CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF BIOMEDICAL ENGINEERING

Department of Biomedical Technology

Detection of artifacts in arterial blood pressure and intracranial pressure signals

Doctoral Thesis

Ing. Valeriia Trukhan

Kladno, *september 2024*

Ph.D. Programme: *Biomedical Engineering*

Supervisor: *doc. Ing. Martin Rožánek, Ph.D.*

Supervisor–specialist: *MUDr. Josef Škola, Ph.D., EDIC.*

Kladno 2024

Declaration

I hereby declare that I have written this thesis myself as a result of my original research (or as the co-author of research papers). All sources of information have been properly stated and referenced.

In Kladno, 6.9.2024 ……..

Ing. Valeriia Trukhan

Acknowledgements

First of all, I want to thank my supervisor, doc. Ing. Martin Rožánek, Ph.D, for his invaluable guidance, encouragement, and support. I would also like to thank, MUDr. Lenka Horáková, Ph.D., DESAIC and MUDr. Josef Škola, Ph.D., EDIC., for the teamwork, support and helpful discussions, which made this process much easier and more interesting.

And finally, I am thankful to my family and friends who have always supported and believed in me, even when I doubted myself. Your constant support, love, and encouragement have been my greatest source of strength, and I am incredibly lucky to have you by my side on this journey. I could not do that without you.

Abstract

In neurocritical care, particularly in traumatic brain injury (TBI) cases, one of the primary goals is to ensure sufficient cerebral blood flow to prevent secondary brain injury. A key factor in achieving this is maintaining cerebral perfusion pressure (CPP) within an optimal range. The pressure reactivity index (PRx), calculated as a moving-window Pearson's correlation between mean arterial pressure (MAP) and intracranial pressure (ICP), is commonly used to determine this optimal CPP. However, false-positive PRx values can occur due to artifacts in arterial blood pressure (ABP) or ICP signals, potentially leading to incorrect interpretation of cerebral autoregulation state.

In this thesis, I developed an artifact detection algorithm based on short-time Fourier transform (STFT). The algorithm was first tested on simulated stereotypical artifacts in ABP signals of various shapes and durations, achieving sensitivity and specificity rates above 93%. It was then validated on real ABP data with annotated artifacts, where it achieved a sensitivity of 92% and a specificity of 90%.

As a result, I created a Python plug-in for the ICM+ software that integrates the developed detection algorithm and calculates a PRx reliability index. The PRx reliability index indicates the percentage of artifacts within a given time interval, allowing for an assessment of the quality of the PRx value during that period. This plug-in can be used for both offline and real-time ABP analysis, potentially improving the accuracy and interpretation of PRx indices.

Keywords

artifacts, detection, arterial blood pressure, pressure reactivity index, short-time Fourier transform, signal processing, plug-in

Contents

List of Abbreviations

Abbreviation	Meaning
TBI	Traumatic Brain Injury
CPP	Cerebral Perfusion Pressure
PRx	Pressure Reactivity Index
MAP	Mean Arterial Pressure
ICP	Intracranial Pressure
ABP	Arterial Blood Pressure
STFT	Short-Time Fourier Transform
EEG	Electroencephalogram
PPG	Photoplethysmogram
CSF	Cerebrospinal Fluid
CBF	Cerebral Blood Flow
CPPopt	Optimal Cerebral Perfusion Pressure
ECG	Electrocardiogram
SAI	Signal Abnormality Index
EDSS	End-Diastole Slope Sum
SESS	Slow Ejection Slope Sum
ICU	Intensive Care Unit
DBN	Deep Belief Network
CNN	Convolutional Neural Network
AUC	Area-Under-the-Curve
EMD	Empirical Mode Decomposition
STFT-ANN	Short-Time Fourier Transform Artificial Neural Network
DWT	Discrete Wavelet Transform
TCD	Transcranial Doppler
NIRS	Near-Infrared Spectroscopy
COx	Cerebral Oximetry Index
nICP	Non-invasive Intracranial Pressure
HDF5	Hierarchical Data Format version 5
FFT	Fast Fourier Transform
MAD	Median Absolute Deviation
STD	Standard Deviation

Figures

1 Introduction

The availability of high-frequency biological signal recordings presents an opportunity for advanced calculations and analyses, offering deeper insights into physiological processes [1]. In neurocritical care, particularly in cases of traumatic brain injury (TBI), one of the key objectives is to ensure adequate cerebral blood flow and prevent secondary brain injury. A crucial aspect of achieving this is maintaining cerebral perfusion pressure (CPP) within an optimal range [2–4]. One method for determining this optimal CPP is through the pressure reactivity index (PRx) [5], which is calculated as a moving-window Pearson's correlation coefficient between mean arterial pressure (MAP) and intracranial pressure (ICP) [6]. Negative PRx values indicate normal cerebral autoregulation, whereas positive values, particularly those greater than 0.3, may suggest a failure in autoregulation [6–9]. However, false-positive values of PRx can arise due to artifacts in arterial blood pressure (ABP) or ICP signals [10].

An artifact is a part of a biosignal that does not have a physiologic origin in the examined organ [11]. Artifacts in both ABP and ICP signals can arise from a variety of sources, including biological and technical factors. Due to the nature of neurocritical care, artifacts in ICP signals are less common, as the ICP catheter, once installed, typically remains undisturbed, which minimizes the occurrence of artifacts. In contrast, artifacts in ABP signals are encountered more frequently. Therefore, this thesis will primarily focus on detecting artifacts in ABP signals.

Several methods have already been developed to detect and remove artifacts from ABP signals, including machine learning techniques, pulse image analysis [12–15], time series analysis [16], and the use of numerical averages (trend data)[13, 17–19]. Although manual detection and removal of artifacts is possible, it is a time-consuming process. Therefore, in this work, we aim to develop an artifact detection algorithm based on the short-time Fourier transform (STFT). To our knowledge, STFT has not been previously applied for artifact detection in ABP signals, though it has been used in studies involving electroencephalogram (EEG) and photoplethysmogram (PPG) signals [20–22]. We selected STFT for its simplicity and efficiency, as it does not require large annotated datasets like machine learning algorithms, and can operate more quickly, making it suitable for real-time data analysis.

2 State of the Art

2.1 Biological signals

A biological signal, or biosignal, is a signal produced by the living organism's existence or caused by physical action on an organism from the outside [23]. From the physical point of view, biosignals do not have to be just electrical signals. We can divide biosignals into several types: bioelectric, biomagnetic, bioacoustic, biochemical, biomechanical, and biooptical [11]. This thesis will focus on biomechanical signals such as arterial blood pressure and intracranial pressure.

2.1.1 Arterial blood pressure

Arterial blood pressure is the force that blood exerts on the artery walls. Changes in blood pressure create a typical waveform (see Figure 2.1). The first slope represents blood ejection from the ventricles, where the upper value is systolic pressure. The second upstroke represents a relaxation of the ventricles, where the lowest point is the diastolic pressure. The dicrotic notch indicates the end of systole and the beginning of diastole [24].

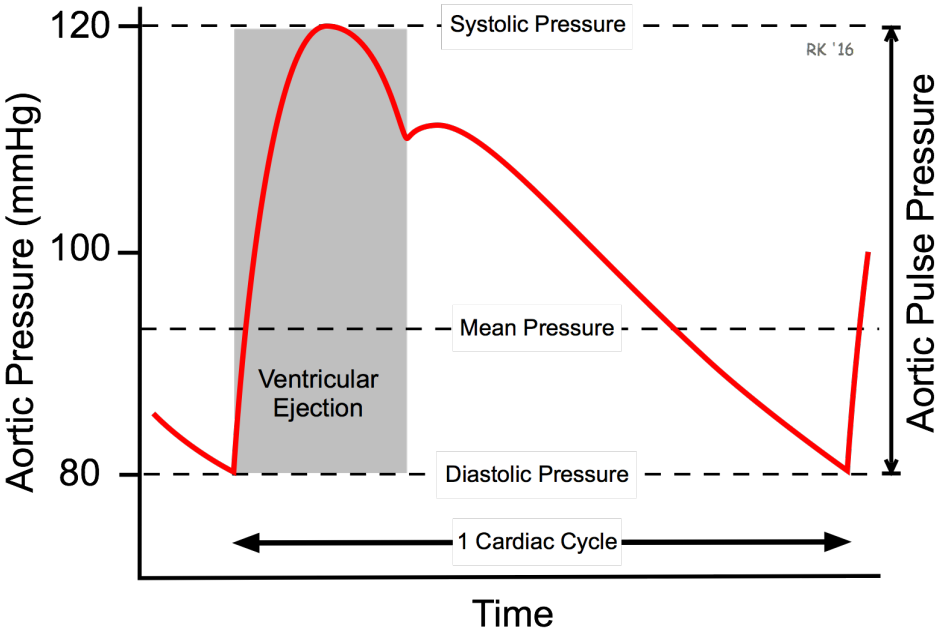


Figure 2.1: Arterial blood pressure waveform [25].

ABP can be measured invasively and noninvasively. During noninvasive blood pressure measurement, a cuff for constriction of the brachial artery and the device that captures Korotkoff sounds (auscultatory method) or oscillation of the cuff volume (oscillatory method) are used. Unfortunately, it is possible to obtain only discrete values from the noninvasive measurement [11].

We use invasive blood pressure measurement for a thorough analysis because it lets us capture continuous waveforms in high-resolution. The measurement is usually

done via a catheter inserted in an artery, with data displayed and captured using a vital sign monitor, see Figure 2.2. The pressure sensors can be outside of the patient's body (extravascular) or inserted directly into the artery at the catheter tip (intravascular). The catheter is a thin plastic tube inside which is usually a few canals. A particular canal can be filled with the fluid that transmits pressure changes from the measurement place to the pressure sensor (transducer). A transducer converts the mechanical impulse of the pressure wave to the electrical signal. Most of the systems include a three-way valve near the transducer, which allows a fast flush of the whole fluid-filled system (frequently used in practice to flush small blood clots at the tip of the catheter). After insertion, the catheter is continuously flushed with a small intravenous fluid volume, preventing coagulation [11]. The fluid flow is negligible, approximately 3-5 ml per hour, because of the catheter's small diameter.

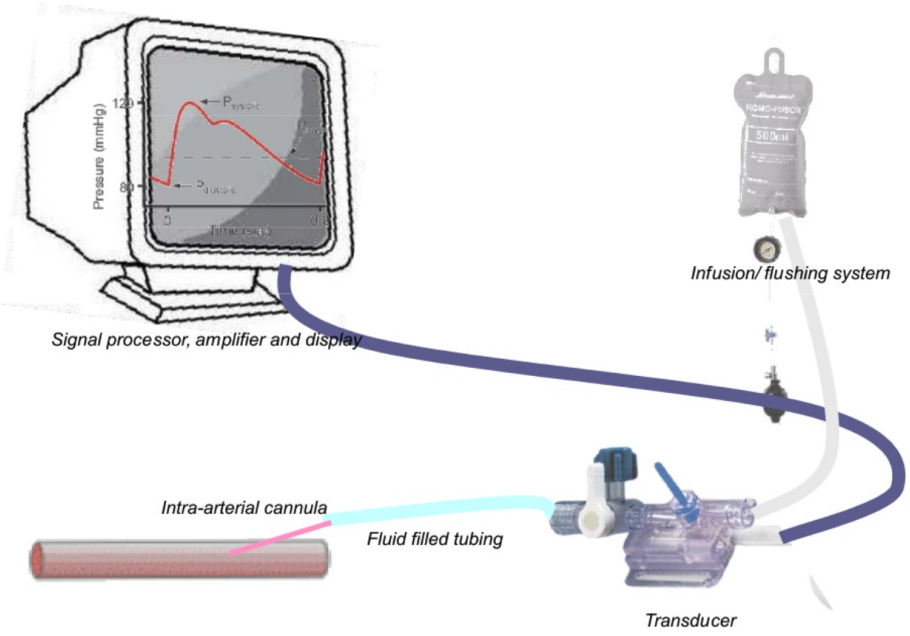


Figure 2.2: Components of the IBP measurement system [26].

For proper measurement of the ABP, the transducer must be at the level of the right atrium. If the transducer's placing is above or below the level, it will lead to the measurement error due to the hydrostatic pressure caused by the fluid in the catheter. For measurement of the ABP for calculation of cerebral perfusion pressure, the transducer must be at the middle cranial fossa level to estimate transcranial perfusion [11, 27].

2.1.2 Intracranial pressure

Monitoring of the intracranial pressure is usually indicated in the case of assumed or emerging intracranial hypertension. In many pathological conditions, value of ICP is increased, which can lead to blood flow disorders in the brain and subsequent neurologic damage. Measurement of the ICP could be performed epidural, subarachnoid, intraventricular, or intraparenchymal (most frequent). The typical intracranial pressure value for adults is 8-15 mmHg, according to Lundberg.

Physiological ICP's waveform has three characteristic waves (see Figure 2.3). Pressure amplitude of the particular waves can change in pathological states. It is essential to be aware that the amplitude of the ICP is increasing proportionally to the mean value of the intracranial pressure [28].

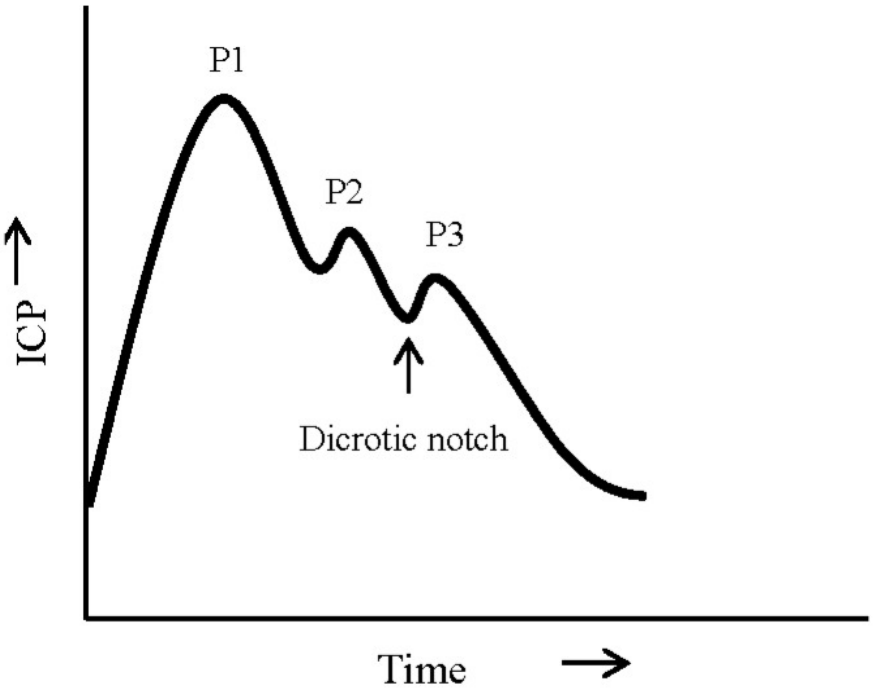


Figure 2.3: An ICP waveform, where P1 is a percussion wave, P2 is a tidal wave, and P3 is the dicrotic wave [29].

The Monro-Kellie doctrine is explained by the direct relationship between volume and pressure within the cranial cavity (see Figure 2.4). That is, the cranial cavity serves as a completely rigid compartment with three incompressible components inside: brain, blood, and cerebrospinal fluid (CSF). If the volume of one component increases, the volume of the other will decrease to have the intracranial pressure remain constant. The compensation can occur through the displacement of CSF into the spinal canal or through increased absorption, as well as by reducing cerebral blood volume through blood vessel constriction. However, if these compensatory mechanisms become overwhelmed—such as when there is a large or rapid increase in one of the components—intracranial pressure rises, which can result in potential brain damage or herniation. This principle is crucial for clinicians to understand how conditions like traumatic brain injury, brain tumors, or hydrocephalus affect intracranial dynamics and pressure [28].

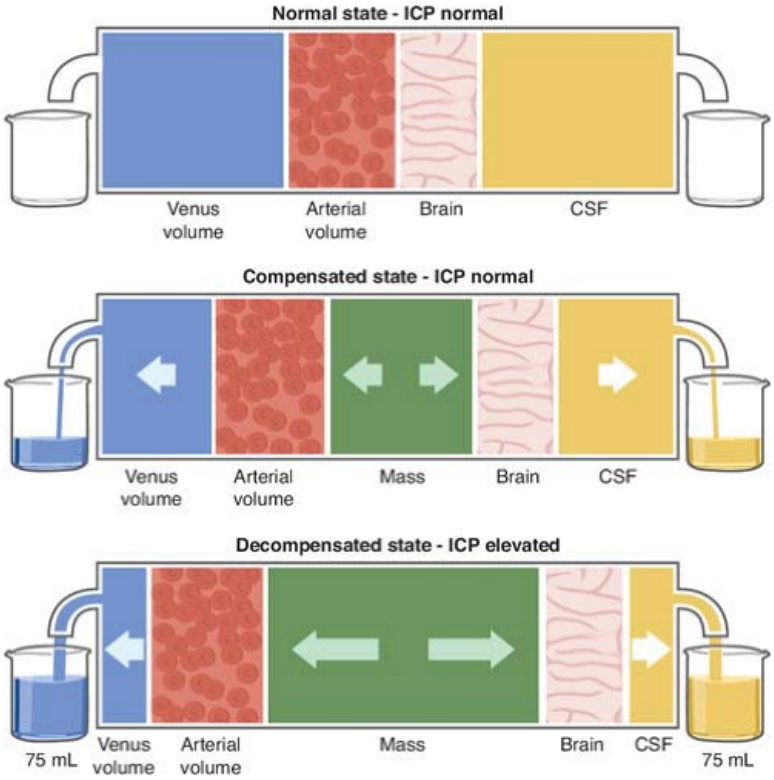


Figure 2.4: Monro-Kellie doctrine [30].

The gold standard of ICP measurement in terms of technology is still the intraventricular catheter with a pressure transducer or in combination with electronic measurement. The intraparenchymal catheter positioning method has the same informative value as the intraventricular, but is less invasive [28].

Nowadays, the intracranial pressure measurement is essential for patients after traumatic brain injury because it helps to preserve proper CPP.

2.1.3 Cerebral perfusion pressure

Cerebral perfusion pressure is defined as the difference between MAP and mean intracranial pressure. MAP is usually automatically calculated from invasive blood pressure. Cerebrovascular pressure autoregulation (Fig. 2.5) protects the brain from changes in CPP by regulating the vascular resistance to ensure stable cerebral blood flow (CBF). Too low values of CPP can lead to ischemia, and too high values lead to hyperaemia. Ensuring an adequate cerebral perfusion pressure helps preventing the secondary injury in patients with TBI [3, 28].

According to the latest TBI guidelines [31], the desired range of CPP should be maintained between 60 and 70 mmHg. However, it should be noted that the ideal perfusion pressure may vary from individual to individual [32, 33]. One of the methods of calculating optimal CPP (CPPopt) is derived by plotting PRx indices against CPP, resulting in a "U" shaped curve, see Figure 2.6. The CPPopt value will be at the point on the curve with the lowest PRx index [3, 5].

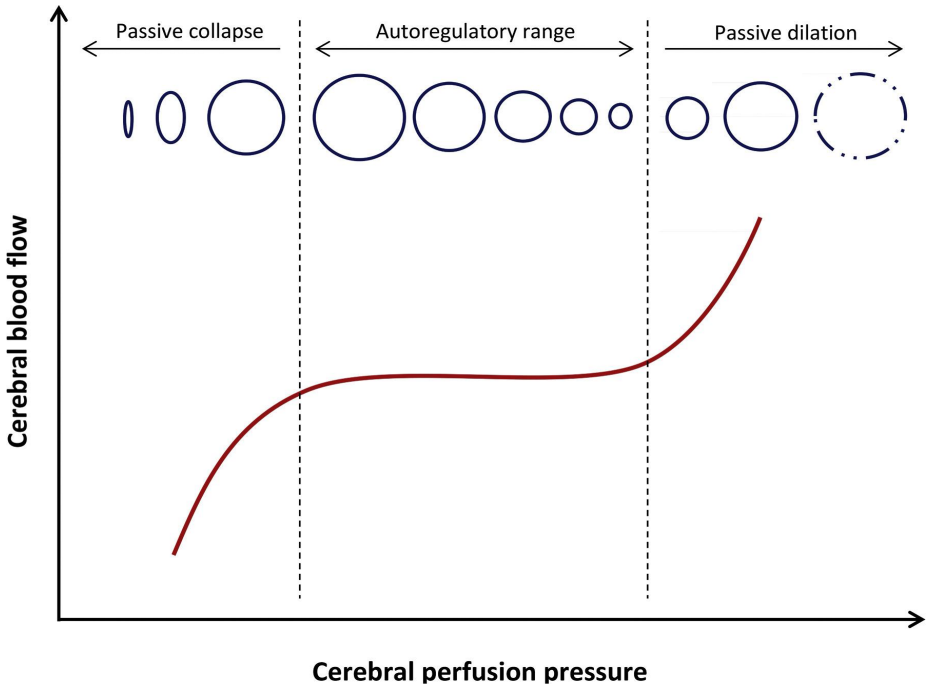


Figure 2.5: Pressure autoregulation curve [34].

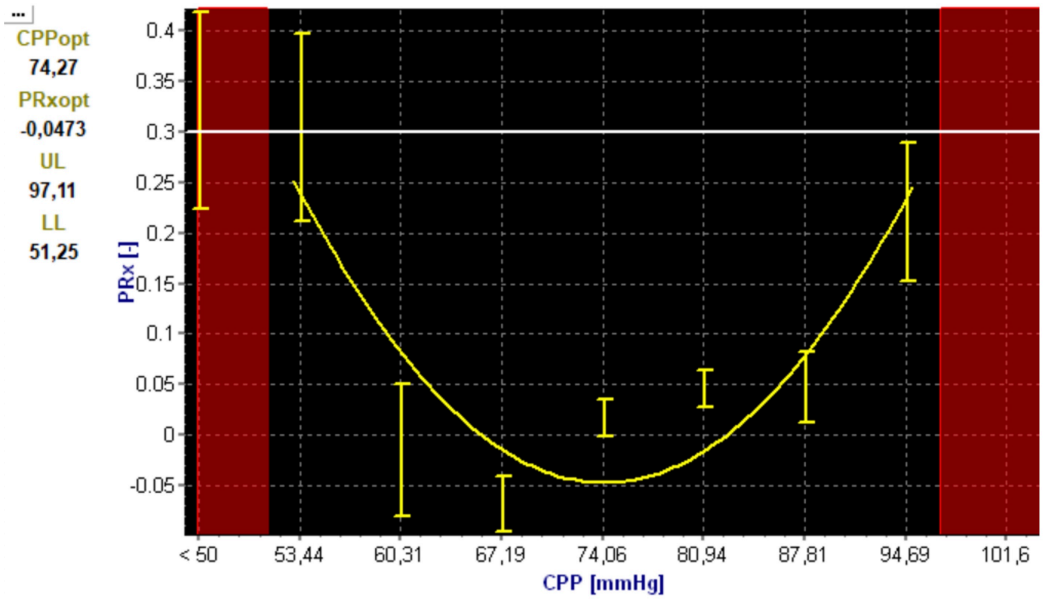


Figure 2.6: Relation between PRx and CPP values [35].

2.1.4 Pressure-reactivity index

The pressure reactivity index is the Pearson correlation coefficient between ICP and MAP. PRx indicates the state of autoregulation of blood vessels of the brain. Negative values of PRx (ICP decrease with MAP increase) represent normal autoregulation, see Figure 2.7. On the contrary, positive values greater than 0.3 indicate a failure of autoregulation [36].

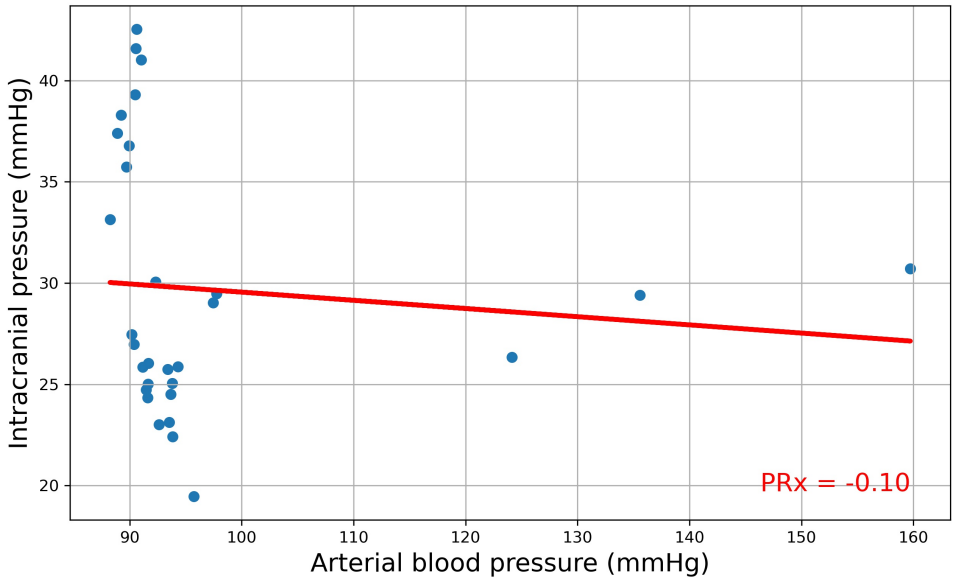


Figure 2.7: PRx curve.

2.2 Artifacts and interference signal

An artifact is a part of a biosignal that does not have a physiologic origin in the examined organ. Artifacts are well-known from clinical practice, but it is often complicated to determine their source and, in some cases, to distinguish them from a useful biological signal. We can divide artifacts into technical and biological. Technical artifacts include electrostatic potentials, electromagnetic interference, power-line noise, impulse interference, and issues with electronic components and circuits. Besides, some examination methods have specific artifacts. Into the biological artifacts, fall motion artifacts and biological factors such as blood clots or thrombosis of the arterial line in the invasive blood pressure measurement [11, 12].

2.2.1 Artifacts in ECG and EEG

Most often, the artifacts are well described and studied in electrocardiographic (ECG) and EEG signals. In recent Littmann's publication was presented a review of electrocardiographic artifacts [37]. This review describes various artifacts, such as motion artifacts, artifacts caused by other devices, loose leads, broken wires, and several other artifacts that simulate clinical conditions [37]. Another publication, by Islam et al., provided a detailed overview of EEG artifacts [38]. EEG artifacts are similar to the ECG artifacts. However, they have some specific additional types, such as eye blinks and eye movements (ocular artifacts), ECG pulses (cardiac artifacts), and different head muscle artifacts [38]. The EEG signal is also more susceptible to electromagnetic interference from neighbouring cables due to the electrodes' close location on the brain map [38].

2.2.2 Artifacts in ABP

Various artifacts can occur during invasive blood pressure measurements, such as air bubbles or blood clots in the fluid-filled tube, the transmission of mechanical vibrations from other devices to the pressure transducer, and constriction of the tube between the catheter and transducer. Additionally, the artifact that frequently occurs in the IBP signal is the transducer flushing [11].

2.2.3 Artifacts in ICP

As for intracranial pressure, there are similar types of artifacts compared to the ABP signal, such as motion artifacts, connection and human errors, and problems in monitoring devices [39]. Moreover, during coughing and sneezing, the physiological value of the ICP may temporarily rise to 60 mmHg, which can be recognised as a false alarm [28].

2.3 Automatic detection and elimination of artifacts in the ABP

In a study by Khan et al., a comprehensive review of the most prominent algorithms for detecting artifacts in ABP signals was presented. Some research has focused on identifying artifacts in numerical values, utilizing averaged ABP and MAP values, while others have analyzed entire waveforms using machine learning techniques and algorithms from image analysis, time series analysis, and signal abnormality detection [17].

Li et al. simultaneously measured ABP with ECG waveforms and recorded ABP artifacts. These researchers analyzed the bedside monitor data obtained from the MIMIC II database, involving more than 6000 hours of monitoring. They were able to identify six different kinds of ABP artifacts: saturation at maximum and minimum ABP levels, pulse pressure reduction, square wave, high frequency, and impulse artifacts. Nevertheless, the concrete origins of those artifacts stay unclear. Additionally, they constructed an algorithm to explore the "artifact pollution" in the ABP signal, and found that diastolic blood pressure is less vulnerable to noise than systolic and mean blood pressure [16].

A different algorithm has been proposed by Zong et al., which makes the assessment of ABP signal quality and examines the ECG-ABP signal relationship in a fuzzy logic approach. Based on the data available in MIMIC database data, they found that their algorithm significantly reduced false ABP alarms due to artifact [40].

Sun et al. had proposed a signal abnormality index (SAI) algorithm for detecting nonstandard segments in ABP waveforms, also using MIMIC II data and checked the algorithm for effectiveness in comparison to human experts. The SAI algorithm has shown robustness and ability to detect typical waveforms of ABP from noise and artifacts for further analysis [41].

However, these methods relied on ECG signals, which might not always be available. In order to overcome this, Zhang et al. proposed a novel methodology for the detection of ABP artifacts in an ECG-independent approach. They proposed two new features - the end-diastole slope sum (EDSS) and the slow ejection slope sum

(SESS) - to refine the SAI. The experiment's results provided evidence that with these two new characteristics, the specificity of the SAI was largely increased [42].

Cao et al. developed an algorithm that rapidly identified nonphysiological artifacts. This study used data from 1852 trauma intensive care unit (ICU) patients admitted to Vanderbilt University Medical Center. They used signal processing statistics to assess the efficiency of this filter and formed logistic regression models both before and after the application of filter to the ABP signal for the prediction of mortality and morbidity [43].

Choi et al. developed an adaptive digital filter system using a capacitive sensor to effectively reduce motion artifacts in blood pressure signals. By synchronizing capacitance data, which varies with motion, with the corrupted blood pressure signal, their system successfully restored the blood pressure signal, achieving up to 95% accuracy in artifact reduction. This method offers a cost-effective alternative to traditional accelerometers, though it has limitations in handling rapid or non-linear movements [44].

Son et al. have developed a deep belief network (DBN) model for the identification and elimination of artifacts in blood pressure waveforms. Artifacts, obtained through the ICM+ software, were classified to have originated from motion, blood clots, thrombosis in the arterial line, cuff inflation, and transducer flushing. Results showed that their developed DBN model outperformed the popular SAI algorithm for artifact detection [12].

In another study, Pasma et al. developed three various learning algorithms for ABP artifact detection: lasso, restrictive logistic regression, neural network, and support vector machines and compared them with the performance produced by manual artifact identification carried out by two trained researchers. They were introducing an artifact-detecting algorithm that would replace the identified artifacts with interpolated values from the blood pressure detected at the arterial line. In a comparison to identify the performance of the algorithm, it was then compared with manual artifact identification [13].

Lee et al. used a deep learning model that combines a stacked convolutional autoencoder and a convolutional neural network (CNN) to effectively remove artifacts from ICP and ABP signals in patients with TBI. Their approach integrates CNN-based automated artifact removal with a data transformation method capable of converting continuous signal data into representative images. The model demonstrated high accuracy, with prediction rates of 97% for ABP and 94% for ICP artifacts, significantly reducing the occurrence of critical clinical events [14].

Finally, Rinehart et al. trained machine-learning algorithms to identify three specific error states: transducer damping, transducer misplacement high, and transducer misplacement low. The results demonstrated that the algorithms achieved high accuracy, with area-under-the-curve (AUC) values exceeding 0.9 for all error types, particularly excelling in detecting damped waveforms [15].

2.4 Automatic detection and elimination of artifacts in the ICP

There have not been done many studies regarding detection and removing artifacts from the ICP signal. Feng et al. proposed an Empirical Mode Decomposition (EMD) method to detect artifacts in the ICP signal. For the elimination of the artifacts

an iterative filtering method was used. The research was performed on signals from 59 neuro ICU patients. In most artifact episodes, this method was effective, apart from recognising artifacts with a small amplitude, right after artifacts with high amplitude [39].

The same research team designed another study to compare three approaches of artifact correction: EMD, wavelet transformation and median filtering. EMD method had the best performance results, but it is more suitable for offline signal analysis, because of the high computational time. For the online analysis, it is more convenient to use the median filter approach [45].

Several methods have been proposed for the automated identification and removal of artifacts from ABP and ICP signals, using techniques such as machine learning, time-series analysis, and signal abnormality detection. While machine learning offers promising solutions, it demands substantial computational power and large, annotated datasets, which makes it less practical for real-time applications where quick processing is crucial.

In contrast, I chose to explore the use of the short-time Fourier transform (STFT) for artifact detection due to its simplicity and efficiency. STFT does not require large annotated datasets and could be faster, making it more suitable for online data analysis in real-time settings.

2.5 Short-time Fourier transform (STFT)

The STFT could be defined as a Fourier transform of a windowed sequence or linear filtering operation. We can obtain a frequency spectrum of the stationary signal in time using this method. Ideal ABP and ICP signals are stationary, i.e. spectral contents are not changing over time. Nevertheless, real signals contain artifacts that make them nonstationary. Hence, STFT can allow us to localize some artifacts in time. The result of the short-time Fourier transform is a matrix of complex numbers, where each complex number represents the magnitude and phase for a particular frequency at a particular time period [46].

2.5.1 STFT for artifact detection

STFT is not usually used for artifact detection. There were just several articles on this topic, and most of them were focused on artifact detection in EEG signals. Yücelbaş et al. presented a methodology to automatically determine the starting and ending time points of sleep spindles in EEG using short-time Fourier transform–artificial neural networks (STFT–ANN), EMD and discrete wavelet transform (DWT) methods. STFT had lower accuracy in comparison with EMD and DWT, because it had low resolution at higher frequencies [20].

Taherisadr et al. designed a study to improve the identification and localization of artifacts in EEG signals. They proposed a method combining three complementary techniques: TF analysis (STFT), multi-resolution analysis, and machine learning. The proposed approach outperformed the standard EEG signal processing method (1D wavelet) when used for artifact detection [21].

Another research team focused on detecting artifacts in photoplethysmography signal, which is closer to arterial pressure frequency-wise. They used deep learning

methods on STFT images as input and analysed quality. The model achieved good results with an accuracy and a sensitivity higher than 98% [22].

2.6 ICM+

ICM+ is a special clinical research software developed by the team from the University of Cambridge (UK) with over 35 years of neuroscience experience. That software facilitates collection of high-resolution data from different analogue and digital devices. Practical applications of the ICM+ can be very diverse: CSF dynamics investigation, monitoring of cerebral autoregulation and other cerebrovascular characteristics, and noninvasive brain monitoring based on Transcranial Doppler (TCD) and near-infrared spectroscopy (NIRS). ICM+ has useful features for signal analysis, like statistical tools and real-time calculations of cerebral autoregulation (PRx, cerebral oximetry index (COx)), CPPopt, brain compliance, cerebrovascular compliance, arterial wall properties, non-invasive ICP (nICP), and complexity of homeostatic regulation (Entropy). Furthermore, that software has artifact elimination tools [47].

