Prediction of Intradialytic Hypotension using Photoplethysmography

Erik Wallenborg

Department of Electrical and Information Technology

December 22, 2011

Abstract

Acute intradialytic hypotension is the most common complication during a hemodialysis treatment. In the doctoral dissertation "Signal Modeling and Detection in Nephrologic and Cardiac Applications" [1], a method for prediction of this complication using photoplethysmography is demonstrated. This thesis will implement and evaluate this method on a larger database.

It will be shown that the database gathered at Lund University Hospital in 2010 can not be used for the prediction of acute intradialytic hypotension with this method. Because of this, the method itself could not be evaluated. It will also be shown that there are clear differences between the signals from the Nonin Medair Life Sense and the wireless Nonin 4100 pulse oximeter sensors.

Contents

Chapter 1

Introduction

1.1 Dialysis

The kidneys have three main functions: (i) removing excess fluid and waste products, (ii) regulating the electrolyte level and the acid-base balance, and (iii) production of hormones such as renin (blood pressure regulator) and EPO (red blood cells regulator). All of these abilities makes the loss of kidney function a very serious medical condition with fatal outcome if untreated [2].

Today there are three treatments for kidney failure; transplanting a new kidney, peritoneal dialysis (PD) and hemodialysis (HD). A transplant is the desired treatment but can not always be done immediately or sometimes not at all, depending on when and if a matching kidney can be found. Most kidney failure patients will therefore need some sort of dialysis, where the kidney functions are mimicked for the patient. A patient requires dialysis treatment 3 times a week lasting 3 to 5 hours. Since the kidneys can no longer handle the removal of excess fluid (peeing), the treatment will remove about 2 liters of fluid, depending on treatment settings. Removal of more than 3% of the patients bodyweight in fluids might lead to complications. The major difference between the two dialysis types is where the blood is treated [2].

In PD you fill the peritoneal cavity (the cavity around your stomach and intestines) with a solution. The membranes between the blood vessel and the solution works as a filter allowing the transport of waste products and excess fluid to the solution by diffusion and osmotic pressure. The dirty solution can then be drained (see Figure 1.1) [2].

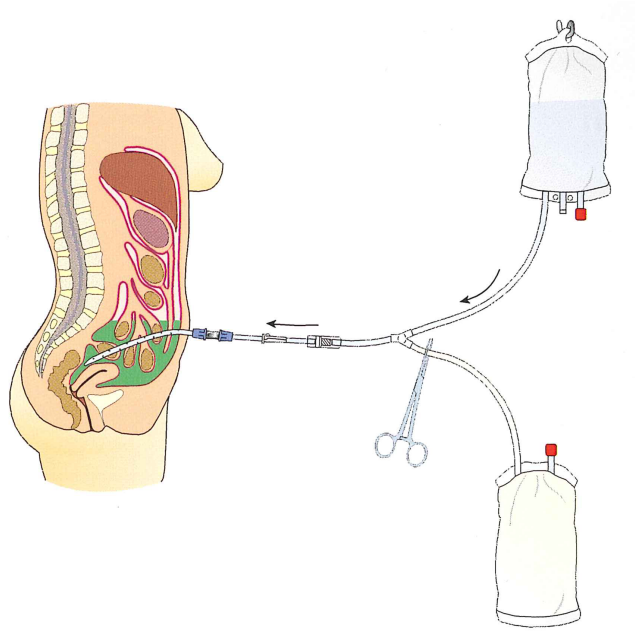In HD a machine is instead used to pump the blood outside the body,

Figure 1.1: *Peritoneal dialysis with the solution (green) being filled into the peritoneal cavity. Reproduced from [2].*

