineering, Technology and Science (INESCTEC).

1.1. Motivation

The study of human physiology is of profound importance, given its central role in advancing medical knowledge. Consequently, both real and virtual approaches to understanding human physiology hold significant value for society. In particular, the development of virtual human twins provides a foundation for simulating and analyzing the behavior of the human body under specific conditions. For this purpose, several software platforms have been created to explore this field, aiming to replicate physiological processes with the highest possible precision [7].

A Virtual Human Twin (VHT) is defined as a digital representation of an individual's health or disease state. VHTs are developed using computational models, and are designed to mimic and predict the behavior of their physical counterparts, including interactions with comorbidity and additional health conditions [8]. The integration of Digital Twin (DT) technology into medicine represents a transformative paradigm shift toward precision and personalized healthcare. Originally developed in engineering as virtual replicas of physical systems, digital twins are now being applied to biomedicine to construct patient-specific, dynamic models capable of predicting disease progression, optimizing therapeutic strategies, and improving clinical decision making [8].

Human Digital Twin (HDT) and VHT are conceptually related, as both rely on computational models to represent human physiological behavior. HDT's are typically individualized and continuously updated with real-world data to enable monitoring and prediction, whereas VHT's provide high fidelity physiological and anatomical models.

Thus, the development and use of reliable HDT is of great importance for medical purposes, particularly in the context of precision medicine. Such models can also serve as complementary tools to support clinical decision making and provide an additional reference during medical discussions.

By combining heterogeneous data sources, including multi-omics, imaging, electronic health records, wearable devices, and environmental factors, HDTs enable a “patient-in-silico” that evolves alongside its real-world counterpart, thereby supporting real-time monitoring and individualized treatment planning [1]. These high resolution virtual models can be computationally treated with thousands of potential therapies to identify the most effective interventions for a specific patient, bridging the gap between complex molecular mechanisms and clinical application.

Furthermore, the fusion of artificial intelligence with mechanistic modeling enhances predictive accuracy while preserving interpretability, allowing HDTs to integrate causal biological relationships and adapt to dynamic patient responses. Despite challenges, including data integration, model standardization, and the representation of multiscale biological complexity, medical digital twins hold the potential to redefine personalized medicine by enabling individualized diagnostics, proactive interventions, and tailored therapeutic strategies across diverse disease contexts.

The use and study of personalized digital twin human simulators across different scenarios opens new avenues of research. Such simulators are particularly valuable in non clinical contexts, where individuals may wish to model ordinary activities, such as walking or running for a specified duration, and observe the resulting physiological responses. Most existing studies on virtual human physiological simulators, however, have focused primarily on clinical applications, addressing urgent medical needs such as surgeries, severe injuries, or other critical conditions [1].

In 1993, the creation of the Physiome Project highlighted how society can benefit from advancements in human body simulation. This initiative includes the Virtual Physiological Human (VPH), a global effort aimed at developing next generation computational technologies capable of integrating all available patient specific information and generating predictive models that simulate how an individual’s health may evolve under specific conditions [9], since then, the existing software platforms for human physiology simulation are primarily designed for clinical applications, such as BioGears and others [10, 11, 12], so these tools are highly valuable in the medical field, as they enable the simulation of scenarios including asthma attacks, airway obstructions, anesthesia, traumatic brain injuries, sepsis, and other critical conditions. Therefore, the open-source physiological simulation platforms provide advanced frameworks, but their primary design is oriented toward clinical and pathological use cases.

The availability of open-source physiological simulation platforms has grown substantially in recent years, enabling increasingly sophisticated representations of human systems and processes. In recognition of this scientific and technological progress, the European Union announced in December 2023 the creation of the European Virtual Human Twins Initiative, a large scale research program aimed at developing integrated, multi scale, and interoperable digital replicas of the human body to support personalized and predictive healthcare. This initiative represents a natural evolution of the digital twin paradigm applied to biomedicine, as envisioned by [13] in their proposal of the Virtual Human Twin as a “moon-shot project” for digital health research. According to the authors, the VHT initiative aspires to establish a unifying framework where data-driven and mechanistic models can coexist, fostering reproducibility, accessibility, and collaboration across the European research landscape [13].

This work will focus on personalized digital human simulators in non clinical scenarious, which is a gap and unexplored area. The research in this area offers a novel approach to advancing disease prevention, enhancing physical performance, and promoting overall well-being. Developing such simulators for non clinical applications can provide valuable insights into the functioning of the human body under normal conditions, such as recovery following physical exercise and the onset of fatigue. The current limitations of open-source physiological platforms highlight the need to extend their functionalities and adapt existing capabilities to novel and previously unexplored contexts, for example, resting states, post-exercise physiological responses, and immersive applications such as virtual reality environments [1].

1.2. Objectives

This work aims to investigate the open-source physiological simulation platforms for human body simulation and explore potential applications that are not yet fully developed, researching and developing new approach and modification on the existing model, building upon their clinical knowledge and adding non clinical knowledge based on real data to be able to model the human in different scenarios such as modeling the recovery state following physical activity. It is expected that new functionalities of the system will be demonstrated, as well as existing capabilities applied to novel and previously unexplored scenarios, for example, rest states, post-exercise physiological behavior, virtual reality scenarios, HDT simulators, etc. The goal is to highlight new features and potential uses of the software in the context of different activities, as well as to identify future implemen-

tations within the research center research lines, aiming at its application in subsequent research and development activities.

To reach such objective it will be needed to combine different backgrounds such as biodesign approaches, system/electronics development and/or integration, physics, mathematical equations, signal processing, machine learning methodologies, cloud programming and system results analysis. Our objectives are summarized as:

- Study and understand the open-source physiological simulation platforms and their limitations to implement new functionalities and optimize existing ones.
- Study how to organize and integrate real and simulated data into the proposed system.
- Define, design, and prepare test scenarios for experimental evaluation.
- Design a model to the resting phase after physical exercise and validate its accuracy in different scenarios.
- Apply signal processing and machine learning techniques to analyses data acquired during real-world monitoring scenarios.
- Collect data and perform real-world experiments with participants in order to obtain real-world data for the validation of the proposed mathematical equations.
- Generate and conduct several real-world tests to evaluate the performance and reliability of the proposed system.
- Study how to personalize the system for each user by considering different data sources and input modalities, as well as the validation of these data.
- Public release of the code to foster reproducibility and future work in the community.

1.3. Contributions

We believe that the contributions to the field will be diverse. First, the development and study of non clinical human virtual twins is already a contribution, considering the lack of research on this particular area. Additionally, it is important to note that improvements in software for non clinical cases will benefit several areas of health research and prevention, particularly in applications such as military scenarios, where it may be necessary to simulate a day under stress or in extreme temperatures. Improving the study and understanding of the human body under non clinical scenarios contributes to disease

prevention and enables a more informed and personalized understanding of each patient. Thus, we hope to make a meaningful contribution to this field.

Considering the existing gap in the literature, this work aims to contribute to the advancement of HDT. The updated versions incorporate new methodologies for calculating Heart Rate (HR), arterial pressure, and lactate concentration. Furthermore, this thesis discusses the development and implementation of a new function within the software, which, to the best of our knowledge, is not currently available in existing systems.

As a final outcome, this work presents a HDT capable of operating continuously in real time, 24 hours per day, while dynamically responding to different stimuli and actions, namely physical exercise, running, stress, and other physiological conditions. In addition, the development of a real time user interface significantly enhances the usability and applicability of the system. Consequently, the contributions of this work to the field are substantial, particularly through the public release of the source code for future research and the detailed explanation of how new functions and functionalities can be integrated into the HDT in real time.

1.4. Dissertation Structure

This dissertation is organized into eight main chapters, structured to guide the reader through the development of the work and to facilitate a comprehensive understanding of the topic.

The first chapter presents an introduction to the theme and its scientific relevance. It outlines the motivation for the study and the reasons that justify the importance of investigating this topic in greater depth. The objectives of the dissertation, both general and specific, are also defined, as well as the expected outcomes. Additionally, this chapter describes the main contributions that the dissertation aims to provide.

The second chapter, State of the Art, examines previous studies and research related to the subject. It summarizes the existing state of the art, identifying the most relevant findings and methodologies used in earlier work. This chapter is organized into subsections to clarify and contextualize each study reviewed in the literature. Additionally, the second chapter describes the limitations of the software studied, addressing the main objectives of this thesis.

The third chapter, Research Concept and Methodology, details the methodological framework adopted in this research. It explains the techniques, models, and procedures

used to conduct the preliminary study, as well as the rationale behind their selection. The use cases are defined in the contexts of physical activity, control room operations and agriculture. Each use case is further discussed and explained in detail. Furthermore, it presents the initial use case diagram and describes how it will be progressively developed throughout the thesis.

The fourth chapter, Twin Approximation Based in Real Data, presents the first concrete outcome of the thesis, namely the creation of a large dataset and a database consisting of several diverse patients performing different activities. Following this, a user interface was developed to enable practical interaction with the database, as well as an interpolation framework based only on real-world data.

The fifth chapter, Researching New Models to Enhance the BioGears Physiological Digital Human, presents and discusses all modifications made to the BioGears engine. It explains the rationale behind these modifications, as well as the expected outcomes. Finally, preliminary validations are performed to assess whether the implemented changes function as intended.

The sixth chapter, PhysTwin: A Personalized Human Digital Twin System, presents the core contribution of this thesis. In this chapter, the already modified BioGears engine is combined with the Twin Approximation based on real data approach. This enables the injection of real-world, patient specific data into the BioGears engine, resulting in a Personalized Human Digital Twin. Additionally, a user interface was developed to facilitate its use.

The seventh chapter, System Validation, presents the evaluation of the Personalized Human Digital Twin. This chapter assesses whether the model behaves as intended and identifies potential gaps and limitations. These findings are documented and discussed in the preceding chapter. Therefore, this chapter focuses on validation of the proposed approach.

Finally, the eighth chapter, Conclusion, summarizes the work developed in this dissertation. It presents the final conclusions of the study, highlighting the achieved results and explicitly discussing the identified gaps and limitations. It also outlines directions for future work and possible extensions of the proposed approach.

2. State of the Art

In this section, the state of the art of various human body simulations methodologies will be discussed, including mathematical models, data-driven models in Human Digital Twins and Virtual Human Twins. Furthermore, the section will address relevant use cases that will be further investigated and explored. The aim is to critically review and synthesize the existing literature and previous research in the field.

2.1. Virtual Human Twin (VHT)

The study of human body simulation models represents a significant milestone for our society, as it demonstrates the interest and willingness of researchers to deepen our understanding of the human body and its diverse functionalities, these tools are highly valuable in the medical field, as they enable the simulation of diverse scenarios, such as DTs in cancer research and therapy, DTs in the context of network medicine, DTs in cardiology, DTs in surgery, and DTs in orthopedics [14].

In Figure 2.1, the possible design choices for virtual health systems are illustrated, highlighting the progression from generic digital models to personalized and dynamically updated Digital Twins, depending on personalization, real-time updating, decision support usage, and the level of human intervention.

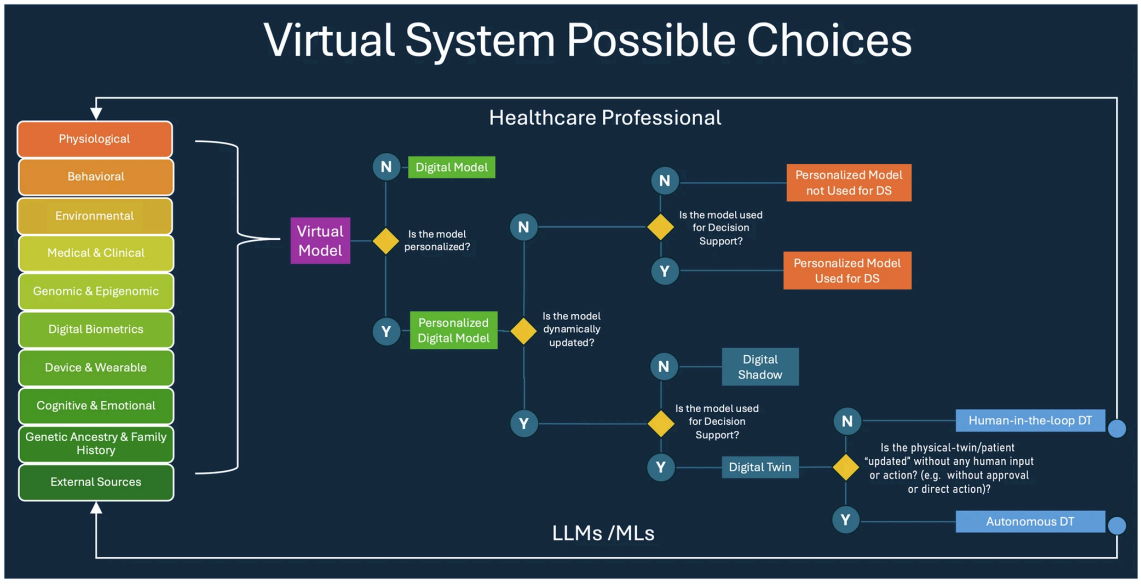

Figure 2.1: Data-flow into a virtual model. Reproduced from [1]

Following this data flow, the progression from a virtual model to a personalized digital model, with dynamic updating and decision support capabilities, leads to the Human Digital Twin concept. Therefore, a VHT can be understood as a generic, foundational

model of human physiology. When this model is subsequently personalized with an individual's real world data and endowed with capabilities for dynamic updating, it evolves into a HDT. Accordingly, this framework can be further detailed by focusing on the Digital Twin roadmap presented in Figure 2.2.

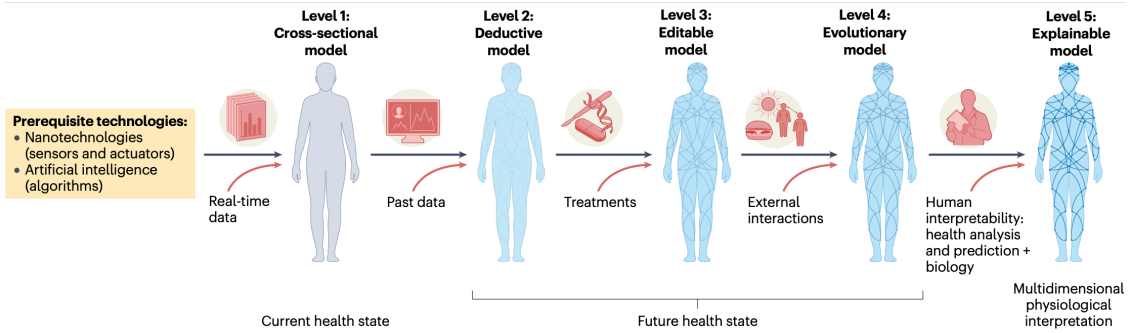

Figure 2.2: Five-level Digital Twin roadmap. Reproduced from [2].

Figure 2.2 illustrates the five-level roadmap of a Digital Twin. Each level represents a progressive increase in model complexity and system integration. The first level relies on real-time data acquisition, where system inputs are obtained directly from live measurements. The second level builds upon this foundation by incorporating both real-time and historical data to train predictive models through deductive and data-driven approaches.

The third level extends the previous ones by explicitly considering human interventions, such as medical treatments or drug administration. The fourth level further enhances the Digital Twin by accounting for interactions between the human body and the external environment. Finally, the fifth level integrates all preceding layers and employs explainable artificial intelligence techniques to uncover the underlying reasoning behind health analysis and prediction.

The challenges involved in the development of the VHT are numerous and multifaceted. The difficulties are compounded by the geographical fragmentation of the VHT research community, with expertise dispersed across various institutions and countries [14]. One of the most critical obstacles is the establishment of efficient mechanisms for data sharing, which must accommodate diverse legal, ethical, and institutional constraints while ensuring the protection of patient privacy and compliance with data governance regulations. Moreover, the standardization of models and algorithms remains a fundamental requirement. Without common ontologies, interoperable data formats, and widely accepted markup languages, the integration of heterogeneous computational models into a cohesive simulation framework becomes exceedingly complex [14].

Another key challenge is the creation of robust technological infrastructures capable of supporting high performance computing, large scale data storage, and seamless communication across distributed virtual laboratories. This includes the development of federated repositories for both biomedical data and computational models, as well as advanced tools for data exploration, visualization, and processing, such as the Multimode Application Framework (MAF). Additionally, modelling human physiology itself presents significant scientific hurdles, such as the integration of multi-scale processes, the incorporation of uncertainty, and the need for automatic processing of vast and diverse biomedical datasets. Addressing these challenges is essential for transforming the VPH from a theoretical concept into a practical, predictive tool that can support both research and clinical decision-making.

The primary objective of this thesis is to develop and advance the last two levels of the proposed roadmap in Figure 2.2. Since the software infrastructure and tools required for the first three levels are already available and demonstrate high accuracy in clinical cases, this work focuses on advancing the integration of environmental interactions. Consequently, this work is expected to contribute primarily to the fourth and fifth levels of the HDT framework.

2.2. Human Body Mathematical Simulation

A Human Body Mathematical Simulator is defined as a system that uses mathematical formulations, such as Ordinary Differential Equations (ODEs) to simulate and compute theoretical values of human physiological processes. Approximating the human body through a mathematical model is a useful approach because it enables the simulation, analysis, and validation of these equations. It is also common to construct electrical circuit analogues that replicate the behavior of the human body or of specific physiological subsystems.

The electric system uses the lumped-parameter model that represents the human body's systems. The approximation of physiological systems through lumped-parameter circuits has been extensively studied in previous research, and it has been shown to accurately reproduce the hemodynamic behavior of the arterial system [15]. This modeling approach enables the development of a system capable of simulating human physiological processes with high computational efficiency, significantly reducing both simulation time and resource requirements.

This type of model takes inputs such as age, sex, weight, and other relevant variables, processes them through its internal architecture, and produces specific outputs. These outputs correspond to physiological data, as illustrated in Figure 2.3.

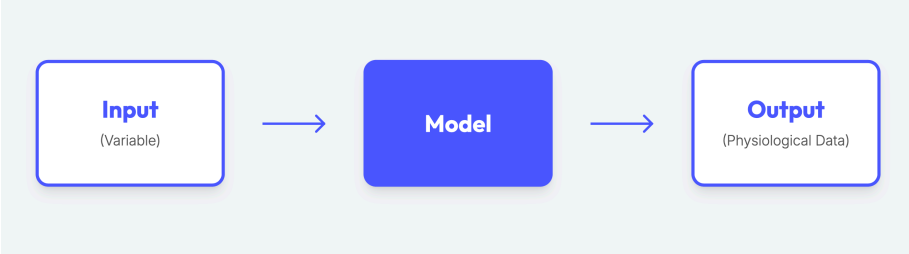

Figure 2.3: Model Architecture

Mathematical simulators of the human body have evolved into an essential methodology for analyzing physiological behavior under controlled, reproducible conditions. A widely adopted strategy is the use of lumped-parameter models, which approximate distributed physiological structures by collections of discrete elements such as resistances, compliances, masses, and inertances. This approach enables the transformation of complex physiological processes into ordinary differential equations that can be efficiently solved and systematically manipulated. For instance, mass–spring–damper formulations have long been used to characterize the mechanical response of the body under dynamic loading, effectively capturing the interaction among rigid and wobbling masses representing skeletal segments and soft tissues [16].

In cardiovascular simulation, lumped-parameter networks model vessels through electrical analogs resistances for viscous losses, compliances for vascular elasticity, and inductances for blood inertance, an approach that allows precise analysis of hemodynamic variables such as pressure, flow, and ventricular-vascular coupling. Recent work has demonstrated the power of these models in replicating full-body circulatory dynamics, including pathological states such as heart failure, cardiogenic shock, aortic stenosis, and mitral regurgitation, by tuning parameters of the time-varying elastance model and vascular impedances to match clinical conditions [17].

Extensions of the lumped-parameter framework have also incorporated cardiorespiratory interactions, showing that respiratory-induced fluctuations in intrathoracic pressure significantly modulate arterial waveforms, and validating these predictions against invasive brachial artery measurements in clinical patients [18].

Considering a whole body physiology engine/framework, there is a significant level of detail and several limitations that will be encountered when developing this kind of tool. In

this context, there are numerous open-source software tools to simulate a virtual human body, thus Nadeem et al. [19] conducted a comprehensive review to show the current state of the art in applying DTs and the technologies used to further their application in healthcare. These tools are therefore primarily focused on clinical scenarios, where the main goal is to simulate and accurately verify clinical cases, as can be seen in Table 2.1.

Software/Initiative	Focus	Specialization	Limitations
BioGears[1]	Whole-body physiological engine	Modular, real-time capable engine for simulating integrated human physiology	Limited accuracy for extended exercise or non clinical physiological responses
Pulse Physiology Engine[2]	Whole-body simulation framework	Enhanced computational stability, cross-platform support, and extended models for hemorrhage, exercise, and energy metabolism	Less documentation and fewer validation studies
VPH/Physiome Project[3]	Multiscale modeling and integration framework	Specialization in patient-specific modeling and cross-scale data integration	Not a ready-to-use simulation engine; requires extensive model coupling

Table 2.1: Comparison between major open-source physiological simulation frameworks and their specializations.

2.2.1. BioGears Physiology Engine

BioGears is a C++ based full body human physiological simulation engine that employs mathematical models to represent physiological systems. It enables the simulation of physiological responses to various conditions, including diseases, injuries, drug administration, and environmental changes. Through this capability, it is possible to model the behavior of multiple human body systems, such as the cardiovascular, respiratory, and

[1] https://www.biogearsengine.com
[2] https://pulse.kitware.com
[3] https://physiomeproject.org/about/the-virtual-physiological-human

nervous systems, among others. Additionally, medical conditions like anemia, pneumonia, and others can be simulated; and the software includes mathematical models for a wide range of systems, medical interfaces, and substances for real-time computation and retrieval of accurate physiology state [3].

The BioGears software allows for the simulation of drug or medication infusions, providing a versatile platform for various physiological and pharmacological studies. BioGears has been widely employed in medical research, including applications in sepsis modeling [20], burn injury simulations [21], and pharmacokinetic studies [22]. One of the main advantages of BioGears is that it does not require additional C++ programming, as its software architecture is designed with three layers of abstraction. This design facilitates integration and provides a more efficient and user-friendly application programming interface (API), as seen in Figure 2.4.

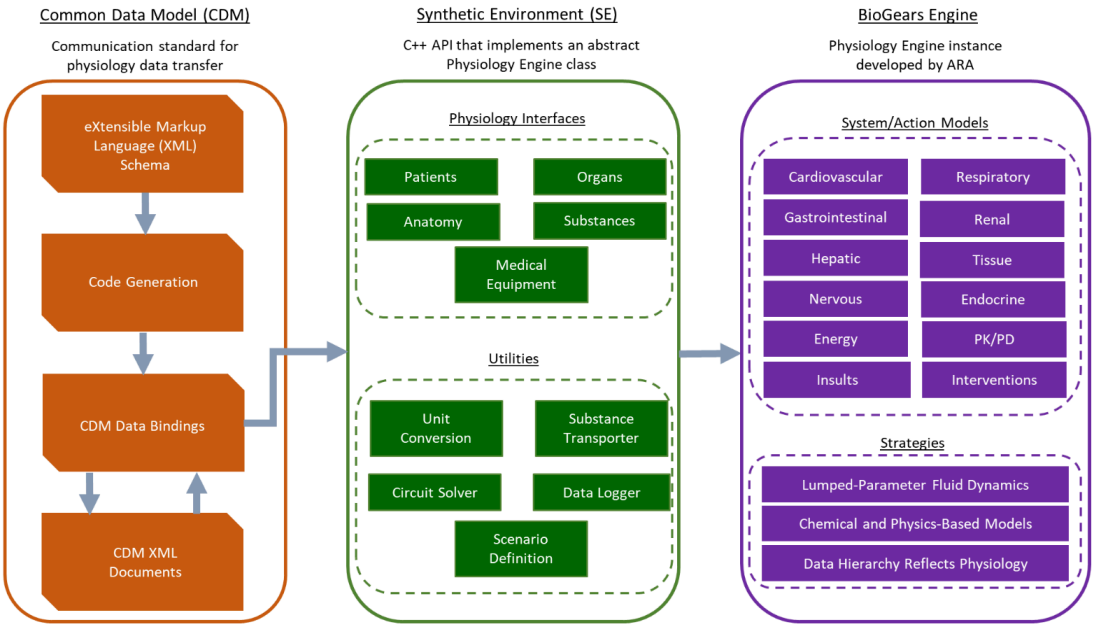

Figure 2.4: Overview of the BioGears engine software structure. Reproduced from [3].

BioGears operates by simulating a virtual patient subjected to specific physiological conditions. To achieve this, it allows the definition of patient parameters, enabling the customization of individual characteristics such as age, sex, weight, height, and other physiological attributes. Furthermore, it is possible to introduce additional conditions or interventions; its possible to initialize the patient with specific chronic and/or disease states via conditions, modify the patients external environmental conditions (weather, submerge in water, etc.), apply various actions (acute insults/injuries, interventions, conscious breathing, exercise, etc.) to be applied to the patient, and the patient can also

interact with equipment models, such as an Anesthesia and/or an Electrocardiogram (ECG) Machine as well as an Inhaler via actions, as demonstrate in Figure 2.5

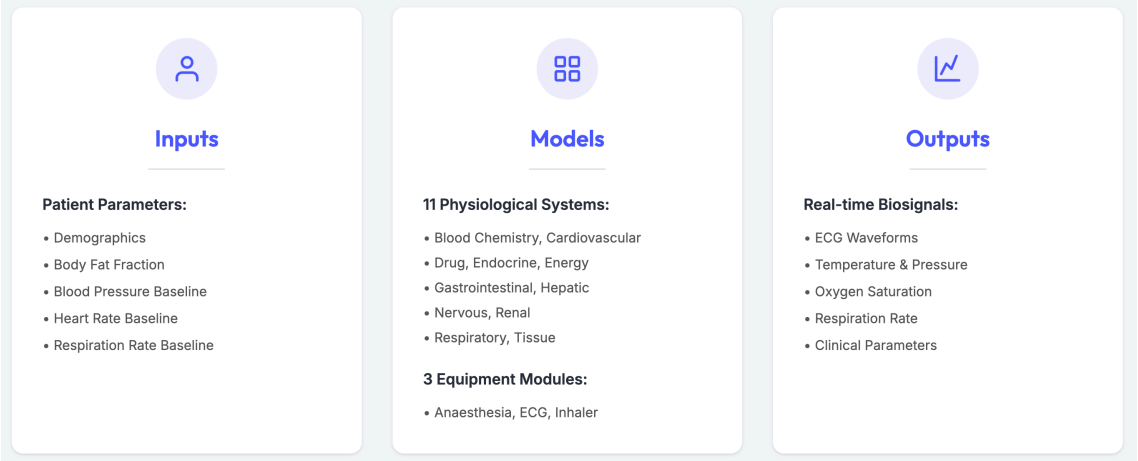

Figure 2.5: BioGears Model Architecture

The simulation within the BioGears software is performed using a lumped-parameter model that represents the circulatory and respiratory systems. As an example, Figure 2.6 illustrates the equivalent electrical circuit of the cardiac system. This circuit estimates blood pressure, flow, and volume in organs represented by multiple compartments. Each compartment is modeled using lumped parameter elements, such as resistors and capacitors, while inductors may also be included to account for inertial effects. Consequently, the system is discretized into nodes interconnected by paths.

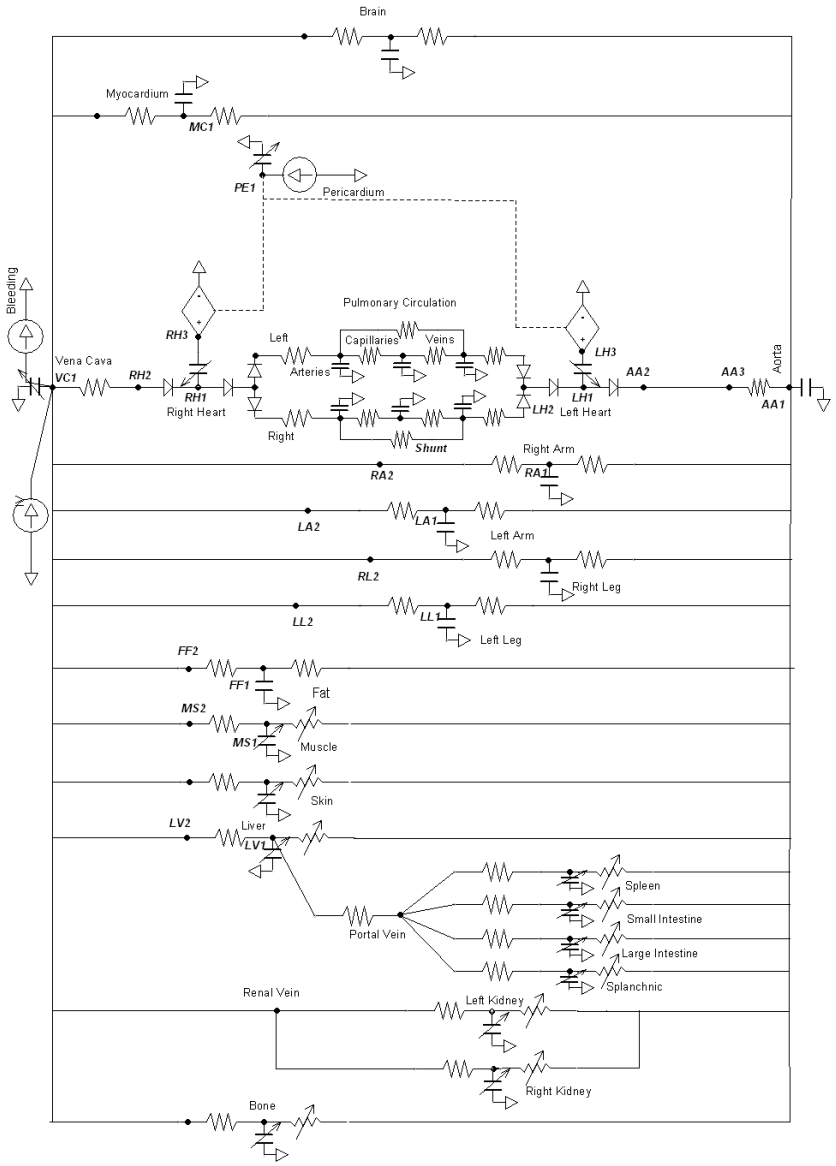

Figure 2.6: The cardiovascular equivalent electric circuit of BioGears Engine. Reproduced from [4].

