

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
Department of Control Engineering and Information Technology



Ákos Antal
TDK research

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
Department of Control Engineering and Information Technology

ÁKOS ANTAL
TDK research

Development of a Neural ODE-based parameter identification
method for the implementation of personalised fluid therapy

Supervisors:

Dr. Balázs Benyó

Dr. Ákos Szlávecz

Bálint Szabó

BUDAPEST, 2024

Contents

Abstract	V
Kivonat	VI
List of figures	VIII
1 Introduction	1
1.1 Objectives	1
2 Data and Methods	2
2.1 Data	2
2.2 Methods	3
2.2.1 Three Chamber Model	3
2.2.2 NeuralODE	6
2.2.3 Early stopping	8
3 Results and discussion	11
3.1 Simulation of the Three Chamber Model	11
3.1.1 Calculation of the initial values of the parameters	11
3.1.2 Definition of the differential equation describing the model	12
3.1.3 Solving the defined differential equation and the results of the simulation	13
3.2 Identification of the parameters of the Three Chamber Model	16
3.2.1 The metrics and error function of the identification	16
3.2.2 The parameter constraints	17
3.2.3 Hyperparamters	22
3.2.4 Assessment of the identification's robustness	24
3.2.5 Evaluation of the results	28
4 Conclusion	30
4.1 Limitations and future work	30
Acknowledgement	32

Abstract

Acute Circulatory Failure (ACF) is a condition when the heart and circulatory system are unable to provide adequate blood supply to meet the metabolic needs of the organs. It occurs in around 30% of patients in intensive care. Fluid therapy is the most commonly used treatment for ACF, but it fails in a significant proportion of cases (about 50%). Currently, there is no clinically widely applicable non-invasive perfusion marker to assess the effectiveness of fluid therapy, which would allow direct measurement of the impact of therapy on the circulation. Unfortunately, the consequences of unsuccessful fluid therapy have a negative impact on patient outcomes, increasing the length of intensive care unit stay and, in turn, the cost of treatment.

Recently published research suggests that determining the Stressed Blood Volume (SBV) parameter, which characterises the circulatory system, may provide an opportunity to assess the effectiveness of fluid therapy. SBV is a parameter of the Three Chamber Model describing the cardiovascular system. Its value can be determined by parameter identification of the model. The identification method used in this thesis is the Neural ODE method, a member of the Physics-informed Neural Network family. For the adaptation of the Neural ODE method to the given parameter identification of the Three Chamber Model and for the validation of the method, an experimental data set containing measurements on model animals was used, which allowed the analysis of the developed method in extreme physiological situations and an analysis of the accuracy of the identification. In addition to the algorithms developed, the report also presents the potential use of the outcomes in clinical decision-making.

Keywords: parameter identification, NeuralODE, cardiovascular model, personalized intensive therapy

Kivonat

A szervezet azon állapotát, amikor a szív és a keringési rendszer nem tudja biztosítani a megfelelő vérellátást a szervek anyagcsere szükségleteinek kielégítéséhez akut keringési elégtelenségnek (Acute Circulatory Failure – ACF) nevezzük, mely állapot az intenzív osztályon ápolott betegek kb. 30%-ánál fellép. Az ACF kezelésére a folyadék terápia a leggyakrabban alkalmazott kezelési eljárás, azonban az esetek jelentős részében (kb. 50%-ában) ez sikertelen. Jelenleg nincs klinikailag széleskörűen alkalmazható, nem invazív perfúziós marker a folyadékterápia eredményességének értékelésére, mely lehetővé tenné a terápia keringésre gyakorolt hatásának közvetlen mérését. A sikertelen folyadékterápia következményei sajnos negatívan befolyásolják a beteg kezelésének kimenetelét, megnövelik az intenzív osztályon történő kezelés időtartamát és ezzel párhuzamosan a kezelés költségeit.

A közelmúltban publikált kutatások alapján valószínűsíthető, hogy a keringési rendszert jellemző ún. Stressed Blood Volume (SBV) paraméter meghatározása lehetőséget adhat a folyadékterápia hatásosságának vizsgálatára. Az SBV a kardiovaszkuláris rendszert leíró ún. Three Chamber modell egyik paramétere. Ennek értékét a modell paramétereinek identifikációjával meg lehet határozni. A dolgozatban alkalmazott identifikációs eljárás a Physics-informed Neural Network hálózat-családba tartozó ún. Neural ODE módszer. A módszer Three Chamber modellre történő adaptációjának kidolgozásához, illetve annak validációjához modell állatokon végzett kísérletek eredményei kerültek felhasználásra, mely adatsorozat lehetővé tette a kidolgozott módszer egymástól lényegesen eltérő élettani helyzetekben történő elemzését, illetve az identifikáció pontosságának alapos elemzését. A dolgozat a kidolgozott eljárások mellett bemutatja az eredmények diagnosztikai döntéshozatal során történő felhasználási lehetőségeit is.

Kulcsszavak: paraméter identifikáció, NeuralODE, kardiovaszkuláris modell, személyre szabható intenzív terápia

List of Figures

2.1	The schedule of interventions performed on the examined pigs [1].	2
2.2	The schematic representation of the Three Chamber Model [2].	4
2.3	One heartbeat of the synthetic data (top) and of measurement data (bottom).	6
2.4	Model of the residual neural network.	7
2.5	The difference and similarity between ResNets and NeuralODEs [3].	8
3.1	Simulation results on synthetic data using the calculated initial parameters with the Euler and Tsit5 methods.	14
3.2	Simulation results on a measurement using the calculated initial parameters with the Euler and Tsit5 methods.	15
3.3	The seven model parameters as functions of the number of training epochs. During the model training, the learning rate was 0.01, which gives more stability but results in slower progress. The different colours show results obtained with different exponential coefficients: (a) $\lambda = 10$; (b) $\lambda = 100$; (c) $\lambda = 500$ and (d) $\lambda = 1000$	18
3.4	The seven model parameters as functions of the number of training epochs. During the model training, the learning rate was 0.1, which gives less stability but results in faster progress. The different colours show results obtained with different exponential coefficients: (a) $\lambda = 10$; (b) $\lambda = 100$; (c) $\lambda = 500$ and (d) $\lambda = 1000$	19
3.5	The seven model parameters as functions of the number of training epochs. During the training, a ReLU activation function was used to ensure positive parameter values. The different colors show results obtained with different learning rates: (a) learning rate= 0.1; (b) learning rate= 0.01 and (c) learning rate= 0.02.	20
3.6	The seven model parameters as functions of the number of training epochs. During the training, new parameters were introduced using the log transform to ensure positive parameter values. The different colours show results obtained with different learning rates: (a) learning rate= 0.1; (b) learning rate= 0.01 and (c) learning rate= 0.02.	21
3.7	Errors of each metric and the total error as a function of training epochs. The errors were calculated for identifications using one of the following learning rates: 0.005, 0.01, 0.05, 0.1, 0.2. The <i>base</i> section of the measurements on subject #4 were used to create the plot.	23

3.8	Errors of each metric and the total error as a function of training epochs for the first 75 epochs and the two learning rates resulting in lowest errors (learning rate $\in \{0.1, 0.2\}$). The <i>base</i> section of the measurements on subject #4 were used to create the plot.	24
3.9	The initial value grid for the E_a and SBV parameters in the benchmark test.	25
3.10	The initial and identified values of SBV in the benchmark test.	27
3.11	The final losses and number of iterations during learning as functions of initial parameter values of SBV in the benchmark test.	27
3.12	Identification results on the measurement data of subject #4 compared to stroke volume values derived from measurement data. The different intervention sections during the experiment are labeled: base; RM1 (first recruitment manoeuvre); fluid (fluid infusion); RM2 (second recruitment manoeuvre) and endotoxin (endotoxin infusion) – detailed in Section 2.1. .	28

1 Introduction

In the case of Acute Circulatory Failure (ACF), the introduction of intravenous fluid – called fluid therapy – is a widely used intervention to stabilize heart function [4]. However, increasing the volume of fluid within the circulatory system does not inherently enhance circulation. In some instances, this approach may even prove detrimental. The Three Chamber physiological model (TCM) provides a concise description of the cardiovascular system as a network of three elastic chambers connected by resistive elements. Fluid responsiveness – describing the effectiveness of fluid therapy – can be predicted by the identification of the Three Chamber Model. One limitation of this approach is the need for aortic pressure measurements, which are invasive and not commonly used in clinical settings. Peripheral blood pressure measurements can serve as a surrogate for aortic pressure through the Tube-load model [5], which estimates the relationship between peripheral and aortic pressure using three model parameters. In a previous study, I investigated the identification of this model [6].

The stressed blood volume (SBV) is defined as the total pressure generating blood volume. SBV can be used in clinical practice to indicate fluid responsiveness [7], so it is the main target of using the Three Chamber Model in this research. The Three Chamber Model has seven model parameters, one of which is SBV. By identifying the parameters of the model using measurement data, the biomarker is also obtained, and can be used to predict fluid responsiveness.

1.1 Objectives

The aim of this work is to develop and evaluate a NeuralODE-based method for the parameter identification of the Three Chamber Model. To achieve this goal, first the simulation of the cardiovascular model is implemented. This is used to develop the NeuralODE-based parameter identification method. This study also aims to evaluate the developed identification method. A benchmark test is conducted using synthetic measurement data, that allows for the quantitative evaluation of the method. The seven model parameters of the TCM all contain valuable information about the state of the patient’s cardiovascular system. The results of the identification can be used for diagnostics and therapy. My work focuses on one model parameter, the stressed blood volume, which can be used to predict fluid responsiveness.

2 Data and Methods

2.1 Data

The experimental data set was obtained from a series of animal experiments conducted at the Centre Hospitalier Universitaire de Liège as part of a collaborative project with the University of Canterbury. The specifics of the animal experiments are detailed in the doctoral dissertation of Rachel Genevieve Rose Smith, entitled "Model-based Hemodynamic Management of Critically Ill Patients" [1]. This section provides a concise overview of the procedures. Accordingly, the following sections are based on the aforementioned PhD thesis.

In the experiments, a cohort of seven pure Piétrain pigs, with an average weight of 23.5 kg (range: 18.5 to 29.0 kg), was used. The subjects were initially sedated and anaesthetised. The sedation and anaesthesia were maintained via a continuous infusion of pharmaceutical drugs, namely sufentanil, thiobarbituric acid and nimbex, delivered via a superior vena cava catheter. The pigs were mechanically ventilated. High-fidelity pressure catheters were utilised in the proximal aorta (P_{ao}), femoral artery (P_{abd}) and vena cava (P_{cv}) in order to ascertain the blood pressure. Additionally, measurements were taken of the left ventricular pressures (P_{LV}) and volumes (V_{LV}). The flow into the aorta (Q_{ao}) was measured using an aortic flow probe positioned on the proximal aorta in the vicinity of the aortic valve. Once the requisite probe had been located, the thorax was held closed using the appropriate clamps. All data were acquired at a sampling rate of 250 Hz and subsequently recorded as a single Notocord data file.

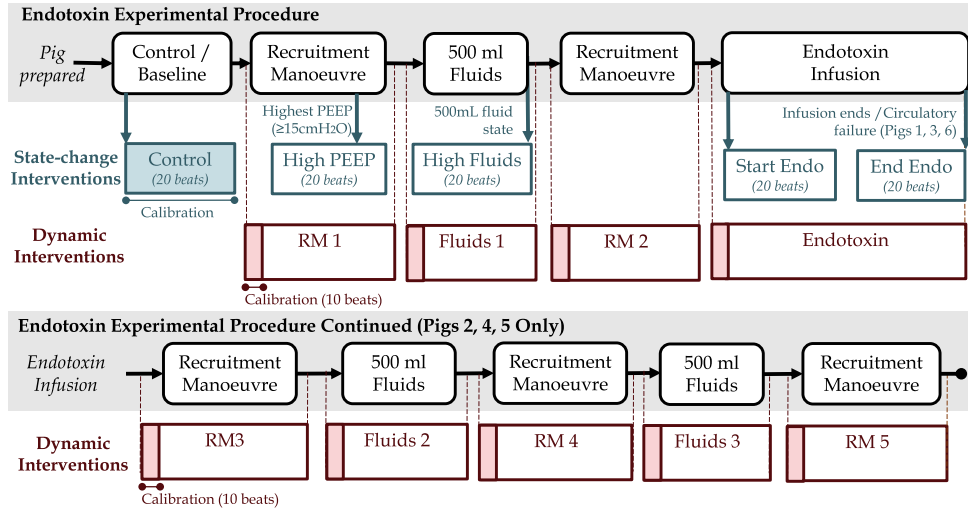


Figure 2.1: The schedule of interventions performed on the examined pigs [1].