2.6.1 Artifact management

There are two approaches to artifact elimination in ICM+: manual and automatic. Firstly, using the manual method, selected specific parts of the signal where the artifacts are located must be selected and then removed. With this removal method, the user can choose from two options. The first one (so-called series) removes the artifact from only one parameter (signal), for example, arterial blood pressure, but all other measured and calculated parameters remain unchanged. The current calculated parameters, based on the already deleted section, will remain unchanged. If we want to prevent this aspect, we have to select the option of so-called global artifact removal and delete the affected part for all signals [47].

The second way to eliminate the artifacts is to define an original formula used to detect artifacts automatically. Software developers provide all detailed instructions for this task. The calculation uses peak-to-peak detection and removes mainly high-frequency artifacts. However, the recognised artifact is not correctly removed, and a noticeable part of the artifact remains in the signal and thus is still included in subsequent calculations. Therefore there is a space for improving artifact detection and elimination in that software [47, 48].

2.6.2 HDF5 files

Hierarchical Data Format version 5 (HDF5), which is used in ICM+ software, is a versatile and efficient data format widely used for storing large and complex data sets in various scientific fields. This format is particularly suitable for the neurocritical care environment, where it solves the problem of archiving heterogeneous data generated by many devices. The hierarchical, self-describing structure of the HDF5 format supports the storage of both small and large datasets and allows for the organized and homogeneous storage of multimodal data such as clinical annotations, low-frequency numerical data, high-frequency waveform data, and summaries of trend data and calculated parameters. Each dataset in the HDF5 file is described using attributes,

making the data self-describing and facilitating interoperability. The ability of the file format to compress the data significantly reduces storage requirements while maintaining fast access times. The compatibility of the HDF5 format with various programming languages and scientific tools, including MATLAB and Python, further enhances its usefulness in neuroinformatics. This flexibility and efficiency make the HDF5 format an ideal choice for creating multi-center databases and standardizing data storage, supporting advanced research and individualized patient management in critical care environments [49].

2.6.3 Plug-in for ICM+

In ICM+, it is possible to use a custom plugin developed in Python software. It allows adding various user-definable functions that can be applied on input signals from vital sign monitors and other devices. The function can contain some complex calculations or statistical analysis that ICM+ does not include. It is possible to employ the designed functions on both offline data and real-time signals [47].

3 Aim of Dissertation

The primary aim of this dissertation is to address the critical issue of artifact detection in arterial blood pressure and intracranial pressure signals, which are crucial to the accurate calculation of the Pressure Reactivity Index. The PRx is a correlation coefficient between MAP and ICP, with positive values above 0.3 indicating a failure of autoregulation. However, the reliability of the PRx index can be compromised by false-positive values resulting from artifacts in ABP or ICP signals. These artifacts, if not properly treated, can lead to misinterpretation of the state of cerebral autoregulation.

Current solutions, such as the ICM+ software, offer features for artifact detection and removal, yet they are not without limitations. Manual artifact removal, while effective, is time-consuming. Automatic artifact elimination is beneficial for addressing small, high-frequency artifacts, but large artifacts, which have a significant impact on MAP and consequently on the PRx index, pose a greater challenge. Therefore, the focus of this research is on developing a robust algorithm specifically designed for the detection and removal of large ABP artifacts, as these have the most substantial effect on the calculation of the PRx index.

The dissertation will prioritize the detection of ABP artifacts, as the ABP signal is more prone to interference compared to ICP. The goal is to identify and eliminate the longest and most significant artifacts that could distort the MAP and, by extension, the PRx index. This selective detection approach is intended to maximize the quality of the remaining signals for accurate PRx calculation, rather than creating a universal detection tool that might excessively reduce the amount of usable data.

The primary focus of this dissertation is on detecting ABP artifacts, with the identification of ICP artifacts serving as a secondary objective. Since ICP signals are generally less prone to interference and a compromised ABP signal renders PRx calculations unusable, the emphasis on ICP artifact detection will be limited and supplementary to the main research focus.

In the subsequent phase of this dissertation, I will develop a specialized algorithm for artifact detection. This algorithm aims to effectively identify and eliminate significant ABP artifacts, thereby enhancing the reliability of PRx calculations and improving the assessment of cerebral autoregulation.

Following the development of the detection algorithm, I plan to create a Python plug-in for the ICM+ software. The output of this plug-in will be a "PRx reliability index," which will inform clinicians of the percentage of artifacts present within a given time interval.

4 Pseudonymisation of the HDF5 files from the ICM+ software

4.1 Methods

When working with data from real patients, it is crucial to perform pseudonymization and remove all personal identifiers to prevent unauthorized re-identification. Before packaging vital signs records into an HDF5 file using the ICM+ software, there is an option to anonymize the data (strip patient identifiers), which we always use. However, during my analysis of the final HDF5 file, I discovered that sensitive data still remains, making it possible to re-identify patients. Examples include actual timestamps, diagnosis notes, and other details. Although the HDF5 file is generated automatically at the end of the recording, it often isn't fully anonymized, and patient identifiers may still be present.

The pseudonymization process involves several steps. The first steps were handled by our intensivist colleague from Masaryk Hospital in Usti nad Labem. These included renaming all files with special identifiers (e.g., TBI_001) and transferring sensitive patient information files, which are automatically generated at the end of the recording, to a secure folder. Only our intensivist colleague has access to files containing personal patient information.

Afterward, I delete the automatically generated HDF5 file because it was created before pseudonymization and may still contain personal patient data. Once these initial steps are completed by my colleague, I can repackage a new HDF5 file from the main .ICMP file. This step is also crucial for another reason: sometimes the automatically generated HDF5 file lacks timeline information, and repackaging the recording helps resolve this issue.

The new HDF5 file is then exported to the Jupyter notebook for timeline pseudonymisation and deleting diagnosis notes. Using Python, I erase sensitive patient information from the annotations folder of the HDF5 file, see Figure 4.1 (on the right). Then I generate a random number from 25 to 100 using *seed* and *randint* functions. This number is then used as a number of years for a time shift, so it is not possible to find out the actual date.

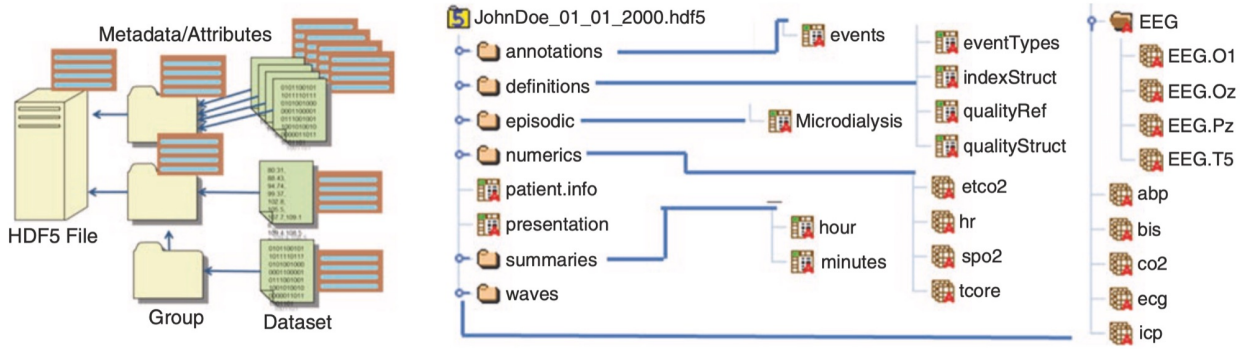


Figure 4.1: On the left is an HDF5 file concept. On the right is a structure of an HDF5 file generated in ICM+ software [49].

To keep the current day of the week after the time shift, I calculate the old and new day of the week using the function *datetime.fromtimestamp().strftime('%w')*. *A* difference between the old and new day of the week is converted to milliseconds and then added to random numbers of years calculated in the previous step.

The last step is adding the random number of years (with a preserved day of the week) to the timeline of trend data, waveform data, and quality data. Information about the timeline is saved in the " index" and "quality" tables of the "waves" and "numerics" data type groups. For pseudonymisation of the timeline, I am rewriting "starttime" column by adding the random number of years (converted to milliseconds) to the original value.

4.2 Results

I created the script that performs pseudonymization for secure future analysis in Python (or other programs like MATLAB). The script takes the original HDF5 file as input and generates a pseudonymized HDF5 file as output. The resulting file has a shifted timeline and no longer contains sensitive patient information.

4.3 Discussion

To protect HDF5 files from unauthorized re-identification, I wrote a pseudonymization script in Python. This script is useful not only for secure signal analysis in Python and other software (e.g., MATLAB) but also for displaying signals in ICM+ software with a pseudonymized timeline and without any sensitive personal information. Although the authors of the ICM+ software published an article about the contents of HDF5 files exported from their program [49], some of the information was inaccurate. For instance, the timeline format in the quality table from the article was incorrect, which complicated the script development process. Most of the HDF5 file details had to be explored and examined manually to ensure there were no mistakes.

5 Effect of downsampling on a signal frequency spectrum

5.1 Methods

While working with ICM+ software and the HDF5 files it generates, I noticed that the software automatically downsamples signals like ABP and ICP when packaging patient data into a single HDF5 file. For example, the ICP signal is downsampled from 200 Hz to 100 Hz. Therefore I decided to examine the effect of downsampling on a signal frequency spectrum.

I first exported a 5-second segment of the ICP signal into a .CSV file. After downsampling it to 100 Hz, I attempted to apply a fast Fourier transform (FFT). However, the FFT could not be performed due to missing data. To fix this, I used interpolation to fill in the data gaps and repeated the downsampling. Before running the FFT, I normalized both the original and downsampled signals. Finally, I compared both FFT spectra by plotting them on a single graph to visualize any differences.

5.2 Results

Below are displayed original ICP signal (see Figure 5.1), and the same signal downsampled to 100 Hz (see Figure 5.2).

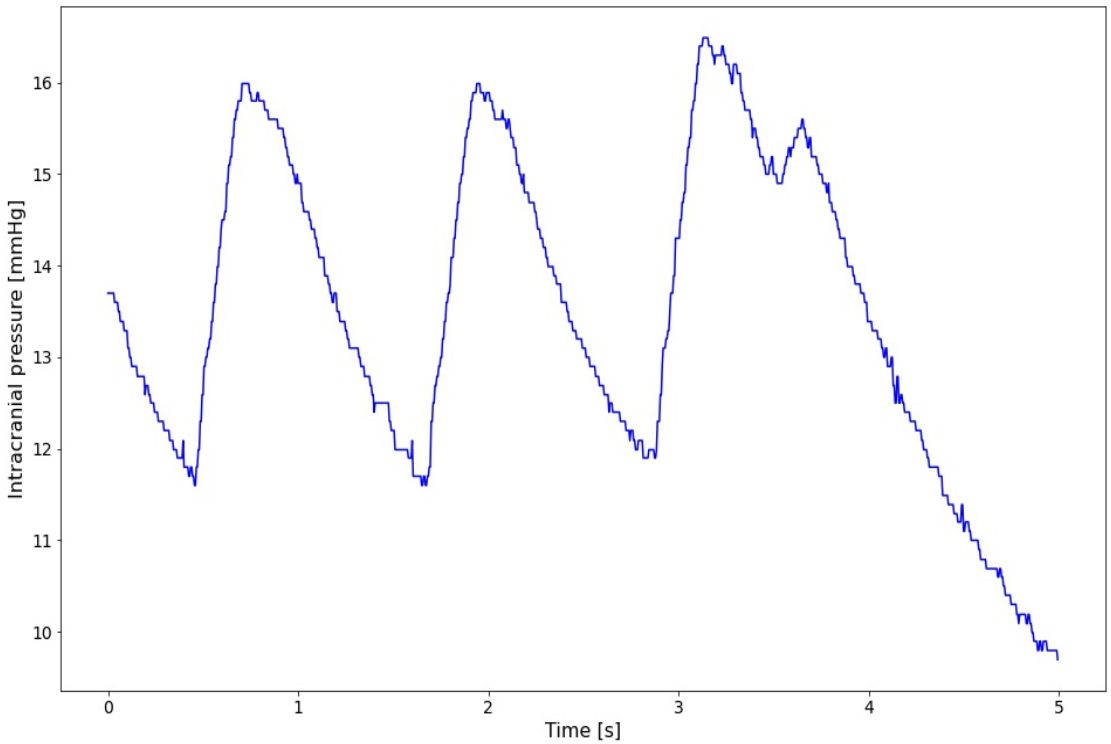


Figure 3.1: Original ICP signal (after imputation) - sampling frequency 200 Hz.

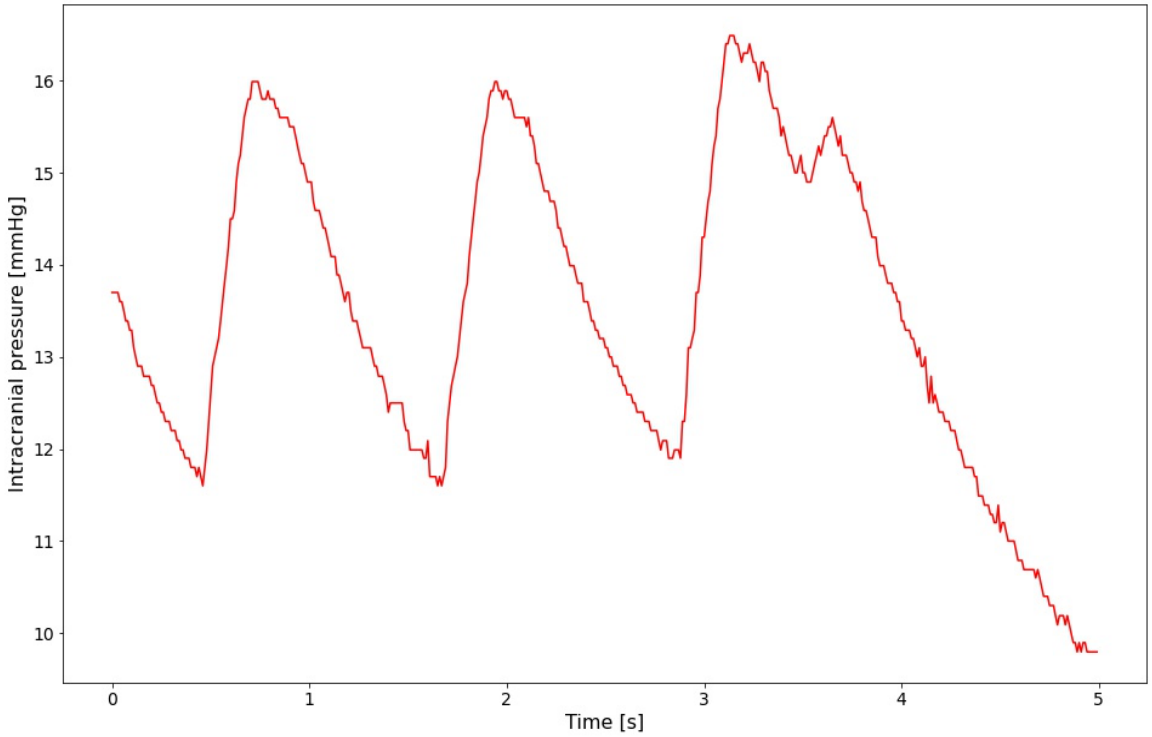


Figure 5.2: Downsampled ICP signal (after imputation) - sampling frequency 100 Hz.

Figures 5.3 and 5.4 display a comparison of the original ICP signal FFT with downsampled signal FFT with different x-axis ranges.

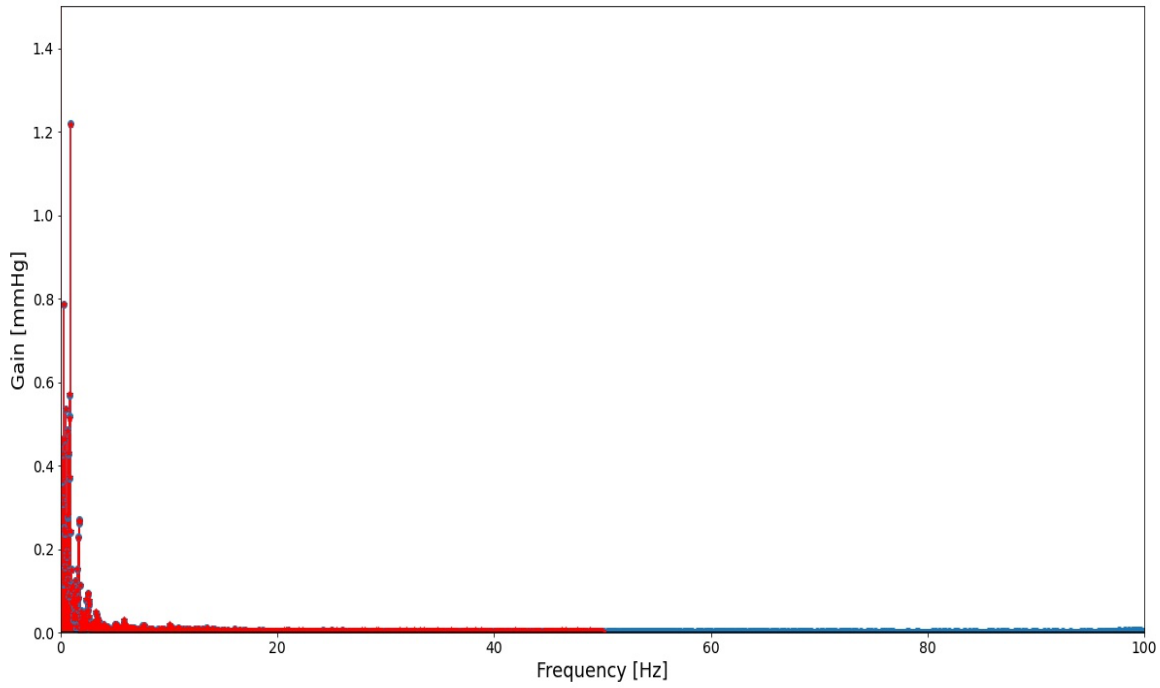


Figure 5.3: Comparison of original signal FFT with downsampled signal (100 Hz) FFT - x-axis range from 0 to 100.

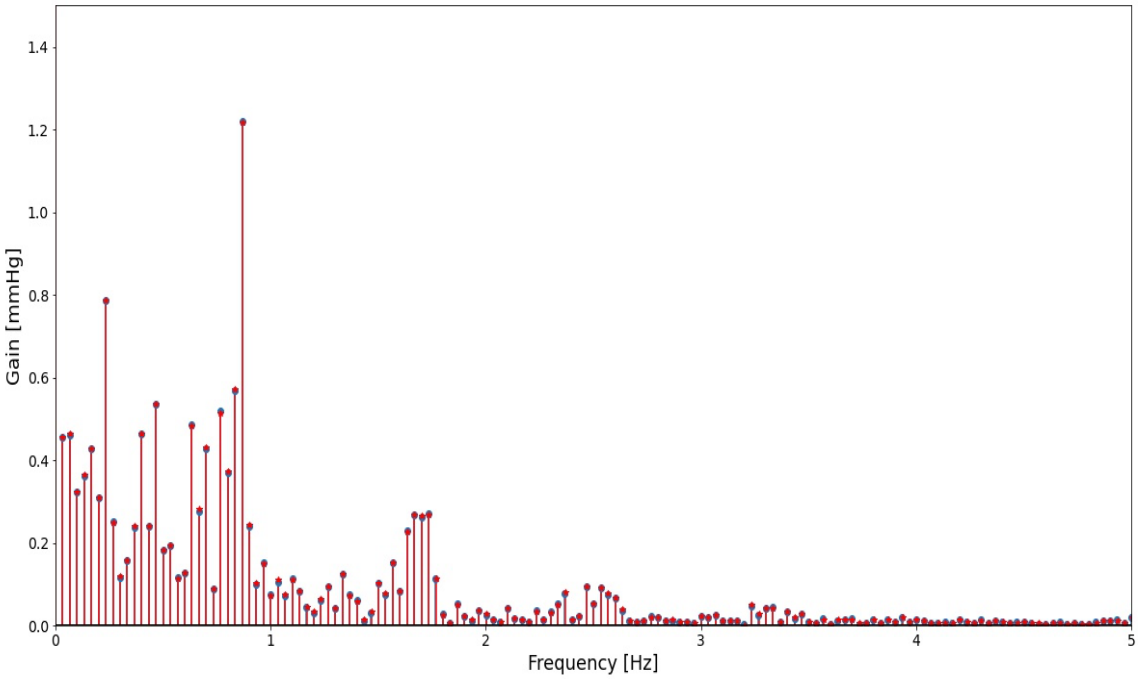


Figure 5.4: Detailed comparison of original signal FFT with downsampled signal (100 Hz) FFT - x-axis range from 0 to 5 Hz.

5.3 Discussion

Since signals exported to HDF5 format in ICM+ software are automatically downsampled, I examined the effect of downsampling on the frequency spectrum. As shown in Chapter 5.2, the appearance of the original signal is very similar to the signal downsampled to 100 Hz, making it difficult to notice any differences. In the first part of the frequency spectrum (from 0 Hz to 5 Hz), both signals are identical. However, beyond 50 Hz, the downsampled signal has no frequency spectrum due to the Nyquist theorem, which states that the sampling frequency must be at least twice the maximum frequency of the original signal. As a result, the downsampled signal's frequency spectrum cannot contain frequencies higher than half of the sampling rate. In our case, 100 Hz is still a sufficient sampling frequency for arterial and intracranial pressure signals. However, if we use these same HDF5 files in the future for signals that require a higher sampling rate (e.g., ECG), we must ensure that the actual sampling rate is appropriate for those signals.

6 Circadian variability of PRx index

6.1 Methods

It is often observed on bedside monitors that increased PRx variability occurs during the morning and mid-morning hours, likely due to artifacts from nursing care. Although this phenomenon has not been described in published studies, it may significantly limit the usefulness of PRx as a real-time parameter for clinical decision-making, including CPPopt calculations. To investigate how artifacts in ABP and ICP signals impact the circadian variability of the PRx index, we initiated a project in collaboration with colleagues from the Neurointensive Care Unit at the Department of Anesthesiology in Masaryk Hospital, Usti nad Labem.

The first step was to select suitable software for signal analysis. While ICM+ software could be used for this purpose, it is not very robust. Therefore, I chose to use the Python programming language and Jupyter notebooks, which provide greater flexibility and capability for our analysis. I also had to decide on the best format for exporting the raw ABP and ICP signals. ICM+ software offers several options, including comma-separated text files (.CSV), tab-delimited relative time text files (.ASC), and hierarchical data format version 5 files (.HDF5). The HDF5 format was the best choice because it allows for the storage of large volumes of numerical data and can be easily accessed using Python [49].

A retrospective signal analysis was conducted on HDF5 data records from 19 patients with traumatic brain injury. The methodology flowchart is shown in Figure 6.1. The HDF5 files were imported into Python using the *h5py.File(HDF5 file, 'r+')* function from the H5PY library. If the file was correctly exported from the ICM+ software, the "waves" group would contain the following datasets: 'art.index', 'art', 'icp.index', and 'icp'. These datasets store high-resolution ABP and ICP values, as well as information about each signal's timeline. Using this timeline information (see Figure 6.2) and the *np.linspace(starttime, stoptime, length)* function, I reconstructed the timelines of the ABP and ICP signals.

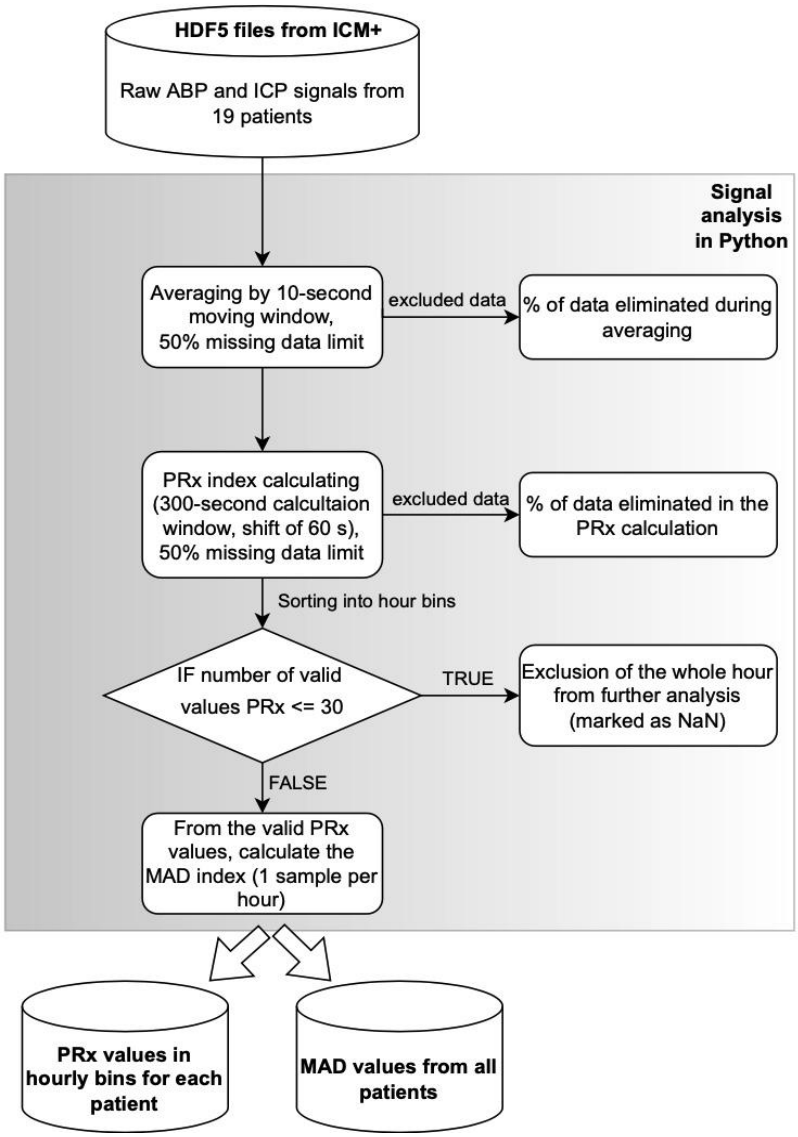


Figure 6.1: Methodology flowchart.

```
array([(b'startidx', b'index of the first sample in this continuous data block'),
       (b'starttime', b'modified UNIX format time stamp (microseconds since 1/1/1970)'),
       (b'length', b'number of data samples in this data block'),
       (b'frequency', b'data sampling frequency [Hz]')],
      dtype=[('field', 'O'), ('description', 'O')])
```

Figure 6.2: Description of columns in the “index” tables.

If an HDF5 file contained manually labeled artifacts, these were recorded in the "quality" tables, which include the timestamps for the start and end of each artifact. Using this information, I labeled all artifacts as NaN values in the ‘art’ and ‘icp’ datasets.

Next, I converted the timelines of the ABP and ICP signals from UNIX format in milliseconds to the standard “yyyy-mm-dd HH:MM.ms” format to simplify further calculations. Since the array containing pressure values and the timeline consists of

billions of rows, performing any analysis would be computationally demanding. To address this, I used the high-performance Python library Vaex, which optimizes calculations and speeds up the analysis process.

After preprocessing the data, I averaged both signals using a 10-second moving window. Before calculating the average, we ensured that the number of valid (non-NaN) values was higher than 50% (the missing data limit). After averaging, we calculated the percentage of eliminated data. Using a similar approach, we calculated Pearson correlation coefficients (PRx indexes), with the only difference being a 300-second calculation window and a 60-second shift. The missing data limit was also set to 50%.

All PRx values were then grouped into hourly bins (from 0 to 23), with the condition that if the number of valid values was less than 30, all values for that hour would be labeled as NaN. The sorted PRx indexes were saved to a .CSV file, and the median absolute deviation (MAD) was calculated for each hourly bin.

After analyzing all 19 patient records, I calculated the median of all MADs for each hourly interval. To better illustrate the variability, I visualized all MAD values using a boxplot.

6.2 Results

PRx indices from 19 patients were sorted into hourly bins and analyzed to assess circadian variability. For each patient, PRx indices and median absolute deviations (MAD) were calculated for each hourly bin. All MAD values were then combined, and a median was calculated for each hour. The graphs below display the calculated medians of the MADs (see Figure 6.3) and a boxplot illustrating the variability of the MADs (see Figure 6.4).

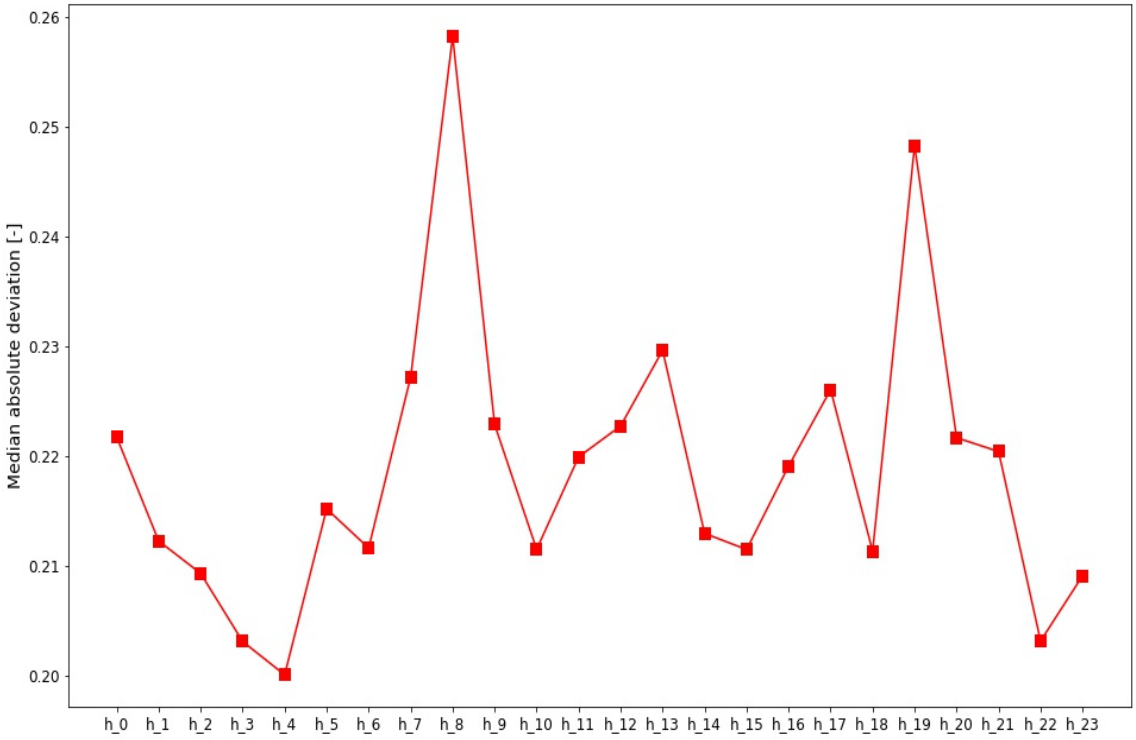


Figure 6.3: Hourly changes of MADs medians.

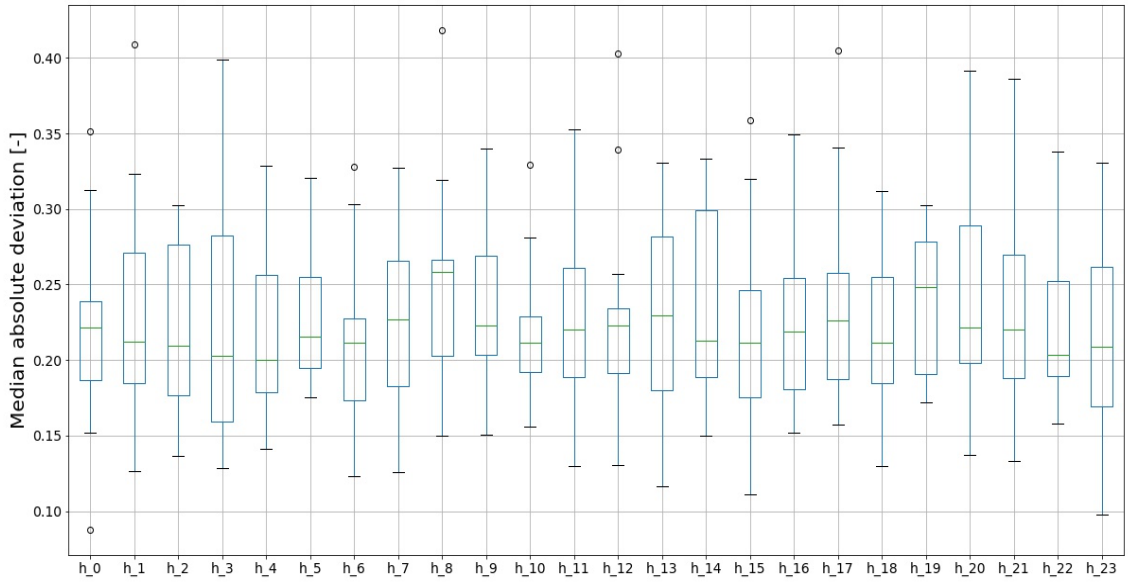


Figure 6.4: Boxplot of MADs showing hourly changes.

6.3 Discussion

We initiated a project to assess how artifacts in ABP and ICP signals impact the circadian variability of the PRx index. In this study, we analyzed data from 19 out of 27 patients, excluding eight due to a high percentage of missing values. Additionally, some PRx indices were replaced with NaN values if more than 50% of data were missing within a particular hour. The results of the signal analysis can be found in Chapter 6.2. Figure 6.3 shows increased PRx variability during certain morning and evening hours. However, these results are based on patients without labeled artifacts. To obtain a more accurate analysis, it is essential to label all artifacts and evaluate how much of the variability peaks are influenced by artifacts.

7 Simulated artifacts

7.1 Methods

For the development of detection algorithms, "simulated" artifacts are sometimes used for the sake of better algorithm specification and tuning [22]. For our first study [23], where we investigated the effect of the presence of different artifacts on the calculation of the PRx index, we simulated the most common artifacts.

We conducted an analysis of 935 hours of multimodal monitoring data from patients with acute brain injury to identify and characterize common artifacts in ABP and ICP signals. The data contained only vital signs without patient descriptions, and were provided by our colleagues at the Neurointensive Care Unit at the Department of Anesthesiology, Perioperative and Intensive Medicine, Masaryk Hospital in Usti nad Labem, Czech Republic.

The identified artifacts were categorized into five types: rectangular, fast impulse, sawtooth, isoline drift, and constant ICP value. These artifacts were described based on their shape, duration, and amplitude changes observed in the original data.

To simulate these artifacts, we first identified 20 undisturbed 10-minute segments of ABP and ICP waveforms that were free from artifacts. These segments were carefully selected to ensure that they had stable ICP (7–15 mmHg), a MAP of 90 ± 5 mmHg, and a PRx value below 0.3, indicating preserved cerebrovascular autoregulation.

