

Characterization of left ventricle sarcomere properties using deep learning

by

Dr. Miquel Crusells Girona

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Medical Physics

Universidad Nacional de Educación a Distancia, UNED



Advisors:

Dr. Javier Bermejo Thomas
Dr. Pablo Martínez-Legazpi Aguilo

Spring 2023

Abstract

Characterization of left ventricle sarcomere properties
using deep learning

Dr. Miquel Crusells Girona

Independientemente de su origen, la disfunción diastólica está presente en la práctica totalidad de las enfermedades estructurales del miocardio. Por ello, es de especial relevancia para la práctica clínica el conocer los cambios que pueden producirse en las propiedades mecánicas del corazón en estas circunstancias, especialmente en el ventrículo izquierdo, más propenso al fallo al trabajar a mayor presión. Actualmente, existen modelos directos que relacionan las propiedades del sarcómero con las propiedades de las cámaras y su función. No obstante, dado que las propiedades del sarcómero son prácticamente imposibles de obtener *in vivo* o *in silico*, no existen modelos que solucionen el problema de calibración inverso. La presente tesina tiene como objetivo evaluar diferentes redes neuronales convolucionales que predigan las propiedades del sarcómero en base a pares de variables hemodinámicas sintéticas, obtenidas mediante CircAdapt, en una metodología de *transfer learning*. Los resultados de estos análisis muestran que, cuando las señales no incorporan ruido blanco, los niveles de exactitud son muy elevados, alrededor del 90%, para cualquiera de las parejas de variables analizadas. Al introducir ruido blanco en las señales, la combinación de presión y volumen del ventrículo izquierdo mantiene este elevado nivel de exactitud. La precisión disminuye al 75% al sustituir estas variables por otras surrogadas, que pueden obtenerse en la práctica clínica de manera no invasiva, como el *strain* o la presión arterial, con una subida del error relativo hasta el 15% con una probabilidad del 95%.

Irrespectively of its origin, diastolic dysfunction is present in virtually all structural myocardial diseases. For this reason, it is of high interest to clinicians to understand the changes that occur in the mechanical properties of the heart, with a special interest in the left ventricle, as it is more prone to failure given its high working pressure. Direct models exist that can relate sarcomere properties to chamber properties and their function. However, since sarcomere properties are virtually impossible to measure *in vivo* or *in silico*, no model exists to solve the corresponding inverse problem. The present thesis assesses several convolutional neural networks that target sarcomere mechanical properties using different pairs of synthetic hemodynamical variables, which have been obtained with the CircAdapt model, in a transfer learning approach. Results show that, when no white noise is considered, accuracy levels above 90% are observed regardless of the hemodynamical variables used in the training process. When white noise is introduced, LV pressure and volume maintain the same level of accuracy. However, this accuracy drops to 75% when the former variables are substituted by other surrogated ones, which can be measured with non-invasive procedures, such as strain or arterial pressure, with an increase in relative error of the parameters up to 15% with a 95% probability.

Contents

Contents	i
List of Figures	ii
List of Tables	iii
1 Introduction and state of the art	1
1.1 Motivation	1
1.2 Anatomy and physiology of the cardiac muscle	2
1.2.1 The heart and its physiology	2
1.2.2 Pressure-volume loop	4
1.2.3 PV loop analysis: mechanical properties of the left ventricle	6
1.3 Characterizing mechanical properties of the left ventricle: a state of the art	8
1.3.1 Theoretical model for diastolic pressure	8
1.3.2 Local vs global optimization	9
1.3.3 CircAdapt model and sarcomere properties	10
1.4 Convolutional neural networks	12
1.4.1 Principles of convolutional neural networks	12
1.4.2 Layers of convolutional neural networks	13
1.5 Objectives and scope	15
2 Methods and materials	17
2.1 Information technology	17
2.1.1 Code and libraries	17
2.1.2 Hardware	18
2.2 Definition of input signals	18
2.2.1 Input signals	18
2.2.2 Spectrogram and normalization	20
2.3 CNN architecture	21
2.3.1 Structure	21
2.3.2 Trainable parameters	22
2.4 Training method	24
2.4.1 Neural activation	24
2.4.2 Loss function and optimization algorithm	25
2.5 Hyperparameters	26

3	Results and discussion	28
3.1	Learning results	28
3.1.1	LV pressure vs LV volume	29
3.1.2	LV pressure vs septal strain	31
3.1.3	LV pressure vs LV strain	32
3.1.4	Arterial pressure vs septal strain	33
3.1.5	Arterial pressure vs LV strain	34
3.1.6	Summary of learning results	35
3.2	Discussion of the results	37
4	Conclusions and future research	40
4.1	Conclusions	40
4.2	Future lines of research	41
	Bibliography	42

List of Figures

1.1	Anatomy of the heart [22].	2
1.2	Configuration of the sarcomere [8].	3
1.3	Summary of events and evolution of hemodynamic variables for the cardiac cycle [14].	4
1.4	Typical PV loop for a healthy patient [14].	5
1.5	Relationship between the PV loop and the end-diastolic pressure-volume relationship (EDPVR) [7].	6
1.6	End-systolic and end-diastolic pressure-volume relationships [7].	7
1.7	LV pressure as a function of time and constant time of relaxation [11].	7
1.8	Decomposition of total pressure into an active pressure p_a and a passive pressure p_p [4].	9
1.9	Architecture of the CircAdapt model [1].	11
1.10	Schematic representation of the convolution of an image and a filter [25].	13
1.11	Neural configuration used in a fully-connected layer [16].	14
1.12	Schematic representation of a typical convolutional neural network combining different types of layers [23].	15
2.1	Representative PV loops with and without noise used in the training of the CNN.	19
2.2	Representative spectrogram of a LV pressure signal with noise.	21
2.3	ReLU and sigmoid activation functions [16].	25
3.1	Learning convergence for LV pressure - LV volume.	29

3.2	Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - LV volume calibration.	30
3.3	Learning convergence for LV pressure - septal strain.	31
3.4	Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - septal strain calibration.	32
3.5	Learning convergence for LV pressure - LV strain.	33
3.6	Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - LV strain calibration.	34
3.7	Learning convergence for arterial pressure - septal strain.	35
3.8	Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for arterial pressure - septal strain calibration.	36
3.9	Learning convergence for arterial pressure - LV strain.	37
3.10	Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for arterial pressure - LV strain calibration.	38

List of Tables

2.1	Proposed CNN architecture.	23
2.2	Proposed hyperparameters for each convolutional layer.	26
2.3	Proposed number of neurons for each fully-connected layer.	27
3.1	Summary of learning results for the assessed CNNs.	36
3.2	Summary of errors (70% and 95% probabilities) obtained in the parameters for the assessed CNNs.	37

Chapter 1

Introduction and state of the art

1.1 Motivation

Diastolic dysfunction is present to some degree in virtually all structural myocardial diseases and, independent of its primary cause, this condition has become a major source of cardiovascular morbidity and mortality [21]. Diastolic dysfunction is a generic term encompassing different abnormalities of the mechanical chamber properties of the left ventricle (LV). From a clinical perspective, this condition may resolve into elevated left atrial pressure, resulting in upstream pulmonary congestion and, eventually, heart failure.

For this reason, it is of high interest to clinicians to understand the changes that occur in the mechanical properties of the heart, since they are the major determinants of both diastolic and systolic function. While it is difficult and risky to measure LV intrinsic properties *in vivo*, an *operating* approach can be taken via catheterism by measuring and analyzing the corresponding pressure-volume (PV) loops. Current methods of analysis either attempt to isolate diastolic properties by focusing on specific temporal windows of diastole, in which independent regressions are sought [19], or attempt to calibrate properties with a global optimization method using complete pressure-volume curves for diastole [4]. Even though significant improvements have been made in the understanding of the relevance of diastolic chamber properties, up to this point all treatments to improve clinical outcome on the basis of modifying these properties have been unsuccessful [4].

Owing to the fact that chamber properties are intrinsically related to the mechanical behavior of sarcomeres, some multiscale models have been proposed to relate both quantities in a direct way [1, 2]. However, since these sarcomere properties are virtually impossible to measure *in vivo* and are decoupled from chamber ones, no model exists to solve the corresponding inverse problem, and therefore the relation between sarcomere properties and chamber properties is not fully disclosed.

While classical methods of optimization have proven useful for the calibration of mechanical properties of the heart, new machine learning algorithms, in particular deep learning techniques, are currently emerging in a multitude of medical disciplines [9]. It is thus mandatory to analyze the accuracy and efficiency that these new methods may

provide to the search of the relationship between sarcomere properties, chamber properties and PV loops.

1.2 Anatomy and physiology of the cardiac muscle

To begin with, some preliminary concepts regarding the anatomy and physiology of the heart are introduced in order to better understand the calibration problem at hand.

1.2.1 The heart and its physiology

The heart is a fist-sized organ that pumps blood throughout the body. Anatomically speaking, the heart is composed by four main chambers that are made of cardiac muscle and are powered by electrical impulses. These chambers are, as shown in Fig. 1.1, the right and left atriums, and the right and left ventricles.

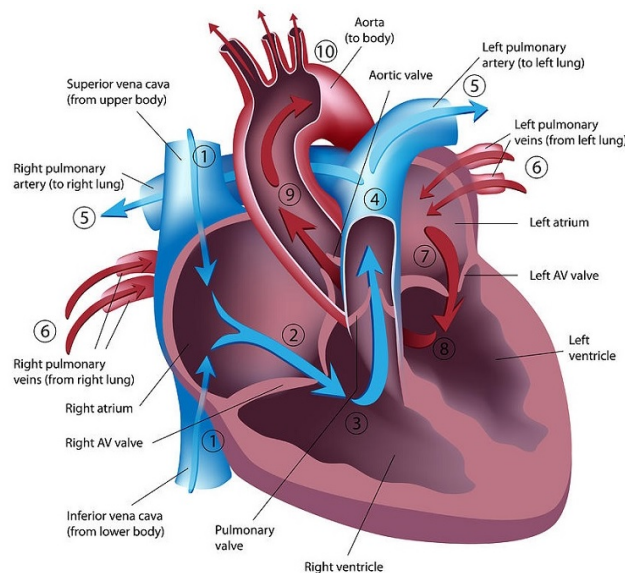


Figure 1.1: Anatomy of the heart [22].

The smallest functional unit of cardiac muscle, and striated muscle in general, is the sarcomere, which corresponds to the segment of a myofibril between two Z disks, as shown in Fig. 1.2. Sarcomeres are composed of long fibrous filament proteins that produce contraction and relaxation of the muscle with the slide of thick filaments (myosin) past thin filaments (actin).

As depicted in Fig. 1.1, venous blood returning from the body (1) drains into the right atrium (2) via the superior and inferior vena cava and coronary sinus. The right atrium pumps this blood through the tricuspid valve into the right ventricle (3) which, in turn, pumps the blood through the pulmonary semilunar valve into the pulmonary arteries (4,5) to be oxygenated in the lungs. Blood returning from the lungs via the four

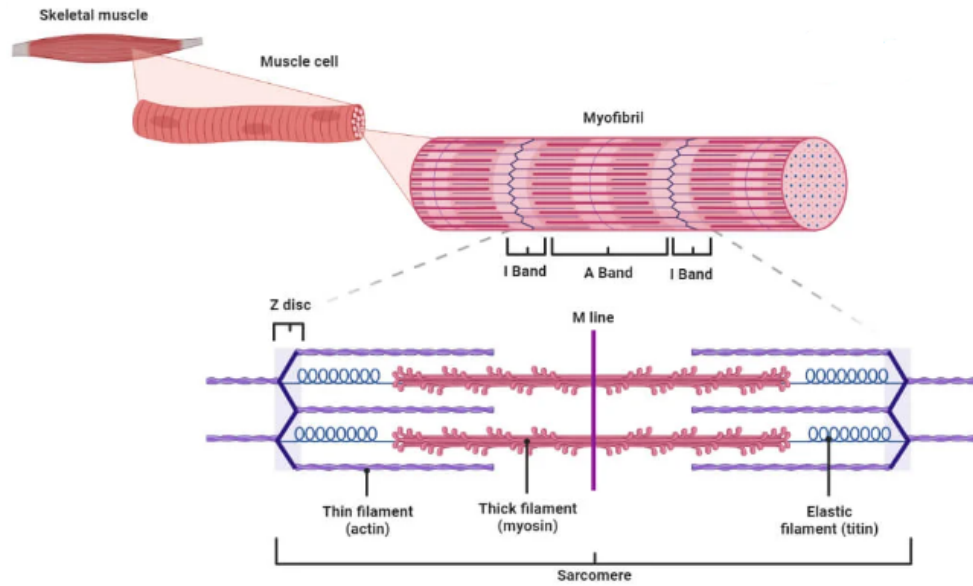


Figure 1.2: Configuration of the sarcomere [8].

pulmonary veins (6) drains into the left atrium (7), and this left atrium pumps the blood through the mitral valve into the left ventricle (8). Finally, the left ventricle pumps the blood through the aortic semilunar valve into the ascending aorta (9,10) and finally to the systemic circulation. This process is known as the *cardiac cycle* and conforms the main purpose of the heart: to be the blood pump for the human body.

Since filling and ejection of blood in the heart occur simultaneously in both sides (left and right), the cardiac cycle is usually divided into two parts, which are of utmost importance for the problem at hand: *systole* and *diastole*. During ventricular systole, both ventricles contract and vigorously eject blood from the heart while the two atria are relaxed (atrial diastole). On the other hand, during ventricular diastole, the heart relaxes and expands, receiving blood into both ventricles (early diastole), followed by the contraction of the two atria that pump blood into their respective ventricles (atrial systole). As a result, ventricular systole is a succession of three steps: an isovolumetric contraction followed by a rapid ejection of blood, with a final reduced ejection, and often represents around 38% of the cardiac cycle in healthy patients. On the contrary, ventricular diastole starts with an isovolumetric relaxation followed by a rapid filling, and continues with a slower filling and a final atrial systole. Ventricular diastole lasts about 62% of the cardiac cycle in healthy patients, and is hence responsible for the blood filling of the heart. Fig. 1.3 shows a summary of the events and evolution of principal hemodynamic variables during the cardiac cycle.

The cardiac function is often analyzed by means of four relevant properties that are related to its correct performance:

- The *end-systolic volume* (ESV) is the volume of blood left in a ventricle at the end of systole, and is usually around 40-60 mL.