The pigs were subjected to a series of haemodynamic interventions during the course of the measurements. The schedule of the interventions is presented in Figure 2.1. The summary of the intervention is as follows:

- **Recruitment Manoeuvres (RM):** A number of respiratory recruitment manoeuvres in which positive end-expiratory pressure (PEEP) is increased in steps of 5 cmH₂O. Increasing positive end-expiratory pressure reduces systemic venous return to the right heart and increases pulmonary resistance. The reduction in flow in and out of the right ventricle leads to a corresponding drop in flow into the left ventricle and a drop in stroke volume (SV). The effect of PEEP changes may be reduced due to the opening and then clamping of the chest for placement of the aortic flow probe. Pigs 2, 4, 5 and 7 underwent five recruitment manoeuvres. Pigs 1, 3, and 6 died before completion of the experimental protocol and underwent two recruitment manoeuvres.
- **Fluid Infusion:** Intravenous infusions of saline solution, delivered at a rate of 500mL over 30 min. This intervention aims to increase circulatory volume and ventricular preload, thus SV. Pigs 2, 4, 5 and 7 underwent three fluid infusions. Pigs 1, 3, and 6 died before completion of the experimental protocol and underwent one infusion.
- **Endotoxin:** An infusion of endotoxin (E. Coli lipopolysaccharide) at a rate of 0.5mgkg⁻¹ over 30 minutes to produce a septic shock-like response: inflammation, capillary leakage, decreased afterload, hypovolemia, tissue hypoxia and eventual cardiac failure. Pigs 2, 4, 5 and 7 completed the full endotoxin infusion. Pigs 1, 3, and 6 died from circulatory failure before completion of the infusion.

2.2 Methods

This chapter summarizes the models used in this research. A part of the methods used in this paper were also employed in a previous study; therefore, the description of some of the methods is derived from that earlier work [6].

2.2.1 Three Chamber Model

The Three Chamber Model (TCM) is a lumped parameter model of the cardiovascular system [2]. The model comprises three elastic chambers, representing the left ventricle (lv), the aorta (ao) and one vena cava (vc). The first chamber is characterised as an active chamber, while the remaining two are classified as passive chambers. The two passive chambers are described by the following equations:

$$P_{ao} = E_{ao}V_{S,ao}(t), \quad (2.1)$$

$$P_{vc} = E_{vc}V_{S,vc}(t), \quad (2.2)$$

where P is pressure, E is elastance, and V_S is the blood volume of the chamber, each at the anatomical location indicated by the indexes. The left ventricle is the active chamber,

represented by the following equation [8]:

$$P_{lv} = E_{lv}e(t)V_{S,lv}(t), \quad (2.3)$$

where E_{lv} is the maximum elastance, and $e(t)$ is the normalised pressure-volume ratio, which is a T-periodic function - T being the duration of a heartbeat - ranging from nearly 0 (during filling) to 1 (at end-systole) [2]. The product of $e(t)$ and E_{lv} represents the time-varying elastance of the left ventricle, which makes it an active chamber. The representation of the TCM is shown in Figure 2.2.

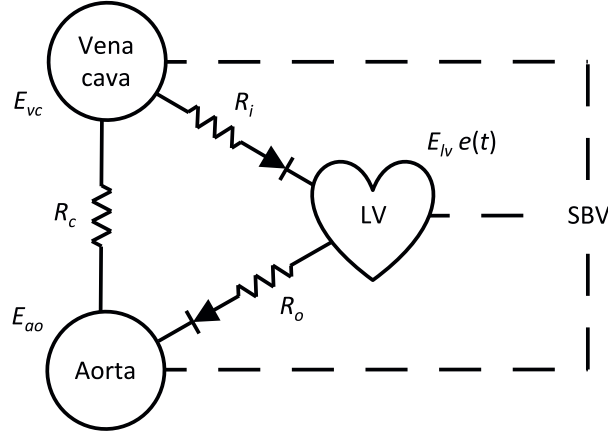


Figure 2.2: The schematic representation of the Three Chamber Model [2].

The chambers are connected through valves. Each connection has a resistance element. The value of R_o represents the resistance of the aortic valve, while R_c corresponds to the resistance of the systemic circulation. The value of R_i , on the other hand, incorporates all resistances from the tricuspid valve to the mitral valve, including the entire pulmonary circulation. Flow is described in accordance with Ohm's law, with the addition of two one-way valves:

$$Q_c(t) = \frac{P_{ao}(t) - P_{vc}(t)}{R_c}, \quad (2.4)$$

$$Q_i(t) = \begin{cases} \frac{P_{vc}(t) - P_{lv}(t)}{R_i}, & \text{if } P_{vc} > P_{lv}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.5)$$

$$Q_o(t) = \begin{cases} \frac{P_{lv}(t) - P_{ao}(t)}{R_o}, & \text{if } P_{lv} > P_{ao}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.6)$$

The rate at which the volumes change is given by the continuity equations:

$$\dot{V}_{S,lv} = Q_i(t) - Q_o(t), \quad (2.7)$$

$$\dot{V}_{S,ao} = Q_o(t) - Q_c(t), \quad (2.8)$$

$$\dot{V}_{S,vc} = Q_c(t) - Q_i(t). \quad (2.9)$$

The summation of the three continuity equations yields

$$\dot{V}_{S,lv} + \dot{V}_{S,ao} + \dot{V}_{S,vc} = 0. \quad (2.10)$$

By integrating Equation 2.10, the sum of chamber volumes is a constant and a model parameter:

$$V_{S,lv} + V_{S,ao} + V_{S,vc} = \text{SBV}. \quad (2.11)$$

The TCM has eight parameters: the three resistances (R_o , R_c , R_i), three elastances (E_{lv} , E_{ao} , E_{vc}), the sum of the volumes of the chambers (stressed blood volume (SBV)) and the cardiac period (T).

It has been demonstrated that stressed blood volume (SBV) may serve as a potential index of fluid responsiveness. The estimation of SBV can be achieved through the utilisation of the Three Chamber Model, thereby circumventing the necessity for highly invasive measurement techniques. Consequently, the assessment of stressed blood volume can be employed in clinical practice as an indicator of fluid responsiveness [7].

To assess the robustness of the developed parameter identification method, a dataset is necessary for which the ground truth parameters are known. This can be achieved by simulating the Three Chamber Model using known constant parameter values. The simulation parameters can be found in Table 2.1. The obtained synthetic dataset contains blood pressure data for all chambers, blood flow data for the Aorta chamber and volume data for the Left ventricle chamber. One heartbeat of the synthetic dataset compared to measurement data is visualized in Figure 2.3.

	E_h	E_a	E_v	R_i	R_o	R_c	V_{s3}
Parameter values	1.4	1.2	0.2	0.05	0.04	2.0	250.0

Table 2.1: The parameter values used to create the synthetic dataset.

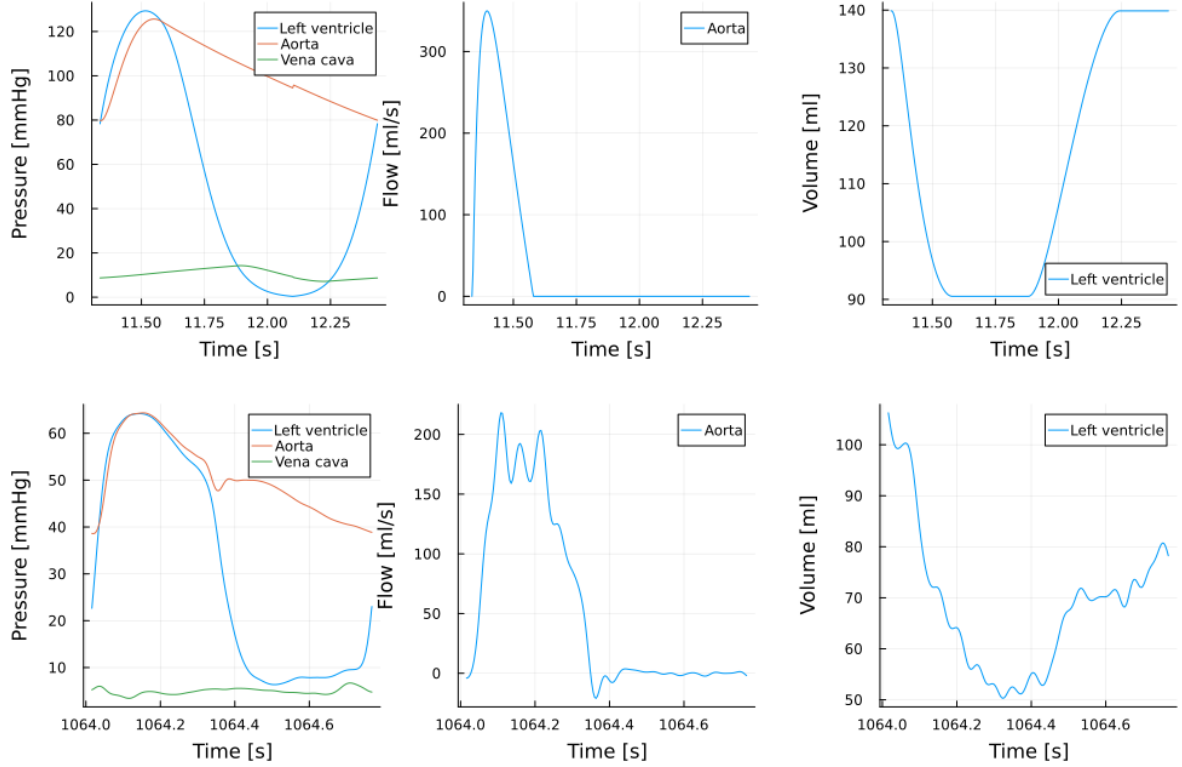


Figure 2.3: One heartbeat of the synthetic data (top) and of measurement data (bottom).

2.2.2 NeuralODE

Neural Ordinary Differential Equations (NeuralODEs) represent a novel category of deep neural networks that integrate the principles of differential equations with those of artificial neural networks [3]. The conjunction of these two disciplines appears to be a logical consequence, given that both differential equations and neural networks are concerned with defining arbitrary nonlinear transformations. In the case of neural networks, this is achieved using linear weights and a non-linear activation function for each layer. Ordinary differential equations define a non-linear transform by defining the derivative of the examined variable and thus leverage a priori knowledge of the structure of the problem.

As our machine learning models become more sophisticated and require larger and larger amounts of data, differential equations have become an attractive option for specifying nonlinearities in a learnable (via the parameters) but constrained form [9]. Neural Ordinary Differential Equations are one of the ways ODEs and artificial neural networks can be combined. The starting point is the Residual Neural Network (ResNet). ResNets are neural networks that were introduced to combat the vanishing gradient problem. They are characterized by skip connections with identity mappings as illustrated in Figure 2.4. These connections facilitate the backpropagation of gradients through deep networks.

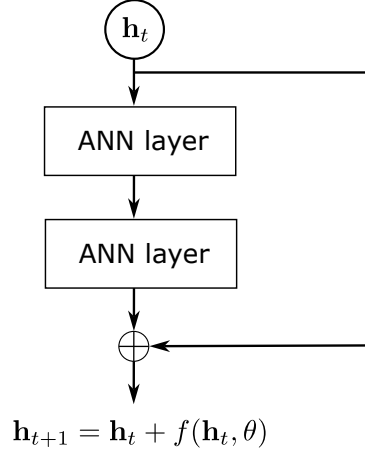


Figure 2.4: Model of the residual neural network.

The output is the sum of the layer output and input/hidden state:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta) \quad (2.12)$$

where $t \in 0 \dots T$, h_t is the state variable vector of the t^{th} layer of the neural network ($\mathbf{h}_t \in \mathbb{R}^D$) and θ represents the weights of the network. Equation 2.2.2 can be seen as an Euler discretisation of a continuous transformation. If so, we can produce the original differential equation by adding more layers to the ResNet and taking smaller steps. In the limit, the neural network can be defined by an ODE:

$$\frac{d\mathbf{h}(t)}{d(t)} = f(\mathbf{h}(t), t, \theta). \quad (2.13)$$

Starting from the input layer $\mathbf{h}(0)$, we can define the output layer $\mathbf{h}(T)$ to be the solution to this ODE initial value problem at some time T . This value can be computed by a black-box differential equation solver, which evaluates the hidden unit dynamics f wherever necessary to determine the solution with the desired accuracy. Figure 2.5 contrasts these two approaches [3].