Using MATLAB (version R2019a, MathWorks, Natick, Massachusetts, USA), we developed mathematical models of the identified artifacts by replicating the observed characteristics, such as duration and amplitude, based on real data. These models allowed us to accurately simulate the artifacts within the waveform segments, as detailed in Table 7.1, which outlines the specific mathematical functions used for each type of artifact.

Table 7.1: Mathematical functions for simulation of artifacts

Type of artifact	Mathematical function for artifact simulation
Rectangular	$A(t)_{new} = A(t) \cdot 0.1 + \bar{A} \cdot \left(1 + \frac{R_{amp}}{100}\right)$
Fast impulse	$A(t)_{new} = A(t) + \bar{A} \cdot \left(1 + \frac{R_{amp}}{100}\right)$
Saw tooth	$A(t)_{new} = sin(t) \cdot \bar{A} \cdot \left(1 + \frac{R_{amp}}{100}\right) + \bar{A}$, where $t = 0{:}\frac{\pi}{2 \cdot F_s}{:}\frac{\pi}{2}$
Isoline drift	$A(t)_{new} = A(t) + sin(t) \cdot \bar{A} \cdot \frac{R_{amp}}{100}$, where $t = 0{:}\frac{\pi}{F_s}{:}\pi$
Constant ICP value	$A(t)_{new} = 8$

Here is the breakdown of the symbols used in the equations:

- $A(t)$ represents the original signal amplitude at time
- $A(t)_{new}$ represents the new signal amplitude at time t after the artifact has been added.
- $\bar{A}$ represents the average amplitude of the original signal over a specified time window.

- R_{amp} specifies the percentage increase or decrease in amplitude due to the artifact.
- t is time variable that ranges from the start to the end of the artifact's influence on the signal.
- F_s represents sampling frequency of the signal.

These simulated artifacts were inserted into the previously selected undisturbed segments at a specific time point (4 minutes and 1 second) within the 10-minute segment. We inserted each artifact either into one or both pressure signals (ABP and ICP) to assess their impact on the PRx calculation.

7.2 Results

We identified two primary groups of artifacts in the dataset: stereotyped and complex. Stereotyped artifacts (see Figure 7.1) were defined by their shape, duration, and amplitude rise (see Table 7.2), allowing them to be categorized into types such as rectangular artifacts, fast impulses, sawtooth artifacts, isoline drifts, and constant ICP values. In contrast, complex artifacts lacked these clear and consistent characteristics, making them difficult to categorize or model using standard analytical methods. On average, each patient had 166 artifacts in a 24-hour signal segment, with variations in both length and amplitude [10].

Table 7.2: Parameters of simulated artifacts (duration and amplitude rise) [10].

Type of artifact	Parameters of modelled artifacts	
	Duration (s)	Amplitude rise (%)
Rectangular	4; 15; 30; 60	25; 50 ;75; 100
Fast impulse	0.04	25; 50; 75; 100; 125
Saw tooth	30; 45; 90	30; 60
Isoline drift	15; 30; 60; 120	15; 30
Constant value	4	Constant value of ICP 8 mmHg

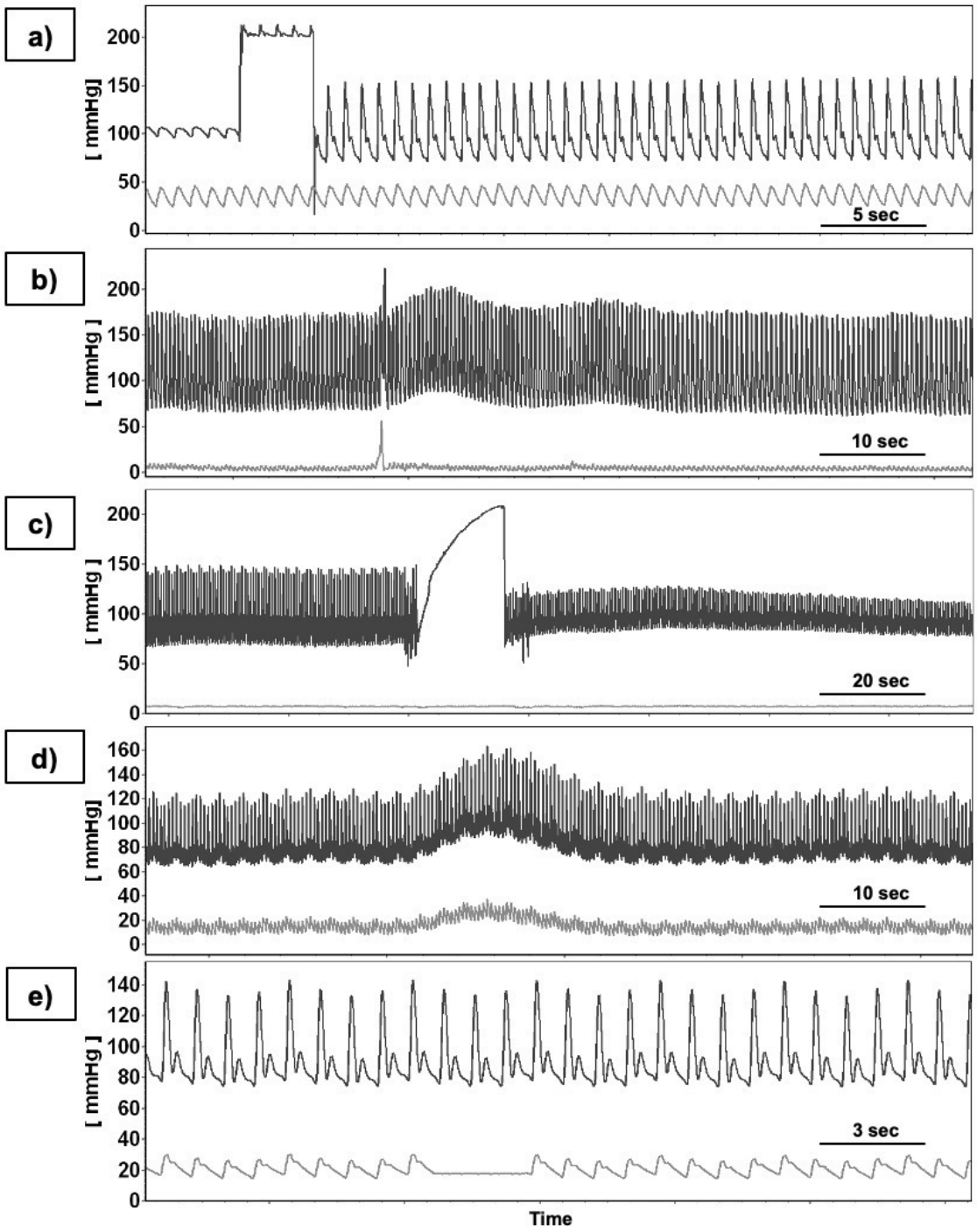


Figure 7.1: Example of observed stereotyped artifacts: a) rectangular, b) fast impulse, c) saw tooth, d) isoline drift, e) constant value [10].

In addition to identifying and categorizing the artifacts present in the original data, we created simulated versions of these artifacts. Each type of artifact—rectangular, fast impulse, sawtooth, isoline drift, and constant ICP value—was mathematically modeled based on the observed characteristics such as shape, duration, and amplitude rise. These simulated artifacts (see Figure 7.2) were then systematically inserted into the undisturbed signal segments.

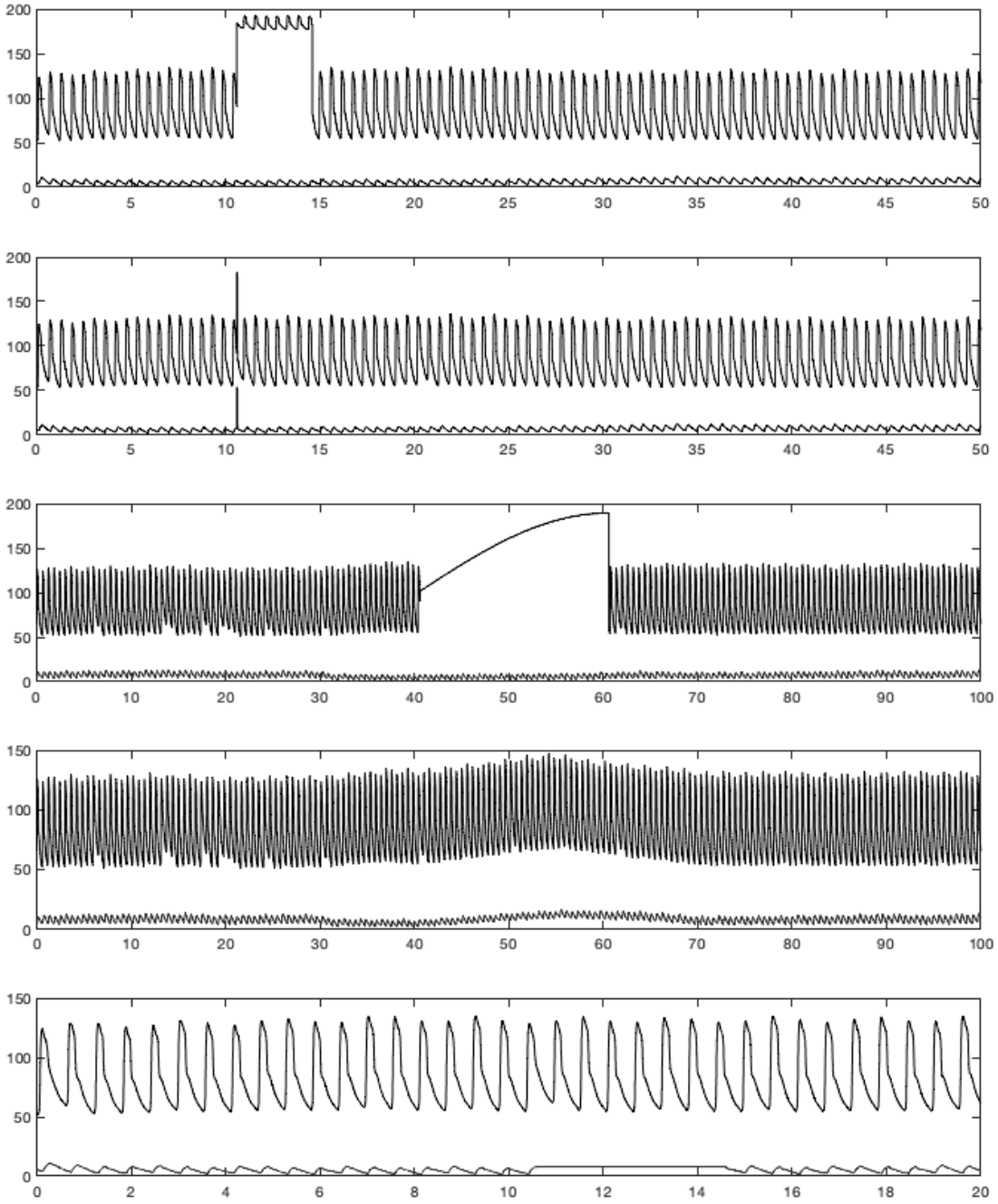


Figure 7.2: Example of simulated artifacts: a) rectangular, b) fast impulse, c) saw tooth, d) isoline drift, e) constant value [10].

Four types of artifacts—rectangular, fast impulse, sawtooth, and isoline drift—were inserted into both ABP and ICP signals. Even the smallest rectangular artifact (lasting 4 seconds and causing a 25% amplitude increase) led to a rise in PRx above 0.3 in 55.4% of cases. Longer artifacts (15 seconds or more) caused the PRx to exceed 0.3 in all samples, with values often approaching 1.0 (for a 15-second artifact: PRx 0.88 [0.81–0.93]), see Figure 7.3. The sawtooth artifact had an even stronger effect, with a 30% amplitude increase causing PRx to reach 1.0, regardless of duration. In contrast, the isoline drift affected PRx only when the drift lasted at least 1 minute and the isoline rose by nearly a third, causing a PRx increase above 0.3 in 21.6% of cases. The

fast impulse artifact did not cause a significant change in PRx. When artifacts were introduced only into the ABP, the impact on PRx was much smaller, with critical values reached in just 4.3% of rectangular and sawtooth artifacts. Lastly, the simulation of a constant ICP value for 4 seconds did not result in a significant change in PRx [10].

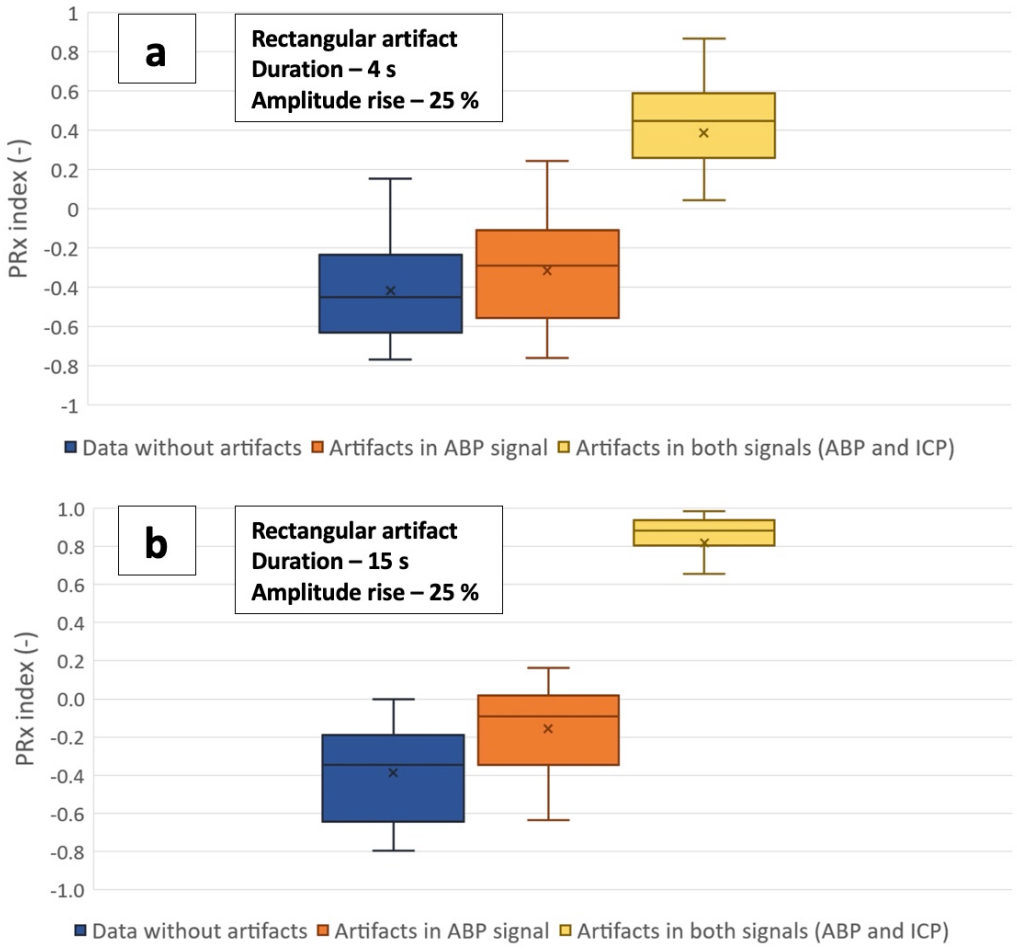


Figure 7.3: The effect of the rectangular artifact on PRx value: a) rectangular artifact lasting 4 s with an amplitude rise of 25% in the ABP and in both signals, b) rectangular artifact lasting 15 s with an amplitude rise of 25% in the ABP and in both signals [10].

7.3 Discussion

We simulated the most common stereotyped artifacts by mathematically modeling their characteristics based on the observed data. While we did not simulate every possible type of artifact—given the large variety—we focused on those that are most prevalent. Our observations showed that many artifacts in real signals were complex, but their individual components often aligned with stereotypical artifacts. As a result, we concentrated on simulating only a few basic types and examined their impact on the PRx index to identify which artifacts have the greatest influence and what should be prioritized during artifact detection. The rectangular and sawtooth artifacts caused the most significant changes in PRx, while other artifact types—fast impulse, isoline drift, and constant ICP—did not significantly affect PRx values.

8 Effect of high-frequency artifacts on pressure reactivity index

8.1 Methods

As previously outlined, one of the primary objectives of this dissertation was to detect artifacts in the ABP signal to ensure its correct use in the calculation of the PRx index. At the outset of this project, our goal was to identify the factors that could influence the PRx index value and determine which artifacts needed to be detected.

In one study, we simulated various types of artifacts and introduced them into either the physiological ABP signal, the ICP signal, or both simultaneously [10]. We then calculated the PRx index before and after the insertion of these artifacts. The key finding from this study was that the impact of artifacts on PRx varied depending on their shape, duration, and whether they were present in one or both signals. Initially, we used a fast pulse as a stand-in for high-frequency artifacts and found that it had no significant effect on the PRx index calculation. However, through further discussions with the team and analysis of real ABP signals, I identified another common form of high-frequency artifact: high-frequency noise. This noise significantly distorts the arterial pressure signal and appears to have the potential to affect the PRx index calculation, which required further verification. Therefore, we decided to test this hypothesis [50].

We extracted data from an anonymized database of physiological signals recorded in the Neurointensive Care Unit at Masaryk Hospital in Usti nad Labem, Czech Republic. ABP and ICP were monitored using specific equipment, and the data were captured using ICM+ software (version 8.6, Cambridge Enterprise Ltd., Cambridge, UK). We selected twenty 10-minute signal segments that were free from any artifacts, based on specific physiological criteria [50].

Further data processing was performed using Python (version 3.8.8, Python Software Foundation, Wilmington, Delaware, USA), which included generating the noise, processing the signals, and calculating the PRx values. The comparison focused on assessing the effects of high-frequency noise on the PRx values [50].

We simulated high-frequency noise in the ABP waveform using a band-pass finite impulse response (FIR) filter with a Kaiser window. To determine the appropriate frequency range for the noise, we applied a STFT to a real ABP signal with noise. The noisy signal was passed through a 5-second STFT window, and we plotted 20 frequency spectrum curves (see Figure 8.1). By analyzing these curves, we identified the frequencies most affected by noise and selected the 5–25 Hz range for the noise simulation. This range was chosen because it accurately represented the high-frequency noise typically found in ABP signals, allowing us to model the noise effectively for subsequent analysis [50].

PRx indices were then calculated from both the original and noisy ABP signals using a 300-second window and a 60-second shift. To compare the impact of noise on PRx variability, we calculated the median absolute deviation for both signal types [50].

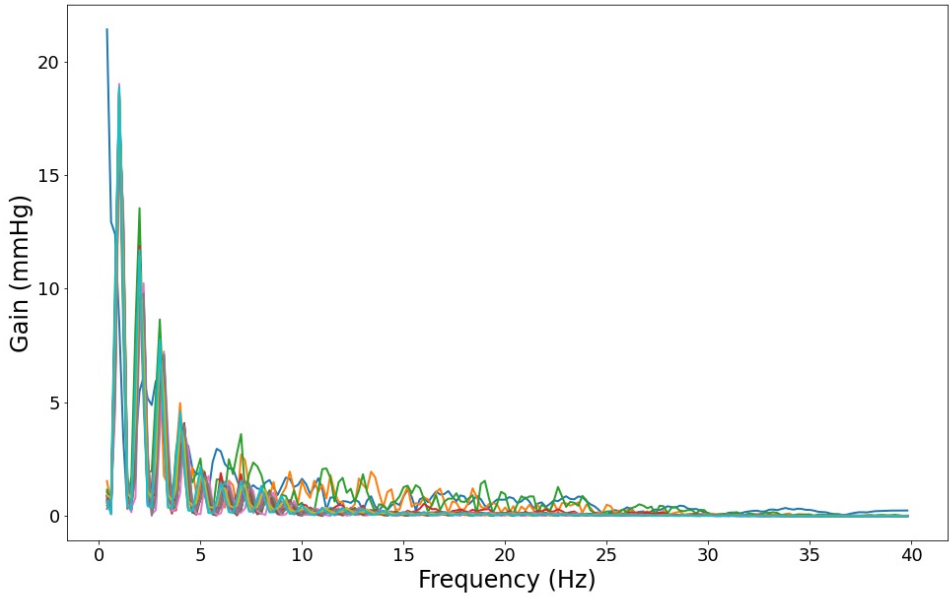


Figure 8.1: 20 spectral curves of the ABP signal with real noise plotted on top of each other [50].

8.2 Results

In this study, we examined the impact of high-frequency noise in ABP signal on PRx calculations. We compared the original ABP signal, the ABP signal with simulated high-frequency noise, and the ABP signal with real noise (see Figure 8.2). The analysis showed that the average MAD for the original PRx values was 0.1129, while for the PRx values calculated from the noisy ABP signal, it was 0.1123 (see Figure 8.3). This small difference indicates that high-frequency noise in the ABP waveform had a negligible impact on the PRx calculation [50].

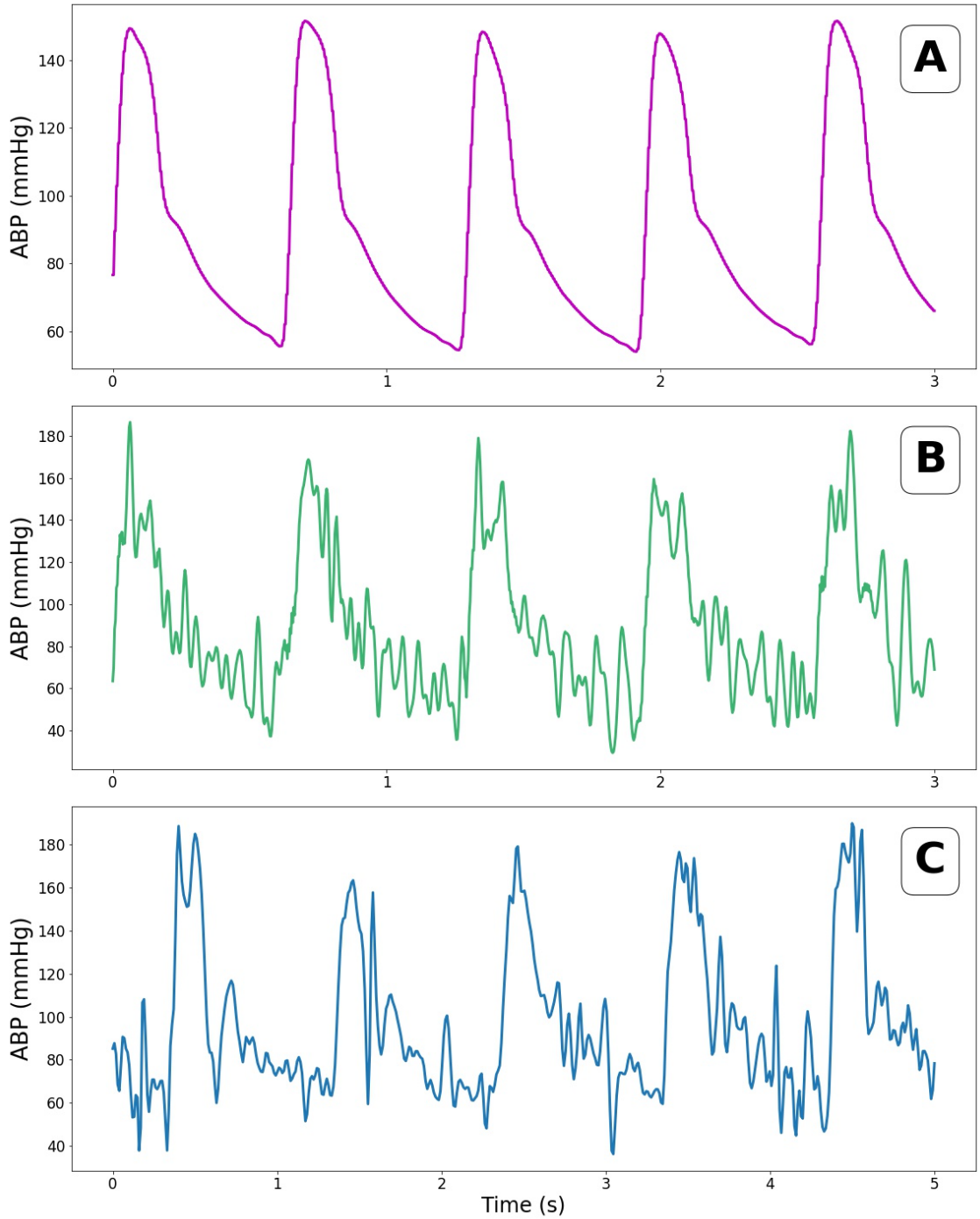


Figure 8.2: A comparison of real and simulated artifacts. A – undisturbed ABP signal without artifacts; B – undisturbed ABP signal with simulated high- frequency noise; C – ABP signal with real high-frequency noise [50].

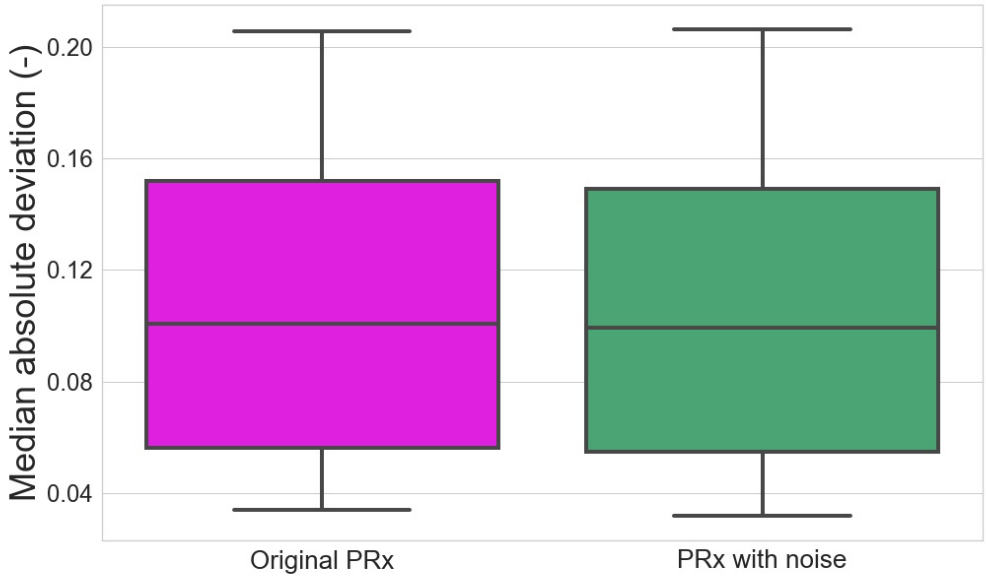


Figure 8.3: A comparison of the calculated MAD values between the original PRx and the noisy PRx [50].

8.3 Discussion

Our findings indicate that high-frequency artifacts in the ABP waveform, within the 5–25 Hz range, do not significantly impact the calculation of the PRx. Although these artifacts meet the standard definition of noise, their effect on PRx is minimal, supporting the notion that not all artifacts need to be removed for accurate PRx calculation. This approach could simplify artifact detection algorithms, making them less computationally demanding without compromising the accuracy of clinical decision-making. However, it is important to consider that this study used simulated noise, and further analysis with real noise is necessary to fully validate these conclusions [50].

9 Choosing a calculation window of Short-time Fourier transform for artifact detection

9.1 Methods

In Section 2.3 (Automatic Detection and Elimination of Artifacts in the ABP), we reviewed various artifact detection methods developed by other research teams. However, none have applied the STFT for this purpose. We aimed to test the use of STFT to detect different types of artifacts.

I simulated five typical types of artifacts (see Table 9.1) in Matlab software (MATLAB R2020a) and inserted them into the original undisturbed arterial blood pressure signal (or into the ICP signal, in case of the constant ICP value artifact). The appearance and duration of the simulated artifacts are based on the observed artifacts in real signals. I exported 50-second or 100-second part of the signal with an artifact (depending on the artifact duration) to the .CSV format for further signal analysis in Python.

In Jupyter notebooks using *signal.stft* function (SciPy library), I have performed STFT on the exported signals disturbed by the different artifacts. I have specified a time series of measurement values (signal section with the artifact), window length, and sampling frequency to use this function, see Table 9.1. Different window lengths were always chosen considering artifact duration. According to the literature [46], the window length must be selected to assume that the signal is stationary during this period. I left the window type as the default - Hanning window. I have also left the length of overlap window at the default setting, which is set to half of the calculation window.

Table 9.1: Duration of simulated artifacts and corresponding set parameters for STFT calculation.

Artifact type	Modelled artifact duration [s]	STFT calculation window length [s]	Sampling frequency [Hz]
Rectangular	4	1; 3; 5; 7; 10	300
Saw tooth	20	5; 7; 10; 20; 25	
Isoline drift	30	5; 7; 10; 20; 30; 35	
Constant ICP value	4	1; 4; 5; 7; 10	
Fast impulse	0.04	0.5; 1; 3; 5; 7	

In the end, I performed STFT on the three signal sections with obvious complex artifacts. Complex artifacts can not be described in terms of their shape, duration, and amplitude rise. I have chosen 5 seconds as the STFT calculation window length because it had good results on different artifacts.

9.2 Results

Figures 9.1-9.10 display different types of artifacts that were analysed using an STFT 5-second window. In Annex A are displayed the same artifacts, but analysed using different STFT window lengths.

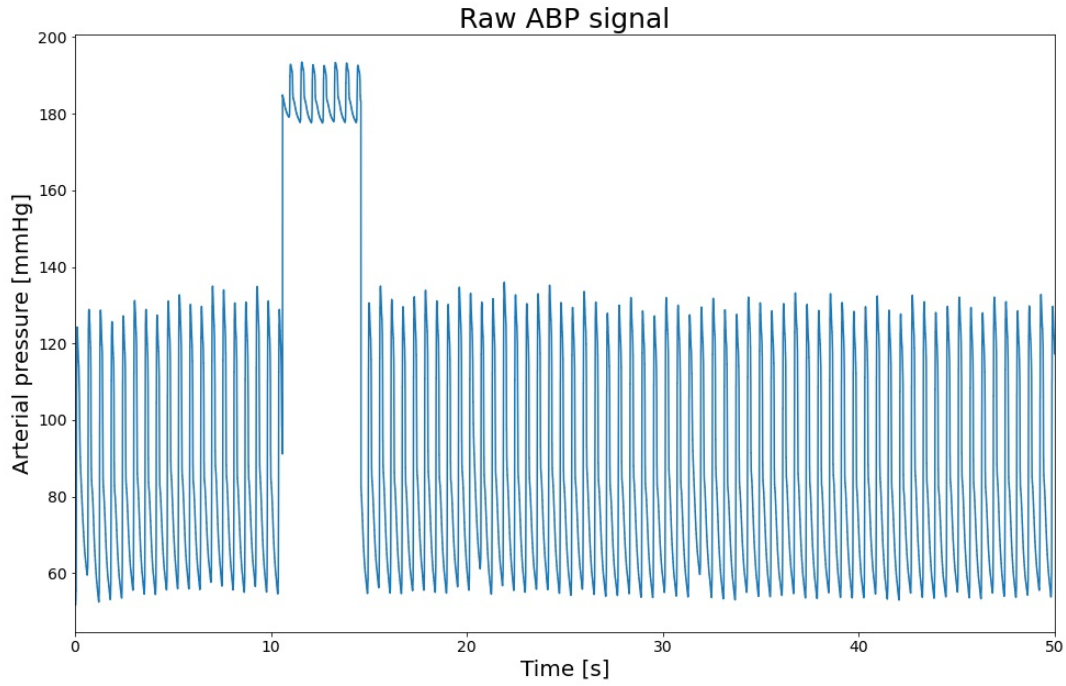


Figure 9.1: Rectangular artifact in ABP signal - duration 4 s.

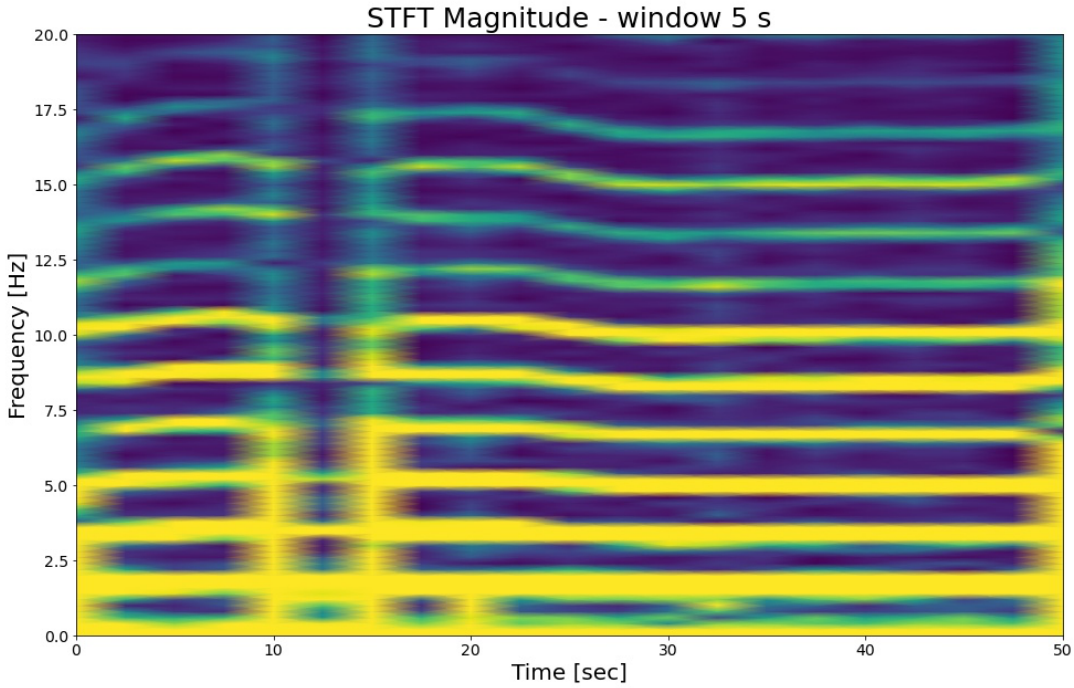


Figure 9.2: Calculated STFT with a 5-second moving window on a signal with a rectangular artifact.