past a filter (dialyzer) and back to the patient (see Figure 1.2). In the dialyzer a solution on the other side of the filter membrane attracts waste and excess fluid by diffusion and osmotic pressure. The treatment is set up with the amount of excess fluid that needs to be removed, tempature of the returning blood (if it is cold the patient can be heated via the machine) and several other parameters [2].
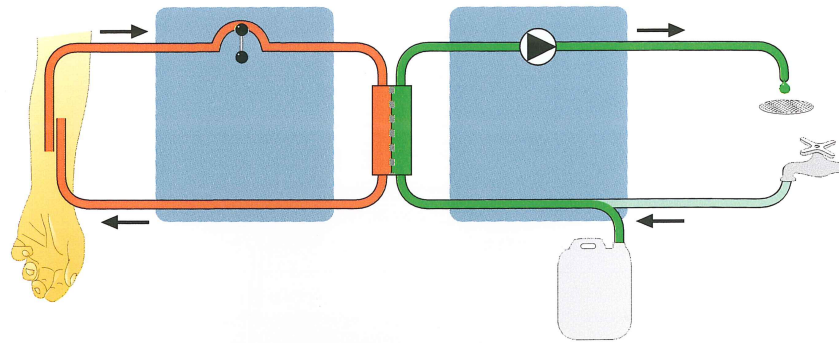
Figure 1.2: *Simple schematic diagram of the blood (red) and solution (green) flow during hemodialysis. Reproduced from [2].*

1.2 Hypotension

Symptomatic hypotension (low blood preassure followed by nausea, vomiting or even fainting) is the most common complication during a HD treatment [2]. The cause of hypotension can be one of many, such as infections, bleeding, allergic reaction, heart failure, or the loss of fluid during a HD treatment [3]. Hypotension is not only unpleasant, it can also be life-threatening if left untreated as vital organs may not get enough oxygen [4]. It has been shown that the speed of the excess fluid removal during a HD treatment correlates to the number of patients that suffer from acute hypotension [2]. Detecting hypotension in advance and enabling interuption of the HD treatment is therefore very useful.

1.3 Photoplethysmography

Photoplethysmography (PPG) is an optical technique that can be used for detection of blood volume changes in the microvascular bed of tissue. A near infrared light emitting diode together with a photodetector is used to measure the perfusion in either a toe, earlobe or finger. The PPG signal consists of a pulsatile (AC) part following the heartbeats and a slowly varying (DC) part attributed to respiration, sympathetic nervous system activity and thermoregulation (see Figure 1.3) [5]. The shape of a PPG pulse can also vary with health and age (see Figure 1.4) [1]. The technique has several uses such as, but not limited to, measurement of oxygen saturation, blood pressure, heartrate and cardiac output. A wide range of medical devices has PPG sensors. One of the more common is the pulse oximeter that measures heartrate and oxygen saturation [5].
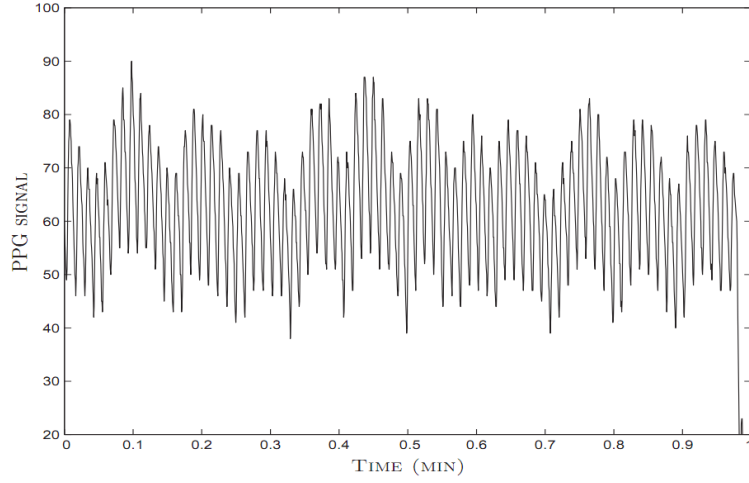
Figure 1.3: *PPG signal measured at the finger. Two components can be observed: an "AC" component related to the heart rate (1.27 Hz), and a "DC" component related to respiration (0.19 Hz). Reproduced from [1].*
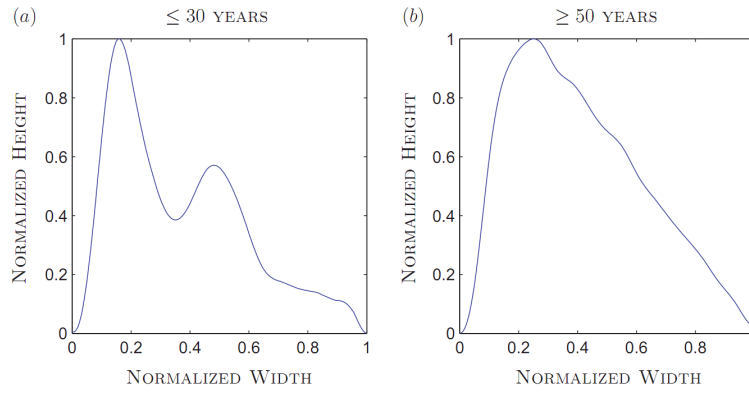