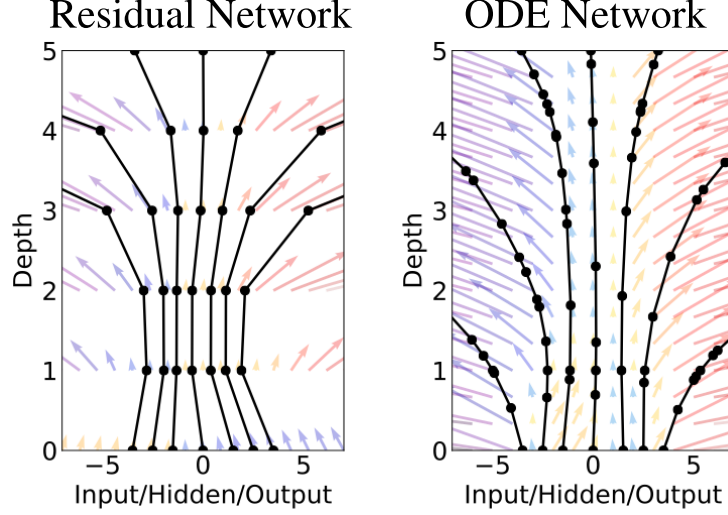


Figure 2.5: The difference and similarity between ResNets and NeuralODEs [3].

The primary challenge in training continuous-depth networks is performing backpropagation, which is the method used to calculate the derivative of the loss function with respect to each parameter. The authors of the NeuralODE method treat the ODE solver as a black box, and compute gradients using the adjoint sensitivity method, which applies to all ODE solvers. The adjoint sensitivity method calculates gradients by solving a second, augmented ODE backwards in time. The approach is scalable, has a low memory cost, and enables explicit control of numerical error [3].

Using neural network models with ODE solvers has numerous benefits, such as:

Memory efficiency: The adjoint sensitivity method provides a way to compute gradients of a scalar-valued loss with respect to all inputs of any ODE solver without backpropagating through all the operations of the solver. Not storing any intermediate quantities of the forward pass allows for training the models with constant memory cost as a function of depth, a major bottleneck of training deep models.

Adaptive computation: Modern ODE solvers provide guarantees about the growth of approximation error, monitor the level of error, and adapt their evaluation strategy on the fly to achieve the requested level of accuracy. This allows the cost of evaluating a model to scale with problem complexity. After training, accuracy can be reduced for real-time or low-power applications.

Continuous time-series models: Unlike recurrent neural networks, which require discretising observation and emission intervals, continuously-defined dynamics can naturally incorporate data which arrives at arbitrary times [3].

2.2.3 Early stopping

The majority of numerical optimisation techniques employed in machine learning operate in an iterative manner, with gradient-based methods representing the most prevalent approach. The solution of these algorithms progresses towards the optimum with each

iteration; however, a substantial number of iterations may be required for convergence. Furthermore, in the context of supervised learning, overfitting can be attributed to an excess of optimisation iterations, which ultimately impairs the model’s generalisation capabilities. One potential solution to the excessive computational cost and overfitting is early stopping. The objective of early stopping is to halt the optimisation process prior to reaching complete convergence. The primary component of an early stopping technique is the stopping criterion, which informs the algorithm of the appropriate point at which to conclude the training process. The criterion employed for early stopping is typically selected in an ad hoc manner, or alternatively, the optimisation process is terminated interactively [10].

In the process of training a machine learning model, the parameters are optimized with the objective of reducing the error evaluated on the training set. However, following a substantial number of iterations, the error rate on the training set continues to decline, whereas the error rate on the validation set begins to rise, suggesting that the model’s ability to generalise is becoming increasingly impaired. This phenomenon is referred to as overfitting. The model exhibits a high degree of precision in replicating the characteristics of the training set, yet its performance on similar but unseen data is decreasing. In order to combat overfitting, a number of techniques may be employed, including a reduction in the number of parameters, the utilisation of regularisation terms, or the implementation of early stopping [10].

The fundamental technique for early stopping in supervised learning tasks is as follows. The data should be divided into two distinct sets: a training set and a validation set. The parameters should be optimised using the training set, with the objective of reducing the error evaluated using the aforementioned training set. During the process of optimisation, it is also necessary to calculate the error on the validation set. In an ideal scenario, the training errors will converge towards zero. However, due to the phenomenon of overfitting, the validation error will initially decrease before subsequently increasing. The training process should be terminated when the validation error begins to increase. This straightforward technique is effective in theory; however, when utilising real-world datasets, the validation error as a function of training epochs is not smooth and numerous local minima are present, making it challenging to ascertain the optimal point at which to halt the training process.

L. Prechelt proposes three more sophisticated classes of early stopping criteria [10]. In the first class, the criterion is defined using a term called *generalization loss*:

$$GL(t) = 100 \cdot \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right), \quad (2.14)$$

where $E_{va}(t)$ is the current, and E_{opt} is the minimum of the error evaluated on the validation set. The stopping criterion is defined as the *generalization loss* reaching a pre-defined threshold.

In the second class, another term, the *training progress* is used:

$$P_k(t) := 1000 \cdot \left(\frac{\sum_{t'=t-k+1}^t E_{tr}(t')}{k \cdot \min_{t'=t-k+1}^t E_{tr}(t')} - 1 \right), \quad (2.15)$$

where E_{tr} is the error evaluated on the training set and k is the length of the examined training strip. The *training progress* defines “how much was the average training error during the strip larger than the minimum training error during the strip?” [10]. The criterion of the second class is that the ratio of the *generalization loss* and the *training progress* reaches a pre-defined threshold.

The stopping criterion of the third class is that the *generalization loss* increases in s successive strips.

3 Results and discussion

3.1 Simulation of the Three Chamber Model

The equations and variables of the Three Chamber Model (TCM) are described in Section 2.2. In short, the cardiovascular system is described by one active chamber (the left ventricle), with varying elastance representing activity, and two passive chambers (the aortic and the venous systems) with constant elastances. The three chambers represent the arterial system (aorta), the venous system including the entire pulmonary circulation (vena cava) and the heart (left ventricle). The relation between pressure and volume is given by the three elastances – $E_{lv}(t)$, E_{ao} , E_{vc} –, and between pressure difference and flow by three resistances (input, output and circulatory resistances – R_i , R_o , R_c). These six parameters together with the total stressed blood volume (SBV) and the cardiac period (T) form the parameter vector of the model. The stressed blood volume is defined as the total pressure generating blood volume. SBV can be used in clinical practice to indicate fluid responsiveness [7], so it is the main target of using the Three Chamber Model in this research.

The simulation of the TCM starts with choosing initial values of the parameters and then simulating several beats using the model equations. The required steps include

1. the calculation of correct initial values,
2. the definition of the differential equation describing the model, and
3. solving the differential equation for a given number of heartbeats.

3.1.1 Calculation of the initial values of the parameters

It is essential to accurately determine the initial values of the model parameters for the purpose of identifying the model and thereby reducing the time required for minimisation of the cost function, while simultaneously ensuring the presence of fewer local minima between the initial and optimal values. In the context of the simulation, the optimal initial values are not a primary concern. However, as this work also addresses the identification process, a detailed account of the precise initial value definition is provided in this section. The initial parameter values derived by A. Pironet et al. [11] are used in this work. For the definition of the initial parameter values, a synthetic dataset containing pressure, volume and flow values is used. One heartbeat of the used test data is shown in Figure 2.3.

The initial values of the parameters are calculated as follows:

1. **T**: the cardiac period is calculated as the distance between successive minima of the central arterial blood pressure.

2. R_o : the output resistance – that is the flow resistance between the central artery and the left ventricle of the heart – is calculated using the pressure and volume curves during ejection: $R_o = \frac{\int_{t_{BE}}^{t_{EE}} (P_{lv}(t) - P_{ao}(t)) dt}{SV}$, where t_{BE} and t_{EE} represent the beginning and end of ejection, P_{lv} , P_{ao} are the pressures of the left ventricle and the aorta, and SV is the stroke volume.
3. R_i : the input resistance – that is the flow resistance between the vena cava and the heart (containing the whole pulmonary circulation) – is defined similarly, but for filling instead of ejection and the pressures of the vena cava and left ventricle are used: $R_i = \frac{\int_{t_{BF}}^{t_{EF}} (P_{vc}(t) - P_{lv}(t)) dt}{SV}$.
4. R_c : the circulatory resistance – that is the flow resistance between the central artery and the vena cava – can be calculated using the mean pressures of the aorta and vena cava: $R_c = T \frac{\bar{P}_{ao} - \bar{P}_{vc}}{SV}$.
5. E_{lv} : the left ventricular elastance consists of a constant E_{lv} term and a time-dependent driver function $e(t)$: $E_{lv}(t) = E_{lv}e(t)$. The constant is calculated as the maximum elastance: $E_{lv} = \max_T \frac{P_{lv}(t)}{V_{lv}(t)}$. The driver function is obtained as: $e(t) \approx \frac{P_{lv}(t)}{E_{lv}V_{lv}(t)}$, then the resulting curve may be approximated by its first N Fourier coefficients: $\hat{e}_q = \frac{1}{T} \int_0^T \exp\left(-\frac{2jq\pi t}{T}\right) dt$, with $q = 0, 1, \dots, N$. The driver function can be included in the parameter set, however, in this work it is handled as constant and it is calculated based on left ventricular pressure and volume measurements.
6. E_{ao} : the aortic elastance initial value is calculated by fitting the following equation during diastole: $P_{ao}(t) \approx \exp\left(-\frac{E_{ao}(t - t_{EE})}{R_c}\right) P_{ao}(t_{EE})$, where t_{EE} represents time at the end of ejection.
7. E_{vc} : the initial value of the venous elastance is calculated using the following equation: $E_{vc} \approx 2 \frac{PP_{vc}}{SV}$, where PP_{vc} is the difference of the maximum and minimum pressure in the vena cava.
8. SBV : the total stressed blood volume was calculated using: $SBV \approx \bar{V}_{lv} + \frac{\bar{P}_{ao}}{E_{ao}} + \frac{\bar{P}_{vc}}{E_{vc}}$.

During the calculation of the initial values of the parameters, the left ventricular, central arterial and central venous pressure measurements (or synthetic data) are used.

3.1.2 Definition of the differential equation describing the model

During the simulation of the model, the stressed blood volume of the three chambers is calculated by solving a differential equation. The pressure and flow values can be

calculated after the simulation using the chamber volumes and the model parameters. The governing differential equation of the model is described by the following equation:

$$\frac{d}{dt} \begin{bmatrix} V_{s,ao} \\ V_{s,vc} \\ V_{s,lv} \end{bmatrix} = \begin{bmatrix} Q_i - Q_o \\ Q_o - Q_c \\ Q_c - Q_i \end{bmatrix}, \quad (3.1)$$

where the input, output and circulatory flows can be calculated using valve functions:

$$\begin{bmatrix} Q_i \\ Q_o \\ Q_c \end{bmatrix} = \begin{bmatrix} \text{valve} \left(\frac{P_{vc} - P_{lv}}{R_i} \right) \\ \text{valve} \left(\frac{P_{lv} - P_{ao}}{R_o} \right) \\ \frac{P_{ao} - P_{vc}}{R_c} \end{bmatrix}. \quad (3.2)$$

The valve functions represent the valves of the heart. The function returns 0, if the argument is not positive, and returns the argument otherwise:

$$\text{valve}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.3)$$

The pressures of the chambers are calculated using the previous values of chamber volumes and the elastances:

$$\begin{bmatrix} P_{ao} \\ P_{vc} \\ P_{lv} \end{bmatrix} = \begin{bmatrix} E_{ao} V_{s,ao} \\ E_{vc} V_{s,vc} \\ E_{lv} e(t) V_{s,lv} \end{bmatrix}. \quad (3.4)$$

The differential equation can be written in a condensed form, using only parameters and the previous values of the stressed volume vector as follows:

$$\frac{d}{dt} \begin{bmatrix} V_{s,ao} \\ V_{s,vc} \\ V_{s,lv} \end{bmatrix} = \begin{bmatrix} \text{valve} \left(\frac{E_{vc} V_{s,vc} - E_{lv} e(t) V_{s,lv}}{R_i} \right) - \text{valve} \left(\frac{E_{lv} e(t) V_{s,lv} - E_{ao} V_{s,ao}}{R_o} \right) \\ \text{valve} \left(\frac{E_{lv} e(t) V_{s,lv} - E_{ao} V_{s,ao}}{R_o} \right) - \frac{E_{ao} V_{s,ao} - E_{vc} V_{s,vc}}{R_c} \\ \frac{E_{ao} V_{s,ao} - E_{vc} V_{s,vc}}{R_c} - \text{valve} \left(\frac{E_{vc} V_{s,vc} - E_{lv} e(t) V_{s,lv}}{R_i} \right) \end{bmatrix}. \quad (3.5)$$