Using BioGears, a wide range of long term scenarios, referred to as Showcase Scenarios, can be explored. Within the software, these are available as HowTo-Scenarios. More than 60 such scenarios are provided, allowing users to examine a variety of physiological conditions and analyze the resulting outputs. One example of a Showcase Scenario, available on the BioGears Engine website [5], is the simulation of an asthma attack, as illustrated in Figure 2.7.

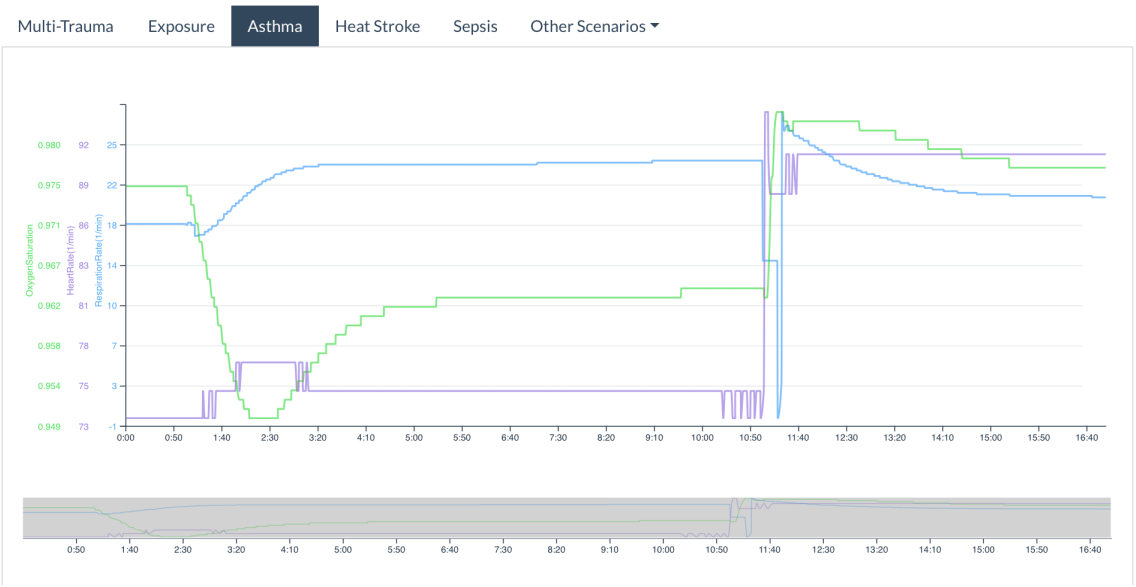

Figure 2.7: ShowCase Scenarios of Asthma Attack. Reproduced from [5].

This demonstrates the capabilities of the BioGears Engine, which makes it possible to simulate clinical scenarios with high precision and to evaluate the outcomes across multiple physiological systems and variables.

2.2.2. Pulse Physiology Engine

The Pulse Physiology Engine (PPE) started as a fork of the BioGears 6.1.1 Physiology Engine, and is a C++ based, open source, multi-platform, comprehensive human physiology simulator that drives medical education, research, and training technologies. The engine can be used as a standalone application or integrated with simulators, sensor interfaces, and models of all fidelities [11].

The major systems in the PPE use zero dimensional lumped-parameter circuit analogs, which means that there is no spatial geometry, so each organ is represented as a separate block, therefore PPE is a human mathematical simulator, given that it uses mathematical equations (lumped-parameter circuits) to simulate and generate data, and there is no continuous real data used during the process. The PPE consists of numerical models that represent various systems of the human body, including feedback mechanisms and interactions between these systems, pharmacokinetics/pharmacodynamics, and medical devices. Each system is described by differential equations, which are solved using transient analysis with a shared time step. Currently, the models execute with a time step of 20 ms, allowing all relevant physiological features to be captured while maintaining real-time simulation [23]. Inside the PPE it is possible to define patient parameters, such as height, weight, systolic and diastolic pressure; simulate specific

chronic and/or disease states via conditions; modify external environmental conditions; apply various actions (acute insults/injuries, interventions, conscious breathing, exercise, etc.) to the patient; have patient interact with equipment models, such as an Anesthesia and/or an ECG Machine.

The main difference between the BioGears Engine and the Pulse Physiology Engine lies in how simulations are executed and results are accessed. BioGears functions primarily as a computational software platform where physiological simulations are run and outputs are typically exported as CSV files for subsequent analysis. In contrast, PPE is designed as a full application that allows users to perform simulations interactively and visualize physiological data in real time. While both engines rely on mathematical models of human physiology, PPE emphasizes real-time interactivity and immediate feedback, making it particularly suitable for applications requiring dynamic exploration of physiological responses.

2.2.3. VPH/Physiome Project

In 1993, during a Bioengineering in Physiology commission held in Glasgow, the concept of the Physiome Project was introduced. The central idea was to establish a quantitative description of physiological dynamics and the functional behaviour of the intact organism. The project was therefore conceived with the primary goal of developing a multi-scale modelling framework capable of supporting the understanding of physiological function, enabling models to be combined and interconnected in a hierarchical manner. As illustrated in [12], electromechanical models of the heart, for example, require the integration of ion-channel models, myofilament mechanics, and signal-transduction pathways at the subcellular level, which must then be linked to models of tissue mechanics, wavefront propagation, and coronary blood flow, each of which may have been developed independently by different research groups.

According to the Physiome Project [12], the models developed within its framework can be described at several levels: the conceptual level, the mathematical level, the formulation level, and the solution level. Given the diversity of expertise and research groups involved, it becomes essential to establish a framework capable of unifying these components. For this reason, the Physiome Project proposes that the integrated structure, which includes MathML, FieldML, scripting modules, and organisational constructs such as components, variables, and imports, be referred to as “ModelML”.

The Physiome Project follows an open-source policy, grounded in the principle that its developments must support both public-good science and industrial participation. As a consequence, any modifications or extensions made within its ecosystem are required to be returned to the public domain. The project also provides a wide range of software tools, including:

- OpenCMISS-Iron: A distributed parallel mathematical modelling environment for complex bioengineering problems.
- OpenCMISS-Zinc:A library for building interactive modelling and 3-D visualisation applications
- Cmgui:Standalone application for visualising and interacting with mathematical field models.
- FieldML: An XML language and API for mathematical field model interchange.

Furthermore, the project encompasses several research efforts across different physiological systems of the body, including the cardiac, respiratory, and other major systems.

2.3. Human body simulation based on real data

The simulation of the human body using real data can be described as a methodology that aims to reproduce physiological behavior based on actual measurements. This approach relies on data obtained from wearables and other sensing devices to learn and model a patient's physiological dynamics. This type of research can be conducted using a variety of data-driven approaches, including the application of machine learning algorithms to model and simulate specific human body structures, as well as the analysis of real-world data to predict physiological responses to novel stimuli.

The primary advantage of a real human body data approach is that it does not require a large number of input parameters to initiate a simulation. Typically, it relies on a single specific metric (e.g., heart rate) and, based on the observed data, can simulate future states and make predictions. Consequently, it is not necessary to include additional inputs such as age, sex, or height. The model or algorithm itself learns from the data and refines its predictions accordingly. However, this approach can also introduce biases and limitations, as predictions are generally constrained to individual patients for whom data are available, limiting the generalizability of the model.

This chapter focuses on three use cases in which human body simulation based on real data can be applied, and discusses how these applications are implemented. The con-

sidered use cases include HR estimation, seizure related scenarios, and other relevant cases. For each use case, the state of the art is discussed.

To the best of our knowledge, there are currently no software platforms capable of providing a comprehensive, whole-body simulation driven entirely by real physiological data. Existing solutions are typically focused on specific organs or physiological subsystems, and the available multi-system models (e.g., BioGears) rely predominantly on mathematical and mechanistic formulations rather than real-time data-driven approaches. Consequently, the review and analysis of these tools and previous studies must be conducted on a subsystem by subsystem basis.

2.3.1. Heart Rate Estimation

The monitoring of HR is a key component in both general healthcare and preventive medicine, and it is particularly crucial for individuals engaged in sports and high performance activities [24]. The increasing availability and adoption of wearable devices enable the acquisition of continuous real-world ECG data from specific individuals, providing valuable real-time insights into their cardiovascular status. In this context, the ability to accurately model and predict HR has significant importance; for athletes, it may represent the difference between achieving optimal performance and experiencing potentially harmful overexertion.

Research on HR prediction has a long history and encompasses a wide range of computational approaches. Traditional techniques include AutoRegressive Integrated Moving Average (ARIMA), Random Forest (RF), Support Vector Machines (SVM), Kalman Filters, Singular Spectrum Analysis (SSA), and Copula-based methods, as well as a variety of neural network architectures.

Alsheikhy et al. [25], conducted a study in which HR was estimated through video recordings of a patient's hand. Their approach employed multiple deep learning architectures, including AlexNet, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and ResNet50V2, to infer physiological signals from visual data. The reported results were encouraging, with only two participants exhibiting noticeable measurement inaccuracies. Several data-driven approaches have been proposed to predict HR using physiological signals. For example, Saranya et al. [26] developed a method combining RPA and a Knowledge-Based Deep Neural Network with Stacked Autoencoders (KDNN-SAE), relying on physical contact through ECG acquisition. Pramukantoro et al. [27] proposed a heartbeat classification model based on RR-interval

measurements obtained from the Polar H10 wearable sensor, aiming at real-time and continuous monitoring. Additionally, Irfan et al. [28] implemented a hybrid approach using autoregressive modelling and Convolutional Neural Networks (CNNs), also requiring ECG contact, to predict cardiac arrhythmias, achieving classification accuracies of 98.37%, 99.59%, 98.41%, and 99.35%, respectively.

Overall, these studies illustrate the range of techniques and sensing modalities available for the prediction and assessment of HR using data-driven methods.

2.3.2. Epileptic Seizure

Epilepsy is a chronic neurological disorder that is characterized by recurrent and unpredictable seizures and millions of people are affect by it [29]. Thus, the study and better comprehension of seizure is essential to improve the patients quality of life. The detection of seizure os primarily made by electroencephalogram (EEG) due to its low cost, portability, and high temporal resolution [30].

Considering the increasing scientific interest in data-driven approaches for epilepsy management, Saadoon et al. [31], conducted a scoping review analyzing recent machine learning and deep learning solutions for seizure prediction based on EEG signals. Their results highlight the rapid growth of research focusing on integrating temporal and spectral feature representations into predictive models, particularly the use of advanced architectures such as CNNs and attention-based networks to improve robustness, sensitivity, and early-warning capability. This work reinforces the relevance of developing AI-based predictive systems to anticipate seizure onset and enable earlier clinical intervention.

In this research domain, Upadhyay et al. [32] proposed a HybridConvMobileNet architecture that integrates 1D-CNN and MobileNet structures to enable efficient and accurate seizure prediction, reporting 99.69% accuracy and 99.68% sensitivity. Additionally, Jana et al. [33] employed a 1D-CNN approach combined with the NSGA-II algorithm for seizure detection, while Wei et al. [34] introduced an AFC-GCN, a compact Graph Convolutional Network designed for robust detection performance. In another study, Kapoor et al. [35] developed an IoT-based framework incorporating a Deep-CNN and a hybrid cuckoo finch optimization algorithm to improve seizure prediction. Furthermore, Sonawane et al. [36] explored a deep CNN enhanced with a smart societal optimization algorithm, achieving a prediction accuracy of 86.73%.

Overall, the literature demonstrates a broad range of data-driven strategies for seizure prediction, reinforcing the continued growth and diversity of machine learning-based methodologies designed to anticipate epileptic events.

2.3.3. Others cases

Data-driven human simulators extend far beyond HR and seizure prediction. This represents a broad research domain with multiple additional applications. In this context, we provide a concise overview of other relevant works in which data-driven simulators have been successfully employed, demonstrating high predictive performance and accuracy.

Diabetes is a chronic disease that affects millions worldwide. It is a complex condition influenced by genetic, environmental, and dietary lifestyle factors. This underscores the need for a comprehensive, patient-centered diabetes management system that can be tailored to individual patients. In this context, Sarani et al. [37] developed a digital twin framework for personalized diabetes management. Their approach integrates a wide array of healthcare data sources, including electronic health records (EHRs), data from wearable devices, mobile health applications, and patient-generated data. This framework enables personalized blood glucose regulation, individualized glucose prediction, exploration of healthcare data through the PHKG user interface, and the generation of tailored dietary recommendations.

2.4. Other Approaches for Human Body Simulation

The simulation and prediction of human body responses are not exclusively achievable through mathematical or data-driven methods; alternative approaches can also be employed to achieve these objectives. A widely used approach involves Large Language Models (LLMs) to learn and predict patient states.

LLMs represent one of the most significant advancements in contemporary artificial intelligence. These models are based on transformer architectures [38], and are typically trained on extensive textual datasets, which endows them with the ability to understand and generate diverse, multilingual texts that resemble natural human language. Several pre-trained LLMs are already available, and it is common practice to fine-tune a specific LLM to adapt the pre-trained model using a domain-specific dataset, in order to develop an AI system tailored to a particular subject or application.

Considering this, Abelenda et al.[39] conduct a study and show a predicting glucose time series in patients diagnosed with diabetes type 1, using a LLM based model, with

a reasonable forecasting results, with similar MAE and RMSE values compared to novel and sophisticated ANN-based models, and with the lowest STD in both metrics.

In another study, Makarov et al. [40], used a large language model fine-tuned on clinical data to predict patient health trajectories, thereby enabling the creation of digital twins. Their model, named Digital Twin Generative Pretrained Transformer (DT-GPT), leveraged electronic health records without requiring data imputation or normalization, effectively addressing real-world data challenges such as noise and limited sample sizes.

Therefore, the application of LLMs to digital twin development in healthcare represents a promising methodological paradigm. However, this approach is accompanied by several notable constraints. First, LLMs fundamentally require text, based input, which necessitates the transformation of clinical data, often numeric or structured into textual representations. While feasible, this conversion may abstract or oversimplify biomedical details.

Second, LLMs typically demand extensive and diverse training datasets to achieve robust, generalizable performance, posing practical challenges in domains where large-scale, high-quality patient data may be limited or difficult to assemble.

Finally, while LLMs excel in generating natural language outputs, their primary output modality remains textual. In contexts requiring precise numerical predictions, statistical summaries, or quantitative simulations of patient states, such as forecasting laboratory trends or physiological parameters, a purely textual output may lack the precision and interpretability afforded by numeric or graphical representations. This gap suggests that hybrid approaches, combining LLM based reasoning with numeric modeling frameworks, may be necessary for comprehensive clinical forecasting and digital twin implementation.

2.5. Limitations in BioGears Physiology Engine

The BioGears Physiology Engine is a powerful software platform and research tool for simulating clinical scenarios across diverse patient profiles, as described in more details in Section 2.2.1. However, when considering non clinical contexts and the creation of highly personalized patients, certain limitations become apparent.

Since the BioGears Engine was primarily developed with clinical scenarios in mind, certain limitations become evident when simulating non clinical situations. While patient creation relies on the parameters described in the previous paragraph, aspects such as nutrition and food storage are not considered, which can be crucial under specific

conditions. Moreover, when simulating non clinical activities, such as a ten-minute run, the software may not respond as expected.

However, even under clinical test conditions, it is possible to identify some limitations of the software, as will be presented in the following sections. Therefore, several clinical and non clinical scenarios will be examined using the same configuration in order to evaluate the limits of BioGears. All of these discussions are intended to justify the limitations of the software that must be overcome to enable its long term use in non clinical scenarios.

Once the software was successfully installed and fully operational, it was possible to generate the How-To scenarios, which are pre-computed models provided by the software. The following scenarios have the same configuration, that is the generation of cases for a 22-year-old fit male soldier and a 44-year-old sleep-deprived male.

2.5.1. Asthma Attack

The first scenario analyzed was the Asthma Attack, the graphs shown in the Figure 2.8 represent the physiological response in terms of HR and inspiratory-expiratory rate.

From a pathophysiological standpoint, an acute asthma attack is characterized by bronchoconstriction, which elevates airway resistance [41]. This increased ventilatory load triggers a compensatory sympathetic response, manifested as an elevation in HR, visible in both panels immediately following the onset marker (red dashed line).

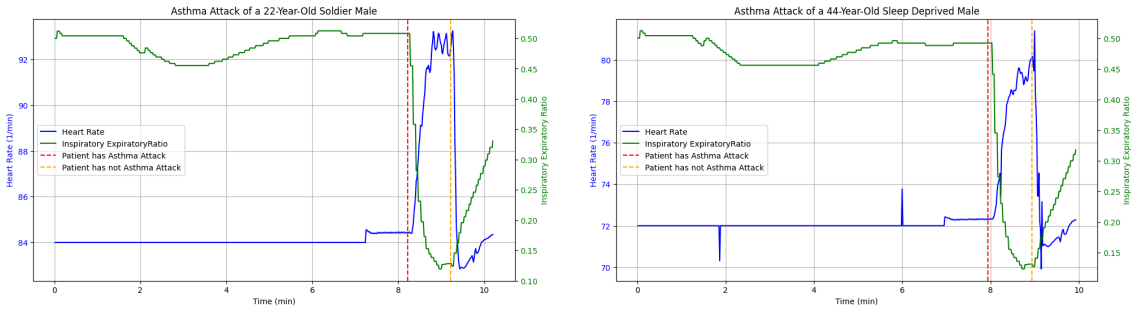

Figure 2.8: Asthma Attack Response

Both simulations converge to pre attack baseline values following pharmacological or spontaneous resolution, indicating that the BioGears engine correctly implements the underlying closed-loop cardiorespiratory control. It should be noted that these patient models are native to BioGears; the present analysis serves to verify that the engine produces physiologically consistent dynamic responses prior to the development of custom non clinical scenarios.

2.5.2. Brain Injury

Using the source code, it was also possible to generate the How-To scenario for a brain injury case. After running the simulation, the resulting plots are shown in the Figure 2.9, illustrating the brain injury response, specifically, the HR and the left eye pupillary response (pupil size modifier).

The two physiological variables tracked, serve as clinically meaningful proxies for the neurological state of the patient. Elevated intracranial pressure (ICP), as occurs in Traumatic Brain Injury (TBI), compresses the oculomotor nerve, disrupting parasympathetic tone to the iris sphincter and impairing the pupillary light reflex [42]. In BioGears, this is encoded as a reduction in the pupillary response size modifier, which in both panels drops sharply from unity toward zero following the onset of the high-severity injury event, consistent with the clinical presentation of a fixed, non-reactive pupil, a hallmark of uncal herniation.

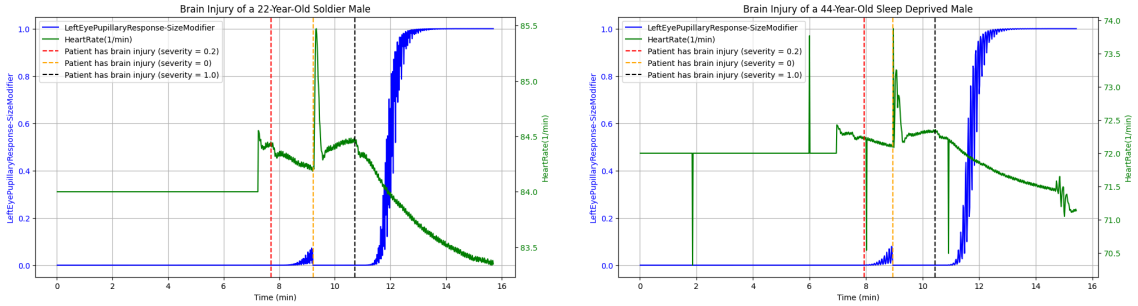

Figure 2.9: Brain Injury Response

With this, it is possible to observe once again that the BioGears engine is a reliable tool for clinical case simulations. Moreover, it is notable that there is a certain level of fidelity and accuracy in the results.

Taken together, the simulations demonstrate that BioGears implements a mechanistically coherent model of TBI progression, capturing both the neurological (pupillary) and cardiovascular (Cushing) components of the injury response. This lends confidence to the engine's suitability as a simulation substrate for the present work, prior to the introduction of custom non clinical scenarios.

2.5.3. Running Example

This represents a non clinical scenario in which the focus is on the post-exercise resting phase, an aspect that is both highly interesting and important to study, the two subjects in this case underwent a ten-minute running exercise. Analyzing the resting phase allows

for the observation of how the simulated patient responds to physical activity, providing insights into overall physiological condition and resilience.

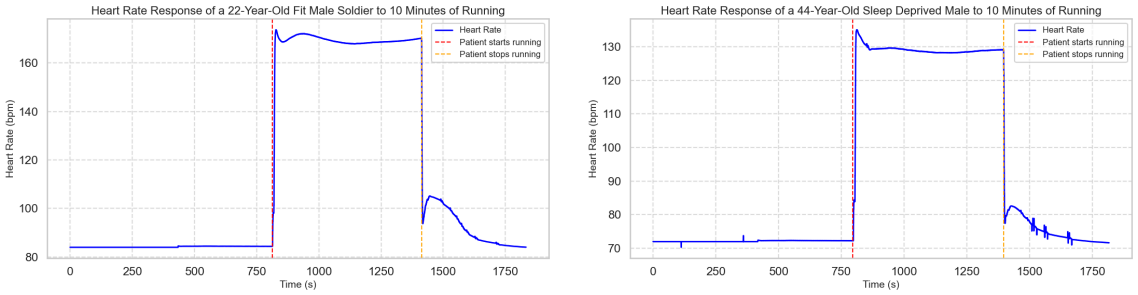

Figure 2.10: Heart Rate Response

Notable differences can be observed between the two graphs, particularly in the maximum HR values, which is expected given that the maximum HR can be defined by using Tanaka et. al, [43] formula, defined by Equation (2.1).

$$HR_{\max} = 220 - 0.7 \cdot \text{age} \tag{2.1}$$

Furthermore, distinct peaks are evident at the beginning of the exercise, corresponding to sudden increases in HR, and similar peaks occur when the exercise ends. As a result, the current simulation does not adequately capture the physiological response at the onset of exercise, nor does it fully represent the post-exercise resting phase. This constitutes a significant limitation of the BioGears software in modeling non clinical scenarios involving physical activity.

Another limitation observed in the software is that, depending on the scenario, after a non- clinical simulation (e.g., the running example), if the simulation time is extended to 15 or 20 minutes after the exercise, the patient enters an irreversible state, in other words, the patient dies. At first, it was unclear why this occurred. However, after running the simulation for both a 22-year-old fit male soldier and a 44-year-old sleep-deprived male, each performing a 10-minute run followed by a resting period of 10 minutes or more, we observed that both patients died, as shown in the Figures Figure 2.11. It is notable that once both patients start running, the lactate production rate increases, and the hepatic gluconeogenesis rate also rises, which is expected. However, when the patients stop and enter the resting phase, both the lactate production rate and the hepatic gluconeogenesis rate continue to increase, which is not expected. This suggests that there is a limitation in the software for this type of scenario.

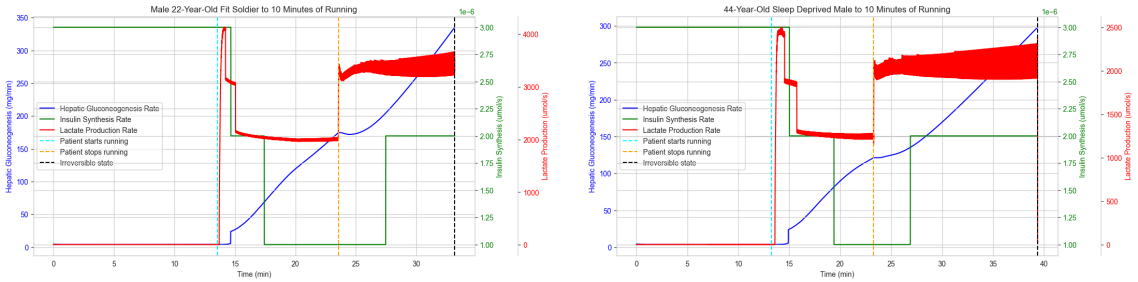

Figure 2.11: Hepatic Gluconeogenesis, Insulin Synthesis and Lactate Production Response

2.5.4. Patients Creation

In the BioGears software, it is possible to create a patient model by specifying a few basic physiological parameters, after which the software automatically calculates the remaining ones. It is possible to define attributes such as sex, age, weight, height, body fat fraction, diastolic arterial pressure, systolic arterial pressure, respiration rate, and heart rate. Based on these inputs, the software computes other important physiological parameters, resulting in the creation of a complete virtual patient. For this study, a patient was created as a 23-year-old fit male.

Once the virtual patient was generated, it was possible to run simulations of various physiological scenarios, including an asthma attack, a brain injury, and a running condition. The results of these simulations are presented in the Figure 2.12.

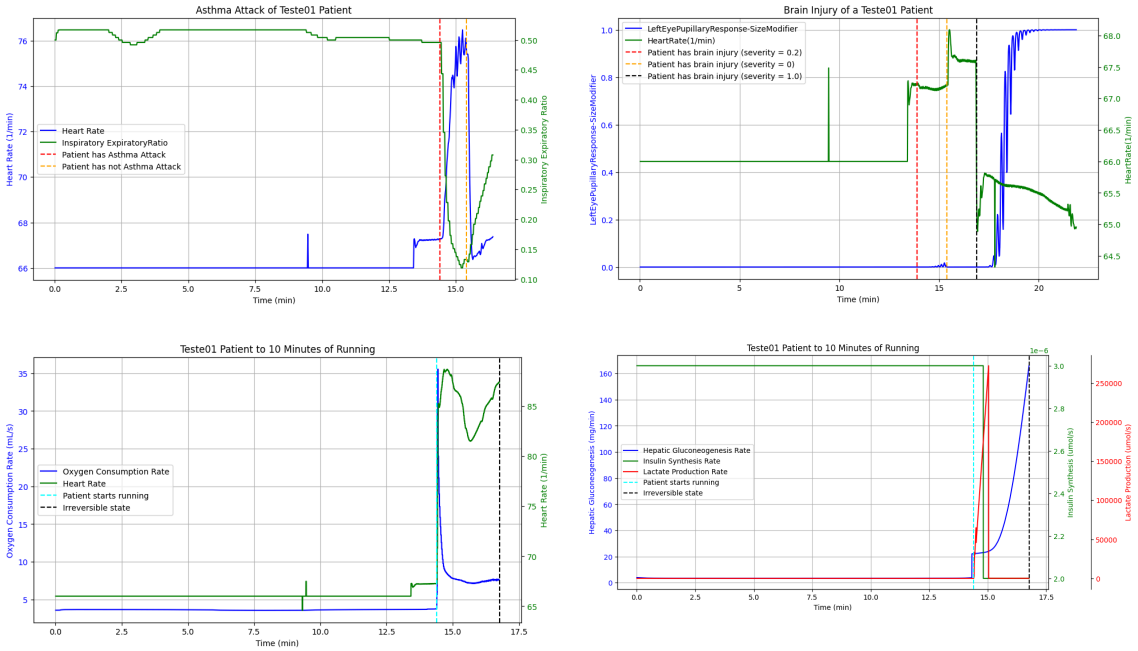

Figure 2.12: Asthma Attack, Brain Injury and Running Examples for Teste01 Patient

Therefore, based on the results, it is notable that the software once again produces coherent outputs for clinical scenarios. However, when applied to non clinical conditions, it tends to converge towards an “irreversible state”, that is, death.

3. Research Concept and Methodology

In this section, the methodology adopted for the development of this thesis is presented. The final methodology resulted from several brainstorming sessions with researchers from the BRAIN group at INESC TEC, together with an extensive review of the literature and additional investigations conducted to design the proposed framework. The structure of the dissertation is outlined, including the preliminary and the main stages of the research Roadmap.

3.1. Designing Personalized Human Digital Twin Concept

The use case structure is divided into two distinct subsystems:

- “Twin Approximation Based on Real Data”
- “Virtual Twin Generator”

as seen in Figure 3.1. The first one represents the foundational phase of the architecture, comprising a relational database and a UI. This database stores patient profiles, historical biological data, and contextual descriptions of patient activities. Through the UI, users can visualize this information and synthesize realistic datasets derived from the database records.

The latter subsystem, the “Virtual Twin Generator,” is responsible for the actual generation and initialization of the HDT. Because constructing an HDT inherently relies on empirical, real world data to optimize parameter accuracy, a sequential dependency was established between the two systems. Specifically, the “Twin Approximation Based on Real Data” subsystem must be deployed first to build a comprehensive data repository. This repository then serves as the reference baseline from which the “Virtual Twin Generator” extracts the necessary datasets to generate a highly precise HDT.