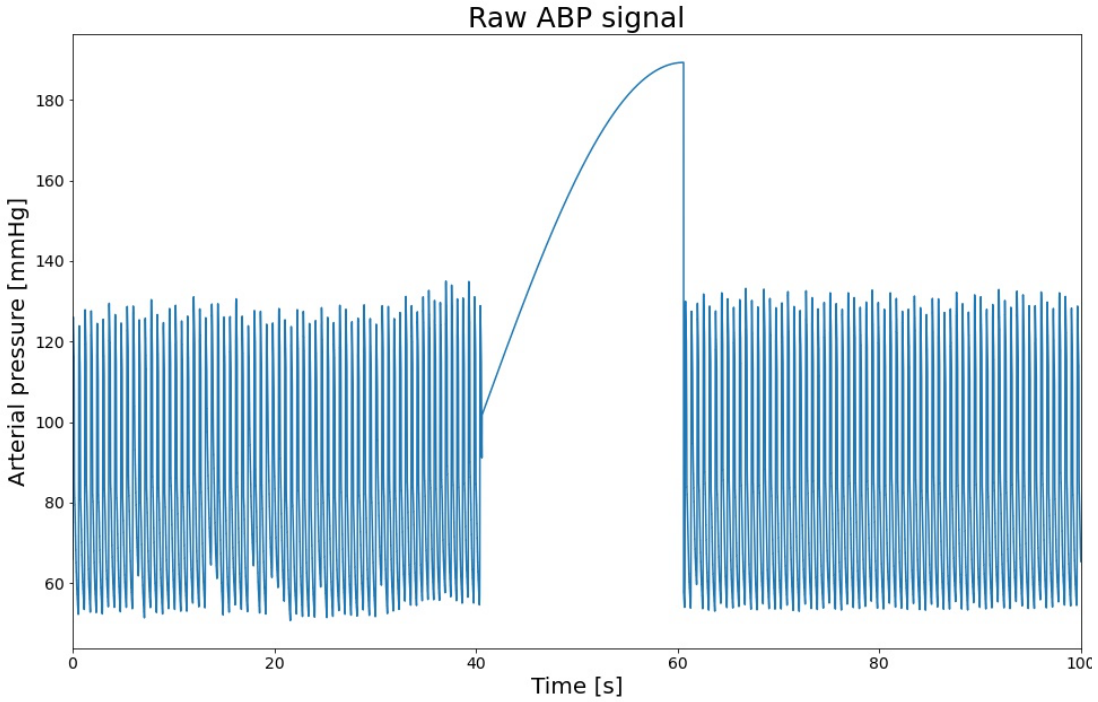


Figure 9.3: Saw tooth artifact in ABP signal - duration 20 s.

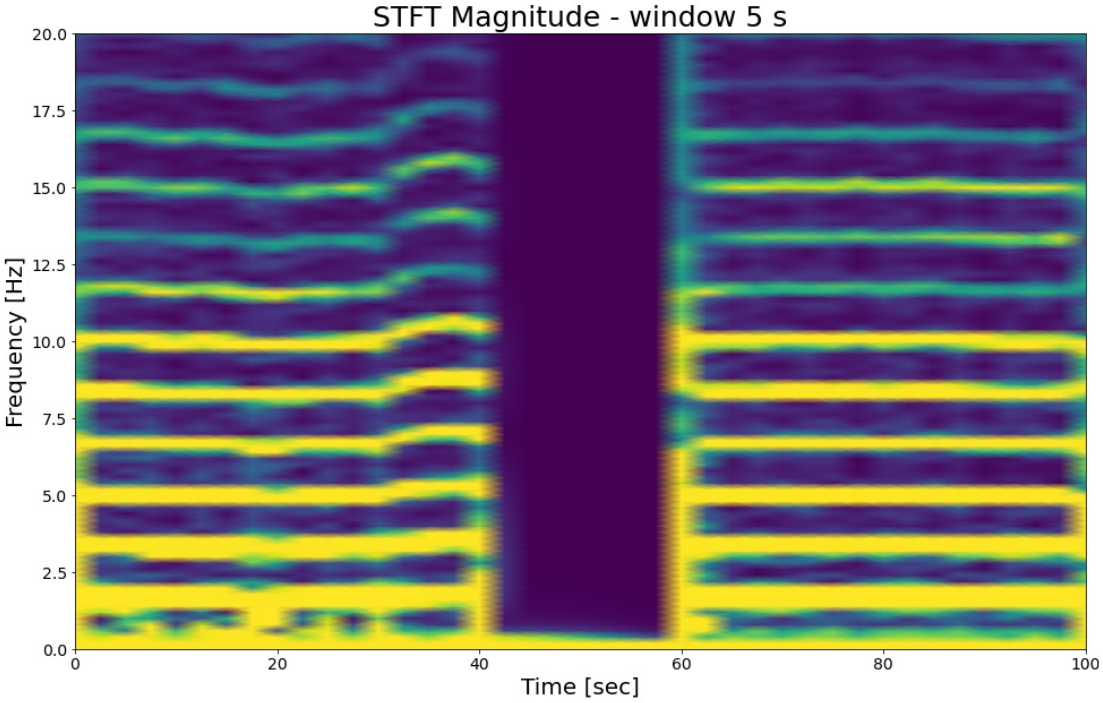


Figure 9.4: Calculated STFT with a 5-second moving window on a signal with a saw tooth artifact.

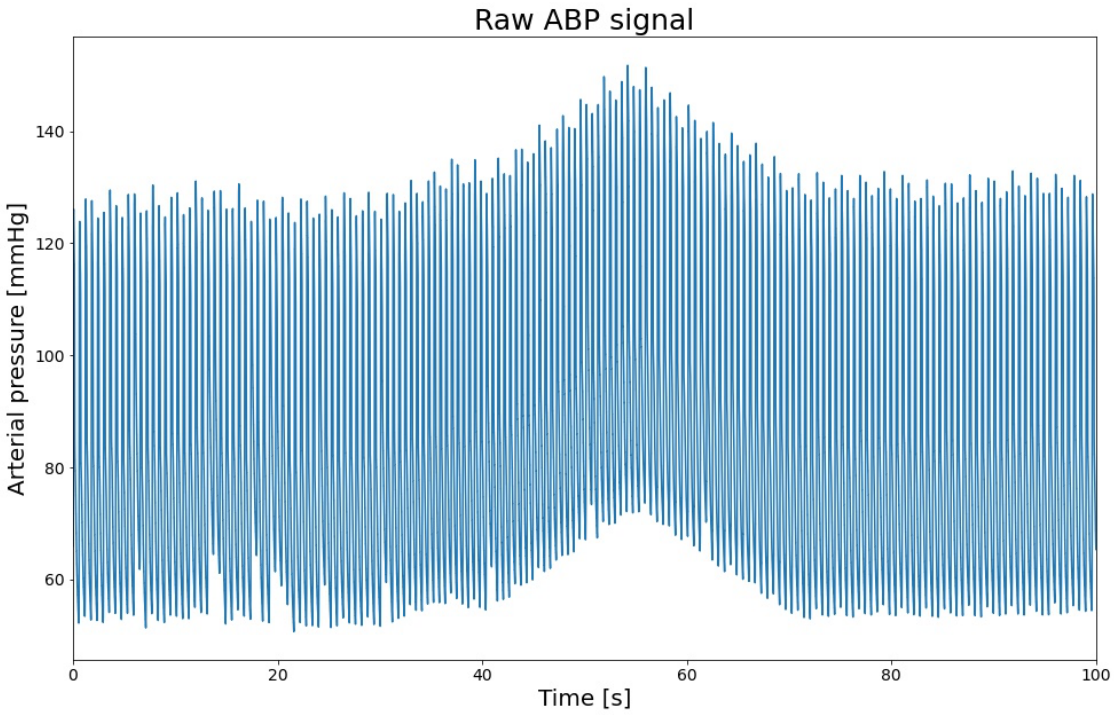


Figure 9.5: Isoline drift artifact in ABP signal - duration 30 s.

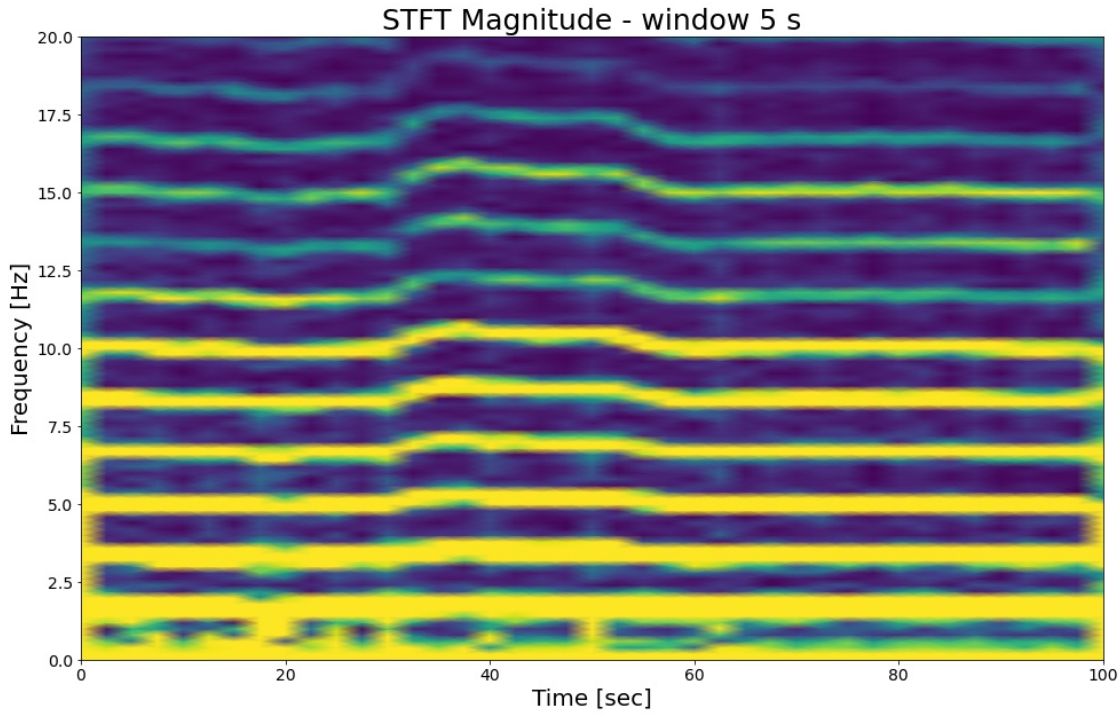


Figure 9.6: Calculated STFT with a 5-second moving window on a signal with an isoline drift artifact.

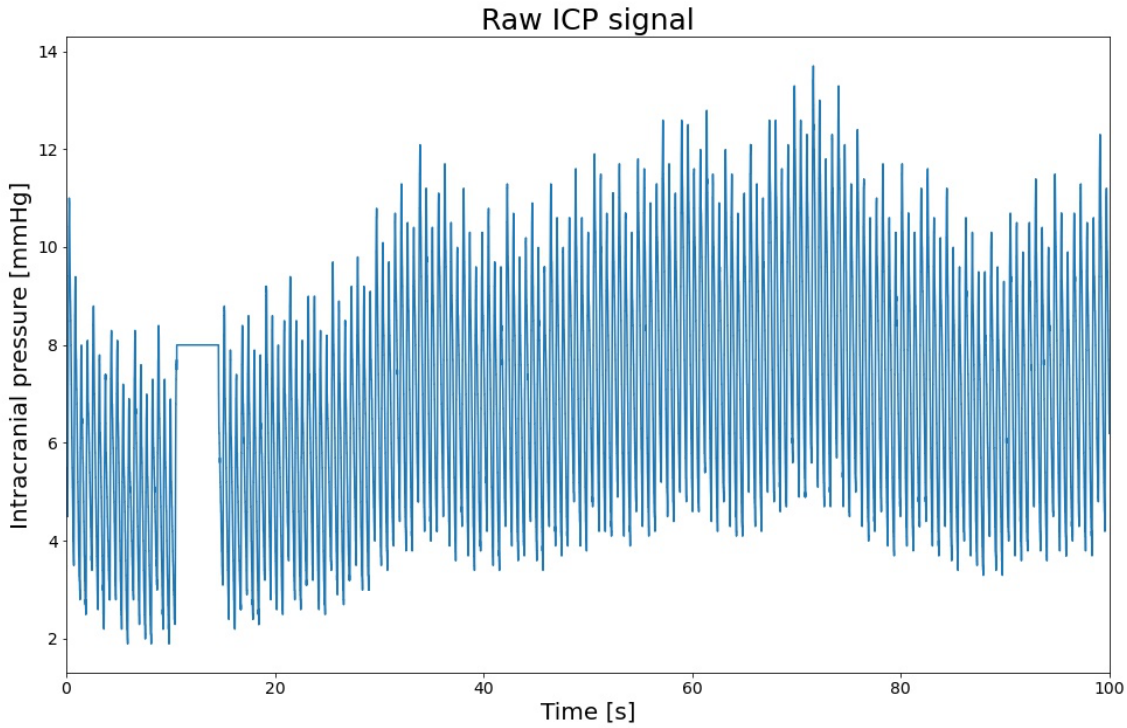


Figure 9.7: Constant value artifact in ICP signal - duration 4 s.

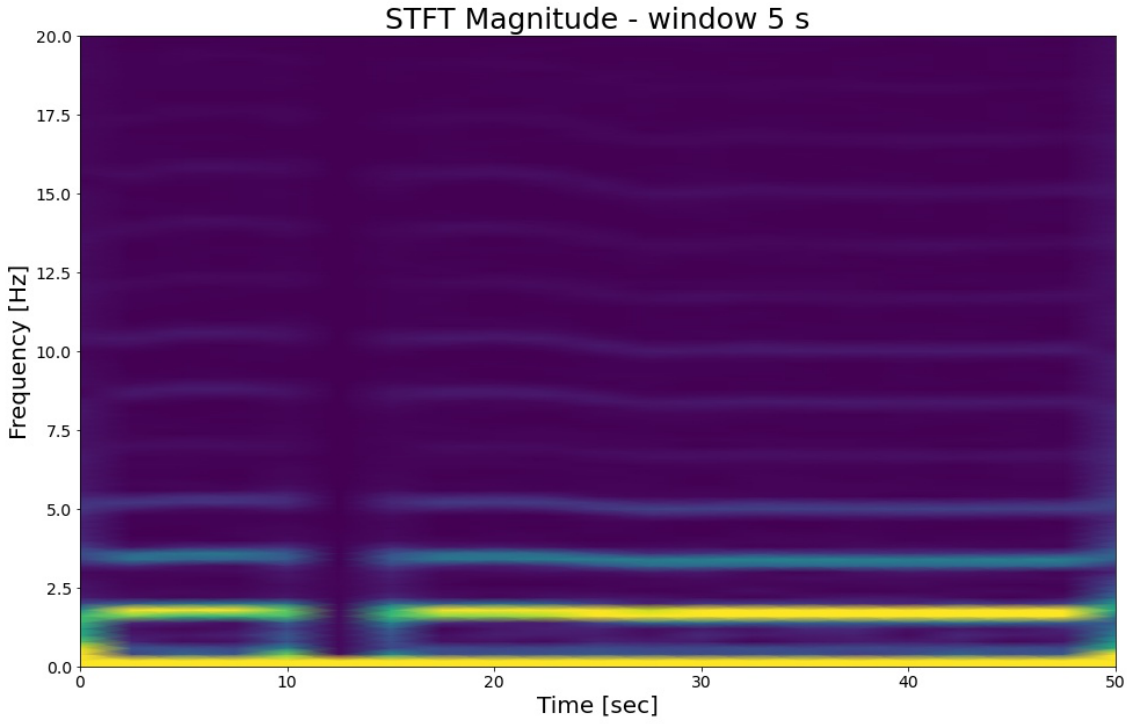


Figure 9.8: Calculated STFT with a 5-second moving window on a signal with a constant ICP value artifact.

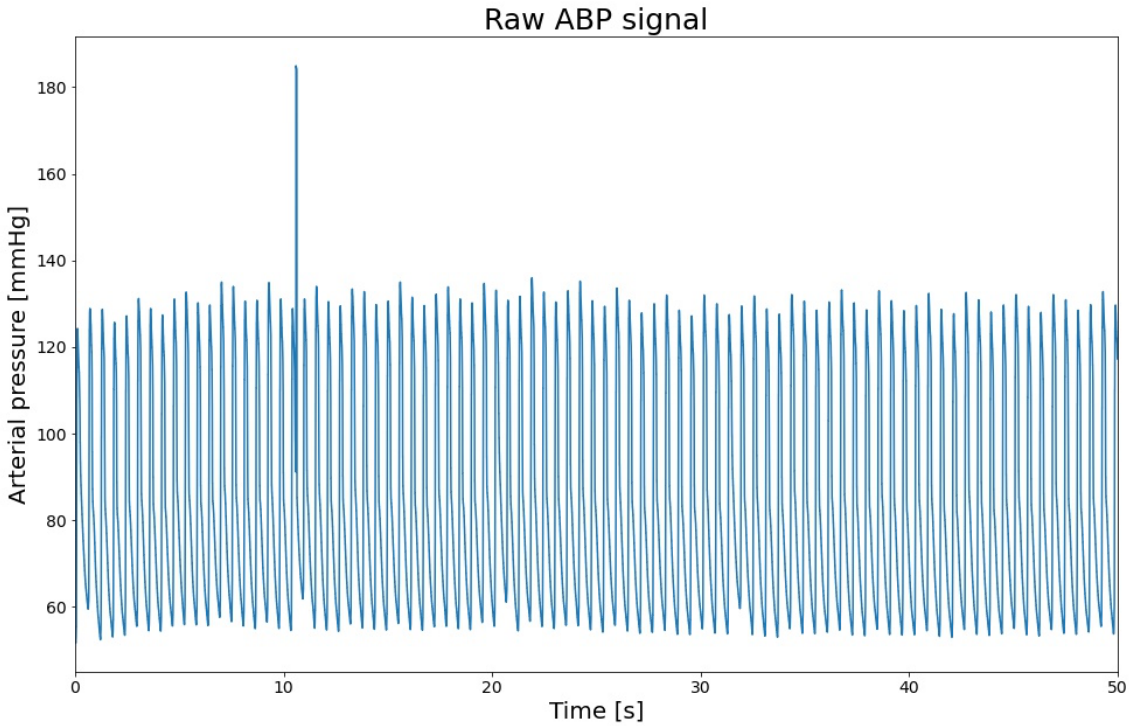


Figure 9.9: Fast impulse in ABP signal - duration 0.04 s.

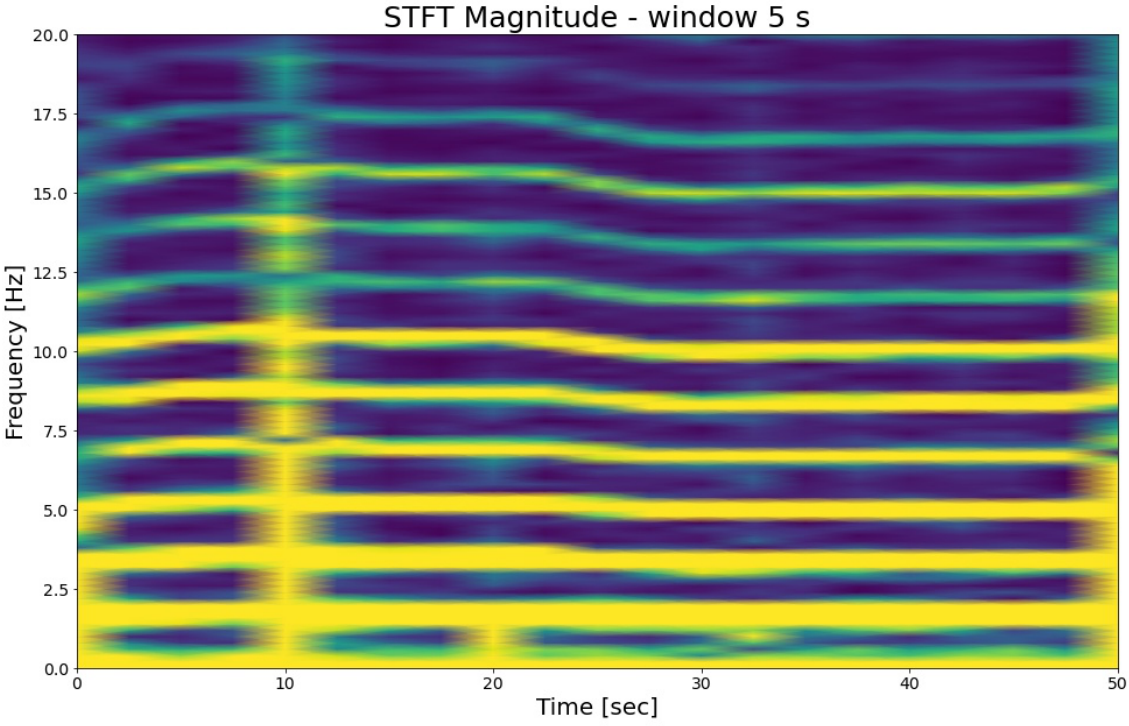


Figure 9.10: Calculated STFT with a 5-second moving window on a signal with a fast impulse artifact.

Figures 9.11-9.16 show an STFT that was performed on different complex artifacts from the real ABP signal.

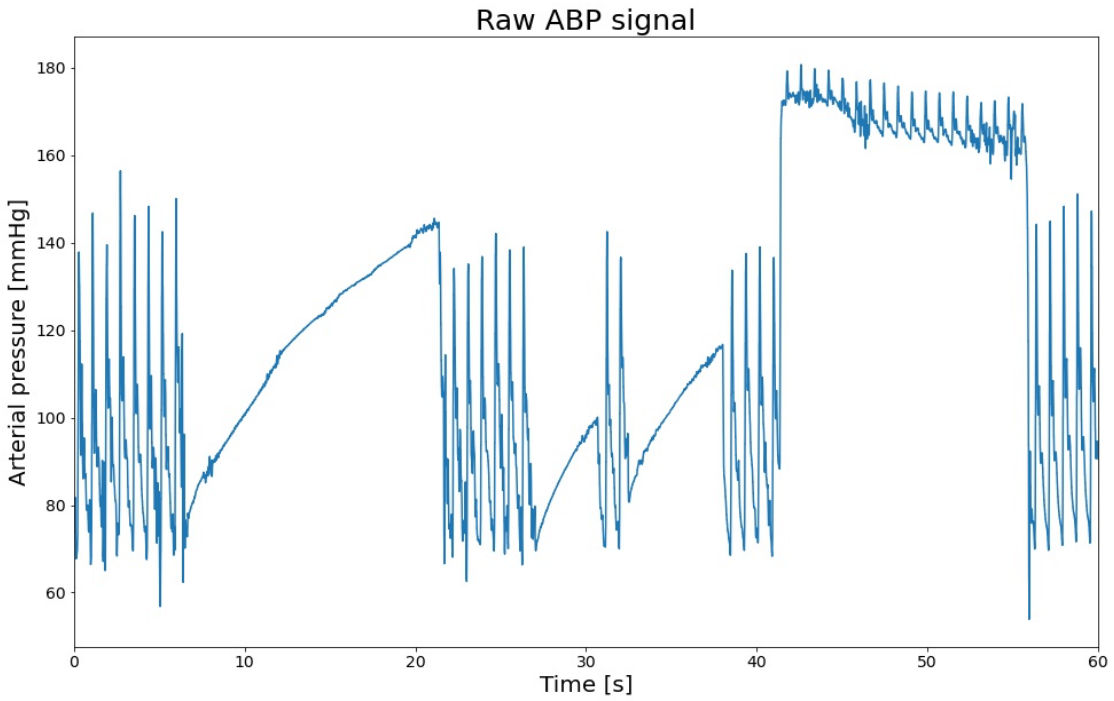


Figure 9.11: Real artifact in a raw ABP signal 1.

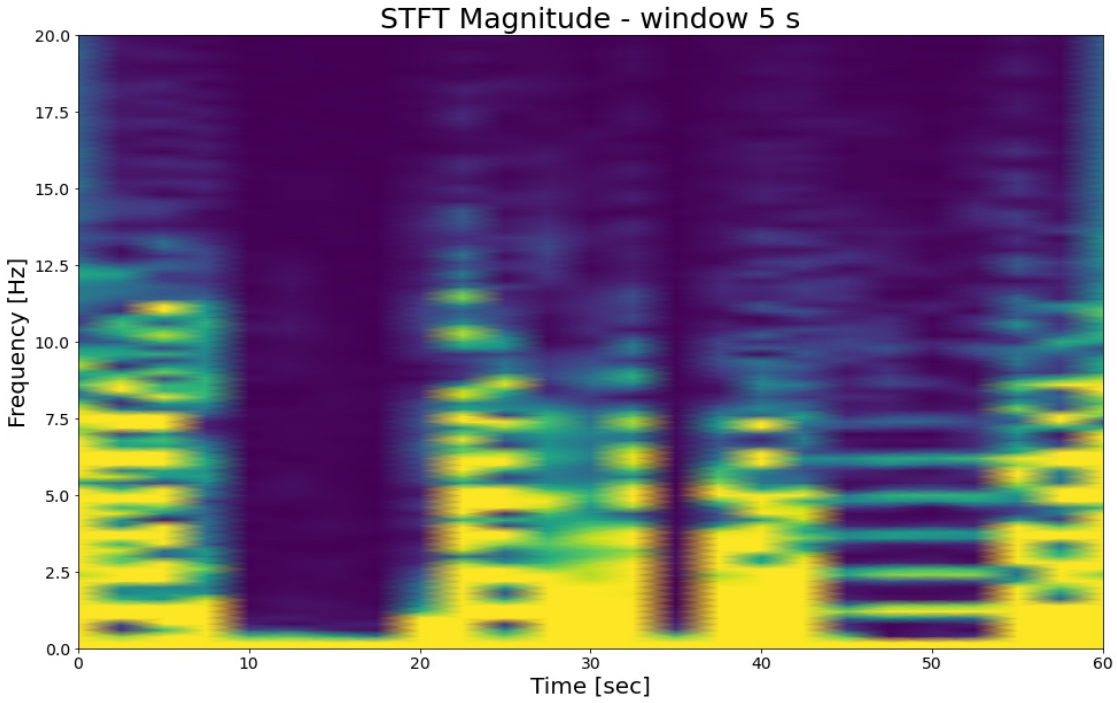


Figure 9.12: Calculated STFT with a 5-second moving window on a signal with a real artifact 1.

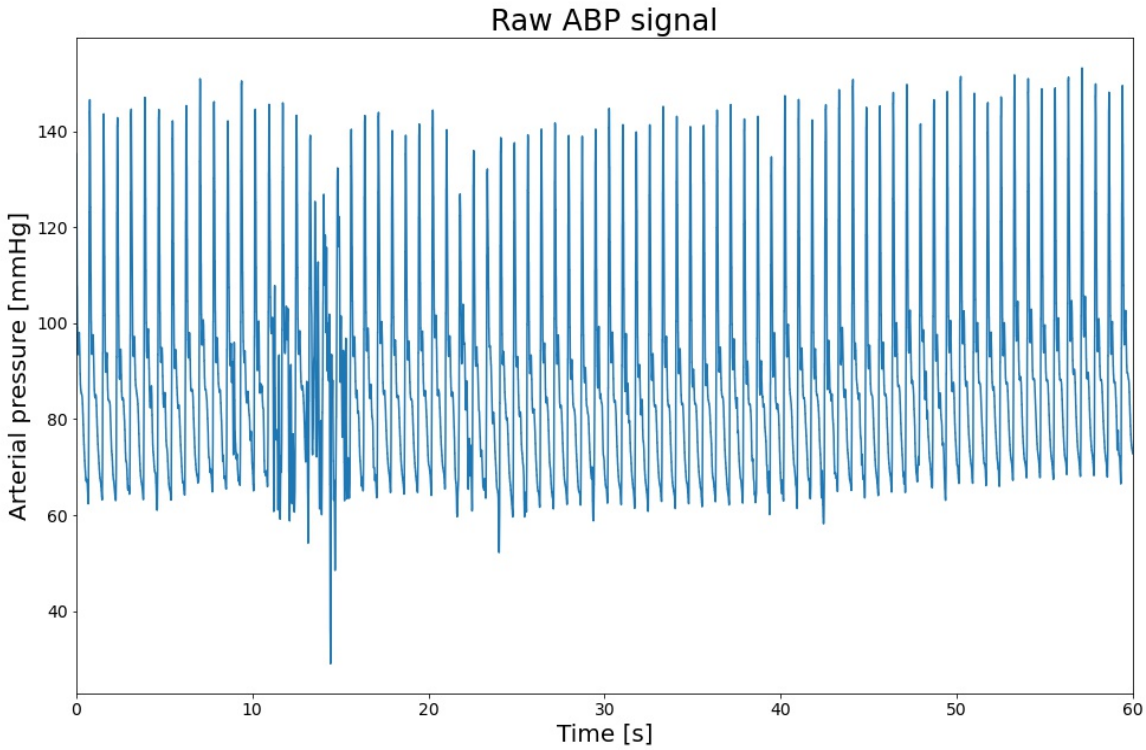


Figure 9.13: Real artifact in a raw ABP signal 2.

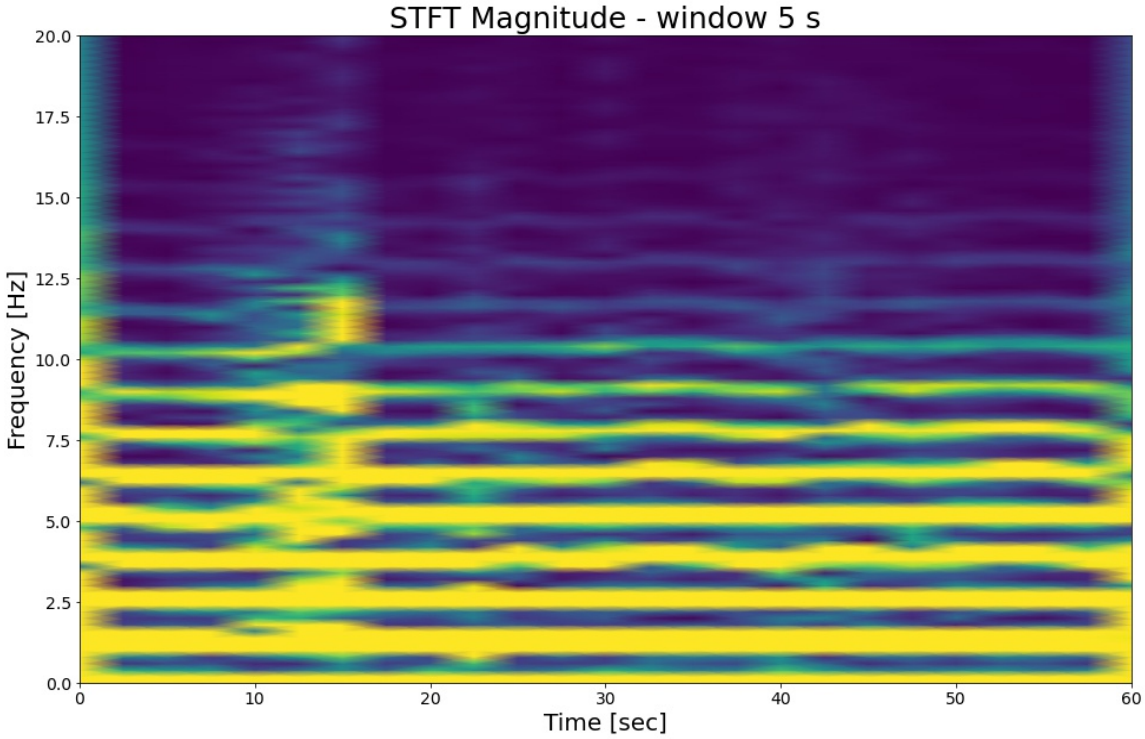


Figure 9.14: Calculated STFT with a 5-second moving window on a signal with a real artifact 2.

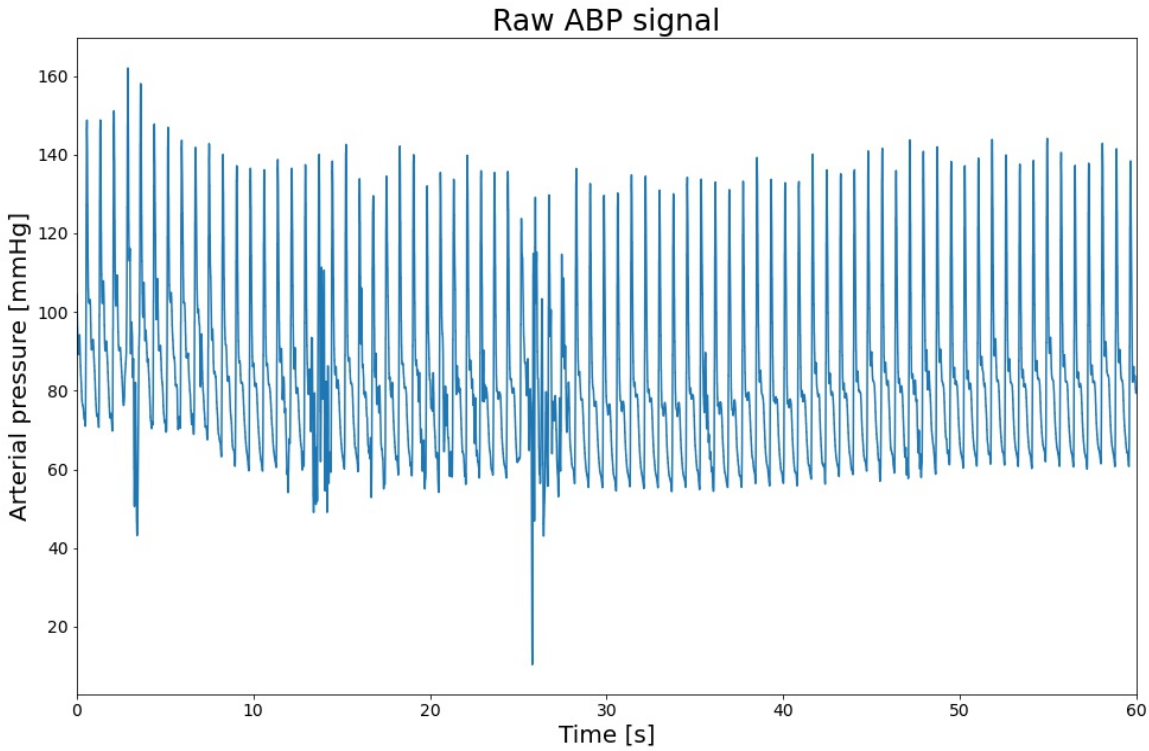


Figure 9.15: Real artifact in a raw ABP signal 3.