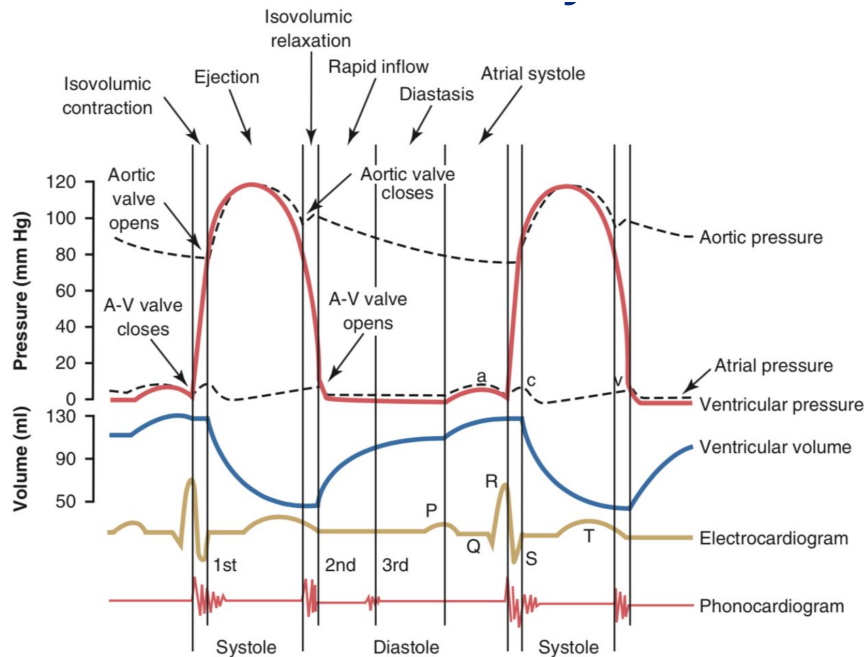


Figure 1.3: Summary of events and evolution of hemodynamic variables for the cardiac cycle [14].

- The *end-diastolic volume* (EDV) is the volume of blood in a ventricle at the end of diastole. Note that this is the maximum volume that can be stored in the heart, normally around 120-130 mL.
- The *stroke volume* (SV) is the total volume of blood ejected by a ventricle during systole, and can thus be computed by the difference $SV = EDV - ESV$. A typical value is around 70 mL.
- The *ejection fraction* (EF) corresponds to the ratio of stroke volume to end-diastolic volume and thus represents the total blood that is mobilized in a heartbeat. The EF for healthy patients is usually between 60-65%.

1.2.2 Pressure-volume loop

A useful operating diagram of the cardiac cycle consists of representing the left-ventricle (LV) pressure as a function of the left-ventricle (LV) volume. This plot is often called a pressure-volume (PV) loop. Fig. 1.4 illustrates a typical PV loop for a healthy patient. The PV loop representation allows one to rapidly identify the main parts of the cardiac cycle.

- Phase I (segment AB) corresponds to the filling phase of the cardiac cycle. In this part of the cycle, the LV pressure is low and approximately constant (around 2-3 mmHg), while the volume rises to EDV. This phase starts with the opening of the mitral valve (point A) and finalizes with the closure of this same valve (point B).

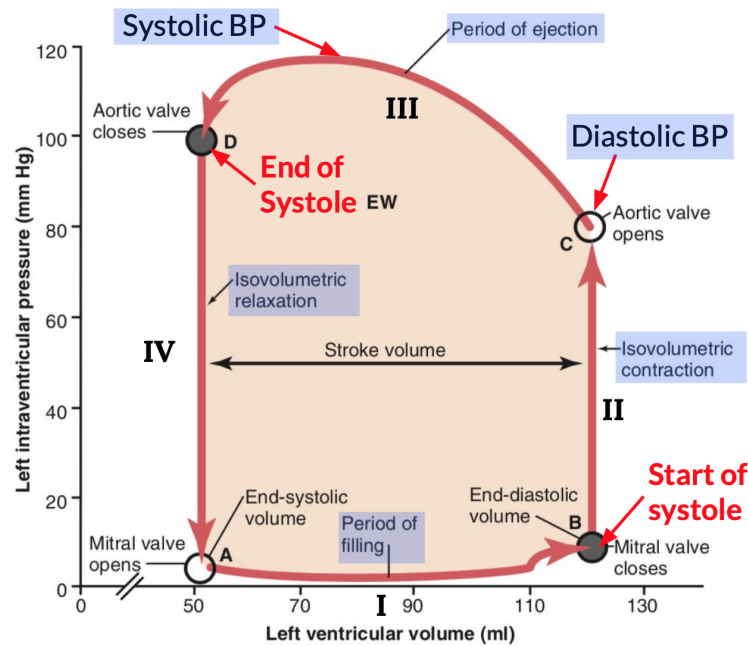


Figure 1.4: Typical PV loop for a healthy patient [14].

- Phase II (segment BC) corresponds to the isovolumetric contraction of the cycle. In this part, the LV volume remains approximately constant at EDV while the pressure increases rapidly to around 80 mmHg. This phase starts when all cardiac valves are closed (point B) and ends with the opening of the aortic valve (point C), point at which the diastolic blood pressure (BP) is measured.
- Phase III (segment CD) corresponds to the ejection phase of the cardiac cycle. In this part, the LV volume decreases to ESV while pressure rises to a peak of around 120 mmHg and then slowly decreases to about 100 mmHg. This phase starts with the opening of the aortic valve (point C) and remains until its closure (point D). The maximum pressure in this phase is the common measurement of systolic blood pressure (BP).
- Phase IV (segment DA) corresponds to the isovolumetric relaxation of the cycle. In this part, the LV volume remains approximately constant at ESV and the pressure quickly drops to 2-3 mmHg. This phase starts with the closure of the aortic valve (point D) and finishes at the opening of the mitral valve (point A).

The PV loop representation is extremely useful in so that it allows a precise mathematical formulation of the cardiac cycle. From this mathematical formulation, the principal mechanical properties of the left ventricle can be extracted.

1.2.3 PV loop analysis: mechanical properties of the left ventricle

From the PV loop, it is possible to derive some mechanical properties that will allow clinicians to assess the cardiac performance of both chambers¹.

The first relevant mechanical property is related to the end-diastolic pressure-volume relationship (EDPVR). At end-diastole, the left ventricle is completely relaxed. If blood is pumped inside this relaxed ventricle, there is no resistance at first, up to a volume V_0 , usually called *equilibrium volume* or *unstressed volume*, which corresponds to the maximum volume at which LV pressure is still zero. From this point on, an increasing resistance is observed in order to expand the ventricle, which is defined in the relationship between the end-diastolic pressure and volume, namely, the EDPVR. As a result, the moment of end-diastole for a given heartbeat is necessarily a point of the EDPVR, as shown in Fig. 1.5.

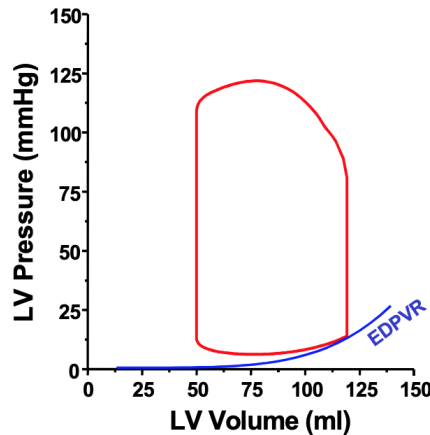


Figure 1.5: Relationship between the PV loop and the end-diastolic pressure-volume relationship (EDPVR) [7].

In a similar way, one can analyze the capacity of the left ventricle to pump blood. Analogously to the former definition, one can define the end-systole pressure-volume relationship (ESPVR) as all possible points of end-systole. It is relevant to note that, since a V_0 volume is required for a positive pressure, ESPVR and EDPVR intersect at point $(V_0, 0)$, as observed in Fig. 1.6.

As a result, the *elastance* E of the left ventricle can be defined as the derivative of LV pressure with respect to LV volume, and is usually measured either at end-systole, in order to assess the pumping capacity of the heart, or at end-diastole, in order to assess its mechanical recovery

$$E = \frac{\partial P}{\partial V} \quad (1.1)$$

¹Note that, in the present work, we have only focused on the left ventricle.

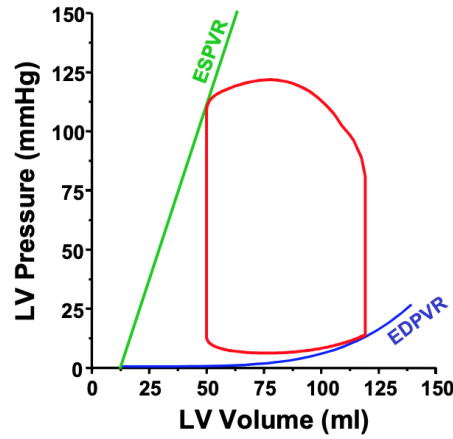


Figure 1.6: End-systolic and end-diastolic pressure-volume relationships [7].

The next relevant mechanical property refers to how fast LV pressure decreases during isovolumetric relaxation (phase IV in Fig. 1.4). For this phase, one can plot LV pressure as a function of time and mathematically fit the resultant curve, as shown in Fig. 1.7.

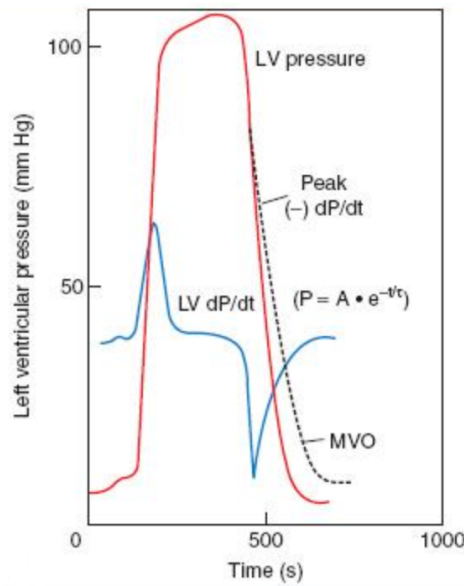


Figure 1.7: LV pressure as a function of time and constant time of relaxation [11].

Since the actual curve $P(t)$ during isovolumetric relaxation approximately behaves like an exponential, the mechanical property defining the relaxation process of the left ventricle can be thought as the time constant of this decay. For this reason, the time constant of relaxation, or simply the relaxation index, τ , is often defined as

$$\tau = \max_t \left(-\frac{P}{dP/dt} \right) \quad (1.2)$$

As will be discussed in subsequent sections, this exponential behavior admits two different models: one with a zero asymptote and one with a non-zero asymptote. It is relevant to note that both of these approaches are compatible with the definition in Eq. 1.2.

1.3 Characterizing mechanical properties of the left ventricle: a state of the art

Due to the relevance that left-ventricle mechanical properties have in the assessment of a healthy heart both in terms of systolic function and diastolic function, several models have been proposed.

1.3.1 Theoretical model for diastolic pressure

A well-accepted model to describe diastolic pressure considers the addition of an active pressure p_a and a passive pressure p_p , both acting through the full diastolic period [4].

In this model, active relaxation pressure p_a is modeled by an exponential function

$$p_a(t, V) = (P_0 - p_p(V)) \cdot e^{-t/\tau} \quad (1.3)$$

where P_0 measures LV pressure at the onset of diastole, t is the diastolic time measured from end-systole to mitral valve closing and τ is the time constant of relaxation (Eq. 1.2). Eq. 1.3 has an asymptote at $p_a \rightarrow 0$ when $t \rightarrow \infty$. Oftentimes, this behavior is not considered appropriate and a twitch independent tension, a *resting tone* R_t , is added to the formulation, so that $p_a \rightarrow R_t$,

$$p_a(t, V) = (P_0 - p_p(V)) \cdot e^{-t/\tau} + R_t \quad (1.4)$$

Passive mechanical diastolic properties are governed by a piecewise PV logarithmic relationship,

$$p_p(t, V) = \begin{cases} S_- \log \frac{V - V_d}{V_0 - V_d} & \text{when } V < V_0 \\ -S_+ \log \frac{V_m - V}{V_m - V_0} & \text{when } V > V_0 \end{cases} \quad (1.5)$$

where S_- and S_+ represent the stiffness constants for each piecewise function, V_d is the LV dead volume asymptote and V_m is the maximal achievable volume asymptote. Note that, when $V = V_0$, $p_p = 0$. Given that the slope of p_p should be continuous for a relatively homogeneous body undergoing elastic deformation, S_- can be expressed in terms of S_+ as

$$S_- = S_+ \frac{V_0 - V_d}{V_m - V_0} \quad (1.6)$$

and thus decreasing the number of independent parameters of the model by one.

Active and passive mechanical properties act simultaneously during the full diastolic period, from aortic valve closure (AVC) to mitral valve opening (MVO). Therefore, the estimated diastolic pressure P is a function of time t and volume V

$$P(t, V) = p_a(t, V) + p_p(t, V) \quad (1.7)$$

Fig. 1.8 shows the contribution² of the active and passive pressures to give the total pressure for a PV loop.

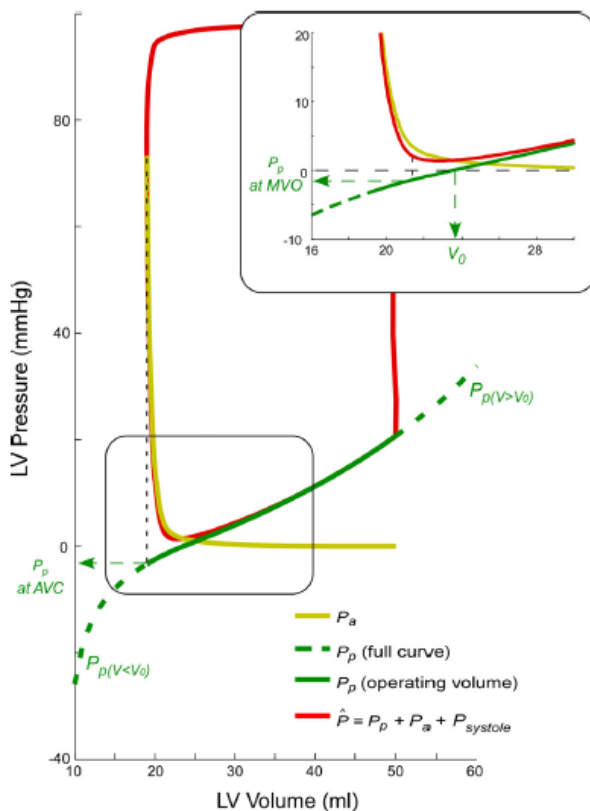


Figure 1.8: Decomposition of total pressure into an active pressure p_a and a passive pressure p_p [4].

1.3.2 Local vs global optimization

The equations presented in Section 1.3.1 define a model that can be calibrated for the diastolic properties τ , V_0 , S_+ , V_m and V_d . Indeed, given a series of PV loops, a least-squares regression can be implemented to approximate the former chamber properties. Nonetheless, two approaches have been proposed to implement the regression of these variables: a local [19] and a global optimization [4].