Thus, this pipeline enables the construction of a HDT driven by empirical, real world data. Furthermore, both subsystems are designed to incorporate a UI to streamline data visualization, as well as to support data streaming and export functionalities for subsequent analysis. In short, this architecture represents a comprehensive ecosystem that effectively integrates multiple complex systems and subsystems.

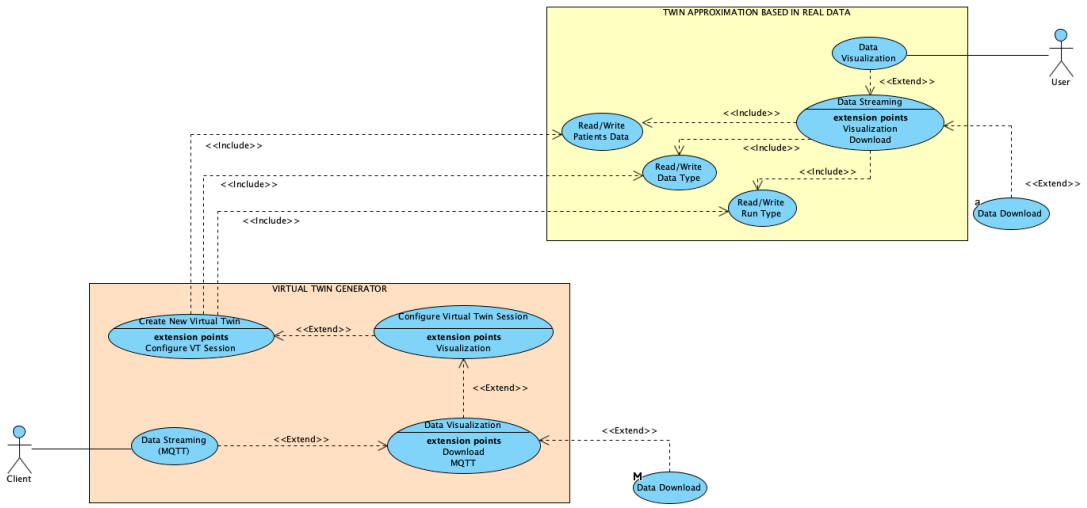

Figure 3.1: Use Case Diagram for the Concept Framework.

3.2. Exploring Concept Idealization - Can BioGears be Improved for Non Clinical Scenario ?

Considering the limitations identified in Section 2.5 and the concept presented in Section 3.1, the initial approach consisted of collecting real-world data to investigate whether it would be possible to develop a reliable simulator for certain physiological parameters, such as heart rate, based solely on real world measurements. To this end, a preliminary framework was conceived and designed to assess whether the collected data were sufficient to accurately reproduce these physiological variables.

The main purpose of these preliminary experiments was to validate our hypothesis that real-world collected data can be used to generate mathematical equations representing specific physiological systems of the human body. Once these equations are generated, it becomes possible to extract patient specific parameters, these parameters can then be incorporated into a dedicated software platform to create a personalized digital representation of the patient.

The initial and preliminary acquisition was conducted at INESCTEC, FEUP. The collected data comprised the HR and skin temperature of a specific individual. All measurements were obtained using a wearable device, specifically the VitalStickers. Two types of data collection were performed.

The first collection occurred at INESCTEC, where a Python script was developed to record the participant's ECG, skin temperature, and treadmill speed. Following data acquisition, a processing phase was implemented, as the raw signals frequently contained

noise. Consequently, signal cleaning and filtering were necessary to obtain usable data. Even after preprocessing, not all data were deemed suitable for further analysis.

For this initial data collection, the only participant was the author. ECG data were collected using the VitalSticker while running on a treadmill at different speeds. A total of $N = 6$ sessions were conducted, each following a different speed profile or exercise protocol. The main objective was to investigate whether it is possible to personalize a patient's physiological model using real world data.

The second data collection was conducted using the VitalSticker in conjunction with a mobile phone, and was performed during daily activities, such as attending the gym and engaging in physical exercise. The collected signals comprised ECG data and skin temperature measurements. In this phase, no direct speed or activity intensity data were available; however, the start and end times of each gym session or exercise period were recorded, allowing the identification of active intervals.

Due to the inherent noise associated with wearable devices, extensive signal processing was required. This processing aimed to mitigate noise artifacts and to assess the quality and reliability of the recorded data, ensuring that only valid and physiologically meaningful segments were retained for the final analysis.

Once the data have been collected, it was processed and subsequently made available for analysis. A model was then be developed to simulate the HR. Using the data in combination with relevant equations, it was possible to determine physiological parameters and construct a model representing the HR of the specific individual.

In this context, three phases or models was considered. The first corresponds to the resting phase, during which the HR is minimal and the individual is in a stable state. The second corresponds to the exercise phase (e.g., walking or running), during which the model will incorporate the previous HR values and project their evolution over time under physical activity. The third corresponds to the recovery phase, during which the HR gradually returns to the resting level.

Once the modeling equations have been completed, they will be integrated into software platforms such as BioGears, allowing the models to be implemented within the system. This integration will be applied exclusively for non clinical scenarios. Consequently, this approach will enable more accurate simulation and modeling of patient responses under non clinical conditions.

3.3. Use Case Definition and Data Collection

Considering the models presented in the previous sections, there are numerous cases in which human body simulations, both mathematical and data driven, can be applied. For all of these cases, it is important to note that the study can be further extended by considering external factors, such as temperature, humidity, wind, and other environmental conditions that may influence physiological responses. Incorporating these factors into the models allows for a more comprehensive analysis and a deeper understanding of human body behavior under varying conditions.

The use cases were defined as physical activity, control room operations and agriculture. They were selected based on the BRAIN team's previous research in these areas and the greater potential for further analysis and experimentation. Each use case will be described in detail, including a brief review of the state of the art.

3.3.1. Physical Activity

At first glance, the potential is evident in the context of physical activities, where it is possible to simulate a patient running, walking, or performing exercises that involve strength and other factors. For this use case, the benefits are noticeable, as a patient can be simulated under normal conditions, allowing researchers to study and analyze what happens when they start running or walking. In this field, Hlis et al. [44], conducted a study demonstrating that DTs enable the simulation of athletes and patients during running, cycling, swimming, and other forms of physical activity. The study also highlights the potential of DT technology not only for elite athletes, but also for rehabilitation and general fitness applications, allowing safe and data-driven simulations of human exertion under varied conditions.

In another study, Barricelli et al. [45], developed a team of HDT within the SmartFit framework to monitor athletes' behavior, including physical activity, sleep, nutrition, and mood over consecutive days. The proposed system is capable of predicting performance outcomes and generating actionable recommendations for improvement, effectively simulating how modifications in daily habits may influence future performance.

In this research area, the BRAIN team at INESC TEC has developed several studies. One particularly relevant work involved the analysis of military personnel during training exercises conducted in the context of the ARmy Technological EXperimentation (ARTEX) event. This study investigated how soldiers' physiological responses change under low,

moderate, and high intensity physical activity. Data acquisition included ECG signals, along with several other physiological and contextual measurements.

Building upon this previous work, the present thesis aims to advance the state of the art by simulating individual physiological states, with a particular focus on HR. The ultimate goal is to develop a HDT tailored to each individual, enabling the simulation and analysis of how the human body responds under specific operational conditions.

3.3.2. Control Room Operators

Another potential use case involves control room operators. In this scenario, there is minimal physical activity, as workers typically remain seated for extended periods while monitoring screens and simultaneously directing people or vehicles. For instance, air traffic controllers in airport towers or operators in electric grid control rooms spend long hours seated. The focus in this case is on external, non physical stimuli and their effects on the individual. Specifically, this allows the study of how stress and anxiety impact physiological responses and overall bodily function.

Recent research has explored the use of HDT frameworks to model and simulate human operator behavior in industrial control environments, demonstrating the feasibility of using cognitive and behavioral data to replicate operator responses in process control tasks. For instance, Adaptive Control of Though Rational (ACT-R) based HDT has been developed to emulate control room operator performance using eye-tracking and cognitive modeling techniques [46].

Additionally, emerging work in human-centric digital twins aims to integrate the monitoring of physical, mental, and emotional states in order to assess worker well-being and to identify stress or anxiety levels through multimodal sensor data and advanced analytics. In the same study, video recordings were also employed to analyze facial micro-expressions during work activities, as these involuntary expressions are closely linked to human emotional responses and may reveal underlying feelings even before the individual becomes consciously aware of them [47].

Rodrigues et al. [48], demonstrates that stress can have a significant influence on performance during the execution of specific tasks. In the analyzed scenarios, the results indicate that when participants are exposed to stress, both the average response time and the number of errors increase. Based on these findings, it is expected that, by leveraging the available dataset, stress levels can be estimated and quantified, as well as their impact on HR. This study was conducted by BRAIN team at InescTec.

Also, the BRAIN team is part of the AI4REALNET European project, that aims to have a human-centered AI approach, and have a system that adapts to the human mental state and reactivity increases the empathy between the human and the AI [49].

Therefore, considering control room operators, it is expected that they operate under a certain level of stress due to the high-intensity nature of their tasks. In this context, the prediction and analysis of an individual's physiological state are highly valuable. If stress levels can be quantified and reliably simulated, it becomes possible to estimate the recovery time required before the individual can safely go back to the operational activities, among several other potential applications.

3.3.3. Agriculture

Another potential application is in the field of agriculture. It is well known that agricultural workers often spend hours walking through fields to harvest specific fruits. For example, kiwi pickers must traverse the field extensively while collecting the fruits, frequently bending and walking for prolonged periods. This scenario provides an opportunity to study non clinical cases. In particular, for older workers, such simulations can serve as a valuable tool to assess and recommend safe durations for work activities.

Digital twin technology is being actively developed in the agricultural domain, however, HDT concepts have primarily emerged in other sectors and have not yet been applied to agricultural work tasks, only for other types of agents, including studies involving the monitoring and control of agricultural machinery.

As related work in this domain, the BRAIN team at INESC TEC conducted a study in agricultural environments, specifically in kiwi orchards and vineyards. The study focused on analyzing workers' activity metrics, including the distance walked, the duration of standing postures, the frequency of bending movements associated with harvesting tasks. Furthermore, the physiological components were collected, namely HR, skin temperature, and respiration rate. In addition, a map-matching analysis was performed to reconstruct and predict workers' trajectories, allowing the identification of areas with higher activity density. This analysis provided insights into spatial patterns of harvesting, which were subsequently related to fruit yield and quality in specific regions of the fields.

As future work, the objective is to develop a simulation framework for the visualization and analysis of workers' activities. Based on the available input data, the system is expected to estimate physiological outputs, such as HR, among other indicators. In this

context, environmental factors, including temperature and humidity, will play a crucial role in accurately simulating workers' behavior and predicting future physiological states.

3.4. Dissertation RoadMap

The dissertation roadmap is presented in Figure 3.2, where it is shown as a Gantt chart to facilitate visualization.

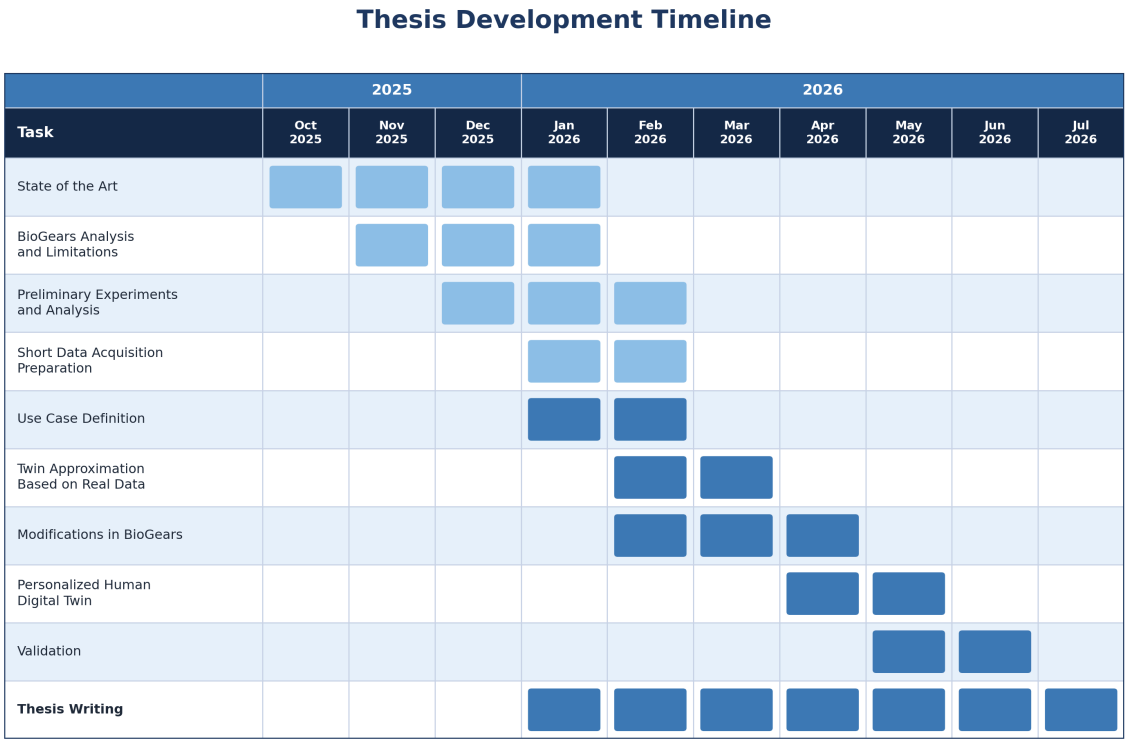

Figure 3.2: Gantt chart showing the planned research activities and their temporal distribution.

The thesis roadmap is structured into two main phases: a preliminary phase and a main research phase. The preliminary phase comprises the state of the art review and the execution of initial exploratory experiments, which were described in detail in the previous chapters. The state of the art review is conducted to assess the current level of development in the field, to contextualize the present work within existing research, and to identify relevant research gaps.

The BioGears analysis and its limitations are also presented in the state of the art. This is an important phase, as it is during this stage that the limitations of the software and the main identified gaps are addressed. In this phase, the software is studied and further analyzed in order, first, to understand its limitations and, second, to identify how these gaps can be addressed and how corresponding improvements can be implemented.

The preliminary experiments and analysis phase, which includes use case definition and data collection, is where the initial experiments are conducted. These experiments allow for a more precise refinement and delimitation of the specific application domain and the use cases to be addressed. Through this preliminary experimentation, it was possible to identify and validate the required inputs, models, and outputs necessary for the implementation of the defined use cases.

The main research phase is divided into five distinct tasks. The first task involves Twin Approximation based on real data. As will be discussed in detail in the following sections, this phase is fundamental, as it serves as the basis for the subsequent tasks. In this phase, a database is created in parallel with a software tool designed to read, interpolate, and visualize real world data for several patients.

The BioGears modifications constitute the core contribution of this thesis. After the limitations have been identified and the gaps defined, this stage focuses on investigating how the original software can be modified through small but impactful changes, aiming to produce more accurate and realistic outputs. Therefore, this phase is crucial for the subsequent work.

The subsequent task, Personalized Human Digital Twin, focuses on the integration of real and virtual data, which represents the central objective of this dissertation: the development of an advanced HDT framework for non clinical applications. Subsequently, in the validation phase, the proposed methodology is systematically evaluated in order to validate the previous research outcomes, identify limitations, and determine potential areas for improvement.

Finally, during the thesis writing phase, all the work developed throughout the previous phases is consolidated and formally documented in the dissertation; and in parallel, during the final phase of the thesis, an experimental prototype deployment was also carried out to create a virtual machine capable of hosting the final software. In In this way, the roadmap provides a comprehensive overview of the temporal and methodological flow of the research presented in this thesis.

4. Twin Approximation Based in Real Data

In this chapter, the creation of a database, a user interface, and the mathematical background for data interpolation are presented and explained. This constitutes the first concrete output of this thesis, in which the initial user interface is designed and implemented.

4.1. Data Organization

Before the software update, namely, the improvement of the VHT into HDT, we initially addressed the problem by constructing a large dataset comprising heterogeneous patients engaged in a variety of activities, including low intensity exercise, high intensity exercise, and baseline conditions.

Using this dataset, we developed a preliminary Twin approximation in the form of a User Interface (UI) and User Experience (UX) that enables the retrieval of specific patients. Once a patient is identified, their corresponding real-world data can be visualized. It is important to emphasize that, at this stage, no prediction techniques are applied; the primary objective is to establish a comprehensive dataset consisting exclusively of real data collected from real patients under non clinical conditions. The dataset is essential, as it provides the structure and overall organization required for its reuse in virtual modeling.

At this stage, a Use Case Diagram for the Twin approximation interface was developed, as illustrated in Figure 4.1. The figure presents the main use cases of the system, where the user is able to edit existing data, add new data, and create a new Virtual Twin.

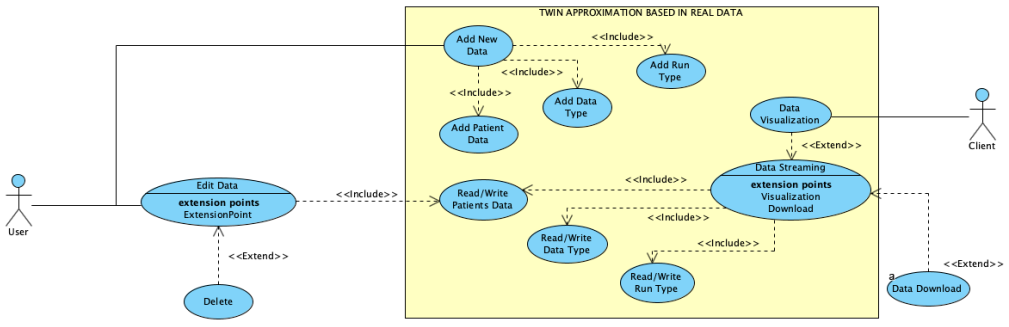

Figure 4.1: Use Case Diagram for the Twin Approximation Based in Real Data.

The system “Twin Approximation Based in real Data” is where the Twin approximation is performed. This process relies on the previously constructed database, which contains data from multiple patients. Based on user defined parameters and specific requests, the system is able to generate a Virtual Twin that approximates, as closely as possible, the characteristics of a real individual. If the parameters provided by the user yield no results, that is, if no patient matches the specified criteria, the UI notifies the user accordingly. In such cases, the system does not attempt to broaden the parameter space or perform an extended search; instead, the process flow is intentionally interrupted.

This design choice was made based on the primary objective of the Twin Approximation, which is to represent, as accurately as possible, the real data of a generated Virtual Twin. Consequently, relaxing or expanding the input parameters was deemed inappropriate, as it could compromise the fidelity of the representation.

4.2. Database Organization

The database was designed so that all data can be easily and intuitively modified, added, or removed. To achieve this, the proposed data structure is shown in Figure 4.2. The idea is simple and straightforward.

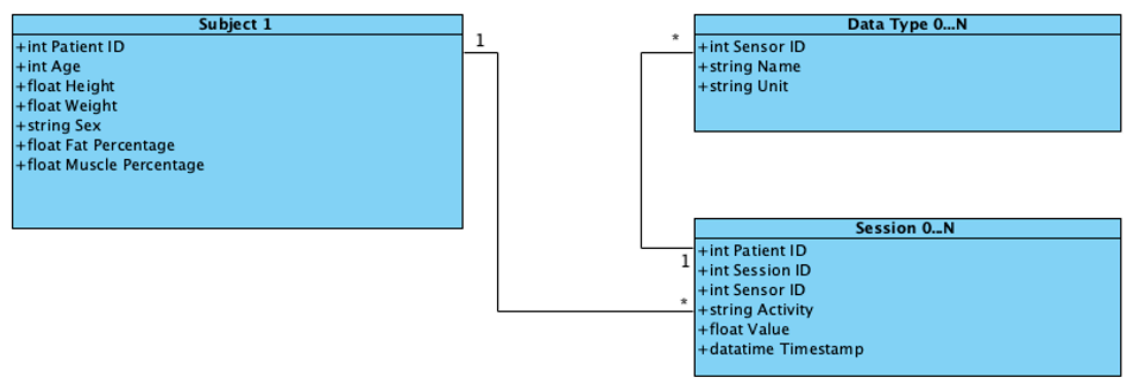

Figure 4.2: Database Architecture.

First, there is the Subject, which corresponds to the patient. A patient must be added with at least the following required fields: ID, sex, and age. All other parameters (height, weight, fat percentage, and muscle percentage) are optional and, if not provided, are automatically interpreted as ‘None’. It is also possible to add additional parameters if needed. The Data Type section defines the metadata associated with sensors. It is mandatory to specify the following fields: Sensor ID, name, and unit. For example;

```
Sensor ID 1;
Name: Heart Rate;
Unit: BPM
```

This structure allows the user to define an arbitrary number of data types, as well as rename or delete existing ones.

Finally, there is the Session. A session links the Patient ID, Sensor ID, Session ID, activity (string), a float value associated with the session, and a timestamp. With these three separated components, it is possible to add new patients or new data types without affecting existing data. When adding new data to the database, it is only necessary to provide:

- Patient ID
- Sensor ID
- Session ID
- Activity
- Value
- Timestamp

Therefore, this provides a simple and direct way to insert new data. Additionally, it enables modification and deletion of existing data.

For the creation of the database, DB Browser for SQLite [50] was used. This is a high quality, open source visual tool designed for users who need to create, browse, and edit SQLite and SQLCipher database files. Considering the data structure described in the previous paragraph, the data were first prepared and organized before being inserted into the database. A preprocessing stage was performed to convert the data into a suitable format prior to storage, as this approach simplifies data insertion and reduces the need for additional processing when retrieving the data in the future.

This database was constructed by aggregating datasets from INESC TEC and PhysioNet [51], more specifically, the dataset “Treadmill Maximal Exercise Tests from the Exercise Physiology and Human Performance Lab of the University of Malaga” [52]. The Figure 4.3 illustrates the final structure of the database after all data had been preprocessed and successfully inserted. In total, the database contains data from $N = 890$ patients and $N = 4351745$ recorded sessions.

Patients	DataSet	Description	Variables
N = 08	Vital Signs Data Collection	Subjects performing various activities, such as jumping and push-ups.	Heart Rate (bpm), Skin Temperature (ºC), Oximeter (%) Respiration Rate (breaths/min), MET
N = 11	TSST DataSet	Workers under stress conditions.	Heart Rate (bpm)
N = 838	Treadmill Malaga	Different individuals performing treadmill exercise at varying speeds.	Heart Rate (bpm), Respiration Rate (breaths/min), VO2 (ml/kg/min), VCO2 (ml/kg/min)
N = 33	CFT DataSet	Control and CFT subjects under the MIST protocol.	Heart Rate (bpm), MET

Table 4.1: Database Description.

As can be seen in Table 4.1, four different datasets were used, each covering one of the considered use cases. The first dataset includes several activities, such as push-ups, jumping, and others. The second is entirely dedicated to stress responses, while the third focuses exclusively on treadmill exercises and, therefore, physical activity. The last dataset corresponds to the Cold Face Test protocol.

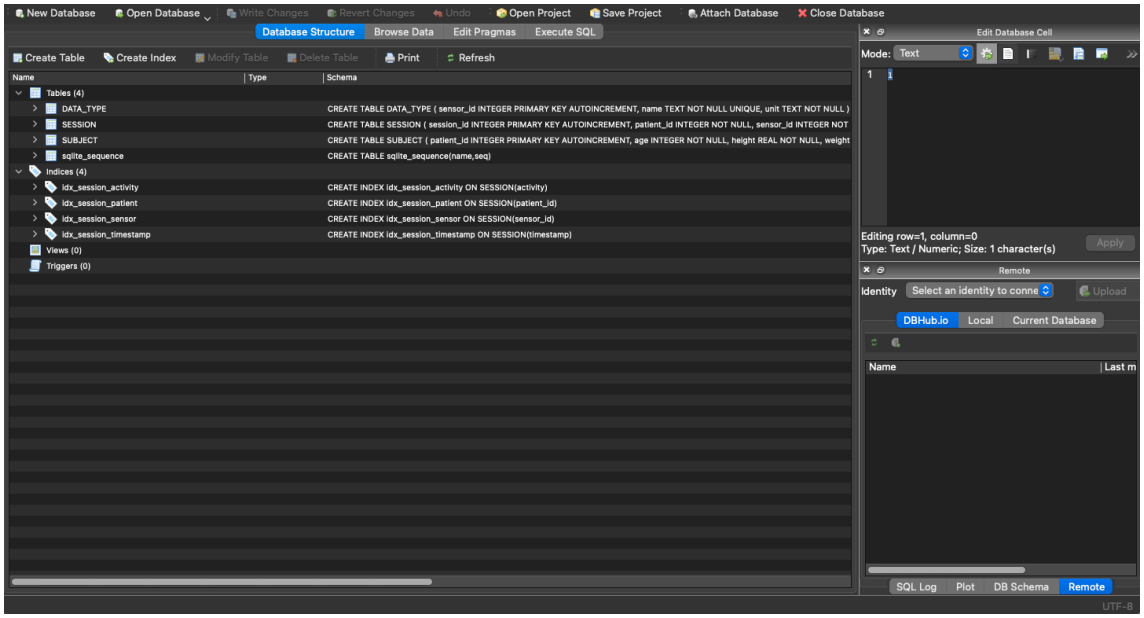

Figure 4.3: Database Example.

4.3. Mathematical Reconstruction Approach

Considering that the system currently operates exclusively with real data and does not yet incorporate any simulation capabilities, only the data stored in the database described in Section 4.2 are available. Consequently, the user interface must account for situations in which the user wishes to visualize data over a time period that exceeds the duration covered by the available records in the database. The mathematical approach that was used was the following.

Let $s = (s_1, ..., s_m)$ denote the recorded signal for a given activity α (e.g. heart rate during "high exercise"), sampled at timestamps $t_1 < ... < t_m$ spanning a real recorded duration $T_{\text{real}} = t_m - t_1$. Two operations are then combined; a temporal reparametrisation of the recorded curve, and a statistical synthesis around a population level anchor.

$$\tau_i = \frac{t_i - t_1}{T_{\text{real}}} \in [0, 1] \tag{4.1}$$

For a requested duration T_{req} (possibly $T_{\text{req}} > T_{\text{real}}$), a new grid of $N = \max\left(10, \left\lceil T_{\text{req}} \cdot f_s \right\rceil\right)$ points is built :

$$\tau'_j = \frac{j}{N-1}, \quad j = 0, ..., N-1 \tag{4.2}$$

and the curve is reconstructed by piecewise linear interpolation in the normalized time domain:

$$g(\tau) = s_i + (s_{i+1} - s_i) \cdot \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}, \quad \tau_i \le \tau \le \tau_{i+1} \tag{4.3}$$

Because $\tau' \in [0, 1]$ always lies inside the domain of g, this step is an interpolation, g is never evaluated outside $[\tau_1, \tau_m]$. What is in fact extrapolated is the protocol duration, not the signal value in physical time,

$$\hat{s}(t) = g\left(\frac{t}{T_{\text{req}}}\right), \quad t \in \left[0, T_{\text{req}}\right] \tag{4.4}$$

while the original recording corresponds to $s_{\text{real}(t)} = g\left(\frac{t}{T_{\text{real}}}\right)$. The mapping between the two is an affine time dilation

$$t \mapsto \lambda t, \quad \lambda = \frac{T_{\text{req}}}{T_{\text{real}}} \tag{4.5}$$

i.e. the algorithm assumes temporal self-similarity, it keeps the morphology of the observed curve and stretches or compresses it by λ to fit the requested duration. When K patients contribute data for the same activity, g is the average of their individually normalised shapes,

$$\overline{g}(\tau) = \frac{1}{K}\sum_{k=1}^{K} g_k(\tau) \tag{4.6}$$

The shape is z-normalized,

$$\hat{z}_j = \frac{\overline{g}\left(\tau_j'\right) - \mathrm{mean}(\overline{g})}{\mathrm{std}(\overline{g}) + \varepsilon} \tag{4.7}$$

and combined with a population-level anchor μ and spread σ:

$$\hat{s}_j = \max\left(0, \mu + 0.4\sigma\hat{z}_j + \eta_j\right), \quad \eta_j \sim N\left(0, (0.1\sigma)^2\right) \tag{4.8}$$

Transitions between consecutive activities follow a first order kinetic model of the same family used to fit the HR rise time constant:

$$\frac{d\,\mathrm{HR}}{dt} = k(\mathrm{HR}_\infty - \mathrm{HR}) \Longrightarrow \mathrm{HR}(t) = \mathrm{HR}_0 + (\mathrm{HR}_\infty - \mathrm{HR}_0)\left(1 - e^{-kt}\right) \tag{4.9}$$

with $\tau = 1/k$ obtained by bounded nonlinear least square restricted to the first ≈ 90 s of the recording.

Equation (4.5) shows that the time dilation λ rescales all apparent kinetic constants of the synthetised trace, a transient with real time constant τ appears in the synthetised signal with $\tau_{\mathrm{apparent}} = \lambda\tau$. This is the main limitation of a purely data driven synthesis strategy; it preserves the statistical morphology of the recorded signal but not its underlying physiological kinetics once the requested duration departs from the recorded one.

4.4. User Interface

In the Twin Approximation, a user interface was also developed with the objective of facilitating interaction in a more intuitive manner. The interface can be described as follows; first, the user is prompted to enter the human physiological parameters, namely height, weight, sex, and age, as can be seen in Figure 4.4. Based on this input, the system retrieves and displays a set of patients with similar characteristics.

