PERVASIVE CARDIOVASCULAR AND RESPIRATORY MONITORING DEVICES

Chapter 5 Modeling and simulation of biomedical systems

ABSTRACT

In this chapter, we describe the principles of modeling biomedical systems. The objective is to put together components and techniques studied in Chapters 2-4 in the context of modeling a biomedical device. We describe models that allow us to generate data, add noise and interference, include tolerance of the electronic components, assess signal quality, and so on. In addition, we consider systems that are modeled using electrical circuits and show how one can propagate uncertainties through the models. We show how one can develop an end-to-end model with different sub-models and where processing is done in software or analog hardware. In the end, we briefly discuss the power consumption in the system.

Keywords: modeling, uncertainty propagation, parameter estimation, noise, motion artifacts, interference, signal quality assessment, signal quality index, power estimation, stop breathing detection, Windkessel model, charge amplifier, real-time systems.

CONTENT

5.1. Introduction

In this chapter, we will discuss modeling of a physiological system and circuits and provide examples of different models, performance analysis, parameter optimization, uncertainty and sensitivity analysis, etc. This chapter is needed to prepare for the next part of the book, where we will model particular devices, such as a blood pressure monitoring device. We have already seen in Chapters 3 and 4 several models of transducers and circuits; however, these transducers and circuits were not presented as parts of the biomedical devices, and additional analyses that include uncertainty propagation and sensitivity analysis were not performed.

This chapter starts with a description of the data collection process and some common physiological signal databases. Even if we include a model for signal generation, the parameters of the model need to be evaluated with experimental data. Data can be obtained in the following ways:

1. Performing data collection, which is necessary when developing a new device to fit the parameters of the device and to evaluate the performance of the device
2. Using already collected data from databases such as Physionet
3. Using in-silico data which is generated using computational models
4. Generating data from developed models of the physiological system as well as models of interaction between the device and the subject.

The first three items will be covered in Section 5.2, while the models (item 4) will be described using an example in Section 5.3. Item 4 is important for us because it will allow us to evaluate the effect physiological and other parameters have on the accuracy of the device. In Section 5.3, we will also perform parameter estimation using provided experimental data.

A general model of a biomedical system is shown in Fig. 5.1. The model includes not only the device itself but also the model of an interaction of the physiological system and the sensor, as well as the noise and artifacts model. This approach will allow us to understand the effect of all simulated parameters of the system on the output, including some physiological parameters. However, physiological models are normally very complex, and we will not focus on modeling the physiological system itself in detail because this is outside the scope of this book.

Noise and artifact models are covered in Section 5.4.2. Analysis of signal quality is provided in Section 5.4.3. In Fig. 5.1, the signal quality index (SQI) is computed at the end of the pipeline. However, we can compute the SQI after each block as well as at the input to the device. Error is normally evaluated at the end, as shown in Fig. 5.1.

A circuit-based model is presented in Section 5.5 to analyze uncertainty propagation.

After converting the signal from analog to digital, the signal is further processed in software (Section 5.6). We will briefly discuss modeling real-time signal processing in software followed by offline processing. Finally, as an example, we will consider the design of a system that generates an alarm in case of breathing cessation. We will simulate a signal obtained from a breathing belt. A breathing belt is a device that a subject wears around her or his chest used to extract the breathing signal based on the displacement of the chest during inspiration and expiration. This signal can be processed to provide breathing rates over time, analyze breathing patterns and indicate whether the breathing rate is too high or too low. We will model and simulate a simplified system that can extract the breathing signal and detect that the subject stopped breathing for a longer period than a predefined threshold. This design will be based on two design paradigms. The first design paradigm is completely hardware-based, while the second design paradigm is based on the approach where minimum required processing is done in hardware to obtain the digital signal, and further processing is performed in software.

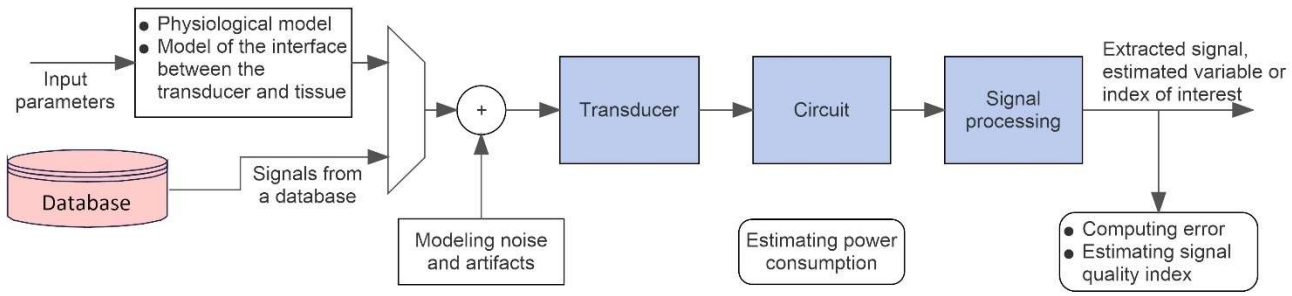


Fig. 5.1 General model of a biomedical system. The block before the plus sign represents a multiplexer that will allow us to use either signals from the database or signals generated from our models.

Several example models are explained and simulated in this chapter. First, we introduce the Windkessel model to model the aortic pulse and show the process of estimating model parameters. The signal quality index was assessed on ECG data. Uncertainty is propagated through the circuit that converts the signal from the piezoelectric transducer into voltage to measure acceleration. The power consumption of the A/D converter is evaluated in the end. We have already mentioned a simulated system for the detection of breathing cessation.

Acronyms and explanations

REB	Research Ethics Board
PWV	pulse wave velocity
MSE	mean square error
LSB	least significant bit
DFT	Discrete Fourier Transform
STFT	Short-time Fourier Transform
SDR	spectral distribution ratio
SQI	signal quality index
$kSQI$	Kurtosis-based SQI
pSQI	spectral power ratio based SQI
$tSQI$	template matching based SQI

Variables used in this chapter include:

- tol tolerance of a resistor
- $S(f)$ spectral density function
- β noise color parameter
- $x(t)$ signal of interest
- B amplitude of a narrowband signal

- $v(t)$ white Gaussian noise
- T_K the temperature in degrees Kelvin
- $Kurt$ kurtosis
- μ the mean
- σ the standard deviation
- $y_j, y(t)$ the output of a function, model or a device
- y_{REF} the reference measurements
- L the number of samples in a block or a window needed to produce one output sample
- T_s sampling period
- T_{sy} sampling period of the signal at the output (if it is different from the sampling period of the signal at the input)
- T_{proc} time to process L samples
- E energy
- P power
- f_{clk} clock frequency
- V_{dd}, V or V^+ supply voltage
- T_s sampling frequency

5.2. Data collection and databases

When developing any biomedical device, we need data to set the parameters of the device. Data is normally obtained by performing pilot studies in which a new device is tested on human subjects against a reference device or a trained medical practitioner. In order to perform the pilot study, Research Ethics Board (REB) approval must be obtained first. Next, the study participants need to agree with the proposed protocol by signing an informed consent form. This section starts with a description of the ethics application process.

We often start data collection in the environment where the device is developed. However, colleagues in companies and students/researchers at universities and research centers might not be the appropriate population for testing the device. Nevertheless, some physiological changes can be caused by simple maneuvers, which might be good enough for the initial test of some devices. Hence, a subset of common maneuvers is also presented in this section.

We can use the data from existing databases or simulated data before data collection. A description of several online databases and simulators is presented at the end of this section.

5.2.1. Ethics application

Before any data collection, the first step is obtaining the REB approval. The ethics board focuses on three major aspects of the study: foreseeable risks, potential benefits and ethical implications of the project. Risks are related to protecting the subjects' privacy by not exposing their identity and ensuring they are safe. In Canada, the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans – TCPS 2 (2018) is commonly followed when submitting and reviewing Ethics applications [TCPS2_18].

The ethics application normally includes the main form, recruitment letter and informed consent form. In the main form, the researchers address the team of experts in the REB regarding the way the study will be conducted and the privacy will be preserved. In the consent form, the researchers address the subjects and explain, at a very high level, all the aspects of the study so that the subjects can understand the study. The experiments cannot begin before obtaining approval from the REB and the subject's signature on the informed consent form.

The REB is looking for the following when reviewing the application:

- description of the project,
- description of the recruitment process,
- description of what the subject needs to do during the study (participation),
- risk for the subject,
- benefits to the subject and the community,
- privacy, confidentiality and data conservation, and
- consent process and consent document.

5.1. Who does evaluate REB applications?

After submitting the application, the application gets evaluated first by the Ethics officer. The application can be considered a minimum risk; in this case, it is commonly evaluated by two members of the REB as well as the Ethics officer. If the Ethics officer assumes that the application has some issues, mainly regarding privacy or risks to participants, then the application goes to the full board review, where it is discussed and evaluated by all members of the REB during a face-to-face meeting. Being a REB member, I saw many engineering REB applications written in a hurry. These applications take most of the time for the REB to discuss. REB members are volunteers, and it is important to think of their time too. The informed consent form also needs to be carefully written too. The researcher and the subject sign it, and it represents the researcher's promise to the subject that the study will be conducted in a way it is claimed in the form which was approved by the REB.

The description of the project provides a rationale for the project and the background literature. The recruitment process deals with several important aspects, including the participants' inclusion and exclusion criteria. In some cases, it is unnecessary or too dangerous to include people with some conditions. The inclusion criterion could be the age range (for example, subjects should be at least 18 years old). An example of an exclusion criterion is excluding cardiac patients from the studies that require significant physical effort. The recruitment process also needs to describe if there is a risk of coercion and how it is handled, and how potential participants will be notified about the incoming study. Next, the subjects' participation during the study needs to be detailed, including the type of activities, duration, location, physical effort required, and so on. In addition, it will also include the description of a device or a method that is tested as well as the description of all reference devices used. Potential harmful effects and risks to the participants need to be discussed in detail. Next, how the private information of participants will be protected needs to be explained. In some cases, it will not be possible to guarantee the protection of the identity of participants – then limits of privacy and anticipated risks to participants need to be clearly explained. The last part is related to the consent process and consent document. The consent process needs to be clearly explained. The consent document should contain information needed for participants to understand the study, its goals and risks.

REB's mandate is not to evaluate the scientific aspects of the project; however, a project description with references needs to be included so that the REB has the complete information required to evaluate the application. It is also very important for the researchers themselves to write the full application and put down on paper the study plan that is then evaluated by themselves as well as by the third party (REB), thereby improving the study.

5.2.2. Maneuvers

In this section, we will discuss some additional aspects of data collection. The first step in testing pervasive wearable devices is testing on healthy subjects. However, these subjects often do not show the same level of variations of physiological variables of interest as subjects with conditions would show. For example, the variations of some variables of interest can be smaller (blood pressure values) in a healthy group without maneuvers or larger (heart rate variability) than in the group with medical conditions. This is why several different maneuvers have been introduced to emulate variations of some of the physiological parameters of interest. By performing the maneuvers, subjects can rapidly change, for example, blood pressure which can help evaluate a newly developed blood pressure device. Maneuvers can also be performed on patients to evaluate the effects of some actions.

5.2.2.1. Valsalva maneuver

The Valsalva maneuver is an attempt at expiration made with the closed nose and the mouth. It can result in a very rapid increase in blood pressure and heart rate. Therefore, it can be used in pilot studies where it is required to increase the blood pressure artificially.

5.2.2.2. Muller maneuver

After a forced expiration, an attempt at inspiration is made with a closed mouth and nose, representing the reverse of a Valsalva maneuver. It is normally performed when testing sleep apnea since it emulates the obstruction of breathing.

In order to understand the effect of change as well as the range of physiological parameters that are affected by the maneuvers, we show through simulations in Fig. 5.2 the effects of the Valsalva and Muller (sometimes written as Mueller) on breathing by observing the tidal volume, oxygen saturation, arterial blood pressure, and heart rate. A number of other parameters are affected by these two maneuvers but are not shown here. The simulated data is obtained by running the software package PNEUMA 3.0 [Cheng10]. Valsalva maneuver starts about a time instant of 40 s with inspiration and breath-hold shown in Fig. 5.2a) as a constant large volume of air in the lungs. Valsalva maneuver has four phases where the first two phases are related to the period of forced expiration, and the other two phases occur after the occlusion is released. In the first phase, there is a sharp increase in arterial blood pressure and some increase in heart rate. In phase 2, the arterial blood pressure decreases while the heart rate increases significantly. In phase 3, the breathing is resumed, and arterial blood pressure drops significantly while the heart rate is still high. Finally, in phase 4, there is an increase in blood pressure before returning to normal and a decrease in the heart rate.

We can also observe the changes in the measured parameters due to the Muller maneuver during the period between 130 s and 150 s. During that time, the subject exhales, and therefore subject's tidal volume is low and relatively constant. There is a significant decrease in oxygen saturation (SaO_2) during the maneuver and immediately after. Please note that we will analyze and introduce these signals in Part II of the book. Here, we just want to show how significant the effect of different maneuvers can be.