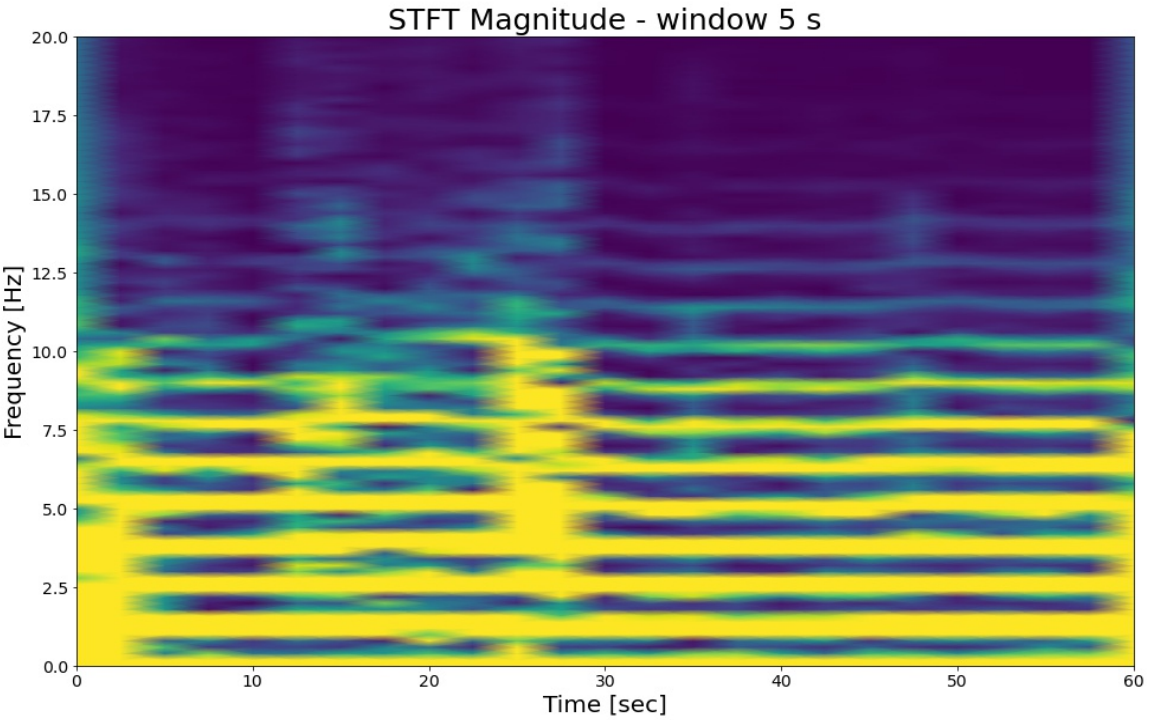


Figure 9.16: Calculated STFT with a 5-second moving window on a signal with a real artifact 3.

9.3 Discussion

I performed the short-time Fourier transform on various artifacts, with the results described in section 9.2. As expected, a 5-7 second STFT window is generally sufficient to detect changes in the signal, particularly for rectangular artifacts, sawtooth artifacts, and even fast impulses. However, for drift isoline artifacts, detection is ineffective, as they are not easily visible on the frequency spectrum. When I applied STFT to constant ICP values, noticeable changes in the spectrum were only observed with window lengths up to 5 seconds. I also applied STFT with a 5-second window to artifacts from real signals, showing that it effectively captures both complex artifacts with a large area under the curve and short interferences.

10 Algorithm for detection of simulated artifacts

10.1 Methods

10.1.1 First attempts at creating a detection algorithm

After selecting the appropriate window length for the STFT calculation to detect basic artifacts, I began developing the detection algorithm. The primary challenge was determining the right parameters to decide whether an artifact was present in a given location.

The STFT provided outputs including the frequency range, timeline, and a matrix containing the amplitude (and phase) values for each frequency component within each time segment. Both frequency and time resolution of the STFT spectrum were crucial, with frequency resolution set at 0.2 Hz and time resolution at 2.5 seconds. Initially, I limited the frequency range for analysis to between 0.4 Hz and 20 Hz, excluding higher frequencies that were irrelevant for my purposes and the 0.2 Hz range due to the dominance of the DC component. Figures 10.1 and 10.2 below show examples of the frequency spectrum for a normal ABP signal and one with an artifact.

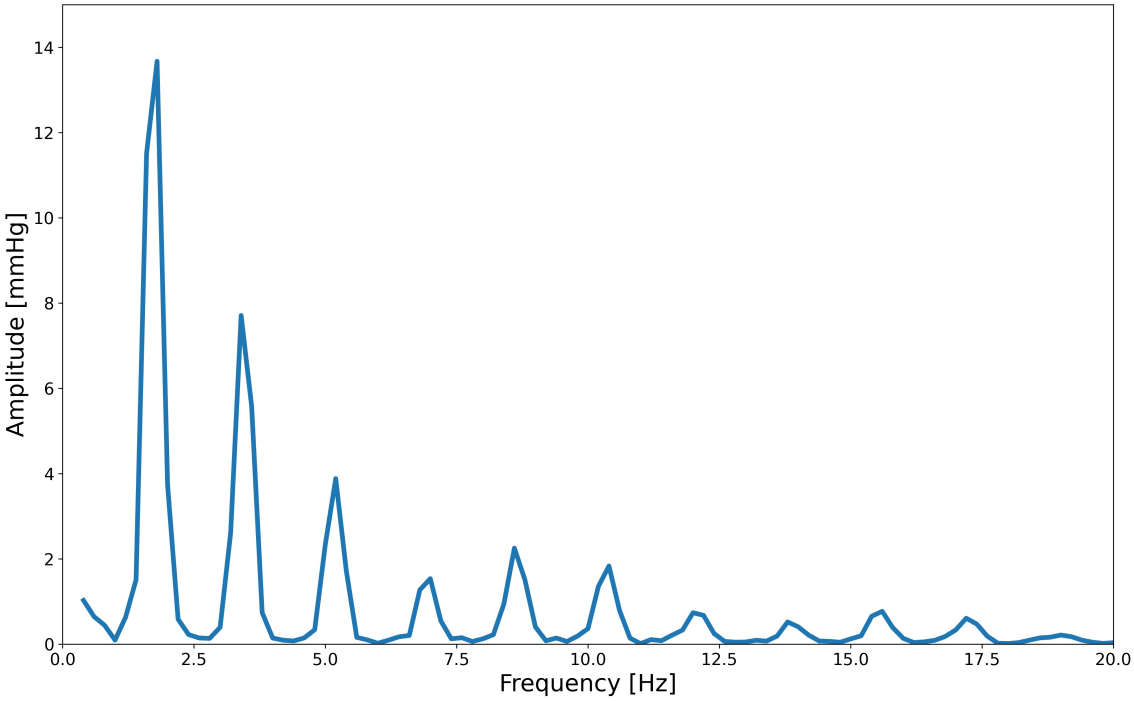


Figure 10.1: Frequency spectrum of a normal signal

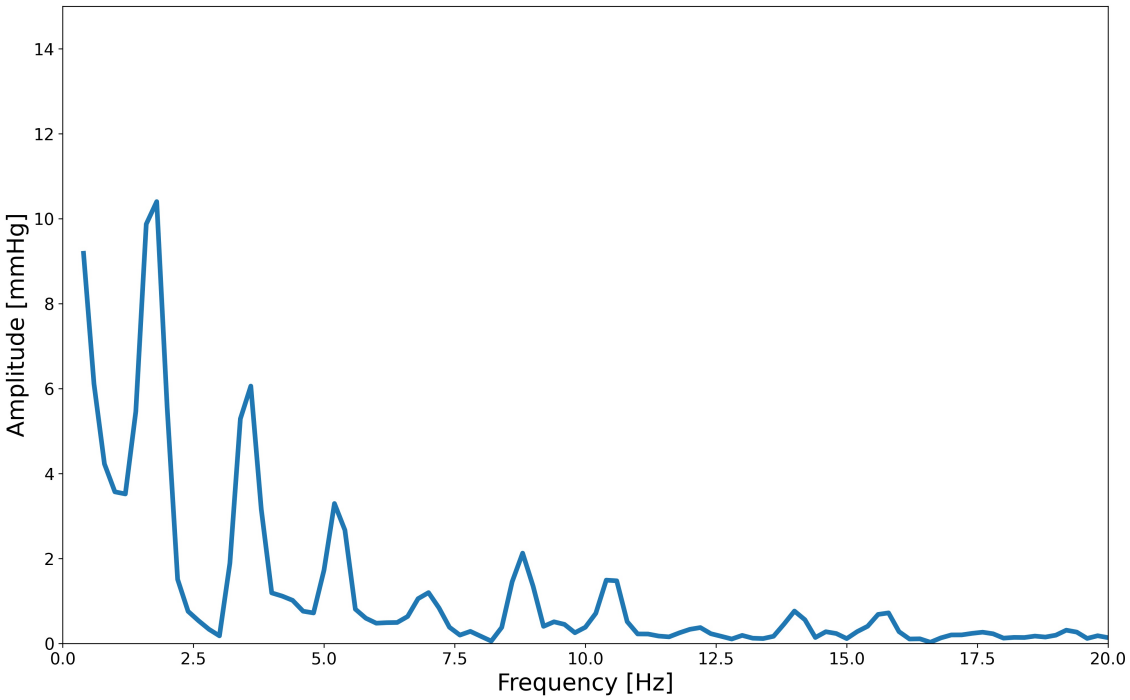


Figure 10.2: Frequency spectrum of signal with rectangular artifact

Next, I attempted to convert the matrix of complex amplitude (and phase) values of each frequency component into a 1D signal. I started by converting the complex numbers into absolute values, and then processed these values across different frequencies for each time point. I calculated the median frequency, mean, median, standard deviation, and sum for each segment (see Figure 10.3). As shown in the figure, only some of these parameters indicated changes in the spectrum. A significant finding was that simple parameters like the sum and mean were effective in capturing the start and end of the artifact. Additionally, changes in standard deviation (STD) were helpful in identifying the duration of the artifact. Therefore, I experimented with combining these parameters and introducing thresholds to help the algorithm identify artifacts.

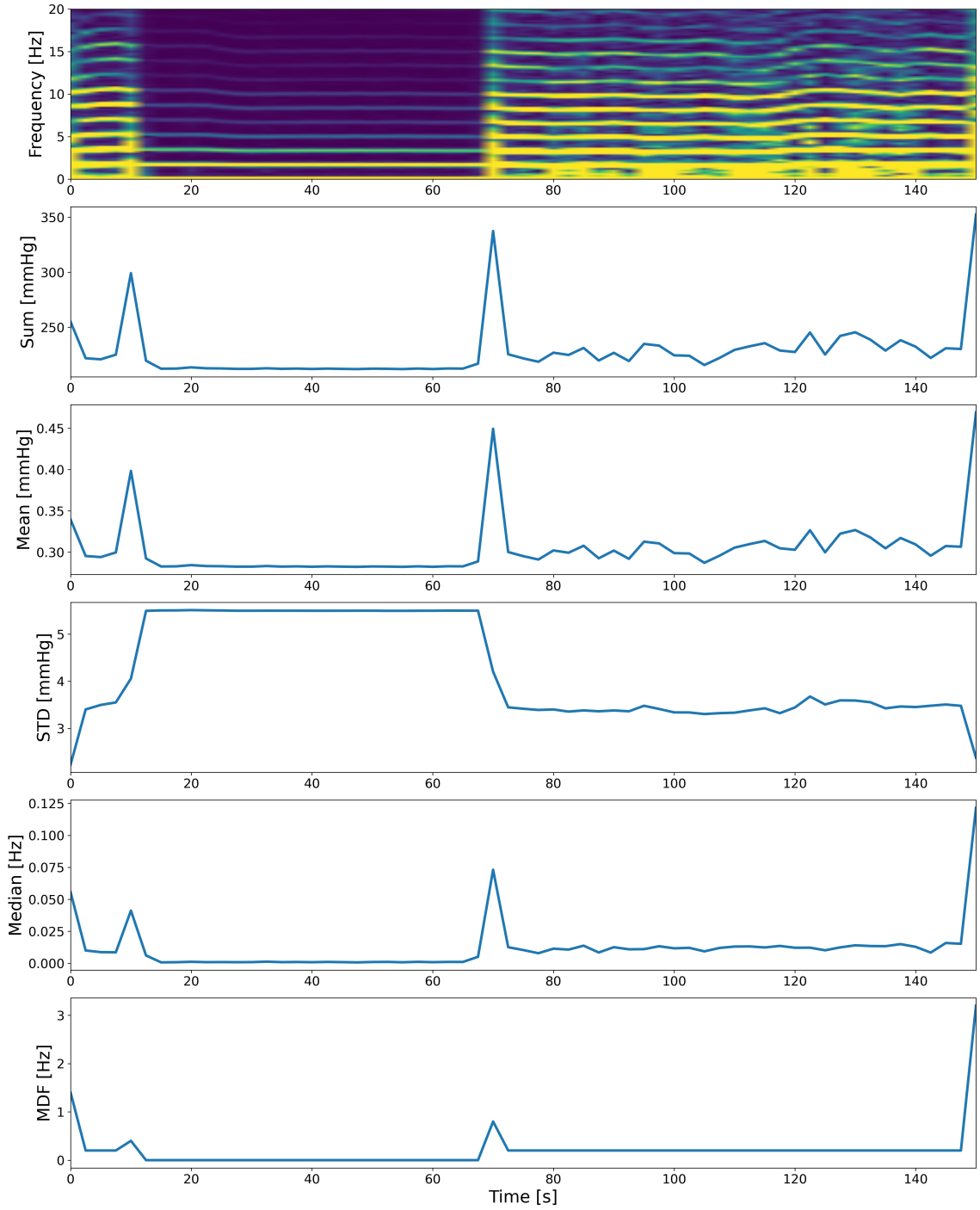


Figure 10.3: Demonstration of tested parameters on the signal with rectangular artifact: sum, mean, standard deviation (STD), median, median frequency (MDF)

However, the standard deviation did not behave consistently across all artifacts; sometimes an artifact increased the STD, while other times it decreased. As a result, I chose not to use the STD in the first version of the algorithm and instead relied on the sum of the amplitude spectrum at each time point.

I then considered monitoring changes in two frequency ranges: 0.2 to 2 Hz and 10 to 20 Hz. My hypothesis was that changes in the power spectrum at these frequencies might help detect different types of artifacts. I calculated the sum of the power spectrum in various frequency ranges for different types of simulated artifacts

(see Table 10.1). The table shows that changes due to baseline drift were not visible in the 10 to 20 Hz range, so I focused further calculations on the 0.2 to 2 Hz range.

Table 10.1: Changes caused by different types of artifacts in different spectral ranges

Artifact type	Power from 0.2 to 2 Hz	Power from 10 to 20 Hz
Rectangular	Increase in total power	Decrease in total power
Saw tooth	Sinusoidal increase in total power	Decrease in total power
Baseline drift	Increase in total power	--------
Fast impulse	Increase in total power	Decrease in total power

When experimenting with different thresholds for artifact detection, I encountered a problem: using absolute values for the power spectrum sum was not feasible, as normal values varied across different recordings and patients. To address this, I standardized the data to establish universal thresholds for artifact detection. The standardized value (z) was calculated using the baseline value (x), mean (μ), and standard deviation (σ):

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Initially, I set the threshold for the standardized total power spectrum value at 1. However, testing revealed that this limit did not work universally across all artifact types, so I adjusted it to 0.5*maximum. Further testing showed that the algorithm performed well with short artifacts but struggled with longer ones. To improve detection, I narrowed the frequency range under investigation to 0.2 to 1 Hz. The algorithm's performance improved, detecting almost all relevant artifacts, though some large artifacts still posed challenges. Therefore, we decided to add additional conditions.

We hypothesized that the dominant frequency in the spectrum would correspond to the heart rate, with its multiples representing harmonic components. We added a condition to monitor the dominant frequency (heart rate) and marked it as an artifact if it fell outside the 0.5 Hz to 3.33 Hz range, corresponding to a heart rate between 30 bpm and 200 bpm. I also examined the harmonic components of the dominant frequency to identify which varied most due to artifacts. Testing revealed that the second harmonic component yielded the best results.

10.1.2 Final algorithm used for detection of simulated artifacts

I applied all the knowledge acquired during the algorithm's development to create its final version, which was used to detect simulated artifacts. The whole data processing is shown in Figure 10.4.

Finally, we conducted a retrospective analysis of high-frequency multimodal monitoring data from a critical care database, focusing on arterial blood pressure (ABP) waveforms. Data were collected from patients in the neurointensive care unit at Masaryk Hospital in Ústí nad Labem, Czech Republic, using Carescape B850 monitors

and sampled at a frequency of 200 Hz. From this anonymized database, we selected 20 segments of unperturbed ABP curves, each lasting 10 minutes, totaling 200 minutes of data. These segments were then divided into four 2.5-minute segments, resulting in a final set of 80 artifact-free segments, all maintaining a mean arterial pressure (MAP) of 90±5 mmHg.

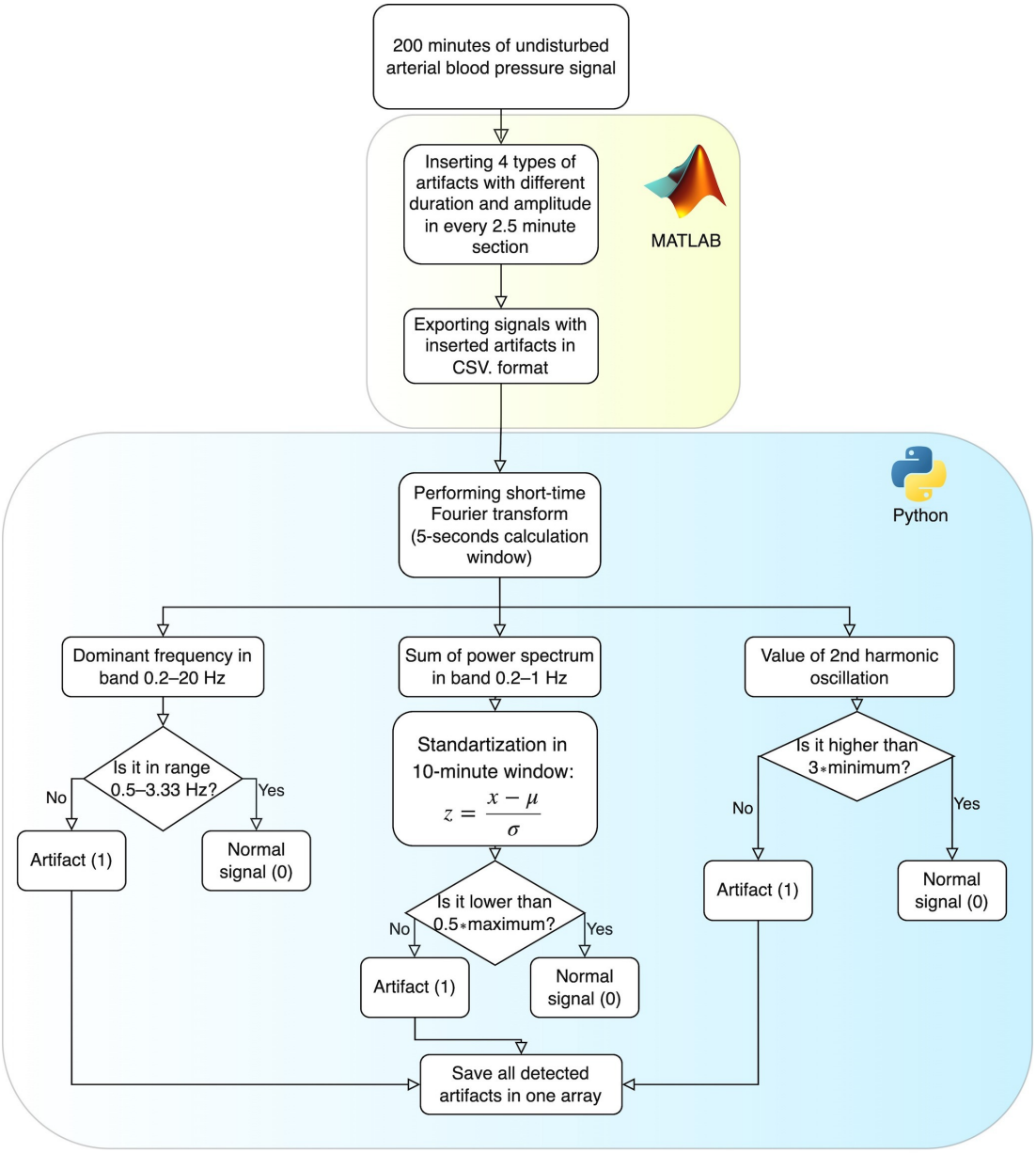


Figure 10.4: Flowchart showing data processing and artifact detection in three rules.

We introduced four types of simulated artifacts—rectangular, fast impulse, sawtooth, and baseline drift—into the ABP waveforms at varying durations and amplitudes. These artifacts were modeled based on characteristics observed in real patient data and were inserted into each of the 80 segments. To detect changes in the frequency domain caused by these artifacts, we applied a STFT using a 5-second window with 50% overlap. This approach allowed us to capture minor changes in the signal, which are critical for identifying artifacts.

To identify artifact-containing segments, we employed an algorithm with three decision-making rules. The first rule analyzed the dominant frequency component within the power spectrum, focusing on the 0.2–20 Hz range. Segments were flagged as containing artifacts if the dominant frequency fell outside the 0.5–3.33 Hz range, which corresponds to the fundamental harmonic of the heart rate in the ABP signal. The second rule evaluated the standardized power spectrum within the 0.2–1 Hz range to detect low-frequency artifacts in segments where the pulsating signal was preserved. If the standardized power spectrum exceeded 50% of the maximum value within a ten-minute window, the segment was classified as containing an artifact. The third rule examined the value of the second harmonic of the dominant frequency component. If this value was less than three times the minimum value of the power spectra calculated within the same window, the segment was identified as containing an artifact.

Segments that passed all three rules were determined to be artifact-free, while those that triggered any of the rules were classified as containing artifacts. We then evaluated the performance of this detection algorithm by calculating sensitivity and specificity based on the accurate identification of inserted artifacts.

10.2 Results

We inserted four types of artifacts—rectangular, fast impulse, saw tooth, and baseline drift—into ABP signals and evaluated the performance of an algorithm based on the STFT to detect these artifacts, see Table 10.2. The algorithm demonstrated high sensitivity in detecting rectangular (93.35%) and sawtooth (94.83%) artifacts, with specificity exceeding 99% for both. However, it showed low sensitivity for detecting baseline drift (5.02%), and it did not detect fast impulse artifacts. The false positive rate was 0.00% for all artifact types, durations and amplitudes.

Table 10.2: Results of detection for each type of simulated artifact

Artifact type	Sensitivity (%)	Specificity (%)
Rectangular	93.35	99.34
Saw tooth	94.83	99.14
Baseline drift	5.02	98.78
Fast impulse	0	99.82

10.3 Discussion

The algorithm's success in identifying rectangular and sawtooth artifacts demonstrated its effectiveness in detecting significant disturbances in the ABP signal. The use of STFT enabled the observation of changes in the frequency spectrum over time, making it a valuable tool for fast and reliable artifact detection. The simplicity of the STFT-based approach, compared to more complex machine learning methods, is particularly beneficial for real-time data analysis.

Although the algorithm struggled with fast impulse and baseline drift artifacts, this is not considered problematic. In our previous work [10], we found that the impact of artifacts on the PRx varies depending on their type, duration, and amplitude. The

fast impulse artifact had minimal impact because the PRx calculation involves a 10-second averaging process that acts as a low-pass filter, suppressing short artifacts. Similarly, baseline drift showed minimal effect, and its origin may not always be artificial. Therefore, we focused on detecting rectangular and sawtooth artifacts, achieving over 90% sensitivity and specificity. While the sensitivity for baseline drift was low, the specificity remained above 90%, ensuring accurate identification of artifact-containing signals.

We also observed that while the algorithm effectively detected artifacts with sharper edges, it was less sensitive to the middle parts of rectangular artifacts. This suggests a potential area for further improvement. Despite these limitations, the algorithm's high specificity ensures that artifact-free segments are reliably identified, which is crucial for maintaining the accuracy of secondary calculations like the pressure reactivity index. Further testing with real clinical data will be essential to refine and validate the algorithm's performance in practical applications.

11 Updated algorithm for detection of real artifacts

11.1 Methods

11.1.1 Detection of artifacts in ABP

After fine-tuning the artifact detection algorithm on simulated data, I proceeded to test it on data from real patient containing actual artifacts. I had access to a 9-day patient record with annotated artifacts, provided by colleagues at the Neurointensive Care Unit at Masaryk Hospital in Usti nad Labem, Czech Republic.

The first step involved modifying the algorithm and its functions to accommodate the different data formats. The simulated artifacts were stored in .CSV format, while the real data were in .HDF5 format. After adjusting the computational functions, I encountered several challenges, starting with the sheer volume of data being processed. Initially, when I ran the algorithm on the raw data, it did not complete within 30 minutes. Given that raw arterial pressure signals often contain tens of millions of values, efficient processing was crucial. To address this, I utilized the Vaex Python library, which optimizes the handling of large datasets. Additionally, I divided the original recording into 10-minute segments and processed them sequentially. These steps significantly accelerated the data processing.

Another major challenge was the discontinuity in the frequency spectrum at the start and end of each 10-minute analysis window. These discontinuities were consistently flagged as artifacts by the detection algorithm, which needed to be resolved. I addressed this by introducing a 1-minute overlap between consecutive analysis windows. This overlap allowed me to replace the points in the frequency spectrum that were affected by discontinuities with those from the subsequent window, resulting in a continuous frequency spectrum across the entire recording.

However, I encountered another optimization issue due to the large volume of data in the frequency spectrum post-STFT. Calculating the dominant frequency, the value of the second harmonic component, and the standardized power spectrum across different frequency ranges was time-consuming. To improve efficiency, I split the data into smaller, 5-minute segments with zero overlap and processed them sequentially.

I then applied the same three artifact detection rules described in Figure 10.4. To evaluate the algorithm's performance, I used the annotated artifact record within the HDF5 file to calculate the sensitivity and specificity. Since the primary goal of this algorithm is to improve the interpretation of PRx values, I ran the annotated artifact record through a 10-second averaging window (as in the PRx calculation) to filter out very short artifacts and those that do not affect PRx calculations. Given that the HDF5 file had hard-coded artifact times, I converted them to match the format of the detected artifacts to ensure consistent temporal resolution. After consulting with clinicians, I set the acceptable range of mean arterial pressure (MAP) changes to ±5 mmHg. If the absolute change in mean ABP was less than 5 mmHg and an artifact was annotated at that point, it was considered insignificant or too short.

Unfortunately, the algorithm's sensitivity was around 65% and specificity was around 96%, which was unsatisfactory. Despite adjusting threshold values and modifying detection conditions, the performance remained low. Consequently, I

decided to rework the algorithm and use a different detection approach, see Figure 11.1.

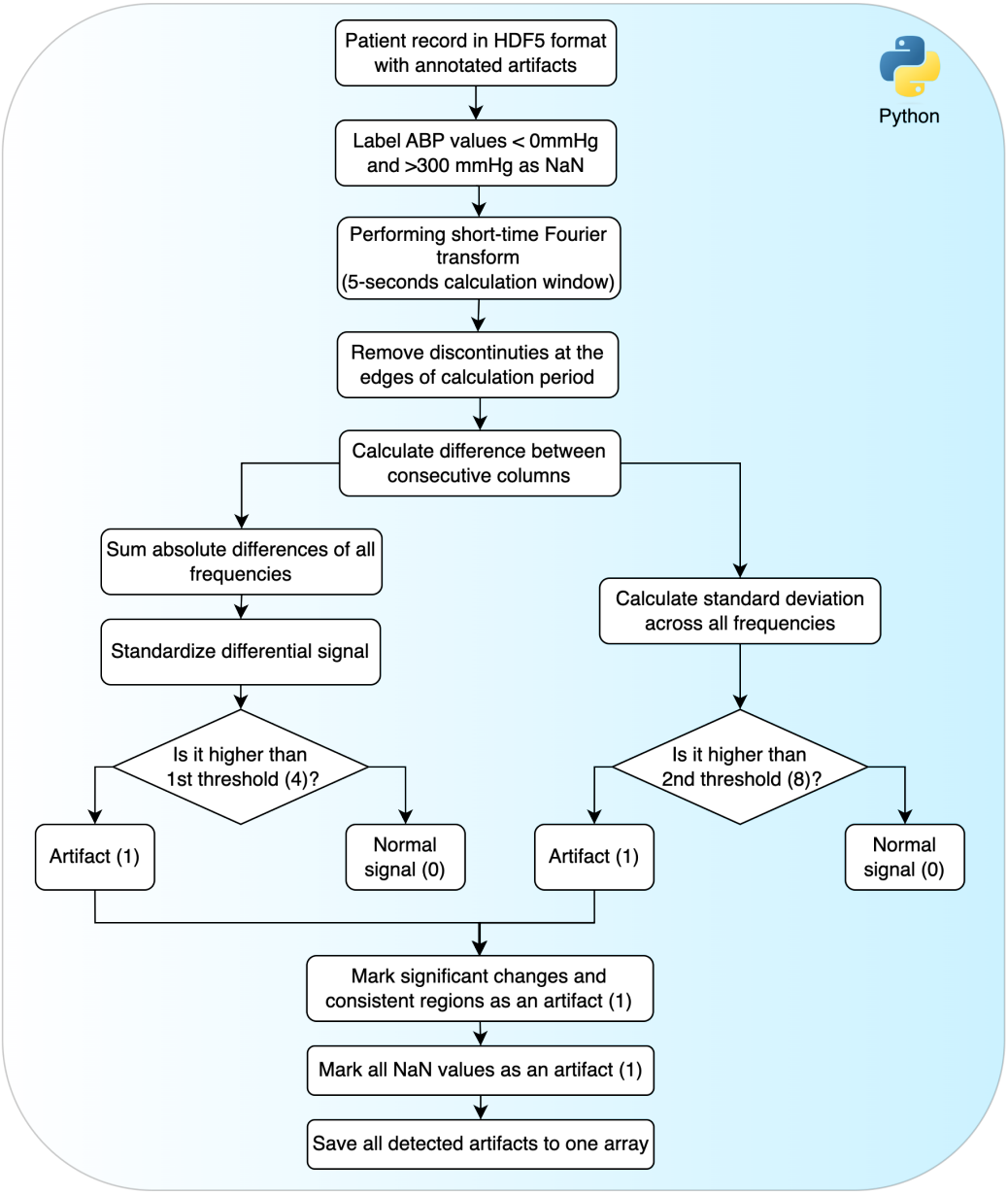


Figure 11.1: Flowchart illustrating the data processing and artifact identification steps in the new algorithm for real data

I shifted the algorithm's focus to detecting changes (differences) in the frequency spectrum. I retained all the steps from the original algorithm up to and including the calculation of the STFT. I then converted the STFT matrix to absolute values for easier processing and calculated the difference between adjacent columns (time steps). Next, I computed the sum of the absolute differences across all frequencies to obtain a single value for each time step. For future application on other patient records, I standardized this differential signal using a specific formula. I introduced a condition to identify significant changes in the frequency spectrum, which would account for the beginning and end of large artifacts as well as some short ones. Initially, I set a

threshold value of 5 for the standardized differential signal, marking points above this value as artifacts.

Upon testing the first condition, I realized that detection was still not optimal, as it mainly identified the beginning and end of large artifacts. To consistently detect spectral regions affected by artifacts, I added a standard deviation calculation across all frequencies, yielding one value per time step. I set an initial threshold of 7 for detecting consistent spectral regions; if the standard deviation exceeded this value, it indicated an artifact at that time point.

I developed a specialized function to determine the optimal threshold parameters, which I subsequently used as criteria for detecting artifacts. This function allowed me to systematically test various threshold values within a relevant range. I then detected artifacts based on these thresholds and computed the sensitivity and specificity against the artifacts annotated by the clinician colleague. A performance metric, derived from the sensitivity and specificity, was calculated to assess how closely these values approached 100%. The threshold values that minimized this metric (indicating the closest approach of sensitivity and specificity to 100%) were selected for the final version artifact detection algorithm. In the end, the threshold for evaluating the differential spectrum signal was set to 2, and the threshold for evaluating consistent spectral regions was set to 8.

In the final step, I flagged all NaN values (corresponding to signal dropouts) and values outside the 0-300 mmHg range as artifacts. This resulted in a record of all detected artifacts (arrays of zeros and ones) matching the time range of the STFT spectrum.

To assess the algorithm's effectiveness, I recalculated the sensitivity and specificity using the annotated artifacts averaged over a 10-second window.

11.1.2 Detection of artifacts in ICP

After successfully tuning the algorithm on the ABP signal, we decided to apply it to the ICP signal, assuming that the frequency spectrum of these signals would be similar, with only the amplitude differing.

To evaluate the algorithm's efficiency on the ICP signal, I used an annotation file from our clinical specialists that thoroughly marked the artifacts in the ICP signal. I analyzed a 66-hour patient record containing these annotations. Since the detection algorithm was primarily designed to improve the interpretation of PRx values, I processed the annotated artifact data using a 10-second averaging window, as done with the ABP signal, to filter out short-duration artifacts that would not significantly impact PRx index calculations. The threshold for allowable changes in ICP artifacts was set at 5 mmHg. If the absolute change in mean ICP at a given time was less than 5 mmHg and an artifact was annotated, the artifact was considered insignificant.

The algorithm applied to the ICP signal was identical to the one used for the ABP signal, with adjustments only to the thresholds to optimize detection capability, sensitivity, and specificity. The same optimization function used for ABP was applied to fine-tune these thresholds. In the end, the threshold for evaluating the differential spectrum signal was set to 4, and the threshold for evaluating consistent spectral regions was set to 8.

11.2 Results

The artifact detection algorithm, after undergoing significant fine-tuning and testing on real patient data, demonstrated varying levels of success in detecting artifacts in both ABP and ICP signals.

11.2.1 Detection of artifacts in ABP

The algorithm was first applied to ABP data, using a 9-day patient record with annotated artifacts provided by colleagues from the Neurointensive Care Unit at Masaryk Hospital. Initially, the algorithm's performance was suboptimal, with a sensitivity of approximately 65% and a specificity of around 96%. Despite efforts to adjust threshold values and detection conditions, these results were not satisfactory. Consequently, the detection approach was reworked to focus on changes in the frequency spectrum.

After reworking the algorithm, sensitivity and specificity were reassessed. The final version of the algorithm, which included the calculation of differential spectrum signals and consistent spectral regions, achieved a notable improvement. The threshold for evaluating the differential spectrum signal was set to 2, and the threshold for evaluating consistent spectral regions was set to 8. The recalculated sensitivity was 92% and specificity was 90%. The figures below (Figures 11.2-11.3) show examples of raw and averaged ABP signals, annotated artifact signals averaged using a 10-second window, and artifacts detected by our algorithm.