3.1.3 Solving the defined differential equation and the results of the simulation

The simulation of the model is the solution of the differential equation in Equation 3.5. There are a huge number of differential equation solvers capable of solving this task. In this work, the Tsitouras 5/4 Runge-Kutta method (Tsit5) is used as it is implemented in the DifferentialEquations.jl Julia package [12]. A huge advantage of solving the model in Julia is the possibility to use the implementation of NeuralODE, which means the

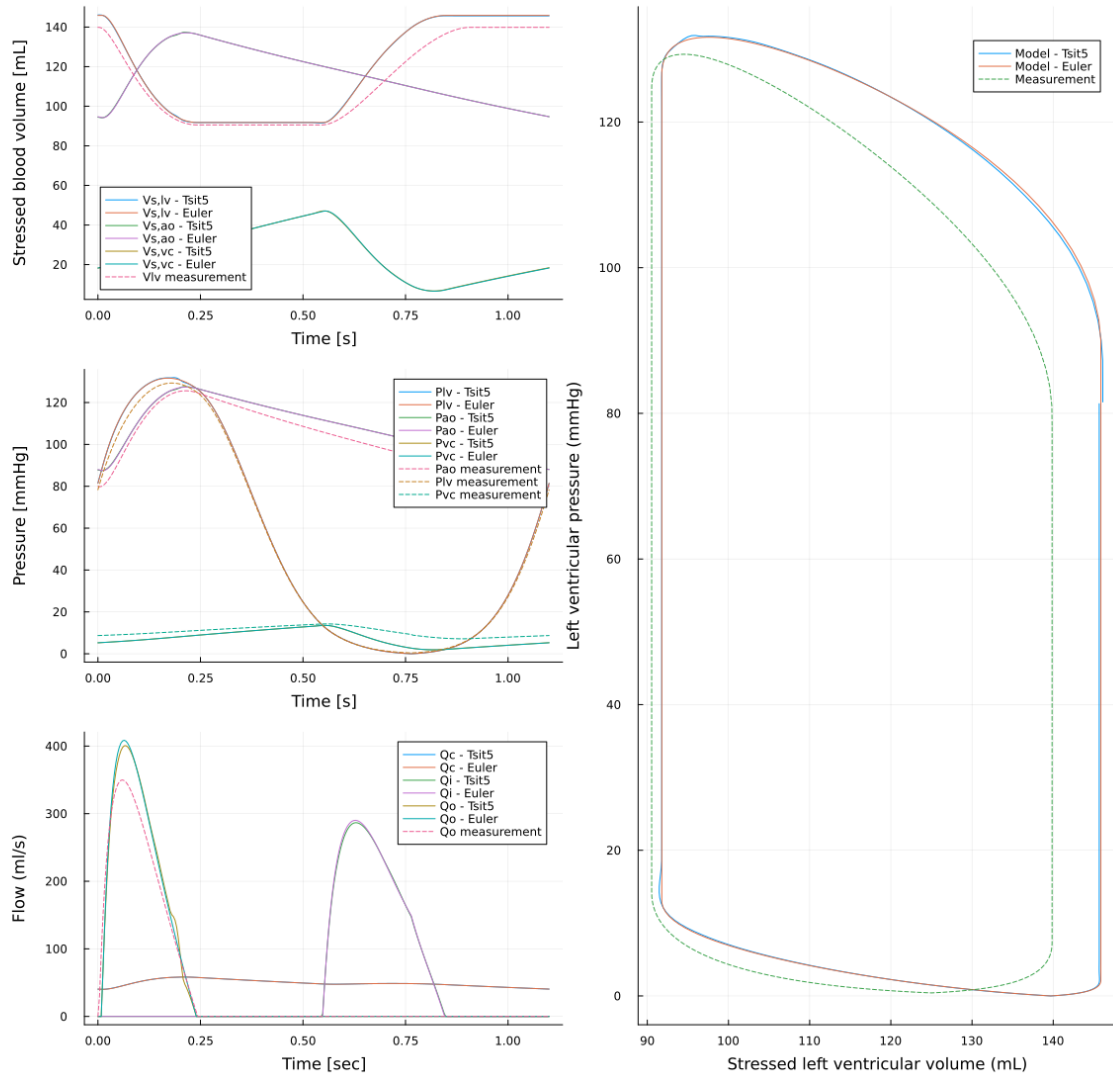


Figure 3.1: Simulation results on synthetic data using the calculated initial parameters with the Euler and Tsit5 methods.

calculation of the reverse-mode derivative of an ODE initial value problem. The model is also solved with the Euler method for comparison.

During the simulation, the initial value problem is solved for a longer time period (approximately 10 heartbeats), so that the transient is not present in the solution. The results shown in this review only contain the end of the simulation where the transient can be neglected.

The simulation results using the synthetic data presented in Section 2.2 and both the Euler and Tsit5 methods are shown in Figure 3.1 together with the data used for the initial parameter value calculation. The results using a randomly chosen subset of the measurement data are shown in Figure 3.2. On the left, the volumes, pressures and flows are presented. On the right, the so-called pressure-volume loop (PV loop) is shown, where the pressure of the left ventricle is plotted as a function of the stressed left ventricular volume during a heartbeat. Based on the Figure, the Euler and Tsit5 methods produce

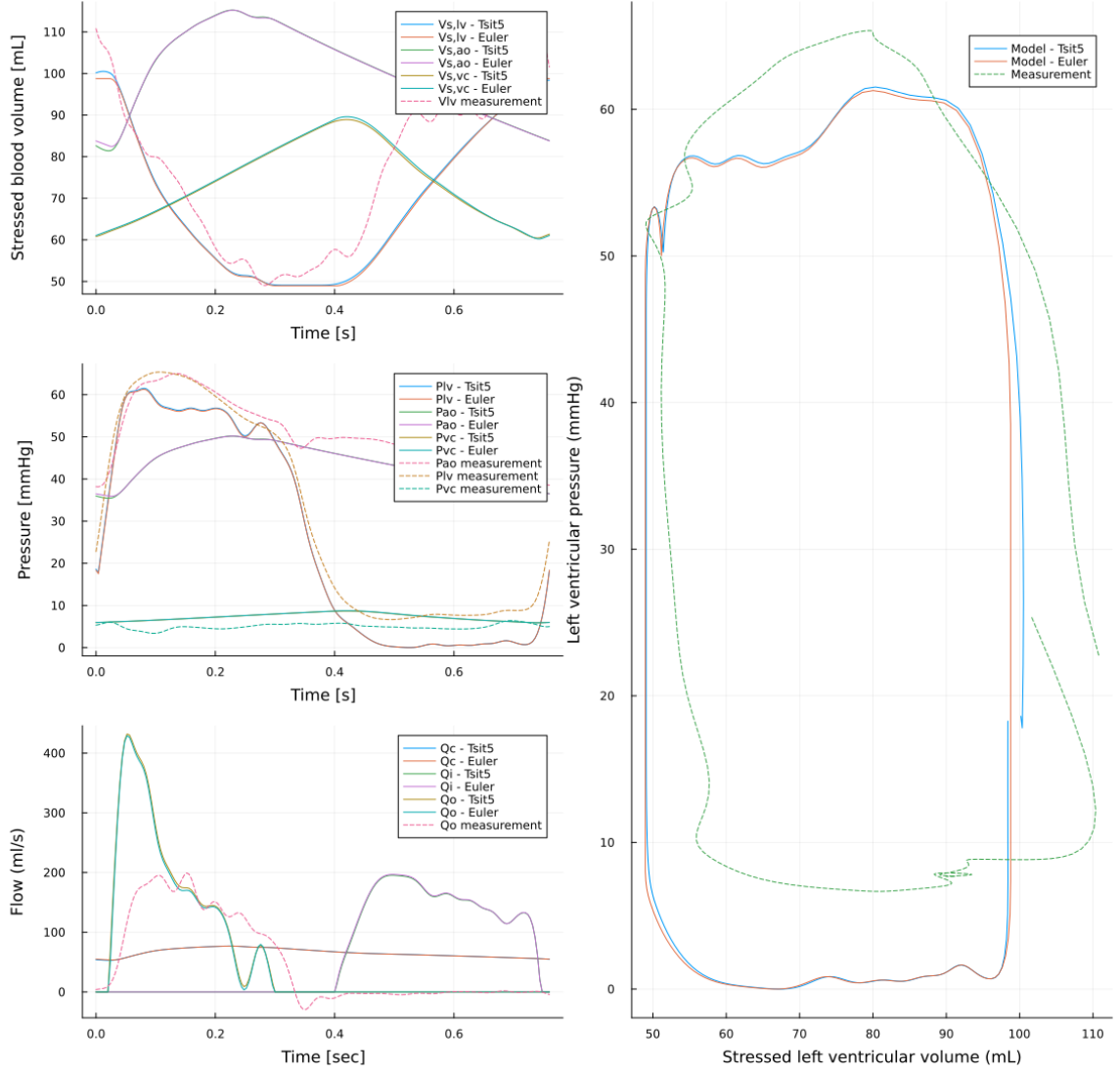


Figure 3.2: Simulation results on a measurement using the calculated initial parameters with the Euler and Tsit5 methods.

similar results, however, not exactly the same. The difference is visible at the peak of Q_o and on the PV loop, where the Euler method produces a steady state after simulating 10 heartbeats – the loop is closed –, while using the Tsit5 method results in a non-continuous PV loop, which can be attributed to transients.

The simulations approximate the synthetic "measurement" data well, especially considering that the parameters are not identified, only estimated with the methods described in Section 3.1.1.

The measurement data seen in Figure 3.2 is visibly less regular. The simulated curves fit the measurement fairly well with the calculated initial parameters and the difference between the Euler and Tsit5 methods is similar to the experiences with the synthetic data.

3.2 Identification of the parameters of the Three Chamber Model

3.2.1 The metrics and error function of the identification

The process of identifying the parameters of the TCM is one aimed at determining the parameter set that will yield the simulation output most closely aligned with the measurements. A cost function is defined which describes the error between the simulation output and the measurement. During the identification process, the parameters are modified with the objective of minimising the cost function. The output vector has to be defined, so that it serves as a measure of similarity between the simulation and measurement data. A. Pironet et al. proposed the following output vector [11]:

$$\begin{bmatrix} \max_T V_{lv}(t) - \min_T V_{lv}(t) \\ \frac{1}{T} \int_T V_{lv}(t) dt \\ \max_T P_{vc}(t) - \min_T P_{vc}(t) \\ \frac{1}{T} \int_T P_{vc}(t) dt \\ \max_T P_{ao}(t) - \min_T P_{ao}(t) \\ \frac{1}{T} \int_T P_{ao}(t) dt \\ \max_T \dot{P}_{ao}(t) \end{bmatrix} = \begin{bmatrix} SV \\ \bar{V}_{lv} \\ PP_{vc} \\ \bar{P}_{vc} \\ PP_{ao} \\ \bar{P}_{ao} \\ dP_{ao}/dt_{max} \end{bmatrix}, \quad (3.6)$$

where the elements of the output vector are the following:

- SV : Stroke volume; the blood volume that the left ventricle pumps per beat,
- $\bar{V}_{lv}$: Mean left ventricular volume over one cardiac cycle,
- PP_{vc} : Venous pulse pressure; the range of vena cava pressure during one cardiac cycle,
- $\bar{P}_{vc}$: Mean vena cava pressure over one cardiac cycle,
- PP_{ao} : Aortic pulse pressure; the range of aortic pressure during one cardiac cycle,
- $\bar{P}_{ao}$: Mean aortic pressure over one cardiac cycle,
- dP_{ao}/dt_{max} : Maximum derivative of aortic pressure over one cardiac cycle.

Our hypothesis is that these seven metrics define the similarity of the simulations to the measurements, and so the aim of the identification is to find the parameters that lead to the output (or metric) vector of the measurement. The paper that introduces these metrics defines the loss function with weights associated with the error of each metric, and uses the standard deviation of the metrics as weights:

$$\text{Loss}_x = \frac{\text{metrics}_{\text{measurement},x} - \text{metrics}_{\text{model},x}}{\sigma_x}. \quad (3.7)$$

3.2.2 The parameter constraints

The model parameters are modified in multiple epochs (training steps) after the model initialisation. In one epoch, the gradient of the defined loss function with respect to the parameter vector is calculated, and then the parameters are updated using the gradients. For the gradient calculation, the ForwardDiff.jl Julia library is used [13], that implements the gradient calculation when an ODE initial value problem is included in the loss function. After calculating the gradients, the ADAM optimizer is used to update model parameters [14].