Figure 1.4: *Normalized PPG pulse shape recorded from the finger of a young and healthy person (a) and an old renal patient (b). Reproduced from [1].*

1.4 Thesis Introduction

In the doctoral dissertation "Signal Modeling and Detection in Nephrologic and Cardiac Applications" [1], Solem demostrates a method for prediction of acute intradialytic hypotension using a pulse oximeter. The source of intradialytic hypotension is considered to be volume depletion due to the ultrafiltration rate (the amount of excess fluid being drained from the patient) being too high. Volume depletion causes reduction in blood volume which in turn causes less blood to reach the capillaries (in e.g. fingers). By monitoring the change in signal amplitude from a pulse oximeter, acute hypotension could be predicted on average 38 min in advance [1].

The method's performance evaluation was done on 25 dialysis treatments among 11 hypotension-prone patients. A total of 7 acute symptomatic hypotensive episodes occurred in 5 treatments (with 2 treatments having double blood pressure drops). The patients were monitored during their treatments and the hand used for PPG measurements was kept still and close to the heart level [1]. Though being enough to show the potential of the method, it is not a very large patient group and further study is needed.

This thesis will implement and evaluate the method in [1] on a larger patient group with over 120 dialysis treatments in total. The treatments are normal without extra supervision. The patient group consists of both hypertension-prone and resistant patients.

Chapter 2

Data and Equipment

2.1 Database

The patient group consists of 26 patients (20 men, 6 women) between 41 and 88 years of age. Each patient had up to 5 treatments for a total of 125 treatments in the patient group. Connection errors during the data collection made 5 recordings unusable. In total, 120 treatments could be used though 14 of these recordings, also due to connection errors, miss data from parts of the whole treatment. 12 out of the 26 patients are prone to hypotension but no hypotensive episodes occurred during any of the treatments.

The data used in this thesis was collected at the dialysis clinic Filialen at Lund University Hospital during the spring of 2010. Nurses at the clinic were instructed in the setup and usage of a computer that recorded the signal from the pulse oximeters. Treatment notes were also filled in for each treatment. This information contains placement and ID number of the pulse oximeter, start and stop times for the treatment, blood pressure before, during and after the treatment and treatment settings such as pump speed and ultrafiltration rate. On the treatment notes certain events are also recorded such as food intake and complications. The complications can consist of the pulse oximeter needing to be adjusted for a better connection or that the treatment had to be stopped due to dialysis machine alarms.

2.2 Equipment

Five wireless Nonin 4100 pulse oximeters (see Figure 2.1), numbered 1 to 5, with a sampling rate of 75 Hz were used for the PPG measurement on

the patients. The sensors where connected via bluetooth to a laptop with a recording program in Labview and the data from the sensors was saved, without processing, as binary files. All treatments were done with Gambro AK 200 S HD dialysis machines (see Figure 2.2).
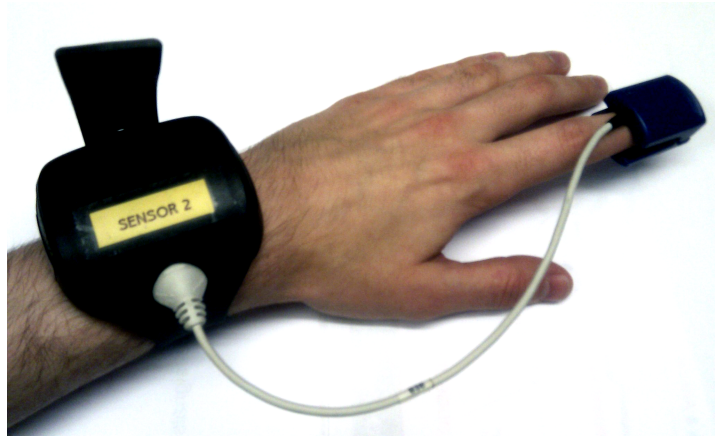