Given that passive volume-dependent forces are low during the isovolumetric period of diastole, a local calibration of the rate of relaxation can be performed during this particular time window of the cardiac cycle. Analogously, assuming that relaxation has completed at end-diastole, parameters coming from passive pressure can be calibrated close to end-diastole. Even though this local optimization procedure gives a relatively

²Note that, in order to generate the complete heartbeat, $P_{systole}$ is added to the diastolic pressure $P = p_a + p_p$.

good approximation of the chamber properties, it is worth noticing that it penalizes discriminatory power, since only a few data points are used to calibrate the properties, while highly depending on the definition of the fitting windows.

In reality, both active and passive contributions act at the same time for the whole diastolic period. For this reason, a global optimization procedure was proposed by Bermejo *et al* [4]. In this approach, the full pressure $P(t, V)$ (Eq. 1.7) is included in a global optimization algorithm with cost function C

$$C = \|P(t, V) - P_{real}\|_2^2 \quad (1.8)$$

where P_{real} stands for the true measured pressure and $\|\cdot\|_2$ is the L^2 -norm of the corresponding vector. It is interesting to note that this method can be used heartbeat-by-heartbeat or it can fit a complete hemodynamic run of inferior vena cava (IVC) occlusion.

While the former two methods evaluate the chamber mechanical properties, it is relevant to note that they lack information about the sarcomere function, which may be of paramount importance for selecting the appropriate targets for pharmacological treatments.

1.3.3 CircAdapt model and sarcomere properties

In order to include more sophisticated constitutive models, a multiscale model called *CircAdapt* was proposed by Arts *et al* [1] based on the interaction of mechanics and hemodynamics of the left ventricle [18]. This CircAdapt model simulates beat-to-beat dynamics of the four-chamber heart with systemic and pulmonary circulation. Being a multiscale model, it incorporates tissue mechanics and its adaptation to mechanical loads, and thus takes into consideration a realistic relation between pressure and volume load.

As described in Ref. [1], the model is based on several rules to combine all levels of the model.

1. For blood vessels: low shear stress dilates the wall, while tensile stress thickens the wall.
2. For myocardial tissue:
 - Strain dilates the wall material.
 - Larger maximum sarcomer strength increases contractility.
 - Contractility increases wall mass.

As far as the implementation is concerned, the CircAdapt model is formed by four types of modules: chamber, tube, valve and resistance. Simulating human anatomy, the heart contains four chambers, and their disposition consists of a large outer chamber (right ventricle) encapsulating a high-pressure-generating inner chamber (left ventricle) to accomodate right-to-left ventricle interaction. The large arteries and veins that are connected to the heart are simulated by nonlinear compliant tubes with a nonlinear characteristic impedance for pressure-flow waves. Chambers and tubes are connected by valve elements whose formulation includes terms for inertia and Bernoulli losses. These

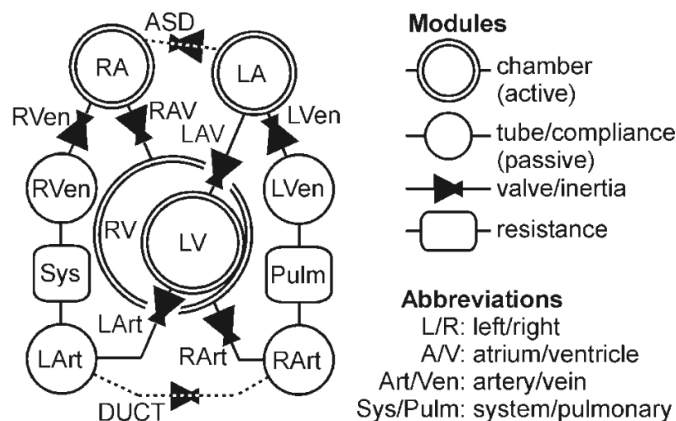


Figure 1.9: Architecture of the CircAdapt model [1].

valve elements can also accommodate defects such as venous outflow into atria or septal defects. The general architecture of the model is shown in Fig. 1.9.

From a computational point of view, the CircAdapt model is formulated by means of a system of differential equations in state variables with respect to time, so that the model generates traces of hemodynamic variables as a function of time for the cardiac cycle. The constitutive equations³ for the sarcomere mechanics are also decomposed into an active and a passive part. On the one hand, the active stress due to myofiber contraction σ_{fa} is a function f of time t , sarcomere length l_s , contractile element length l_{si} and contractility C , multiplied by a scaling factor Sf_{act} ,

$$\sigma_{fa} = Sf_{act} \cdot f(t, l_s, l_{si}, C) \quad (1.9)$$

where C represents the density of cross-bridges per unit length and its dynamics depend on time constants of rise (t_r) and decay (t_d). From Eq. 1.9, one can define the sarcomere properties of relaxation⁴ η_s and contractility ϵ_s by means of the decay time t_d and the slope of the active stress σ_{fa} , respectively.

On the other hand, the passive stress, arising from internal structures such as titin and the extracellular matrix surrounding the myocyte, is described as a nonlinear relation depending on sarcomere length l_s ,

$$\sigma_{fp} = Sf_{pas} \cdot \left(\left(\frac{l_s}{l_{s0}} \right)^{k_1} - 1 \right) \quad (1.10)$$

where Sf_{pas} is a scaling factor, l_{s0} represents stress-free fiber length and k_1 is an exponent which determines the degree of nonlinearity. Note that Eq. 1.10 allows one to define the passive stiffness of the sarcomere k_s as the slope of σ_{fp} .

³Developments for the present thesis only consider sarcomere properties; however, extensive detail of the formulation for all components can be found in Ref. [1].

⁴The subscript "s" will be dropped for simplicity in Chapter 3.

The total fiber stress σ_f generated by the myocardial fibers is thus the composition of the two contributions

$$\sigma_f = \sigma_{fa} + \sigma_{fp} \quad (1.11)$$

As a result, the CircAdapt model becomes a useful tool in order to simulate PV loops with the sarcomere properties as input variables. Obviously, these sarcomere properties are extremely difficult - virtually impossible - to measure in real patients, and would thus be a significant contribution to design an inverse algorithm to calibrate them from real PV data.

1.4 Convolutional neural networks

Deep learning, a subfield of machine learning, has seen a dramatic increase in popularity during the recent years, mostly because of the improvements in computational power and the availability of massive datasets. The greatest successes in this field have come from the area of computer vision, and healthcare and medicine have found good use to it in the diagnostics of diseases with medical imaging [9]. In this regard, convolutional neural networks (CNNs), a type of deep learning technique designed to process data with spatial invariance, have been established as one of the most promising algorithms for medical applications.

Convolutional neural networks (CNNs) are usually used for classification problems; however, recent developments in deep learning use these type of networks to calibrate parameters with time-dependent variables [5, 17, 20, 26]. As a result, it would be a relevant contribution to assess the possibilities that these new methods may provide to the inverse problem at hand.

1.4.1 Principles of convolutional neural networks

Traditional neural networks are modeled on the human brain, whereby neurons are stimulated by connected nodes and are only activated when a certain threshold value is reached. There are two main drawbacks of these networks when it comes to image⁵ processing.

First, these traditional networks process each image pixel by pixel. As a result, the number of weights that are necessary to learn from the images rapidly grows unmanageable when the size of the image increases. Consequently, overfitting becomes a problem when training these types of networks.

Another problem of traditional neural networks is that they are not translational invariant. This statement means that these networks react differently to an image and its shifted version. Given that pixels are processed one at a time, when a network that has been trained with an image encounters a shifted version of this same image, it will try to correct itself, instead of trying to learn the pattern that the figure includes. This

⁵The discussion is based on image processing; however, it applies to any grid-like structured data, as is the case of spectrograms or any other sort of contour plots.

phenomenon results in the weights being too sensitive to the particular image instead of to the pattern that it represents, thus generating significant inaccuracy.

In order to solve the former disadvantages and induce the network to learn the patterns instead of the pixels, the idea of using a convolution arised. A convolution is a mathematical operation that acts as a filter in so that an array f , usually called the convolution filter, is multiplied elementwise by a portion of the image, as shown in Fig. 1.10. As a result, convolution filters act as kernels in the processing of digital images [12]. It is for this reason that convolutional architectures are used to detect patterns in images. It is interesting to note though that the powerfulness of convolutional neural networks is that the coefficients of the filters can in fact be trained, and thus patterns are *learned* instead of *looked for* as in digital image processing.

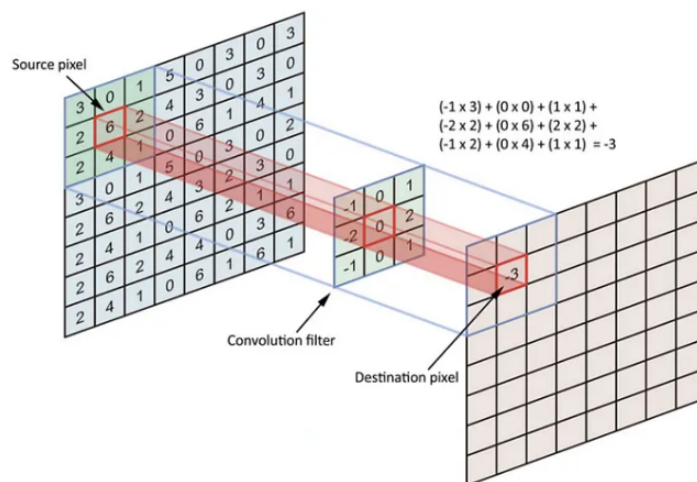


Figure 1.10: Schematic representation of the convolution of an image and a filter [25].

As a result, concatenating convolution operations allows the network to detect patterns with a relatively small number of parameters. This fact both reduces the probability of overfitting and keeps the network simple.

1.4.2 Layers of convolutional neural networks

Convolutional neural networks (CNNs) always include the convolution operator at some point in the architecture; however, they can be decomposed in several types of layers that have different purposes.

The first and most important type of layer in a CNN is the convolutional layer. This layer performs the convolution between a filter and an image, and its objective, as stated before, is to extract patterns from the image. In this setup, filters are not preset, but learned by the training set, and there are hence as many trainable parameters in the layer as filter coefficients plus one bias parameter per node. This layer can also include padding or use a stride. In the first case, extra rows and columns are used around the image so that the size of the output does not decrease excessively, while in the second

case the convolution operation *jumps* as many elements as the stride instead of following consecutive elements from the input image. In terms of output size, if the input image is $n \times n$ and the convolutional layer uses a square filter of dimension f with padding p and stride s , the size S of the output image can be computed as

$$S = \frac{n + 2p - f}{s} + 1 \quad (1.12)$$

Once the convolution is computed, an activation function is usually applied element-wise. This activation function establishes a threshold to give a nonzero value at that particular element. Traditional activation functions are used in convolutional layers, such as *rectified linear units* (ReLU) or sigmoids, depending on the needs of each particular application.

Another sort of layer that is often used in CNNs is a pooling layer. This layer is used to condense the information of an image by *summarizing* its elements with a particular rule. One possibility that is extensively used is *max-pooling*, for which the maximum value of a subimage is selected. It is relevant to note that this type of layer does not learn any feature of the image, as it only condenses the input information. As a result, pooling layers do not have trainable parameters.

Another important layer that can be added to a CNN is a dropout layer. This layer randomly sets input elements to zero with a predefined frequency d_r at each step during training, which helps prevent overfitting [3, 24]. It is relevant to note that, in order to maintain the intensity of the image, inputs that are not set to zero are scaled up by $1/(1-d_r)$, such that the sum over all inputs is unchanged. As a result, no parameters need to be trained in this layer, and d_r acts a hyperparameter that needs to be set beforehand by the user.

If the image needs to be stored in a unidimensional form, a flatten layer is usually added to the CNN. As a result, the function of this layer is to simply relocate all pixels of all channels of an image in a one-dimensional vector disposition. Consequently, no parameters need to be trained in this layer.

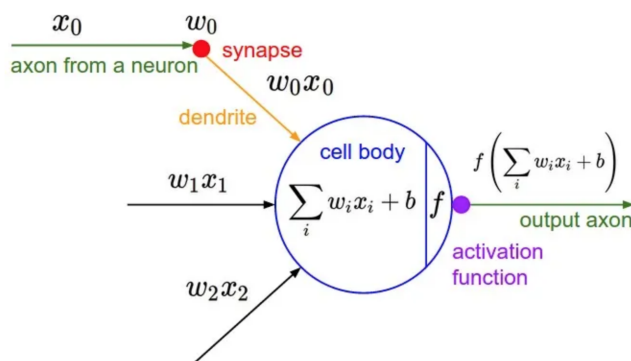


Figure 1.11: Neural configuration used in a fully-connected layer [16].

Finally, a fully-connected layer is a layer that has weights w_i for all possible connections between all elements x_i of the input vector and each neuron, accommodating an

activation function f that establishes a threshold to give a nonzero value, as shown in Fig. 1.11. In fact, this layer acts as a traditional neural network, for which all weights plus a bias in each neuron need to be trained.

In reality, a convolutional neural network includes one or several layers corresponding to the types that have been detailed above. This structure is often known as the *architecture* of the CNN, and, as experience demonstrates, is extremely application-dependant. As an example, Fig. 1.12 shows a schematic representation of a typical convolutional neural network combining the different types of layers.

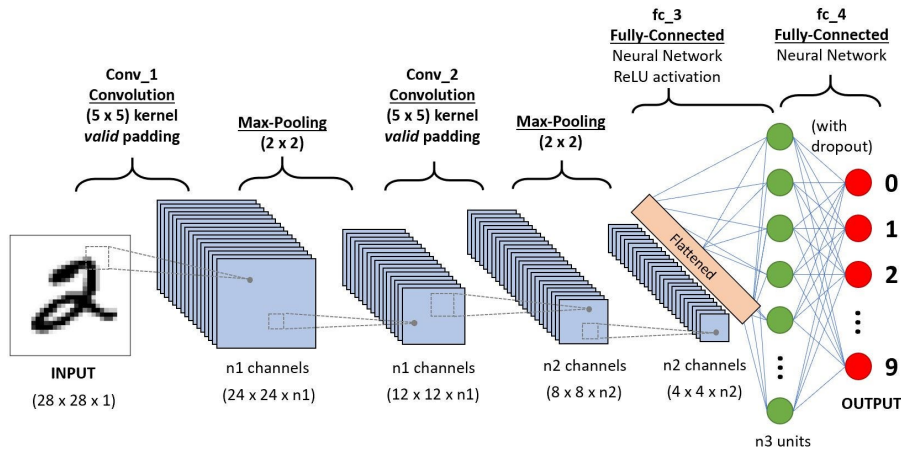


Figure 1.12: Schematic representation of a typical convolutional neural network combining different types of layers [23].