At first, the loss function was defined as the square sum of the error vector of the metrics:

$$\text{Loss} = \sum_{i=1}^7 \left(\frac{\text{metrics}_{\text{measurement},i} - \text{metrics}_{\text{model},i}}{\sigma_i} \right)^2. \quad (3.8)$$

However, the parameter identification process was unsuccessful due to the assignment of values to the parameters after a few iterations that resulted in an unstable ODE solution. The issue was caused by an insufficiently high flow resistance value, which was either close to zero or even negative. This phenomenon not only results in unstable ODE solutions but is also inconsistent with the physiological principles governing the cardiovascular system. To avoid such situations, multiple methods were analysed:

- It is possible to modify the loss function by introducing a constraint. For instance, an exponential function may be incorporated into the loss function, which increases significantly when the parameters fall below 0.001.
- An activation function may be incorporated into the output of the gradient-based method, ensuring that only positive parameters are produced.
- The crucial parameters can be reparametrised using a log transform, thus enabling their values to fall below zero while maintaining the solubility of the ODE.

Exponential term in loss function The loss function can be extended with a penalty term, that ensures that the parameters do not fall below a pre-defined value. I chose an exponential term, that rises rapidly, so that the parameters do not become negative. The loss function is expressed in Equation 3.9:

$$\text{Loss} = \sum_{i=1}^7 \left[\left(\frac{\text{metrics}_{\text{measurement},i} - \text{metrics}_{\text{model},i}}{\sigma_i} \right)^2 + e^{-\lambda(x_i - 0.001)} \right], \quad (3.9)$$

where x is the parameter vector, and λ is the coefficient of the exponential, which sets the slope of the penalty term.

Figures 3.3 and 3.4 show results obtained using an exponential penalty term. First, a lower learning rate was used so that the parameters do not overshoot during the gradient-based method and do not become negative. Based on the results shown in Figure 3.3,

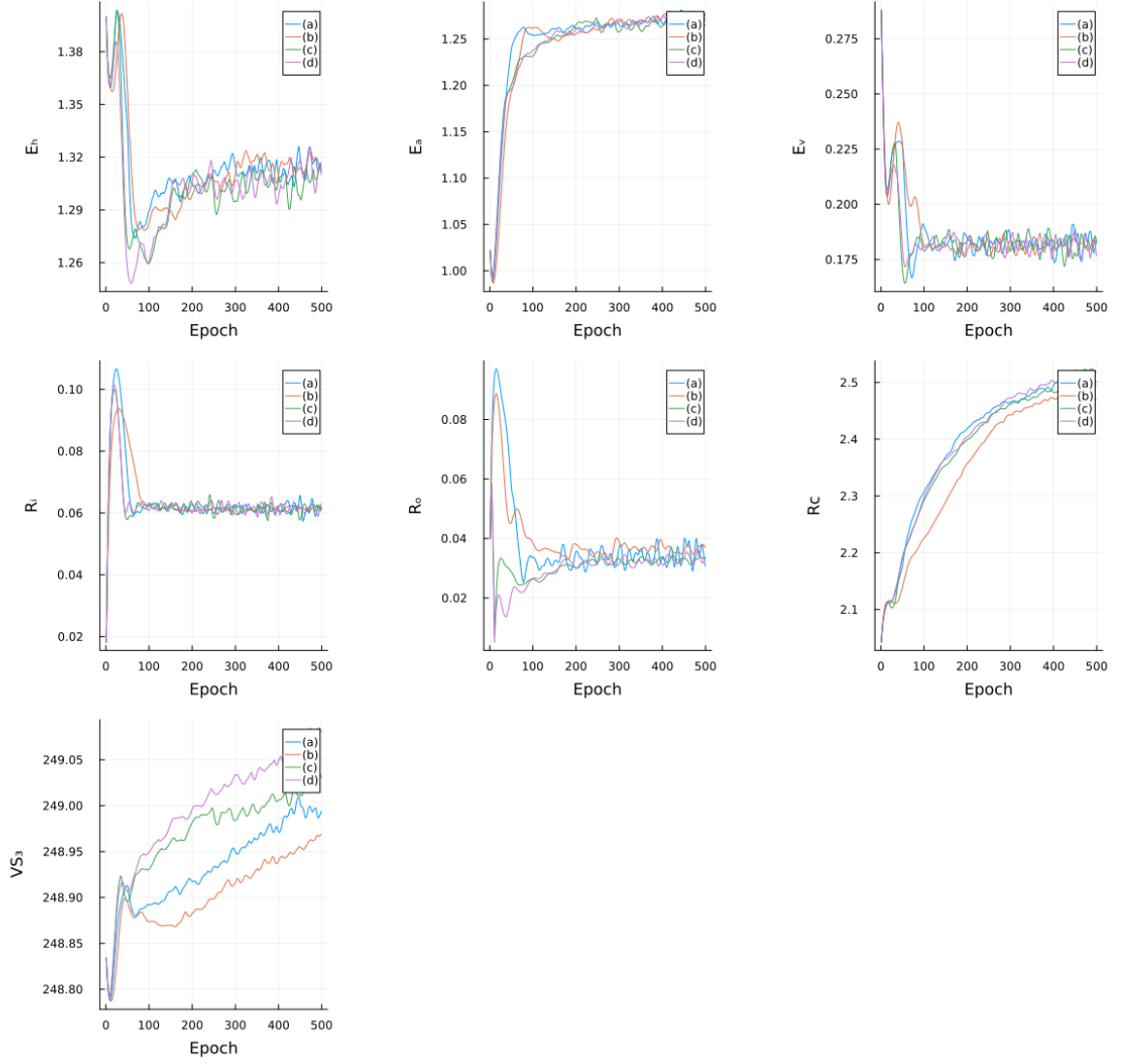


Figure 3.3: The seven model parameters as functions of the number of training epochs. During the model training, the learning rate was 0.01, which gives more stability but results in slower progress. The different colours show results obtained with different exponential coefficients: (a) $\lambda = 10$; (b) $\lambda = 100$; (c) $\lambda = 500$ and (d) $\lambda = 1000$.

the exponential term works as intended, the parameters do not take negative values. However, the learning process is slow, as the VS_3 parameter (also called SBV) has still not converged after 500 epochs. A higher learning rate results in faster learning process. The results shown in Figure 3.4 were obtained using a 0.1 learning rate. As a result of the higher learning rate, the R_o parameter's value changed significantly between two epochs, so the penalty loss – and the gradient – became high, and the parameter value increased excessively.

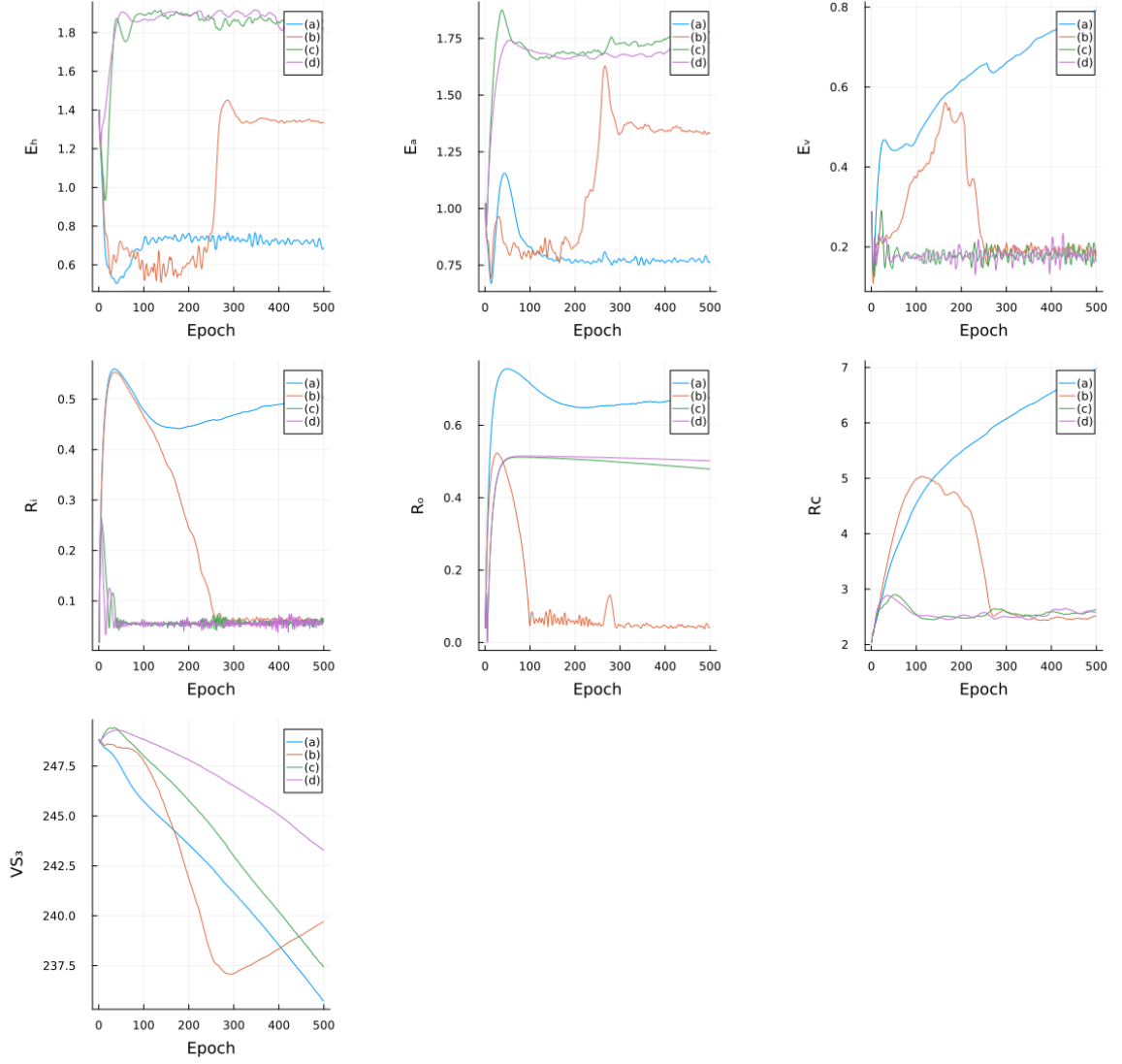


Figure 3.4: The seven model parameters as functions of the number of training epochs. During the model training, the learning rate was 0.1, which gives less stability but results in faster progress. The different colours show results obtained with different exponential coefficients: (a) $\lambda = 10$; (b) $\lambda = 100$; (c) $\lambda = 500$ and (d) $\lambda = 1000$.

Activation function Assuming that the parameters take physiologically correct values once all of them converged, another solution is to use an activation function, that constrains the parameters to remain positive. The optimisation using an activation function is described in Equation 3.10:

$$x_{i+1} = f(\text{Optimizer}(x_i)), \quad (3.10)$$

where f is the activation function, and Optimizer can be any gradient-based optimisation method, such as ADAM. In this work, the ReLU (Rectified Linear Unit) [15] activation function is shifted so that the parameters do not take values below 0.01:

$$f(x) = \max(x, 0.01). \quad (3.11)$$

The results of the identification using the ReLU activation function are shown in Figure 3.5 with different learning rates.