Figure 2.1: *Wireless Nonin 4100 pulse oximeter sensor number 2.*
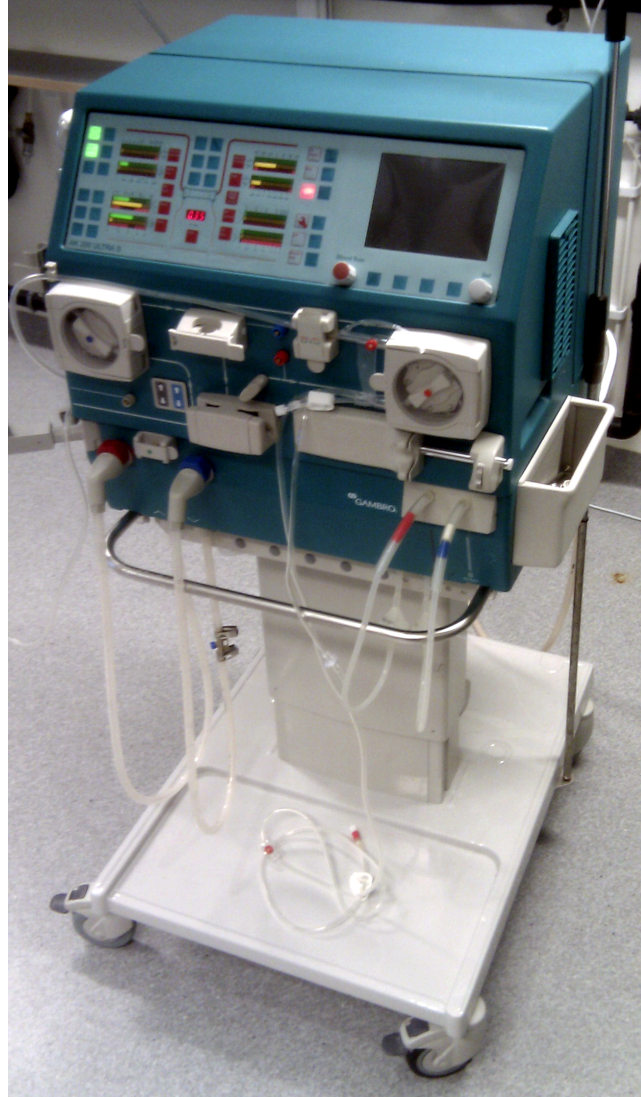
Figure 2.2: *The Gambro AK 200 S hemodialysis dialysis machine.*

Chapter 3

Methods

3.1 Data Extraction

The binary data from the treatments needed to be processed before it could be used in Matlab. The first problem was that of synchronization. At times the logging computer would loose and then re-establish the wireless contact with the pulse oximeters. The data is sent in packages (see Figure 3.1) of 125 bytes (25 frames with 5 bytes each) with the first byte in each frame as a status byte, indicating quality and usability of the signal. In this status byte there is a synchronization bit indicating the beginning of each package with a one (the rest holds zeros). We can therefore remove all packages that start with a one but holds less then 24 zeros in the synchronization bit (as a one should be sent every 25 th status byte). This will create gaps in the data, but since each package also contains a timestamp this is not a problem [6].

Synchronization errors are not the only reason for loss of data. The sensor itself will indicate when the signal is unusable for heartrate or oxygen saturation. This is indicated in the heartrate signal with the fixed value of 511 beats per minute or with a oxygen saturation of 127 percent [6]. These parts of the signal were also removed.

In addition, the status byte also contains other indicators of an unusable signal such as sensor disconnections, artifacts and sensor faults [6]. After analysis, the size of these unusable sequences is shown to be either very short (1-2 data points) or quite long (more than 25 data points, which is roughly one third of the pulse period). These parts of the signal were also removed.