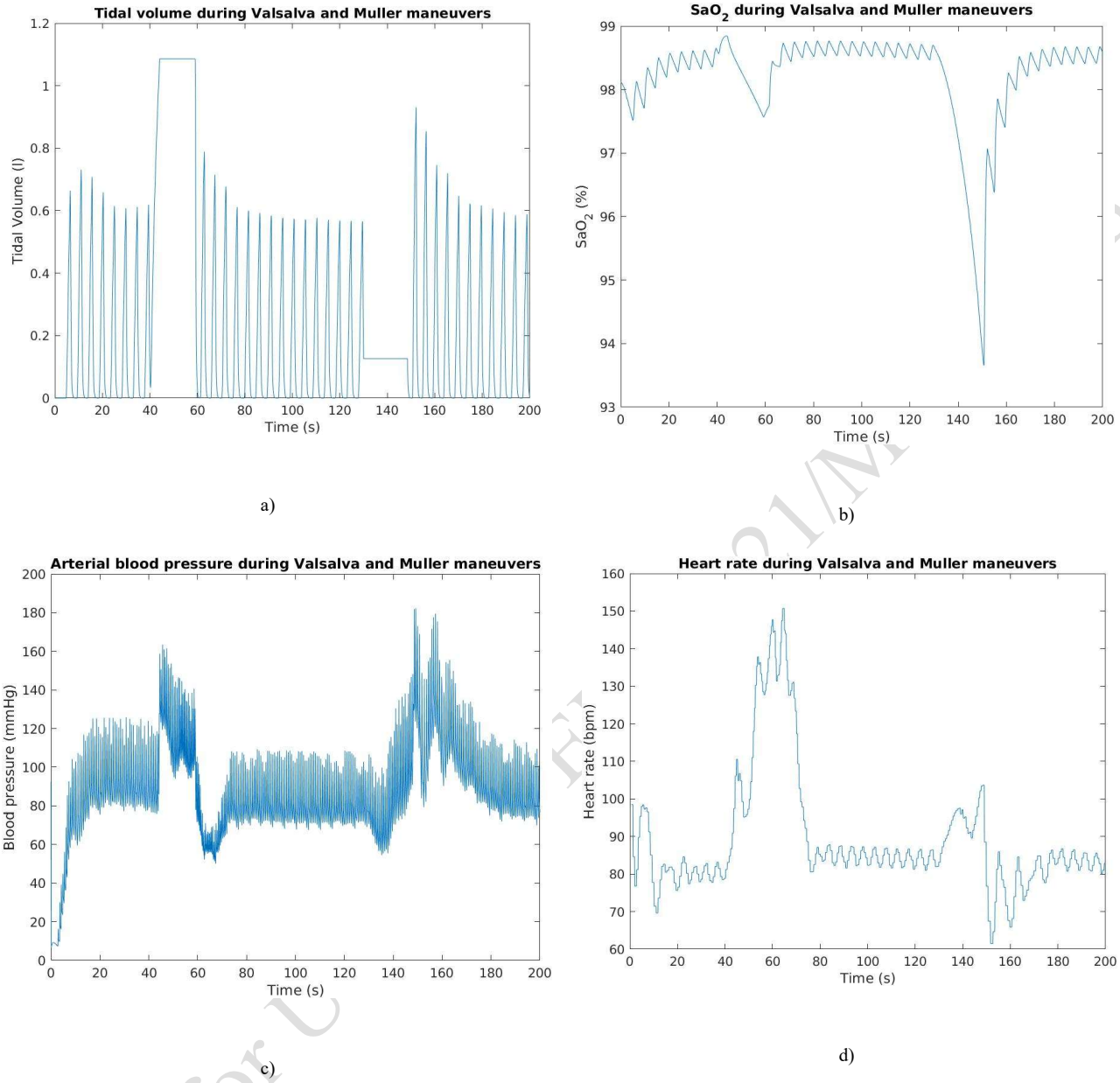


Fig. 5.2 a) Tidal volume, b) Oxygen saturation (SaO2), c) Arterial blood pressure and d) Heart rate during 200 s interval. Valsalva maneuver is performed between 40 s and 60 s, while the Muller maneuver is performed between 130 s and 150 s. The figures are obtained using the simulation packet PNEUMA 3.0 [Cheng10].

5.2.2.3. Deep breathing maneuvers

A deep breathing test requires the subject to breathe in and out at least 6 to 10 times. Respiratory sinus arrhythmia is usually measured using the deep breathing test [Löllgen09]. Respiratory sinus arrhythmia represents variability in the heart rate in synchrony with respiration by shortening the R-R interval in the ECG during inspiration and prolonging it during expiration.

5.2.2.4. Ewing's battery of tests

Ewing's battery of tests was intended to test cardiovascular autonomic function in diabetes patients. Ewing's battery of tests is divided into two groups where the first group evaluates the sympathetic nervous system functions using, for example, the handgrip test. In contrast, the second group investigates the operation of the parasympathetic nervous system by analyzing, for example, heart rate response to deep inspiration and expiration and the Valsalva maneuver [Ewing85], [Spallone11]. Some of these tests are used to induce changes in physiological parameters when developing blood pressure and other biomedical devices.

5.2.3. Databases

PhysioNet is a database of recorded physiologic signals (PhysioBank) freely available online. PhysioNet is supported by the National Institute of General Medical Sciences (NIGMS) and the National Institute of Biomedical Imaging and Bioengineering (NIBIB). One of the PhysioNet datasets is the MIMIC dataset. MIMIC III data set is described in [Aew16] and contains thousands of recordings of multiple physiologic signals collected by patient monitoring devices in adult and neonatal intensive care units.

A large dataset of signals was collected in Great Britain from 4,378 subjects. Among many other parameters, it contains finger PPG pulse waveforms alongside brachial-femoral pulse wave velocities [Elliott14]. Although this database is not available online, data can be accessed after obtaining approval from the data access committee and the Ethics committee.

Another large dataset on atherosclerosis, an important sub-type of arterial hardening and narrowing caused by plaque build-up, is available to researchers through the Multi-Ethnic Study of Atherosclerosis (MESA) [Burke16]. This dataset includes several biological, behavioral, and environmental markers obtained from over 6,800 men and women living in different communities in the U.S.

In the next chapters of the book, we will mention important databases for each device we describe.

5.2.4. Databases based on computational models

Computational (in silico) modeling is the direct use of computer simulation and modeling in biomedical research to simulate physiological processes. Computational modeling of cardiovascular and respiratory functions is important as it allows for:

- evaluating the influence of individual cardiovascular properties and their variations over time on the signal of interest (for example, the arterial pulse wave),
- evaluating the signal of interest at different locations on the body,
- analyzing the effect of aging and different cardiovascular diseases on the signal of interest,
- providing exact measurement references which might be difficult to obtain from in vivo studies,
- developing methods for estimating the uncertainty of measurements,
- generating the signal of interest of virtual patients and,
- performing patient-specific modeling and potentially developing a digital twin.

Major disadvantages of this approach include reliance on modeling hypothesis and setting different parameters of the models where some parameters might be unknown or based on previous studies that might not be completely relevant. Nevertheless, computational modeling can provide additional understanding of the physiological processes of interest.

Modeling and simulation of arterial blood pressure, blood flow velocity, volume flow rate and PPG pulse wave at common measurement sites was described recently in [Charlton19]. The research resulted in a database of pulses of more than 4,000 virtual patients. MATLAB (The MathWorks Inc., Natick, MA) code is provided, allowing for generating new virtual patients with varying parameters.

A database of simulated arterial waves of 3,325 virtual subjects, each with distinctive arterial pulse waveforms, is described in [Willemet15]. The following parameters are varied between subjects: arterial pulse wave velocity (PWV), the diameter of arteries, heart rate, stroke volume and peripheral vascular resistance. It is applied for foot-to-foot PWV simulation and estimating arterial stiffness.

An example of a physiological simulator is PNEUMA which has already been mentioned in the previous section [Cheng10]. PNEUMA is a Matlab-based software that simulates the autoregulation of the cardiovascular and respiratory systems under normal conditions, during sleep, and under different interventions. It allows for obtaining many simultaneous respiratory and cardiovascular signals, including those presented in Fig. 5.2. It can simulate cardiorespiratory responses to different phases of sleep, the effects of mechanical ventilators, Valsalva and Mueller maneuvers, Cheyne-Stokes respiration during sleep, and so on. The simulator is very interesting for this book since it allows for generating a number of signals that can be used for testing different devices/systems with known preset physiological parameters. However, the simulator is designed for physiologists and not for engineers and is not easy to expand to simulate signals from different places on the body.

5.3. Models for signal generation

In the first part of this section, we discuss model types based on their interpretability and how parameter estimation is performed. Then, we show, as an example, the Windkessel model that described the hemodynamics of the arterial system in terms of arterial resistance and compliance. This model is based upon an electrical circuit analogy. Finally, the model parameters are estimated, and the sensitivity analysis is performed.

5.3.1. Types of models

In this book, we will often work with simulated physiological signals. We will review several modeling approaches that are helpful for understanding models used later in the book. They include:

- Physiology-based models are normally presented through a set of differential equations
- Physiology-based models based upon the electrical circuit analogy
- Empirical data-based models

Physiological models are white or gray-box models, and as such, their parameters are often interpretable. They can model the way the signal is generated. The parameters used in the model should have physical or physiological meaning.

Models normally attempt to simulate different patient conditions; therefore, these conditions are inputs to the models. For example, the type of arrhythmia could be modeled by an ECG simulator.

As pointed out in Chapter 6 of [Cerutti11], there are two main ways to assign values to the parameters of an interpretable model, including:

W1. Based on the previous knowledge (forward modeling) – we assign typical parameters to an entire class of subjects with a given pathology. We can use this model to generate signals that are then used as input to a simulated device. In this way, we can better understand a system's behavior for specific pathology or specific ranges of parameters of interest.

W2. Based on parameter estimation or model fitting (inverse modeling) that minimizes the error between the model output and the experimental data. The selection of the cost function to minimize the error and the selection of proper parameters for estimation are normally a challenge. It is pointed out in [Cerutti11] that the unknowns regarding the choice of the parameters and the cost function are one of the bottlenecks in the widespread use of interpretative models in clinical practice.

In the next section, we will show these two ways of assigning the parameters to the model on an example of modeling the arterial pulse based on the Windkessel model. Table 5.1 gives an overview of the models used in the next chapters.

Table 5.1 Different models used in the next chapters.

Signal or variables of interest	Physiological/data-driven model	Device-subject interaction
Arterial pulse	The arterial pressure-volume curve for different arterial stiffness/blood pressure Windkessel model	Effect of the cuff on the arterial volume
Photoplethysmography	Not used in the book. A potential data-driven model can be found in [Tang20]	Light propagation through the tissue
ECG	A potential mathematical model based on differential equations can be found in Chapter 4 of [Clifford06]	Modeling tissue-electrode contact and the effect of sweat
Pulse transit time for continuous blood pressure monitoring	Moens-Korteweg equation	-
Breathing signal	Sine wave signal Breathing pattern simulator	-

5.3.2. Modeling and parameter estimation example: Windkessel model

Three types of computational models are commonly applied as low-dimensional physics-based models of arteries: the Windkessel model, one-dimensional (1-D) models, and tube-load models [Zhou19]. The Windkessel model presents the pulse as a function of time only and is a function of equivalent inertance, compliance and resistance. 1-D and tube-load models represent distributed properties of the arterial system. The 1-D model is based on the Navier–Stokes equation and is used to simulate pressure and flow at any position in the entire arterial tree. Tube-load models are transmission line models, which are made up of multiple parallel tubes with loads representing arteries, and as such, they can model wave propagation and reflection with only a few parameters [Zhou19]. Multiple models connected commonly in series can represent multiple arteries. In this book, we will limit the analysis to the simplest Windkessel model.

5.3.2.1. Example: Windkessel model

The Windkessel model is one of the most used models in studying a systemic arterial system. Windkessel effect explains the phenomenon of the conversion of the pulsatile blood flow into the continuous blood flow in blood vessels.

The systemic arterial system is represented using a three or four-element circuit. In the circuit, the pressure $p(t)$ plays the role of the voltage at different points and the blood flow $Q(t)$ of the current. Pressure is given in mmHg. Flow represents volumetric flux and is defined in units of volume per unit time or ml/s. The four-element circuit representing the Windkessel model is shown in Fig. 5.3. The circuit parameters are:

- Z_0 is the characteristic impedance of the artery $[mmHg \cdot s/ml]$
- R is the peripheral resistance $[mmHg \cdot s/ml]$
- C_a is the arterial compliance [ml/$mmHg$] which is the change in arterial blood volume (ΔV) due to a given change in arterial blood pressure (Δp), i.e., $C_a = \Delta V/\Delta p$ [Spencer63]. Compliance is equivalent to the capacitance in electrical circuits in the following way: Capacitance accumulates the charge and then discharges it to the same circuit. Similarly, compliance represents the ability of arteries to expand during the systole accumulating a volume of blood. During the diastole, this volume of blood is discharged along the arteries.
- L is the arterial inertance $[mmHg \cdot s^2/ml]$ and it is defined as the change in pressure in the arterial segment versus the acceleration of the blood [Spencer87].

The controlled current source represents flow through the aorta $Q_a(t)$. Aortic pressure is $p_a(t)$ while $p_p(t)$ is the pressure over the peripheral resistance and arterial compliance. With the Windkessel model, we are interested in modeling $p_a(t)$.

The Windkessel model is an interpretative model that allows us to achieve the following:

1. Given the aortic flow and the parameters of the model, we can generate aortic and peripheral pressure signals. This corresponds to item W1 in the previous section and will be elaborated on in Section 5.3.2.2.
2. Given the aortic flow and the aortic pressure, we can estimate the values of parameters such as arterial compliance that have physiological meaning and can be interpreted. This corresponds to item W2 in the previous section and will be elaborated on in Section 5.3.2.3.

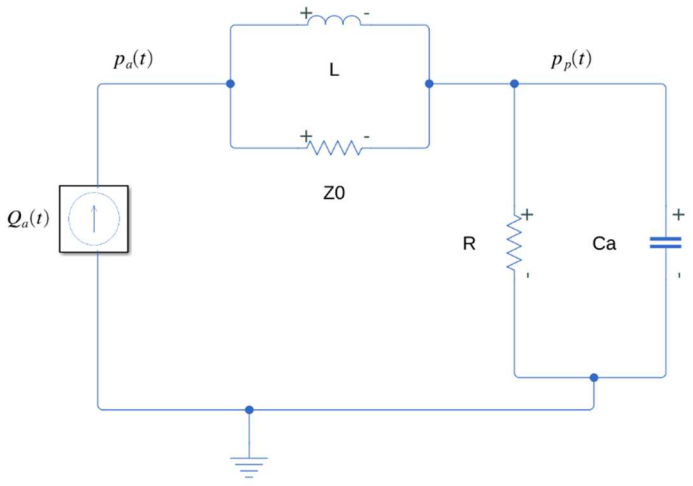


Fig. 5.3 Four-element Windkessel model

5.3.2.2. Windkessel model with parameters assigned based on previous knowledge

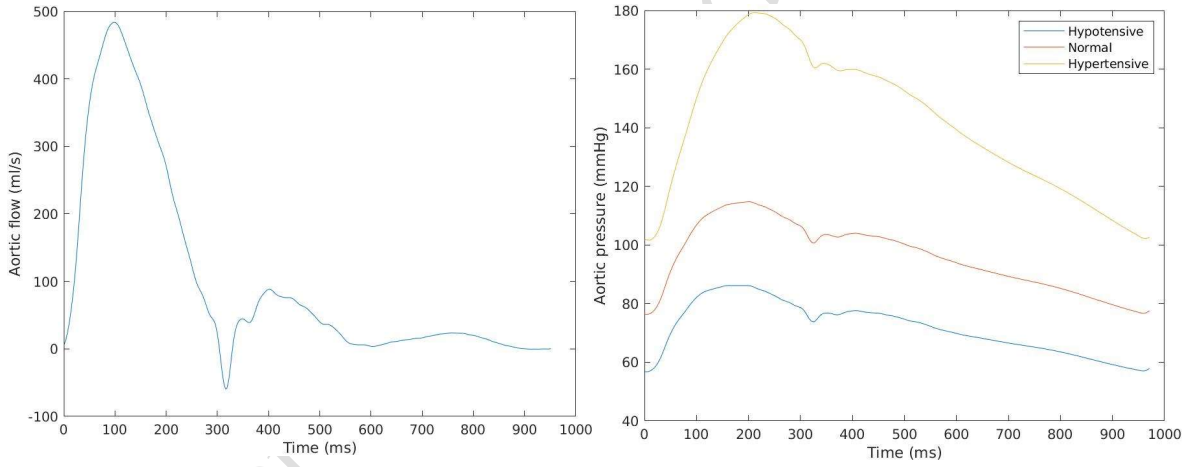
In this section, we will generate an aortic pressure signal based on known parameters of the model shown in Table 5.2. As mentioned in Section 5.3.1 item W1, we assign typical parameters to an entire class of subjects with a given pathology. The parameters are obtained and modified from [Kind10]. Pathologies considered are hypotension (low blood pressure) and hypertension (high blood pressure). Also, the parameters for a subject with normal blood pressure are included. The resistance of the aorta and the peripheral arteries is expected to increase with increasing blood pressure, and arterial compliance will decrease. The inertance also increases with the increase of blood pressure.