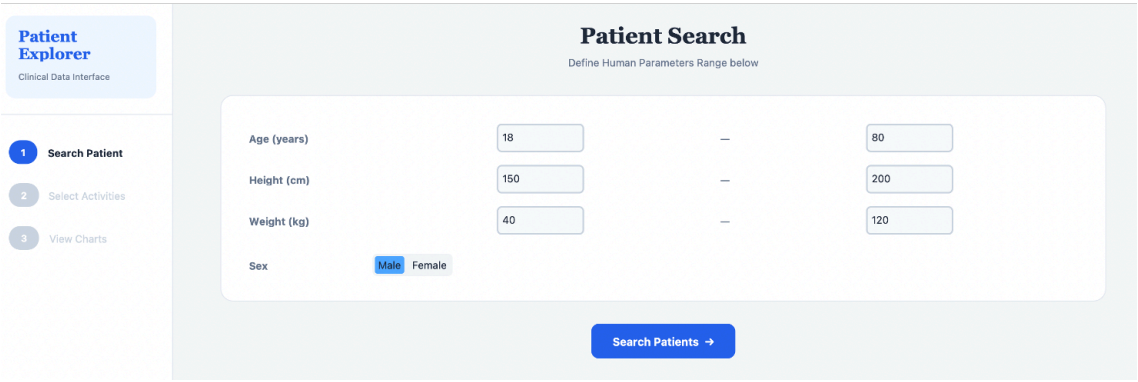

Figure 4.4: Selection of Parameters of interest using the Twin Approximation Interface.

In addition to the patient selection, the interface presents the activities associated with each patient, along with their corresponding duration (e.g., “High Exercise”, 365 seconds). The user can then select the desired activities and define the time intervals for analysis, as verified in Figure 4.5.

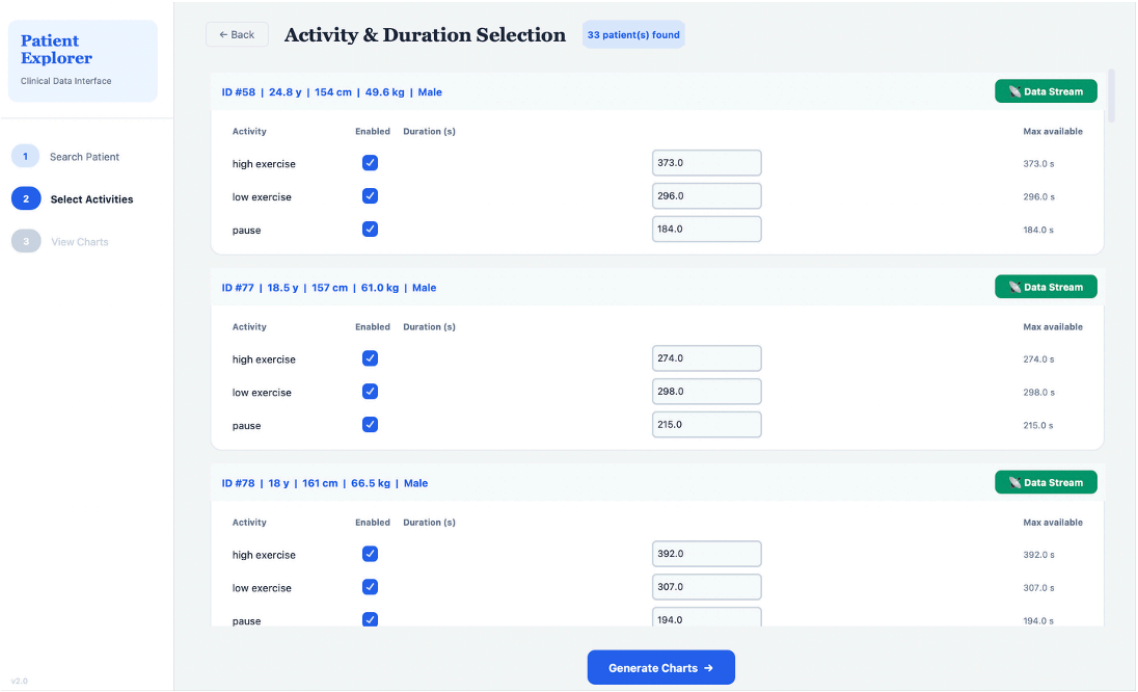

Figure 4.5: Configuration of Patient Activities and Time.

It is important to note that the activity duration in the database may be shorter than the interval requested by the user. When the requested duration is equal to or shorter than the available data, the system uses the raw recorded data directly. However, when the requested duration exceeds the available range, a prediction step is performed, in which the missing portion of the signal is estimated using a time integration based approach to enable full curve reconstruction and plotting, as explained in details in Section 4.3. With

this, the user can defined the time it wants, being less or more than the real and available time in the database.

Once the user has chosen the activities and their corresponding durations, two display modes are available. The first option is a real-time data stream, in which a single patient is visualized at a time, allowing continuous monitoring of the physiological signals. Also, the user can stop, change the speed of simulation and save the specific data. The second option is a full graph view, in which all selected data are displayed simultaneously as a complete time-series representation, as can be seen in Figure 4.6.

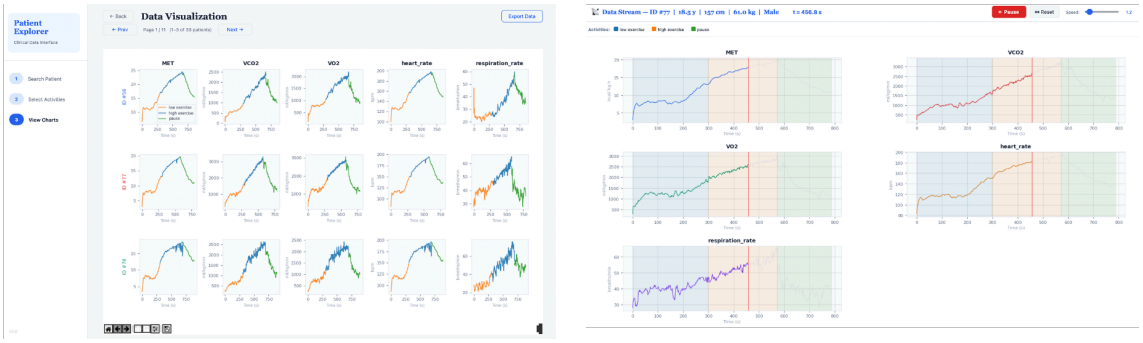

Figure 4.6: Two Displays available. All patients simultaneously or inspect real time data for one patient.

Finally, the user can download the data in either Excel or `.cfg` format, as seen in Figure 4.7. In the Excel file, all patient data are exported, allowing the complete reconstruction and regeneration of all generated graphs. The data is separated by excel sheet using the variables as the name of the sheet. In contrast, the `.cfg` file contains only the parameters required by the BioGears engine, enabling the generation of more personalized configurations and simulation settings, as will be describe in more detail in the further chapters.

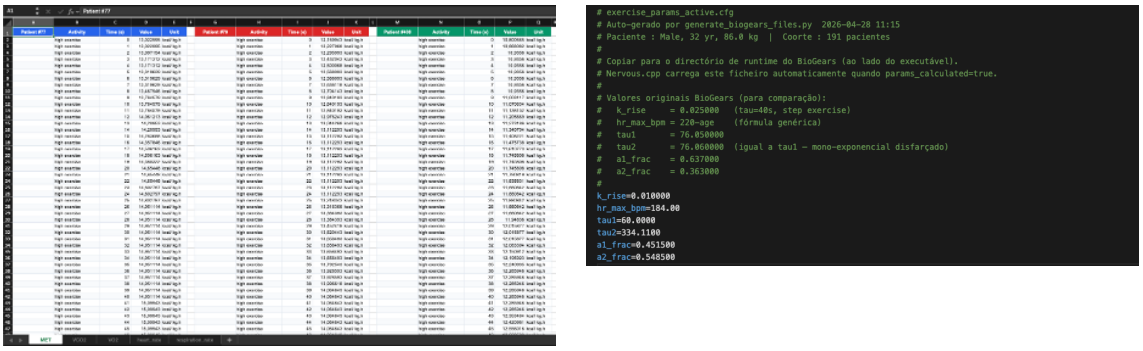

Figure 4.7: Two export format available; Excel of .cfg

Therefore, in short, The Twin Approximation is implemented as a user interface that allows the specification of desired avatar characteristics, such as sex, weight, and height.

Based on these inputs, the system performs a search within a pre-compiled database. Given the parameters selected by the user, the software identifies and returns the most similar patients within the database.

Once an appropriate avatar has been selected, the user may define both the type and duration of the activity, restricted to those already available in the database. It is important to note that, at this stage, no new data are generated; instead, the system exclusively reuses existing data. Subsequently, the user can either visualize the data from all selected patients through graphical representations or monitor the data of a single patient in real time.

Finally, the system provides two options for data export: the results may be saved either as an Excel file or in a .cfg format for future use. In the case where the user employs a `.cfg` file, it can be uploaded into a modified version of BioGears.

5. Researching new Models to Enhance Biogears Physiological Digital Human

In this chapter, all the improvements made to the BioGears Engine are explained, along with the improvements for these alterations and the expected effects of the implemented changes. In addition, preliminary validations are performed to verify whether the alterations are correct and function as intended.

All subsequent improvements and modifications were implemented in the BioGears Engine software [10]. The original software is referred to as Version 1 ($V1$), as it represents the unmodified baseline version. Consequently, all later developments are designated using the notation Vn, where n denotes the corresponding version number, indicating a modified or enhanced version derived from the original $V1$.

5.1. Heart Rate Calculation Modification (V2)

The first improvement based on real human data, Version 2 ($V2$), was introduced following an observation of the HR response in the original software, where the HR exhibited a binary (square wave) behavior. Consequently, the first modification consisted of implementing physiologically consistent HR equations for the subjects.

The reason why HR is represented as a binary signal in the original software is straightforward. When a patient is engaged in an activity (e.g., running, cycling, or generic exercise), the exercise state in `Nervous.cpp` is defined in binary form. In this context, `HasExercise()` is a Boolean variable, and the exercise intensity remains constant throughout the entire simulation. As a result, there is no progressive warm up phase, and the exercise is effectively modeled as a Heaviside step function, $H(x)$. Consequently, the resulting HR response inherits this discontinuity, leading to a square wave like behavior.

However, this represents a fundamental limitation of the `SEExercise` module. In contrast, for cases such as asthma (Clinical case), the corresponding formulation already deviates from a pure step function representation and exhibits a more continuous behavior.

Therefore, the following components were added to the software or developed externally to improve interoperability between the database and BioGears:

1. Equations describing heart rate increase and decay
2. A Python script enabling communication between the database and BioGears

3. Modifications to the BioGears software to read external data and integrate the proposed heart rate equations

The alteration was applied across three different files: `Nervous.h`, `Nervous.cpp`, and `Tissue.cpp`. Additionally, the file `exercise_params.h` was introduced, which is generated by an external Python script, `generate_biogears_files.py`, responsible for extracting patient specific constants. The header file `exercise_params.h` is generated by `generate_biogears_files.py`, based on the output of the function `extract_patient_params()` defined in `prediction_V6.py`. The script `prediction_V6.py` is an external Python module that utilizes the App Twin database to derive patient-specific parameters. All Python scripts were developed to facilitate communication between BioGears and real-world data.

The new block in `Nervous.cpp` introduces a name space that attempts to load the file `exercise_params_active.cfg` from the directory. If the file is not found, the BioGears default values are used.

For the rise in HR, the standard calculation was replaced by a first-order discrete filter:

$$HR_{\text{target}} = HR_{\text{rest}} + (HR_{\text{max}} - HR_{\text{rest}}) \times I \tag{5.1}$$

$$HRscale_{\text{target}} = \frac{HR_{\text{target}}}{HR_{\text{rest}}} \tag{5.2}$$

and;

$$\tau = \frac{1}{k_{\text{rise}}}, \alpha = \frac{\Delta t}{\tau} \tag{5.3}$$

where I represents the exercise intensity provided by `SEExercise`.

For the recovery phase, the HR formulation was also modified to reflect a more physiologically accurate model. When `HasExercise()` becomes ‘False’ after the first exercise event, the system transitions into recovery mode:

$$HR(t) = HR_{\text{rest}} + A_1 \times e^{\frac{-t}{\tau_1}} + A_2 \times e^{\frac{-t}{\tau_2}} \tag{5.4}$$

With the `.cfg` file, the response is decomposed into a fast component (parasympathetic), $T_1 \approx 60 - 80s$, and a slow component (sympathetic + thermoregulation), $T_2 \approx 260 - 540s$ [53, 54].

The resting heart rate, HR_{rest}, is estimated as the median of the trailing 20% of each Baseline segment. When the Baseline label is missing, the $5th$ percentile of the full HR

record is used as a physiologically robust proxy for the resting minimum. A sanity filter restricts accepted values to $30 < HR_{\text{rest}} < 100$ bpm.

The maximum heart rate, HR_{max}, is computed per subject as the $95th$ percentile of the high exercise HR series (rather than the absolute maximum), to suppress motion artifacts and telemetry spikes. The cohort estimate is the median over subjects, defaulting to the Tanaka et. al, [43] formula if no high exercise data is available, defined by Equation (2.1).

With this initial improvement, a preliminary validation was conducted to verify correct functionality. Specifically, the predefined patient `Male_22_Fit_Soldier` was simulated running at a speed of $3.4\frac{m}{s} \approx 12.24\frac{Km}{h}$, followed by a recovery phase.

However, BioGears generated two warnings: “MaxWorkRate and BasalMetabolic values are lower than the computed ones” and “Patient enters an irreversible state because speed is too high.”

Consequently, the next step was to investigate the origin of these issues and determine the admissible limits for the running speed. The model imposes internal physiological constraints that cannot be exceeded without leading to system instability or collapse. In this case, the patient model entered a hypoglycemic state approximately $24s$ after initiating the run at $3.4\frac{m}{s}$. This result is physiologically implausible, as an individual with normal glycogen reserves can typically sustain moderate to high intensity exercise for approximately 60 to 90 min before experiencing severe depletion [55].

Speed (m/s)	Speed (km/h) and Equivalence	BioGears
0.4	1.4 (Low Walk)	Stable
1.0 - 1.5	3.6 - 5.4 (Moderate Walk)	Stable
2.0 - 2.5	7.2 - 9 (Moderate Run)	Stable
3.0	10.8 (Intense Run)	Limit
$\geq$ 3.1	11.16 (High Intense Run)	Collapse

Table 5.1: Comparison between speeds in the BioGears exercise function and its reliable.

Therefore, a range of velocities was systematically evaluated in order to better characterize the limiting speed of the model. For both versions of the software, a consistent protocol was adopted: each simulation began with a baseline phase of two minutes, followed by an exercise phase at the selected velocity lasting 320s (5m, 20s). Subsequently, the patient ceased activity and entered a recovery phase of 300s (5m). The results are illustrated in Figure 5.1 and Figure 5.2.

Additionally, a table summarizing the velocity ranges and their corresponding limits was generated (see Table 5.1). From both tha e graphical analysis and the tabulated results, it is evident that for velocities exceeding $1.5\frac{m}{s}$ $(5.4k\frac{m}{h})$, the software begins to lose stability.

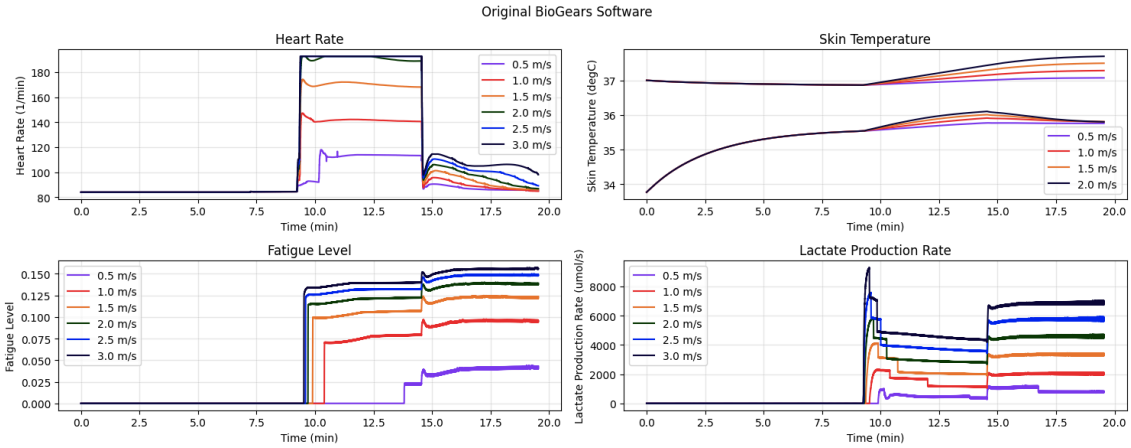

Figure 5.1: Original BioGears software output: heart rate (bpm), skin and core temperature (ºC), fatigue level, and lactate production rate (mmol/s) for different speeds.

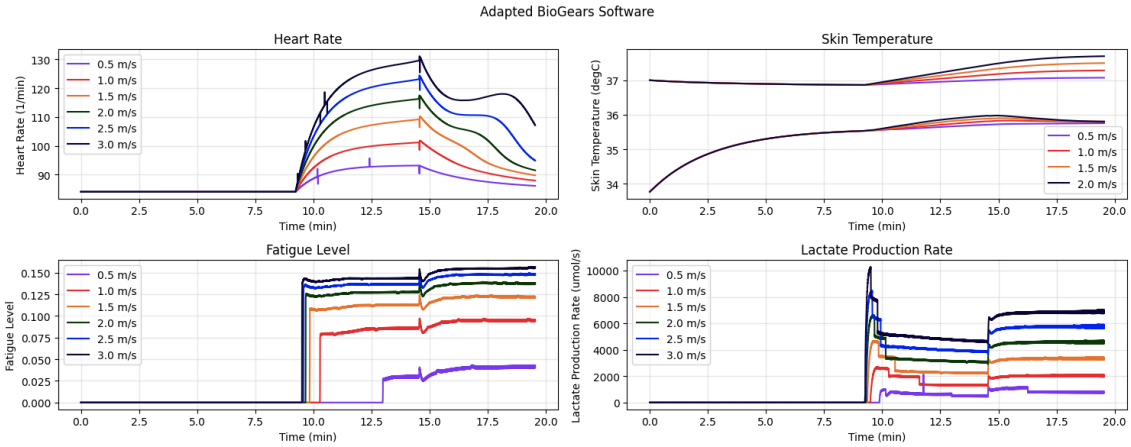

Figure 5.2: Adapted BioGears Software $V2$ output: heart rate (bpm), skin and core temperature (ºC), fatigue level, and lactate production rate (mmol/s) for different speeds.

Considering Figure 5.1 and Figure 5.2, it is possible to observe a significant improvement between the original and adapted implementations. The square wave behavior observed in the original model is no longer present in the adapted version. This improvement is a direct consequence of the implementation of Equation (5.1), Equation (5.2), and Equation (5.4). By modifying the method used to calculate HR within the software, it was possible to replace the abrupt square wave response with a smoother behavior that more closely resembles physiological HR dynamics observed in real subjects.

5.2. Phosphocreatine Circuit Analogy (V3)

Beside the Heart Rate improvement, it was verified that during exercise, the Lactate Production Rate values, as shown in Figure 5.2, become unrealistically high, which does not reflect physiological reality, thus the Version 3 ($V3$) addresses this systematic issue observed in $V2$.

The original model and the $V2$ does not account for Phosphocreatine (PCr), relying solely on anaerobic glycolysis at the onset of exercise. This leads to rapid depletion of intracellular glucose and may result in hypoglycemia [10]. In the original BioGears implementation, when exercise begins, aerobic metabolism is unable to respond immediately to the increased muscular energy demand. This delay leads to the activation of anaerobic glycolysis, resulting in a transient peak in lactate concentration and a concomitant depletion of glucose levels.

To address this limitation, PCr was introduced as a high energy molecule that functions as an energy reservoir [56]. Its role can be analogously compared to a capacitor in an electrical circuit.

The circuit illustrated in Figure 5.3 represents the conceptual implementation of PCr. When the energy demand rises rapidly, the capacitor (PCr) discharges instantaneously, preventing a drop in the “voltage” (i.e., ATP concentration). Subsequently, aerobic metabolism replenishes the capacitor by restoring PCr levels.

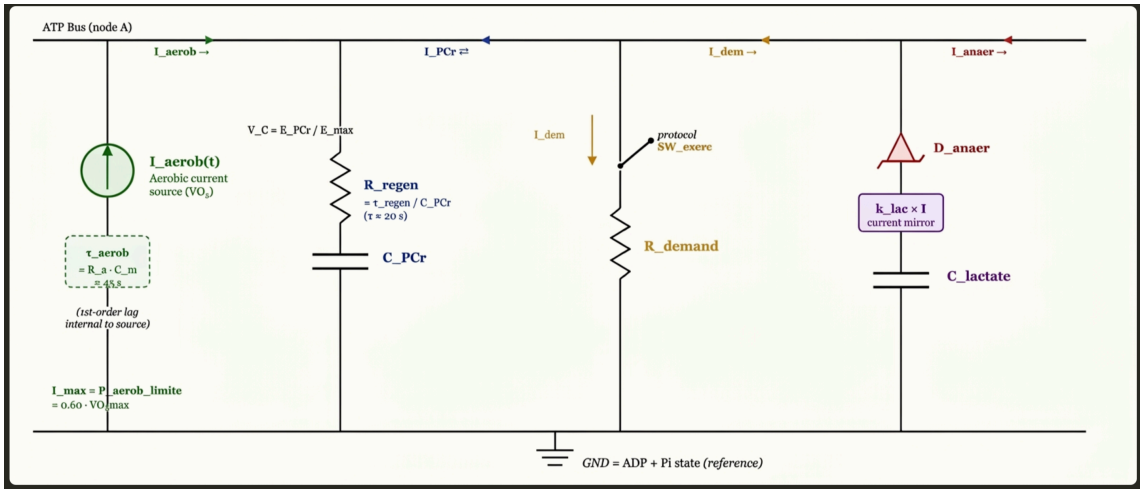

Figure 5.3: PCr Electric Circuit.

The equations that model the circuit from Figure 5.3 are:

$$I_{\text{aerob}}\,(t) + I_{\text{CPCr}}\,(t) + I_{\text{anaerobic}}\,(t) = I_{\text{demand}}\,(t) \tag{5.5}$$

$$I_{\text{PCr}} = C_{\text{PCr}} \times \frac{dV_C}{dt} \tag{5.6}$$

$$I_{\text{anaerobic}} = \max(0, I_{\text{demand}} - I_{\text{aerobic}} - I_{\text{PCrmax}} \times H(V_C - V_{\text{th}})) \tag{5.7}$$

Where, $H()$ is the Heaviside function and τ_{regen} is the regeneration time.

The ATP–PCr system provides ATP through hydrolysis during the initial 15–30 seconds following the onset of exercise, without requiring O_2 and without producing CO_2 or lactate. The recovery process after exercise follows a first-order filter with $\tau_{\text{regen}} \approx 20s$:

$$PCr = PCr + \left(\frac{\Delta t}{\tau_{\text{regen}}}\right) \times (PCr_{\max} - PCr), PCr \leq PCr_{\max} \tag{5.8}$$

During exercise, PCr supplements the energetic deficit before the activation of anaerobic glycolysis:

```
if (tissueNeededEnergy_kcal > 0.0 && s_PCr_J > 0.0) {
  double deficit_J = tissueNeededEnergy_kcal * J_per_kcal;
  double energyFromPCr_J = std::min(deficit_J, s_PCr_J);
  s_PCr_J -= energyFromPCr_J;
  tissueNeededEnergy_kcal -= energyFromPCr_J / J_per_kcal;
}
```

It is important to note that, during this phase, no O_2 is consumed and no CO_2 or lactate is produced. Considering all these modifications, it is expected that:

- HR during exercise remains close to HR_{target}
- The occurrence of hypoglycemia is prevented
- The HR recovery curve exhibits a bi-exponential profile; Equation (5.4)

All modifications from $V2$, the personalized HR control system (first-order rise filter and bi-exponential recovery) in `Nervous.cpp` are preserved unchanged in $V3$. Following the implementation of these modifications, the resulting behavior is illustrated in Figure 5.4.

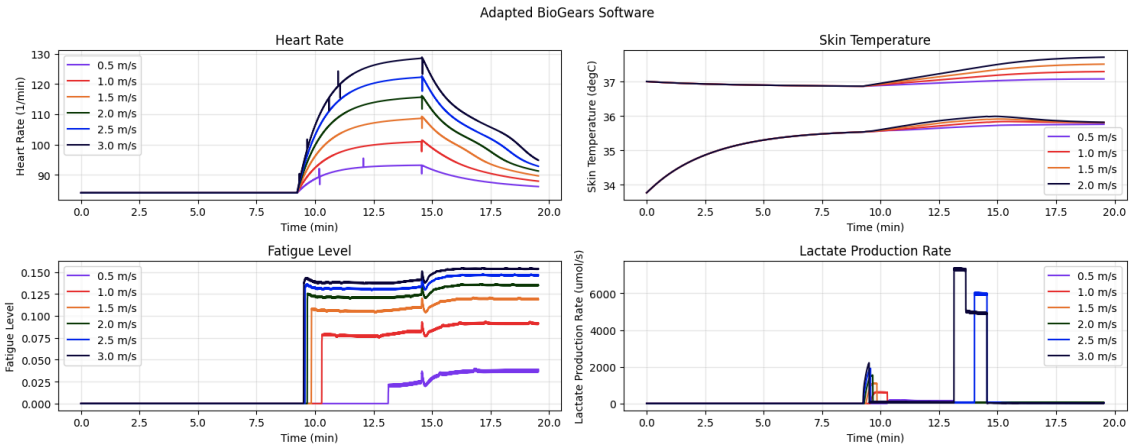

Figure 5.4: Adapted BioGears Software $V3$ Output for Different Speeds.

The differences between Figure 5.2 and Figure 5.4 are subtle; however, they indicate that the modifications are moving the model in the desired direction. A more detailed inspection reveals that the Lactate Production Rate exhibits a substantial reduction in the second graph. This behavior results from the introduction of PCr system. Because the PCr system acts as a short term energy buffer, the onset of exercise is initially supported by PCr utilization rather than by an immediate increase in lactate production. Consequently, lactate accumulation is delayed, which significantly alters the dynamics of the remaining physiological subsystems.

Additionally, the absence of differences between the fatigue levels observed in the graphs does not indicate an inconsistency in the model, but rather a direct consequence of how each variable is defined within the engine. The fatigue level is determined by a thermodynamic balance between ATP demand and the total energy supply from all metabolic pathways. In contrast, the lactate production rate quantifies the production of lactate resulting from anaerobic glucose metabolism.

The software is composed of several cascaded modules, where the output of one module influences the behavior of the others. The impact of the PCr implementation is particularly evident in the HR response. In Figure 5.2, the HR curve displays an abnormal pattern after approximately 16 minutes, whereas this behavior is no longer observed in Figure 5.4.

Finally, the patient no longer enters an irreversible state when subjected to exercise, as observed in Section 2.5.3, indicating a substantial improvement in the overall stability of the simulation.

5.3. Intensity Calculations Modifications (V4)

During physical activities, the engine uses the patient's maximum exercise power ($W_{\max}$), which represents the maximum mechanical power that the patient can sustain during exercise. By default, the engine assumes a value of 1200 W, which is unrealistic for most individuals and leads to excessive fatigue, eventually resulting in death. To address this limitation, Version 4 ($V4$) revised the definition of exercise intensity to represent a fraction of the oxygen uptake reserve ($\%VO_2R$). Additionally, a compatibility layer was introduced to preserve the calibrations established in Version 3 (V_3).

First, $V4$ adopts $\%VO_2R$ instead of $\%VO_2max$:

$$VO_2target = VO_2rest + I \times (VO_2max - VO_2rest) \tag{5.9}$$

With this formulation, it becomes possible to construct a conversion table between $\%VO_2max$ and $\%VO_2R$, as shown in Table 5.2.

%VO2max	%VO2R
40%	33.6%
60%	55.8%
80%	77.9%
100%	100%

Table 5.2: Comparison between speeds in the BioGears exercise function and their equivalents.

The implementation in Energy.cpp was carried out in several steps.