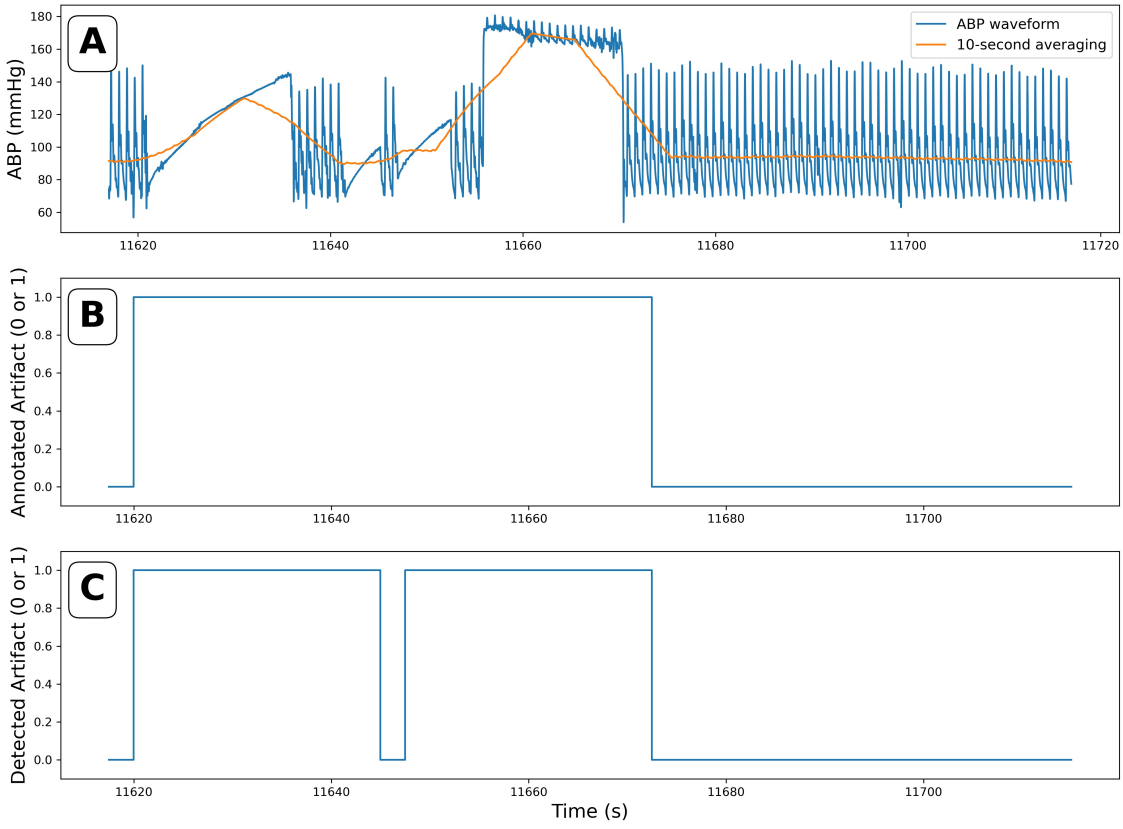


Figure 11.2: An example of complex artifact in real signal: A) ABP signal with 10-second averaging B) Annotated averaged artifact C) Detected artifact by the algorithm.

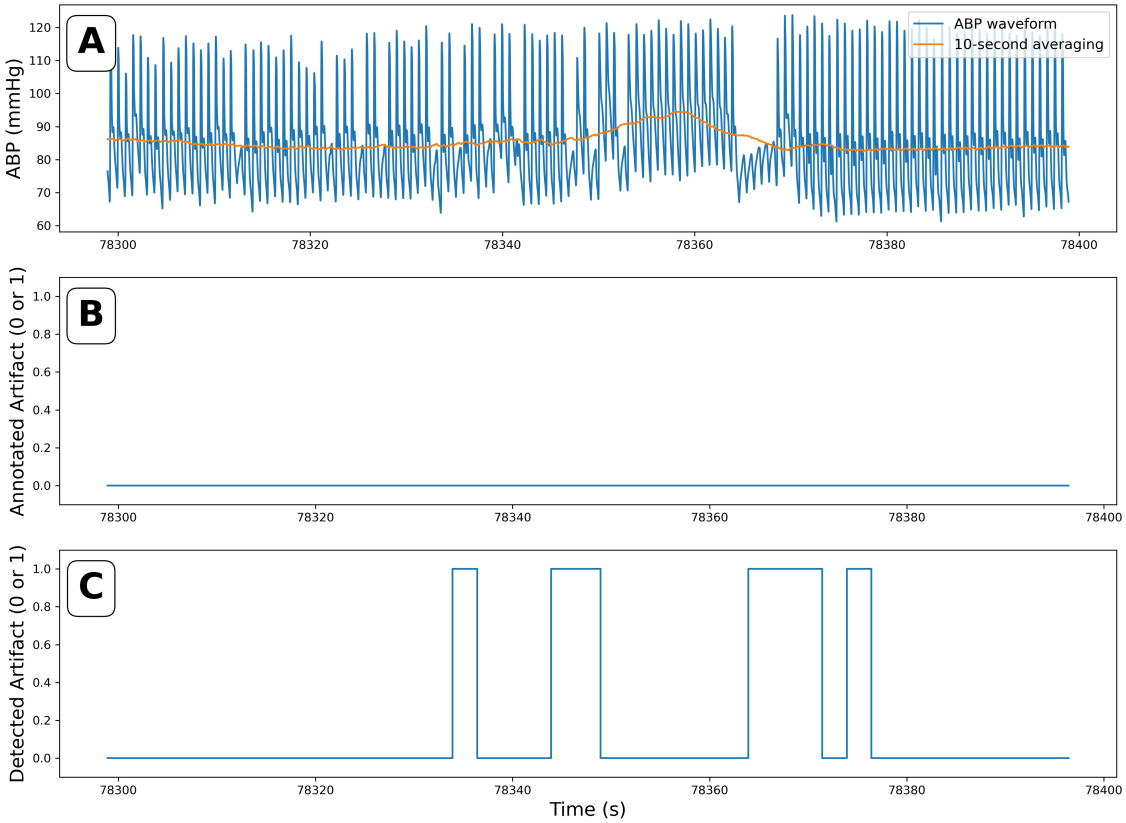


Figure 11.3: An example of insignificant artifact in real signal: A) ABP signal with 10-second averaging B) Annotated averaged artifact C) Detected artifact by the algorithm.

11.2.2 Detection of artifacts in ICP

The artifact detection algorithm was adapted and tested on a 66-hour patient record with detailed artifact annotations for the intracranial pressure signal. After fine-tuning the thresholds specifically for the ICP data, the sensitivity and specificity of the algorithm were both near 80%. The differential spectrum signal threshold was set at 4, while the threshold for consistent spectral regions was maintained at 8. This performance demonstrates that the algorithm effectively detected clinically significant artifacts in the ICP signal, although with slightly reduced sensitivity compared to the ABP signal. These results underscore the algorithm's ability to identify relevant artifacts that could impact PRx calculations, filtering out those of lesser clinical importance.

11.3 Discussion

The updated algorithm for artifact detection in arterial blood pressure and intracranial pressure signals has shown very promising results. The recalculated sensitivity for ABP artifact detection reached 92%, with a specificity of 90%. These metrics indicate that the algorithm is now highly effective in identifying significant artifacts in real ABP signal that could impact clinical decision-making, particularly in the quality evaluation of the PRx index.

The algorithm has been refined to focus on changes in the frequency spectrum, with carefully determined threshold values that optimize its performance. As more

patient records with annotated data become available, these threshold values can be adjusted using the optimization function, allowing for continuous improvement in the algorithm's sensitivity and specificity. This adaptability ensures that the algorithm remains effective and can be further updated to enhance its performance as additional data and clinical insights will be available.

The detection of artifacts in the ICP signal, although incorporated into the study, was a secondary goal. This is because the ICP signal is inherently less prone to interference and artifacts compared to the arterial blood pressure signal. The nature of the ICP signal, combined with the effective 10-second averaging process, resulted in the elimination of many potential artifacts, reducing the overall need for a highly sensitive detection algorithm. Figure 11.4 illustrates this point by showing how many artifacts in the ICP signal disappear due to the 10-second averaging process [40].

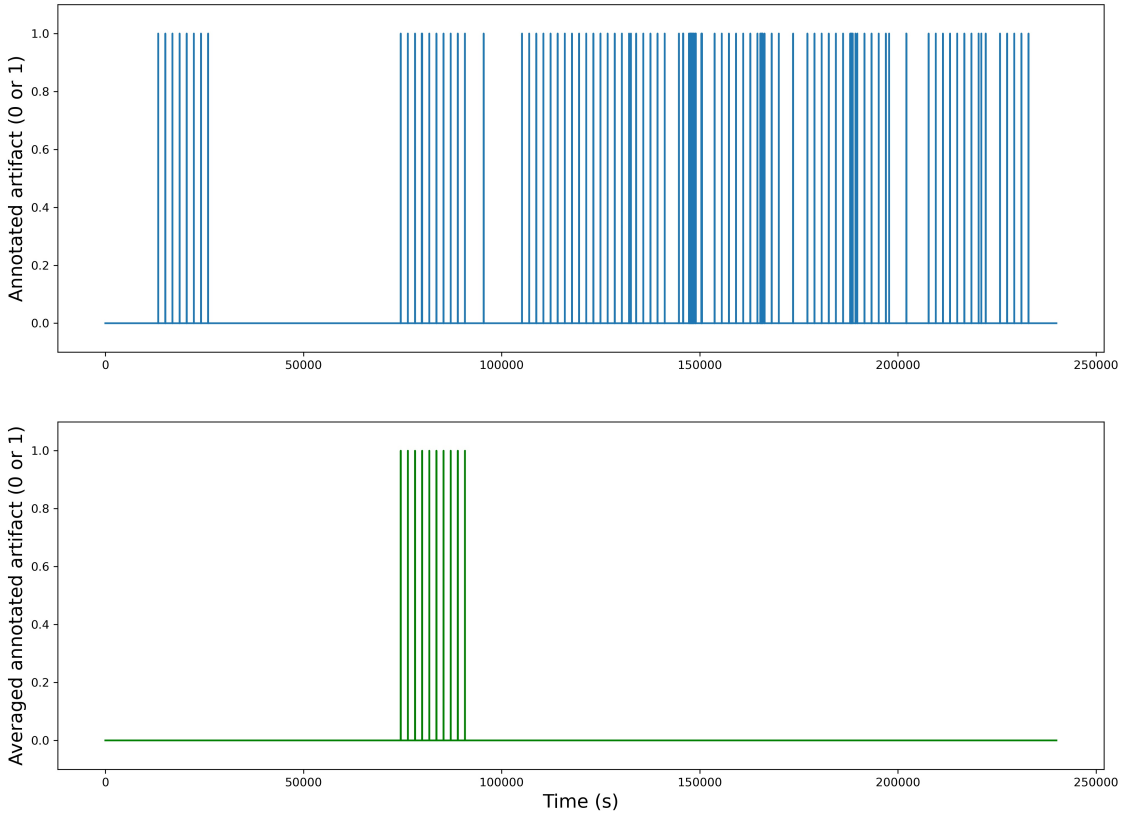


Figure 11.4: Comparison of original annotated artifacts in ICP with artifacts after 10-second averaging

The sensitivity and specificity of approximately 80% achieved in detecting ICP artifacts indicate that the algorithm performs well in identifying significant disruptions in the signal. However, given that many artifacts were already smoothed out or removed through averaging, the necessity for such rigorous detection is less critical in the context of ICP monitoring.

Additionally, it is important to note that the primary focus of artifact detection remains on the ABP signal. This is because any part of the ABP signal that is interfered with or contains artifacts becomes unusable for calculating the PRx index. Therefore,

ensuring the accuracy and cleanliness of the ABP signal is primary, as it directly impacts the reliability of PRx calculations.

Furthermore, it has to be noted that the clinical specialists carefully annotated artifacts, ensuring that they identified everything that, in their expert opinion, met the criteria for an artifact. Since manual annotation is an extremely time-consuming process, and because the annotated data was intended for multiple projects, including the detection of artifacts in ABP for PRx index calculations, the specialists approached the task with a high level of precision and thoroughness. After consulting with the clinical specialists, we set a threshold of ±5 mmHg for the 10-second averaging filter to refine the annotated artifact data. This threshold was designed to filter out short and insignificant artifacts, ensuring that only clinically significant changes in ABP were included in the annotations and used for the statistical evaluation of the detection algorithm.

The main limitation of my detection algorithm is that its performance has been validated on only a single patient record. To optimize the detection thresholds, additional annotated data from various patients would be necessary. As I previously mentioned, annotating artifacts is extremely time-consuming, and clinical specialists have limited time for this task due to other more important responsibilities, such as patient care. However, the detection algorithm is now at a stage where, if new annotated data were available, recalculating thresholds would be straightforward. This would likely result in more universal thresholds with potentially higher sensitivity and specificity across different patients.

12 Python plug-in and PRx reliability index

12.1 Methods

After fine-tuning the detection algorithm on ABP signals, I implemented it within the ICM+ software, which supports custom Python functions for advanced statistical analysis and reporting on both recorded and real-time data.

A primary objective of my work was to develop a "traffic light" system for evaluating quality of PRx indices, which we named the "PRx reliability index." This index indicates the percentage of artifacts within a given time interval, allowing physicians to assess the reliability of the PRx index at that moment.

To calculate the PRx reliability index, accurate artifact detection is essential. I adapted my algorithm to meet the requirements for Python functions within ICM+. This involved creating a Python class that accepts the arterial pressure signal, sampling rate, and thresholds as input, and outputs the percentage of artifacts over a specified time interval. The "traffic light" display is achieved by plotting the signal on the Risk Chart in ICM+, where the chart colors change based on set thresholds.

The prerequisites for using this plug-in in ICM+ include having a 32-bit version of Python installed on the same machine, enabling the use of custom Python functions in the ICM+ software license (via a special Python module), and installing essential Python libraries for data processing (numpy, scipy, stattools). The development environment included ICM+ (version 9.2.4.6, Cambridge Enterprise Ltd., Cambridge, UK), the Python module (version 1.1, Cambridge Enterprise Ltd., Cambridge, UK), and Python IDE (version 3.7, 32-bit, Python Software Foundation, Wilmington, Delaware, USA).

Once all prerequisites were met, I proceeded to create the Python plug-in. ICM+ offers a user-friendly interface for Python script creation, allowing the insertion of custom functions into a predefined template. In this template, I defined the function name, description of my function, number of input signals (e.g., ABP, ICP), and additional function parameters. I set the detection algorithm thresholds as adjustable parameters so users can modify them without altering the Python code. If unchanged, the thresholds default to values I optimized during algorithm tuning.

After setting the necessary parameters, ICM+ generated a Skeleton Python script (see Figure 12.1) and a Python plugin configuration file (see Figure 12.2). I did not alter the generated configuration file. Instead, I integrated my artifact detection and PRx reliability index calculation code into the Skeleton script, following the guidelines provided on the ICM+ website.

```python
import numpy as np
import scipy.signal as signal
import stattools as st

class ArtifactPercent:

    # DO NOT MODIFY this method
    # It will provide backward compatibility wth the older ICM+--Python interface.
    def set_parameter(self, param_name, param_value):
        setattr(self, param_name, param_value)

    # You can append your own code to the constructor, if needed.
    # You should not set here values of parameters declared in your XML
    # configuration file because ICM+ will do it for you.
    # You will have to add your own code, only if you need to initialise some
    # extra data structures which were not declared in the XML config file.
    def __init__(self):
        self.variables = []
        self.sampling_freq = None
        self.file_path = None
        self.threshold1 = 2       # threshold1
        self.threshold2 = 8       # threshold1

    # You can append your own code to the destructor but most likely you will not need it
    def __del__(self):
        pass

    # 'calculate' is the main work-horse function.
    # It is called with a data buffer (one or more) of size corresponding to the Calculation Window
    # It must return one floating-point number
    # It takes the following parameters:
    # sig1 - input variable/signal 1
    # ts_time - part of the data time stamp - number of milliseconds since midnight
    # ts_date - Part of the data time stamp - One plus number of days since 1/1/0001
    # Use the class member self.sampling_freq to calculate the time vector
    # Note: the class member 'self.variables' includes indices of the used variables
    #       These can be used to check if the function has already been called with identical parameters
    #       in order to avoid unnecessery re-calculations
    # Note 2: The self.variables variable and the ts_time and ts_date parameters to the calculate call
    #       are at present only used in the on-line calculation engine, from Primary Analysis onwards.
    #       The will be empty when used in the Virtual Signals part of that engine, nor in the other
    #       off-line tools lke ScriptLab, SignalCalculator or CustomStatistics
    # Note 3: If the function expects to return a single value but a vector is returned, a mean value
    #       will be automatically added on top;  If on the other hand a vector is expected, but a single
    #       value is returned, a vector will be automatically created filled in with that value.
    def calculate(self, sig1, ts_time, ts_date):
        sig1 = np.array(sig1)
        # my_own_code_here
        result = 0.0
```

Figure 12.1: Skeleton Python script.

```
▼<PyToICMPlusConfig>
  ▼<Function Name="ArtifactPercent" Type="Stats" SignalsCount="1">
     <GUID>{[redacted]}</GUID>
     <Description>This function detect artifacts in ABP or ICP signal and returns the percent of
     artifacts in time interval.</Description>
    ▼<Parameter ShortName="threshold1" IsMandatory="True">
       <Caption>threshold1</Caption>
       <Description>threshold for evaluating the differential spectrum signal</Description>
       <Type Name="Float" Min="0" Max="0" DefaultValue="2"/>
     </Parameter>
    ▼<Parameter ShortName="threshold2" IsMandatory="True">
       <Caption>threshold2</Caption>
       <Description>threshold for evaluating consistent spectral regions</Description>
       <Type Name="Float" Min="0" Max="0" DefaultValue="8"/>
     </Parameter>
   </Function>
 </PyToICMPlusConfig>
```

Figure 12.2: Python plugin configuration file.

To adapt my algorithm for use within ICM+, I made several modifications, including omitting optimization steps designed for large dataset processing, which is unnecessary for ICM+ data analysis. When using any function (whether built-in or

custom) to analyze data, ICM+ requires setting a computation window length and a window offset. For testing my Python function, I set the computation window length to 300 seconds and the window offset to 60 seconds, aligning with the default settings for PRx index computation in ICM+.

I also removed functions related to optimizing thresholds, processing annotated artifacts, calculating sensitivity and specificity, and handling discontinuities. To address discontinuities within the ICM+ environment, I marked the first and last points of the detection matrix as 0 after detecting artifacts, as discontinuities typically affect only the edges of the analyzed signal segment.

Most steps described in Section 11.1.1 were incorporated into the Python plug-in for ICM+. Once the detection matrix was generated, containing only zeros and ones, I calculated the percentage of ones relative to the total number of points.

I further refined the Python plug-in using the ScriptLab tab, which allows for processing real signals with custom Python functions. This tab is particularly useful for identifying error messages and warnings during code execution, enabling direct edits within the Python IDE.

Finally, I tested the Python plug-in on a retrospective anonymized record in ICM+. I selected a file from our anonymized records library, re-analyzed the raw data using the Calculations tab, limited the allowable ABP values to a range of 30 to 300 mmHg, and applied the custom Python function to the ABP signal (see Figure 12.3). The results were plotted as a trend curve. For additional clarity, I added a Risk Chart, where the chart color changes according to the PRx reliability index values: a zero value corresponds to green, and a 50% value corresponds to red.

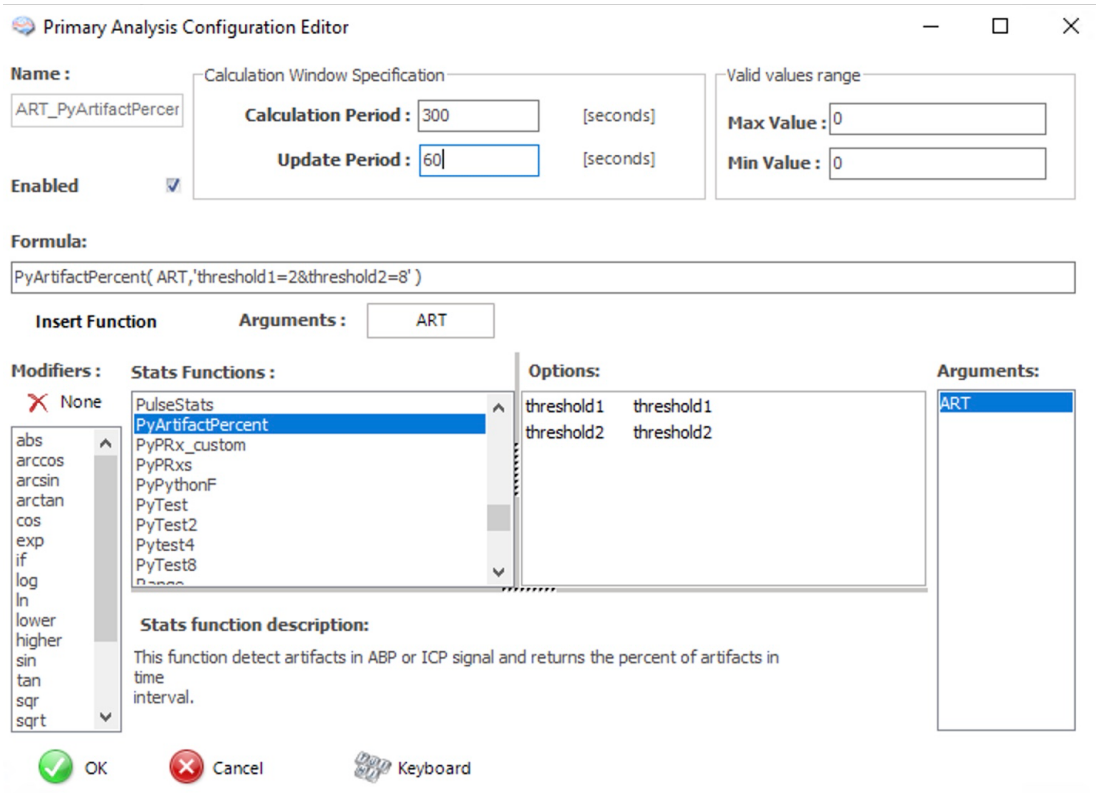


Figure 12.3: Python plug-in setting

12.2 Results

In this section, I developed a Python plug-in for the ICM+ software based on the algorithm I created for artifact detection in ABP signals. The input data for the algorithm includes the ABP signal, the threshold for the standardized differential spectrum signal, and the threshold for consistent spectral regions. The output of the Python plug-in is the PRx reliability index (see Figure 12.4), which represents the percentage of artifacts detected within a specified time interval. The final Python code and the XML configuration file for the plugin are provided in Appendix C.

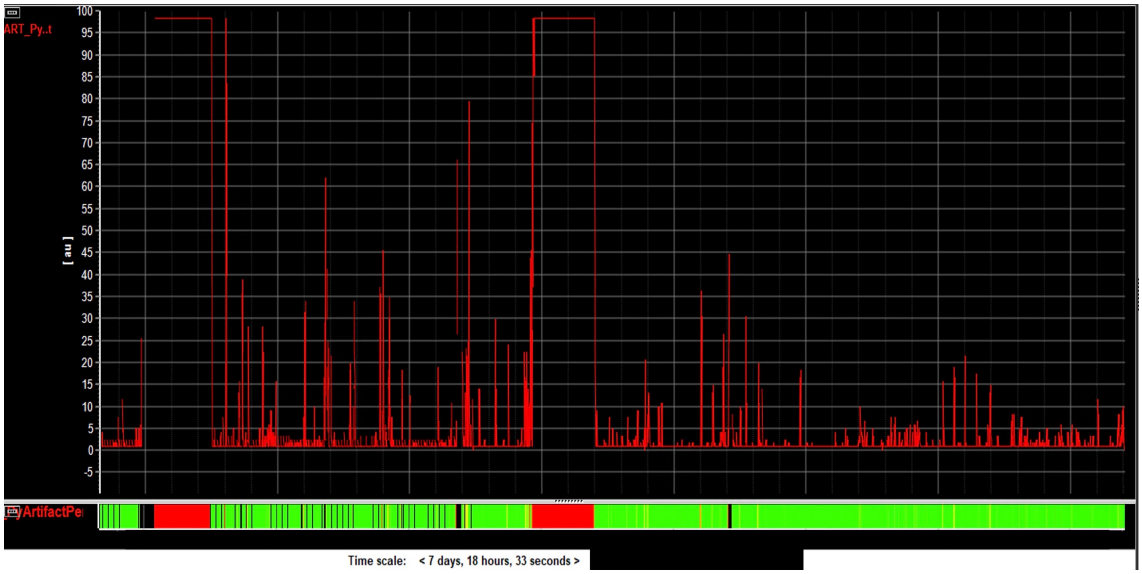


Figure 12.4: PRx reliability index in the ICM+ software

To streamline processing and use of the Python plug-in, I saved an ICM+ configuration profile that can be applied to any patient record with a raw ABP signal. This profile includes the settings for calculating the PRx reliability index and displaying it on a Risk Chart, with the upper limit set to 50%.

12.3 Discussion

The development and integration of the Python plug-in for artifact detection in ABP signals within the ICM+ software represent a significant advancement in real-time and offline data analysis capabilities. This plug-in, based on a finely tuned detection algorithm, offers a robust tool for evaluating the quality of PRx indices, particularly through the creation of the PRx reliability index.

One of the key features of this plug-in is its adaptability. The thresholds for artifact detection, as well as the computation window length, can be easily modified by users without the need to alter the underlying Python or XML code. This flexibility ensures that the plug-in can be applied to data from different patients and adapted to varying clinical scenarios. As more patient records with annotated data become available, the algorithm's thresholds can be fine-tuned using the built-in optimization functions, further enhancing its accuracy and reliability.

The PRx reliability index, displayed through a "traffic light" system on the Risk Chart in ICM+, provides a clear visual representation of the percentage of artifacts detected within a given time interval, typically 5 minutes. This index is crucial for clinicians as it helps them assess the reliability of the PRx index at any given moment. By setting a high limit of 50% on the Risk Chart, we align with the same threshold used for artifact calculation in the PRx index within the ICM+ software, ensuring consistency across different analyses.

A significant advantage of this system is that it can be used for both online and offline data analysis. Whether analyzing real-time patient data or reviewing recorded signals, the plug-in provides fast feedback on the presence of artifacts, enhancing the decision-making process.

The PRx reliability index, displayed through a "traffic light" system on the Risk Chart in ICM+, provides a clear visual representation of the percentage of artifacts detected within a given time interval, typically 5 minutes. This index is crucial for clinicians as it helps them assess the reliability of the PRx index at any given moment by showing the percentage of artifacts in real-time. By setting a high limit of 50% on the Risk Chart, we align with the typical missing value limit set within the ICM+ software for PRx calculations, ensuring consistency across different analyses. Although the ICM+ software accounts for a certain percentage of missing values in PRx calculations, it does not visibly display the percentage of artifacts or specify their nature. By integrating this plug-in, clinicians gain a more comprehensive understanding of the data's reliability, offering insights that are not immediately apparent in standard PRx calculations.

During the development of the Python plug-in and the implementation of the PRx reliability index, I encountered several technical issues. The most significant problem was that the Python function I initially created wasn't communicating with ICM+ at all, and error messages related to memory allocation were being generated. At first, I suspected the issue was due to an incompatible version of the Python IDE with ICM+. I tried multiple different versions of Python, but none resolved the problem. I even modified the internals of my Python function in various ways and eventually simplified it to the point where it only output a single number, but it still didn't work. Finally, I discovered that the issue wasn't with my function or the Python IDE, but rather with the incompatibility between the versions of the Python module and the ICM+ software. After updating ICM+ to the latest version, my Python plug-in started working correctly.

The Python plug-in I developed also has some limitations. The primary limitation, as discussed in Chapter 11, is that the detection algorithm on which the plug-in is based has been validated and tuned using annotated data from only a single patient. To optimize the detection thresholds, additional annotated data from multiple patients would be needed. Another limitation is that, because I do not have access to the source code of ICM+ or insight into how it handles and executes my function, I cannot guarantee 100% functionality or ensure that the results will be the same as when using the same functions in a standard Python IDE.

13 Conclusions

In this work, I explored artifact detection in arterial blood pressure and intracranial pressure signals, as well as how these artifacts affect the PRx index. The main achievement was the development of an algorithm for detecting artifacts in real ABP signals. This algorithm was tested on a 9-day patient dataset, achieving sensitivity and specificity greater than 90%. Additionally, I created a Python plug-in for the ICM+ software based on this algorithm, which calculates a PRx reliability index. This index could significantly improve the accuracy of interpreting the quality of the PRx index.

I began by analyzing HDF5 files containing ABP and ICP signals from the ICM+ software, addressing various issues with these files. I wrote a pseudonymization script to remove personal identifiers and shift the timeline, ensuring the data was protected against unauthorized re-identification. I also investigated the impact of downsampling on the frequency spectrum due to automatic downsampling when exporting to HDF5 files. Since we work with ABP and ICP signals, I found that downsampling did not significantly affect the frequency spectrum.

Next, I examined how artifacts in ABP and ICP signals influence the circadian variability of the PRx index. I analyzed data from 19 patients without labeled artifacts and found that PRx variability was higher during certain morning and evening hours (see Figure 3.1). However, without more labeled artifacts (annotated patient records), we cannot conclusively attribute these variations to artifacts.

I then simulated various types of stereotypical artifacts to help develop and test the artifact detection algorithm. We also studied the effect of these simulated artifacts on the PRx index to determine which artifacts should be detected and which could be ignored. Additionally, we assessed the impact of high-frequency noise on PRx and confirmed that it has an insignificant effect, as it is filtered out by the 10-second averaging process during PRx calculation.

To detect artifacts, I developed an algorithm using short-time Fourier transform (STFT). Initially, I focused on selecting the optimal calculation window and chose a 5-second window, which was sufficient for detecting changes in the frequency spectrum caused by both complex artifacts and short interferences.

I then worked on an efficient method to evaluate changes in the frequency spectrum for artifact detection. The initial version of the algorithm showed good results on rectangular and sawtooth artifacts, achieving sensitivity and specificity above 93%. However, it was less effective at detecting baseline drift artifacts and fast impulses. Since these artifacts do not significantly affect the PRx index, and clinicians often disagreed on their artificial nature, this was not a major limitation.

The next step was to validate the algorithm using real patient data. The original algorithm, with a sensitivity of around 65%, was less effective on real ABP data, so I developed and tested a new version that improved sensitivity and specificity to over 90%. Artifact detection in the ICP signal was a secondary objective because ICP is less prone to interference compared to ABP. Additionally, the 10-second averaging process effectively filtered out many minor artifacts, reducing the need for a highly sensitive detection algorithm. Nevertheless, the algorithm performed well, achieving about 80% sensitivity and specificity for detecting significant disruptions in ICP. Although the detection algorithm is not universal and targets only clinically significant artifacts, I believe it is sufficient for assessing the quality of PRx indices.

Finally, I integrated the detection algorithm into a Python plug-in for the ICM+ software. The plug-in outputs a PRx reliability index, helping to assess the quality of PRx indices. It can help analyze offline patient records and potentially be used for real-time analysis.

This Python plug-in and the detection algorithm can be further adapted and improved as more annotated patient data becomes available. The algorithm's thresholds can be fine-tuned using built-in optimization functions, further enhancing its accuracy and reliability.

14 List of References

[1] Tas J, Czosnyka M, Van Der Horst ICC, et al. Cerebral multimodality monitoring in adult neurocritical care patients with acute brain injury: A narrative review. *Front Physiol* 2022; 13: 1071161.

[2] Abecasis F, Dias C, Zakrzewska A, et al. Monitoring cerebrovascular reactivity in pediatric traumatic brain injury: comparison of three methods. *Childs Nerv Syst* 2021; 37: 3057–3065.

[3] Klein SP, Depreitere B, Meyfroidt G. How I monitor cerebral autoregulation. *Crit Care* 2019; 23: 160, s13054-019-2454–1.

[4] Gomez A, Froese L, Bergmann TJG, et al. Non-Invasive Estimation of Intracranial Pressure-Derived Cerebrovascular Reactivity Using Near-Infrared Spectroscopy Sensor Technology in Acute Neural Injury: A Time-Series Analysis. *Sensors* 2024; 24: 499.

[5] Aries MJH, Czosnyka M, Budohoski KP, et al. Continuous determination of optimal cerebral perfusion pressure in traumatic brain injury*: *Critical Care Medicine* 2012; 40: 2456–2463.

[6] Czosnyka M, Smielewski P, Kirkpatrick P, et al. Continuous Assessment of the Cerebral Vasomotor Reactivity in Head Injury. *Neurosurgery* 1997; 41: 11–19.

[7] Czosnyka M, Piechnik S, Richards HK, et al. Contribution of mathematical modelling to the interpretation of bedside tests of cerebrovascular autoregulation. *Journal of Neurology, Neurosurgery & Psychiatry* 1997; 63: 721–731.

[8] Sorrentino E, Diedler J, Kasprowicz M, et al. Critical Thresholds for Cerebrovascular Reactivity After Traumatic Brain Injury. *Neurocrit Care* 2012; 16: 258–266.

[9] Steiner LA, Czosnyka M, Piechnik SK, et al. Continuous monitoring of cerebrovascular pressure reactivity allows determination of optimal cerebral perfusion pressure in patients with traumatic brain injury: *Critical Care Medicine* 2002; 30: 733–738.

[10] Rozanek M, Skola J, Horakova L, et al. Effect of artifacts upon the pressure reactivity index. *Sci Rep* 2022; 12: 15131.

[11] Penhaker M. *Lékařské diagnostické přístroje: učební texty*. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2004.

[12] Son Y, Lee S-B, Kim H, et al. Automated artifact elimination of physiological signals using a deep belief network: An application for continuously measured arterial blood pressure waveforms. *Information Sciences* 2018; 456: 145–158.

[13] Pasma W, Wesselink EM, van Buuren S, et al. Artifacts annotations in anesthesia blood pressure data by man and machine. *J Clin Monit Comput* 2021; 35: 259–267.

[14] Lee S-B, Kim H, Kim Y-T, et al. Artifact removal from neurophysiological signals: impact on intracranial and arterial pressure monitoring in traumatic brain injury. *Journal of Neurosurgery* 2020; 132: 1952–1960.

[15] Rinehart J, Tang J, Nam J, et al. Detection of arterial pressure waveform error using machine learning trained algorithms. *J Clin Monit Comput* 2022; 36: 227–237.

[16] Li Q, Mark RG, Clifford GD. Artificial arterial blood pressure artifact models and an evaluation of a robust blood pressure and heart rate estimator. *BioMed Eng OnLine* 2009; 8: 13.

[17] Khan JM, Maslove DM, Boyd JG. Optimized Arterial Line Artifact Identification Algorithm Cleans High-Frequency Arterial Line Data With High Accuracy in Critically Ill Patients. *Critical Care Explorations* 2022; 4: e0814.

[18] Imhoff M, Bauer M, Gather U, et al. Statistical pattern detection in univariate time series of intensive care on-line monitoring data. *Intensive Care Medicine* 1998; 24: 1305–1314.

[19] Du CH, Glick D, Tung A. Error-checking intraoperative arterial line blood pressures. *J Clin Monit Comput* 2019; 33: 407–412.

[20] Yücelbaş C, Yücelbaş Ş, Özşen S, et al. Automatic detection of sleep spindles with the use of STFT, EMD and DWT methods. *Neural Comput & Applic* 2018; 29: 17–33.

[21] Taherisadr M, Dehzangi O, Parsaei H. Single Channel EEG Artifact Identification Using Two-Dimensional Multi-Resolution Analysis. *Sensors* 2017; 17: 2895.