Table 5.2 Parameters for the circuit elements of the Windkessel model shown in Fig. 5.3 for hypotensive, normal and hypertensive subjects – modified from [Kind10].

Parameter	Z_0 $[mmHg \cdot s/ml]$	R $[mmHg \cdot s/ml]$	C_a [ml/$mmHg$]	L $[mmHg \cdot s^2/ml]$
Hypotensive	0.039	0.71	1.95	0.011
Normal	0.043	0.95	1.5	0.015
Hypertensive	0.05	1.4	0.7	0.02

Fig. 5.4a) represents the blood flow at the input of the model. Fig. 5.4b) shows a pressure pulse generated from the Winkessel model based on parameters from Table 5.2 for hypotensive, normal and hypertensive subjects. Normal diastolic blood pressure is 77 mmHg and normal systolic blood pressure is 115 mmHg (minimum and maximum of the red pulse). Please note that the pulses are shown for the steady-state. It takes about 5-10 seconds for the pulses to reach the desired amplitude (steady-state) when simulating the circuit presented in Fig. 5.3. Therefore, the pulses shown in Fig. 5.4b) were extracted from the pressure signal at the time range between 15.9 s and 16.9 s.

The arterial pulses can then be used as an input to the systems for oscillometric or continuous blood pressure measurements. We can evaluate how different physiological parameters affect blood pressure as well as the outcome of our device/system.



a) b)

Fig. 5.4 a) Arterial flow $Q_a(t)$ measured from a healthy subject used as an input to the model. b) Aortic pressure generated from the Windkessel model with the parameters for hypotensive, normal and hypertensive subjects from Table 5.2. Waveforms for the flow and pressure are shown for the steady-state.

5.3.2.3. Windkessel model with sensitivity analysis and parameter estimation

In this section, we will assume that the values of the circuit parameters are unknown and need to be estimated. They will be estimated based on the known input, which is the flow $Q_a(t)$ given in Fig. 5.4a) and the known pressure signal given in Fig. 5.4b), which is the blue signal for a hypotensive subject. The pressure signal corresponds to the hypotensive subject, and therefore, we should expect to obtain similar parameters as shown in the second row of Table 5.2. The following steps were followed:

1. Sensitivity analysis to determine the initial values of the parameters for the estimation
2. Selection of parameters that need to be optimized and their minimum and maximum values
3. Fitting the parameters (parameter estimation) of the model.

Parameter estimation through optimization could be performed without the sensitivity analysis. However, we expect that the optimization algorithm will perform better if good initial points are given and if the range of the parameters over which the optimization is performed is reduced. Therefore, we will perform sensitivity analysis first to find the values of the parameters that will be used as initial values for the parameter estimation.

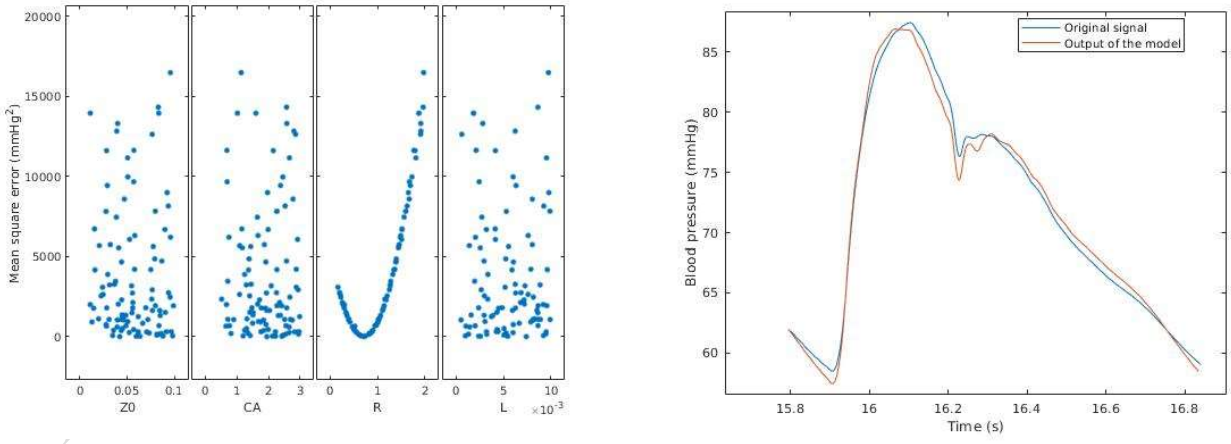
Sensitivity analysis was performed by generating 100 random values for each of the circuit parameters. The random numbers are generated from a uniform distribution. We will assume that we have a general idea about the range of parameters which is about 50% larger than the largest values of the parameters in Table 5.2 and 50% smaller than the smaller values of parameters in Table 5.2. We assume there is no correlation between parameters when generating random numbers, even though we know this is not the case. When determining what values to take out of 100 random values, we used mean square error (MSE) metrics between the samples of the measured pressure pulse and the generated pulse from the model during the steady-state (the segment of the pulse was taken between 15.9 s and 16.9 s). MSE is computed as:

$$MSE = \frac{1}{L}\sum_{k=1}^{L}[p_a(k) - \hat{p}_a\,(k)]^2$$

where L is the number of samples in the segment, $p_a(k)$ is the true aortic pressure and $\hat{p}_a\,(k)$ is the estimated aortic pressure obtained at the output of the model. The MSE versus different parameter values is shown on a scatter plot in Fig. 5.5a). We can see in this figure that only values of the peripheral resistance R show correlation with the mean square error. Therefore, the peripheral resistance R is the most sensitive parameter in the model.

After performing sensitivity analysis, the **selection of parameters** for which the MSE was minimum was performed. These parameters were considered as initial values for the optimization step. The selected values for our implementation were: $Z_0 = 0.013\ mmHg \cdot s/ml$, $R = 0.73\ mmHg \cdot s/ml$, $C_A = 1.65$ ml/$mmHg$, $L = 0.0015\ mmHg \cdot s^2/ml$. If we compare these values with the values of the parameters for the hypotensive subject, we will see that they are close but that there is still space for improvement. Minima and maxima of the parameters for the optimization are set as 50% smaller and larger than the parameters' values.

These parameters are used as an input to the optimization algorithm to perform **parameter estimation**. The optimization was done using Matlab function *fmincon,* which finds the minimum of a constrained nonlinear multivariable function. We used MSE as our cost function. The optimization reduced MSE from 2.35 $mmHg^2$ to 0.81 $mmHg^2$. The original pulse as well as the pulse obtained from the model after optimization, are shown in Fig. 5.5b). The obtained parameters after optimization are $Z_0 = 0.039\ mmHg \cdot s/ml$, $R = 0.71\ mmHg \cdot s/ml$, $C_a = 1.95$ ml/$mmHg$, $L = 0.011\ mmHg \cdot s^2/ml$. We conclude that the optimization methods provided acceptable results with low MSE. For more advanced methods for estimating parameters of the Windkessel model, please refer to [Kind10].



a) b)

Fig. 5.5 a) Scatter plot of the 100 randomly generated values of the circuit parameters versus the mean square error ($mmHg^2$) between the estimated and the measured pulse. b) Reference or measured aortic pressure pulse in blue and the pulse obtained from the Windkessel model after parameter estimation in red.

5.4. Modeling noise and assessing signal quality

In this section, we will describe the following topics:

1. presenting uncertainties in the parameters of the electrical components
2. modeling noise and motion artifacts
3. quantifying the effects of noise and motion artifacts on signal quality.

Tolerance of the resistors and noise will be used later in the chapter. Therefore, in this section, we will cover the blocks of Fig. 5.1 that are not on the main pipeline: modeling noise and artifacts and estimating SQI.

5.4.1. Modeling uncertainties in parameters of electrical components

5.4.1.1. Sampling from distributions related to tolerances or errors of common components

Let us consider several electronic components. Parameters of resistors are their resistance and tolerance. Examples of tolerance values are 0.1%, 1%, 2%, and so on. It is commonly assumed that the resistor values can be modeled with a uniform distribution. If the tolerance is given as the variable tol, then the values of the resistor can be sampled from the uniform distribution in the range $[R \cdot (1 - tol), R \cdot (1 + tol)]$.

If the resistor values follow a normal distribution and tolerance is presented using a coverage factor k (for example, $k = 2$) then we would first obtain standard uncertainty $u = tol/k$ and then sample from the distribution $N(R, u^2)$.

LSB error of an A/D converter is commonly presented using a uniform distribution, and it takes values [0, LSB] in the case of A/D converters that perform truncation and [-LSB/2, LSB/2] for A/D converters that perform rounding. Both resistor tolerance and A/D converter resolution errors are Type B errors based on GUM (see Chapter 2).

5.4.1.2. Determining uncertainty of the components based on data from the datasheet

Let us consider a sensor whose data sheet includes several errors, such as linearity and hysteresis errors. The total uncertainty of the sensor is then computed as the combined uncertainty. Let us assume that the linearity and hysteresis errors of a fictitious force sensor are 0.2 N over the full-scale and 0.3 N over the range, respectively, and they are both given at 95 % confidence. They are both type B measurements. In this case, combined uncertainty is computed as $u_c = \sqrt{(0.3\ N)^2 + (0.2\ N)^2} = 0.36\ N$ at 95 % confidence [Figliola12].

5.4.2. Noise

5.4.2.1. Types of noise

Noise from sensors, electrodes and amplifiers is often significant compared to the signal level because many bioelectric signals are in the μV range. In addition, interference from other devices and the environment, along with interference from other organs in the body, are also significant sources of disturbances.

Noise can be classified in different ways. Considering the portion of the overall signal bandwidth that the noise occupies, it can be classified as **narrowband** and **wideband/broadband**. **White noise** (such as thermal noise) occupies the whole bandwidth and is, therefore, broadband. Narrowband noise would be 60 Hz noise from the device itself.

Noise can also be classified as **low**, **middle** and **high**-frequency noise. In biomedical instrumentation, low-frequency noise can cause baseline wander.

If the unwanted signal comes from external sources, it is called **interference**. For example, electrodes and electric cables can be capacitively coupled to a nearby electric field. When the impedance of the electrode and cable pairs differ, the capacitive coupling will result in a potential difference due to interference. For example, a 60 Hz signal picked up from the power lines is an interference signal.

We have already introduced and explained the quantization noise when discussing A/D converters. For more information about quantization noise, please refer to Chapter 4.

Noise can also be additive or multiplicative. **Additive noise** refers to the unwanted signal added to the signal of interest. **Multiplicative** noise modulates the amplitude of the signal. In biomedical instrumentation, we commonly consider additive noise.

We can model the noise by looking at its spectral content. This can be modeled by a single parameter representing the slope of a spectral density function $S(f)$ that decreases monotonically with frequency [Sameni08]:

$$S(f) \propto \frac{1}{f^{\beta}}$$

where β is **the noise color parameter**. Different types of noise can be modeled in this case, including:

- **white** noise (β = 0),
- **pink** or **flicker** noise (β = 1),

- **brown** noise or the **random walk process** (β = 2).

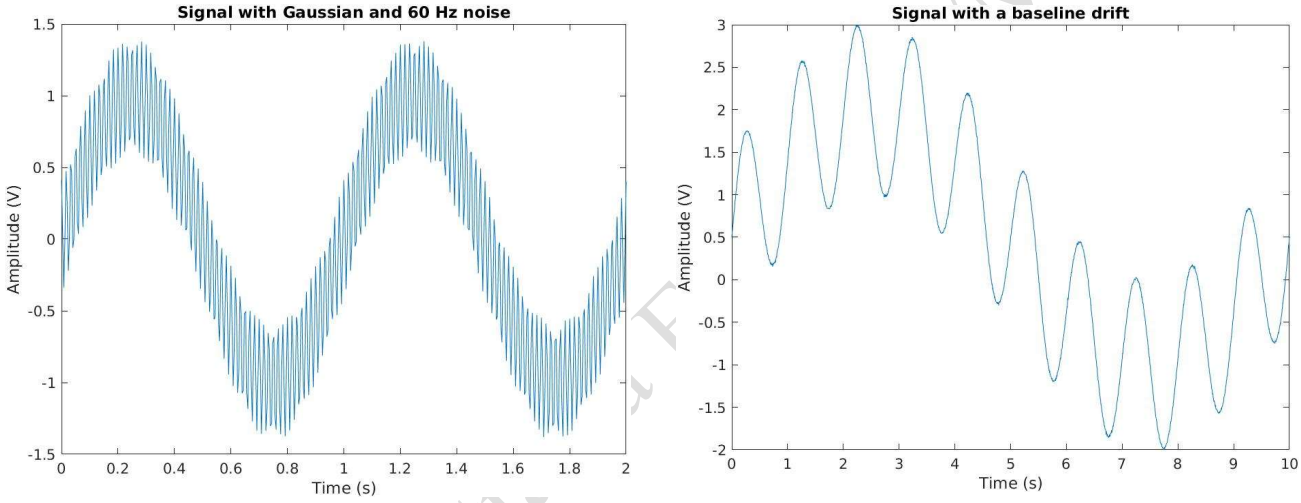
The simulation of pink or brown noise is normally done by generating white noise, transforming the noise signal into the frequency domain by applying Discrete Fourier Transform (DFT), altering frequency components of the DFT according to the defined slope of the spectrum and then performing the inverse DFT to transform the signal back into the time domain.

5.4.2.2. Modeling noise added to a biomedical signal

The signal with additive narrowband and white noises can be modeled as shown below:

$$y(t) = x(t) + Bsin(2\pi ft) + v(t),$$

where $y(t)$ is the noisy signal at the output, $x(t)$ is the clean signal of interest, B is the amplitude of the narrowband noise, f is the interference or narrowband noise frequency. In addition, $v(t)$ is the Gaussian white noise that is usually zero mean and has a standard deviation σ that is either considered known or estimated from the data. Let us assume that the signal of interest is a cardiac signal with an amplitude of $1\ V$, simulated in Fig. 5.6 as a sinewave signal of frequency 1 Hz sampled at 250 samples per second. An example of incorporating high-frequency noise is shown in Fig. 5.6a), where an interference sinewave signal with an amplitude of $0.4\ V$ and frequency of $60\ Hz$ is added to the signal of interest. This is an example of a powerline interference. Another example of a narrowband signal that is added to the original sinewave signal is shown in Fig. 5.6b). Here, the baseline interference is modeled as a sinewave signal with an amplitude of 1.5V, frequency of 0.1 Hz and a mean of 0.5 V. This can be used to model the baseline drift of ECG and PPG signals.



a) b)

Fig. 5.6 Input sinewave signal with an amplitude of $1\ V$ and a frequency of $1\ Hz$ corrupted with Gaussian white noise with standard deviation $0.01\ V$ and a) $60\ Hz$ noise corresponding to power line interference and b) $0.1\ Hz$ corresponding to baseline drift.