In [6] we also note that the way the sensor and Matlab handles most and least significant bytes (MSB, LSB) differ, and this needs to be handled. Also, the order of the bits needs to be rearranged before they can be read.
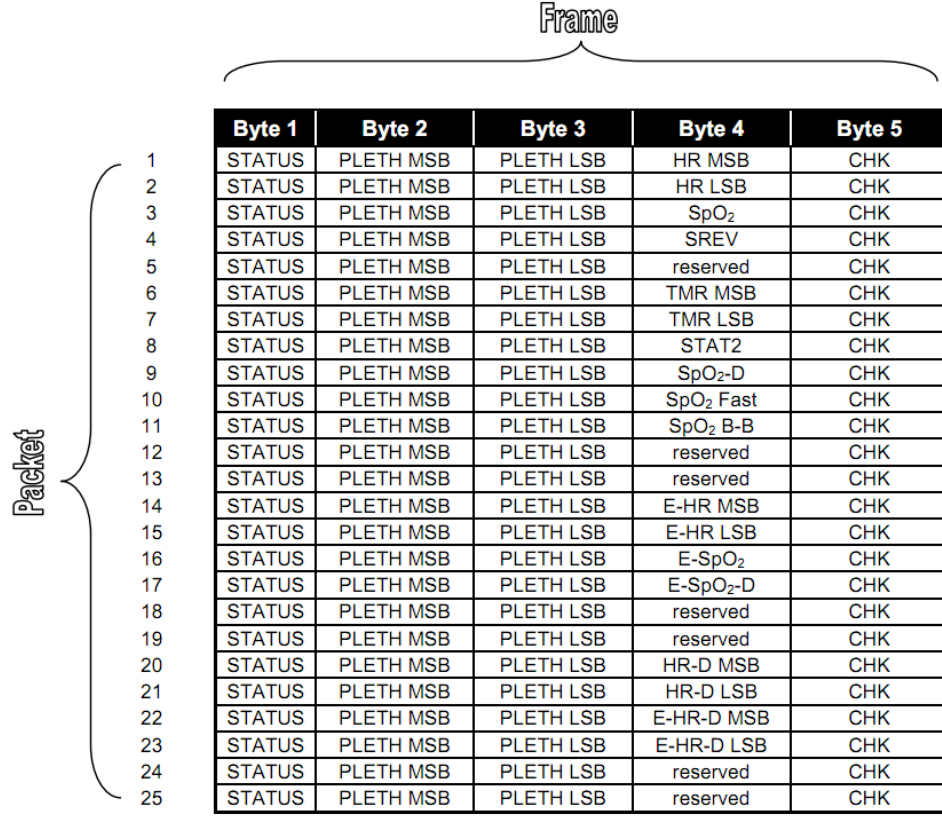
Figure 3.1: *The data structure of one package with 25 frames of binary data. Reproduced from [6].*

The table below (from [6]) shows the generic format for the bits of the pulse (denoted HR0-8) signal.

	7	6	5	4	3	2	1	0
HR MSB	x	x	x	x	x	x	HR8	HR7

	7	6	5	4	3	2	1	0
HR LSB	x	HR6	HR5	HR4	HR3	HR2	HR1	HR0

A Matlab script was written for opening, processing and saving the binary PPG data as .mat files (see appendix A.1). The script took over an

hour to run but saved a lot of time since all the corrections made above only needed to be performed once. Included in the data processing was the task of transferring all relevant non-binary data from the treatment notes by hand. The start and end of the treatment was saved with each PPG .mat file as well as all blood pressure measurements. The sensor identification number was also noted.

3.2 Prediction Algorithm

As described in [1], the method for prediction of acute intradialytic hypotension is the calculation of the relative magnitude of the capillary pulse (RMCP) (i.e. the pulse measured from the finger capillaries). As mentioned, photoplethysmography can be used for detection of blood volume changes in the microvascular bed of tissue [5]. Therefore the RMCP should decrease as less blood reach the extremities when volume depletion occurs.

The method for prediction of hypotension with the RMCP can be divided into three steps; baseline correction, RMCP computation and alarm evaluation.

3.2.1 Preprocessing

As previously could be seen in Figure 1.3, the PPG signal consists of a pulsatile AC and a slowly varying DC part. To make an alarm prediction the RMCP signal is required (which should only be effected by the AC part) and therefore the DC part of the signal needs to be removed [1].