1.5 Objectives and scope

Some of the most relevant inputs that clinicians have to assess systolic and diastolic function are the mechanical properties of the heart. However, the measurement of these properties is usually complicated and requires invasive tests for patients. Because of this reason, several models have been proposed to obtain these properties from real patients' data by means of the least possible invasive techniques, such as ecocardiography imaging. Furthermore, even though it is virtually impossible to measure sarcomere properties in an experimental way, recent developments have found direct relationships between chamber properties and sarcomere properties [1, 18], while no model has been derived for the corresponding inverse problem.

Since recent deep learning applications have been successfully applied in calibration problems with time-dependent variables [5, 17, 20, 26], the present thesis aims at finding a useful convolutional neural network architecture that can be applied to PV loops' data in order to infer heart sarcomere properties. Since no real data is available to calibrate the neural network, the CircAdapt model [1] is used to simulate numerous syntethic

heartbeats, and these are used as input data for the calibration of the neural network in a transfer learning approach.

With this methodology, the proposed thesis also assesses several combinations of hemodynamical variables as explanatory variables to sarcomere properties in order to find the most relevant model in terms of accuracy and clinical simplicity.

Chapter 2

Methods and materials

2.1 Information technology

The following section presents the information technology that has been used to design and train the proposed convolutional neural networks.

2.1.1 Code and libraries

The programming language that is mostly used to design and train deep learning models is currently *python*, which is indeed the one that has been used in the present work. In particular, the code has been compiled with python 3.11. There are, however, two main deep learning libraries that are available in python: Tensorflow and Pytorch. While the first one, developed by Google, was the very first package that allowed the deployment of deep learning models, the second one, developed by Facebook, is getting more and more attention in the present days. For the problem at hand, there are not significant differences between the two, and Tensorflow has been selected for the coding. In particular, the API *tensorflow.keras* is used to design the layers and train the network.

Apart from the deep learning library, several additional libraries are required to manipulate the data and compute the required quantities. Other relevant libraries that are used in the present work are listed below.

- *numpy* and *scipy* are used to perform mathematical operations. In particular, the function *spectrogram* from the package *scipy.signal* is used to obtain the spectrograms of the different signals.
- *pandas* is used to structure results into dataframes.
- *sklearn.train_test_split* is used to fix a random split of the input sample into a train and a test sample.

2.1.2 Hardware

The second relevant IT aspect is the hardware in which the training and evaluation of the CNN has been performed. All trainings of the CNNs that are assessed in the present work have been performed in a local machine with 16 Gb of RAM and a CPU Intel(R) Core(TM) i5-9400 CPU @ 2.90GHz. With this hardware, each CNN training has taken about an hour of computation time.

2.2 Definition of input signals

The first step in the definition of a convolutional neural network (CNN) consists of preparing the input variables in an appropriate format. Given that convolutional neural networks require a vast amount of data to report meaningful results, and that sarcomere properties are virtually impossible to measure *in vivo* or *in silico*, there have been no possibility to train the proposed CNN with real data. For this reason, a transfer learning approach has been selected. With this approach, input variables are obtained numerically with the multiscale model CircAdapt [1], and are subsequently used to train the neural network. This approach has indeed proven successful in calibration studies with time-dependent variables [17, 20, 26].

2.2.1 Input signals

As mentioned previously, input variables are obtained with the multiscale model Circadapt [1]. Using different sarcomere properties (stiffness, relaxation and elastance) as an input to this model, 44.315 heartbeats have been computationally simulated, and results have been saved in a data structure with the following fields:

- LVP_F corresponds to the pressure inside the left ventricle.
- AO_P_F corresponds to the aortic pressure, oftentimes referred to as arterial pressure in this document.
- LVV_F is the volume of the left ventricle.
- Septal_Strain_F represents the strain of the heart septum.
- LV_Strain_F is the strain of the left ventricle wall.
- time_F corresponds to the time scale of the heartbeat.
- StiffLV_F is the stiffness k_s of the sarcomere according to Section 1.3.3, and corresponds to the first target variable of the model.
- RelaxLV_F is the relaxation η_s of the sarcomere according to Section 1.3.3, and corresponds to the second target variable of the model.
- ElastanceLV_F is the elastance or contractility ϵ_s of the sarcomere according to Section 1.3.3, and corresponds to the third target variable of the model.

Once the relevant data is stored, a data quality audit is performed to ensure that it does not contain missing or invalid values, which is a crucial step in order to obtain meaningful results with machine learning algorithms. In this process, problematic data points were removed from the calibration sample.

Once the input data has been purged from errors, a random extraction of 37.000 heartbeats is selected to conform the input sample that will feed each CNN training. This random selection is performed in order to include variability in the calibration of the CNN. Indeed, since input variables have been obtained with a deterministic model, namely CircAdapt, it will not be complicated for the CNN to obtain a combination of weights that give a high accuracy¹. As a result, to design a meaningful transfer learning methodology, uncertainty needs to be incorporated into the data.

In order to challenge the transfer learning methodology, results obtained with the former input sample will be compared with another CNN calibration including a different random sample of 37.000 heartbeats in which white noise is to be added. To this purpose, a white noise signal with $\mu = 0$ and $\sigma = 0.4$ will be added to the pressure input variables, while a white noise signal with $\mu = 0$ and $\sigma = 0.2$ will be added to the volumetric input variables. Fig. 2.1 shows, as a matter of illustration, three representative PV loops without noise and including noise for different sarcomere properties. This figure shows that a higher level of uncertainty is added to phases I and III of the cardiac cycle (Fig. 1.4), while phases II and IV remain less affected.

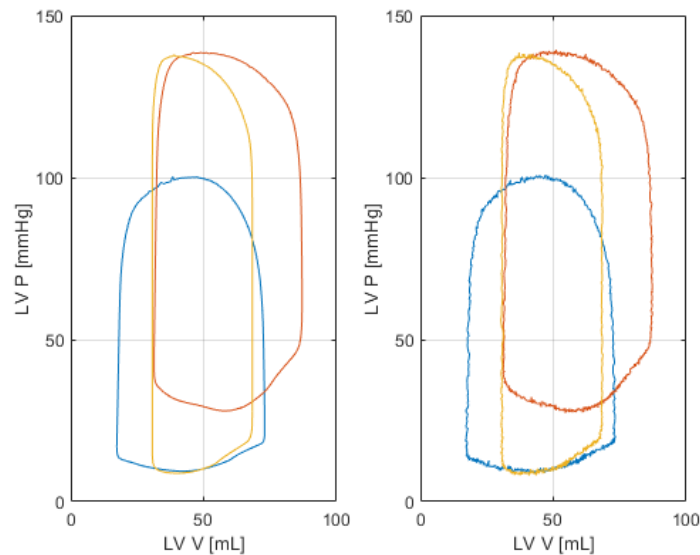


Figure 2.1: Representative PV loops with and without noise used in the training of the CNN.

¹This effect is indeed observed in the results of Chapter 3, for which all models that are calibrated without noise reach an accuracy above 90%.

2.2.2 Spectrogram and normalization

After having defined the input sample that will be used to train the CNN, 1D signals for pressure and volumetric variables need to be transformed into 2D images, for which convolutional neural networks are the most efficient [10]. A traditional method to perform this transformation, which will be used in the present study, is by means of spectrograms. A spectrogram is a visual representation of the frequency spectrum of a signal as a function of time. In this representation, the x and y axes represent time and frequency, respectively, while the color intensity of the point (x, y) depicts the power of frequency y at time x .

All signals that are simulated by the CircAdapt model and subsequently used to train the proposed CNN are sampled at a frequency $\nu = 1.573$ Hz, forming a vector of size $l = 750$ temporal data points, and the spectrogram of each of these signals is created using a Hamming window of size w . Also, a percentage p of overlap between Hamming windows together with an FFT resolution n_{ft} are considered. With this disposition, the spectrogram will be an array of size $n_f \times n_t$ with

$$n_f = \text{int} \left(1 + \frac{n_{ft}}{2} \right) \quad ; \quad n_t = \text{int} \left(\frac{l - n_o}{w - n_o} \right) \quad (2.1)$$

where int is the function that truncates a floating number to its integer part and n_o is the number of overlapped points between Hamming windows, i.e. $n_o = pw$. For the proposed CNN, these features are set to $w = 75$, $p = 0.6$, $n_{ft} = 2^{10}$ and, as a result, the dimensions of the resulting spectrograms are $n_f = 513$ and $n_t = 23$. Fig. 2.2 shows a representative spectrogram corresponding to a LV pressure signal with noise. From this figure, one observes that spectrograms have relatively low frequency contributions that are packed in a regular pattern, as expected.

Since normalization of input data is critical for the training of convolutional neural networks [6, 13], the input heartbeats and target variables are normalized to the interval $[0, 1]$. This normalization, which is performed at the spectrogram level, is achieved by dividing the spectrogram values of each signal by their maximum value. It is relevant to note from this operation that the variable that is being normalized is frequency power, and thus the level of the signal is dismissed in the process. This assumption, which will be referred to as the *normalization hypothesis*, is a relevant assumption of the proposed model. Nonetheless, this hypothesis has a relevant advantage for the proposed network, as it allows a fixed and independent range for the input images, making all spectrograms comparable, and thus ensures a proper mathematical normalization² for the CNN.

Finally, the pressure and volume-related spectrograms for each heartbeat are gathered into a single 2-channel image that represents that heartbeat, and that will be fed to the CNN for training.

²Note that the input images that are used to train the CNN do not consider RGB values, but spectrogram values, which do not have an upper bound as opposed to RGB images.

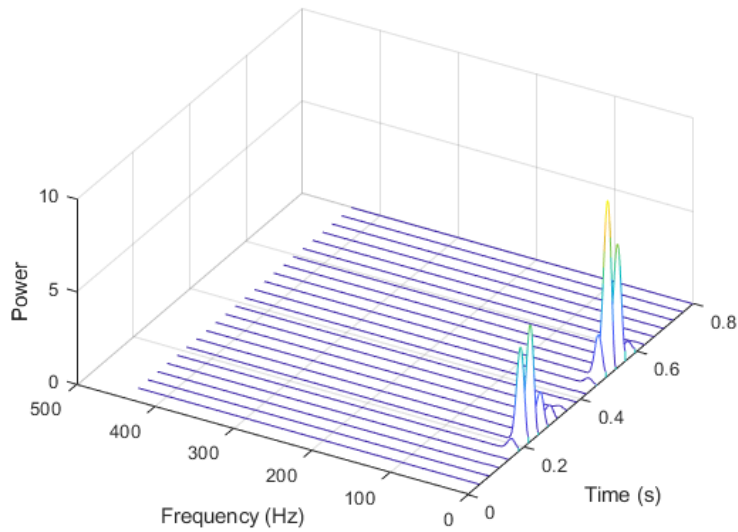


Figure 2.2: Representative spectrogram of a LV pressure signal with noise.

2.3 CNN architecture

As has been said, there are no applications of neural networks to the sarcomer properties problem to the best of the author’s knowledge. As a result, and since CNN architectures are significantly problem-dependant, similar parameter calibration problems with time-dependent variables from other fields have been studied. Literature in this regard has been found in the field of Civil Engineering, for which hysteresis loops’ parameters have been successfully calibrated with CNNs ([17],[20]), as well as in the field of Electrical Engineering, in which properties for power cells have also been calibrated with success via CNNs [26].

In general terms, a basic principle needs to be followed in order to design the CNN architecture. Namely, the CNN has to be as simple as possible. Indeed, including more parameters than necessary or too deep a learning process may lead to better results in terms of convergence, but may lack generalization and thus be susceptible of overfitting. As a result, the choosing of hyperparameters is not an easy task in new problems. Since no literature is at hand, the proposed architecture has been, in general, obtained with a trial and error process, looking into the usual configurations of networks utilized for calibration problems. Some hyperparameters of this structure are susceptible of investigation, while others will remained fixed in order to permit a tractable problem.

The main characteristics of the proposed CNN are discussed in the following sections.

2.3.1 Structure

The proposed CNN has been structured into three 2D convolutional layers with 2D max-pooling, a dropout layer, a flatten layer and three dense fully-connected layers:

1. The first layer of the network is a 2D convolutional layer that takes the input spectrogram, with size $513 \times 23 \times 2$, and performs a 2D convolution with 32 filters of size 3×3 with a 1×1 stride. As a consequence, the size of the output image is $511 \times 21 \times 32$. Additionally, the first layer contains a max-pooling layer with a 2×2 padding, which reduces the size of the output image by two to $255 \times 10 \times 32$.
2. The second layer of the network is another 2D convolutional layer that takes the output of the previous layer and performs a 2D convolution with 64 filters of size 3×3 and a stride of 1×1 . As a result, the size of the output image is $253 \times 8 \times 64$. The second layer also includes a max-pooling layer with a 2×2 padding, which reduces the size of the output image by two to $126 \times 4 \times 64$.
3. The third layer of the network is another 2D convolutional layer that takes the output of the previous layer and performs a 2D convolution with 128 filters of size 3×3 and a stride of 1×1 . As a result, the size of the output image is $124 \times 2 \times 128$. The third layer also includes a max-pooling layer with a 2×2 padding, which reduces the size of the output image by two to $62 \times 1 \times 128$.
4. The fourth layer of the network performs a dropout of 15% of the variables³, hence changing no dimension of the image.
5. The fifth layer of the network flattens the input image, a 3D matrix, to a unidimensional vector of dimension $62 \times 1 \times 128 = 7.936$.
6. The sixth layer of the network is a fully-connected layer with 90 neurons, thus reducing the dimension of the output vector to 90.
7. The seventh layer of the network is another fully-connected layer with 40 neurons, hence outputting a 40-dimensional vector.
8. The eighth and last layer of the network is another fully-connected layer with 3 neurons, one per calibrated parameter, thus outputting the final 3-dimensional vector of calibrated parameters.

Table 2.1 summarizes the structure of the proposed CNN.

2.3.2 Trainable parameters

CNN coefficients that are used to weigh the input variables in a neuron are widely known as CNN parameters. These are chosen with an optimization algorithm in the process that we call training. In the proposed CNN, model parameters depend on each particular layer:

- The first convolutional layer has 9 filter elements (3×3) for a total of 32 filter nodes, and an input image of 2 channels. As a result, the layer will have a total of $2 \times 9 \times 32 + 32 = 608$ parameters, including biases. Note that the max-pooling layer does not add any additional parameters to the layer.