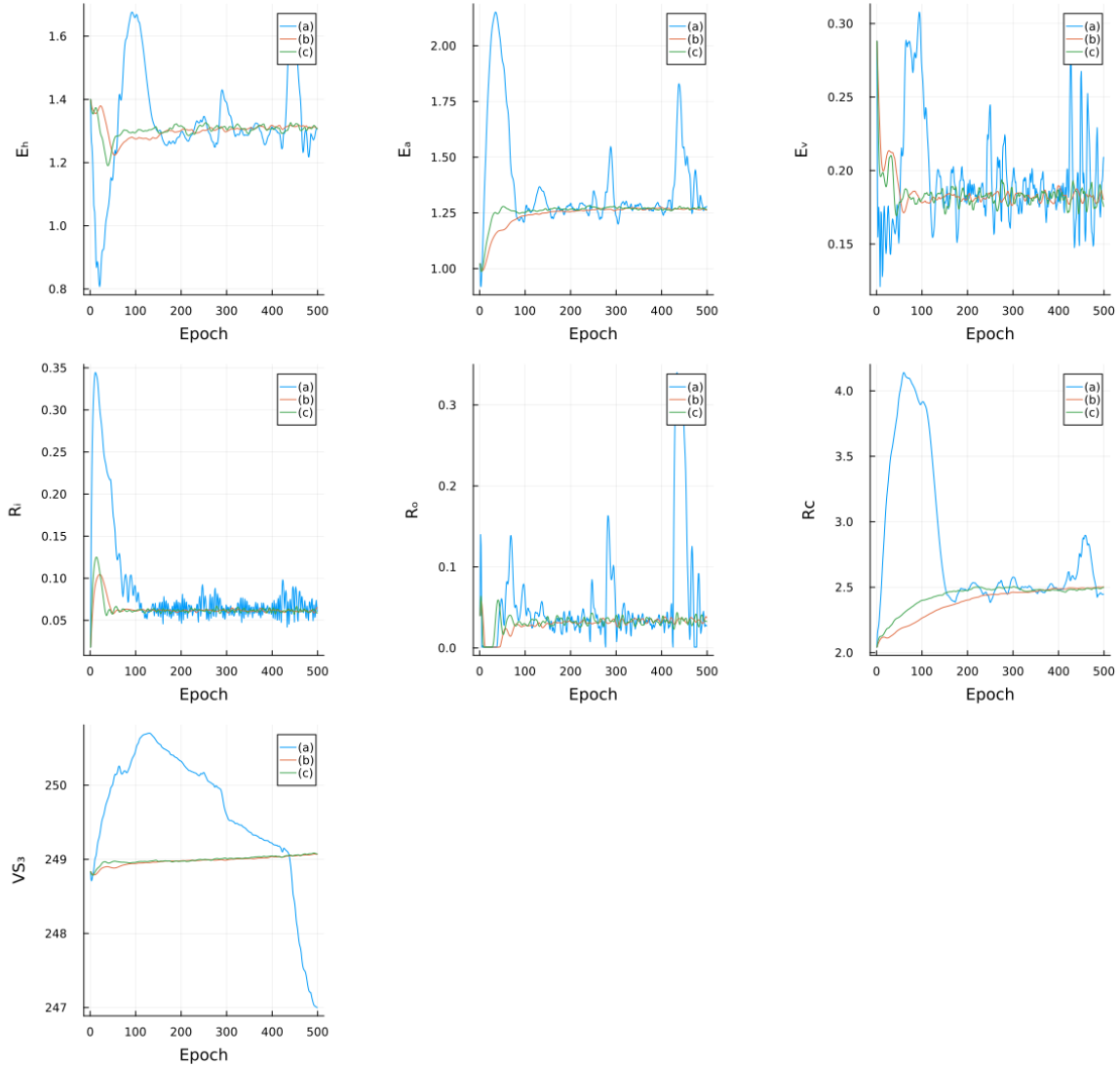


Figure 3.5: The seven model parameters as functions of the number of training epochs. During the training, a ReLU activation function was used to ensure positive parameter values. The different colors show results obtained with different learning rates: (a) learning rate= 0.1; (b) learning rate= 0.01 and (c) learning rate= 0.02.

Log transform The final method presented for avoiding the instability of the solved ODE during the parameter optimisation process is based on a log transform. For an original parameter x_i , it is possible to introduce a new parameter $\hat{x}_i$ so that $x_i = e^{\hat{x}_i}$. The newly introduced $\hat{x}_i$ parameter can be optimized, and the original parameter values can be calculated using the definition of the new parameter whenever needed in the ODE calculation. This way, the new parameter may take negative values, and the original parameters will still not decrease below zero based on the properties of the exponential function.

In the present work, the first six parameters – $E_h, E_a, E_v, R_i, R_o, R_c$ – are prone to take negative values, so these are replaced with the new parameters using the log transform. The results obtained using the log transform on the parameters are shown in Figure 3.6.

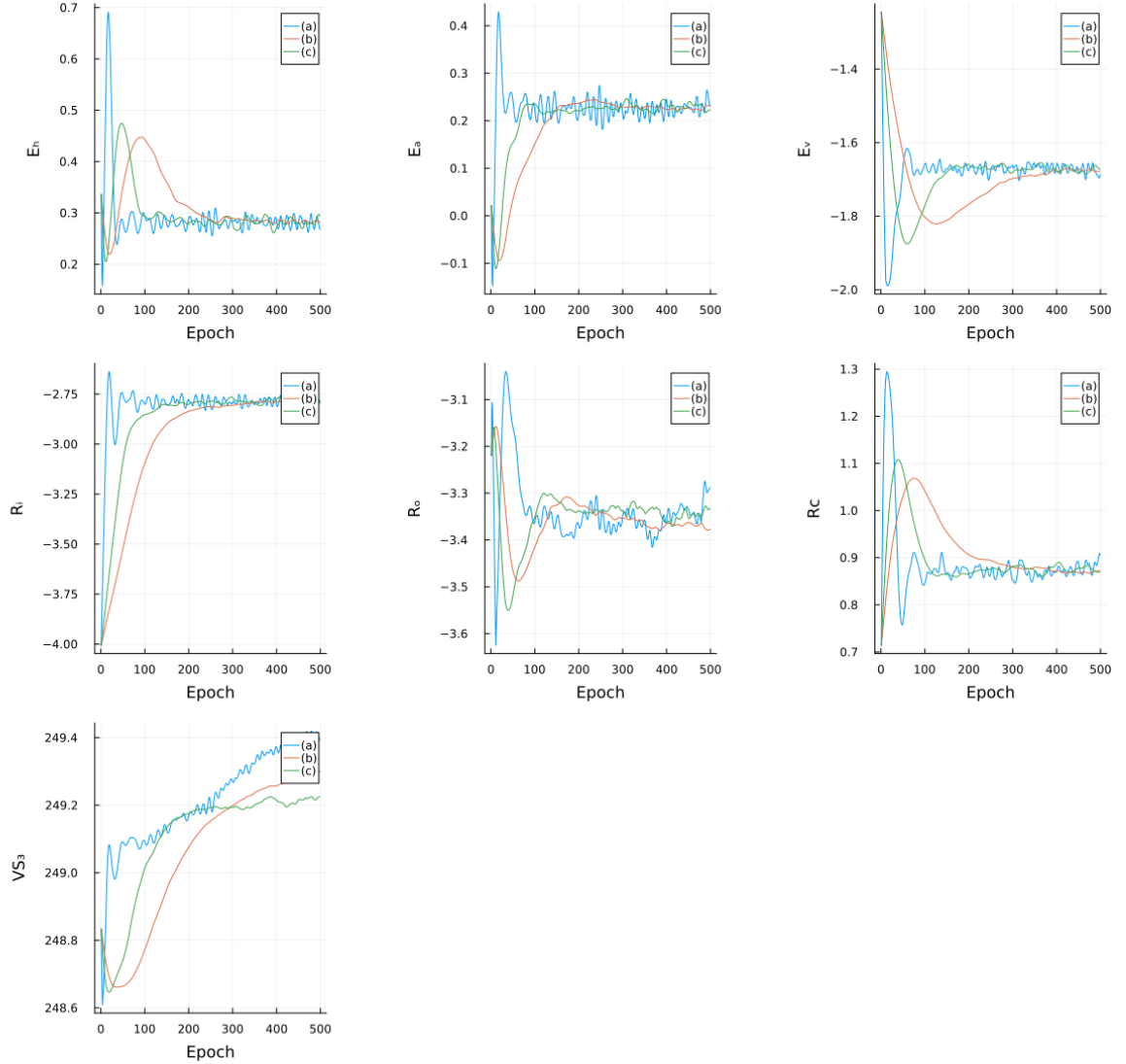


Figure 3.6: The seven model parameters as functions of the number of training epochs. During the training, new parameters were introduced using the log transform to ensure positive parameter values. The different colours show results obtained with different learning rates: (a) learning rate= 0.1; (b) learning rate= 0.01 and (c) learning rate= 0.02.

Evaluation of the parameter constraints In order to determine the most appropriate method, the three described solutions that are used to avoid unstable ODE solutions were subjected to a comprehensive analysis. As a result of using an **exponential term** to augment the loss function, the parameters do not take negative values, however, the high slope of the exponential term causes instability of the parameters during the learning process. The high gradient results in high parameter changes that may result in the algorithm finding a local minimum instead of the global optimum. This phenomenon is not

experienced in settings with low learning rate, however, to speed up the training process, a higher learning rate is required, which results in instability.

The other two methods – using an activation function and using the log transform – do not involve the modification of the loss function. In the case of using an **activation function**, the loss function is evaluated, the parameters are changed according to the computed gradients, and the designed activation function is applied to the changed parameters. This method avoids negative parameter values and unstable ODE solutions, however, it interferes with the training process and may slow down the process of finding the global minimum. As the activation function does not change all of the parameters – only those with low or negative values –, the ratio of the parameters can change, which can result in higher losses. High loss function values cause extreme parameter changes that may lead to local minima or extended time of learning required.

Applying the **log transform** to six of the seven parameters solves the problem of unwanted negative parameter values. This method utilizes that the exponential function is a positive-values function. By introducing a temporary variable $\hat{x} = \log(x)$, the new variable $\hat{x}$ can take negative values as well, and will not result in unstable ODE solutions. This method does not involve the modification of the loss function, and has the advantage of not interfering with the gradient-based optimisation process as the activation function does. Due to its advantageous properties, this method was used in this work.

3.2.3 Hyperparameters

In order to achieve optimal identification, it is necessary to optimise the hyperparameters. It is essential to determine the number of epochs to be used for the identification of each beat in order to prevent premature termination before the optimal parameters have been identified or an excessive running time due to a too high epoch number.

The optimal epoch number can be applied using an **early stopping monitor**. The method analyses the result of each epoch and determines the point at which optimal identification has been reached, and the process is terminated. This results in optimal running time and convergence. In this work, the early stopping parameters were as follows:

- The minimum epoch number: 150
- The stopping criterion: the range of the SBV parameter values of the last 150 epochs is smaller than 1 for 5 consecutive epochs.

The proposed method uses a version of *training progress* defined in Section 2.2. The solution is based on experimental evidence and enables the implementation of a robust early stopping monitor.

The **learning rate** defines how much the parameters are changed in one epoch – the higher the learning rate, the more aggressive the parameter’s change towards the

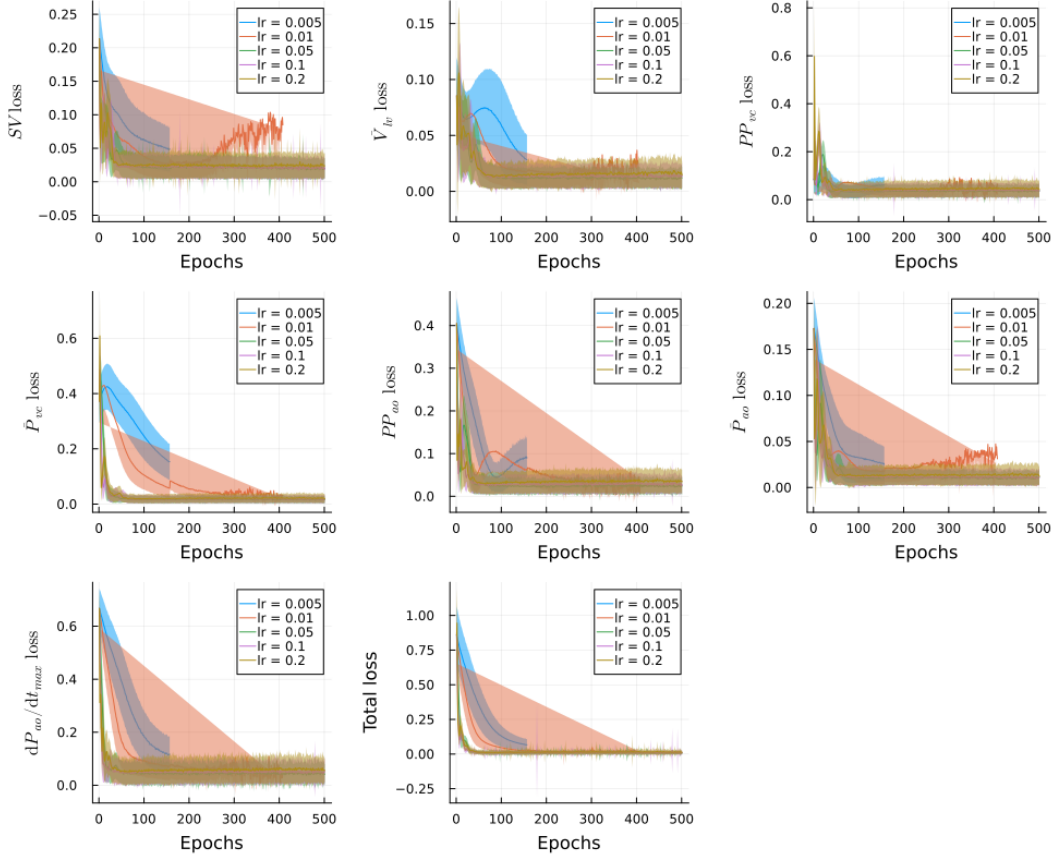


Figure 3.7: Errors of each metric and the total error as a function of training epochs. The errors were calculated for identifications using one of the following learning rates: 0.005, 0.01, 0.05, 0.1, 0.2. The *base* section of the measurements on subject #4 were used to create the plot.

negative gradient, although this can also result in overshooting and negative parameter values. A learning rate that is set too low may result in a longer running time and in the identification of a local minimum rather than the global minimum.