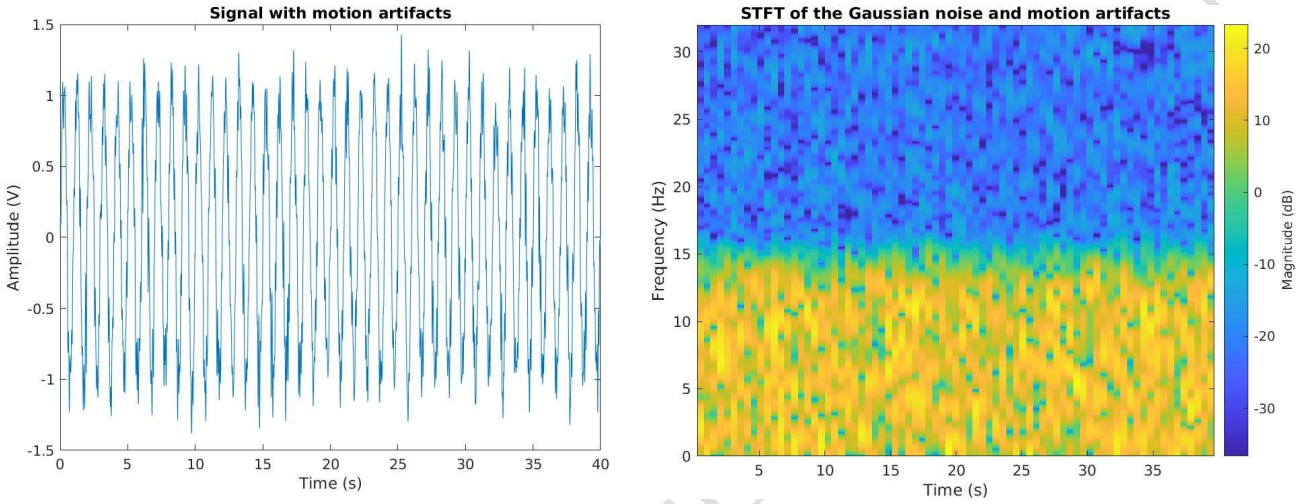
5.4.2.3. Modeling motion artifacts

Motion artifacts are one of the most common disturbances for biomedical devices and they affect all the signals discussed in this book. The frequency range of motion artifacts overlaps with the frequency range of the biomedical signals, and therefore, it cannot be filtered out easily using traditional filtering methods. These artifacts can arise due to factors such as walking, motion of the patient's arm on which the sensor is attached (such as movement of the arm up and down), and also environmental conditions such as taking measurements in a moving vehicle. These artifacts tend to degrade the quality of the measured signal.

Motion artifacts from patient movement and movement from a vehicle are inherently low-frequency signals. There are multiple ways to simulate continuous motion artifacts, such as ones due to walking. A simple way is to generate Gaussian white noise and then pass it through a lowpass filter, thus only keeping the low-frequency components of the signal which resemble the noise introduced by motion artifacts [Thakkar04]. This approach is shown in Fig. 5.7, which shows the motion-corrupted signal in the time domain as well as the time-frequency domain representation of the noise. The signal of interest is the sinewave that simulates a cardiac signal with an amplitude of 1 V and frequency of 1 Hz sampled at 250 samples per second. The continuous motion artifact is modeled as a Gaussian noise that is zero mean and 0.4 V standard deviation and is filtered by a lowpass filter with a cut-off frequency of 10 Hz. We can see that this motion artifact is causing changes in the amplitude of the signal of interest. Time-frequency domain representation in Fig. 5.7b) is

obtained by applying the Short-time Fourier Transform (STFT) to the motion artifact signal with the following parameters: Discrete Fourier Transform (DFT) length of 8,096 points windowed by a Hamming window of 1 s length with 50% overlap. STFT is a DFT computed over short and overlapping signal segments, and it provides information about both time and frequency for situations in which frequency components of the signal vary over time. On the other hand, the standard DFT provides the frequency information averaged over the entire time interval. The STFT spectrogram shows an intensity plot of the STFT magnitudes over time. As shown in Fig. 5.7b), the signal with motion artifacts occupies an almost fixed portion of the frequency spectrum over time. For more information about Short-time Fourier Transform, please refer to [Mitra01].

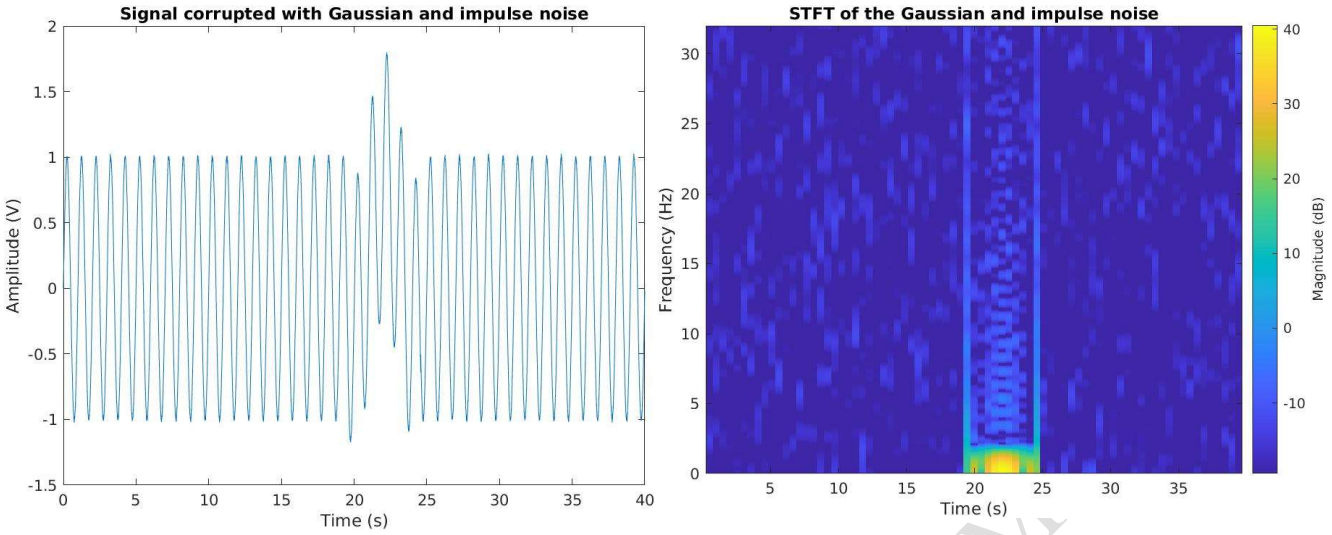
Three types of more realistic motion artifacts models are shown in [Farago21]: ARMA model, hidden Markov model and recurrent neural networks. Describing these models is outside of the scope of this book.



a) b)

Fig. 5.7 a) Input sinewave signal with an amplitude of $1\ V$ and a frequency of $1\ Hz$ corrupted with Gaussian white noise with standard deviation of $0.01\ V$ and a lowpass filtered Gaussian noise with the amplitude of 0.4 V that simulates motion artifacts. b) Time-frequency domain of the Gaussian white noise and the motion artifact signals, shown in the frequency range from 0 Hz to 32 Hz.

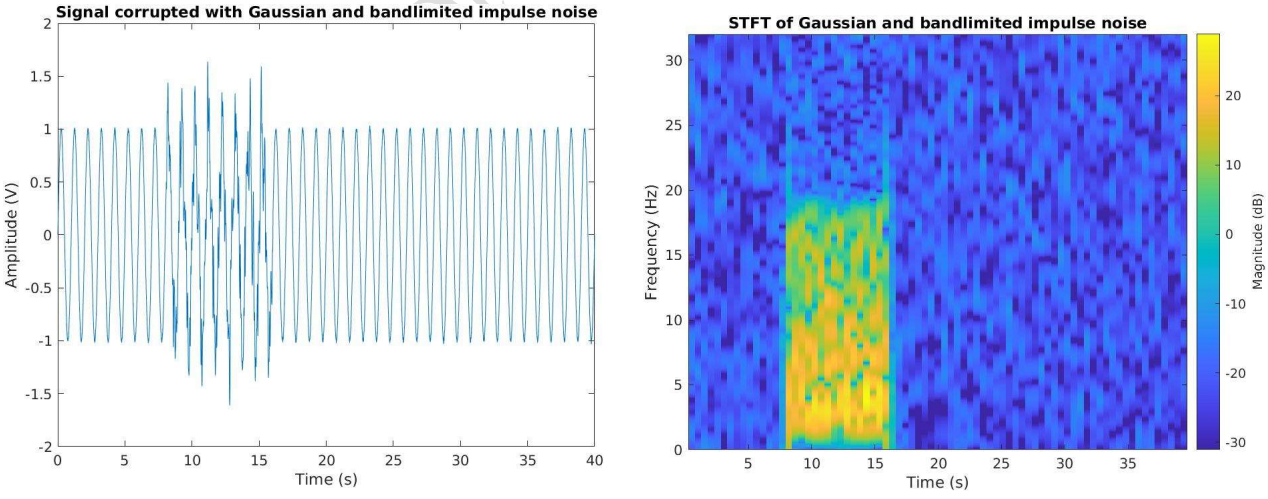
Very often, the motion artifacts are not continuous but occur over a short time intermittently. An example includes a quick movement of the arm while measuring the blood pressure. That movement will result in a pulse on top of the signal of interest. Impulse artifact can be simulated by the *sinc* function [Li09] in the discrete domain as $a_{imp}(n) = m \cdot sin\ (\pi n T_s/\eta)/(\pi n T_s)$. The central lobe of the *sinc* function can be used as an impulse artifact and can be added to the signal of interest. The duration of the artifact is determined by the factor η. Fig. 5.8a) shows the signal of interest (that is the same as the one described in Fig. 5.7) with the impulse motion artifact. The impulse is modeled to occur anywhere randomly. For the *sinc* function, we selected η =1.5 resulting in about $5\ s$ long movement artifacts. The amplitude of the movement artifact is $m = 1.2\ V$. Fig. 5.8b) shows the time-frequency domain representation of the motion artifact signal computed using the Short-time Fourier Transform with the same parameters as in Fig. 5.7. As seen from the figure, change in the frequency components over time occurs only for the duration of the motion artifact. A similar motion artifact model can be applied to biomedical signals such as the PPG with some modifications in amplitude and duration.



a) b)

Fig. 5.8 a) Input sinewave signal with an amplitude of 1 V and a frequency of 1 Hz corrupted with an impulse noise represented using the *sinc* function in the time domain. b) Time-frequency domain of the Gaussian white noise and the motion artifact signals signals shown in the frequency range from 0 Hz to 32 Hz.

In [Li09], motion artifacts are modeled as Brown noise, filtered using a band-pass filter in the frequency range between 1.8 Hz and 18 Hz. This corresponds to the motion artifacts obtained by, for example, dragging the clothes over the electrode or the transducer [Li09]. Fig. 5.9a) shows the signal of interest that is the same as the one described in Fig. 5.7 corrupted with the described motion artifact in the interval between 8 s and 17 s, while Fig. 5.9b) shows the time-frequency representation of the motion artifact signal computed using the Short-time Fourier Transform with the same parameters as in Fig. 5.7. The time when the motion artifact occurs as well as the duration of the artifact, are modeled as uniform random variables. The amplitude of the motion artifact before being filtered was 5 V.



a) b)

Fig. 5.9 a) Input sinewave signal with an amplitude of 1 V and a frequency of 1 Hz corrupted with an impulse brown noise. b) Time-frequency domain of the Gaussian noise and motion artifact signal signals shown in the frequency range from 0 Hz to 32 Hz.

5.4.3. Signal quality

Signal quality is the degree to which a set of signal characteristics fulfill the requirements.

As presented in [Orphanidou18], quality assessment aims to identify and quantify the instances of artifact in a signal segment so that the information of interest extracted from that signal segment can be ignored or corrected. Acceptable signal quality depends on the application of interest. The output of the algorithm is a **Signal Quality Index** (SQI) that can be binary or have several degrees of acceptability.

The **quality factors** are factors that affect the quality of the data. They include the quantization noise, noise introduced by filters and amplifiers, motion artifacts, environmental noises and interference. The procedure for extracting the SQI normally involves several steps, including:

- Feature extraction, where these features are supposed to carry important information regarding signal quality. Some of the features include:
    - morphological features of the signals or pulses,
    - physiological feasibility,
    - time-domain and spectral features,
    - statistical features,
    - comparing outputs of multiple different algorithms that process the same data.
- Using features and decision methods based on empirical thresholds or machine learning to come up with **quality indices**
- Combining the indices into a single SQI.

Algorithms for extracting some of the features are universal, while the thresholds or methods for comparison are different among biomedical signals. For example, to determine the signal-to-noise ratio (SNR) of any signal, the ratio between in-band spectral power and out-of-band spectral power is computed. In-band frequencies are different for different physiological signals. For example, the range between 5 Hz to 14 Hz could be used as an in-band frequency for the ECG signal.

Morphological features are different for every physiological signal. For example, for ECG signal, the time and amplitudes of waves of interest, including P wave, T wave and QRS complex, are the features. **Template matching** could be used to compare different pulses against a reference pulse. It is important to check if the extracted features are physiologically feasible. Features such as heart rate, the time difference between different points on the pulse, extracted amplitudes and their ratios can all be checked for feasibility.

Time-domain features include, for example, the rise and fall time of the pulse. **Spectral features** include ratios of the spectral power among different frequency regions of interest.

Statistics features include the mean, variance, skewness, kurtosis and others.

Outputs of different algorithms processing the same signal segment should be the same when the signal is of good quality. However, if the signal is not of good quality, then different algorithms will be affected differently by noise or interference. Therefore, we should expect that the results of the algorithms would be significantly different. An example of this method is using two different algorithms for detecting QRS complex in the ECG signal. The SQI is defined as the total number of matched QRS complex beats between the two algorithms divided by the total number of beats.

SQI can be computed for a **segment of interest,** such as a 10 s interval or *each detected pulse*. Below, we will show examples of SQIs computed based on statistical and spectral features where SQI is computed for the overall segment of interest. SQI based on template matching example is shown as well – it allows for evaluating SQI of every pulse.

Statistical feature: Kurtosis

The reason behind using kurtosis of the biomedical signal is to distinguish between noise and the signal. The random noise samples often follow Gaussian distribution, whereas biomedical signals do not follow it. The kurtosis of any univariate Gaussian distribution is 3 and therefore the kurtosis of the empirical distribution obtained from the signal samples is compared to this value. The estimate of kurtosis of a signal can be represented as:

$$Kurt(x) = E\{x - \mu\}^4/\sigma^4$$

where $x(n)$ is the signal segment of L samples, μ and σ are the mean and the standard deviation of the signal $x(n)$ respectively and $E\{x-\mu\}$ is the expected value for $x-\mu$. In order to compute the SQI for the ECG signal, the authors in [Li08] selected a threshold of 5 for kurtosis. Therefore, if the kurtosis of the distribution of a signal in a given sequence is greater or equal 5, the kurtosis-based SQI (kSQI) is $kSQI = 1$, otherwise $kSQI = 0$.