³Note that the dropout rate is thus 15%.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 511, 21, 32)	608
max_pooling2d (MaxPooling2D)	(None, 255, 10, 32)	0
conv2d_1 (Conv2D)	(None, 253, 8, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 126, 4, 64)	0
conv2d_2 (Conv2D)	(None, 124, 2, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 62, 1, 128)	0
dropout (Dropout)	(None, 62, 1, 128)	0
flatten (Flatten)	(None, 7936)	0
dense (Dense)	(None, 90)	714330
dense_1 (Dense)	(None, 40)	3640
dense_2 (Dense)	(None, 3)	123
=====		
Total params: 811,053		
Trainable params: 811,053		
Non-trainable params: 0		

Table 2.1: Proposed CNN architecture.

- Likewise, the second layer, which has 9 filter elements (3×3) for a total of 64 filter nodes, and an input image of 32 channels, will have a total of $32 \times 9 \times 64 + 64 = 18.496$ parameters, including biases. As before, the max-pooling layer does not add any additional parameters to the layer.
- The third layer has 9 filter elements (3×3) for a total of 128 filter nodes, and an input image of 64 channels. As a result, this layer will have a total of $64 \times 9 \times 128 + 128 = 73.856$ parameters, including biases. Once again, the max-pooling layer does not add any additional parameters to the layer.
- Since the dropout and flatten layers do not learn from the data, they only put some values to zero and rearrange the size of the image, no parameters are added to the model for these layers.
- The sixth layer is a fully-connected layer with 90 neurons and an input vector of dimension 7.936. For this reason, there are as many parameters as the product of neurons times the input values plus biases, $90 \times 7.936 + 90 = 714.330$.
- The seventh layer is another fully-connected layer with 40 neurons, and an input vector of dimension 90. For this reason, this layer will have a total of $40 \times 90 + 40 = 3.640$ parameters, including biases.
- Finally, the last layer has 3 neurons with an input vector of dimension 40, and thus has $3 \times 40 + 3 = 123$ parameters including biases.

In summary, the eight layers of the proposed CNN form a model with a total of 811.053 parameters. Since not one of these is restrained, there is a total of 811.053 degrees of freedom that will need to be trained from the input data. Table 2.1 shows a summary of the architecture with the associated parameters.

2.4 Training method

Once the architecture has been set, a training algorithm needs to be selected. Generally speaking, the training method requires three main features: the neural activation, the loss function and the optimization algorithm.

2.4.1 Neural activation

A neural activation function needs to be selected for each layer of the network that requires learning. As a result, for the CNN at hand, a neural activation function is to be selected for each of the three convolutional layers and for each of the three fully-connected layers. Note that the dropout layer and the flatten layer do not actively learn from the data, and thus do not require any neural activation.

- For the three convolutional layers, the rectified linear unit in Fig. 2.3, also known as *ReLU* activation function, is selected. Since these layers perform the convolutional operation of the input image with several filters f , the output will be given by

$$z = ReLU(\text{conv}(x, f)) \quad (2.2)$$

where x and z are the input and output values for each convolutional operation (conv) and f is the filter used at point x . The advantage of the ReLU function is that it allows a nonlinearity component to the output of the convolution while allowing a distinction⁴ in the activation of different neurons.

- For the first two fully-connected layers, the rectified linear unit is also selected according to the same advantages as in the previous case. In these cases, however, each neuron performs a linear operation $y = \mathbf{A}x + \mathbf{B}$, where x and y are the input and output images to the neuron, respectively, and \mathbf{A} and \mathbf{B} are the matrices of neural weights and neural biases. With the inclusion of a ReLU activation function, a threshold is required for the neurons to be activated, in such a way that

$$z = ReLU(\mathbf{A}x + \mathbf{B}) \quad (2.3)$$

- For the last fully-connected layer, which gives the outputs of the model, the neural activation function is set to be a sigmoid function. This selection owes to the fact that disregarding negative values of $\mathbf{A}x + \mathbf{B}$ is not appropriate in the output layer since it represents a loss of information, and hence a ReLU function has to be

⁴Note that the neuron does not activate for negative values of the input variable if a ReLU function is used. This allows unrelated neurons to completely deactivate in a particular training step.

discarded. In this regard, since the problem at hand is a calibration problem and not a classification problem, a sigmoid function complies with the requirement⁵, distributing all values of the real line. As a result, the output values for the last layer are

$$z = \text{Sigmoid}(\mathbf{A}x + \mathbf{B}) \quad (2.4)$$

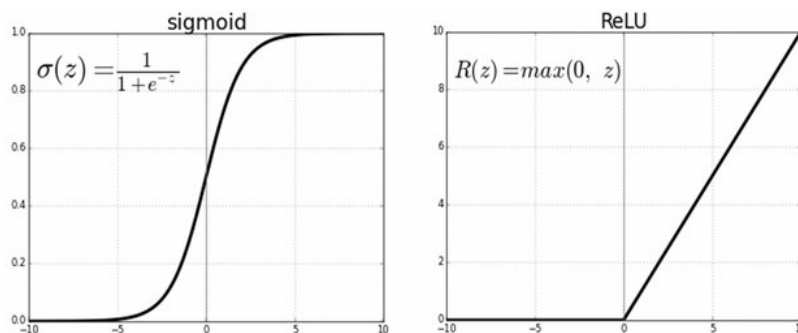


Figure 2.3: ReLU and sigmoid activation functions [16].

2.4.2 Loss function and optimization algorithm

Another important feature that needs to be selected to be able to train a CNN is the loss function, which gives the error of the prediction made by the neural network at each training step. Since the problem at hand is a calibration problem, the loss function \mathcal{L} is set to the Mean Square Error (MSE), which has the form

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (z_{pred} - z_{real})^2 \quad (2.5)$$

where N is the number of data points in the sample, z_{pred} is the predicted value for the output z and z_{real} is the true value for the output z . This loss function has two main advantages for the calibration problem at hand, namely:

1. It does not penalize overpredicting against underpredicting, or viceversa, since the error is symmetric, i.e. $(z_{pred} - z_{real})^2 = (z_{real} - z_{pred})^2$.
2. It heavily penalizes outlier predictions, since their corresponding MSE makes the loss function increase rapidly. It is worth noticing that this property is sometimes seen as a downside, especially for the cases in which outlier predictions are not very relevant to the output. Since the problem at hand deals with cardiac properties, no outliers are to be allowed.

⁵It is interesting to note that the sigmoid function does not output mutually exclusive values for the three neurons. This is appropriate in a calibration problem, but would not have been admissible in a classification problem.

After defining the loss function, an optimization algorithm needs to be selected. The objective of this optimization algorithm is to look for the optimal parameters that minimize the total loss function \mathcal{L} . Following recent developments in convolutional neural networks, the *Adam* algorithm has been selected [15]. Adam, derived from adaptive moment estimation, is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. For this reason, it requires very little memory while being robust and well-suited to a wide range of non-convex optimization problems in the field machine learning.

2.5 Hyperparameters

Those CNN parameters that need to be defined beforehand by the user are known as *hyperparameters*. Consequently, these will not be trained by the learning algorithm itself and another, more manual, selecting procedure will need to be put in place. There are two different types of hyperparameters: those depending on the nature of the layers and those related to the learning algorithm. For the CNN at hand, the hyperparameters that emerge from the layer architecture are:

- Technically speaking, the number of layers and their nature (convolution, fully-connected, dropout) are considered as hyperparameters of the network. Nevertheless, since no literature is available for the problem at hand, the model architecture has been obtained by a trial and error process, and will be considered as fixed for all developments of this thesis. In this regard, the optimization of this architecture remains as an interesting open question⁶.
- Convolutional layers have the following main hyperparameters: number of filter nodes n_f , filter size⁷ $f \times f$ and strides $s_1 \times s_2$. Furthermore, the max-pooling layers have the size of the pooling, $p_1 \times p_2$, as additional hyperparameters. The selection of these has been based on the research done in the field of Civil Engineering [17, 20], together with an additional consideration: s_1 , s_2 , p_1 and p_2 need to be small, since the input image size does not have a high-frequency resolution according to Fig. 2.2. The proposed architecture hyperparameters are described in Section 2.3.1 and summarized in Table 2.2.

Layer	n_f	$f \times f$	$s_1 \times s_2$	$p_1 \times p_2$
Convolution 1	32	3×3	1×1	2×2
Convolution 2	64	3×3	1×1	2×2
Convolution 3	128	3×3	1×1	2×2

Table 2.2: Proposed hyperparameters for each convolutional layer.

⁶See Section 4.2 for a detailed description of aspects that remain open for the problem at hand.

⁷Since rectangular filters are rarely used in practice, only square filters of size f are considered in the discussion.

- The dropout layer requires a dropout rate d_r , which relates to the number of input variables that are set to zero in the next layer input image. Since this hyperparameter's objective is to avoid overfitting [24], it tends to be lower in calibration problems than in classification problems. For the problem at hand, d_r has been set to 0.15 according to the relevant recommendations related to small datasets and calibration problems [3, 24].
- Fully-connected layers have essentially one hyperparameter, namely the number of neurons n_n . Its selection has been made in a trial an error basis, with the reference of the two Civil Engineering papers [17, 20]. In this regard, more neurons have been necessary to fully capture sarcomer characteristics than those used in the former references. It is worth noticing that the last layer requires only three neurons, since three parameters are required as the output of the network. Table 2.3 summarizes the selected number of neurons for each layer.

Layer	n_n
Fully-connected 1	90
Fully-connected 2	40
Fully-connected 3	3

Table 2.3: Proposed number of neurons for each fully-connected layer.

The Adam optimization algorithm allows for multiple hyperparameters to be selected. Nevertheless, almost all of them have been set to the keras default parameters, with two exceptions:

- The learning rate has been set to 0.001.
- The train-test split of the sample is set to 25% for validation and 75% for training.

Both of these hyperparameters have been set according to common practice and the reference literature [17, 20, 26].

Chapter 3

Results and discussion

3.1 Learning results

The objective of the present thesis is to design a convolutional neural network (CNN) that is capable of calibrating stiffness, relaxation and elastance of the cardiac sarcomere by means of two variables, one that is related to the pressure of the chamber and one that is related to its volume. The available data for these two variables, acquired with CircAdapt, is:

1. Regarding the pressure variable, it is possible to train the CNN either with LV pressure or with arterial pressure. It is worth noticing that the second one would be preferable to the first one since it can be non-invasively obtained.
2. Regarding the volumetric variable, the available data to train the CNN is LV volume, septal strain and LV strain. These variables can be measured clinically with a simple Doppler echocardiography.

In order to train the best possible network, five groups of pressure-volume variables are assessed, for a given heartbeat:

- LV pressure and LV volume
- LV pressure and septal strain
- LV pressure and LV strain
- Arterial pressure and septal strain
- Arterial pressure and LV strain

It is worth noticing from the previous pairs that the combination *arterial pressure and LV volume* is not considered in the analysis since arterial pressure does not seem to be a good predictor of the sarcomere properties and strain is a shifted version of LV volume.

According to the above-mentioned combinations of variables, two different samples of 37.000 heartbeats are used to train two CNNs for each pair, one in which white noise is introduced according to Section 2.2 and one with direct output results from CircAdapt. For validation purposes, a split of 25% is considered in all calibrated CNNs. The following sections show the training results for each of the assessed models.

3.1.1 LV pressure vs LV volume

The first model uses the two variables that are the most difficult to obtain clinically: LV pressure and LV volume. These, in fact, are the variables that, from an anatomical point of view, are thought to best correlate to sarcomere stiffness, relaxation and elastance. Learning results for the training of this CNN are shown in Fig. 3.1, both considering and not considering noise.

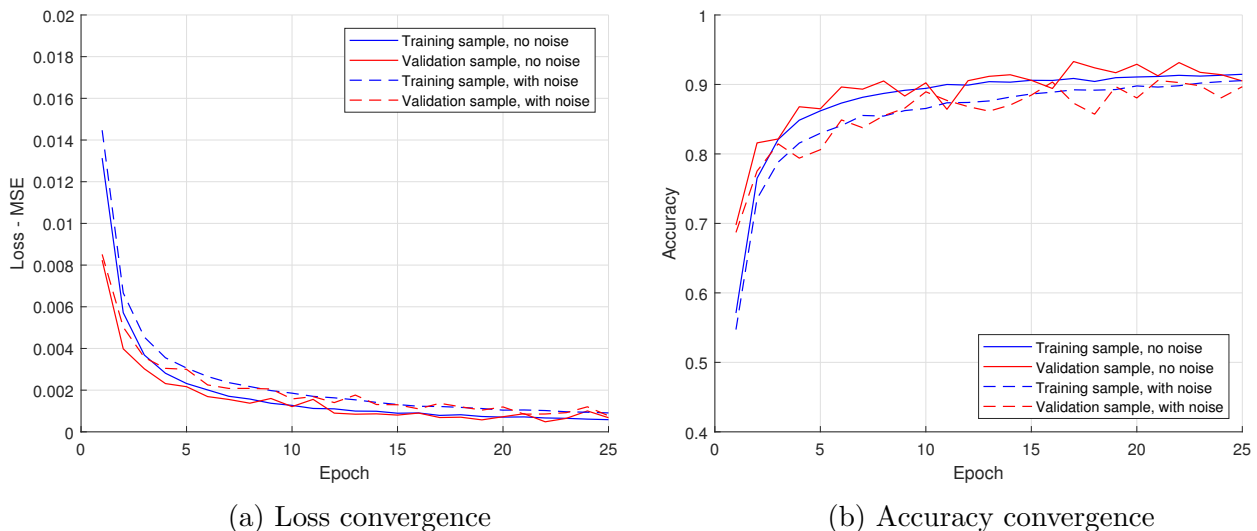


Figure 3.1: Learning convergence for LV pressure - LV volume.

From the former results, one first observes that the model without noise (solid lines) obtains better predictions for the sarcomere properties than the model with noise (dashed lines), as expected. Moreover, another property that emerges from the results in Fig. 3.1 is the fact that the validation sample closely accompanies the training sample both in loss and in accuracy, owing to the fact that both samples obtain the parameters from the CircAdapt software. Also, relevantly enough, it is observed that the difference between the model with noise and the model without is very limited, with a final difference in accuracy that is less than 1% according to the results shown in Table 3.1.

Given that real signals will have some degree of noise, it is interesting to investigate the error that is committed in the parameters for the case with noise. Defining the error ε as

$$\varepsilon = z_{real} - z_{pred} \quad (3.1)$$

with z being the stiffness proportion k/k_{max} , the relaxation proportion η/η_{max} or the elastance proportion ϵ/ϵ_{max} , of the validation sample, Fig. 3.2 shows the discrete probability¹ of ϵ being in a particular bin of width 0.01.