- First, the runtime and helper structures were introduced.

```
struct VO2ParamsRT {
  bool params_calculated = false;
  double vo2_rest_L_per_min = -1.0;
  double vo2_max_L_per_min = -1.0;
  double efficiency = 0.21;
  void TryLoad(const std::string& cfg_path = "exercise_vo2_params.cfg");
};
VO2ParamsRT g_vo2; bool g_vo2_loaded = false;
// I_VO2R -> ΔMetabolicRate (W) via Weir 1949
inline double VO2R_to_extraMetabolic_W(double I, double rest, double max) {
  if (I < 0) I = 0; if (I > 1) I = 1;
  return I * (max - rest) * 351.6667;
}
// Inverse: When the user passes desiredWorkRate in Watts
```

```
inline double extraMetabolic_W_to_VO2R(double W, double rest, double max);
inline double legacy_Wmax_frac_to_vo2r(double Il, double rest, double max);
```

- Second, the `Energy::Exercise` method in the `SEExercise::Generic` mode was modified to interpret the intensity directly as $\%VO_2R$.

```
if (genericEx.Intensity.IsValid()) {
  exerciseIntensity = genericEx.Intensity.GetValue(); // [V4] %VO2R direct
} else if (genericEx.DesiredWorkRate.IsValid()) {
  // [V4] DesiredWorkRate is now EXTRA METABOLIC RATE
  DesiredWorkRate_W = genericEx.DesiredWorkRate.GetValue(PowerUnit::W);
  exerciseIntensity = extraMetabolic_W_to_VO2R(DesiredWorkRate_W, vo2_rest_Lpm,
vo2_max_Lpm);
  if (exerciseIntensity >= 1.0) {
    exerciseIntensity = 1.0;
    Warning("Desired work rate corresponds to >100% VO2R. Clipping to 100% VO2R
(VO2max).");
}
  genericEx.DesiredWorkRate.Invalidate();
}
```

- Additionally, modifications were required in `Nervous.cpp`, specifically to redefine the denominator semantics. Instead of using the arbitrary value $W_{\max} = 1200W$, the formulation was updated to employ a physiologically consistent definition based on VO_2R, while preserving the neural response dynamics.

```
inline double vo2r_to_legacy_Wmax_frac(double I_vo2r,
double vo2_rest_Lpm,
double vo2_max_Lpm) {
  if (I_vo2r < 0.0) I_vo2r = 0.0;
  if (I_vo2r > 1.0) I_vo2r = 1.0;
  const double deltaVO2_Lpm = I_vo2r * (vo2_max_Lpm - vo2_rest_Lpm);
  const double extraMetabolic_W = deltaVO2_Lpm * 351.6667;
  return extraMetabolic_W / 1200.0;
}
```

The Central Signal Process (sympathetic/vagal) was also updated to ensure consistency with the new intensity definition. Additionally, modifications were applied to the elastance feed forward block. Therefore, with the changes introduced in $V4$, it is expected that the intensity values will be easier to compute and integrate into the software.

5.4. Model Improvement for High Stress Moments (V5)

During the brainstorming of use cases and the definition of the final set of use cases, one of the scenarios considered was a control room operator under stress. For this purpose, the selected protocol was the Cold Face Test (CFT). However, after reviewing the BioGears software, it was observed that the CFT is not implemented. Consequently, Version 5 (V5) was developed to address this limitation.

Version 5 (V5) of the software introduces a new function, the CFT, specifically through the implementation of `SEColdFaceTest` in the BioGears engine. In addition, V5 does not introduce major structural or logical modifications; instead, it mainly consists of a recalibration of the previous versions, aiming to optimize the software performance as much as possible.

The CFT function was developed because no equivalent feature exists in the BioGears engine. The only related implementation available was designed for full body submersion scenarios, in which the patient is completely submerged in water. However, for the CFT, only the patient's face should be exposed to cold water, where cold stimulus applied to the facial region induces bradycardia and peripheral vasoconstriction, known as the diving reflex [57].

Therefore, the `SEColdFaceTest` was developed by proposing a direct modulation of the `m_VagalSignal_Hz` and `m_SympatheticPeripheralSignal_Hz` signals within `Nervous.cpp`. The implementation was designed to preserve the decoupling between the vagal and baroreflex responses, which is an essential condition for maintaining the physiological fidelity of the simulation.

As previously discussed, and considering [48], the physiological quantification of operational stress plays a central role in HDTs for non clinical applications. In this context, the CFT represents a simple yet effective test that provides a quantitative assessment of the vagal response. Consequently, its implementation constitutes an important contribution to the software, since the original version does not provide any comparable physiological test or evaluation mechanism.

The theoretical background of the CFT function was first investigated. In a simulated diving response, cardiovagal activation induces bradycardia, while sympathetic activation contributes to the maintenance or elevation of blood pressure. Within this physiological framework, the CFT can be interpreted as a simplified version of the diving response, producing similar cardiovascular adjustments, albeit with reduced magnitude.

Considering [58, 6], the expected physiological response in the `SEColdFaceTest` implementation includes a peak bradycardic effect occurring approximately between $[30, 60]$ seconds, with a HR reduction in the range of $[-10\%, -25\%]$. Additionally, an increase in systolic blood pressure is expected, with a peak occurring approximately between $[20, 40]$ seconds, corresponding to a variation of $[+5, +15]$ mmHg.

For its implementation in the software, a new SEPatientAction was first defined. The action is parameterized by a severity level $\in [0, 1]$, and a duration $[s]$. A severity value equal to 0 corresponds to no activation, whereas a severity value equal to 1 represents the maximum stimulus intensity.

```
49  // SEColdFaceTest.h
    #pragma once
51  #include <biogears/cdm/patient/actions/SEPatientAction.h>

53  namespace biogears {
    class SEScalar0To1; class SEScalarTime;
55
    class BIOGEARS_API SEColdFaceTest : public SEPatientAction {
57  public:
      SEColdFaceTest();
59    ~SEColdFaceTest() override;
      static constexpr const char* TypeTag() { return "SEColdFaceTest"; }
61    const char* classname() const override { return TypeTag(); }
      void Invalidate() override;
63    bool IsValid()  const override;
      bool IsActive() const override;
65    bool HasSeverity() const;        SEScalar0To1& GetSeverity();
      bool HasDuration() const;        SEScalarTime& GetDuration();
67    void ToString(std::ostream& s) const override;

69  private:
      SEScalar0To1*  m_Severity;
71    SEScalarTime*  m_Duration;
    };
73  }
```

The signal modulation is implemented in `Nervous.cpp`, within the block responsible for computing the autonomic efferent signals. BioGears is a deterministic, lumped-parameter model in which the autonomic effectors act through independent algebraic paths on the HR; `SEAcuteStress` (an already existing module of the BioGears engine).

It is worth clarifying the exact mechanism of `SEAcuteStress`, since this distinction is what justifies the decision not to reuse or extend it for the CFT. `SEAcuteStress` is parameterized solely by a severity scalar $\in [0,1]$ and acts exclusively within `Endocrine::ReleaseEpinephrine()`. The resulting epinephrine mass is injected into the circulation and subsequently raises HR and vasoconstriction through the model's standard humoral (blood-borne hormone) pathway, with kinetics governed by hormone release, distribution, and clearance rates rather than by an instantaneous neural signal.

The `SEAcuteStress` drives the sympathoadrenal axis, which modulates HR via β-adrenergic receptors at the sinoatrial node, whereas `SEColdFaceTest` engages the vagal arm of the trigeminocardiac reflex, reducing HR through direct vagal output. Crucially, the engine does not retain memory of past autonomic states, the effect of the CFT vanishes the instant its severity returns to zero.

To compensate this missing cross-talk while remaining within the BioGears action API, we introduced a phenomenological gain-reduction proxy, in the CFT condition, the effective severity of `SEAcuteStress` during the AT phase is multiplicatively scaled by a constant factor $k_{\text{CFTGainReduction}} \in (0,1)$, as defined in Equation (5.10).

$$\sigma_{\text{AT}}^{\text{eff}} = \begin{cases} \sigma_{\text{AT}}^{\text{nom}} & , \text{ Control condition} \\ \sigma_{\text{AT}}^{\text{nom}} \cdot k_{\text{CFTGainReduction}} & , \text{ CFT condition} \end{cases} \tag{5.10}$$

Therefore, both conditions use the `SEAcuteStress`, however, the CFT receives it with a lower severity. In the BioGears internal chain, this results in a lower EpinephrineBasalReleaseRate, which leads to a lower HeartRateModifier. Consequently, a lower HR peak is observed.

6. PhysTwin: A Personalized Human Digital Twin System

This chapter presents the final outcome of this thesis: a real time user interface built upon the latest version of the software model and its associated modifications. The final architecture is presented, and each component of the resulting use case is discussed in detail.

6.1. Use Case and System Architecture

Considering all that was discussed and done in the previous sections; the final system architecture can be seen in Figure 6.1, where the use case diagram developed considering all the aspects mentioned is presented. In the following sections, the process of using real data to generate personalized configurations will be explained. The complete methodology is described in detail in Section 6.2 and Section 6.3.

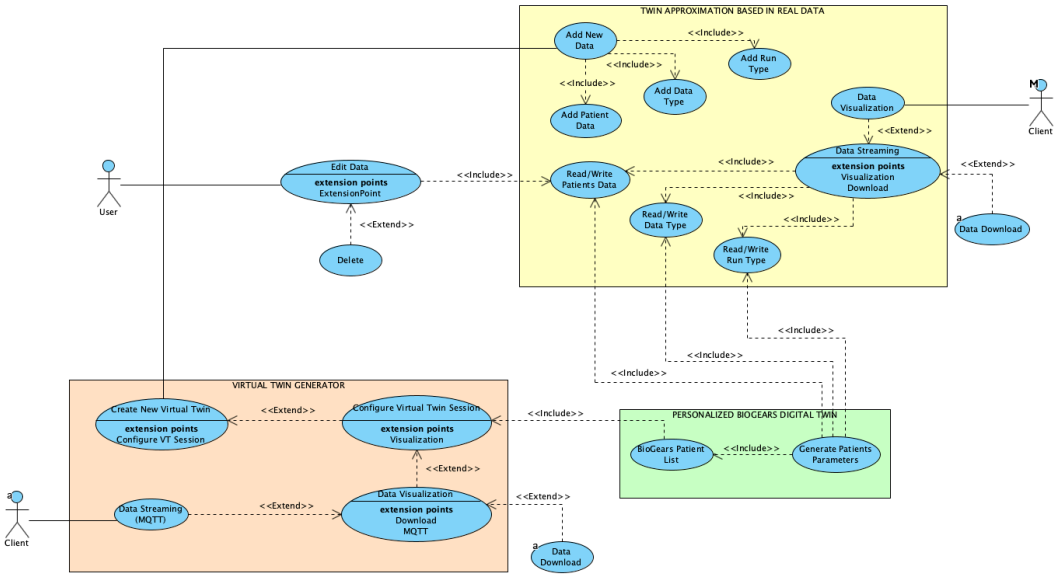

Figure 6.1: Final System Architecture Use Case Diagram.

Although the use case diagram is diverse, it remains easy to understand. First, the user can interact with the 'Twin Approximation based on real data' module by adding, editing data or visualizing the data as described in Section 4. In addition, the user can also create an avatar, i.e., a virtual human digital twin. This is achieved through the 'Virtual Twin Generator'. Once inside the Virtual Twin Generator, several use cases are available. In summary, this is where the user selects the patient to be simulated and starts the simulation.

6.2. Personalized BioGears Digital Twin Patient Creation

Figure 6.1 illustrates a component designated as the “Personalized BioGears Digital Twin”. In this system, it’s primary functionality involves generating a new patient profile within the BioGears engine. The patient specification is of fundamental importance, as it is through these parameters that the engine initializes the patient model and performs all subsequent calculations. Therefore, it is essential that these values accurately represent the real patient and are as close to reality as possible. A wide range of parameters can be defined when creating a new simulation. A complete list of available variables and their corresponding values can be found in [10].

The incorporation of real world data into the patient variables plays a crucial role in improving the accuracy of the simulation. For example, if the default value of a physiological parameter, such as heart rate, is set to 70 bpm, and the incorporation of real world measurements adjusts this value to 65 bpm, the entire engine is affected, since all BioGears modules are interconnected. Consequently, even small changes in the initial conditions can propagate through the system and produce significant effects on subsequent calculations.

Thus, the creation of new patients cannot be executed via the UI. Because the software is pre compiled and runs from a binary executable, users are restricted from dynamically creating patients at runtime. The BioGears engine includes $(N = 24)$ predefined patient profiles. However, if desired, the user can create additional patient profiles.

To introduce a new patient, the configuration must be manually hardcoded. This requires accessing the `bio_V5/core_V5/share/data/patients` directory and adding a new `.xml` file containing the patient’s physiological data. Furthermore, this data must strictly adhere to a specific format and sequence, as documented in [10]. An example of how the `.xml` file must be for a patient can be seen:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Patient xmlns="uri:/mil/tatrc/physiology/datamodel" contentVersion="Biogears_7.5.0+85"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="uri:/mil/tatrc/
physiology/datamodel BioGearsDataModel.xsd">
  <Name>Female_30_Normal</Name>
  <Sex>Female</Sex>
  <Age readOnly="false" unit="yr" value="30"/>
  <Weight readOnly="false" unit="lb" value="130"/>
  <Height readOnly="false" unit="in" value="64"/>
  <BodyFatFraction readOnly="false" value="0.18"/>
```

```
  <DiastolicArterialPressureBaseline readOnly="false" unit="mmHg" value="73.5"/>
  <HeartRateBaseline readOnly="false" unit="1/min" value="72"/>
  <RespirationRateBaseline readOnly="false" unit="1/min" value="18"/>
  <SystolicArterialPressureBaseline readOnly="false" unit="mmHg" value="114"/>
</Patient>
```

Following this modification, the entire source code must be recompiled to integrate the new patient profile into the executable binary.

However, once a new patient is successfully created and the system is recompiled, the existing architecture automatically accommodates these future profiles. This is because the core source code is designed to dynamically scan and read all available patient files within the `bio_V5/core_V5/share/data/patients` directory. Thus, while the manual configuration and generation of a functional patient profile present initial technical challenges, any successfully integrated profile remains permanently available throughout the subsequent pipeline.

6.3. BioGears and Real Data Fusion

The integration of real world data with the BioGears engine represents a core contribution of this work. As detailed in Section 4.4, the user can generate a configuration (`.cfg`) file containing the patient's specific physiological parameters. Once generated, this file is parsed by the BioGears engine, which overrides its default baseline values with the personalized ones. Consequently, these calculated and patient specific parameters serve as a direct input to drive the simulation engine, as seen in Figure 6.2.

To ensure seamless integration between pre existing profiles and newly created ones as described in Section 6.2, the data fusion process must occur dynamically upon patient selection. To achieve this, the developed system triggers an automated workflow whenever a user switches the active patient profile, This process only occurs when the user switches between patients. The entire workflow is triggered exclusively during a patient exchange, due to the need to regenerate and update all associated data for the new configuration.

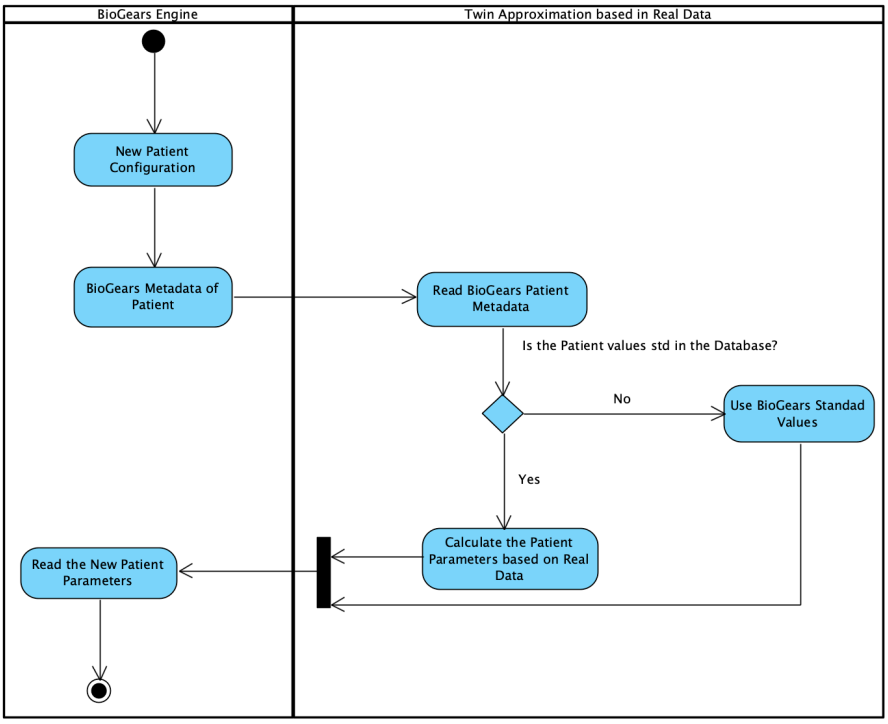

Figure 6.2: BioGears and real data fusion flowchart.

First, BioGears transmits the patient's physiological metadata (e.g., sex, age, height, and weight) to the "Twin Approximation Based on Real Data" system, the actually metadata that the BioGears sends to the python script was discussed and presented in Section 6.2. This system processes the metadata to identify or construct the most visually and physiologically similar profile, calculating the necessary parameters. This process is performed by reading the metadata and querying the database to identify the most similar patient. The most similar patient is determined based on real world data using a default tolerance of ±5 units for each metric. This tolerance can also be adjusted by the user. For example, if the patient's weight is 80 kg, the software will, by default, search for patients whose weights fall within the range of 75– 85 kg.

Parameters	Brief Explanation	Mathematical Approach
k_{rise}	Rate constant of the first-order (low-pass) filter governing the exponential rise of HR toward the target set by exercise intensity.	$HR(t) = HR_{\text{rest}} + (HR_{\text{target}} - HR_{\text{rest}})\left(1 - e^{-k_{\text{rise}}\, t}\right)$
HR_{max}	The original BioGears engine uses an age-based formula; here it is replaced by Tanaka et. al, [43].	$HR_{\text{max}} = 220 - \text{age} \quad (\text{default})$ $HR_{\text{max}} = 220 - 0.7 \times \text{age}$
τ_1	Time constant of the **fast** component of post-exercise HR recovery.	$HR(t) = HR_{\text{rest}} + A_1 e^{-\frac{t}{\tau_1}} + A_2 e^{-\frac{t}{\tau_2}}$
τ_2	Time constant of the **slow** component of recovery.	$HR(t) = HR_{\text{rest}} + A_1 e^{-\frac{t}{\tau_1}} + A_2 e^{-\frac{t}{\tau_2}}$
$A1_{\text{frac}}$	Fraction of the initial HR offset $(HR_{\text{peak}} - HR_{\text{rest}})$ attributed to the fast component τ_1.	$A_1 = a1_{\text{frac}}\,(HR_{\text{peak}} - HR_{\text{rest}})$
$A2_{\text{frac}}$	Complementary fraction attributed to the slow component τ_2, with $a1_{\text{frac}} + a2_{\text{frac}} = 1$ by construction of the fit.	$A_2 = a2_{\text{frac}}\,(HR_{\text{peak}} - HR_{\text{rest}})$

Table 6.1: Parameters passed to the BioGears Engine.

Once the candidates are retrieved (typically more than one), the algorithm computes the parameters to be passed to BioGears, see Table 6.1. Once generated, these parameters are fed back into the BioGears engine, closing the execution loop. BioGears then uses these parameters to overwrite the default values, resulting in a more personalized patient configuration within the simulation. This continuous feedback loop ensures fluid, automated communication across all system components.

The parameters presented in Table 6.1 were selected because they are required for the accurate calculation of heart rate rise and decay, as described by Equation (5.1), Equation (5.2), and Equation (5.4). Therefore, these parameters play a crucial role in the heart rate estimation process, given the equations employed in the model.

6.4. User Interface and Client Service

Considering all the improvements made to the software, as described in Section 5, and given that the latest version ($V5$) is the most optimized one, a real time user interface was

designed and implemented as the final outcome of this work using the code presented in BioGears $V5$.

The software was developed with user practicality and ease of use in mind. The main objective is that, after the initial execution, the system can run continuously without interruption, similarly to a virtual human operating in real time. Naturally, there are limitations to this approach; for instance, if the simulated patient reaches an irreversible physiological state (i.e., death), the engine must be restarted. Whenever the patient configuration is modified through the User Interface, the software automatically restarts the engine in order to load the newly selected patient profile and apply the corresponding physiological parameters.

As the first official version, the software supports the implementation of several non clinical scenarios that were previously studied and further developed, namely:

- Run (Treadmill); `SEExercise::SERunning`
- Cycle (Bicycle); `SEExercise::SECycling`
- Exercise (generic); `SEExercise::SEGeneric`
- Cold Face Test; `SEColdFaceTest`
- Acute Stress; `SEAcuteStress`
- Sleep; `SESleep`

Therefore, it is possible to subject the patient to exercise, stress conditions, the CFT, and other protocols, as seen in the Figure 6.3. For the real time user interface, a module within BioGears $V5$ was developed to read data in real time and expose a local server (localhost) for visualization purposes. It is worth noting that when the patient is first exposed to the CFT and subsequently subjected to stress, the physiological response to stress is expected to be attenuated compared to a baseline (i.e., non CFT) condition [6]. In other words, without prior exposure to the CFT, a stronger physiological response is typically observed.

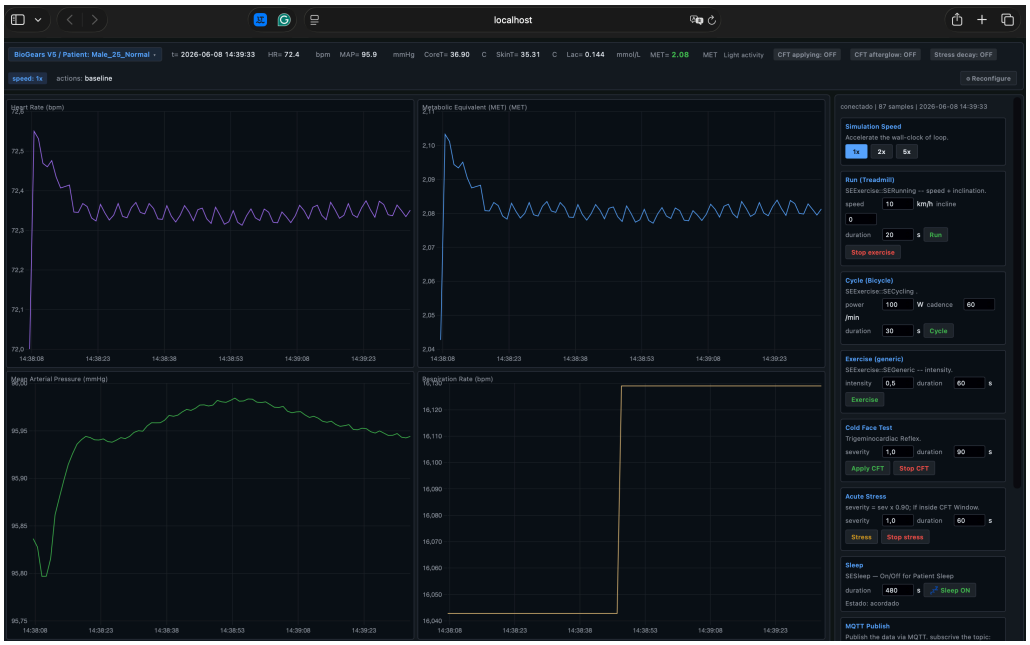

Figure 6.3: Real Time User Interface.

Additionally, the Message Queuing Telemetry Transport (MQTT) protocol with the Mosquitto was added to enable data transmission through the UI. For this purpose, the user can use the broker.hivemq service [59]. The MQTT topic can be customized by the user. By default, the topic is set to `biogears/data/vitals`; however, the user may define any topic as desired. The only point to note is that, if the topic is changed, all users or applications that subscribe to the MQTT data stream must be informed of the new topic. By default, to view the data being transmitted via MQTT in the terminal, assuming Mosquitto is already installed, it is only necessary to type `mosquitto_sub -h broker.hivemq.com -p 1883 -t biogears/data/vitals`.

Also, it was added two main configurations, the first is that the user must select four parameters to be displayed in the graph, as well as any number of parameters to be transmitted via MQTT. After the initial configuration, whenever the software is reopened, these settings can be modified by clicking the `Reconfig` button, where Figure 6.4 shows this configuration for the four parameters to be displayed in the graph and Figure 6.5 shows the configuration for the MQTT protocol.

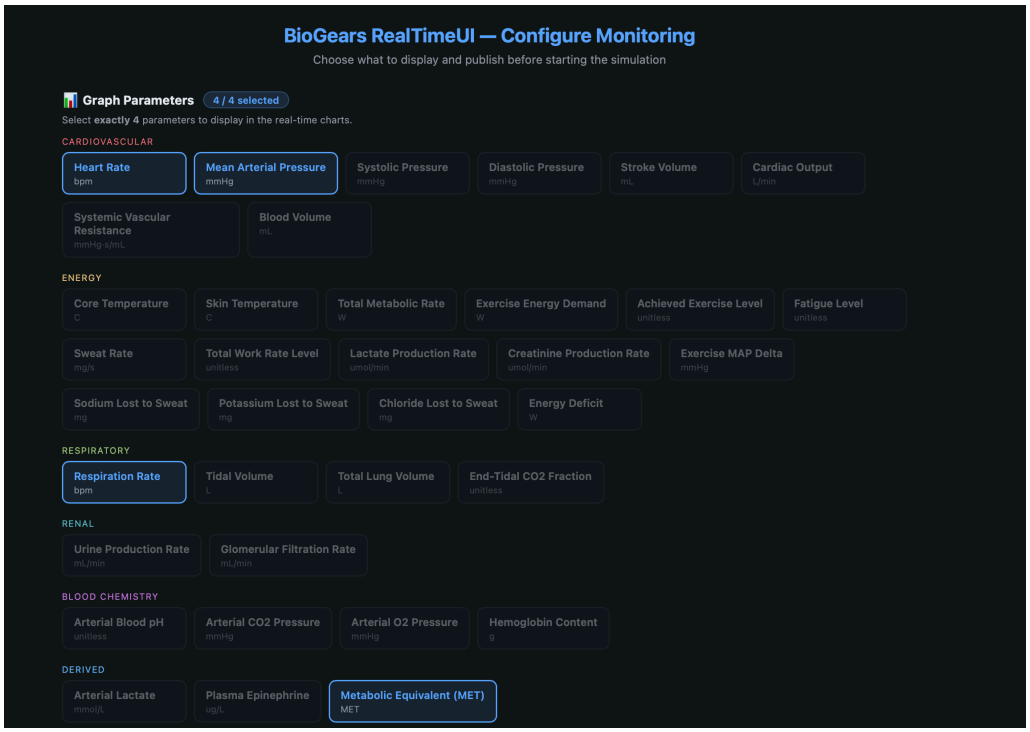

Figure 6.4: Real Time User Interface Configure Graph Parameters Monitoring.

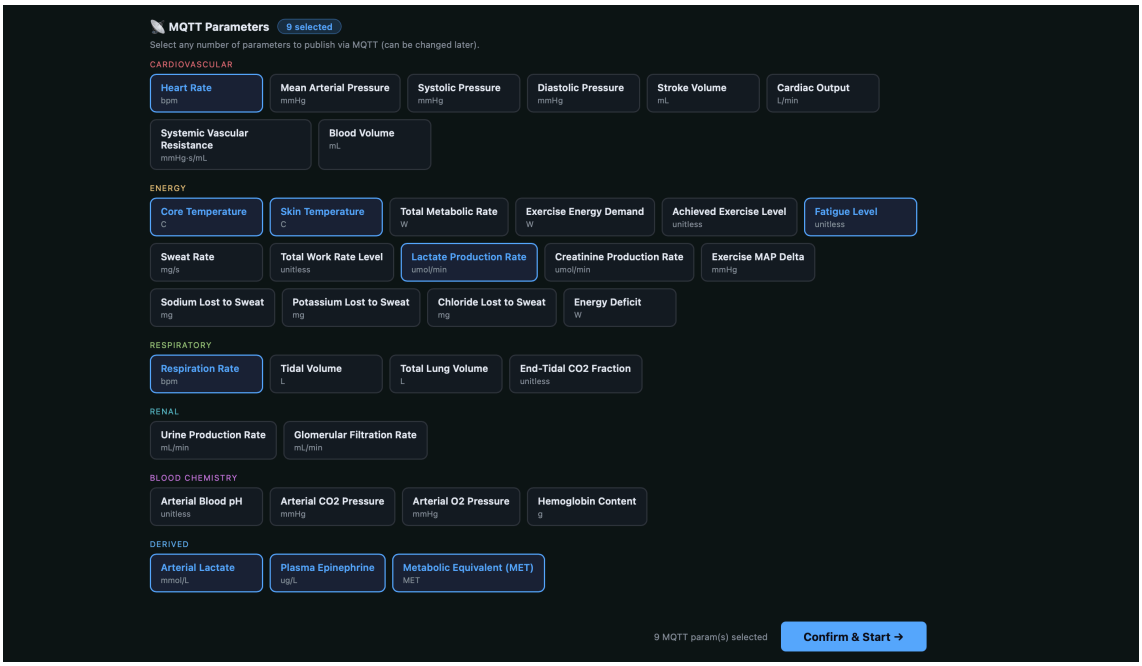

Figure 6.5: Real Time User Interface Configure MQTT Parameters Monitoring.

The second option is the Diary mode, which was developed for the use cases in which the user intends to generate and simulate a patient over a 24-hour period. In this mode, the user can define the patient's daily routine and initiate the diary simulation, as seen in Figure 6.6. Once started, the diary continuously executes the predefined schedule in a loop, with each simulated cycle corresponding to a 24-hour period. The simulation remains active until the user explicitly terminates the diary execution.

Figure 6.6: Real Time User Interface Diary Example.

To provide an easier way to launch and use the UI, the script `biogears_launcher.py` was developed. This Python script automatically launches the `Howto-realtimeui.cpp` application and starts the UI, as seen in Figure 6.7. To further simplify its use, the script was also compiled into an executable (`.exe`) file. Therefore, it can be used as a standalone application; the user only needs to run the executable, and all the required processes are started automatically.

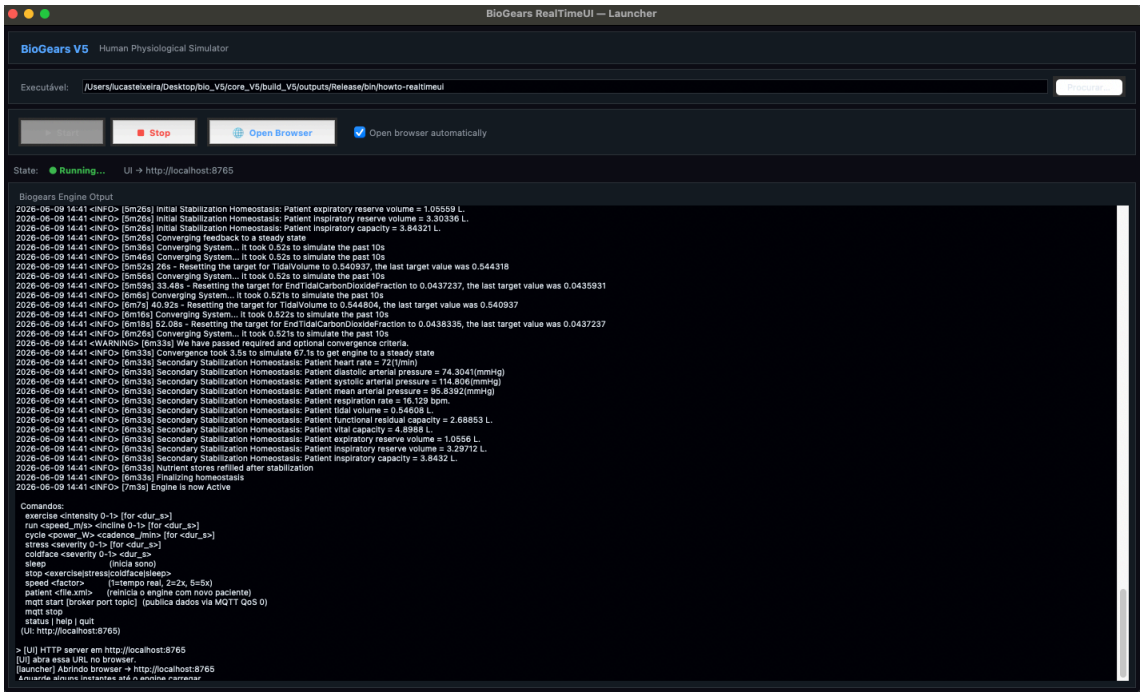

Figure 6.7: BioGears Real Time UI Launcher.

6.5. How to Use the System

Considering that the real-time user interface currently provides six available actions, and that all of these actions, except for `SESleep`, which is an existing feature of the BioGears software, were either developed or modified during the course of this thesis to produce more realistic outputs; it is important to explain both the process of creating a new `SEAction` within BioGears and the methodology used to collect and generate reliable data.

To create a new `SEAction`, the user must first identify the desired output variable, such as HR. Once the target output has been defined, it is necessary to either design an experimental protocol or gather suitable data from reliable sources to be incorporated into the Twin Approx database. After the required data have been collected, Twin Approx can be trained using the real world dataset to determine the parameters that most accurately reproduce the desired physiological response (see Section 4).

Following the training process, the generated model must be validated using a protocol considered reliable and appropriate for the intended application, as was performed in the Section 7 of this thesis. Once satisfactory validation results have been obtained, the generated parameters can be integrated into BioGears. This procedure was followed for the HR model, whose parameters were validated and subsequently stored in the `exercise_params` directory. During simulation, these parameters are used in place of the precomputed values provided by the original BioGears implementation.

Finally, if the required `SEAction` does not already exist within the software, as was the case for `SEColdFaceTest`, the user must implement the new action following the methodology described in the previous sections. This involves creating the corresponding action class, integrating it into the BioGears framework, and ensuring that the newly generated parameters are correctly applied during simulation.

If necessary, all modifications inside `core_V5/projects/howto/RealTimeUI/src/Howto-RealTimeUI.cpp` are free to make, as it is open source code. However, if any modifications are made, it is necessary to recompile the code; that is, enter `core_V5/build_V5` and run `ninja howto-realtimeui`. This step is required because the `.cpp` file is the source code, and the computer executes the compiled binary `howto-realtimeui`. Therefore, when changes are made to the `.cpp` file, the existing binary on disk does not reflect them. After recompilation, the updated binary replaces the previous version on disk.

6.6. System Deployment

Once the user UI was completed and fully functional, as described in Section 6.4, the next step was to make the software available for real world operational use. To achieve this, a Virtual Machine (VM) was deployed within the INESCTEC infrastructure. The specifications of the VM used is described in Table 6.2.

Parameters	Values
RAM	8 GB
Processor	AMD EPYC-Milan Processor
CPU	4 vCPUs
Disk Space	50 GB

Table 6.2: Virtual Machine Parameters.

The entire software stack was documented and configured within the VM, and the application was packaged to be as simple as possible to run it. This approach allows users to interact with the software as they would with any standard desktop application, significantly simplifying deployment and usage. Furthermore, since both the software and the UI support data transmission through MQTT, one of the primary use cases involves a user accessing the VM, creating an avatar within the software, configuring a 24-hour diary, and initiating real-time data transmission.

Once inside the VM, the user only needs to type `./startEngine`, and the entire system will start running automatically. This design was implemented to simplify the usage of the virtual machine. Therefore, the `./startEngine` command acts as the main entry point for launching the application.