Figure 3.7 shows the absolute value of errors for each metric and the total loss function result as a function of the epochs. The plotted results were obtained using a maximum of 500 epochs and an early stopping method that terminates the training process when the convergence criterion is met and at least 150 epochs have been completed. The learning rates employed for the various curves were 0.005, 0.01, 0.05, 0.1 and 0.2. The parameters were identified for 587 distinct heartbeats (comprising the base section of the measurement data for subject #4) using each learning rate. Each curve of the Figure represents the mean and standard deviation of the 587 recorded losses after a specific number of epochs.

As illustrated in Figure 3.7, the identification process yielded the lowest losses when the learning rates were set to 0.1 and 0.2. Therefore, these curves are presented in Figure 3.8 for the sake of clarity. Figure 3.7 illustrates that the loss is minimised in the initial 50 epochs, with a subsequent reduction in the final 400 epochs that is not statistically significant. Consequently, only the first 75 epochs have been plotted in Figure 3.8. The

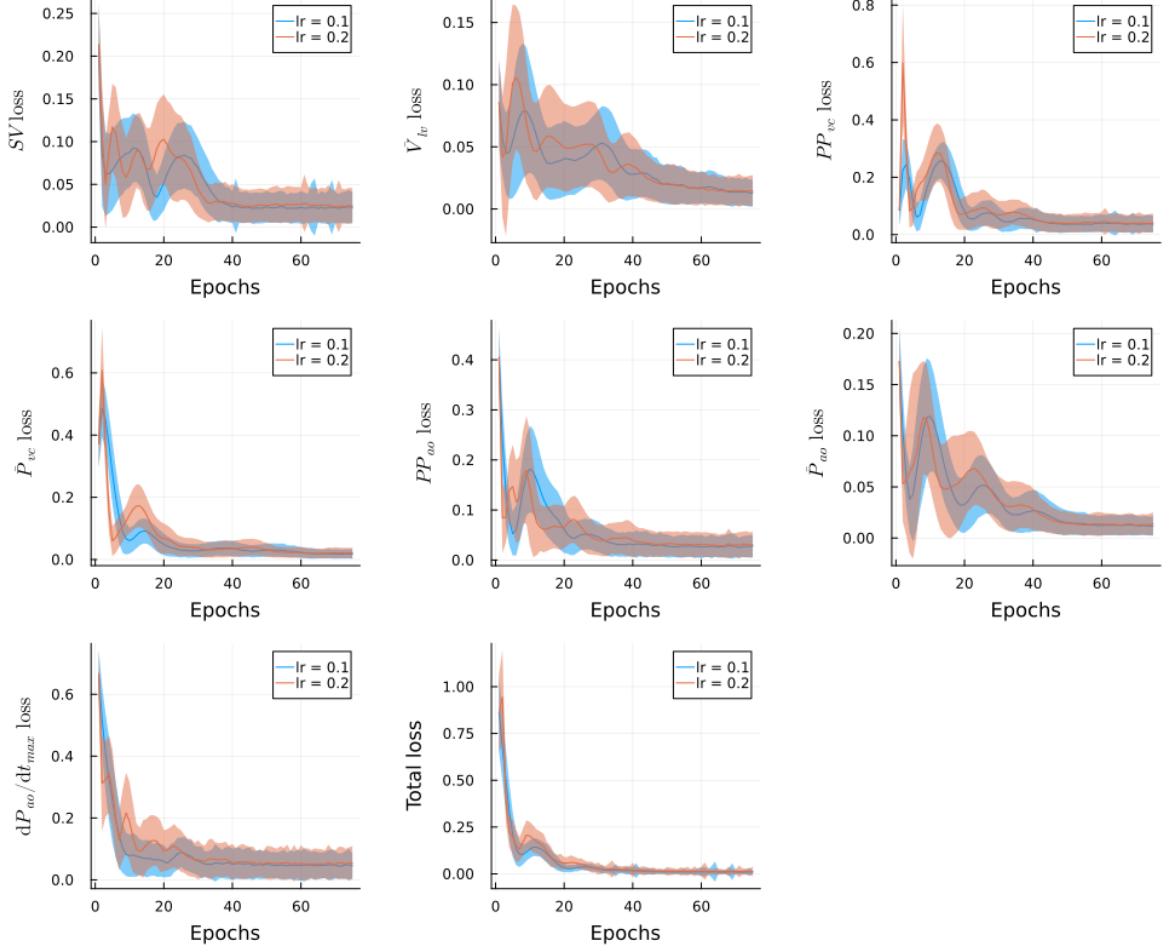


Figure 3.8: Errors of each metric and the total error as a function of training epochs for the first 75 epochs and the two learning rates resulting in lowest errors (learning rate $\in \{0.1, 0.2\}$). The *base* section of the measurements on subject #4 were used to create the plot.

curves exhibit a similar pattern, however, the 0.2 learning rate resulted in longer run times. Therefore, the 0.1 value was selected as the optimal learning rate for this study.

3.2.4 Assessment of the identification’s robustness

To assess the robustness of the implemented NeuralODE-based method regarding the identification of the Stressed Blood Volume parameter, a benchmark test was designed. The synthetic dataset introduced in Section 2.2 is suitable for the benchmark test, as the parameters of the synthetic measurement are known. The robustness of the method is assessed by performing the parameter identification using different initial parameter configurations and evaluating the results of the identification processes. In this work, the initial values of the parameters are either the reference value (the value used to create the synthetic data set) or the reference value shifted by a certain ratio (0.1, 0.2, 0.4, 0.5 or 0.6). Since the model has 7 parameters, the combinatorial space of variation is very large, and

therefore only 2 parameters were shifted at a time, either negatively or positively relative to the reference value. Using this method gives 2100 combinations where two parameters are shifted compared to the reference values, 70 combinations where one parameter is shifted, and one combination where all parameters have the reference values. Figure 3.9 shows the initial parameter value combinations of E_a and SBV , if all other parameters take their reference values.

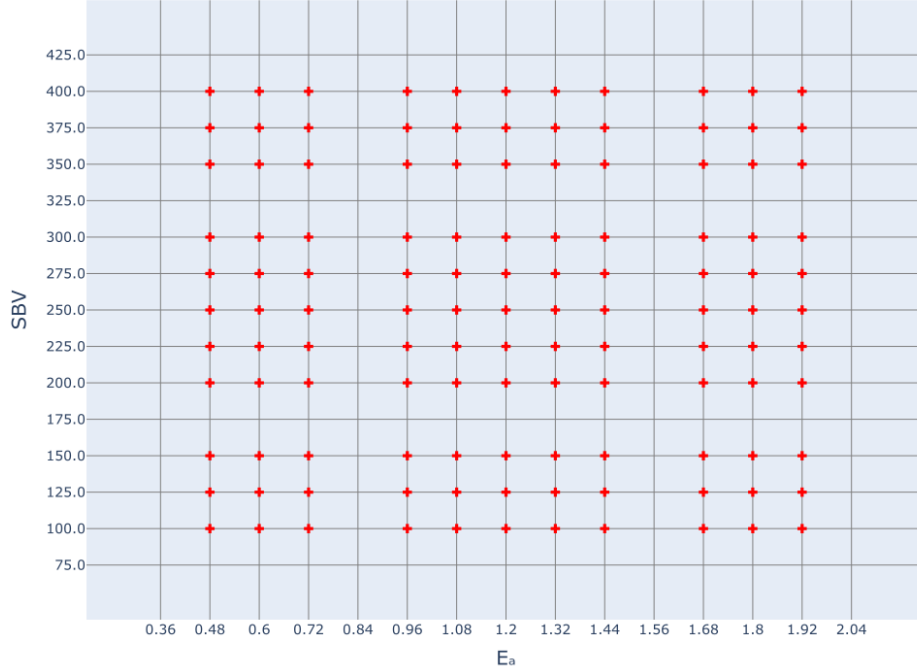


Figure 3.9: The initial value grid for the E_a and SBV parameters in the benchmark test.

Tables 3.1, 3.2 and 3.3 summarize the results of the benchmark test for each parameter. The results are grouped for each parameter according to the initial value in each simulation. The tables show the mean, standard deviation and coefficient of variation of the identified parameter values for each initial value of each parameter. The coefficient of variation is the normalized deviation, calculated as the ratio of the standard deviation and the mean. This represents the results better since the different parameters have different scales, which makes the standard deviations hard to interpret.

Based on the results, some observations can be made about the method:

1. The optimal values of the parameters do not precisely match the simulation values with the applied cost function. Analysing the results when the parameters were initialized from the reference values (offset = 0%), it was observed that the identified parameters did not precisely take on the initial values.
2. The method demonstrated robustness with regard to the initial values of the parameters. The standard deviations and coefficients of variation indicated that the

method was reliable, with the parameters converging even in instances where high initial value errors were present.

offset	E_h	E_a	E_v	R_i	R_o	R_c	V_{s3}
0%	1.37738	1.20084	0.209571	0.0549619	0.040144	2.09681	247.434
10%	1.38619	1.20714	0.210431	0.0544629	0.0397038	2.10466	245.881
20%	1.38054	1.20591	0.210653	0.0545713	0.0397211	2.10691	245.826
40%	1.38545	1.19934	0.210147	0.0545855	0.0397226	2.09581	245.593
50%	1.38422	1.20652	0.210035	0.05464	0.0393665	2.09371	245.445
60%	1.38045	1.1942	0.209536	0.0546288	0.0389615	2.09566	246.308
Reference values	1.4	1.2	0.2	0.05	0.04	2.0	250.0

Table 3.1: The means of the benchmark test results for each parameter.

offset	E_h	E_a	E_v	R_i	R_o	R_c	V_{s3}
0%	0.0164	0.0274	0.0034	0.0013	0.0015	0.0309	0.7336
10%	0.0175	0.0228	0.0037	0.0012	0.0018	0.0355	1.2142
20%	0.0211	0.0219	0.0038	0.0011	0.0012	0.03	1.4752
40%	0.0181	0.0209	0.0034	0.001	0.0025	0.0303	1.9834
50%	0.0156	0.0216	0.0033	0.001	0.0032	0.0306	2.1167
60%	0.0136	0.0199	0.0031	0.0015	0.0029	0.0279	12.3652

Table 3.2: The standard deviations of the benchmark test results for each parameter.

offset	E_h [%]	E_a [%]	E_v [%]	R_i [%]	R_o [%]	R_c [%]	V_{s3} [%]
0%	1.188	2.282	1.609	2.281	3.729	1.472	0.296
10%	1.262	1.89	1.749	2.182	4.605	1.684	0.494
20%	1.529	1.819	1.823	1.966	2.931	1.422	0.6
40%	1.303	1.741	1.601	1.849	6.207	1.447	0.808
50%	1.124	1.786	1.575	1.767	8.155	1.461	0.862
60%	0.988	1.666	1.487	2.751	7.417	1.333	5.02

Table 3.3: The coefficients of variation of the benchmark test results for each parameter.

The analysis of the benchmark results for SBV. Figure 3.10 contains the identification results of the benchmark test for the SBV parameter. The x axis shows the initial value of the SBV parameter for each configuration, while the y axis shows the result of the identification for the stressed blood volume. The red line represents the reference value for SBV, that was used to create the synthetic dataset. The plot assures the statements summarizing the benchmark test results. The optimum of the identified parameters is not at the reference value. Figure 3.10 shows that the SBV parameter is more robust

for higher than for lower initial values. Figure 3.11 shows additional information about the benchmark test. On the left of the Figure, the loss function values at the end of the identifications are shown. Based on the Figure, the initial parameter value does not cause any difference in the final loss value. This means, that the other six parameters compensate for the difference in the final SBV value. On the right of Figure 3.11, the iterations used for each identification are presented. An early stopping method was used which detects convergence and stops the identification process. The plot shows that starting the parameter farther from the optimal value results in higher training epoch numbers.