Signal-to-noise ratio based on the spectral power ratio

We have already mentioned that an estimate of the SNR can be computed as the ratio between in-band spectral power and out-of-band spectral power. In the case of ECG signal, with a typical QRS, all of the power is concentrated below $30\ Hz$ and the peak for the power occurs in the range of $5Hz$ to $14\ Hz$ [Li08]. The spectral distribution ratio is calculated as the sum of the power of the ECG in the range of $5\ Hz$ to $14\ Hz$ to the power of the same signal in the range of $5\ Hz$ to $60\ Hz$. The spectral distribution ratio (SDR) can be represented as:

$$SDR = \int_{f=5}^{f=14} P(f)df \Big/ \int_{f=5}^{f=60} P(f)df$$

The signal quality index (pSQI) is then computed by comparing SDR with a predefined threshold. SDR has a value between 0 and 1. If all the energy of the ECG signal is contained in the range between $5\ Hz$ and $14\ Hz$, SDR would be 1, but that would not correspond to the spectrum of real ECG signal that also has frequency components outside of this range. Therefore, it was empirically determined that signals with SDR above 0.8 and below 0.5 are unacceptable [Li08]. As we will show in Section 5.4.3.1, we selected 0.5 and 0.9 empirically as our thresholds.

We showed an example of computing pSQI for an ECG signal, but a similar analysis can be made for almost any biomedical signal. For other physiological signals, different frequency ranges and thresholds should be selected.

Template matching using average correlation coefficient

Template matching requires that all the pulses are first detected. Then, the Pearson correlation coefficient is computed between each pulse and the *waveform template*. We expect the correlation between the pulses and the waveform template would be high if there is no noise or distortions in the signal. The waveform template is normally unavailable but is computed as the average pulse waveform in a segment of the biomedical signal of interest by performing beat/pulse detection and then ensemble averaging over all pulse waveforms. Ensemble averaging all pulse waveforms in a given segment provides the pulse waveform template.

In [Orphanidou18], the template waveform of PPG signals was obtained by an ensemble averaging 10 pulses. The correlation coefficient was computed between each of the ten pulses and the template waveform and averaged. An average correlation coefficient greater or equal to 0.86 was considered acceptable, resulting in the template matching SQI or 1 (tSQI=1). The average correlation threshold was determined empirically.

To summarize, the following steps are needed:

1. Some applications require that the signal is first filtered using a low-pass filter
2. Finding time instances and amplitude of the fiducial points of interest of each pulse (for example, time instances and amplitude of R peaks of the ECG signal)
3. Extracting pulses and performing ensemble averaging in order to find the waveform template
4. Calculating the correlation coefficient between each pulse and the template
5. Determining tSQI for each pulse by comparing the correlation coefficient of each pulse with the threshold
6. Determining tSQI of the overall segment by averaging the correlation coefficients for the segment.

5.4.3.1. Example of signal quality assessment: ECG signal

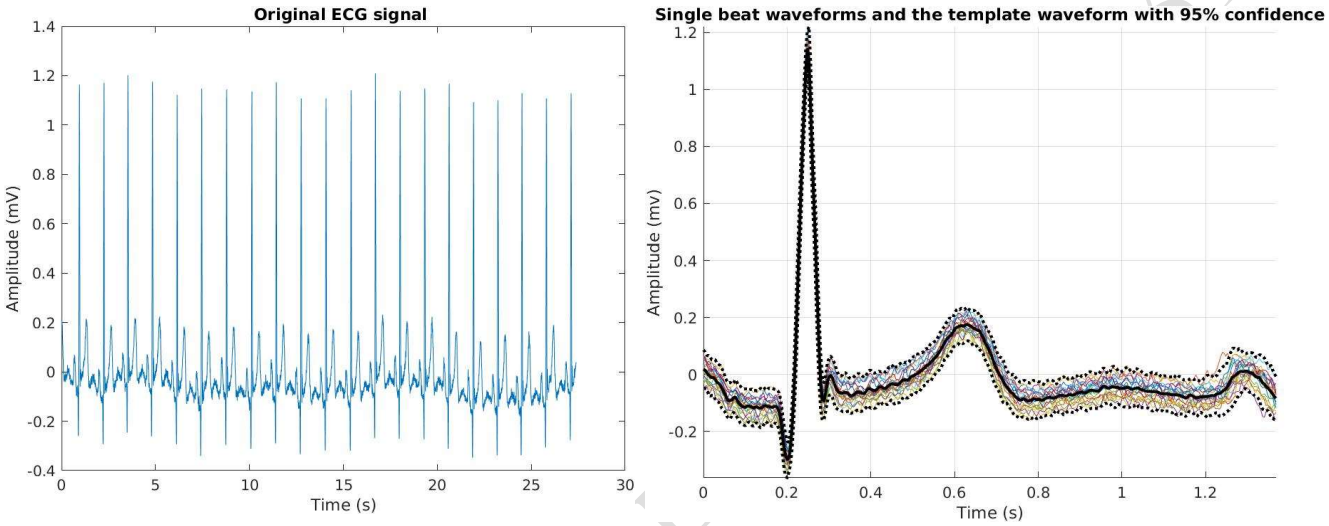
In this section, we will compute previously described SQIs for an example ECG signal. A high-quality ECG signal was collected for about 27 seconds and sampled at 250 samples per second.

We show the SQIs of the ECG signal in 3 cases:

1. No noise is added to the original ECG signal
2. $60\ Hz$ noise with an amplitude of $0.1\ mV$ and random Gaussian noise with a mean of 0 mV and standard deviation of $0.1\ mV$ is added to the original ECG signal resulting in a low SNR of $3.2\ dB$.
3. Random Gaussian noise with a mean of $0\ mV$ and standard deviation of $0.01\ mV$ was added together with the impulse noise represented using the brown noise described in Section 5.4.2.1 with the amplitude of $2\ V$ followed by a bandpass filter with the frequency range of [1 Hz, 10 Hz] resulting in a SNR of $4.2\ dB$.

In Case 1, the original signal is considered without additional artificial interference and noise, as shown in Fig. 5.10a). Fig. 5.10b) shows the template waveform obtained by performing an ensemble averaging of 19 beats. About 1.05 of the total averaged pulse duration is taken when performing averaging. The bold black line in Fig. 5.10b) represents the mean of the averaged pulses, which is actually the template waveform. Dashed lines represent the averaged value of the signal ± two standard deviations. All other lines represent 19 beats overlapped one on the top of the other. The same representation of the averaged pulses is used in Fig. 5.11 and Fig. 5.12. Segmentation of the ECG signal to find the R peaks and detect every beat is done based on the work [Sedghamiz15]. Ensemble averaging is performed based on the work presented in [Parker13].

In addition, even though R peaks of the ECG signal are detected, we start averaging 0.25 s before the R peak. Therefore, the R peaks of all beats are positioned exactly at 0.25 s. Table 5.3 shows that all SQIs are one. The correlation coefficient between the template waveform and every single beat is high, which means that the tSQI for every beat is 1.



a) b)

Fig. 5.10 Case 1: original ECG signal without added noise. a) The waveform of the ECG signal and b) Overlapped waveforms for each ECG beat (different color lines) and their ensemble-averaged template waveform (black line) together with 95% confidence intervals (black dotted lines).

In Case 2, the signal is noisy, as shown in Fig. 5.11a). Fig. 5.11b) shows the template waveform obtained by averaging the 19 beats of the signal shown in Fig. 5.11a). It is quite clear that the noise is high; therefore, the correlation coefficient between each pulse and averaged template waveform is low. Therefore, tSQI for each pulse is tSQI=0, as shown in Fig. 5.11a). The average correlation coefficient shown in Table 5.3 is $r = 0.82$.

Lowpass filter with a cut-off frequency of 15 Hz was applied to the signal. The filtered ECG waveform, as well as its template waveform, are shown in Fig. 5.11c) and Fig. 5.11d), respectively. The average correlation coefficient, in this case, is increased to 0.97, and all tSQIs per pulse are 1, as shown in Fig. 5.11c). Kurtosis and power ratio are shown for the non-filtered signal in Table 5.3. Kurtosis dropped in comparison to Case 1, but it is still much higher than 5, which means that the kSQI=1. The power ratio dropped significantly compared to the power ratio of Case 1, but it is still above 0.5 and, therefore, pSQI=1. This power ratio drop is expected because of the 60 Hz interference since much of the noise spectral components are outside the 5-14 Hz range. Therefore, it is clear that multiple SQIs need to be considered when analyzing the signals.

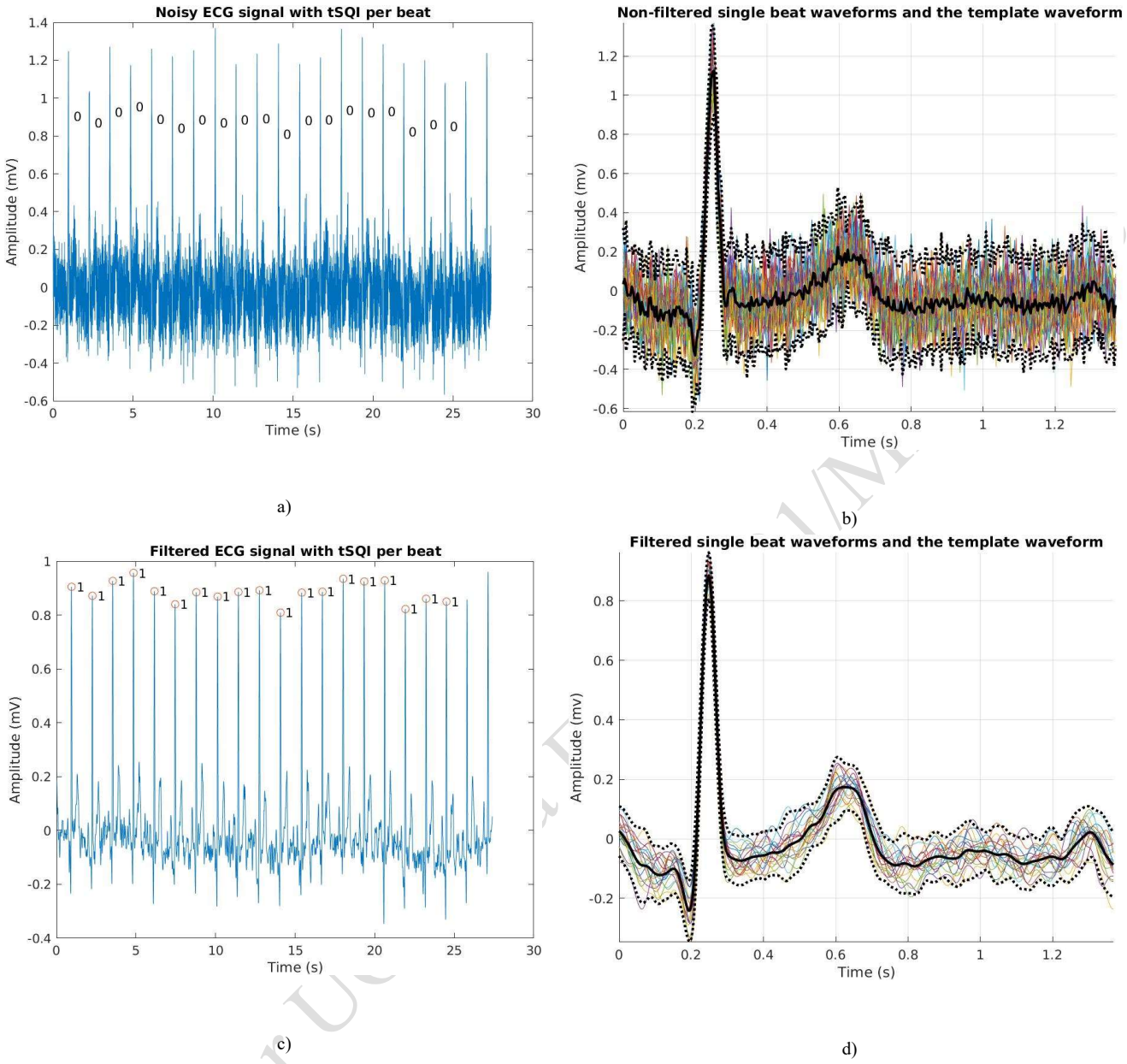
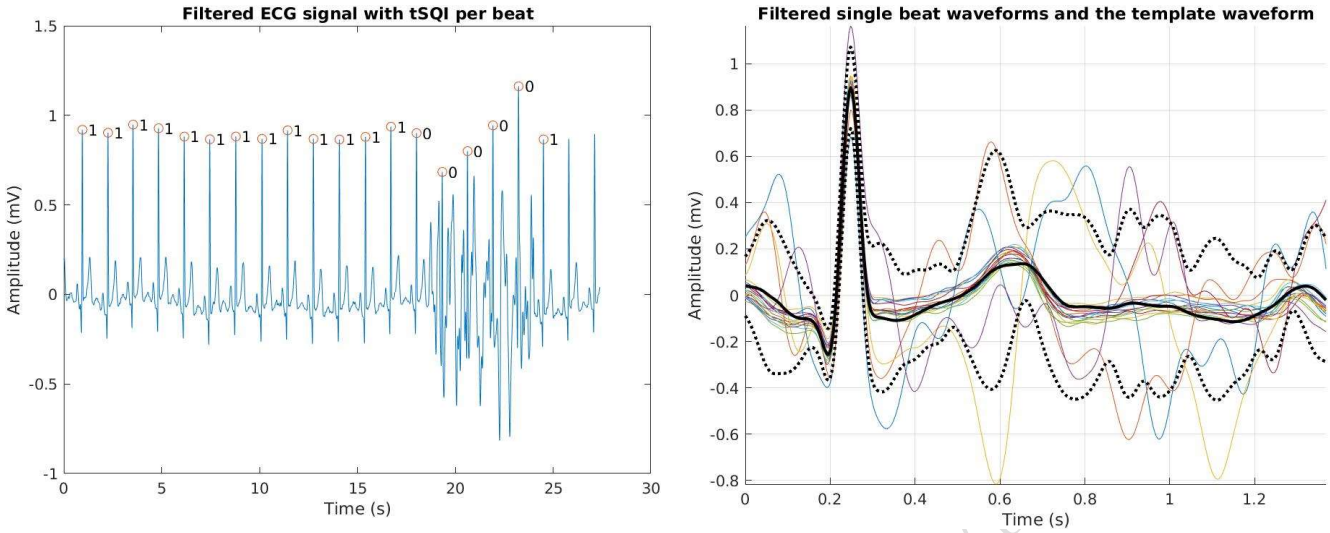


Fig. 5.11 Case 2: ECG signal with Gaussian white noise and 60 Hz interference. a) The waveform of non-filtered ECG signal with tSQIs per beat, and b) Overlapped waveforms of each beat of the ECG signal from a) and the ensemble-averaged template waveform. c) The waveform of the low-pass filtered signal with identified R peaks and tSQIs per beat, and d) Overlapped waveforms of each beat of the ECG signal from c) and the ensemble-averaged template waveform.