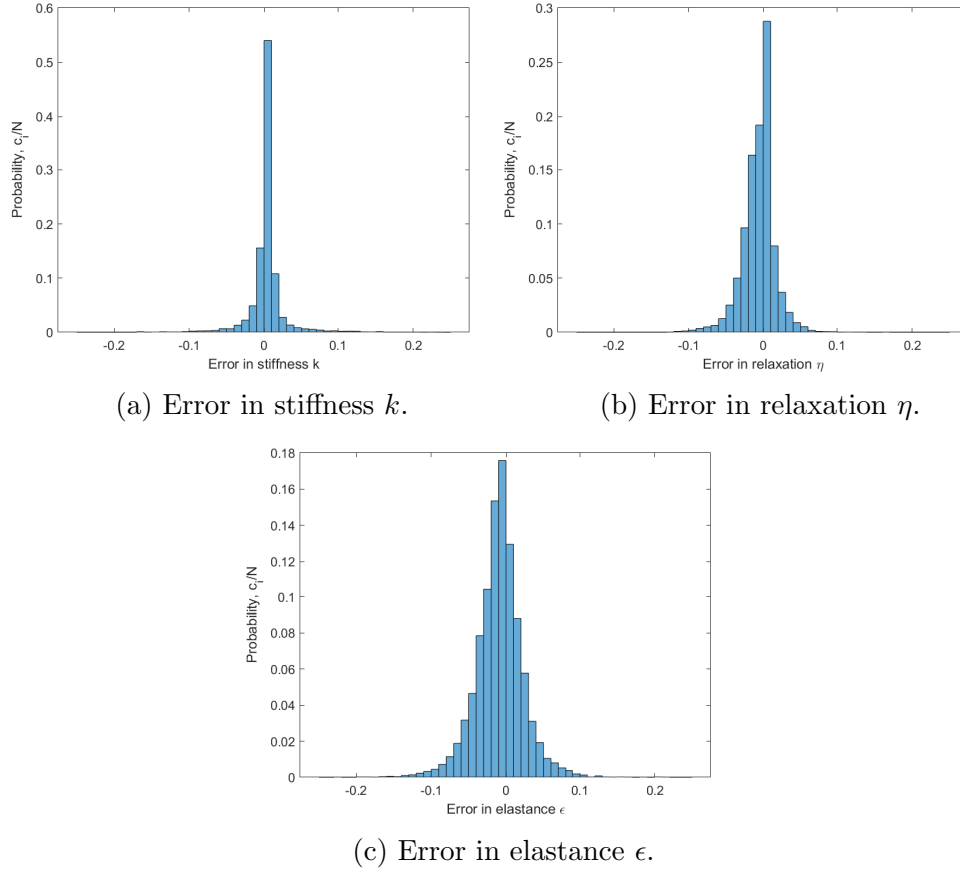


Figure 3.2: Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - LV volume calibration.

From the results in Fig. 3.2, it is first observed that the stiffness is the property that is best calibrated from the data, showing an error² of $\pm 1\%$ almost 70% of the times. In fact, one can observe that the error is lower than $\pm 5\%$ almost 99% of the times, which is an extremely good result for clinical purposes. Relaxation is the next best calibrated parameter from the data, with an error of $\pm 1\%$ almost 50% of the times, followed by elastance, with an error of $\pm 1\%$ almost 31% of the times. From these results, it is also interesting to note that there is no significant bias towards positive or negative errors.

¹Note that the total area under the histogram is equal to 1.

²Note that all reported errors are in terms of the maximum parameter k_{max} , η_{max} and ϵ_{max} .

3.1.2 LV pressure vs septal strain

The second model uses LV pressure and septal strain to calibrate the properties of the sarcomere. Learning results for the training of this CNN are shown in Fig. 3.3, both considering and not considering noise.

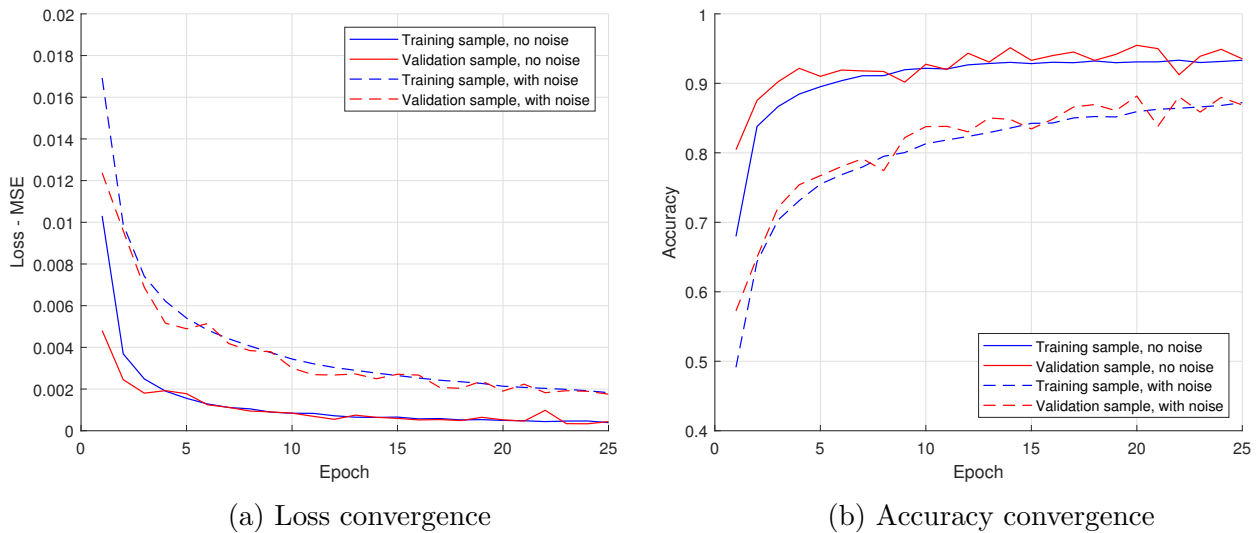


Figure 3.3: Learning convergence for LV pressure - septal strain.

From the previous results, it is observed that the calibration without noise (solid lines) features a better learning convergence than the model considering noise (dashed lines), as expected. Nonetheless, the average difference between the results obtained with and without noise for this case is significantly larger than it was for the LV pressure - LV volume case, as seen in comparison to Fig. 3.1. This may be anatomically explained by the fact that the septal strain is more indirectly correlated to the sarcomere properties than the LV volume is. Additionally, this case also shows that the validation results are very similar to the training results, owing to the relationship that CircAdapt promotes between the two samples.

Defining an error ε in terms of the maximum properties as in the previous section (Eq. 3.1) allows us to investigate the error in which the predictions might incur. In this regard, Fig. 3.4 shows the discrete probability of ε being in a particular bin of width 0.01 for the results including noise.

Results in Fig. 3.4 show an analogous tendency in the error estimates to the one observed in the previous section. Indeed, the stiffness is yet again the property that is best calibrated by the network, showing an error of $\pm 1\%$ almost 65% of the times. Relaxation and elastance follow stiffness with similar errors, with the distribution for the relaxation being slightly wider, at an error of $\pm 1\%$ nearly 32% of the times. Additionally, results show that there is no apparent bias in the model towards positive or negative errors.

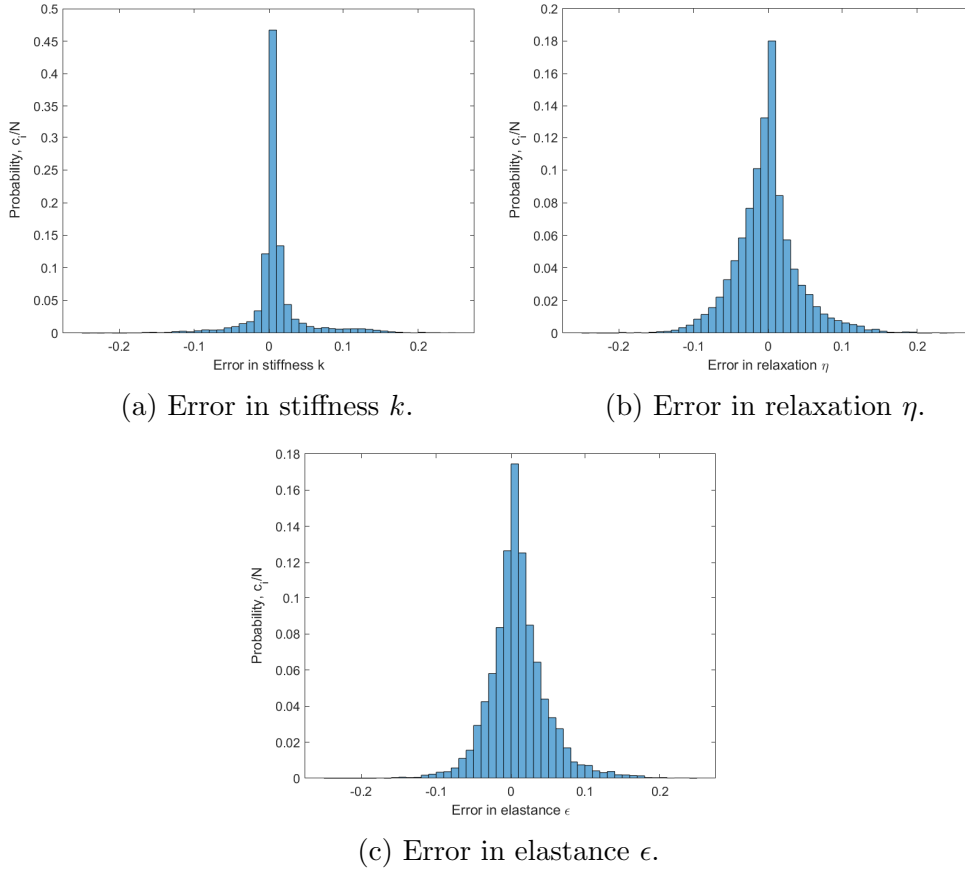


Figure 3.4: Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - septal strain calibration.

3.1.3 LV pressure vs LV strain

The third model uses LV pressure and LV strain to calibrate the properties of the sarcomere. Learning results for the training of this CNN are shown in Fig. 3.5, both considering and not considering noise.

Results from Fig. 3.5 show, once again, that the learning convergence is best for the model without noise (solid lines), as expected. In this regard, the observed average difference between both calibrations, including and not including noise, is aligned with the LV pressure - septal strain calibration, and thus being significantly larger than the one for LV pressure - LV volume. Results in Fig. 3.5 also show that the validation sample obtains similar convergence levels to the training sample in both calibrations, owing to the relationship that CircAdapt promotes between the two.

With an analogous definition for the error ϵ as in previous sections (Eq. 3.1), the relative error referred to the maximum properties is investigated. Fig. 3.6 shows the discrete probability of the error ϵ being in a particular bin of width 0.01 for the results including noise. These results show an analogous tendency in the error estimates as observed in the previous sections. Indeed, the stiffness becomes again the property that is

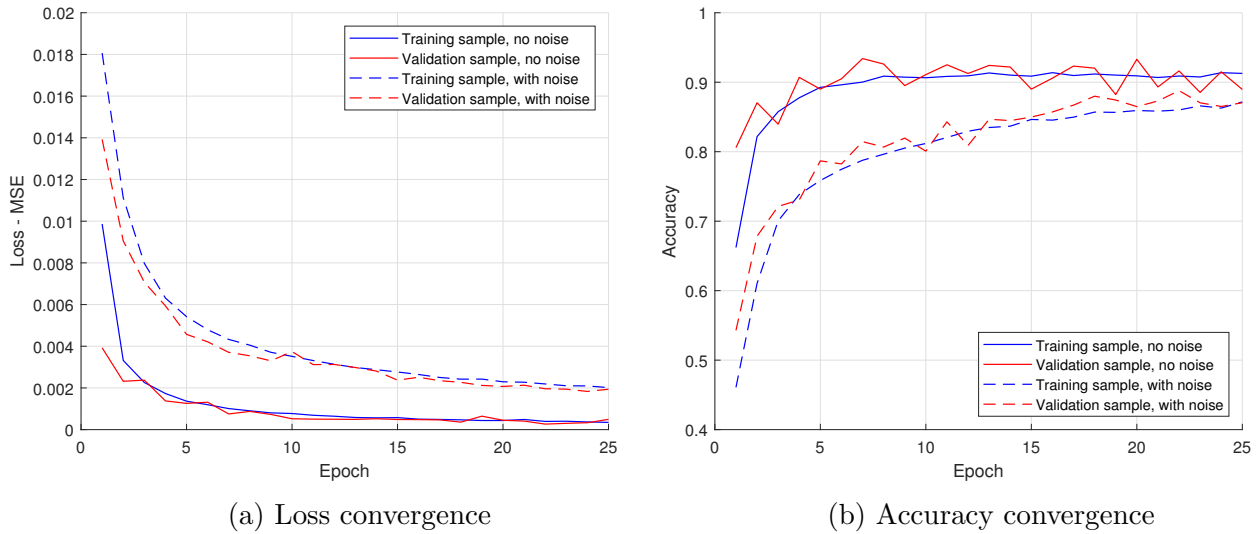


Figure 3.5: Learning convergence for LV pressure - LV strain.

best calibrated by the CNN, showing an error of $\pm 1\%$ almost 52% of the times. Relaxation and elastance follow stiffness with similar errors, namely $\pm 1\%$ for nearly 26% and 25% of the instances, respectively. Once again, no apparent bias is observed towards positive or negative errors.

3.1.4 Arterial pressure vs septal strain

The fourth model uses arterial pressure and septal strain to calibrate the properties of the sarcomere. Learning results for the training of this CNN are shown in Fig. 3.7, both considering and not considering noise.

Results derived from Fig. 3.7 show that the learning convergence is best for the model without noise (solid lines) for this CNN as well. Furthermore, the observed average difference between both calibrations, including and disregarding noise, becomes larger than in any of the previous studies, and thus showing that the combination of arterial pressure - septal strain is much less effective than the former ones. Results from Fig. 3.7 also show that the validation sample and the training sample have similar losses, while a more chaotic behavior is obtained in accuracy for the validation sample. It is also relevant to note, in this case, that the validation sample obtains similar convergence levels to the training sample in both calibrations, owing to the relationship that CircAdapt promotes between the two.

Using an analogous definition of the error ε as in previous sections (Eq. 3.1), one can investigate the relative error referred to the maximum properties. Fig. 3.8 shows the discrete probability of the error ε being in a particular bin of width 0.01 for the results considering noise. These results show the same tendency observed in previous section, for which the stiffness is the best calibrated sarcomere property. In this case, however, the error level is much larger than in previous studies, being $\pm 1\%$ nearly 17% of cases.