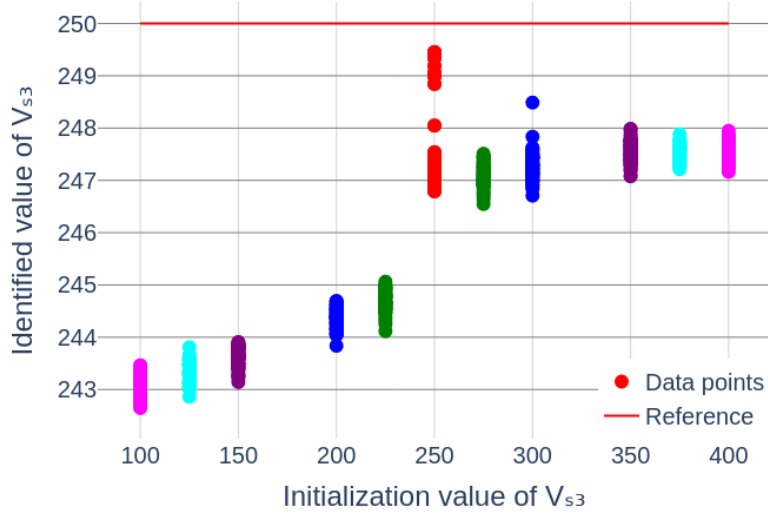


Figure 3.10: The initial and identified values of SBV in the benchmark test.

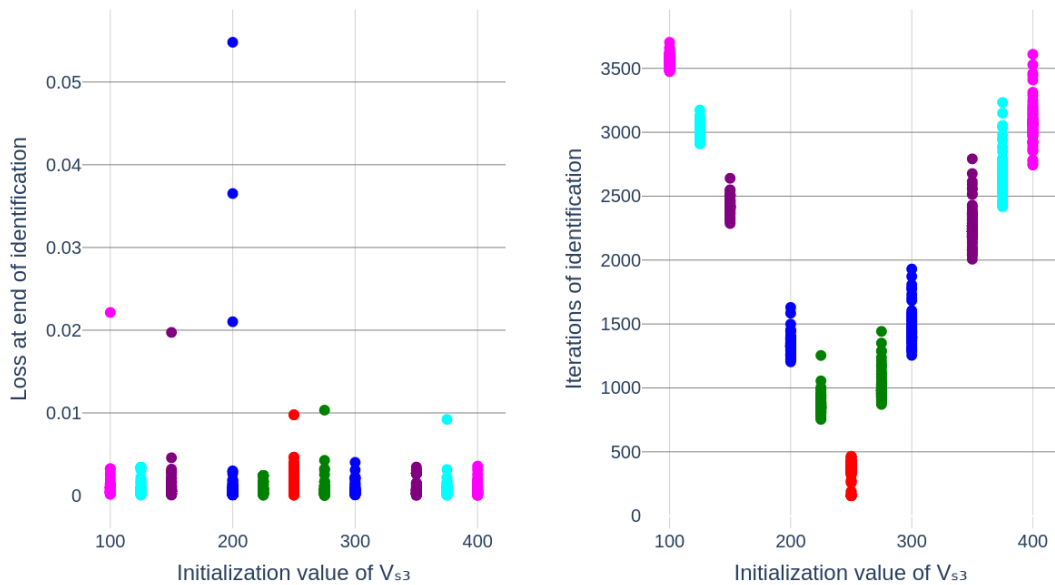


Figure 3.11: The final losses and number of iterations during learning as functions of initial parameter values of SBV in the benchmark test.

Table 3.4 summarizes the results of the benchmark test for the SBV parameter. The mean and standard deviation of the identified SBV values are shown as well as the mean and standard deviation of the error between the metrics of the identified model and the metrics of the synthetic dataset - the metrics are the elements of the output vector described in Section 3.2.1, and the mean squared error is used.

offset	mean of identified value	std of identified value	mean of error	std of error
0%	247.434	0.733568	0.00113481	0.0014412
10%	245.881	1.21422	0.000913275	0.00111032
20%	245.826	1.47519	0.00146624	0.0056749
40%	245.593	1.98344	0.00111674	0.00181506
50%	245.445	2.11669	0.00101705	0.0011429
60%	246.308	12.3652	0.00207306	0.0140853

Table 3.4: The benchmark results for Stressed Blood Volume parameter.

3.2.5 Evaluation of the results

The identified stressed blood volume parameters of the measurement are supposed to follow the trends of the stroke volume. The dataset used in this study contains measurements of the left ventricular volume, which can be used to derive the stroke volume.

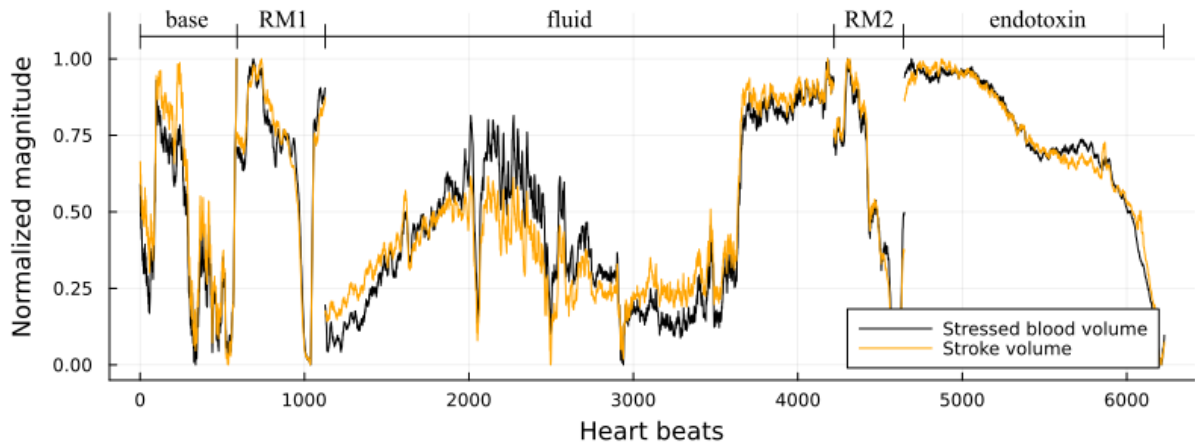


Figure 3.12: Identification results on the measurement data of subject #4 compared to stroke volume values derived from measurement data. The different intervention sections during the experiment are labeled: base; RM1 (first recruitment manoeuvre); fluid (fluid infusion); RM2 (second recruitment manoeuvre) and endotoxin (endotoxin infusion) – detailed in Section 2.1.

The identified parameters in each heartbeat are shown in Figure 3.12 – the stroke volume derived from the measurements in orange, and the stressed blood volume results of the developed identification method in black. A 10 sample moving average is applied

	Base	RM1	Fluid	RM2	Endotoxin
MSE	0.0106	0.0026	0.0085	0.0019	0.0011
COR	0.9834	0.9816	0.936	0.9945	0.9898

Table 3.5: The mean squared error (MSE) and the correlation (COR) between the normalized stroke volume derived from the measurements and the normalized stressed blood volume in each section of the measurement on subject #4.

to each curve to reduce the high frequency variations and improve the illustration of the trends. The curves were transformed to the $[0,1]$ interval to allow for visual comparison of the trends in the curves. On the x axis, the number of heart beats is shown. The Figure contains all measurements on the subject #4 of the experiment, which contained a base section where the subject was not under any intervention, then one recruitment manoeuvre, followed by a fluid therapy. After the fluid infusion, another recruitment manoeuvre was performed, which was followed by an endotoxin infusion that simulates cardiac shock and resulted in the death of the model animal. The results confirm the ability of stressed blood volume to predict fluid responsiveness as it fits to the stroke volume curve.

In Table 3.5, the mean squared errors, and correlation values are shown between the normalized stroke volume and normalized stressed blood volume. The quantitative evaluation of the result of the parameter identification shows high correlation between SV and the SBV model parameter.

4 Conclusion

The aim of this study was to develop and evaluate a NeuralODE-based method for the parameter identification of the Three Chamber Model as presented in detail in Section 3. First, the simulation of the cardiovascular model was implemented. Even though the calculation of the initial values of the parameters does not play a crucial role in the model simulation, the identification process highly depends on initial parameter values, and the simulations also hold more value if the correct parameters are used. The seven model parameter values are defined as functions of measurement data in Section 3.1.1. The differential equation describing the model is solved during the simulation of the TCM. The differential equation governing the model is derived from model equations in Section 3.1.2.

Based on the simulation, a NeuralODE-based parameter identification method was developed. In the process of enhancing the identification method, several challenges had to be overcome, namely (a) parameter transformation to avoid ODE instabilities, (b) definition of the appropriate hyperparameters of the identification process, and (c) developing an appropriate early stopping method. Analysing the results obtained using different solutions for each of these problems, the optimal configuration was developed which avoids instability, enables optimal running time and enhances convergence to the global minimum.

The study also aimed to evaluate the developed identification method. As presented in Section 3.2.4, a benchmark test was conducted using synthetic measurement data that allows the quantitative evaluation of the method. The results of the assessment demonstrated that the developed identification method exhibited robust performance with regard to the initial values of the parameters. Due to the clinical importance of the Stressed Blood Volume model parameter, the results of the benchmark test with regard to this parameter were analysed more extensively. The study concludes that the method is robust to reasonably high deviations in the initial parameter value compared to the reference value, which makes the method applicable in real-life situations where the optimal initial value is not known.

Finally, in Section 3.2.5, the developed identification is applied to measurement data, and the resulting SBV curve is compared to the stroke volume curve derived from measurement data. Based on the evaluation results, the model parameter shows strong correlation with the measurement data, which assures the applicability of the method in real-life fluid responsiveness prediction.

4.1 Limitations and future work

The study presents an evaluation of the developed parameter identification method on a dataset comprising measurements from an animal experiment. The data set contains

measurements taken from subjects in a variety of physiological states, rendering it diverse and suitable for the research in question. To validate the developed method, further research is required utilising a larger dataset, including measurements on human subjects.

Future work will include the development of a method capable of identifying the parameters of the Three Chamber Model using only data that is commonly available in clinical settings. The objective of the project, in which the current research is embedded, is to develop an artificial intelligence-based semi-automated fluid responsiveness assessment method that can be used in intensive care units. This requires the interconnection of the Tube-load model and the Three Chamber Model, as well as the parallel identification of these physiological models.

Acknowledgement

The research is supported by the EKÖP-24-2-BME-362 University Research Scholarship Programme of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.



Reference

- [1] R. G. R. Smith, *Model-based hemodynamic management of critically ill patients*. PhD thesis, University of Canterbury, 2022.
- [2] A. Pironet, P. Dauby, J. Chase, S. Kamoi, N. Janssen, P. Morimont, B. Lambermont, and T. Desaive, “Model-based stressed blood volume is an index of fluid responsiveness,” *IFAC-PapersOnLine*, vol. 48, pp. 291–296, 12 2015.
- [3] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” *CoRR*, vol. abs/1806.07366, 2018.
- [4] A. Messina, F. Collino, and M. Cecconi, “Fluid administration for acute circulatory dysfunction using basic monitoring,” *Ann Transl Med*, vol. 8, p. 788, June 2020.
- [5] R. Smith, L. Murphy, C. G. Pretty, T. Desaive, G. M. Shaw, and J. G. Chase, “Tube-load model: A clinically applicable pulse contour analysis method for estimation of cardiac stroke volume,” *Computer Methods and Programs in Biomedicine*, vol. 204, p. 106062, 2021.
- [6] Á. Antal, “Neural ode based cardiovascular model identification for personalized fluid resuscitation,” *Budapest University of Technology and Economics, Students’ Scientific Conference*, 2023.
- [7] L. Murphy, J. G. Chase, S. M. Davidson, R. Smith, and T. Desaive, “Minimally invasive model based stressed blood volume as an index of fluid responsiveness,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 16257–16262, 2020. 21st IFAC World Congress.
- [8] H. Suga, K. Sagawa, and A. Shoukas, “Load independence of the instantaneous pressure-volume ratio of the canine left ventricle and effects of epinephrine and heart rate on the ratio.,” *Circulation Research*, vol. 32, pp. 314–322, Mar. 1973.
- [9] C. Rackauckas, M. Innes, Y. Ma, J. Bettencourt, L. White, and V. Dixit, “Diffeqflux.jl - A julia library for neural differential equations,” *CoRR*, vol. abs/1902.02376, 2019.
- [10] L. Prechelt, *Early Stopping - But When?*, pp. 55–69. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.
- [11] A. Pironet, P. D. Docherty, P. C. Dauby, J. G. Chase, and T. Desaive, “Practical identifiability analysis of a minimal cardiovascular system model,” *Computer Methods and Programs in Biomedicine*, vol. 171, pp. 53–65, 2019.
- [12] C. Rackauckas and Q. Nie, “DifferentialEquations.jl—a performant and feature-rich ecosystem for solving differential equations in Julia,” *Journal of Open Research Software*, vol. 5, no. 1, 2017.

- [13] J. Revels, M. Lubin, and T. Papamarkou, “Forward-mode automatic differentiation in Julia,” *arXiv:1607.07892 [cs.MS]*, 2016.
- [14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980 [cs.LG]*, 2017.
- [15] A. F. Agarap, “Deep learning using rectified linear units (relu),” *arXiv:1803.08375 [cs.NE]*, 2019.