To be able to apply a filter there must not be any gaps in the data as there are due to the data processing. When the gaps are small (i.e. 1-2 data points), as many are, linear interpolation in between is sufficient. In the case of larger gaps (one third of a pulse up to several pulses) the shape and amplitude of the missing data can not be reconstructed. We therefore linear interpolate over all gaps and save the positions of all reconstructed data. If we later get a false alarm due to this, we can ignore it since we know the alarm is caused by missing data.

To remove the DC part the signal is first downsampled (see Figure 3.2) from 75 Hz to 3 Hz, as this increases the numerical robustness and speed of digital low-pass filtration. This is followed by forward and backward filtering with a second order low-pass Butterworth filter with cutoff frequency of 0.5 Hz. This gives a good estimate of the slow baseline variation, or DC part of the signal. This estimate is then upsampled back to 75 Hz and subtracted from the original signal.
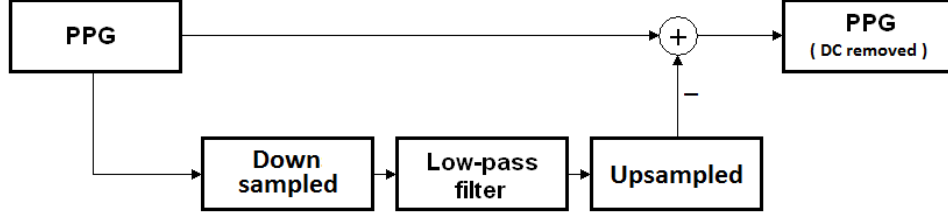
Figure 3.2: *The method for removing the DC part of the PPG signal. The PPG signal is downsmapled, filtered with a low-pass butterworth filter, up-sampled and finaly substracted from the original PPG signal.*

3.2.2 Computation of Relative Magnitude of Capillary Pulse

To compute the RMCP one of several methods can be used. The peak-to-peak amplitude can be used, another method is to compute the envelope. In [1], a third method is used which is to compute the integral of the PPG signal. In the case of a finite number of data points, the integral can be computed by the summation of the absolute values of the PPG signal. As the algrorithm needs to be used in real time and we want to allow time for RMCP computation and evaluation, and since changes in magnitude of the capillary pulse should be slow, computation of the RMCP more frequently than every 5 s is not needed. The RMCP signal variation increases as the length of summation window is decreased. The length of the summation window, L, is therefore set to 1 min. Every $T_s = 5$ s the signal RMCP, $x(n)$, is therefore calculated according to

$$x(n) = \sum_{k=nK-L+1}^{nK} |p(k)|, \tag{3.1}$$

where $K = 375$ (i.e. 5 s $\cdot$75 Hz), $L = 4500$ (i.e. 60 s $\cdot$75 Hz), and $p(k)$ the baseline filtered PPG signal.

3.2.3 Hypotension Prediction

The RMCP signal (3.1) varies from patient to patient and should therefore be normalized (RMCP_{norm}). The median of the RMCP signal of the first 10 min of recorded treatment is used as the normalization factor. Since the

15

RMCP is not expected to drop, a healthy patient should have a RMCP_{norm} of almost 1 during the the whole treatment (as seen in Figure 3.3).

In [1] it is described how to evaluate if the RMCP signal has had a significant change by using statistical methods. The aim is to maximize the probability for detection given a certain probability of false alarm. The following test statistic is introduced (p. 188 in [1]),

$$G(\overline{x}) = 1 + \frac{1}{N} \sum_{n=0}^{N-1} (|x(n) - x_{med}| - |x(n) - 1|) < \gamma', \qquad (3.2)$$

with N being the number of RMCP values evaluated and x_{med} as the median over the samples evaluated, and decides that hypotension is approching if the value of $G(\overline{x})$ goes below a certain threshold γ'. In [1] it is also shown that $G(\overline{x})$ can be replaced by the function

$$x_{med}(m) = \text{median}\{x(m), x(m-1), ..., x(m-N+1)\}, \qquad (3.3)$$

with $N = 60$, i.e. the median of the last 5 min of the RMCP signal (the last 60 values as RMCP is computed every $5s$), with just as good effect. This saves computational power as (3.3) is much faster to compute than (3.2).