```
89 ~/bioGearsEngine/bio_V5/core_V5/build_V5/outputs/Release/bin/howto-realtimeui
```

Through this VM based deployment, it is possible to operate a HDT continuously, 24 hours a day and 7 days a week, under different scenarios and activity profiles.

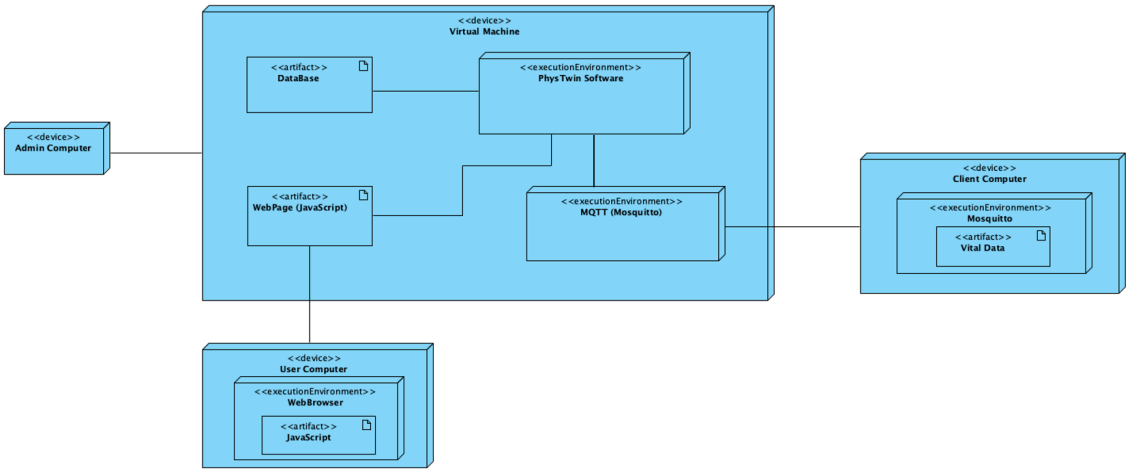

Figure 6.8: System Deployment Diagram.

In Figure 6.8, a simplified diagram is presented to clarify the system architecture. The general workflow is as follows: the user accesses the VM, and once inside, the main code is executed, generating the HTML interface used for data visualization. Additionally, there is the option to run the MQTT broker externally in a publicly accessible configuration using HiveMQ. In this setup, external users can subscribe to the MQTT topics to access and stream the data in real time.

7. System Validation

In this chapter, the validations are made in order to assess whether the modifications introduced in the different versions effectively improved the system. For each protocol, the methodology, the reference papers and the obtained results are presented, providing a comprehensive analysis and a critical evaluation of whether the implemented changes were beneficial.

7.1. Statistical Methods for Validation

The Mean Absolute Error (MAE) was adopted as the primary scalar metric for quantifying the discrepancy between each BioGears simulation and the reference trajectory. For a single physiological variable X of subject s, sampled at N equally spaced time instants over the full protocol, the MAE is defined as:

$$\mathrm{MAE}_{X,s} = \frac{1}{N}\sum_{i=1}^{N}|y_i^{\mathrm{ref}} - \hat{y}_i^{\mathrm{sim}}| \tag{7.1}$$

where y_i^{ref} denotes the reference value of variable X at time t_i and $\hat{y}_i^{\mathrm{sim}}$ denotes the corresponding value produced by the BioGears engine under the same stimulus protocol. The absolute value ensures that positive and negative deviations do not cancel, so Equation (7.1) estimates the expected pointwise distance between the two trajectories in the native units of the variable.

To compare versions of the engine across subjects, a per-variable aggregate was computed as the mean MAE over the simulated subjects S:

$$\bar{\mathrm{MAE}}_X = \frac{1}{|S|}\sum_{s\in S}\mathrm{MAE}_{X,s} \tag{7.2}$$

Finally, the Friedman and Wilcoxon tests require a single scalar per observation, a composite MAE was defined by averaging the per-variable errors across all physiological signals and patients:

$$\mathrm{MAE}_{\mathrm{composite}} = \frac{1}{|S|\cdot|X|}\sum_{s\in S}\sum_{X}\mathrm{MAE}_{X,s} \tag{7.3}$$

It should be noted that Equation (7.3) aggregates quantities with different physical units and is therefore a pragmatic rather than dimensionally rigorous summary; it is

retained as a compact overall indicator, with per-variable analyses (Equation (7.1) and Equation (7.2)) serving as the physically interpretable reference.

All validation protocols presented in this chapter are conducted exclusively using virtual patients generated within the BioGears engine. No real-world physiological data from human subjects was collected as part of this work. Comparisons are therefore made against reference values reported in peer reviewed literature, and any deviation from those values must be interpreted in light of this constraint, the simulator is validated against population level physiological norms, not against individual experimental measurements.

The rationale behind the development of three distinct protocols and the use of real world data from the literature, rather than the database data, is straightforward. Since the parameters used by the final software are derived from the database itself, employing the same dataset for validation would introduce a significant bias. In other words, the database is used to generate the parameters of a specific patient; therefore, comparing the software outputs against the same data used for parameter generation would naturally result in high accuracy.

However, such an approach would be biased and would not provide a reliable assessment of the software's performance. Consequently, independent real world data obtained from the literature were used for validation in order to ensure a more realistic and unbiased evaluation of the proposed methodology.

7.2. Protocol A: Exercise Scenarios

7.2.1. Methodology

For the first validation test; the study by Attar et al. [60] was adopted as the reference for comparison. Based on this work, a corresponding experimental protocol was implemented and applied consistently across all five versions of the BioGears engine.

The protocol comprises three exercise scenarios defined for the same patient. Each scenario consists of a five minute activity period, followed by a two minute recovery interval, ensuring partial physiological stabilization between efforts.

According to the reference study, the prescribed exercise intensities are defined in terms of speeds ($5\frac{km}{h}$ and $9\frac{km}{h}$). However, for the implementation in the BioGears, it was necessary to convert these speeds into equivalent intensity levels. This conversion

introduces an additional modeling step, which may influence the fidelity of the comparison and should therefore be considered when interpreting the results.

For the first three versions ($V1$ – $V3$), the exercise intensity is computed as a fraction of the maximum mechanical power output:

$$Intensity = \frac{W}{W_{\max}} \tag{7.4}$$

where $W_{\max} = 1200W$

In contrast, for versions $V4$ and $V5$, the definition of intensity is reformulated based on the oxygen consumption reserve ($VO_{2,\,\text{reserve}}$), providing a more physiologically meaningful representation:

$$Intensity = \frac{VO_2 - VO_{2,\,\text{rest}}}{VO_{2,\,\max} - VO_{2,\,\text{rest}}} \tag{7.5}$$

This formulation normalizes the metabolic demand relative to the individual's functional capacity, accounting for both resting and maximal oxygen consumption levels. Consequently, after converting locomotion speed into model-specific intensity values, the following correspondence is obtained, as can be seen in Table 7.1:

Speed (km/h)	V1 - V3	V4 - V5
5	0.245	0.331
9	0.414	0.561

Table 7.1: Comparative of Speeds: V1 to V5

These results highlight that, for the same external workload, the newer formulation ($V4$ – $V5$) yields higher intensity values. This reflects the transition from a purely mechanical normalization to a physiologically grounded metric, thereby improving the consistency and interpretability of the model across different subjects and conditions.

7.2.2. Result

First, three patients were considered in the analysis. The first corresponds to the `StandardMale.xml` profile, representing a 44 year old male with a body mass of 77 kg, a height of 180 cm, and a body fat fraction of 0.21. For this subject, the script `prediction_V6.py` was employed to identify the closest matching individual within the Twin Approx database. The retrieved parameters were used to calibrate and fine-tune the model in the later versions of the software (from $V2$ onwards). The results for the first patient can be seen in Figure 7.1 and Figure 7.2.

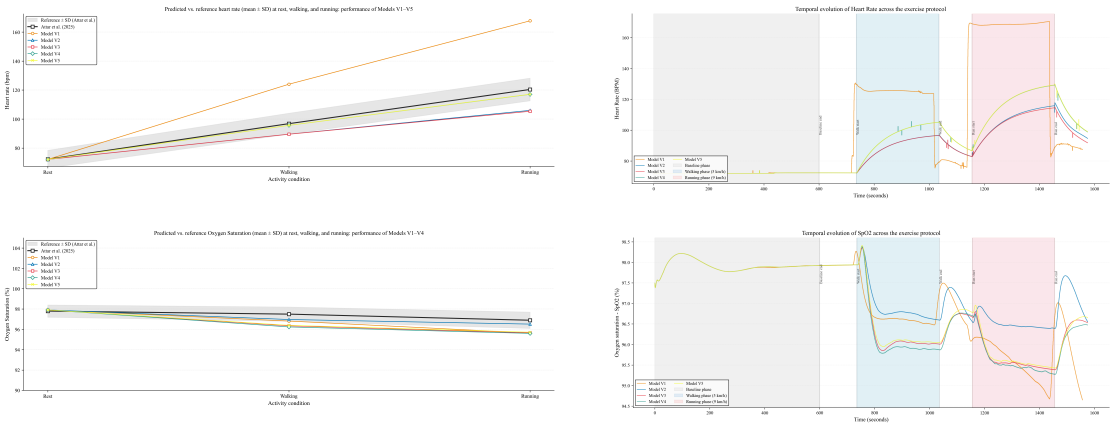

Figure 7.1: Heart Rate and SPO2 for StandardMale Patient

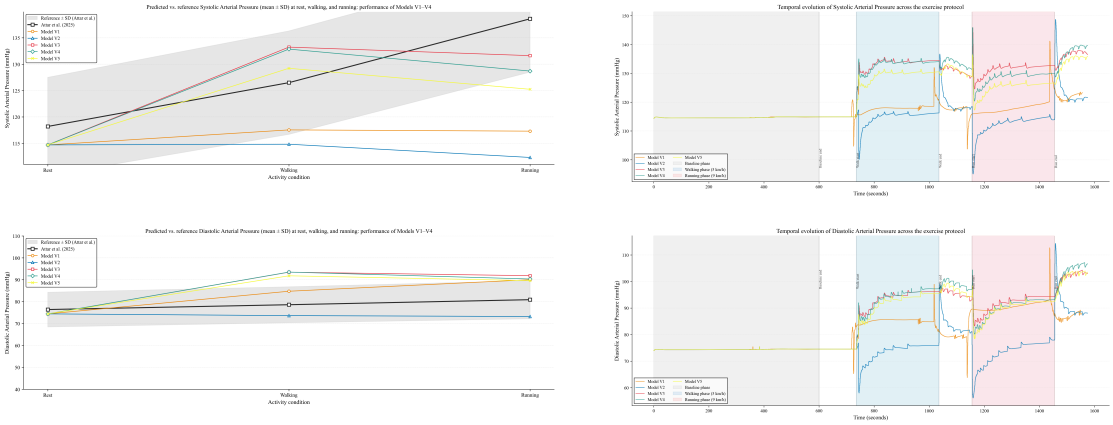

Figure 7.2: Systolic/Dyastolic Arterial Pressure for StandardMale Patient

The second corresponds to the `Male_25_Normal.xml` profile, representing a 25 year old male with a body mass of 77 kg, a height of 180 cm. For this subject, the script same script, `prediction_V6.py` was employed to identify the closest matching individual within the Twin Approx database. The results for the second patient can be seen in Figure 7.3 and Figure 7.4.

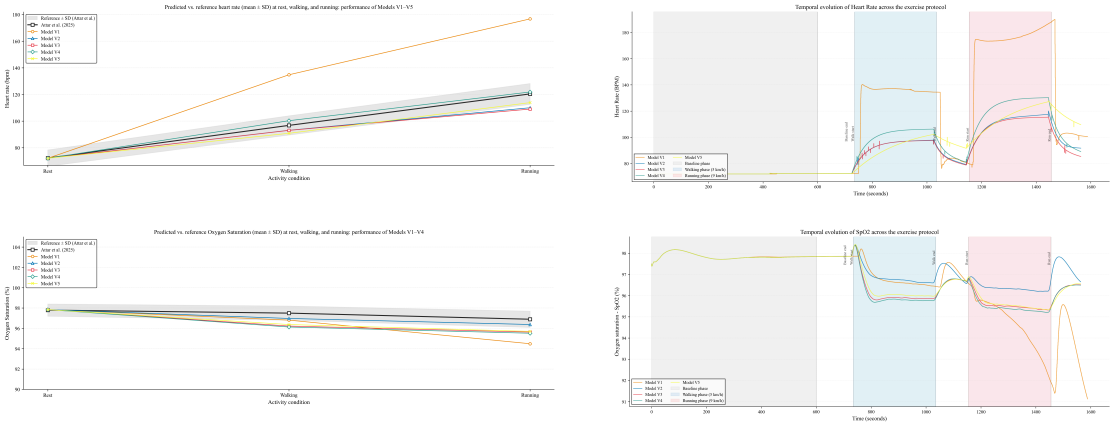

Figure 7.3: Heart Rate and SPO2 for Male 25 Normal Patient

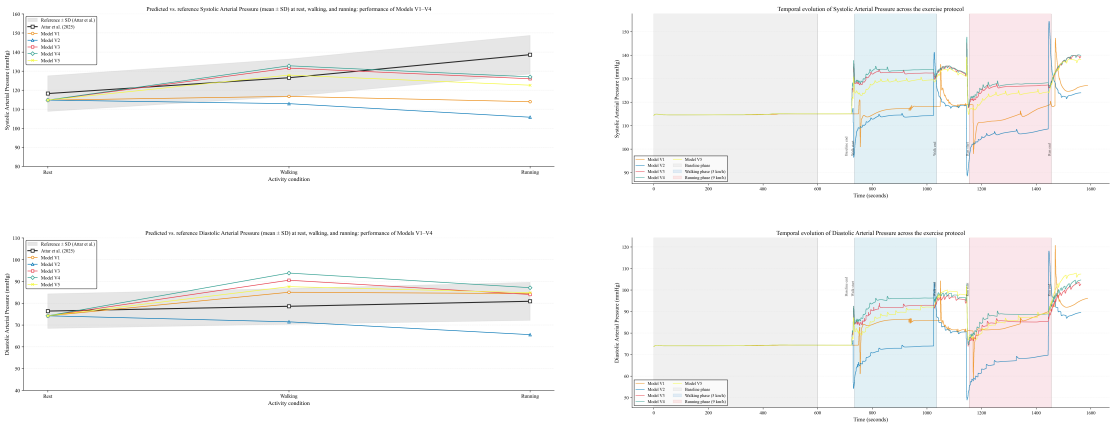

Figure 7.4: Systolic/Dyastolic Arterial Pressure for Male 25 Normal Patient

The third subject corresponds to the `Male_35_Normal.xml` profile, which was introduced due to its absence in the original software. This profile represents a 35 year old male with a body mass of 81 kg and a height of 182 cm. As in the previous cases, the `prediction_V6.py` script was employed to identify the closest matching individual within the Twin Approx database. The resulting parameters were then used to fine-tune the model, ensuring consistency in the calibration procedure across all evaluated subjects. The results for the third patient can be seen in Figure 7.5 and Figure 7.6.

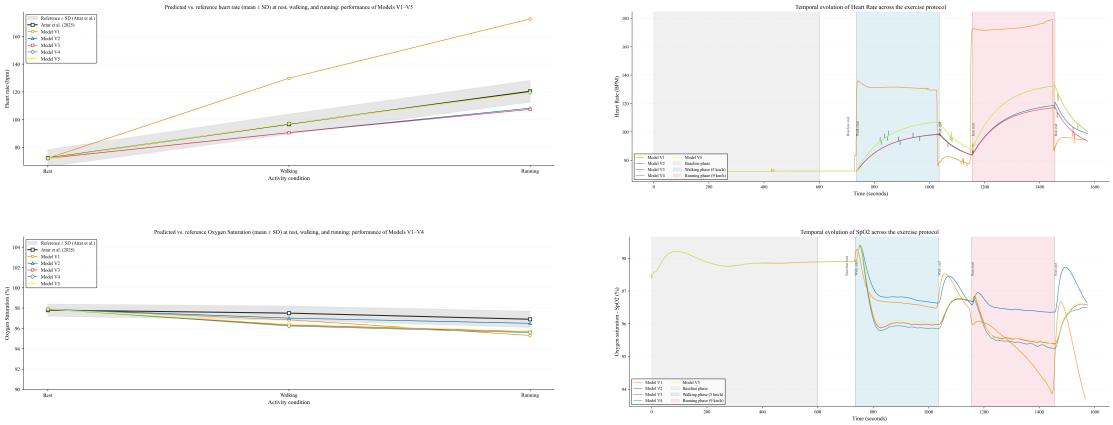

Figure 7.5: Heart Rate and SPO2 for Male 35 Normal Patient

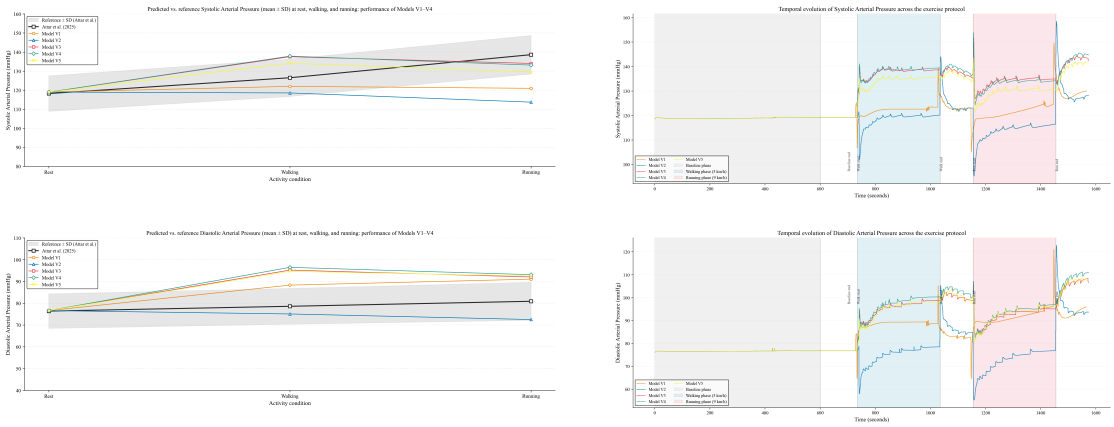

Figure 7.6: Systolic/Dyastolic Arterial Pressure for Male 35 Normal Patient

Therefore, for the validation of the different model versions, a set of performance metrics and statistical indicators was computed. The central hypothesis of this dissertation posits that the performance of the BioGears engine, particularly in the context of non clinical physiological responses, can be improved through targeted physiological recalibration and structural modifications to the underlying software architecture.

Among all evaluated versions, Version 5 ($V5$) demonstrated the best overall performance, achieving the lowest MAE when compared to the reference data. Specifically, $V5$ obtained an MAE of 4.17, outperforming the previous versions ($V4 = 4.39$; $V3 = 5.35$; $V2 = 6.50$; $V1 = 11.28$). This progressive reduction in error across versions indicates that the implemented modifications contribute positively to model accuracy and predictive capability as can be seen in Figure 7.7.

Furthermore, the consistent improvement observed from $V1$ through $V5$ suggests that both the physiological adjustments and architectural refinements play a significant role in enhancing the model's ability to reproduce realistic non clinical responses. These findings reinforce the validity of the proposed approach and highlight the importance of iterative model development guided by quantitative evaluation metrics.

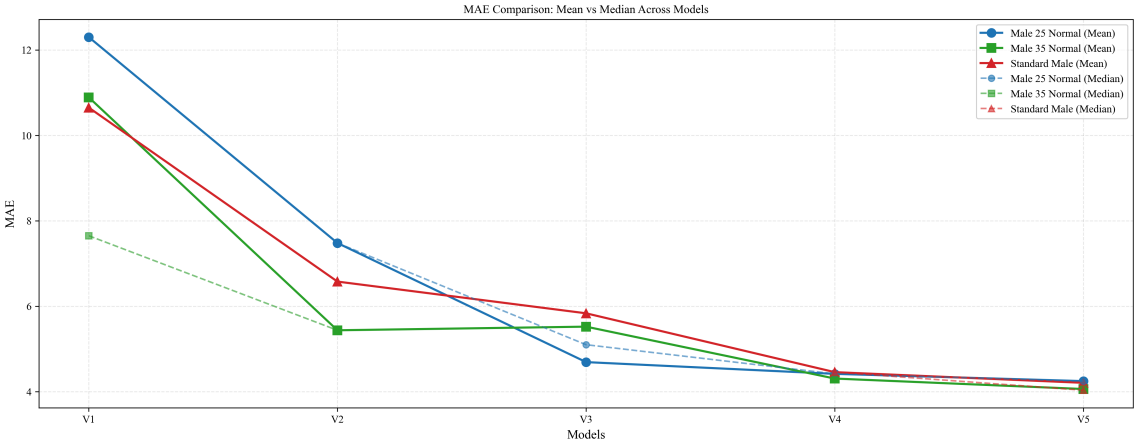

Figure 7.7: MAE Comparison: Mean vs Median Across Models (V1 - V5).

With the computed MAE values, it was possible to derive both the mean and median error metrics for all model versions ($V1$–$V5$). As illustrated in Figure 7.8, $V5$ consistently exhibits the lowest values for both the mean and median MAE across all evaluated scenarios. This result is not merely visual but quantitatively consistent across the datasets, indicating a systematic reduction in prediction error rather than an isolated improvement. The agreement between mean and median metrics further suggests that the performance gain is robust and not driven by outliers or skewed error distributions.

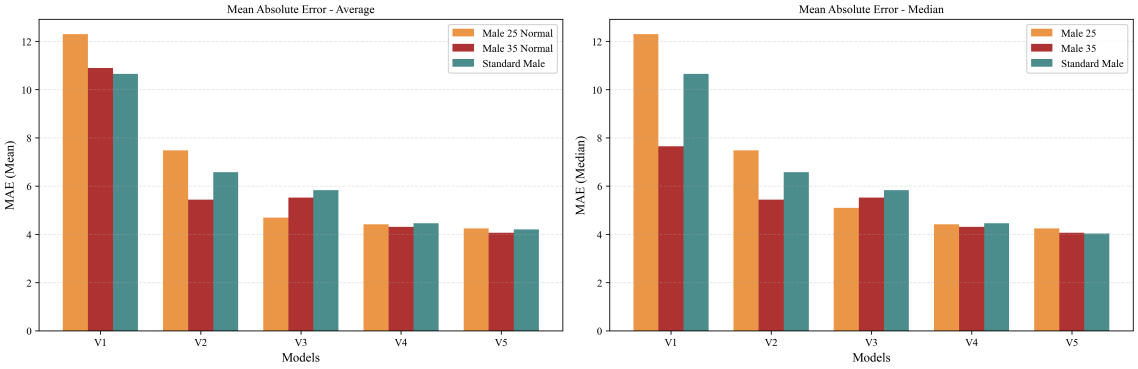

Figure 7.8: Mean Absolute Error Across Models (V1 - V5).

For a more comprehensive and quantitative assessment of the results, boxplots were constructed to analyze the distribution of the MAE across the different model versions. This representation enables not only the evaluation of central tendency but also the dispersion, variability, and presence of outliers in the error distributions. As illustrated in Figure 7.9, a clear trend of error reduction is observed from $V1$ to $V5$ across all considered subjects. In particular, $V5$ consistently exhibits lower median values and a narrower inter quartile range, indicating both improved accuracy and greater stability. Overall, the boxplot analysis supports the conclusion that the iterative modifications introduced

throughout the model development lead to progressive improvements in both precision and reliability.

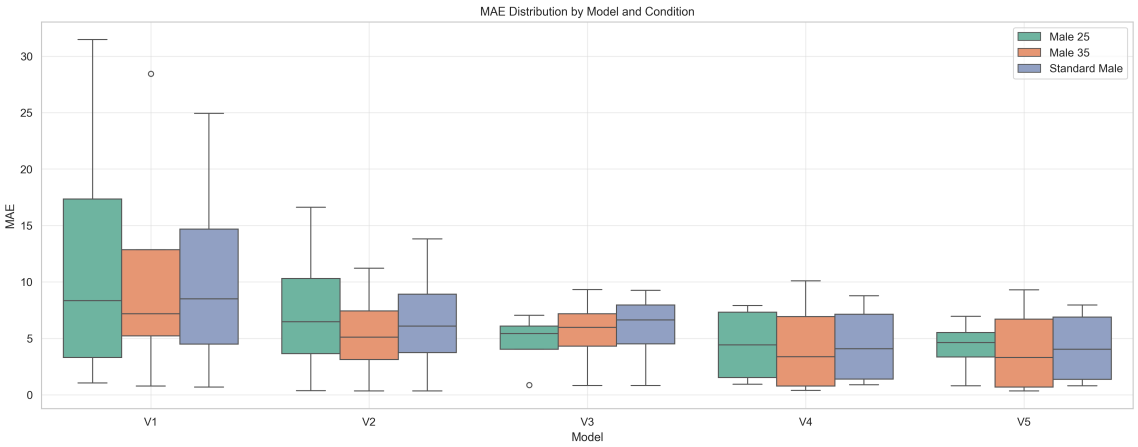

Figure 7.9: BoxPlot of MAE Distribution by Model and Condition (V1 - V5).

To substantiate the qualitative trends apparent in Figure 7.9, a non-parametric inferential analysis was performed on the full set of twelve paired observations per version (3 subjects x 4 physiological variables). Descriptively, the composite mean MAE decreases from $V1 = 11.28$ to $V5 = 4.17$, a reduction of 63.0%, while the standard deviation of the error distribution simultaneously contracts from 11.01 to 3.21. This dual contraction, of both central tendency and dispersion, is consistent with the visual narrowing of the boxplot whiskers and inter quartile ranges from $V1$ through $V5$, and indicates that the iterative recalibration procedure is improving not only the expected accuracy but also the robustness of the simulator across physiological profiles.

A Friedman test across the five versions ($k = 5$, $n = 12$) yielded $\chi^2(4) = 6.13$, $p = 0.19$, which does not reach formal significance at the $\alpha = 0.05$ level; this outcome, however, is attributable to the limited statistical power of the block size ($n = 12$) and to the heterogeneous magnitudes involved when $V1$ and $V5$ compete for ranks simultaneously, rather than to true distributional equivalence. To probe the most stringent pairwise comparison, the transition from the previous best model to the latest one, a one-sided Wilcoxon signed-rank test was applied between $V4$ and $V5$, testing the directional hypothesis $\mathrm{MAE}_{V5} < \mathrm{MAE}_{V4}$.

The test returned $Z = 2.157$, $p = 0.0155$, with an effect size of $r = \frac{|Z|}{\sqrt{N}} = 0.62$. Out of twelve paired observations, eleven (91.7%) showed a lower MAE in $V5$ than in $V4$, with a single marginal regression confined to the HR of the youngest subject. These results provide quantitative corroboration that the progressive reduction of median and inter quartile

range visible in Figure 7.9 reflects a statistically significant and practically meaningful improvement.

It was also done a one-sided Wilcoxon signed-rank test between $V1$ and $V5$, testing the directional hypothesis $\mathrm{MAE}_{V5} < \mathrm{MAE}_{V1}$. The test returned $Z = 1.451$, $p = 0.07335$, with an effect size of $r = \frac{|Z|}{\sqrt{N}} = 0.42$. Out of twelve paired observations, seven (58.3%) showed a lower MAE in $V5$ than in $V1$, while five (41.7%) favored $V1$.

This lack of statistical significance, however, masks a strong metric-dependent dissociation: $V5$ demonstrated dramatically lower MAE for HR ($\Delta = +27.25$) and Systolic Pressure ($\Delta = +4.72$), whereas $V1$ excelled in Diastolic Pressure ($\Delta = -2.21$), with SpO2 yielding a practical tie ($\Delta = -0.01$). These results indicate why $V5$ does not uniformly outperform $V1$ across all physiological parameters.

Considering this, the entire protocol was redesigned by including all 24 patients available in the software, for the $(n = 24)$ patients, the Absolute Percentage Error (APE) was calculated according to the equation described in Equation (7.6). The APE represents the relative error of a single data point expressed as a percentage. From the APE, it is also possible to derive the Mean Absolute Percentage Error (MAPE) and the Median Absolute Percentage Error (MdAPE), which are described by Equations Equation (7.7) and Equation (7.8), respectively.

$$APE_i = 100 \times \frac{|y_i^{sim} - y_i^{ref}|}{|y_i^{ref}|} \tag{7.6}$$

$$MAPE = \frac{1}{N} \sum_i APE_i \tag{7.7}$$

$$MdAPE = median_i APE_i \tag{7.8}$$

Therefore, it was outputed the following results.

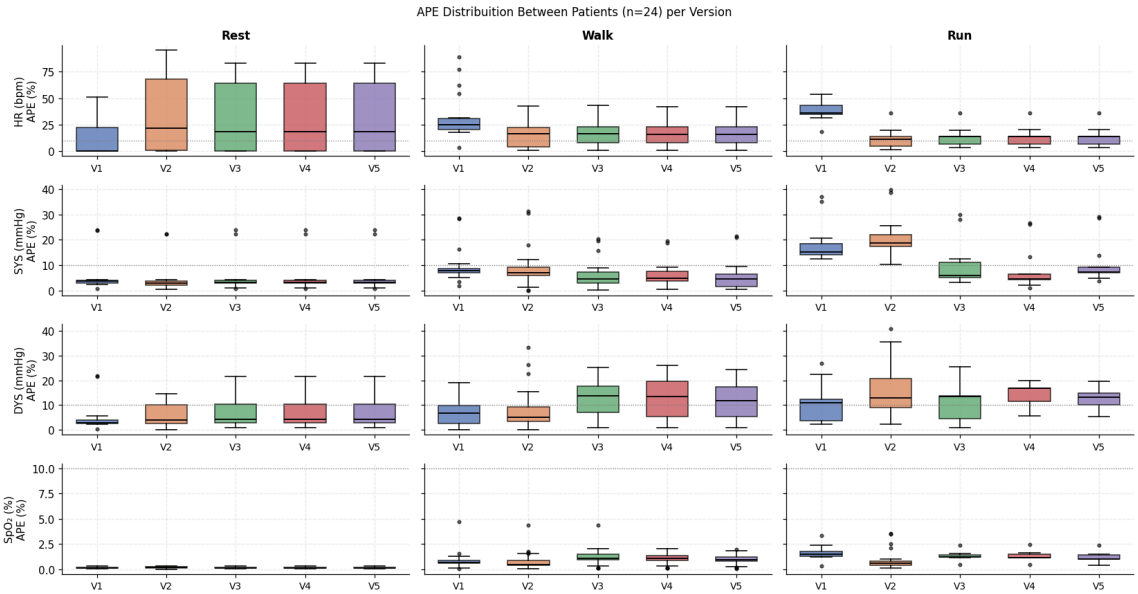

Figure 7.10: APE Distribution between Patient by Model(V1 - V5).

The Figure 7.10 demonstrates a significant reduction in error between the original version (V1) and the subsequent versions, particularly for the HR values in activity. This result was expected, since substantial modifications were made to the HR calculation, including changes to both the governing equation and the parameters used in the software. Smaller, although less pronounced, improvements can also be observed for the other physiological parameters.

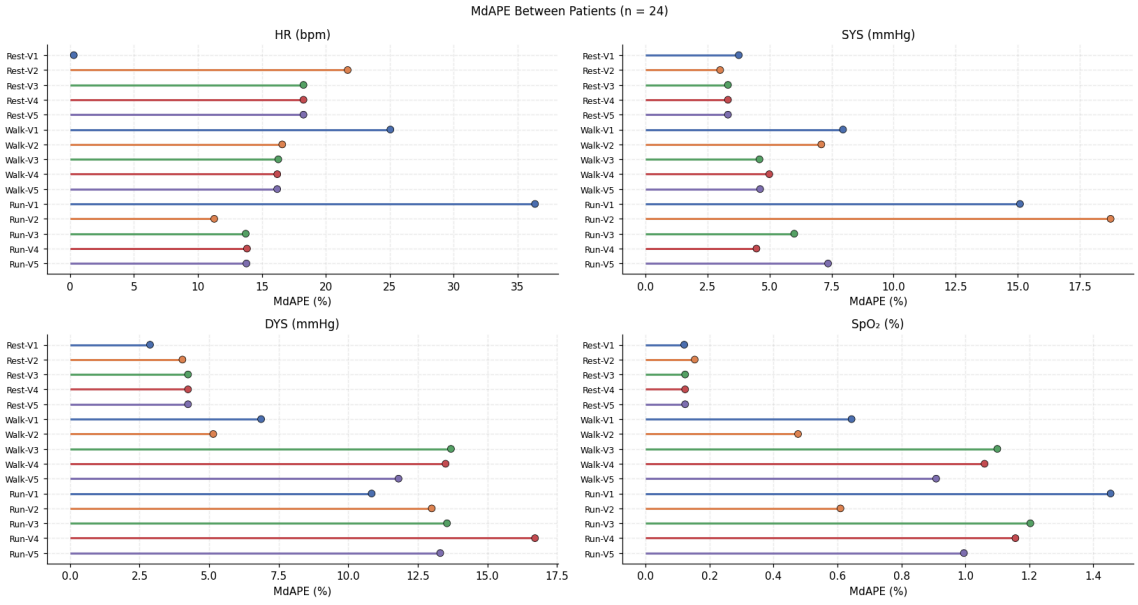

Figure 7.11: MdAPE Between Patients by Model (V1 - V5).

In Figure 7.11, the presence of outlier values becomes particularly evident for the HR metric, especially in Version 1. Similar outlier behavior can also be observed in other metrics, such as SYS and SpO_2. Therefore, these results further strengthen the argument

that the modifications implemented in the software produced more accurate results when compared with the original version.

7.3. Protocol B: Hemodynamic and Respiratory Responses

The first phase of validation Section 7.2 assessed the hemodynamic and respiratory responses across different versions of the software during the transition from rest to low-intensity exercise. However, this validation alone is insufficient for a human digital twin intended for non clinical applications.

7.3.1. Methodology

To address this limitation, Protocol B was designed to reproduce a cardiopulmonary exercise test based on the Bruce protocol [61], and to compare the simulated results with the Metabolic Equivalent Task (MET) values for each patient, using two independent reference sources from the literature. Protocol B employs two distinct equations to estimate the expected MET values. This independence between reference models is methodologically important, as it allows for a more robust and unbiased evaluation of the simulation outcomes. Therefore, the primary objective of this protocol is to address the following research questions:

- Does the BioGears software generate the expected peak MET values?
- Is the inter-stage MET response produced by the simulator consistent with reference data?
- Does the choice of reference model (anthropometric-based vs. age-based) influence the conclusions regarding the accuracy of the software?

The protocol consists of seven stages, each lasting three minutes, with incremental increases in both speed and inclination. The stage parameters presented in the Table 7.2 are not directly derived from the original Bruce protocol [61], but instead from [62]. Consequently, the test follows an incremental structure and is terminated when the subject reaches fatigue.

Stage	Speed (mph)	Speed (m/s)	Inclination (%)	Expected MET
1	1.7	0.760	10	5
2	2.5	1.118	12	7
3	3.4	1.520	14	10
4	4.2	1.878	16	13
5	5.0	2.235	18	15
6	5.5	2.459	20	18
7	6.0	2.682	22	20

Table 7.2: Expected MET

The primary reference used for this protocol is [63], in which $n = 1067$ subjects were recruited to perform the Bruce protocol. The cohort consisted of 547 women and 520 men. In this study, the mean age was 41.6 years, and the peak MET was reported as 11.6 ∓ 2.1. Additionally, multivariate regression equations were proposed for the prediction of MET values;

$$MET_F = 17.502 - 0.062 \times age[y] - 0.100 \times BMI[kg.m]^{-2} - 0.014 \times WC[cm] \quad (7.9)$$

$$MET_M = 19.379 - 0.041 \times age[y] - 0.122 \times BMI[kg.m]^{-2} - 0.023 \times WC[cm] \quad (7.10)$$

In these equations, $WC[cm]$ represents the abdominal circumference. Additionally, to provide a complementary and simpler reference model based solely on age and sex, the equations from [64] for women and from [65] for men were also considered.

$$MET_{\text{Women, age-only}} = 14.7 - 0.13 \times age[y] \quad (7.11)$$

$$MET_{\text{Men, age-only}} = 18.0 - 0.15 \times age[y] \quad (7.12)$$

For the validation and comparison, the outputs from BioGears were analyzed using two different methods for MET estimation. The first method is based on the patient's oxygen consumption rate, such that:

$$1MET = 3.5ml.kg^{-1}.min^{-1} \quad (7.13)$$

Therefore, once that we have the values in the BioGears, its only necessary to do the conversion;

$$MET = \frac{VO_2}{3.5 \times kg} \quad (7.14)$$

The second method consists of estimating MET values from the HR output, using the predictive equations proposed in [66].

$$EE_{\text{HR,men}} = \frac{-55.0969 + 0.6309 \times HR + 0.1988 \times W + 0.2017 \times age}{4.184} \tag{7.15}$$

$$EE_{\text{HR, woman}} = \frac{-20.4022 + 0.4472 \times HR - 0.1263 \times W + 0.074 \times age}{4.184} \tag{7.16}$$

Considering HR in beats per minute (bpm), body mass in kilograms (kg), and age in years (y), the MET estimation is obtained through the following equation:

$$MET = \frac{EE_{\text{HR}} \times 60}{W} \tag{7.17}$$

7.3.2. Pipeline

In the simulations, a total of 24 patients were considered. All patients completed every stage defined in Table 7.2. The composition of the cohort was as follows:

- 2 anonymous templates
- 2 default templates
- 4 female subjects distributed across different ages (18, 30, and 40 years)
- 16 male subjects across a broader age range

This composition was selected to approximate the demographic envelope of the population used in the derivation of the Bruce protocol, while also enabling the evaluation of the simulator's response across a range of physiological parameters.

From Version 2 onward, each patient was associated with an individual `.cfg` file. These files were generated a priori using the `prediction_v6.py` script and contain individualized coefficients for each subject, as described previously. Prior to each simulation, the main C++ driver reads the corresponding configuration file and applies the personalized parameters. In contrast, the original implementation (Version 1) relies on predefined parameters shared across all patients.

The simulation pipeline can be described as follows:

1. For each patient, the C++ driver first performs an initialization step, during which the `.cfg` file is loaded and the output structures are created.
2. This is followed by a baseline period of 5 minutes. After the baseline phase, the Bruce protocol is initiated.

3. Each stage of the protocol is implemented as an instance of `SEExercise::SERunning`.
4. Upon completion of all stages, a recovery phase is introduced using `SEExercise::SEGeneric{Intensity = 0.0}`, effectively removing the exercise stimulus and allowing the analysis of the post-exercise response.

7.3.3. Results

After completing the simulations for all five versions of the software, post-processing was performed for each patient and for each version. First, the temporal signal was segmented into the seven stages of the protocol. For each stage, both the mean and peak MET values were computed.

Subsequently, for each patient and each version, the simulated results were compared against the two reference models described previously. All computed metrics were then aggregated into a structured dataframe, containing the results for every version, to enable further analysis and comparison across the different models.

The first analysis was relatively straightforward. Based on the values defined in Table 7.2, the expected MET for each stage was compared with the corresponding values produced by the software. To ensure consistency, MET was calculated using the two reference models previously described. The results can be seen in Figure 7.12.

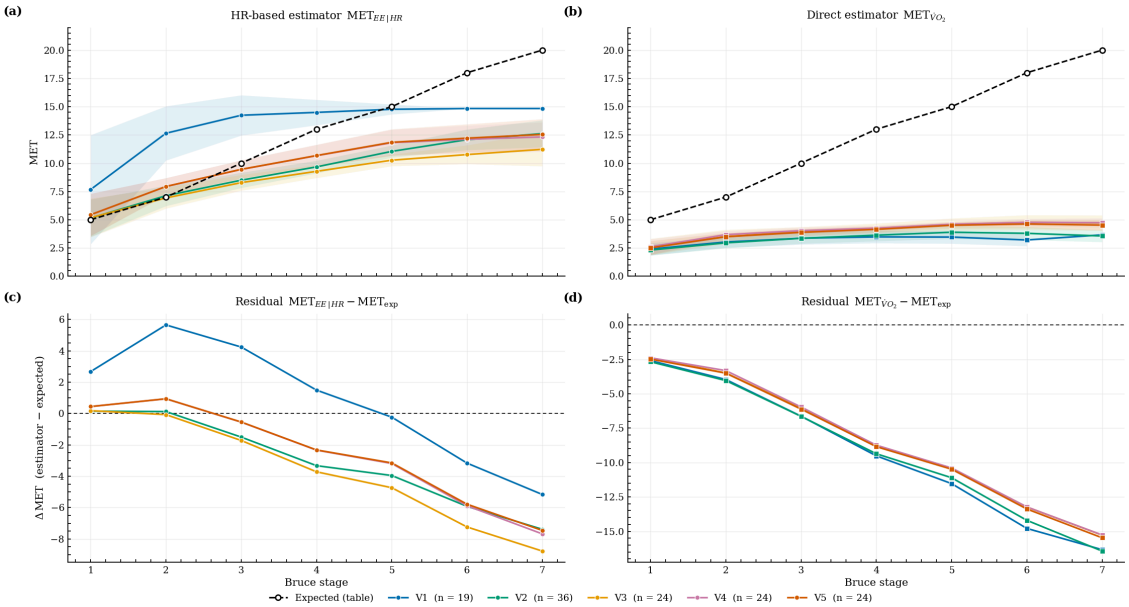

Figure 7.12: (a) HR-based estimator (b) Direct estimator. (c)–(d) Stage-wise residuals against the reference. The HR-based estimator converges from V1 (n = 19) towards a stable response in V4–V5 (n = 24).

Across all five versions, the MET estimates derived from VO_2 exhibit saturation in the range of approximately 3.5 to 5 METs, largely independent of the workload.

Already in Stage 1 (expected MET ≈ 5), the simulated values fall below the reference, and beyond Stage 3 the response becomes nearly flat, with minimal variation across subsequent stages. This behavior is characteristic of the submodel `Energy::CalculateMetabolicRateProduction()`. In this implementation, the `OxygenConsumptionRate` is derived from the `TotalMetabolicRate`, which is effectively constrained by the patient-specific `MaximumWorkRate`. As a result, once this limit is reached, further increases in workload do not translate into higher oxygen consumption, leading to the observed saturation.

In contrast, the MET values predicted using the HR based estimator show a more realistic trend. Considering the modifications introduced in the cardiovascular control mechanisms, particularly in the HR dynamics, the HR based estimates exhibit improved agreement with the expected progression across stages.

Subsequently, additional statistical analyses were performed to quantify the error across the different software versions. In particular, the MAE and the RMSE were computed to evaluate the accuracy of the models.

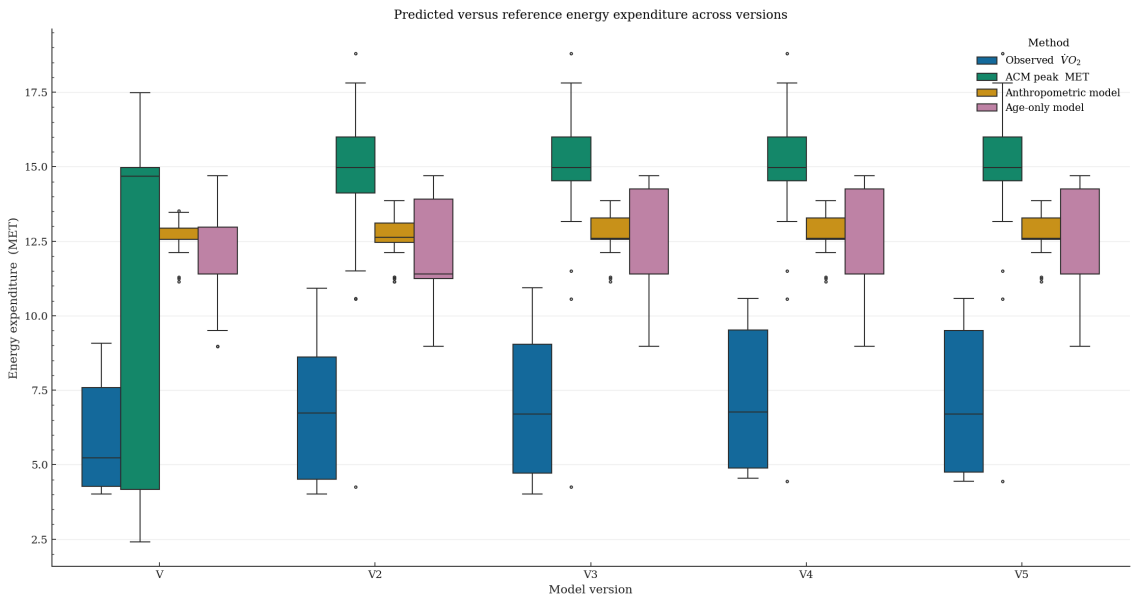

Figure 7.13: Predicted vs. Reference Energy Expenditure Across Versions.

Considering the Figure 7.13, it is possible to note that peak MET_{ACM} is not a BioGears output but the calorimetric reference for the highest Bruce stage attained, computed from the Equation (7.15), Equation (7.16) and Equation (7.17). Conversely, observed VO_2 is the engine reported peak VO_2 converted to MET equivalents and remains capped at ≈ 5 METs across all five versions, in agreement with the per-stage analysis Figure 7.12.

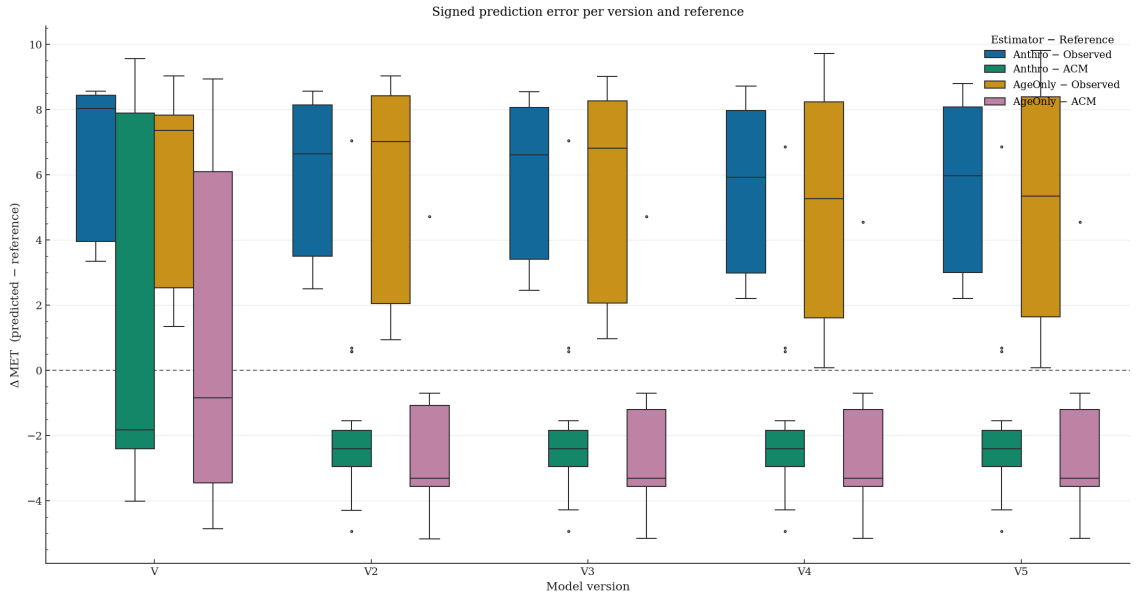

Figure 7.14: Signed Prediction Error per Version and Reference.

In Figure 7.14, it is observed that, relative to the engine's Observed VO_2 (blue and orange distributions), both HR based estimators exhibit a systematic positive bias of approximately $+6$ MET across Versions $V2$ - $V5$. The inter quartile ranges extend roughly from $+3$ to $+8$ MET. This persistent overestimation quantifies, on a per-patient basis, the saturation effect of the BioGears metabolic submodel.

In contrast, when considering the ACM peak reference (green and pink distributions), both estimators reverse this tendency, with values concentrated around $\Delta MET \approx -2.5$ MET for the anthropometric model and ≈ -3 MET for the age-only model. This indicates that, at the highest exercise stage achieved, both estimators slightly underestimate the caloric demand

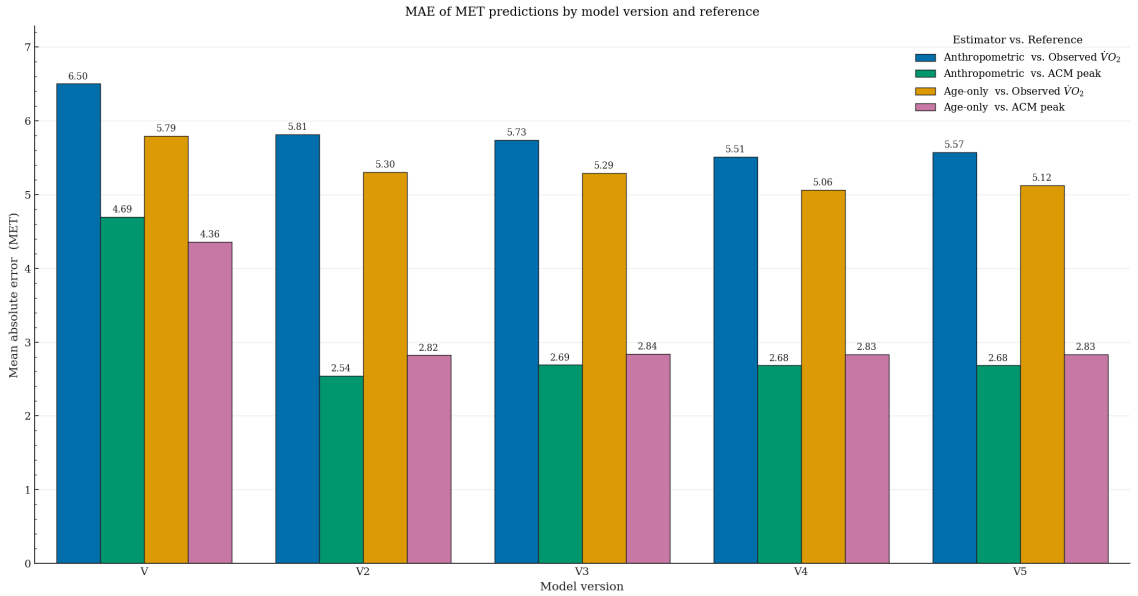

Figure 7.15: MAE of MET Predictions by Model Version and Reference.

Finally, in Figure 7.15, version $V1$ appears as a clear outlier, with MAE values ranging between 4.36 and 6.50 MET, primarily driven by the unstable transient behavior of the HR response. In contrast, versions $V2$ – $V5$ converge to nearly identical values, approximately 5.5 – 5.8 MET against the observed VO_2 reference and approximately 2.7 – 2.8 MET against the ACM peak reference. This reduction reflects a decreased discrepancy between the simulator outputs and the reference models, and highlights the remaining systematic deviation between the saturated metabolic sub model and the calorimetric benchmark.

Quantitatively, $V2$ achieves the lowest MAE of 2.54 MET (RMSE = 2.833, r = 0.790) against the ACM reference, while $V4$ and $V5$ yield practical identical results, confirming that the modifications introduced after $V3$ do not alter the peak metabolic output substantially. In contrast, $V1$ exhibits a substantially higher error (MAE = 4.69, RMSE = 5.52) and a reversed systematic bias ($+1.79$ MET vs. ≈ -2.0 MET for $V2$ – $V5$), consistent with the unstable HR transient response observed in the per stage analysis. The transition from $V1$ to $V2$ therefore represents a 45.8% reduction in MAE, driven primarily by the stabilization of the cardiovascular control sub model.

The absence of improvement beyond $V3$ can be attributed to the nature of the HR based estimator and the stabilization of the cardiovascular sub model, since the estimator relies fundamentally on HR, as can be verified by Equation (7.15) and Equation (7.16). Indeed, HR_{peak} across $V3$ - $V5$ varies by less than 4 bpm on average, confirming that no further cardiovascular improvement propagates into the peak metabolic estimates.

The remaining systematic bias of approximately -2.0 MET therefore reflects a structural limitation of the metabolic sub model saturation rather than a failure of the cardiovascular control.

7.4. Protocol C: Cold Face Test Validation

In addition to the use cases described in the previous sections, the CFT was implemented and adopted as the validation protocol for version 5 (V5). This protocol is exclusive to V5 because the CFT is not a native action in BioGears, it had to be implemented from scratch as part of this work, with the full action class and the corresponding engine side hooks integrated into the simulation pipeline and exposed via the `SEColdFaceTest` API. Calibration of the action was performed against the experimental CFT recordings from the Twin Approximation cohort, which provides CFT data across a diverse set of subjects.

This protocol try to directly addresses Use Case 2, Section 3.3.2, in which control room operators are subject to prolonged cognitive stress.

7.4.1. Methodology

For this protocol, the study by [6] was used as the reference framework to reproduce the experimental design described in the original paper. In their protocol, an initial global baseline measurement was performed to determine the baseline HR. Subsequently, for each MIST condition, with $\mathrm{MIST} \in [1, 2, 3]$, participants were exposed to a mental arithmetic challenge implemented through a computer program with a graphical user interface, while simultaneously being subjected to social-evaluative pressure from a human investigator monitoring their performance.

Since this experimental setup could not be reproduced exactly within the simulation environment, the `SEAcuteStress` condition available in the software was employed to emulate the stress response induced by the computerized task and social-evaluative threat.

For each virtual patient, the entire procedure was conducted twice. The first condition corresponded to the ‘Control’ group, in which no CFT was applied. The second condition corresponded to the CFT group, in which the Cold Face Test was applied. This dual group design was adopted to remain consistent with the methodology described in the original study, which also included separate Control and CFT groups. Furthermore, the use of both groups provides a more robust framework for the a posteriori validation and comparative analysis of the simulated physiological responses.

7.4.2. Results

Considering the methodology described above, and using the complete dataset from the $N = 24$ virtual patients, a comprehensive analysis and validation procedure was conducted to evaluate whether the simulated responses were consistent with the expected physiological behavior.

First, Figure 7.16 was generated, presenting the variation in the HR (ΔHR) in $[\%]$. This figure was produced as an attempt to cross-validate the results against Figure 7.17 reported in the original study [6].

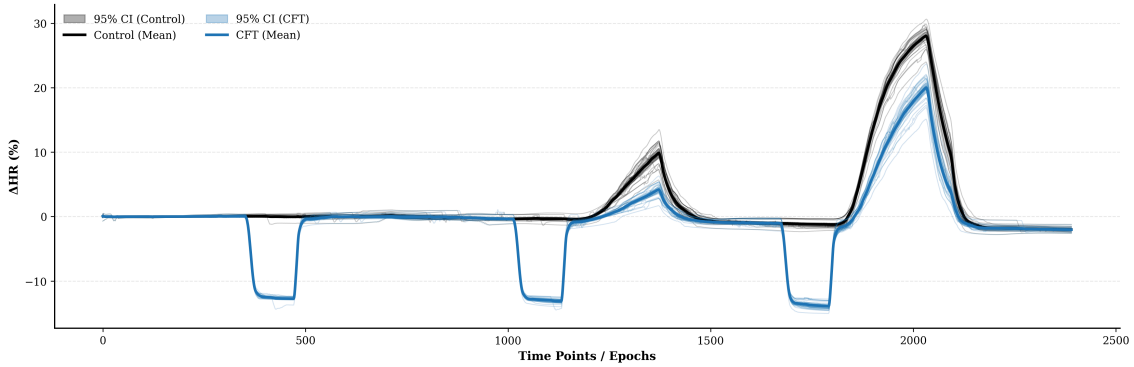

Figure 7.16: Control and CFT During the Entire Protocol.

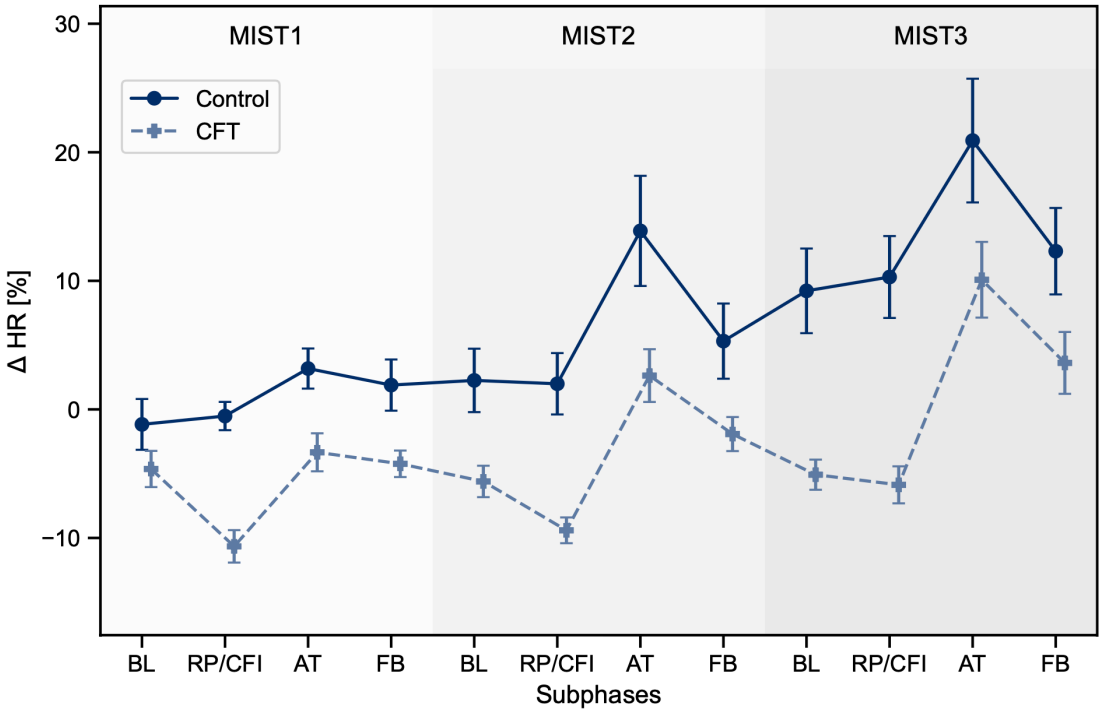

Figure 7.17: Average heart rate per MIST subphase during the conduction of the MIST. Reproduced from [6].

In Figure 7.16, it is possible to observe the data from both the Control and CFT conditions. From these results, it is evident that the CFT condition presents lower values of ΔHR. During the CFT periods, occurring at approximately $[400, 1100, 1700]$, a pronounced decrease in HR can be observed, which is consistent with the expected physiological response to this protocol. Furthermore, after these events, the HR values under the CFT condition remain lower than those of the Control condition, as expected.

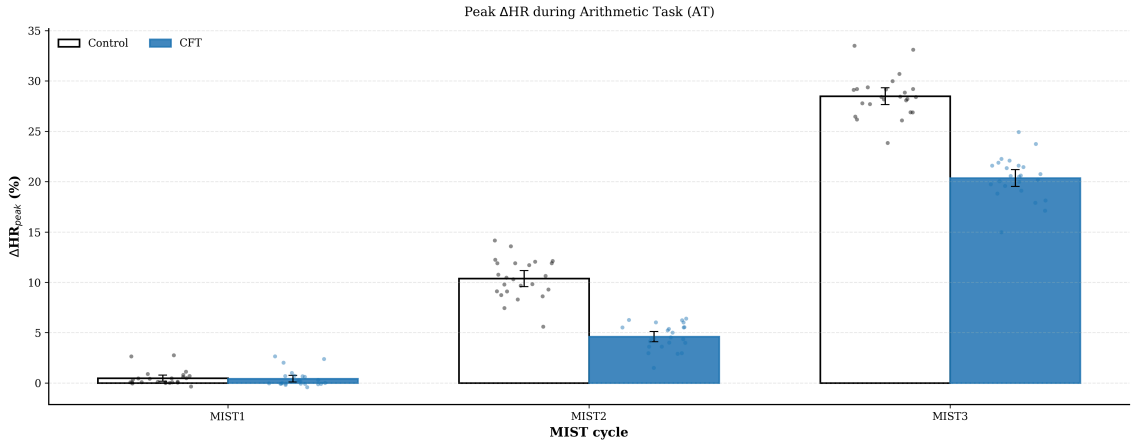

Figure 7.18: Peak Delta HR during Arithmetic Task.

In Figure 7.18, the boxplots demonstrate that the CFT values are lower than the Control values, with this effect being more pronounced in MIST2 and MIST3. In contrast, during MIST1, only minimal changes can be observed. Furthermore, through additional analysis and verification, it was possible to observe that the peak bradycardia was reached after approximately 30 s, with a HR decrease of about 20%, which is consistent with the findings reported by [58, 6].

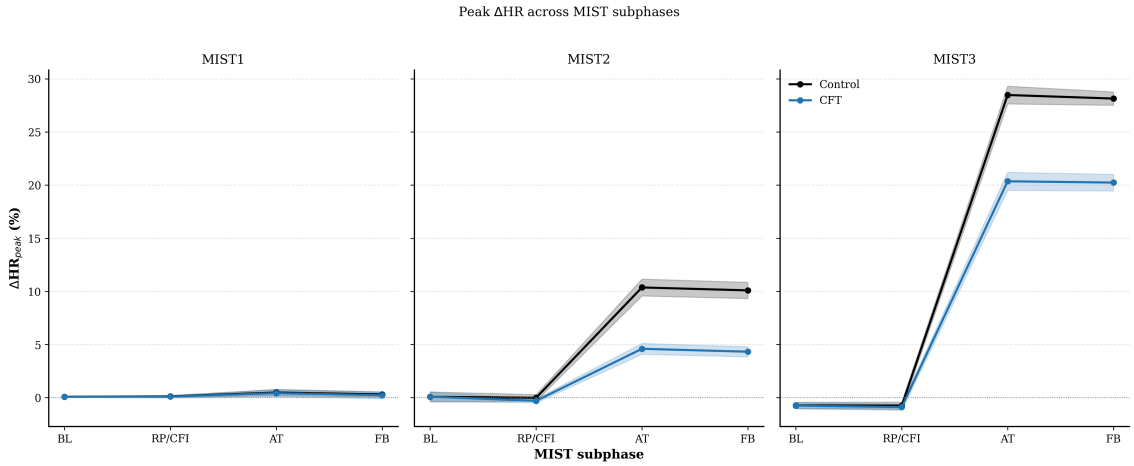

Figure 7.19: Peak HR Across MIST Subphases.Each consisting of Baseline (BL), Resting Period / Cold Face Intervention (RP/CFI), Arithmetic Tasks (AT), i.e stress application, and Feedback (FB) subphases.

In Figure 7.19, it is possible to observe that the values of $\Delta HR_{\text{peak}}\,(\%)$ increase progressively with the MIST level, with MIST 3 presenting higher values than MIST 1. This behavior occurs because the stress intensity is greater in MIST 3. Since the CFT condition remains identical across all MIST stages, the results indicate that CFT reduces the $\Delta HR_{\text{peak}}\,(\%)$ proportionally to the intensity of the applied stress.

It was expected that the values of $\Delta HR[\%]$ would decrease as the protocol progressed. This behavior can be explained by physiological adaptation: after the first and second MIST exposures, the body becomes progressively accustomed to the stimulus and, consequently, exhibits a less pronounced response compared to earlier exposures [67]. The increasing predictability of the stimulus reduces the progressive sympathetic reflex activation, thereby attenuating the cardiovascular response in subsequent events.

However, the BioGears engine, being a mathematical model without short or long term memory mechanisms, is unable to reproduce this adaptive behavior. This represents a limitation of the software. In its current implementation, the baroreflex is modelled as a PID controller that returns the same steady-state setpoint for each event, regardless of prior exposures. As a result, the physiological response remains identical for repeated identical stimuli. This absence of memory in the model leads to a gap in the simulation capability, since it prevents the representation of habituation effects and time dependent physiological adaptation observed in real human responses.

Therefore, considering the results and the statistical analysis, the Paired t-tests revealed a significant reduction of peak ΔHR during the cold face intervention for all three MIST cycles (all $p < 0.001$), confirming the immediate cardiac vagal response.

8. Conclusion

In this work, a software system for simulating virtual human patients under non clinical scenarios was developed. In the firsts chapters, the state of the art was reviewed, along with the identification of existing gaps and limitations. Subsequently, the first outcome was introduced: the Twin Approximation based on real data. This was followed by modifications to the BioGears engine and, finally, the creation of a Personalized Human Digital Twin using real-world data.