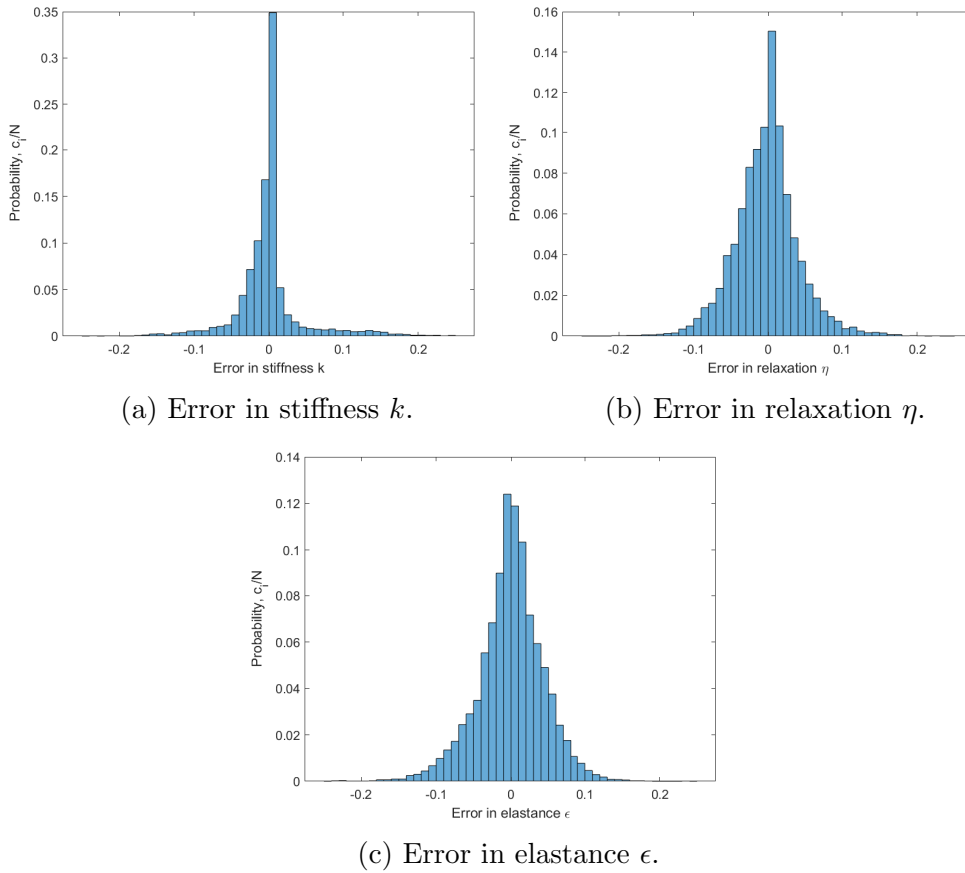


Figure 3.6: Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for LV pressure - LV strain calibration.

Elastance follows stiffness in this calibration, showing an error of $\pm 1\%$ in a bit more than 14% of the instances. The worse-calibrated sarcomere property becomes the relaxation in this case, with an error of $\pm 1\%$ in about 10% of the cases. Additionally, in this case, one observes that the distribution width and skewness is slightly larger for relaxation and stiffness than it is for elastance.

3.1.5 Arterial pressure vs LV strain

The last model uses arterial pressure and LV strain to calibrate the properties of the sarcomere. Learning results for the training of this CNN are shown in Fig. 3.9, both considering and not considering noise.

Results depicted in Fig. 3.9 show a similar tendency in loss and accuracy to the ones observed in the previous section. Indeed, a better convergence is observed for the model without noise than for the one considering noise, and a significant difference is observed in the loss and convergence levels between the two calibrations. This consideration emphasizes yet again that the arterial pressure is not a very precise input variable to calibrate sarcomere properties. Fig. 3.9 also shows signs of a slight overfitting for the calibration

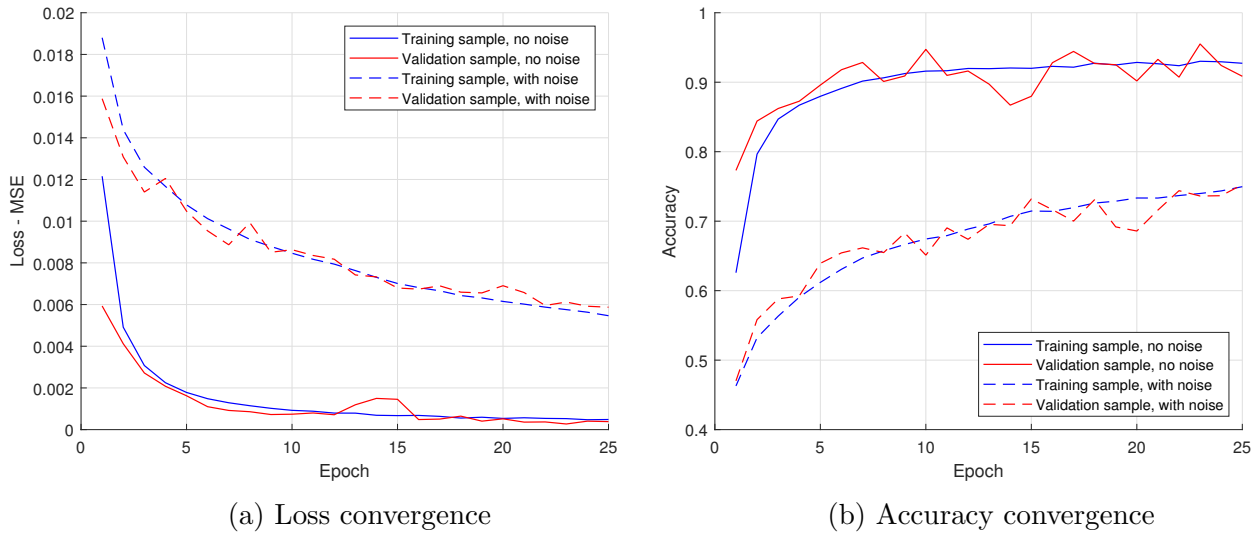


Figure 3.7: Learning convergence for arterial pressure - septal strain.

with noise, since the validation loss and accuracy reach the asymptote while the training loss and accuracy still show a tendency to improve. This fact provokes a stagnation of validation performance compared to training performance, which is oftentimes related to overfitting. However, results for this calibration, as shown in Table 3.1, show that these differences are still not too critical to invalidate the model.

With an analogous definition for the error ε as in previous sections (Eq. 3.1), the relative error referred to the maximum properties is investigated. Fig. 3.10 shows the discrete probability of the error ε being in a particular bin of width 0.01 for the results including noise. These results show the same trend that has been observed in all previous sections, with the stiffness being the best calibrated sarcomere property. In this case, the probability of the stiffness error to be $\pm 1\%$ is about 14%, while the probabilities for the same error are nearly 10% and 11% for relaxation and elastance, respectively. These error levels show that this combination of input variables is the worst to be considered for the calibration of sarcomere properties. Additionally, in this case one observes a slightly wider probability distribution for the relaxation and stiffness than for the elastance.

3.1.6 Summary of learning results

Aiming at the discussion of the results for the previously-calibrated 10 CNNs, Table 3.1 summarizes loss and accuracy for the converged solutions, both for training and for validation samples. Additionally, in order to gauge the clinical correctness of the proposed calibrations, Table 3.2 summarizes validation errors for 70% and 95% probabilities for each sarcomere property.

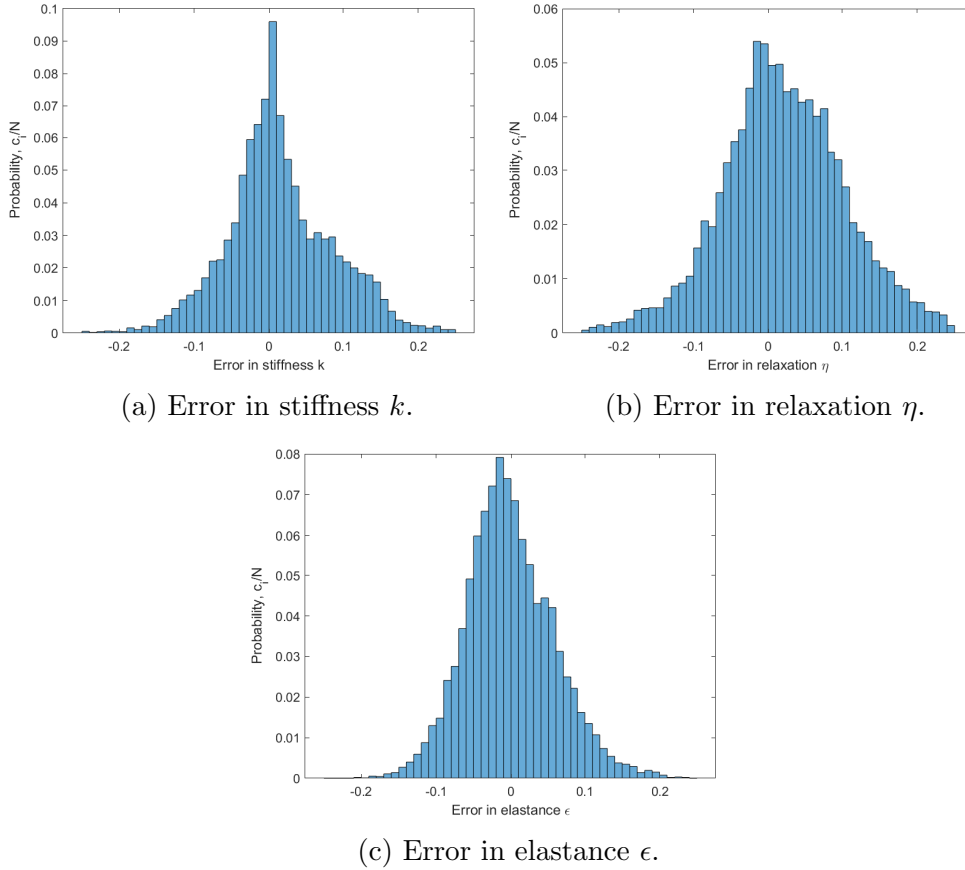


Figure 3.8: Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for arterial pressure - septal strain calibration.

Variables			Training sample		Validation sample	
<i>Press.-related</i>	<i>Vol.-related</i>	<i>Noise</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Accuracy</i>
LV press.	LV vol.	No	$5.842 \cdot 10^{-4}$	91.46%	$6.781 \cdot 10^{-4}$	90.50%
		Yes	$9.077 \cdot 10^{-4}$	90.55%	$7.699 \cdot 10^{-4}$	89.68%
LV press.	Sept. strain	No	$3.889 \cdot 10^{-4}$	93.32%	$4.360 \cdot 10^{-4}$	93.52%
		Yes	$1.8 \cdot 10^{-3}$	87.22%	$1.8 \cdot 10^{-3}$	86.89%
LV press.	LV strain	No	$3.475 \cdot 10^{-4}$	91.27%	$4.927 \cdot 10^{-4}$	88.96%
		Yes	$2 \cdot 10^{-3}$	87.16%	$1.9 \cdot 10^{-3}$	87.05%
Art. press.	Sept. strain	No	$4.852 \cdot 10^{-4}$	92.73%	$3.839 \cdot 10^{-4}$	90.85%
		Yes	$5.5 \cdot 10^{-3}$	74.97%	$5.9 \cdot 10^{-3}$	75.26%
Art. press.	LV strain	No	$4.035 \cdot 10^{-4}$	91.95%	$2.530 \cdot 10^{-4}$	95.56%
		Yes	$5.2 \cdot 10^{-3}$	75.09%	$6.8 \cdot 10^{-3}$	72.56%

Table 3.1: Summary of learning results for the assessed CNNs.

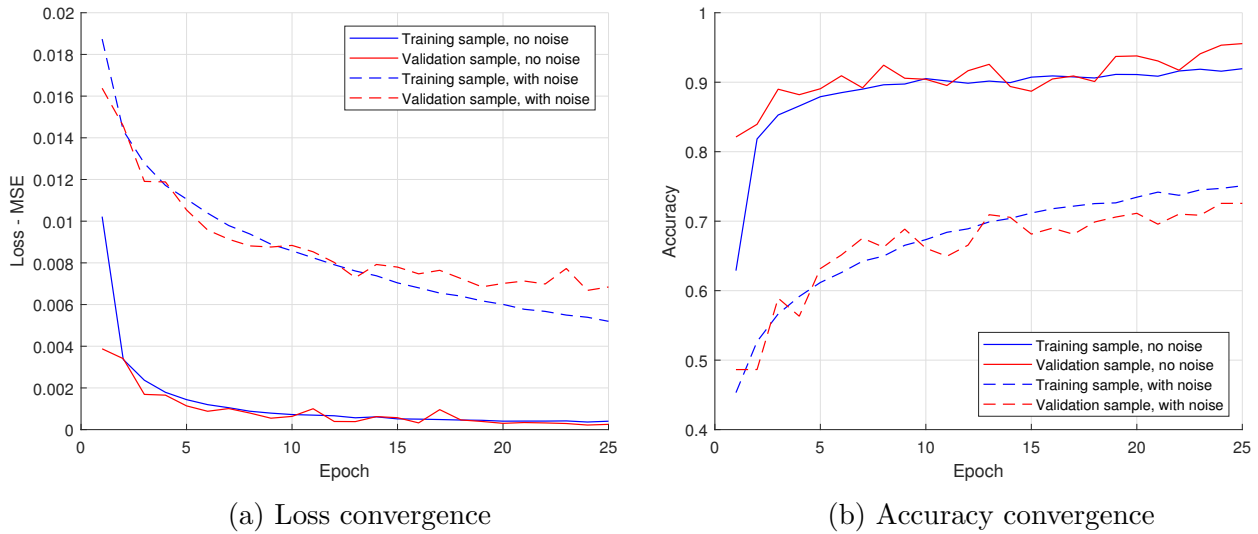


Figure 3.9: Learning convergence for arterial pressure - LV strain.

Variables			Errors		
<i>Press.-related</i>	<i>Vol.-related</i>	<i>Noise</i>	<i>Stiffness k</i>	<i>Relaxation η</i>	<i>Elastance ϵ</i>
LV press.	LV vol.	No	$\pm 1\%$ ($\pm 3\%$)	$\pm 2\%$ ($\pm 4\%$)	$\pm 2\%$ ($\pm 4\%$)
		Yes	$\pm 2\%$ ($\pm 3\%$)	$\pm 2\%$ ($\pm 4\%$)	$\pm 2\%$ ($\pm 4\%$)
LV press.	Sept. strain	No	$\pm 2\%$ ($\pm 2\%$)	$\pm 2\%$ ($\pm 3\%$)	$\pm 2\%$ ($\pm 3\%$)
		Yes	$\pm 2\%$ ($\pm 7\%$)	$\pm 4\%$ ($\pm 8\%$)	$\pm 4\%$ ($\pm 8\%$)
LV press.	LV strain	No	$\pm 1\%$ ($\pm 3\%$)	$\pm 2\%$ ($\pm 3\%$)	$\pm 2\%$ ($\pm 3\%$)
		Yes	$\pm 3\%$ ($\pm 8\%$)	$\pm 4\%$ ($\pm 8\%$)	$\pm 4\%$ ($\pm 8\%$)
Art. press.	Sept. strain	No	$\pm 2\%$ ($\pm 2\%$)	$\pm 2\%$ ($\pm 4\%$)	$\pm 2\%$ ($\pm 4\%$)
		Yes	$\pm 8\%$ ($\pm 13\%$)	$\pm 9\%$ ($\pm 16\%$)	$\pm 9\%$ ($\pm 16\%$)
Art. press.	LV strain	No	$\pm 1\%$ ($\pm 2\%$)	$\pm 1\%$ ($\pm 3\%$)	$\pm 1\%$ ($\pm 3\%$)
		Yes	$\pm 8\%$ ($\pm 13\%$)	$\pm 10\%$ ($\pm 17\%$)	$\pm 10\%$ ($\pm 17\%$)

Table 3.2: Summary of errors (70% and 95% probabilities) obtained in the parameters for the assessed CNNs.