In [1] it was shown that if $x_{med}(m)$ goes below the threshold γ', hypotension will occur. The threshold set to $\gamma' = 0.54$ gave a prediction rate of 100%, and at the same time a false alarm rate of 0% [1]. In Figure 3.4 and 3.5 the method is shown predicting hypotension (compare with a patient with stable blood pressure in Figure 3.3). The Matlab script for prediction algorithm can be seen in appendix A.2.
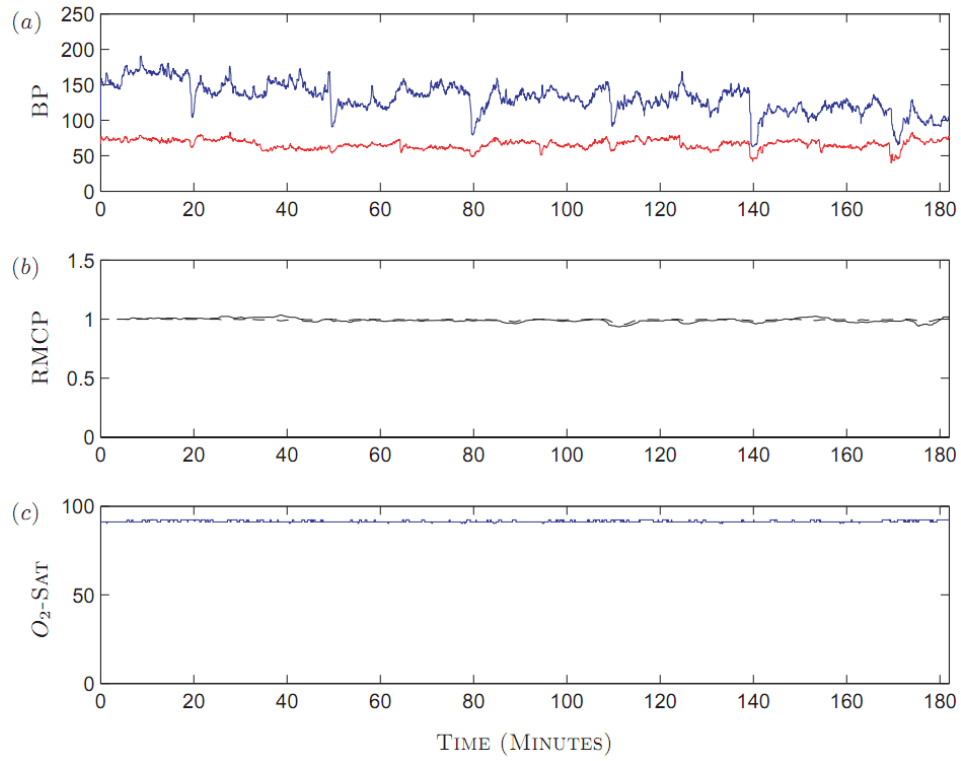
Figure 3.3: *A patient with stable blood pressure during a hemodialysis treatment. (a) Systolic and diastolic blood pressure, BP (mmHg). (b) $RMCP_{norm}$, $x_{med}(m)$ (solid line) and $G(x(m))$ (dashed line). (c) Oxygen saturation, O_2-Sat (%). Reproduced from [1].*
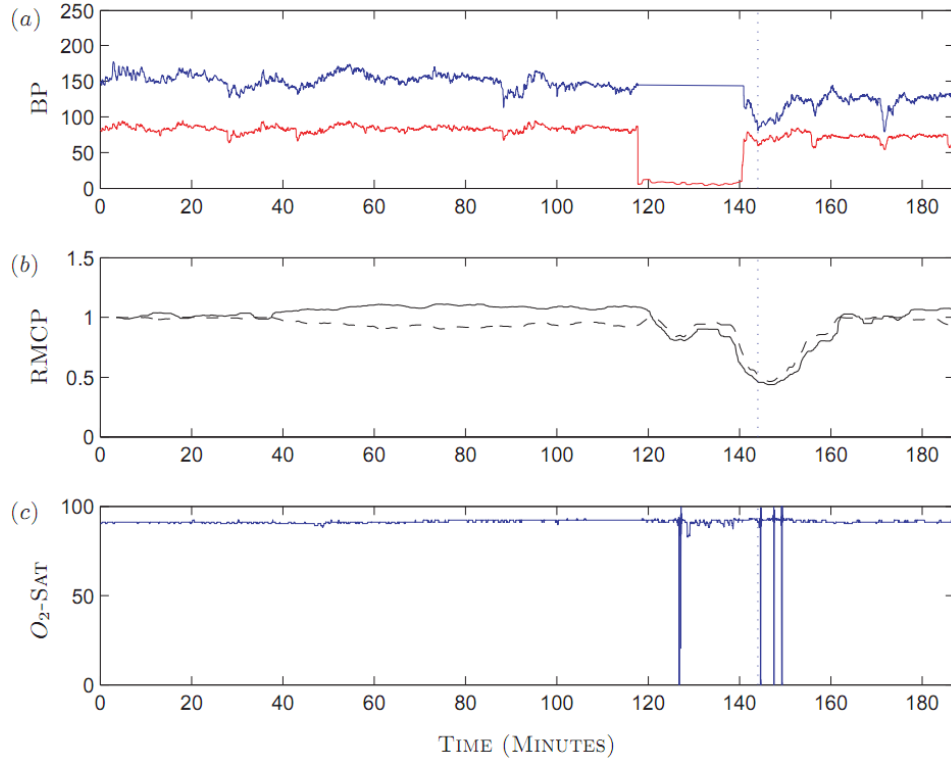
17