[22] Chen J, Sun K, Sun Y, et al. Signal Quality Assessment of PPG Signals using STFT Time-Frequency Spectra and Deep Learning Approaches. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. Mexico: IEEE, pp. 1153–1156.

[23] Mohylová J, Krajča V. *Zpracování biosignálů*. Ostrava: Vysoká škola báňská - Technická univerzita, 2008.

[24] Escabí MA. BIOSIGNAL PROCESSING. In: *Introduction to Biomedical Engineering*. Elsevier, pp. 549–625.

[25] Klabunde RE. Arterial Blood Pressure: Cardiovascular physiology concepts., https://www.cvphysiology.com/Blood%20Pressure/BP002 (2016).

[26] Rahaman U, Dr, Senior. Invasive Blood Pressure Monitoring. 2013, pp. 1–21.

[27] Thomas E, Czosnyka M, Hutchinson P. Calculation of cerebral perfusion pressure in the management of traumatic brain injury: joint position statement by the councils of the Neuroanaesthesia and Critical Care Society of Great Britain and Ireland (NACCS) and the Society of British Neurological Surgeons (SBNS). *British Journal of Anaesthesia* 2015; 115: 487–488.

[28] Ševčík P, Matějovič M. *Intenzivní medicína*. 3., přeprac. a rozš. vyd. Praha: Galén, 2014.

[29] Kawoos U, McCarron R, Auker C, et al. Advances in Intracranial Pressure Monitoring and Its Significance in Managing Traumatic Brain Injury. *IJMS* 2015; 16: 28979–28997.

[30] Monro–Kellie doctrine: Neurosurgical and Neurological Emergencies for Surgeons, https://basicmedicalkey.com/neurosurgical-and-neurological-emergencies-for- surgeons/ (2016).

[31] Carney N, Totten AM, O'Reilly C, et al. Guidelines for the Management of Severe Traumatic Brain Injury, Fourth Edition. *NEUROSURGERY* 2017; 80: 6–15.

[32] Dias C, Silva MJ, Pereira E, et al. Optimal Cerebral Perfusion Pressure Management at Bedside: A Single-Center Pilot Study. *Neurocrit Care* 2015; 23: 92–102.

[33] Tas J, Beqiri E, Van Kaam RC, et al. Targeting Autoregulation-Guided Cerebral Perfusion Pressure after Traumatic Brain Injury (COGiTATE): A Feasibility Randomized Controlled Clinical Trial. *Journal of Neurotrauma* 2021; 38: 2790–2800.

[34] Kooi EMW, Richter AE. Cerebral Autoregulation in Sick Infants. *Clinics in Perinatology* 2020; 47: 449–467.

[35] Trukhan V, Horakova L, Skola J, et al. Effect of Pressure Reactivity Index Calculation Settings on the Range of the Optimal Cerebral Perfusion Pressure. In: *2022 E-Health and Bioengineering Conference (EHB)*. Iasi, Romania: IEEE, pp. 1–4.

[36] Czosnyka M, Czosnyka Z, Smielewski P. Pressure reactivity index: journey through the past 20 years. *Acta Neurochir* 2017; 159: 2063–2065.

[37] Littmann L. Electrocardiographic artifact. *Journal of Electrocardiology* 2021; 64: 23–29.

[38] Islam MK, Rastegarnia A, Yang Z. Methods for artifact detection and removal from scalp EEG: A review. *Neurophysiologie Clinique/Clinical Neurophysiology* 2016; 46: 287–305.

[39] Feng M, Loy LY, Zhang F, et al. Artifact removal for intracranial pressure monitoring signals: A robust solution with signal decomposition. In: *2011 Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society. Boston, MA: IEEE, pp. 797–801.

[40] Zong W, Moody GB, Mark RG. Reduction of false arterial blood pressure alarms using signal quality assessement and relationships between the electrocardiogram and arterial blood pressure. *Med Biol Eng Comput* 2004; 42: 698–706.

[41] J. X. Sun, A. T. Reisner, R. G. Mark. A signal abnormality index for arterial blood pressure waveforms. In: *2006 Computers in Cardiology*. 2006, pp. 13–16.

[42] Zhang P, Liu J, Wu X, et al. A Novel Feature Extraction Method for Signal Quality Assessment of Arterial Blood Pressure for Monitoring Cerebral Autoregulation. In: *2010 4th International Conference on Bioinformatics and Biomedical Engineering*. Chengdu, China: IEEE, pp. 1–4.

[43] Cao H, Norris P, Ozdas A, et al. A Simple Non-physiological Artifact Filter for Invasive Arterial Blood Pressure Monitoring: a Study of 1852 Trauma ICU Patients. In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. New York, NY: IEEE, pp. 1417–1420.

[44] Choi HS, Park HD, Lee KJ. Motion Artifact Reduction in Blood Pressure Signals Using Adaptive Digital Filter with a Capacitive Sensor. In: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Lyon, France: IEEE, pp. 3285–3287.

[45] Feng M, Loy LY, Sim K, et al. Artifact correction with robust statistics for non-stationary intracranial pressure signal monitoring. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 557–560.

[46] Prabhu KMM. *Window functions and their applications in signal processing*. Boca Raton, [Florida]: CRC Press/Taylor & Francis, 2014.

[47] ICM+ homepage: Cambridge Enterprise ICM+, https://icmplus.neurosurg.cam.ac.uk.

[48] Smielewski P, Lavinio A, Timofeev I, et al. ICM+, a flexible platform for investigations of cerebrospinal dynamics in clinical practice. In: Steiger H-J (ed) *Acta Neurochirurgica Supplements*. Vienna: Springer Vienna, pp. 145–151.

[49] Cabeleira M, Ercole A, Smielewski P. HDF5-Based Data Format for Archiving Complex Neuro-monitoring Data in Traumatic Brain Injury Patients. In: Heldt T (ed) *Intracranial Pressure & Neuromonitoring XVI*. Cham: Springer International Publishing, pp. 121–125.

[50] Trukhan V, Horakova L, Skola J, et al. The Effect of High-Frequency Artifacts in Arterial Blood Pressure Waveforms on Pressure Reactivity Index. In: Costin H-N, Magjarević R, Petroiu GG (eds) *Advances in Digital Health and Medical Bioengineering*. Cham: Springer Nature Switzerland, pp. 381–387.

15 My Publications

1. Trukhan, V.; Škola, J.; Horáková, L.; Rožánek, M. Arterial blood pressure waveform artifacts detection using short-time Fourier transform. Lékař a technika – Clinician and Technology. 2024
   Accepted for publication

2. Trukhan, V.; Horáková, L.; Škola, J.; Rožánek, M. The effect of high-frequency artifacts in arterial blood pressure waveforms on pressure reactivity index. In: Costin HN, Magjarević R, Petroiou GG, editors. Advances in Digital Health and Medical Bioengineering: Proceedings of the 11th International Conference on E-Health and Bioengineering (EHB-2023), November 9–10, 2023, Bucharest, Romania. Volume 2: Health Technology Assessment, Biomedical Signal Processing, Medicine and Informatics. Basel: Springer Nature Switzerland AG; 2024. p. 381-7. IFMBE Proceedings. vol. 110. ISBN 978-3-031-62520-6. DOI: 10.1007/978-3-031-62520-6_42.
   Honorable mention for publication and oral presentation

3. Trukhan, V.; Horáková, L.; Škola, J.; Rožánek, M. Effect of pressure reactivity index calculation settings on the range of the optimal cerebral perfusion pressure. In: Proceedings of 2022 E-Health and Bioengineering Conference (EHB). Iasi: Gr. T. Popa University of Medicine and Pharmacy; 2022. ISBN 978-1-6654-8557-9. DOI: 10.1109/EHB55594.2022.9991435.
   1st place in the competition for the best publication and oral presentation

4. Rožánek, M.; Škola, J.; Horáková, L.; Trukhan, V. Effect of artifacts upon the pressure reactivity index. Sci Rep. 2022;12(1). DOI: 10.1038/s41598-022-19101-y.

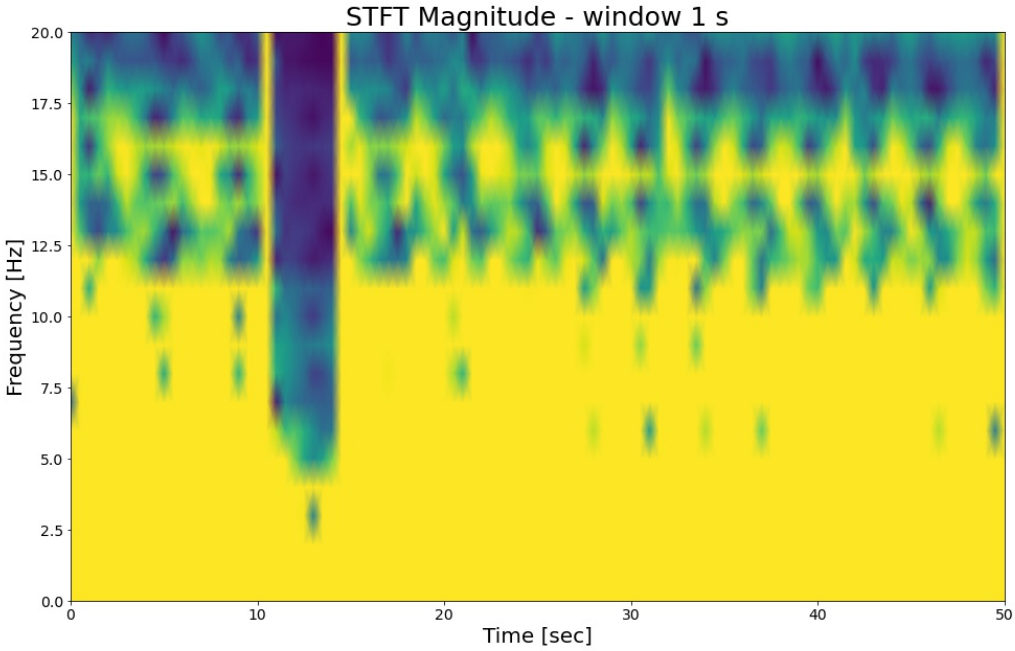
5. Ráfl J, Trukhan V. Databáze klinických dat v intenzivní péči. [Invited unpublished scientific lecture]. In: Digitalizace a umělá inteligence v anesteziologii a intenzivní medicíně - současnost a perspektivy; 2022 May 19; Praha, Czech Republic.

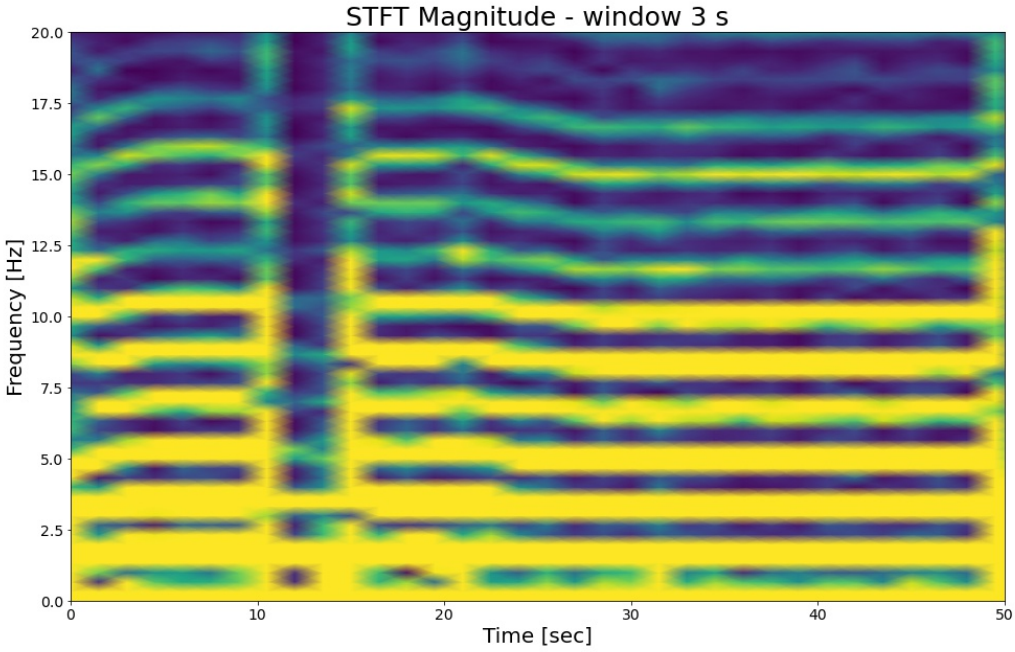
6. Trukhan V, Horáková L, Rožánek M. Program extension for data analysis from operating rooms. In: IEEE E-Health and Bioengineering EHB 2020; 2020 Oct 29-30; Iasi, Romania. Iasi: Gr. T. Popa University of Medicine and Pharmacy; 2020. ISBN 978-1-7281-8803-4.
   Online oral presentation

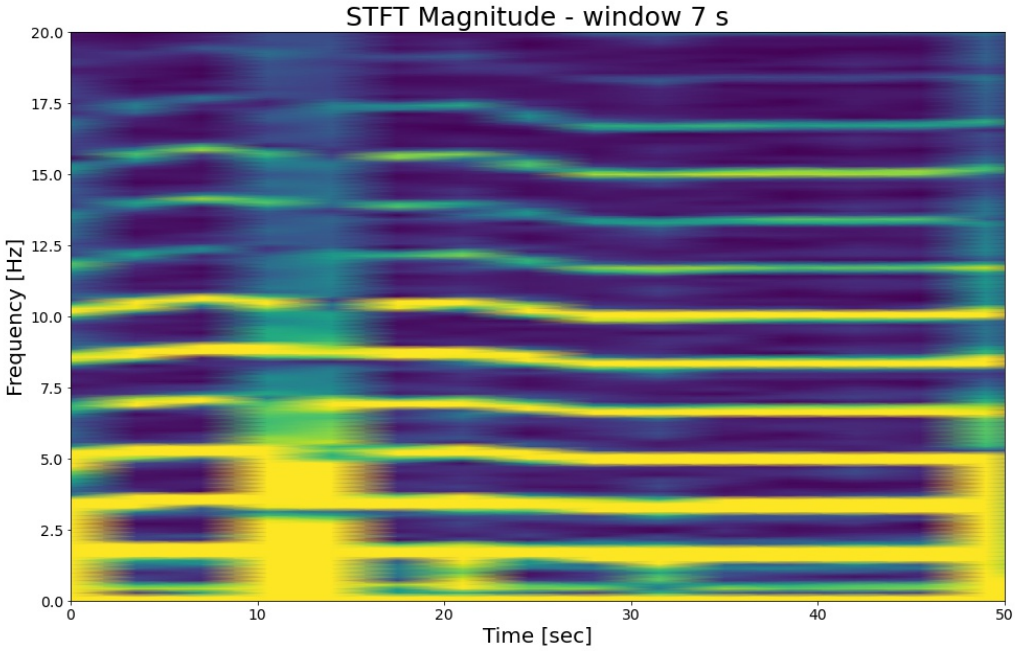
   Partial results of the doctoral thesis were presented at the student conference POSTER 2022 and were awarded in the best poster competition (3rd place).

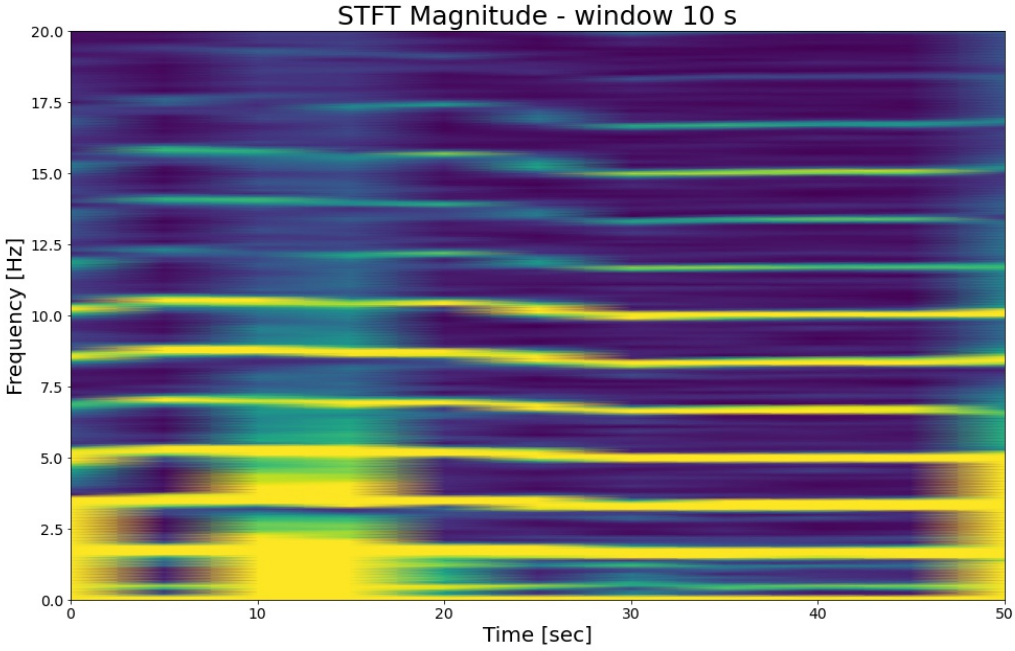
Annex A: Short-time Fourier transform for artifact detection

Rectangular artifact:

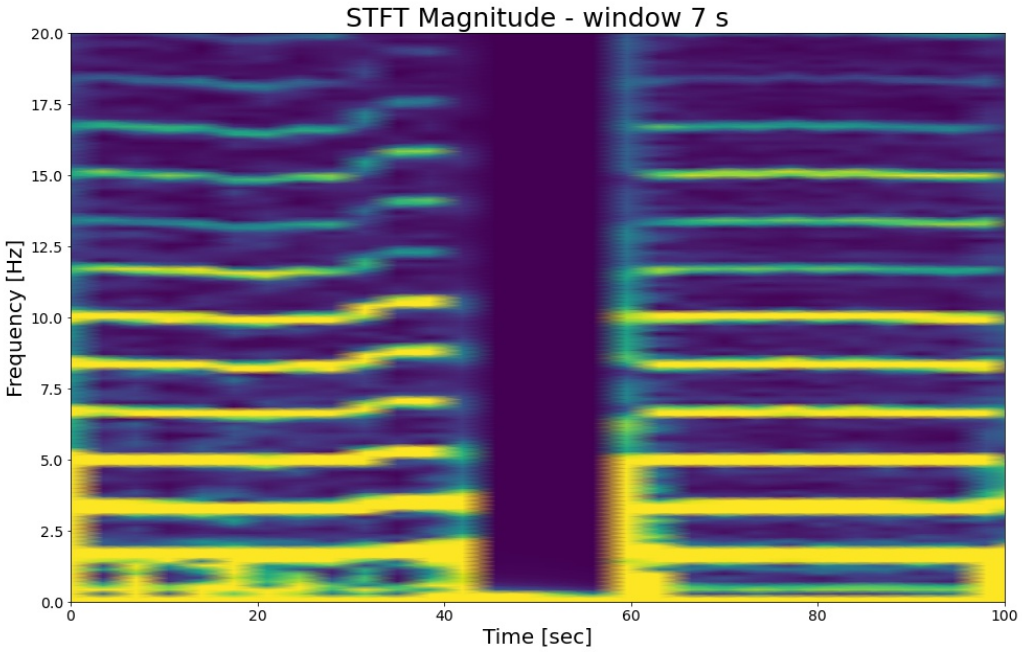


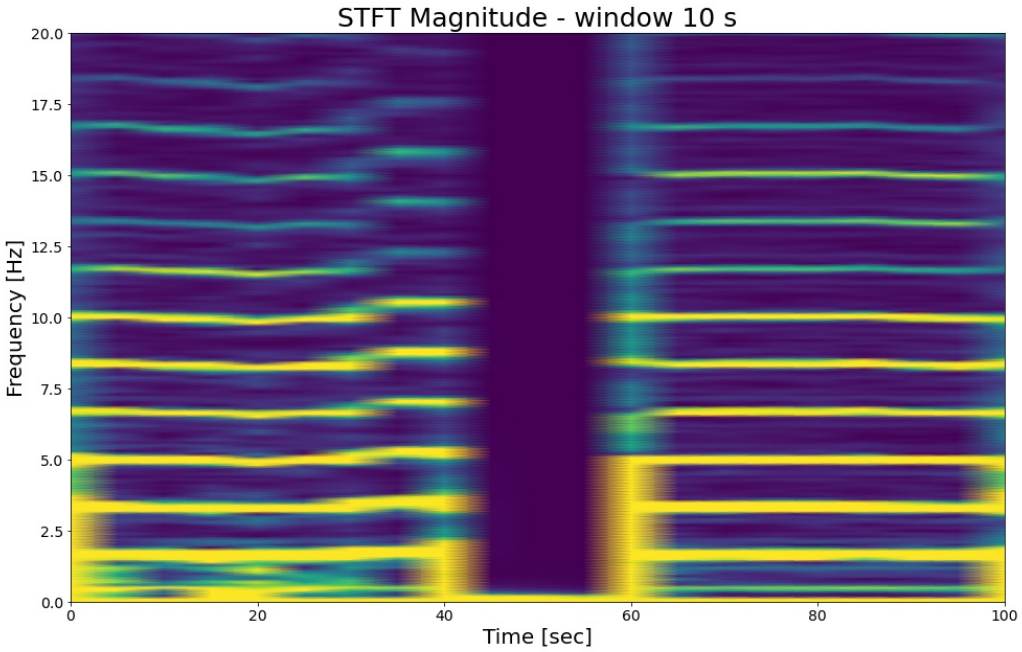


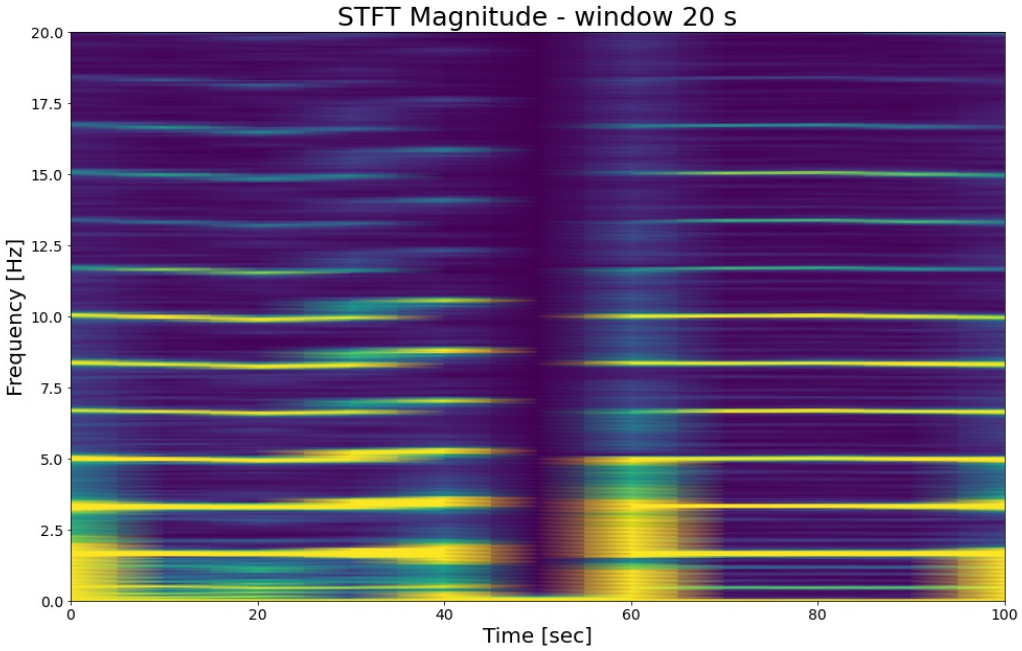

STFT Magnitude - window 7 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50

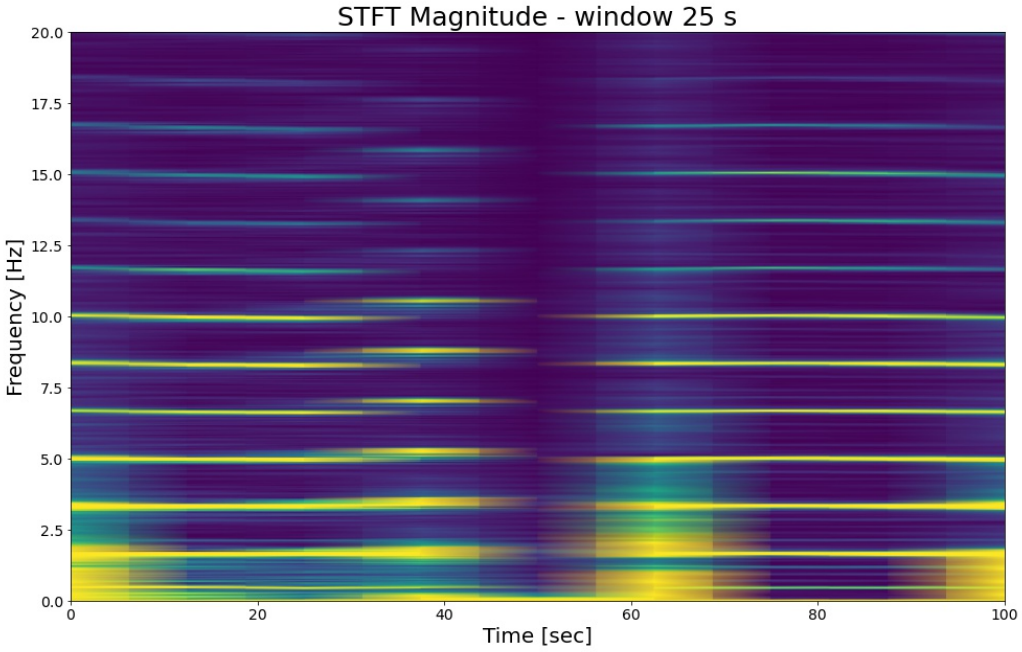

STFT Magnitude - window 10 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50

Saw tooth artifact:

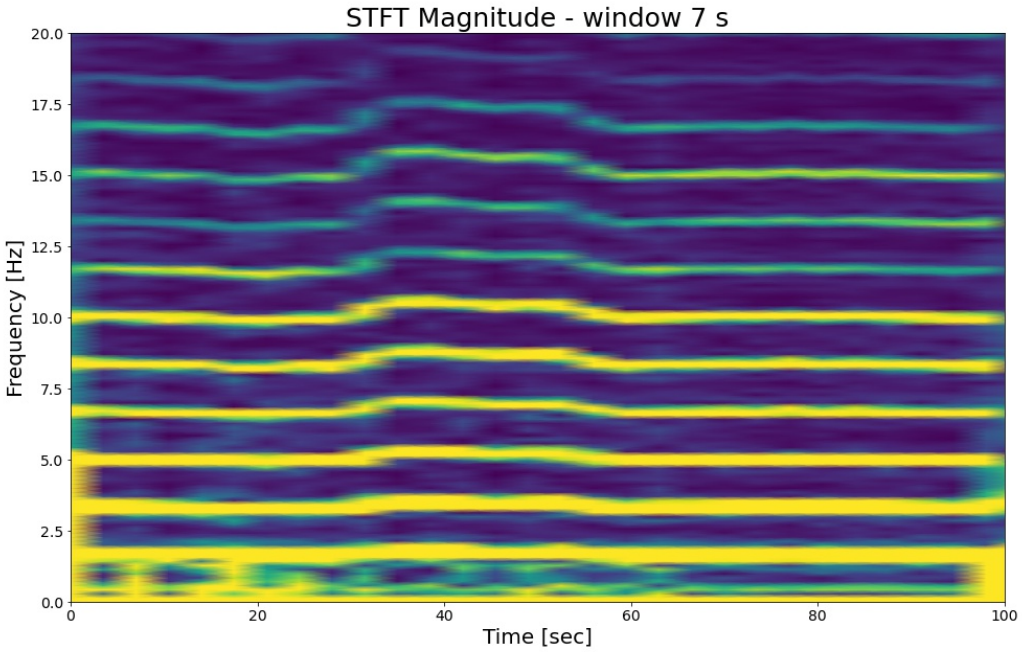


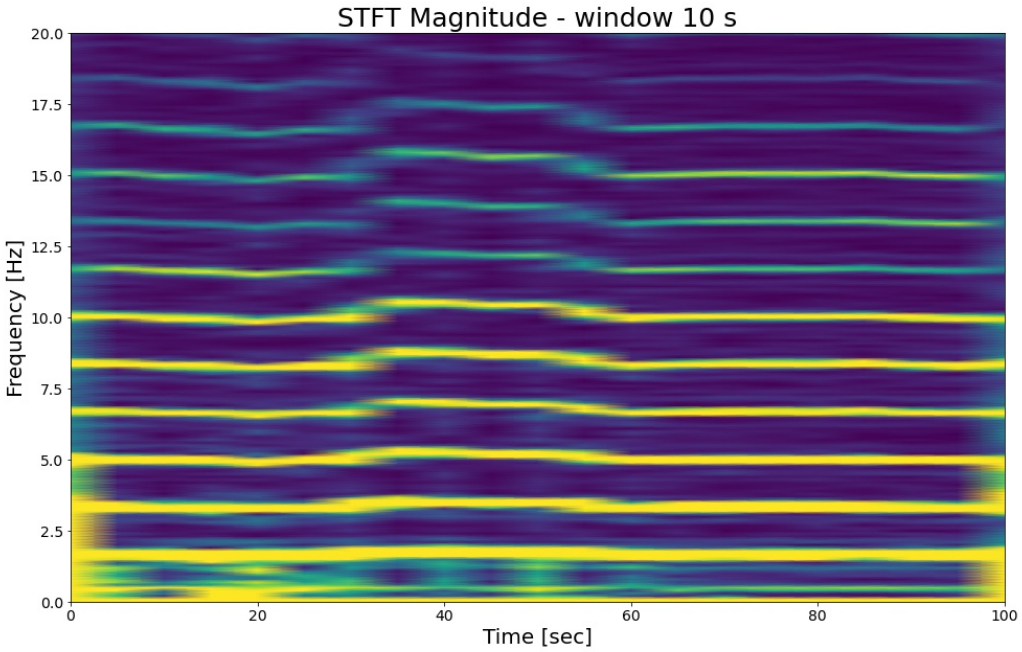


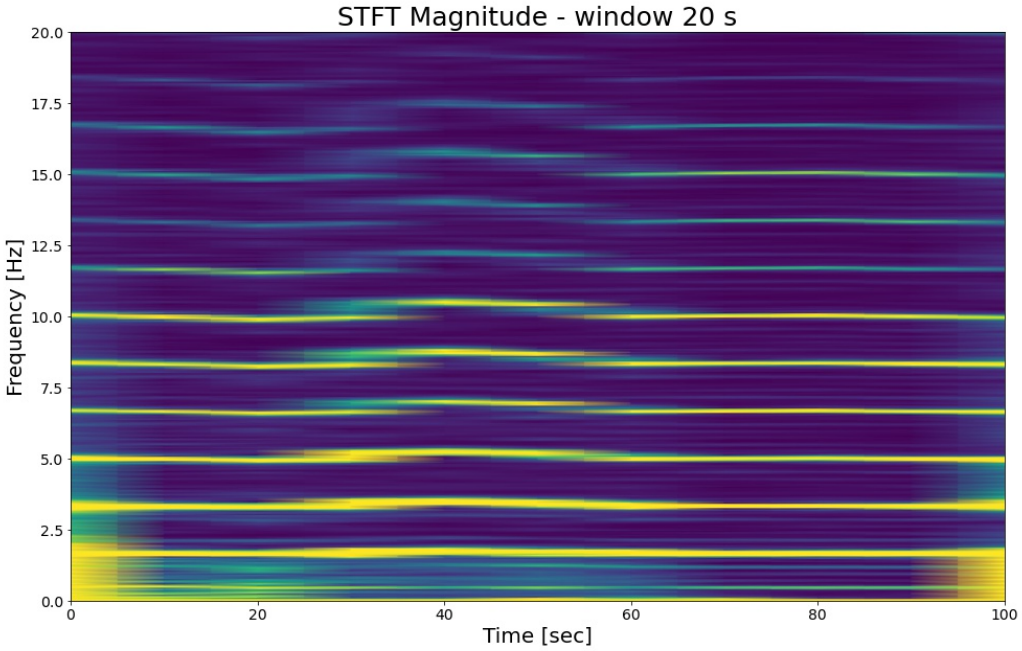

STFT Magnitude - window 20 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
20
40
60
80
100

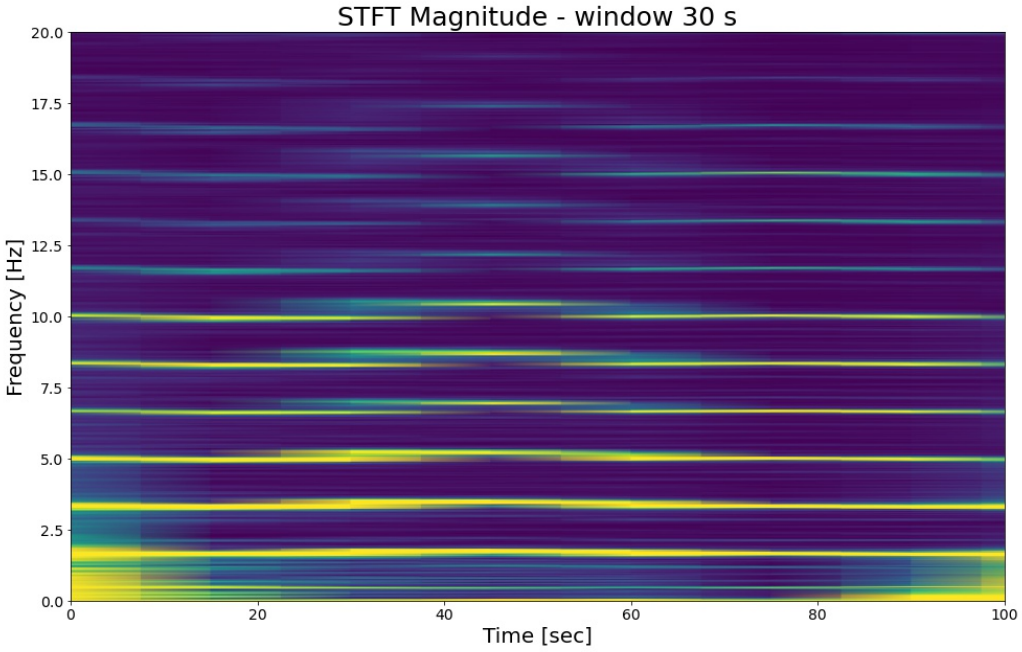

STFT Magnitude - window 25 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
20
40
60
80
100