In Case 3, the signal is noisy only in the range between 18 s and 25 s, as shown in Fig. 5.12a). This situation corresponds to a short-term motion artifact, and it was simulated by adding bandlimited brown noise in the frequency range of 1 to 10 Hz, which overlaps with the frequency range of the ECG signal. Fig. 5.12a) shows the signal after low-pass filtering with a 15 Hz cut-off frequency. Fig. 5.12b) shows the template waveform obtained by averaging the 19 beats of the signal. It is clear that the pulses corrupted by noise have a different shape than the original ECG pulses. Fig. 5.12a) also shows the tSQI computed for every pulse. The correlation coefficient of corrupted pulses is significantly below the threshold; therefore, the tSQI for these pulses is zero. The average correlation coefficient is around 0.87 for both filtered and unfiltered ECG signals, as shown in Table 5.3. As expected, filtering did not help much in this case to improve signal quality, and this type of noise will require a different kind of filter which is outside of the scope of this book. As expected, the power ratio did not change much compared with Case 1 – it is very high. Kurtosis is reduced in comparison to Case 1 because of the added noise.



a) b)

Fig. 5.12 Case 3: ECG signal with brown noise. a) The signal waveform with identified R peaks and tSQIs per beat – the signal is filtered with a low-pass filter with 15 Hz cuff-off frequency and b) Overlapped waveforms for each ECG beat (different color lines) and their ensemble-averaged template waveform (black line) together with 95% confidence intervals (black dotted lines).

Table 5.3 Features and SQIs for the ECG signal with different types of noise

	Kurtosis, kSQI	Power ratio, pSQI	Average r – no filtering, tSQI	Average r with filtering, tSQI
Case 1	Kurt=22.2, kSQI=1	ratio=0.88, pSQI=1	r=0.99, tSQI=1	r=0.99, tSQI=1
Case 2	Kurt=11.8, kSQI=1	ratio =0.63, pSQI=1	r=0.82, tSQI=0	r=0.97, tSQI=1
Case 3	Kurt=11.9, kSQI=1	ratio =0.89, pSQI=1	r=0.88, tSQI=0	r=0.87, tSQI=0

5.5. Uncertainty propagation in systems

The goal of uncertainty propagation is to show how uncertainties in the parameters of the model or the circuit affect the confidence intervals of the estimate at the output. Uncertainty propagation will be done here using Monte Carlo methods. We will show the propagation of uncertainties through a charge amplifier circuit described in Chapter 4.

5.5.1. Example: Uncertainty propagation through a charge amplifier

In this example, we will show uncertainty propagation through the charge amplifier presented in Fig. 4.12 and Example 4.2. To remind the reader, this circuit uses the piezoelectric transducer as an accelerometer. The objective is to amplify the signal obtained from the piezoelectric sensor and estimate the uncertainty at the output if we assume that resistors and capacitors have 1% tolerance.

The input to the circuit represents a sinewave with 20 Hz frequency and an amplitude of 5 pC (peak to peak is 10 pC). This signal is differentiated to obtain the current and then used as an input to the controlled current source $i_{IN} = dq/dt$. The only difference in comparison with Fig 4.12 from Chapter 4 is that the resistor R7 has a value of 130 kΩ to reduce the likelihood of saturating the amplifier.

Uncertainty propagation is performed in the following way:

- M=200 values of the resistors R_f, R_i, R_5, R_7 and R_8 and the capacitor C_f were sampled from a normal distribution with the mean value equal to their nominal value and the standard deviation equal to 0.5% of their nominal value. We assume that 1% tolerance of the component means that 95% values of that component will be in the ±1% of the nominal value (2σ).
- We computed the peak-to-peak amplitude of the signal at the output by subtracting the maximum and minimum values of the signal.
- The simulation was run M times, and M peak-to-peak amplitude values were obtained.
- From these M peak-to-peak amplitude values, we computed the standard error in percentage and the 95% confidence intervals. The histogram of M peak-to-peak amplitude values is also presented. Finally, 95% confidence is computed by sorting the output results and then taking the value for the low confidence interval which corresponds to the 2.5% of the peak-to-peak amplitudes, and for the high confidence interval which corresponds to the 97.5% of the peak-to-peak amplitudes. More accurate results would be obtained if we selected a larger value for M.

Derivation of the sensitivity analysis is given in [AD17], and it is performed using a derivative approach based on the derived expression of the gain: $\left(1+\frac{R_7}{R_8}\right)\frac{1}{C_f}$. Please note that in the calculations in [AD17], it is assumed that the standard error is 1%.

We evaluated the uncertainty at the output using the Monte Carlo method presented above. The uncertainty mainly depends on the variations of the components R_7, R_8 and C_f, as expected. The histogram of the magnitudes is shown in Fig. 5.13. The standard error at the output is estimated to be 0.8%, while the 95% confidence intervals are [2.73 V, 2.81 V]. The mean value is 2.76 V.

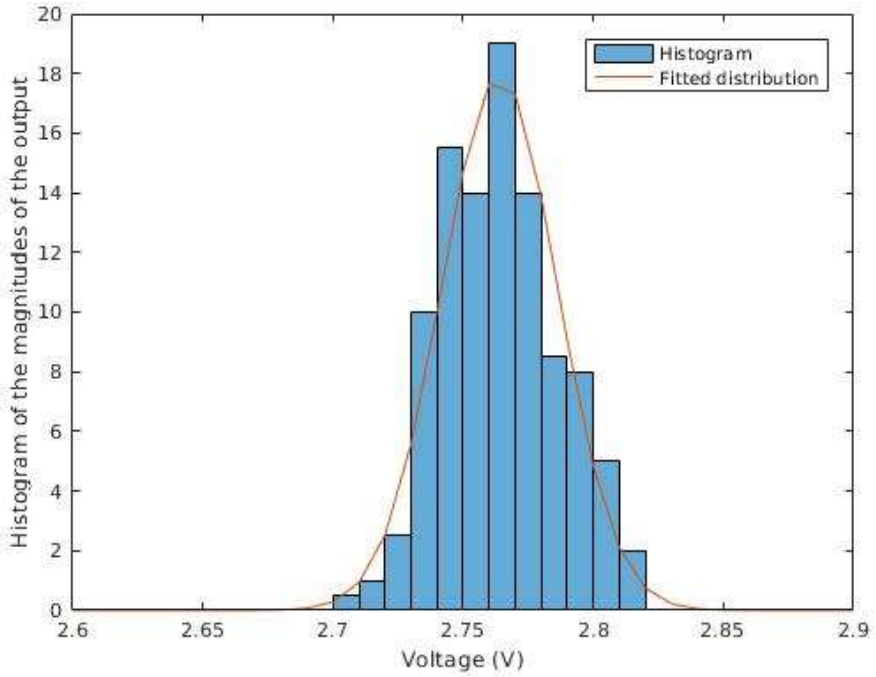


Fig. 5.13 Histogram of the peak-to-peak amplitudes of the output voltage of the charge amplifier in case it is assumed that the tolerance of each component is 1%.

Confidence intervals computed in this way can be used in more complex circuits in which we can assume that the charge amplifier presented here is a black box with the uncertainty quantified using the confidence intervals.

5.6. Modeling software

5.6.1. Real-time signal processing

Real-time applications are unique in regards to how data is received and processed:

- The data comes as a continuous stream
- The data is processed continuously
- Processing time is important since the **latency** (the input to output delay of the system) introduced by the processing algorithm needs to be small.

Real-time processing can be classified based on how frequently the output is obtained [Ackenhusen99]:

- **Stream processing** where all computations related to one input sample are completed before the next input sample arrives
- **Block processing** where each input sample $x(n)$ is stored in a memory or a buffer before any processing is performed on it. After L input samples have arrived, the entire block of samples is processed at once.

Stream processing includes processing the last L samples, where $L \geq 1$, and producing the output signal at each sampling period T_s. If $L = 1$, then the output is produced for each input without taking into account any historical data. If $L > 1$, then processing is based on a sliding window that requires storing the last L input samples for processing. If the processing is completed before the next sample arrives, then the system is performing real-time processing. Therefore, if the time required for processing L samples is T_{proc}, the condition for real-time processing is that $T_{proc} \leq T_s$.

Block processing includes processing the last L samples, where L ≥ 1, and producing the output signal after L samples, which means that the data at the output appears at a different rate from the data at the input. The output sampling period is $T_{sy} = (L - L_{over})T_s$. Blocks can be overlapped by the number of samples L_{over}, where $0 \leq L_{over} < L$ samples. $L_{over} = 0$ means that the blocks are not overlapped. If $L_{over} = L - 1$, then this is stream processing and $T_{sy} = T_s$. Therefore, we can see that stream processing can be presented as a special case of block processing. In order to perform real-time processing, the following condition needs to be met $T_{proc} \leq (L - L_{over})T_s$. Block processing is useful for implementing algorithms, such as discrete Fourier transform, that require processing vectors. The disadvantages of block processing are additional memory requirements and latency. Block processing with non-overlapping blocks is shown in Fig. 5.14, where L=4 input samples are used to produce one output sample and $L_{over} = 0$. An output sample appears after $L \cdot T_s + T_{proc}$.

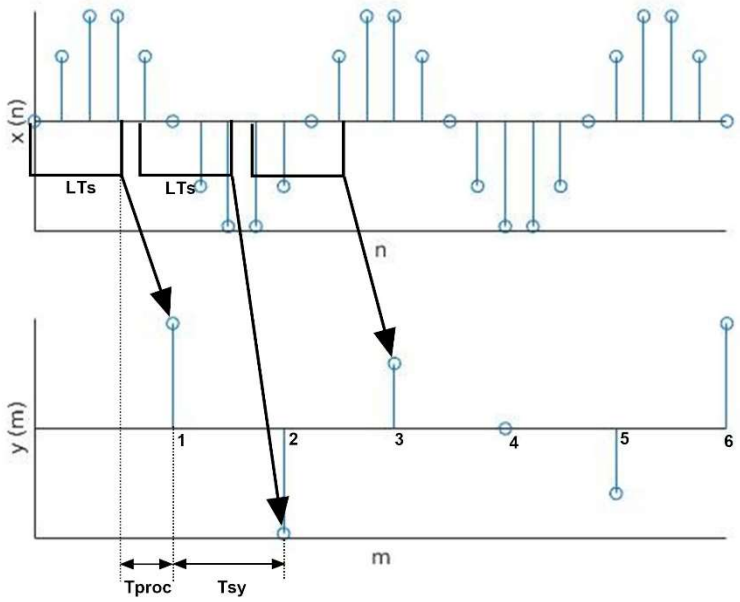


Fig. 5.14 Block processing where blocks are not overlapped

5.6.2. Real-time processing: hardware vs. software

Signal processing can be done in hardware using *analog or digital electronics* or *in software*. Both approaches support stream and block processing; however analog processing is better suited for stream processing. We will show in the next section both types of processing on an example of a circuit for

generating an alarm if breathing cessation takes longer than it was initially expected. Analog implementation in Section 5.6.3.2 is done using an integrator followed by a comparator. In Section 5.6.3.3, the analog signal is converted into the digital signal which is then processed in software.

5.6.2.1. Processing using analog hardware

In Chapters 3 and 4, we covered several analog components that are commonly used in biomedical devices. We also described components used for conversion between the analog and digital domains: A/D and D/A converters. However, we did not present digital electronics. In this book, we will not use digital electronics such as flip-flops, registers, counters, accumulators and so on – processing will be done either using analog electronics or using software. Analog electronics allows us to perform basing arithmetic operations, filtering, integration, differentiation, and similar. It provides very limited opportunities for processing in comparison to the processing opportunities in software.

5.6.2.2. Processing in software

Almost all algorithms nowadays are executed in software. Processing in software is done after the signal is first converted into a digital representation using A/D conversion. Then, the processing is done on a processor which is a part of a device or on a computer. The output of the A/D converter is normally an integer that is in the range defined by the resolution N of the A/D converter $[0,\ 2^N - 1]$. This integer is normally first scaled back into the range that depends on the application. For example, if the A/D converter is used for blood pressure measurement and the range of blood pressure that is measured is between $0\ mmHg$ and $300\ mmHg$, then $2^N - 1$ is scaled to correspond to $300\ mmHg$. After that, the scaled signal is either immediately processed or buffered for block processing.

Please note that processing in software can be done in real-time as well as offline. **Offline** or **batch** processing requires that the data is collected and available after the experiment or the simulation. An example of a device that performs offline processing is an oscillometric blood pressure device. There, all the signals of interest are collected and then processed. There is no real deadline for processing except that the results should be presented to the user as soon as possible to improve the user experience and reduce the wait for the results. However, even if there is a delay in processing data, there will be no consequences like in real-time processing, where delays in processing might result in missing data at the output. Since some medical devices rely on online real-time processing and some on offline processing, we will use both concepts in the book.

5.6.3. Example: stop-breathing detection

A breathing belt is a device a subject wears around her or his chest that can be used to extract the respiration signal based on the displacement of the chest during inspiration and expiration. The breathing belt can be used to estimate breathing rates over time, analyze breathing patterns and indicate that the breathing rate is too high or too low. Here, we will show a simplified system that can extract the breathing signal and detect that the subject is not breathing for some time that is longer than a predefined threshold.

5.6.3.1. Description of the model

In this section, we show an example of a circuit that collects the breathing signal and generates an alarm if the amplitude of the breathing signal is zero for some time. This system can be used as a central sleep apnea detector to detect the number of times the patient had apnea episodes or as a baby monitor system. The system is shown in Fig. 5.16. The model has six sub-models, including breathing signal and noise generation model not shown in the figure, strain gauge with Wheatstone bridge, differential amplifier, low-pass filter, pulse generation, and alarm generation. An alarm is generated if the breathing ceases for about 10 seconds. Please note that $V^+ = 5\ V$ and $V^- = -5\ V$ in this circuit.

Signal and noise generation:

The first block, called Input, represents the breathing simulator. We present breathing as a sinusoidal signal with the frequency f_B. Breathing cessation is modeled as a DC signal with an amplitude of zero. Let us assume that one breath involves inhaling air to full lung capacity, where full lung capacity corresponds to the maximum strain of the strain gauge, and then exhaling all air out of the lungs ending up with zero strain. Assuming a maximum strain of 0.02 and a strain change of 0.01, we represent breathing as signal $-0.01\cos(2\pi f_B t) + 0.01$. In our example, the breathing frequency is 15 breaths per minute which means that the breathing frequency is $f_B = 0.25\ Hz$. Gaussian white noise is also added. The signal together with the activity periods (high level means breathing and 0 level means no breathing) are presented in Fig. 5.15.