3.2 Discussion of the results

With the aim of discovering the best possible convolutional neural network (CNN) to calibrate sarcomere properties, ten different networks have been trained with different pairs of pressure and volumetric variables. The training process has fed each CNN with a different set of 37.000 hearbeats, leaving a 25% of the sample for validation purposes.

As was expected, all five CNNs that do not consider white noise achieve a better performance than their counterparts considering noise. In fact, in all five cases that disregard noise, calibrated CNNs perform at an astonishing level according to Tables 3.1

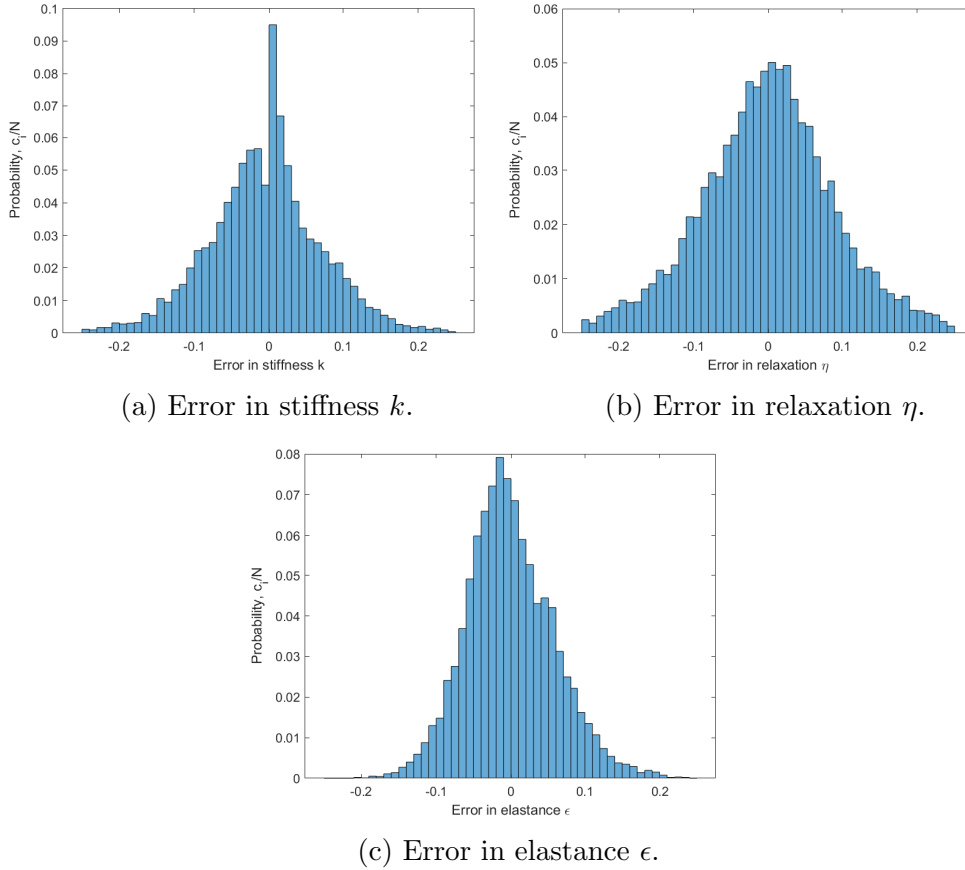


Figure 3.10: Discrete probability of the error in k/k_{max} , η/η_{max} and ϵ/ϵ_{max} for arterial pressure - LV strain calibration.

and 3.2, with accuracies above 90% and relative errors to maximum properties of less than 4% with a 95% probability. This high performance is well-understood, since the corresponding input and target variables are obtained with CircAdapt. This software, which models the functioning of the heart at a multiscale level, embed deterministic relationships in the processed data that are discovered by the CNNs.

With the objective of permitting a transfer learning of the CNNs, some independent white noise, with $\mu = 0$ and $\sigma = 0.2$ for volumetric variables and $\mu = 0$ and $\sigma = 0.4$ for pressure variables, is incorporated in each signal. This noise emmascarates the deterministic relationships within the data, and allows a more realistic evaluation of the CNN performance. When considering noise, accuracy remains high for the LV pressure - LV volume calibration, with levels around 90%, and an error in parameters of less than 4% with a 95% probability. With the substitution of LV volume by strain (either septal or LV), accuracy drops slightly to around 87%, keeping the error under 8% with a 95% probability. These error levels may in fact still be admissible in the clinical practice. Finally, the inclusion of arterial pressure instead of LV pressure makes the accuracy of the CNN plummet to 72 – 76%. In this case, the committed error in the estimation becomes

around 8–10% with a 70% probability and around 13–17% with a 95% probability. These more inaccurate solutions, however, may lead to a less risky evaluation by the doctor, since no invasive clinical procedure is required to obtain the input data. Nevertheless, as it is not possible to measure the real values of the sarcomere properties inside the *working* chamber, large trials should be conducted to assess patient’s normality and see how these properties change in well-characterized disease scenarios.

Two additional considerations can be made regarding the error distributions for the different sarcomere properties. To begin with, it is worth noticing that, for all assessed CNNs, the stiffness is the best-calibrated property of all, with a relatively thin probability distribution as summarized in Table 3.2. In contrast, the error distributions for relaxation and elastance have a wider shape, which correlate to larger errors. Finally, no significant bias is observed towards positive or negative errors, with a slight exception in relaxation and stiffness for the case with arterial pressure and septal strain.

Chapter 4

Conclusions and future research

4.1 Conclusions

Irrespectively of the root cause, diastolic dysfunction is present in virtually all structural myocardial diseases. For this reason, it is of high interest to clinicians to understand the changes that occur in the mechanical properties of the heart, since these active and volume-mediated mechanical properties are the major determinants of diastolic function, as well as systolic function. In this regard, direct models exist nowadays that can relate sarcomere properties to chamber properties and cardiac cycle. However, since these sarcomere properties are virtually impossible to measure *in vivo* or *in silico*, no model exists to solve the corresponding inverse problem.

In order to contribute to this open problem, the present thesis assesses several convolutional neural networks (CNNs) that target sarcomere mechanical properties (stiffness, relaxation and contractility) by means of pairs of synthetic hemodynamical variables such as left ventricle (LV) pressure or volume, left ventricle (LV) or septal strain, or arterial pressure, and proposes the most relevant fit in terms of accuracy and clinical simplicity. Since no real data is available for sarcomere properties, the calibration sample is obtained via the CircAdapt model in a transfer learning approach.

The assessed CNNs are trained with 37.000 spectrograms corresponding to the temporal Fourier decomposition of the time-dependent hemodynamical variables, and the proposed architecture combines three 2D convolutional layers with max-pooling, followed by a dropout layer and three fully-connected layers. Relatively small filters (3×3) are necessary in the convolutional layers to obtain accurate pattern recognition, while two internal fully-connected layers with 90 and 40 neurons suffice for the task.

For every pair of input variables, the training of the neural network, which considers a 25% split for the validation sample, is performed with the raw results from CircAdapt and with results in which white noise has been incorporated. As expected, all five assessed CNNs (LV pressure/LV volume, LV pressure/septal strain, LV pressure/LV strain, arterial pressure/septal strain, arterial pressure/LV strain) that do not consider white noise perform at an astonishing level of accuracy above 90%, with relative errors in the

predicted sarcomere properties of less than 4% with a 95% probability. This high performance is well-understood, since the corresponding input and target variables are linked by the embedded relationships of the multiscale model CircAdapt.

The masking of the CircAdapt relations with white noise allows a more realistic evaluation of the CNN performance. When considering noise, accuracy remains high for the LV pressure - LV volume calibration, with levels around 90% and a relative error in parameters of less than 4% with a 95% probability. With the substitution of LV volume by strain (either septal or LV), accuracy drops slightly to around 87% while controlling the error under 8% with a 95% probability. These error levels may in fact still be admissible in the clinical practice. Finally, the inclusion of arterial pressure instead of LV pressure makes the accuracy of the model plummet to around 74%. In this case, the committed relative error in the estimation becomes around 15% with a 95% probability. Despite these more inaccurate solutions, it is worth noticing that the cost for the patient is virtually zero in these cases, since arterial pressure and strain can be obtained easily with noninvasive tests.

Furthermore, all assessed CNNs demonstrate that the sarcomere stiffness is the best-calibrated property of all, with a relatively thin probability distribution for the relative errors. In contrast, relative error distributions for sarcomere relaxation and elastance have a wider shape, which correlate with higher errors.

4.2 Future lines of research

The work that has been developed in the present thesis has left two main aspects to which more research efforts could be destined.

First, the developments of this thesis have assumed that the sarcomere properties are independent of the level of pressure and volume, since the spectrograms used to train the CNNs have been normalized by their maximum. Even though the results that have been obtained prove relevant accuracies, this hypothesis, which we call the normalization hypothesis, shall be object of further analysis.

Also, given that CNNs trained with arterial pressure show the lowest level of accuracy and the highest relative error, it would be interesting to assess whether this is caused by an improvable architecture that could recognize patterns in a more efficient way, or it is caused by the normalization hypothesis. It might be the case that arterial pressure, which corresponds to a smoother pressure signal, requires the keeping of the level, and thus a relationship of these low levels of accuracy and the former hypothesis might not be surprising.

Bibliography

- [1] Arts, T. et al. “Adaptation to mechanical load determines shape and properties of heart and circulation: the CircAdapt model”. *American Journal of Physiology - Heart and Circulatory Physiology* 288 (2005), pp. 1943–1954.
- [2] Arts, T. et al. “Modeling the relation between cardiac pump function and myofiber mechanics”. *Journal of Biomechanics* 36.5 (2003), pp. 731–736.
- [3] Baldi, P. and Sadowski, P. “Understanding Dropout”. *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. 2013.
- [4] Bermejo, J. et al. “Diastolic chamber properties of the left ventricle assessed by global fitting of pressure-volume data: improving the gold standard of diastolic function”. *Journal of Applied Physiology* 115 (2013), pp. 556–568.
- [5] Buoso, S., Joyce, T., and Kozerke, S. “Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks”. *Medical Image Analysis* 71 (2021).
- [6] Chollet, F. *Deep learning with python*. Manning Publications Co., 2018.
- [7] Columbia University. *Mechanical Properties of the Heart 1 & 2*. Lecture Notes. URL: <https://ccnmtl.columbia.edu/projects/heart/exercises/MechPropHeart/lecture.html>.
- [8] Dahal, S. *Sarcomere: definition, structure, diagram and functions*. The Biology Notes. 2023. URL: <https://thebiologynotes.com/sarcomere/>.
- [9] Esteva, A. et al. “A guide to deep learning in healthcare”. *Nature Medicine* 25 (2019), pp. 24–29.
- [10] Fawaz, H.I. et al. “Deep learning for time series classification: a review”. *Data Mining and Knowledge Discovery* 33 (2019), pp. 917–963.
- [11] Fozzard, H. et al. *The Heart and Cardiovascular System: Scientific Foundations*. Second Edition. Raven, 1991.
- [12] Gonzalez, R. and Woods, R. *Digital Image Processing*. Third Edition. Pearson, 2018.
- [13] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. The MIT Press, 2016.
- [14] Hall, J. E. *Guyton and Hall Textbook of Medical Physiology*. Thirteenth Edition. Elsevier, 2015.

- [15] Kingma, D.P. and Lei Ba, J. *Adam: a method for stochastic optimization*. ArXiv. 2017. URL: <https://arxiv.org/abs/1412.6980>.
- [16] Kizrak, A. *Comparison of activation functions for deep neural networks*. Towards Data Science. 2019. URL: <https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a>.
- [17] Lanning, A., Zaghi, A.E., and Zhang, T. “Applicability of Convolutional Neural Networks for Calibration of Nonlinear Dynamic Models of Structures”. *Frontiers in Built Environment* 8 (2022).
- [18] Lumens, J., Delhaas, T., and Arts, T. “Three-wall segment (TriSeg) model describing mechanics and hemodynamics of ventricular interaction”. *Annals of Biomedical Engineering* 37.11 (2009), pp. 2234–2255.
- [19] Mirsky, I. “Assessment of diastolic function: suggested methods and future considerations”. *Circulation* 69 (1984), pp. 836–841.
- [20] Naderpour, H. et al. “Shear Strength Prediction of Reinforced Concrete Shear Wall using ANN, GMDH-NN and GEP”. *Journal of Soft Computing in Civil Engineering* 6 (2022), pp. 66–87.
- [21] Owan, T.E. et al. “Trends in prevalence and outcome of heart failure with preserved ejection fraction”. *New England Journal of Medicine* 355 (2006), pp. 251–259.
- [22] Physiopedia contributors. *Anatomy of the Human Heart*. Physiopedia. 2023.
- [23] Saha, S. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. SaturnCloud. 2018. URL: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>.
- [24] Srivastava, N. et al. “Dropout: A simple way to prevent neural networks from overfitting”. *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.
- [25] Stewart, M. *Simple introduction to convolutional neural networks*. Towards Data Science. 2019. URL: <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>.
- [26] Wshah, S. et al. “Deep Learning for Model Parameter Calibration in Power Systems”. *2020 IEEE International Conference on Power Systems Technology (POWERCON)*. 2020.

**DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO,
PARA LA DEFENSA DEL TRABAJO FIN DE MASTER**

Fecha: 22/6/2023

Quien suscribe:

Autor(a): **Miquel Crusells Girona**
D.N.I./N.I.E./Pasaporte.: **47762408X**

Hace constar que es el autor(a) del trabajo:

Título completo del trabajo.

Characterization of left ventricle sarcomere properties using deep learning

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el documento que se entrega.

DECLARACIÓN:

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.



Fdo. Miquel Crusells Girona