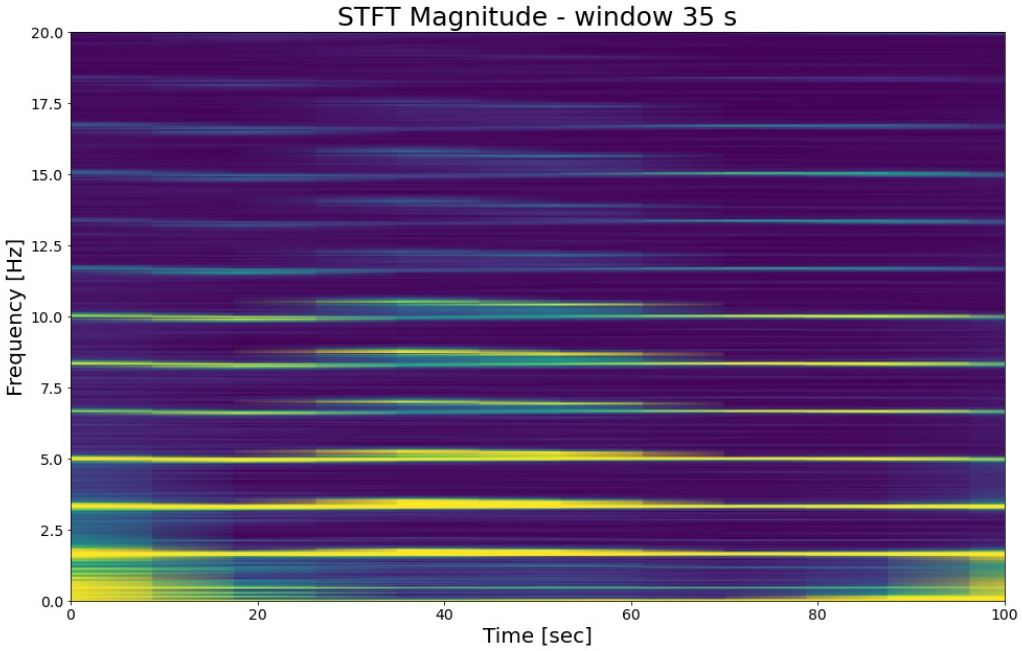
Isoline drift artifact:



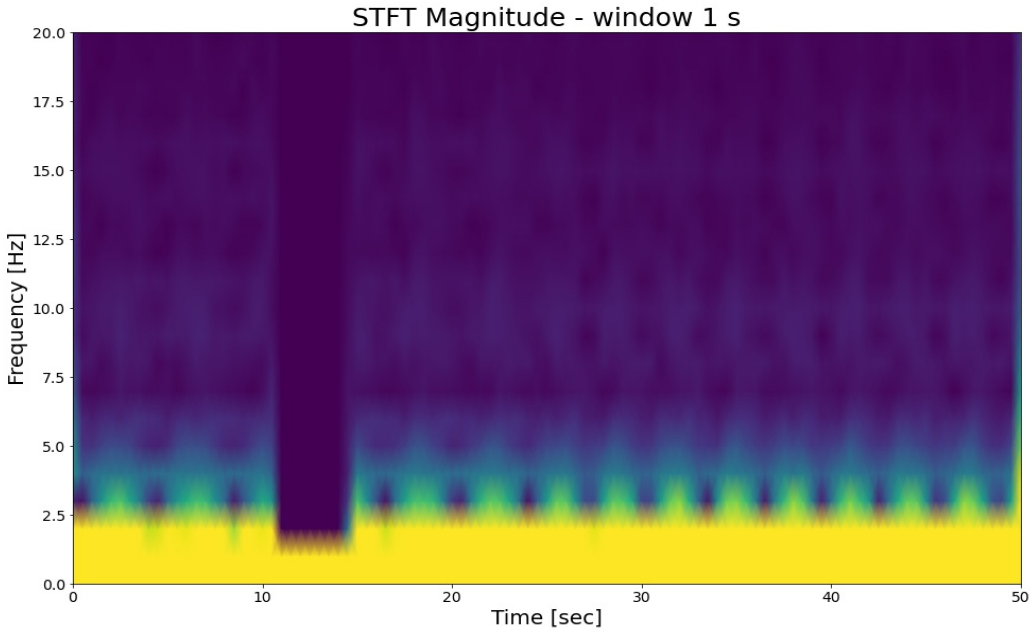


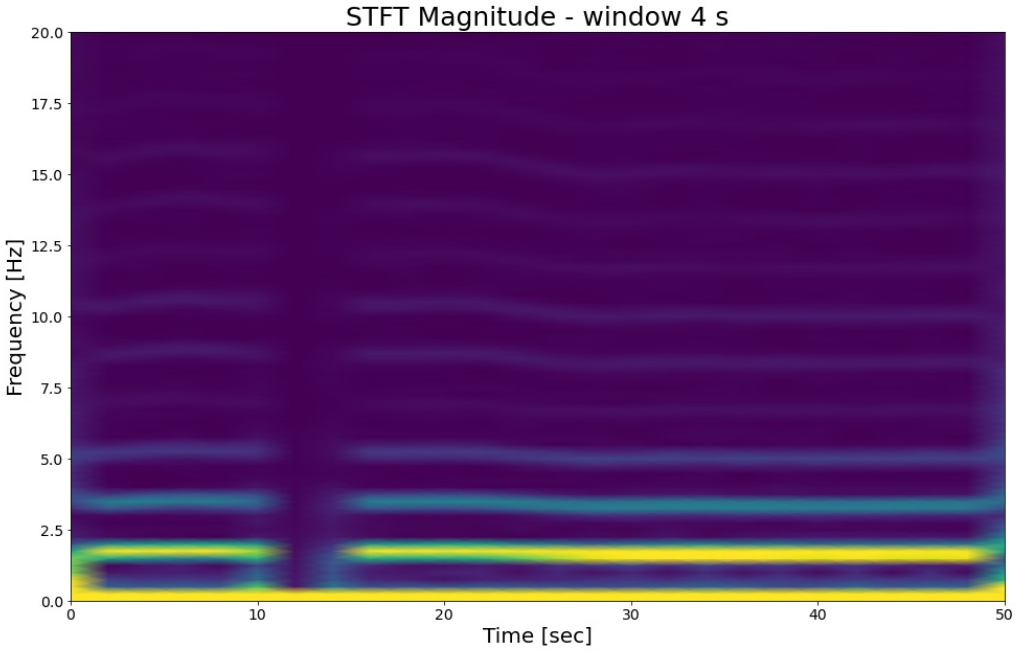

STFT Magnitude - window 20 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
20
40
60
80
100

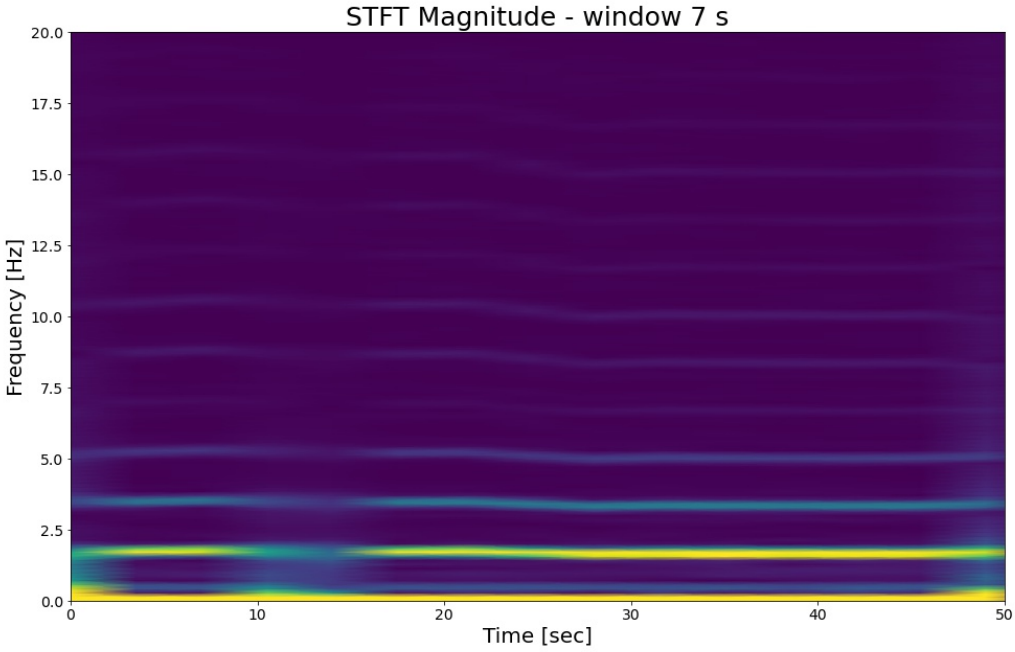

STFT Magnitude - window 30 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
20
40
60
80
100

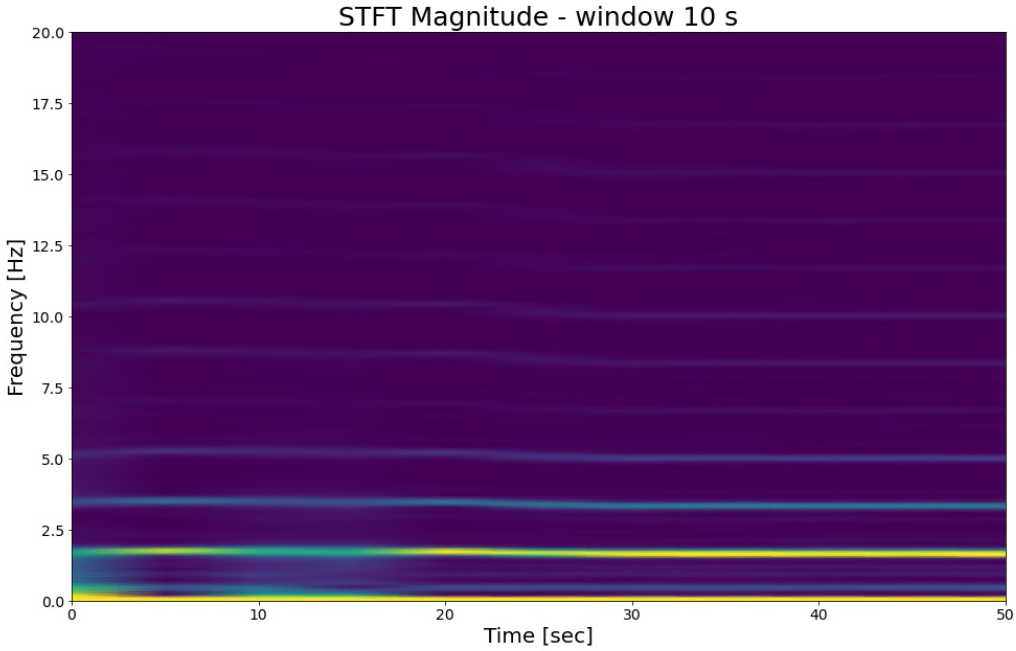


Constant ICP artifact:

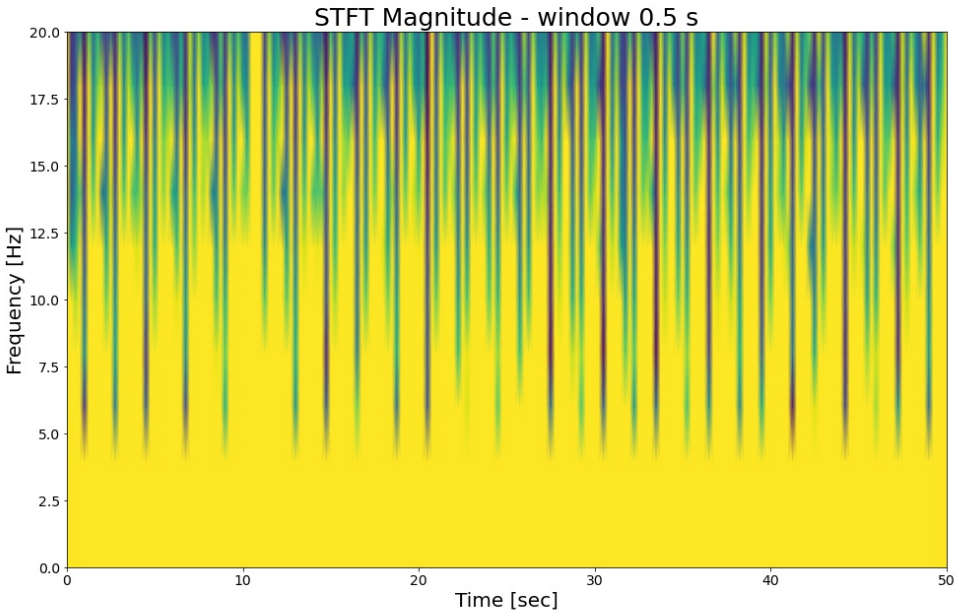


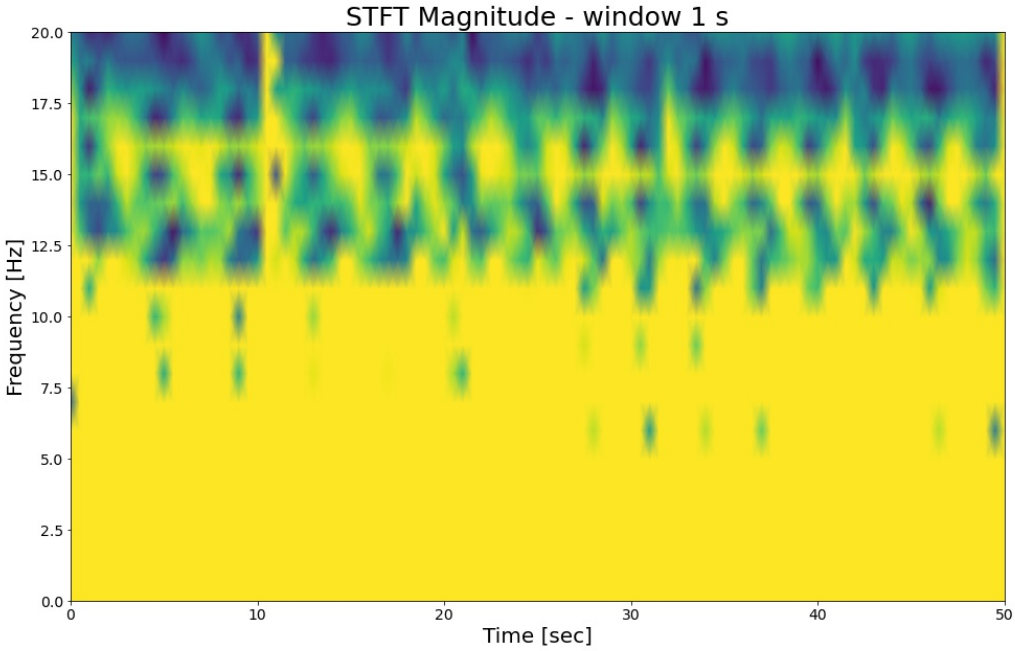

STFT Magnitude - window 4 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50

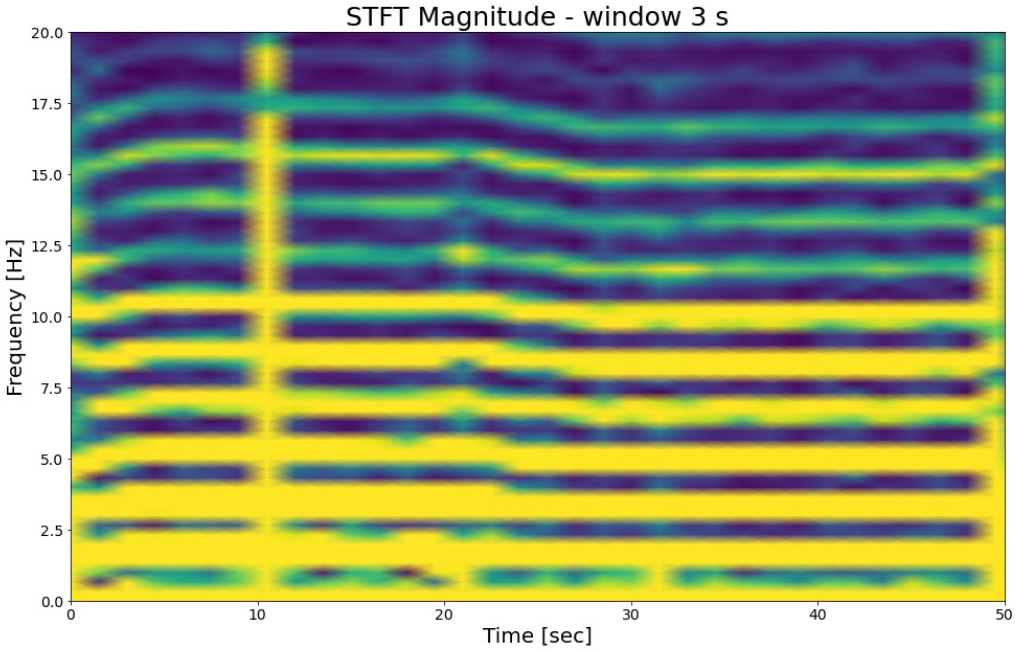

STFT Magnitude - window 7 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50

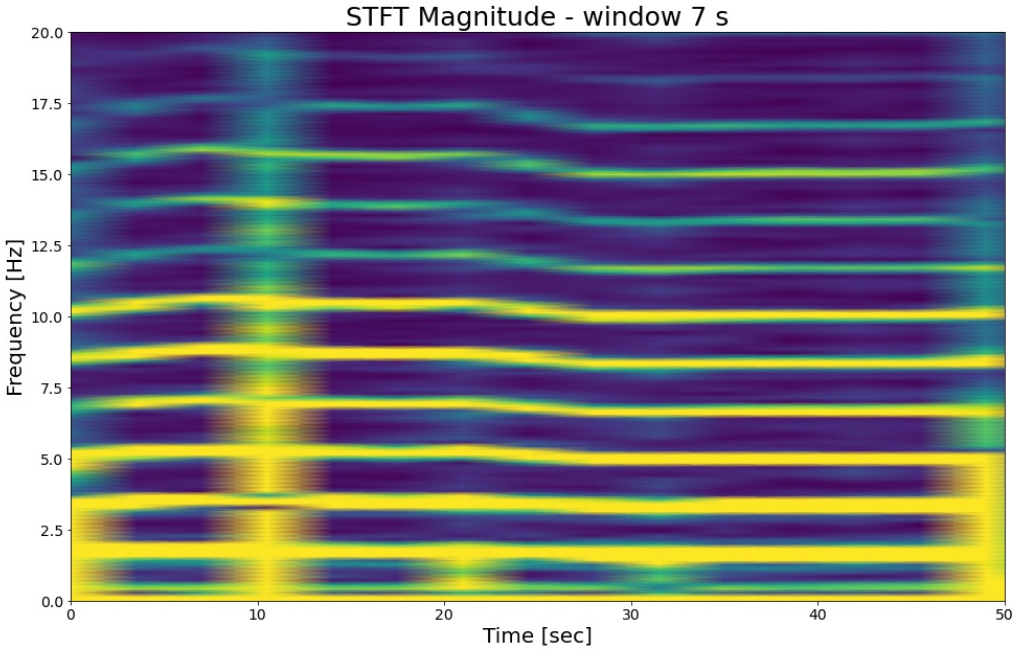


Fast impulse artifact:




STFT Magnitude - window 1 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50


STFT Magnitude - window 3 s
Frequency [Hz]
Time [sec]
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40
50


STFT Magnitude - window 7 s
Frequency [Hz]
Time [sec]
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
10
20
30
40
50

Annex B: Examples of the tested parameters from STFT on different simulated artifacts

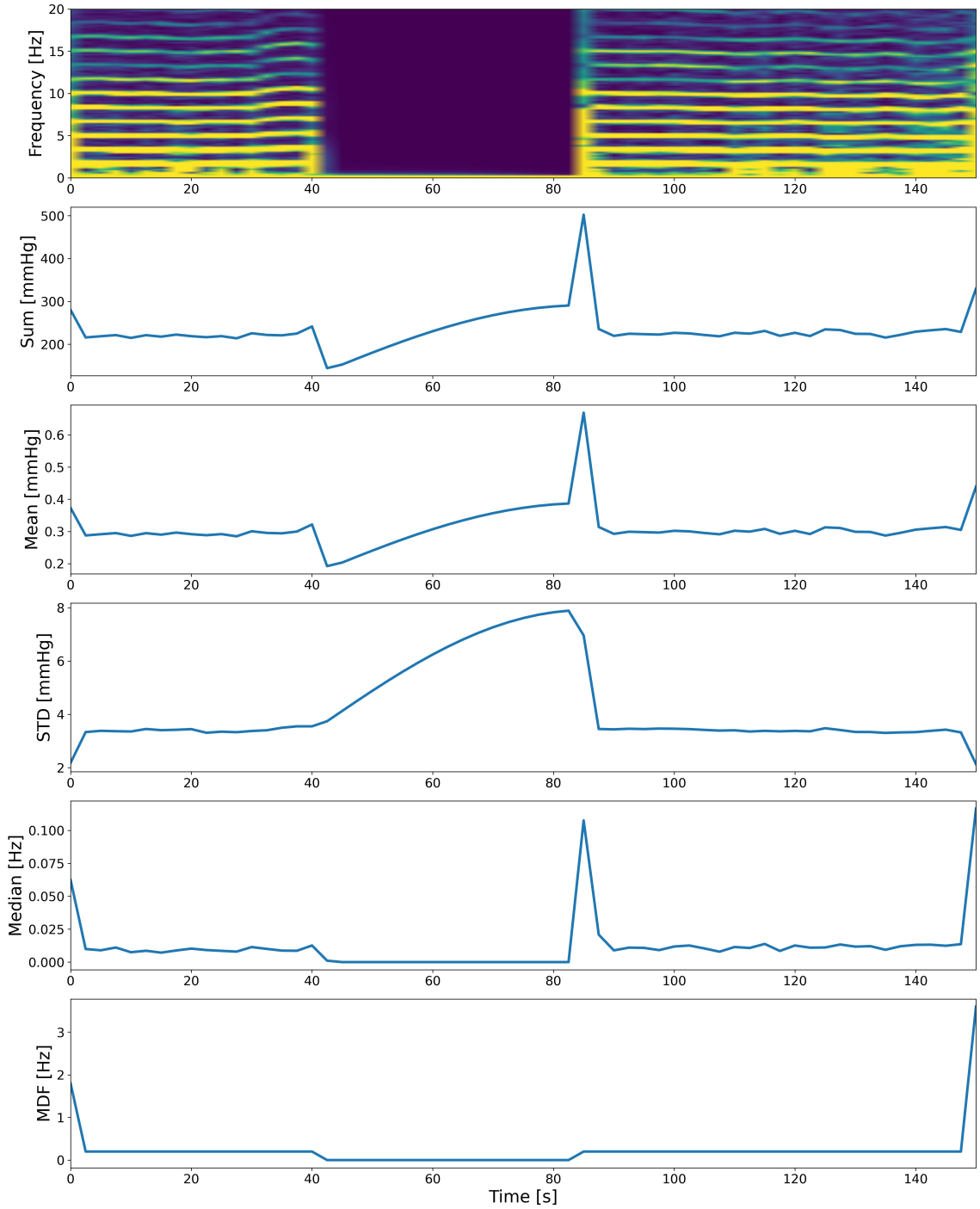


Figure B.1: Demonstration of tested parameters on the signal with saw tooth artifact: sum, mean, standard deviation (STD), median, median frequency (MDF)

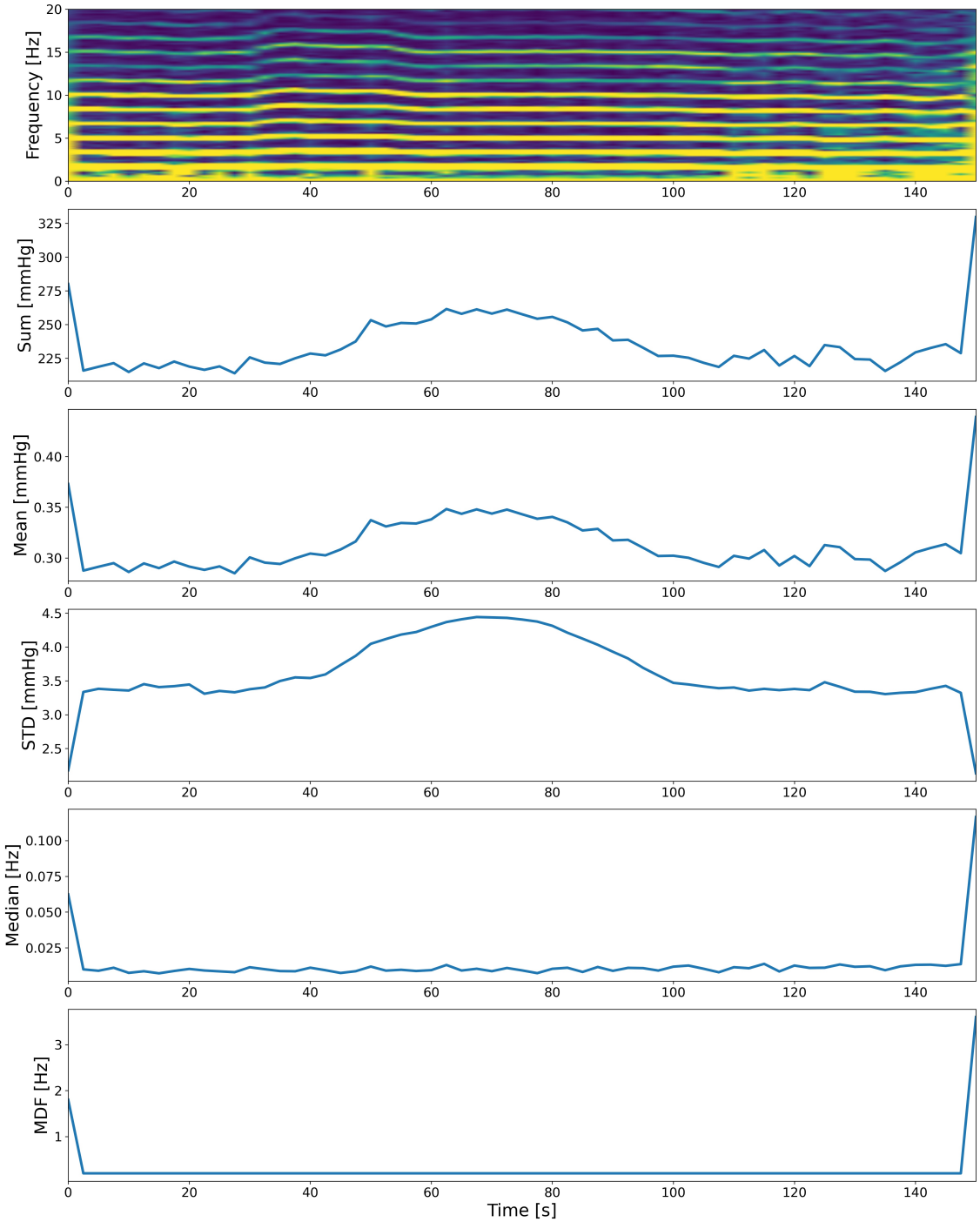


Figure B.2: Demonstration of tested parameters on the signal with baseline drift artifact: sum, mean, standard deviation (STD), median, median frequency (MDF)

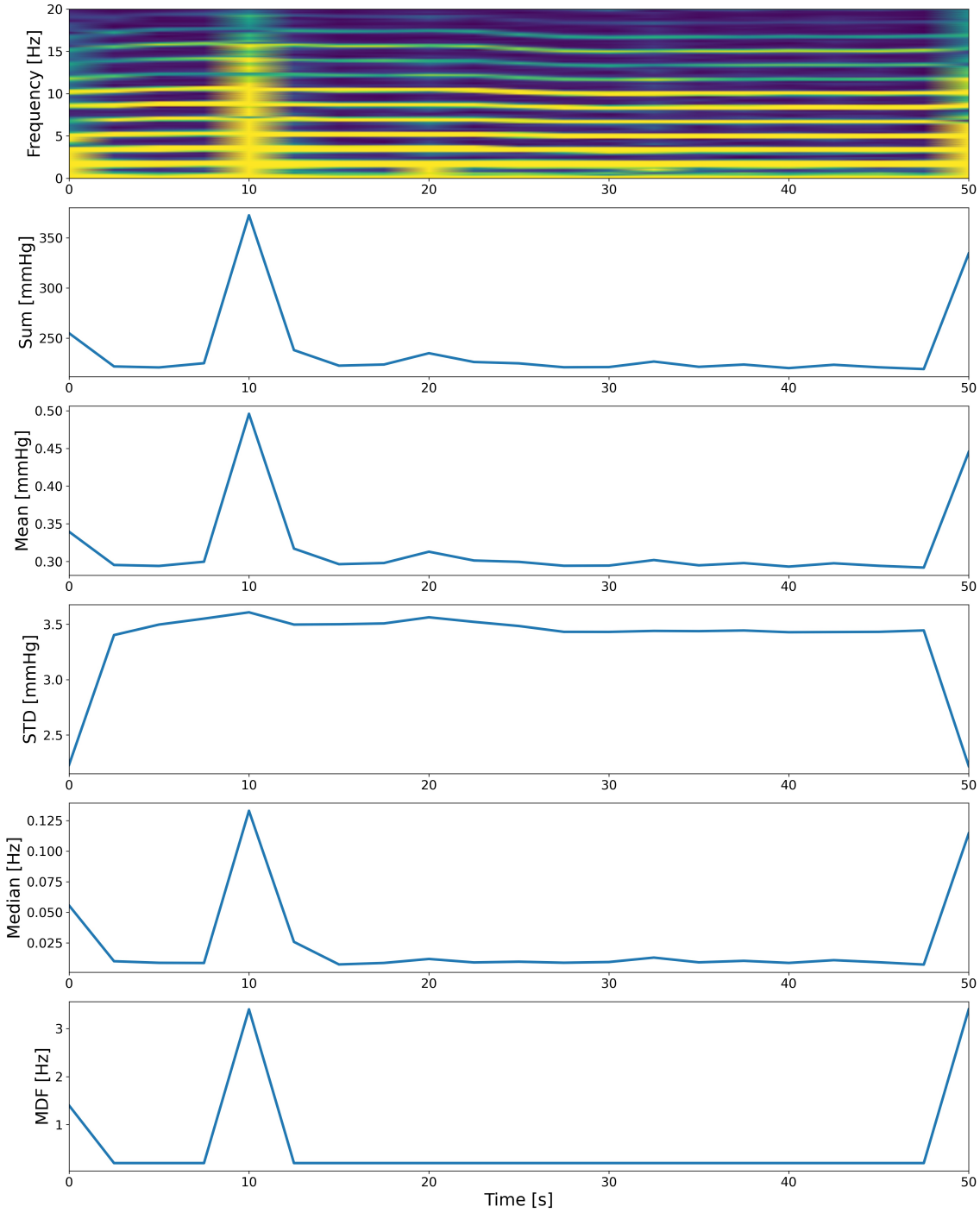


Figure B.3: Demonstration of tested parameters on the signal with fast impulse artifact: sum, mean, standard deviation (STD), median, median frequency (MDF)

Annex C: Python plug-in for ICM+ software

Python script: ICMP_ArtifactPercent.py

```
#import ...
import numpy as np
import scipy.signal as signal
import stattools as st

class ArtifactPercent:

   # DO NOT MODIFY THIS METHOD. It is a part of the ICM+--Python interface.
   def set_parameter(self, param_name, param_value):
      setattr(self, param_name, param_value)

   # You can append your own code to the constructor, if needed.
   # You should not set here values of parameters declared in your XML
   # configuration file because ICM+ will do it for you.
   # You will have to add your own code, only if you need to initialise some
   # extra data structures which were not declared in the XML config file.
   def __init__(self):
      self.variables = []
      self.sampling_freq = None
      self.file_path = None
      self.threshold1 = 2      # threshold1
      self.threshold2 = 8      # threshold1

   # You can append your own code to the destructor but most likely
   # you will not need it.
   def __del__(self):
      pass

   # 'calculate' is the main work-horse function.
   # It is called with a data buffer (one or more) of size corresponding to the Calculation
Window
   # It must return one floating-point number
   # It take the following parameters:
   # sig1 - input variable/signal 1
   # ts_time - part of the data time stamp - number of milliseconds since midnight
   # ts_date - Part of the data time stamp - One plus number of days since 1/1/0001
   # It can also use the data sampling frequency:
   #    self.sampling_freq



   def calculate(self, sig1, ts_time, ts_date):
```

```
        sig1 = np.array(sig1)
        sig1 = np.nan_to_num(sig1, nan = 5000)

        window_size = int(5 * self.sampling_freq)
        f, t, Zxx = signal.stft(sig1, fs=self.sampling_freq, nperseg=window_size)

        detected_artifacts = np.zeros(len(t), dtype=int)
        magnitude = np.abs(Zxx)
        magnitude_diff = np.diff(magnitude, axis=1)
        change_signal = np.sum(np.abs(magnitude_diff), axis=0)
        change_signal = np.nan_to_num(change_signal, nan = 100)

        change_signal_mean = change_signal.mean()
        change_signal_std = change_signal.std()
        standardized_change_signal = (change_signal - change_signal_mean) /
change_signal_std

        threshold = self.threshold1
        artifact_times = t[1:][standardized_change_signal > threshold]
        artifact_indices = np.searchsorted(t[1:], artifact_times).astype(int)

        consistency_threshold = self.threshold2
        spectrum_variability = np.std(magnitude, axis=0)
        consistent_regions = t[spectrum_variability > consistency_threshold]
        consistent_indices = np.searchsorted(t, consistent_regions).astype(int)

        detected_artifacts[artifact_indices] = 1

        if len(consistent_indices) > 0:
           detected_artifacts[consistent_indices] = 1

        #nan_indices = np.argwhere(np.isnan(spectrum_variability)).flatten()
        #if nan_indices.size > 0:
        #   detected_artifacts[nan_indices] = 1

        detected_artifacts[0] = 0
        detected_artifacts[-1] = 0

        artifact_percentage = np.mean(detected_artifacts) * 100

        result = float(artifact_percentage)

        return result
```

Python plug-in configuration file: ICMP_ArtifactPercent.xml

```
<?xml version = "1.0"?>

<PyToICMPlusConfig>
  <Function Name = "ArtifactPercent" Type = "Stats" SignalsCount = "1">
    <GUID>{The value is anonymized}</GUID>
    <Description>This function detect artifacts in ABP or ICP signal and returns the
percent of artifacts in time
interval.</Description>
    <Parameter ShortName = "threshold1" IsMandatory = "True">
      <Caption>threshold1</Caption>
      <Description>threshold for evaluating the differential spectrum
signal</Description>
      <Type Name = "Float" Min = "0" Max = "0" DefaultValue = "2"/>
    </Parameter>
    <Parameter ShortName = "threshold2" IsMandatory = "True">
      <Caption>threshold2</Caption>
      <Description>threshold for evaluating consistent spectral regions</Description>
      <Type Name = "Float" Min = "0" Max = "0" DefaultValue = "8"/>
    </Parameter>
  </Function>
</PyToICMPlusConfig>
```