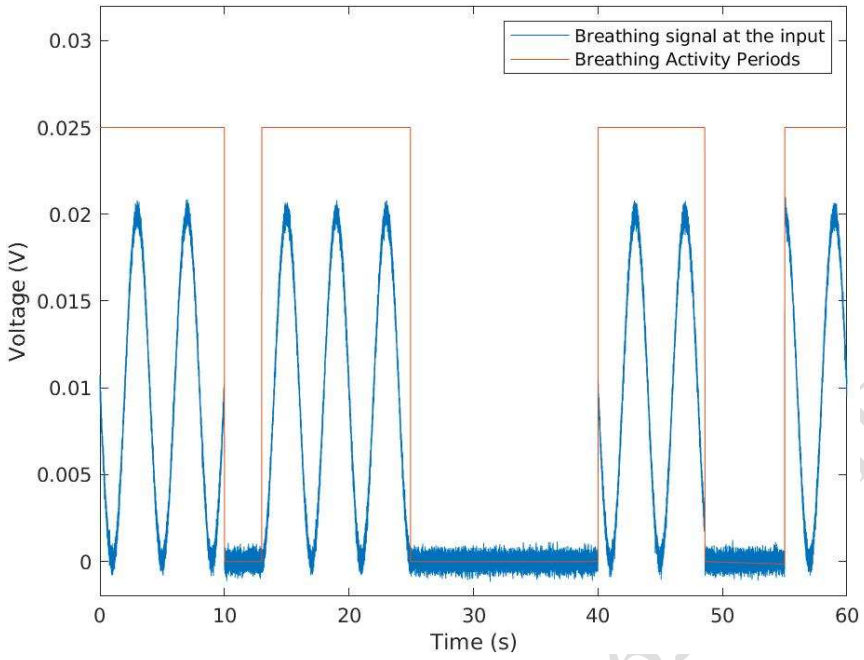


Fig. 5.15 Noisy breathing signal (blue line) and periods of active breathing (high level of the red line) and breathing cessation (low level of the red line).

Transducer and amplifiers:

For the strain gauge, the Gauge factor G is set to 2, and the nominal resistance of the strain gauge is 100 Ω. The change in resistance ΔR for the maximum strain of 0.02 is given as $\Delta R = GR\varepsilon = 4\ \Omega$. In this circuit R =R_1 = R_2 = R_3 = 100 Ω. Please note that the Wheatstone bridge is balanced to $V^+/2$ and not to 0 V.

The voltage V_1 in Fig. 5.16 is the voltage at the output of the bridge, which is also the input to the differential amplifier. The voltage V_1 for a strain of 0.02 is:

$$V_1 = V^+\left(\frac{R_2}{R+\Delta R+R_2} - \frac{R_3}{R_1+R_3}\right) = V^+\frac{\Delta R}{2(2R+\Delta R)} = 49\ mV$$

A differential amplifier is used to amplify the signal V_1. We chose resistors so that the amplification is 100 (R_5 = R_7 =10 $k\Omega$ and R_4 = R_6 = 100 Ω). Therefore, the maximum voltage at the output of the differential amplifier is about 4.9 V.

5.6.3.2. Analog circuit for stop breathing detection

Detection of stop breathing is based on an integrator circuit. When the amplitude of the breathing signal is small, the capacitor C_1 is being charged. During charging, the amplitude at the output of the integrator increases. If the amplitude is greater than some predefined threshold, this means that the alarm needs to be generated. If the amplitude of the breathing signal is large, the integrator will be reset to zero and will not be charged. This is the basic idea behind the analog processing performed to detect stop breathing that takes more than 10 seconds. Please note that this is a very simplified implementation and is susceptible to motion artifacts. Implementation details are presented next.

The lowpass filter is used to filter out random noise. Components C_f = 500 nF and R_8 =100 kΩ are selected to achieve $f_c = 1/(2\pi R_8 C_f)$ =3.2 Hz, which is satisfactory for the breathing signal. The pulse detector circuit converts amplified and filtered breathing signals into pulses shown in Fig. 5.17a). There, the signal at the output of the low-pass filter presented using a blue line in Fig. 5.17a) is compared to a threshold that is set to 0.5 V using the comparator Comp 1. When the output of the comparator is one, the switch is closed, and the capacitor C_1 is connected to the ground through this voltage controlled switch and discharged. The capacitor is charged during the periods of logic zero at the output of the comparator Comp 1, which correspond to the low breathing amplitudes or no breathing signal. The time constant is selected to be 10 sec ($C_1 = 1\ \mu F$ and $R_{10} = 10\ M\Omega$).

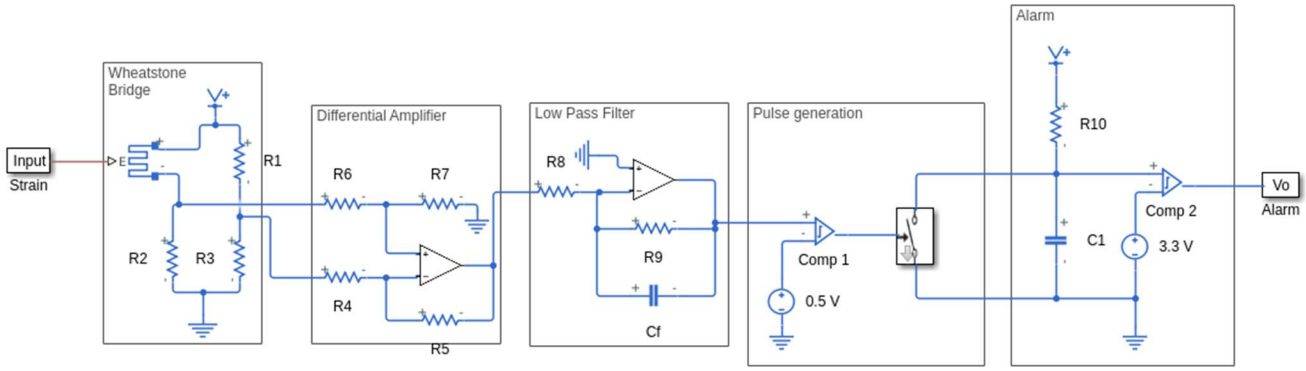
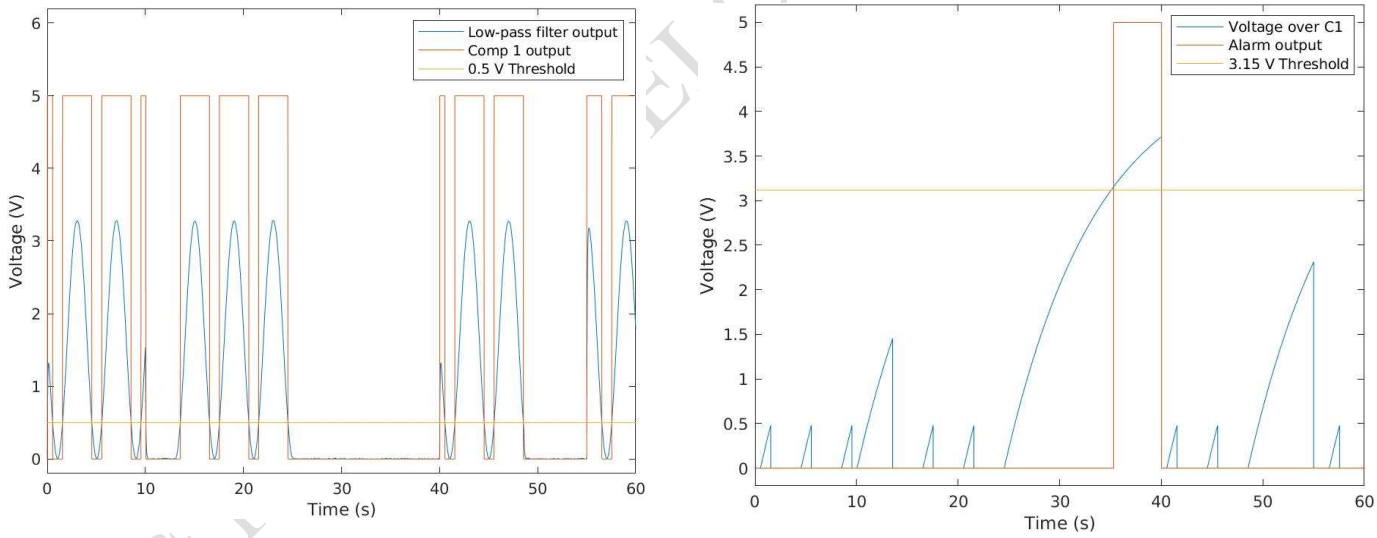


Fig. 5.16 Stop-breathing detection circuit with analog processing

The output of the RC integrator in the block Alarm is compared against the fixed threshold at 0.63 * 5 V = 3.15 V using the comparator Comp 2. The alarm is generated when the signal is above the threshold. This threshold is selected like that because the voltage at the output of the RC circuit will reach 0.63 of its maximum at the time that corresponds to one time constant (see Section 4.4.1). In Fig. 5.17b), the blue line shows the voltage level at the capacitor C_1. When the signal at the output of the low-pass filter is smaller than 0.5 V, the capacitor C_1 will start charging. However, it will not reach the threshold at 3.15 V shown as a yellow line in Fig. 5.17b) unless the signal amplitudes are lower than 0.5 V for more than 10 s. This occurs only about 35 s from the beginning. At that time, the output is one, and it remains one for the rest of the breathing cessation. This is presented as a red signal in Fig. 5.17b).



a) b)

Fig. 5.17 Waveforms for the circuit presented in Fig. 5.16 for the input signal presented in Fig. 5.15. a) Blue: output of the low-pass filter, Red: Output of Comparator 1, Yellow: the threshold at 0.5 V. b) Blue: The voltage over the capacitor C_1, Red: output alarm signal, Yellow: the threshold at 3.15V.

5.6.3.3. Software-based stop breathing detection circuit

The design is shown in Fig. 5.18. The major difference is that analog electronics for detecting stop breathing are replaced with an antialiasing filter, A/D converter and processor. The processor is modeled in Matlab code. It processes data in real-time and outputs 1 V if the breathing signal is below a certain threshold for more than 10 sec. The A/D converter is selected to have 12 bits and a sampling rate of 80 Hz.

The antialiasing filter is designed to perform low-pass filtering of the signal at the output of the differential amplifier. The components were selected as $C_{11} = C_{21} = 800\ nF$ and $R_{11} = R_{21} = 10\ k\Omega$ resulting in a cut-off frequency of 19.9 Hz.

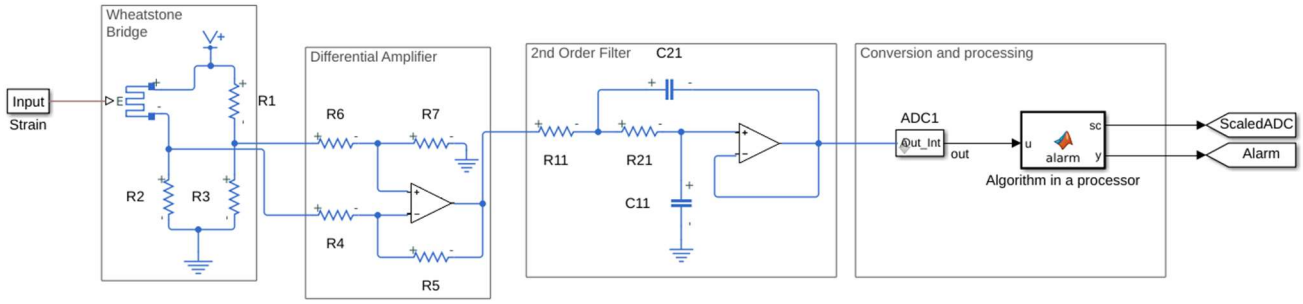


Fig. 5.18 Stop breathing detection circuit where the algorithm for detecting stop breathing is implemented in software.

The algorithm performs the following steps:

- Stores the samples that arrive from the A/D converter sequentially in a memory (modeled as an array)
- Scales data to the range from $0\ V$ to $5\ V$ because the data from the A/D converter is in the range between 0 and $2^{12} - 1$
- Compares the last 10 seconds of data against a fixed threshold of 0.5 V and, if the output is smaller than the threshold, generates the alarm.

Timing diagrams are shown in Fig. 5.19. The blue line represents the scaled output of the A/D converter. The red line represents the alarm generated 10 seconds after the signal was below the threshold of 0.5 V.

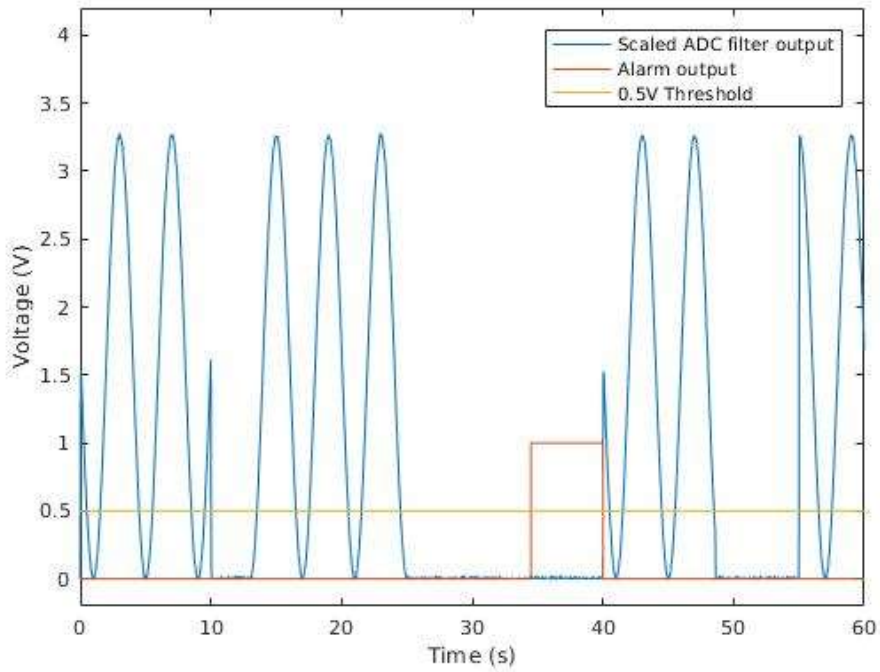


Fig. 5.19 Timing diagram of a stop-breathing detection circuit shown in Fig. 5.18. The signals shown include: Blue: scaled output of the A/D converter, Red: output alarm signal, Yellow: the threshold at 0.5 V.