With the PhysTwin software and the real-time user interface now fully developed, it is possible to identify and discuss the main limitations of the proposed framework, as well as outline potential directions for future work.

BioGears was designed and validated for acute physiological scenarios on the order of minutes to hours. Long-term homeostatic regulation, including nutritional metabolism, sleep physiology, and chronic renal fluid balance, falls outside the engine's validated operating envelope. The 24-hour scheduling architecture proposed here, in Section 6.4 is therefore intended as a framework for sequencing non clinical perturbation scenarios across a daily timeline, not as a claim that the underlying engine sustains full 24-hour physiological validity.

Considering these limitations, an initial mitigation strategy was implemented through the integration of the `SESleep` action. This deliberate intervention was motivated by the fact that `SESleep` provides a binary on/off patient action that allows the BioGears Nervous System to track accumulated sleep and wake durations.

However, the `SESleep` action does not interact with the cardiovascular, respiratory, or thermoregulatory systems. Consequently, physiological variables such as HR, arterial pressure, ventilation, core and skin temperature set points, and shivering thresholds remain entirely independent of the sleep state. While the action therefore offers a physiologically grounded mechanism for representing the accumulation and recovery of sleep debt, it should not be regarded as a substitute for a validated model of nocturnal physiology.

As a result, the incorporation of `SESleep` into the 24-hour diary framework should be interpreted as a targeted extension of the engine's cognitive and fatigue-related representations, rather than as a comprehensive simulation of the physiological processes associated with sleep.

Finally, a complete 24-hour physiological simulation would require the addition of modules for nutritional intake and hepatic glucose regulation. These represent natural extensions of the present work and are identified as future development directions.

As future work, further improvements to both the software and the physiological model can be pursued. One immediate opportunity for enhancement is the use of the existing software framework to execute different simulation scenarios and evaluate how the newly introduced variables affect the model outputs. By systematically adjusting the parameter values and analyzing their impact on representative test cases, it is possible to empirically identify configurations that yield more accurate and physiologically plausible results. Consequently, certain aspects of the model can be improved without requiring structural modifications to the software, but rather through the refinement and calibration of existing parameters.

Another important direction for future work is the incorporation of additional use cases, particularly those involving environmental conditions. Extending the model to account for external factors, such as extreme heat or extreme cold, would represent a significant advancement in the capabilities of the HDT. Such functionality would enable users to investigate how different environmental conditions influence the physiological response of a virtual patient during a specific simulation scenario, like the case of agriculture works.

However, the implementation of this feature requires a comprehensive study to determine how environmental effects can be represented mathematically and integrated into the existing physiological framework. Once the underlying models have been established, further development will be necessary to modify the software architecture and ensure that the simulation accurately reflects the desired environmental influences on the patient.

Additionally, for the future addition of data to the database, it is important to establish a robust approach to assess the quality of the data after preprocessing. It is necessary to define minimum quality thresholds to ensure the reliability of the data being incorporated into the database, considering that these data are used to estimate the model parameters.

References

[1] Brant H. Tudor, Ryan Shargo, Geoffrey M. Gray, Jamie L. Fierstein, Frederick H. Kuo, Robert Burton, Joyce T. Johnson, Brandi B. Scully, Alfred Asante-Korang, Mohamed A. Rehman, and Luis M. Ahumada. “A Scoping Review of Human Digital Twins in Healthcare Applications and Usage Patterns”. In: *npj Digital Medicine* 8 (2025).

[2] Chenyu Tang, Wentian Yi, Edoardo Occhipinti, Yanning Dai, Shuo Gao, and Luigi G. Occhipinti. “A roadmap for the development of human body digital twins”. In: *Nature Reviews Electrical Engineering* 1.3 (February 2024). pp. 199–207.

[3] Austin J Baird, Matthew Mcdaniel, Steven A White, and Nathan Tatum Marin. “BioGears: A C++ library for whole body physiology simulations”. In: *The Journal of Open Source Software* (2020).

[4] Biogears Engine. *BioGears Cardivascular Circuit*. 2018.

[5] Biogears Engine. *BioGears Showcase*. 2018.

[6] Robert Richer, Janis Zenkner, Arne Küderle, Nicolas Rohleder, and Bjoern M. Eskofier. “Vagus activation by Cold Face Test reduces acute psychosocial stress responses”. In: *Scientific Reports* 12.1 (November 2022). p. 19270.

[7] Bergthor Björnsson, Carl Borrebaeck, Nils Elander, Thomas Gasslander, Danuta R. Gawel, Mika Gustafsson, Rebecka Jörnsten, Eun Jung Lee, Xinxiu Li, Sandra Lilja, David Martínez-Enguita, Andreas Matussek, Per Sandström, Samuel Schäfer, Margaretha Stenmarker, X. F. Sun, Oleg Sysoev, Huan Zhang4, and Mikael Benson. “Digital twins to personalize medicine”. In: *Genome Med* (2020).

[8] Yujia Lin, Liming Chen3, Aftab Ali, Christopher Nugent, Ian Cleland, Rongyang Li, Jianguo Ding, and Huansheng Ning. “Human digital twin: a survey”. In: *Journal of Cloud Computing* (2024).

[9] Marco Viceconti and Gordon Clapworthy. “THE VIRTUAL PHYSIOLOGICAL HUMAN: CHALLENGES AND OPPORTUNITIES”. In: *IEEE* (2006).

[10] Biogears Engine. *Biogears Engine Website*. 2018.

[11] Kitware. *Pulse Physiology Engine*. 2025.

[12] Physiome Project. *Physiome Project*. n.d.

[13] Marco Viceconti, Maarten De Vos, Sabato Mellone, and Liesbet Geris. “From the digital twins in healthcare to the Virtual Human Twin: a moon-shot project for digital health research”. In: *IEEE J Biomed Health Inform* (2023).

[14] Manlio De Domenico, Luis M. Rocha, Luca Allegri, Guido Caldarelli, Valeria d'Andrea, Barbara Di Camillo, Jordan Rozum1 Riccardo Sbarbati, and Francesco Zambelli. "Challenges and opportunities for digital twins in precision medicine from a complex systems perspective". In: *npj Digital Medicine* 8 (2025).

[15] Matthew McDaniel, Jennifer Carter, Jonathan M Keller, Steven A. White, and Austin Baird. "Open Source Pharmacokinetic/Pharmacodynamic Framework: Tutorial on the BioGears Engine". In: *ASCPT* (2018).

[16] COMSOL. *Lumped Model of a Human Body*. Online PDF. 2024.

[17] XUN LIU, CHENGHUAI MO, JIAXING LI, HONGYI YU, SHENG HU, DA ZHU, PUMING ZHANG, and YONGHUA LI. "Development of a Lumped Parameter Model of Human Whole Body Circulatory Loop". In: *IEEE* (2024).

[18] Fabrizio Gaudenzi and Alberto P. Avolio. "Lumped Parameter Model of Cardiovascular-Respiratory Interaction". In: *IEEE* (2013).

[19] Mubaris Nadeem, Sascha Kostic, Mareike Dornhöfer, Christian Weber, and Madjid Fathi. "A comprehensive review of digital twin in healthcare in the scope of simulative health-monitoring". In: *Digital Health* 11 (2025).

[20] Matthew McDaniel, Jonathan M. Keller, Steven White, and Austin Baird. "A Whole-Body Mathematical Model of Sepsis Progression and Treatment Designed in the BioGears Physiology Engine". In: *Front. Physiol* (2019).

[21] Matthew McDaniel and Austin Baird. "A Full-Body Model of Burn Pathophysiology and Treatment Using the BioGears Engine". In: *PubMed* (2019).

[22] Rachel B Clipp, Aaron Bray, Rodney Metoyer, M Cameron Thames, and Jeffrey B Webb. "Pharmacokinetic and pharmacodynamic modeling in BioGears". In: *PubMed* (2016).

[23] Bray A, Webb J.B, and Enquobahrie A al. "Pulse Physiology Engine: an Open-Source Software Platform for Computational Modeling of Human Medical Simulation". In: *Springer Nature* (2019).

[24] Juul Achten and Asker E Jeukendrup. "Heart Rate Monitoring: Applications and Limitations". In: *Sports Medicine* 33.7 (2003). pp. 517–538.

[25] Ahmed Alsheikhy, Yahia F. Said, Tawfeeq Shawly, and Husam Lahza. "A Model to Predict Heartbeat Rate Using Deep Learning Algorithms". In: *Healthcare* 11.3 (January 2023). p. 330.

[26] A. K. S. Saranya and T. Jaya. “Early Detection of Heartbeat from Multimodal Data Using RPA Learning with KDNN-SAE”. In: *Computer Systems Science and Engineering* 45.1 (2023). pp. 545–562.

[27] Eko Sakti Pramukantoro and Akio Gofuku. “A Heartbeat Classifier for Continuous Prediction Using a Wearable Device”. In: *Sensors* 22.14 (July 2022). p. 5080.

[28] Saad Irfan, Nadeem Anjum, Turke Althobaiti, Abdullah Alhumaidi Alotaibi, Abdul Basit Siddiqui, and Naeem Ramzan. “Heartbeat Classification and Arrhythmia Detection Using a Multi-Model Deep-Learning Technique”. In: *Sensors* 22.15 (July 2022). p. 5606.

[29] Joseph I. Sirven. “Epilepsy: A Spectrum Disorder”. In: *Cold Spring Harbor Perspectives in Medicine* 5.9 (September 2015). p. a022848.

[30] Kenneth D. Laxer, Eugen Trinka, Lawrence J. Hirsch, Fernando Cendes, John Langfitt, Norman Delanty, Trevor Resnick, and Selim R. Benbadis. “The consequences of refractory epilepsy and its treatment”. In: *Epilepsy & Behavior* 37 (August 2014). pp. 59–70.

[31] Yousif A. Saadoon, Mohamad Khalil, and Dalia Battikh. “Machine and Deep Learning-Based Seizure Prediction: A Scoping Review on the Use of Temporal and Spectral Features”. In: *Applied Sciences* 15.11 (June 2025). p. 6279.

[32] Prabhat Kumar Upadhyay, Priyaranjan Kumar, Manoj Kumar Panda, and Aswini Kumar Samantaray. “Seamless integration for enhanced seizure prediction using HybridConvMobileNet on Typhoon HIL”. In: *Scientific Reports* (November 2025).

[33] Ranjan Jana and Imon Mukherjee. “Efficient Seizure Prediction and EEG Channel Selection Based on Multi-Objective Optimization”. In: *IEEE Access* 11 (2023). pp. 54112–54121.

[34] Boxuan Wei, Lu Xu, and Jicong Zhang. “A Compact Graph Convolutional Network With Adaptive Functional Connectivity for Seizure Prediction”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 32 (2024). pp. 3531–3542.

[35] Bhaskar Kapoor and Bharti Nagpal. *Hybrid Cuckoo Finch Optimization based Machine Learning Classifier for Seizure Prediction Using EEG Signals in IoT Network*. June 2022.

[36] Pratibha S Sonawane and Jagdish B. Helonde. “Smart Societal Optimization-based Deep Learning Convolutional Neural Network Model for Epileptic Seizure Prediction”. In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 12.1 (December 2024). p. 2280551.

[37] Fatemeh Sarani Rad, Rasha Hendawi, Xinyi Yang, and Juan Li. “Personalized Diabetes Management with Digital Twins: A Patient-Centric Knowledge Graph Approach”. In: *Journal of Personalized Medicine* 14.4 (March 2024). p. 359.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *ACL Anthology* (n.d.).

[39] Francisco Jesus Lara-Abelenda, David Chushig-Muzo, Pablo Peiro-Corbacho, Ana M. Wägner, and Cristina Soguero-Ruiz. *Personalized Glucose Forecasting for People with Type 1 Diabetes Using Large Language Models*. 2024.

[40] Nikita Makarov, Maria Bordukova, Raul Rodriguez-Esteban, Fabian Schmich, and Michael P. Menden. *Large Language Models forecast Patient Health Trajectories enabling Digital Twins*. July 2024.

[41] Eirini Kostakou, Evangelos Kaniaris, Effrosyni Filiou, Ioannis Vasileiadis, Paraskevi Katsaounou, Eleni Tzortzaki, Nikolaos Koulouris, Antonia Koutsoukou, and Nikoletta Rovina. “Acute Severe Asthma in Adolescent and Adult Patients: Current Perspectives on Assessment and Management”. In: *Journal of Clinical Medicine* 8.9 (August 2019). p. 1283.

[42] Victor L. Pinto and Adedayo Adeyinka. *Increased Intracranial Pressure*. Updated September 14, 2025. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2026 Jan–. 2025.

[43] Hirofumi Tanaka, Kevin D Monahan, and Douglas R Seals. “Age-predicted maximal heart rate revisited”. In: *Journal of the American College of Cardiology* 37.1 (January 2001). pp. 153–156.

[44] Tilen Hliš, Iztok Fister, and Iztok Fister Jr. “Digital twins in sport: Concepts, taxonomies, challenges and practical potentials”. In: *Expert Systems with Applications* 258 (December 2024). p. 125104.

[45] Barbara Rita Barricelli, Elena Casiraghi, Jessica Gliozzo, Alessandro Petrini, and Stefano Valtolina. “Human Digital Twin for Fitness Management”. In: *IEEE Access* 8 (2020). pp. 26637–26664.

[46] Bharatwaajan Balaji, Mohammed Aatif Shahab, Babji Srinivasan, and Rajagopalan Srinivasan. “ACT-R based human digital twin to enhance operators’ performance in process industries”. In: *Frontiers in Human Neuroscience* 17 (February 2023). p. 1038060.

[47] Saul Davila-Gonzalez and Sergio Martin. “Human Digital Twin in Industry 5.0: A Holistic Approach to Worker Safety and Well-Being through Advanced AI and Emotional Analytics”. In: *Sensors* 24.2 (January 2024). p. 655.

[48] Susana Rodrigues, Joana S. Paiva, Duarte Dias, Marta Aleixo, Rui Filipe, and João Paulo Silva Cunha. “A Wearable System for the Stress Monitoring of Air Traffic Controllers During An Air Traffic Control Refresher Training and the Trier Social Stress Test: A Comparative Study”. In: *The Open Bioinformatics Journal* 11.1 (May 2018). pp. 106–116.

[49] AI4REALNET. *AI For real-World Network Operation*. n.d.

[50] DB Browser SQLite. *DB Browser for SQLite*. n.d.

[51] MIT Laboratory Computational Physiology. *PhysioNet*. n.d.

[52] Denis Mongin, Jeronimo García Romero, and Jose Ramon Alvero Cruz. “Treadmill Maximal Exercise Tests from the Exercise Physiology and Human Performance Lab of the University of Malaga”. In: *PhysioNet* (December 2021). Version 1.0.1.

[53] Katsuji Imai, Hideyuki Sato, Masatsugu Hori, Hideo Kusuoka, Hitoshi Ozaki, Hiroshi Yokoyama, Hiroshi Takeda, Michitoshi Inoue, and Takenobu Kamada. “Vagally mediated heart rate recovery after exercise is accelerated in athletes but blunted in patients with chronic heart failure”. In: *Journal of the American College of Cardiology* 24.6 (November 1994). pp. 1529–1535.

[54] Tiago Peçanha, Natan Daniel Silva-Júnior, and Cláudia Lúcia De Moraes Forjaz. “Heart rate recovery: autonomic determinants, methods of assessment and association with mortality and cardiovascular diseases”. In: *Clinical Physiology and Functional Imaging* 34.5 (September 2014). pp. 327–339.

[55] Javier T. Gonzalez, Cas J. Fuchs, James A. Betts, and Luc J. C. Van Loon. “Liver glycogen metabolism during and after prolonged endurance-type exercise”. In: *American Journal of Physiology-Endocrinology and Metabolism* 311.3 (September 2016). pp. E543–E553.

[56] Paul B. Gastin. “Energy System Interaction and Relative Contribution During Maximal Exercise:” In: *Sports Medicine* 31.10 (2001). pp. 725–741.

[57] Martha E. Heath and John A. Downey. “The cold face test (diving reflex) in clinical autonomic assessment: methodological considerations and repeatability of responses”. In: *Clinical Science* 78.2 (February 1990). pp. 139–147.

[58] Ramesh K. Khurana. “Cold face test: adrenergic phase”. In: *Clinical Autonomic Research* 17.4 (June 2007). pp. 211–216.

[59] HiveMQ. *HIVEMQ*. 2026.

[60] Eyad Talal Attar. “Physiological responses to physical activity: A multi-sensor wearable study across activity intensity, age, gender, body weight, and BMI variability”. In: *Medicine* 104.40 (October 2025). p. e44725.

[61] R.A. Bruce, F. Kusumi, and D. Hosmer. “Maximal oxygen intake and nomographic assessment of functional aerobic impairment in cardiovascular disease”. In: *American Heart Journal* 85.4 (April 1973). pp. 546–562.

[62] M. Jetté, K. Sidney, and G. Blümchen. “Metabolic equivalents (METS) in exercise testing, exercise prescription, and evaluation of functional capacity”. In: *Clinical Cardiology* 13.8 (August 1990). pp. 555–565.

[63] Özge Çetinarslan. “Nomogram-Based Evaluation of Exercise Capacity Based on Metabolic Equivalents (METs), Age, Body Mass Index(BMI), and Waist Circumference”. In: *Acıbadem Üniversitesi Sağlık Bilimleri Dergisi* 16.4 (October 2025). pp. 540–550.

[64] Martha Gulati, Morton F Arnsdorf, Thomas H Marwick, and Roxanne H Wicklund. “The Prognostic Value of a Nomogram for Exercise Capacity in Women”. In: *n engl j med* (2005).

[65] Charles K. Morris, Jonathan Myers, Victor F. Froelicher, Takeo Kawaguchi, Kenji Ueshima, and Alisa Hideg. “Nomogram based on metabolic equivalents and age for assessing aerobic exercise capacity in men”. In: *Journal of the American College of Cardiology* 22.1 (July 1993). pp. 175–182.

[66] Lr Keytel, Jh Goedecke, Td Noakes, H Hiiloskorpi, R Laukkanen, L Van Der Merwe, and Ev Lambert. “Prediction of energy expenditure from heart rate monitoring during submaximal exercise”. In: *Journal of Sports Sciences* 23.3 (March 2005). pp. 289–297.

[67] Joana De Calheiros Velozo, Thomas Vaessen, Jens Pruessner, Ilse Van Diest, Stephan Claes, and Inez Myin-Germeys. “The repeated Montreal Imaging Stress Test (rMIST): Testing habituation, sensitization, and anticipation effects to repeated stress induction”. In: *Psychoneuroendocrinology* 128 (June 2021). p. 105217.