5.7. Modeling power consumption

Power is defined as $P = V \cdot I$ where V is the voltage, and I is the current through the component. The energy that the component or a device requires to operate is

$$E = \int_{t=0}^{T} P dt$$

The power in a digital circuit is defined as a sum of dynamic and static power. Static power is the power that a device consumes independent of any activity or task the core is running because even in an inactive state, there is a low "leakage" current path and some static current. Static current is the DC bias current that some circuits need for their correct work. Dynamic power is the power due to switching transistors. It is the power of the active digital components. It is defined as

$$P = A \cdot C \cdot V_{dd}^2 \cdot f_{clk}$$

where A is the switching factor which is the probability of switching from 0 to 1 or from 1 to 0 in a given clock cycle, C is the load, f_{clk} is the clock frequency and V_{dd} is the supply voltage. Therefore, to reduce the dynamic power, one should reduce the supply voltage and the clock frequency. Normally, the clock frequency and the supply voltage can be reduced together with significant power savings. Power savings are normally made by using devices with multiple power modes in which there is a tradeoff between the power consumption and the operating speed or accuracy. The modes range from normal operating modes, where devices operate at full speed, to different power down, standby and sleep modes depending on the device. The power is computed as the average energy during active and standby modes divided by the total time: $P = (P_{active}T_{active} + P_{standby}T_{standby})/T_{total}$. The initial estimate of the power in different modes can be obtained from a datasheet of the component of interest.

5.7.1. An example of estimating power consumption of an A/D converter

In Chapter 4, we introduced several types of A/D converters. Sigma delta A/D converters perform internally a large number of operations which might be a reason for higher power consumption. Successive approximation A/D converters are typically low power. Low-power A/D converters power down automatically at the end of each conversion phase. In this way, the power consumption is proportional to the sampling rate as we will show in the following example.

Example 1 Let us consider the power consumption of a 16-bit, 100,000 samples per second (sps) successive approximation A/D converter AD7684 from Analog Devices. The parameters of the converter are given in Table 5.4. Compute power consumption as the sampling frequency increases from 100 sps to 100 ksps.

Table 5.4 Operating parameters of A/D converter AD7684 from Analog Devices

Conversion time, Tc	Supply Voltage, Vdd	Operating current at 100 ksps, Io	Standby current, Is
10 µs	2.7 V	0.56 mA	1 nA
	5 V	0.8 mA	1 nA

Solution:

Since the A/D converter switches between the operating and standby modes, the energy E_{adc} required to perform one conversion during one sampling period T_s is:

$$E_{adc} = V_{dd}\,(I_o \cdot T_c + I_s \cdot (T_s - T_c))$$

The power is then $P_{adc} = E_{adc}/T_s$. The plot of power over the range of sampling frequencies is shown in Fig. 5.20.

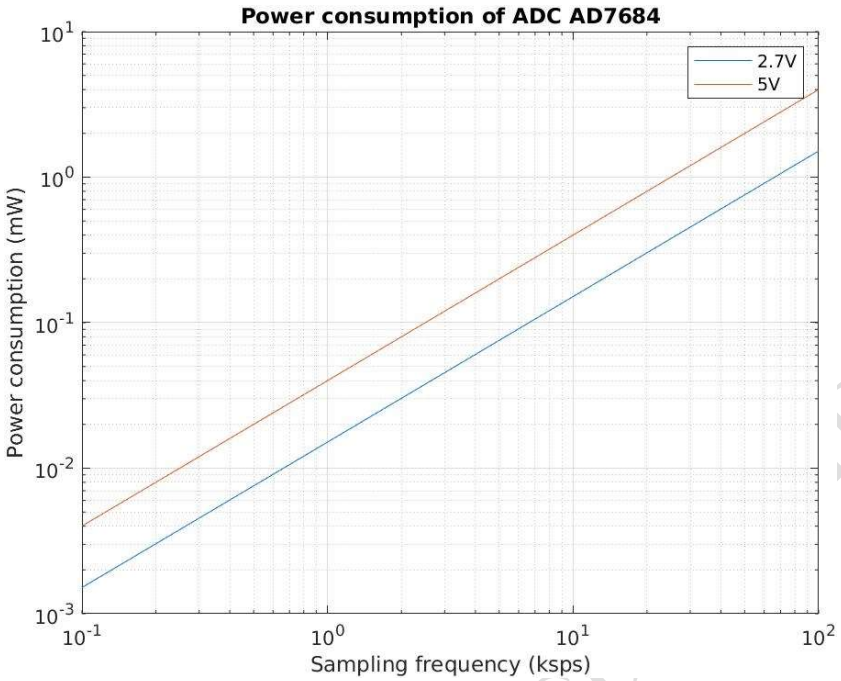


Fig. 5.20 Power consumption for different sampling frequencies and supply voltages of a successive approximation A/D converter.

5.8. Summary

In this chapter, we introduced several important concepts related to modeling and simulation, including how to:

1. Collect, obtain or generate data
2. Fit the data to the model
3. Model the noise
4. Estimate the quality of the signal
5. Propagate the uncertainties through a circuit
6. Analyze streaming data by performing block or stream processing
7. Modeling power estimation.

The following applications were simulated in detail while describing the concepts, including:

1. Windkessel model of the aortic pulse
2. Signal quality analysis of ECG data
3. Uncertainty is propagated through the circuit that converts the signal from the piezoelectric transducer into the voltage to measure acceleration
4. Real-time implementation of the stop-breathing alarm system
5. Power consumption of an A/D converter.

The reader is encouraged to go through the simulations and modify the parameters of the programs to better grasp the concepts. After completing this chapter, the reader should be able to model a biomedical system and its components, including the noise, calculate the signal quality and perform uncertainty propagation.

5.9. Problems

5.9.1. Simple questions

5.1. *Discuss when white, pink, and brown noises appear in biomedical instrumentation.
5.2. *Give examples of low, middle and high-frequency noises in biomedical instrumentation.
5.3. Define the differences between noise, interference, and motion artifacts.
5.4. Define concepts related to signal quality analysis, including factors, features, indices and SQI.
5.5. List advantages of implementation where processing is implemented in software vs. implementation in which processing is implemented in analog electronics.

5.9.2. Problems

5.6. Perform uncertainty analysis in Section 5.5.1 using the derivative approach.
5.7. Explain time and frequency representation of different motion artifacts shown in Fig. 5.7, Fig. 5.8 and Fig. 5.9.

5.9.3. Simulation

5.8. This question is related to the Windkessel model shown in Section 5.3.2.1.

a) Perform sensitivity analysis of the circuit parameters of the Windkessel model shown in Fig. 5.3 by modifying the code provided on the book webpage to include a positive correlation between the peripheral and aortic resistance when generating random numbers and a negative correlation between the resistance and compliance. Discuss the difference in the results – if any.

b) Modify the cost function in the Windkessel model to be the mean absolute error. Comment on the results.

5.9. This question is related to the quality of the ECG signal described in Section 5.4.3.1.

a) Add baseline shift to the ECG signal. Analyze the effect the baseline shift has on the three SQIs. Then filter out the baseline shift and observe the three SQIs again.

b) Change SNR for Case 2 of the signal quality example shown in Section 5.4.3.1 and show the effect of different SNRs on all three SQIs.

5.10. Section 5.5.1 shows uncertainty propagation through a charge amplifier. Perform sensitivity analysis and determine what components affect the charge amplifier the most. Select the tolerance of these components to be 0.1% and then perform uncertainty propagation and determine the confidence intervals at the output.

5.11. This question relates to the software-based stop-breathing detection circuit described in Section 5.6.3.3.

a) In order to simulate the effect of aliasing, add 60 Hz interference to the input signal shown in Fig. 5.18. Observe the effects of aliasing at the output. How much is the 60 Hz signal attenuated at the output of the antialiasing filter?

b) Run the algorithm that averages the data during the 10 s interval instead of just comparing all data samples against the threshold. Then, compare the generated alarm signal at the output against the one presented in Fig. 5.19. Why is there a delay in processing?

c) Add motion artifact of 1 s duration. The artifact should result in a signal amplitude being larger than 1 V. Make sure that the motion artifact occurs during the 15 s stop-breathing intervals. Compare the outputs of the algorithms with and without averaging.

d) Modify both analog and software-based implementation by adding a push button that will be used to stop the alarm. In this case, the alarm will be stopped only when someone presses the button. This is important to ensure that someone heard and reacted to the alarm.

FURTHER READING

In this section, we refer to several relevant resources for further reading. However, please note that this list is not comprehensive and that many excellent and relevant resources might have been omitted unintentionally.

In general, modeling physiological control systems and physical systems in Simulink is covered very well in [Khoo18]. Another excellent book that discusses modeling physiological systems using Matlab and Comsol, mainly based on the physical and finite element model, is [Dokos17].

Modeling of biomedical systems and model-based signal processing is presented in [Cerutti11].

An excellent description of noise sources in the electric circuits and computation of the sensitivity of the output of the electric circuit to different noise sources is presented in Chapter 9 of [Northrop12].

Uncertainty propagation for sensors and measurement systems is covered well in [Figliola12]. Excellent theoretical discussion about uncertainty quantification is covered in [Smith14]. There are also chapters on sensitivity analysis and uncertainty propagation. One of the main references in the field of global sensitivity analysis is [Saltelli08].

An excellent reference for real-time implementation of signal processing algorithms is [Ackenhusen99].

References

[AD17] 12-Bit, 1 MSPS, Single-Supply, Two-Chip Data Acquisition System for Piezoelectric Sensors, Analog Devices, Circuit Note CN-0350, 2017.

[Aew16] J. Aew, T.J. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi and R.G. Mark, "MIMIC-III, a freely accessible critical care database," Scientific Data, 2016, DOI: 10.1038/sdata.2016.35.

[Ackenhusen99] J. G. Ackenhusen, *Real-time Signal Processing: Design and Implementation of Signal Processing Systems*, Prentice-Hall, 1999.

[Burke16] G. Burke, J. Lima, N.D. Wong, J. Narula, "The multiethnic study of atherosclerosis," Global Heart, vol. 11, no. 3, pp. 267–268, 2016.

[Charlton19] P.H. Charlton et al., "Modeling arterial pulse waves in healthy ageing: a database for in silico evaluation of haemodynamics and pulse wave indices," AJP Hear. Circ. Physiol., 2019.

[Cheng10] L. Cheng, O. Ivanova, H.H. Fan and M.C. Khoo, "An integrative model of respiratory and cardiovascular control in sleep-disordered breathing. Respir Physiol Neurobiol. vol. 174, no. 1-2, pp. 4-28, 2010.

[Clifford06] G. D. Clifford, F. Azuaje, P. E. McSharry, *Advanced Methods and Tools for ECG Analysis*, Artech House Publishing, Boston/London; 2006.

[Cerutti11] S. Cerutti, C. Marchesi, *Advanced methods of biomedical signal processing*, Wiley, 2011.

[Dokos17] S. Dokos, *Modeling Organs, Tissues, Cells and Devices using MATLAB and COMSOL Multiphysics,* Springer 2017.

[Elliott14] P. Elliott, "The Airwave Health Monitoring Study of police officers and staff in Great Britain: Rationale, design and methods," Environmental Research, vol. 134, pp. 280-285, 2014.

[Ewing85] D.J. Ewing, C.N. Martyn, R.J. Young, and B.F. Clarke, "The value of cardiovascular autonomic function tests: 10 years experience in diabetes.," Diabetes Care, vol. 8, no. 5, pp. 491–498, 1985.

[Farago21] E. Farago, A.D.C. Chan, "Motion artifact synthesis for research in biomedical signal quality analysis," Biomedical Signal Processing and Control, vol. 68, no. 102611, 2021.

[Khoo18] M. C.K. Khoo, *Physiological control systems: Analysis, Simulation, and Estimation*, Wiley, Second Edition, 2018.

[Kind10] T. Kind, T.J.C. Faes, J.W. Lankhaar, A. Vonk-Noordegraaf, M. Verhaegen, "Estimation of three and four-element windkessel parameters using subspace model identification," IEEE Trans Biomed Eng., vol. 57, pp. 1531-1538, 2010.

[Li09] Q. Li, R.G. Mark & G.D. Clifford, Artificial arterial blood pressure artifact models and an evaluation of a robust blood pressure and heart rate estimator, BioMedical Engineering Online, Vol. 8, no. 13, 2009, https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-8-13, Last accessed on November 15, 2020.

[Li08] Q. Li, R.G. Mark, G.D. Clifford, "Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter," Physiological Measurement, vol. 29, no. 1, pp. 15–32, 2008.

[Löllgen09] Löllgen D, Müeck-Weymann M, Beise RD. The deep breathing test: median-based expiration-inspiration difference is the measure of choice. Muscle Nerve, vol. 39, no. 4, pp. 536-544, 2009.

[Mitra01] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 2nd dd. New York: McGraw-Hill, 2001.

[Northrop12] R. B. Northrop, *Analysis and Application of Analog Electronic Circuits to Biomedical Instrumentation*, CRC Press, Second edition, 2012.

[Orphanidou18] C. Orphanidou, *Signal Quality Assessment in Physiological Monitoring: State of the Art and Practical Considerations*, SpringerBriefs in Bioengineering, 2018.

[Parker13] K.H. Parker, A practical guide to wave intensity analysis, https://kparker.bg-research.cc.ic.ac.uk/guide_to_wia/00_introduction.html, last updated on August 9, 2013, accessed on June 20, 2021.

[Salvi17] P. Salvi, *Pulse Waves: How Vascular Hemodynamics Affects Blood Pressure*, Springer, 2nd edition, 2017.

[Sameni08] R. Sameni, M. Shamsollahi, C. Jutten, "Model-based Bayesian filtering of cardiac contaminants from biomedical recordings," Physiological Measurement, IOP Publishing, vol. 29, no. 5, pp. 595-613, 2008.

[Sedghamiz15] H. Sedghamiz and D. Santonocito, "Unsupervised Detection and Classification of Motor Unit Action Potentials in Intramuscular Electromyography Signals, The 5th IEEE International Conference on E-Health and Bioengineering, EHB 2015, At Iasi-Romania.

[Spallone11] V. Spallone, F. Bellavere, L. Scionti, et al., "Recommendations for the use of cardiovascular tests in diagnosing diabetic autonomic neuropathy," Nutr Metab Cardiovasc Dis, vol 21, pp. 69–78, 2011.

[Spencer87] M.P. Spencer, “Normal blood flow in the arteries,” in *Ultrasonic Diagnosis of Cerebrovascular Disease. Developments in Cardiovascular Medicine*, M.P. Spencer (eds), vol 61. Springer, 1987.

[Tang20] Q. Tang, Z. Chen, R. Ward, R. et al. “Synthetic photoplethysmogram generation using two Gaussian functions,” Scientific Reports, Nature, vol. 10, no. 13883, 2020.

[Thakkar04] P. Thakkar, “The Removal of Motion Artifacts from Non-invasive Blood Pressure Measurements,” Ph.D. thesis, University of Central Florida, 2004.

[TCPS2_18] Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans, TCPS2 2018, Canadian Institutes of Health Research, Natural Sciences and Engineering Research Council of Canada, Social Sciences and Humanities Research Council.

[Willemet15] M. Willemet, P. Chowienczyk and J. Alastruey. “A database of virtual healthy subjects to assess the accuracy of foot-to-foot pulse wave velocities for estimation of aortic stiffness,” American Journal of Physiology: Heart and Circulatory Physiology, vol. 309, pp. H663-675, 2015.

[Zhou19] S. Zhou, X. Lisheng, L. Hao, et al., “A review on low-dimensional physics-based models of systemic arteries: application to estimation of central aortic pressure,” BioMed. Eng. OnLine vol. 18, no. 41, 2019.