Figure 3.4: *A patient during a hemodialysis treatment with hypotension occuring at the dotted line (144 min). (a) Systolic and diastolic blood pressure, BP (mmHg). (b) $RMCP_{norm}$, $x_{med}(m)$ (solid line) and $G(x(m))$ (dashed line). (c) Oxygen saturation, O_2-Sat (%). Reproduced from [1].*
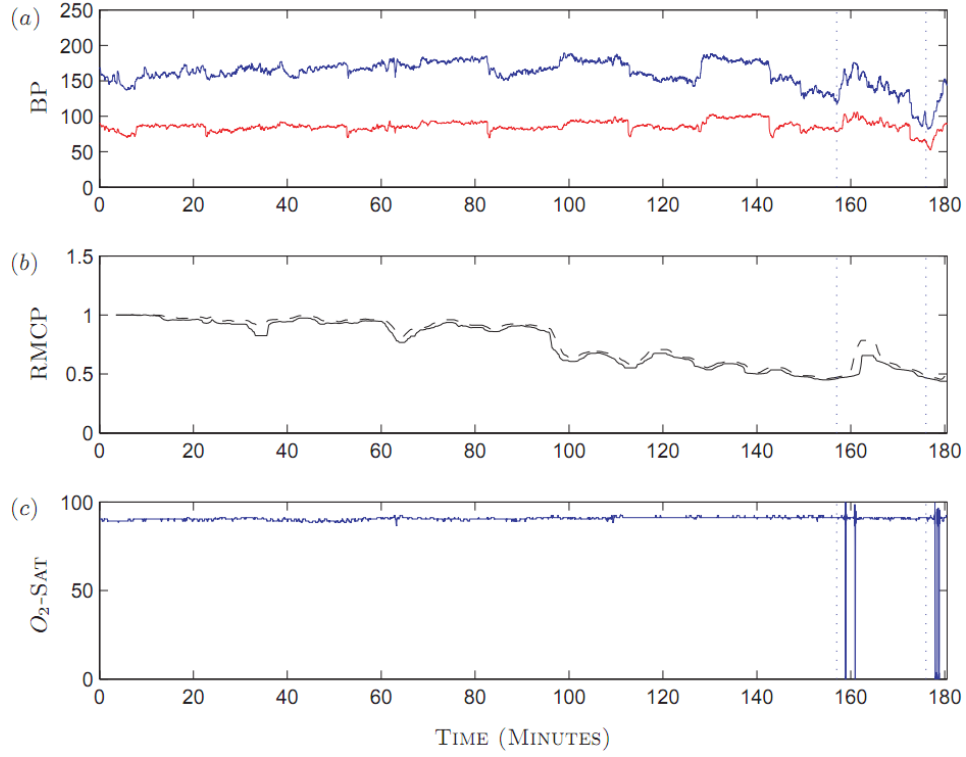
Figure 3.5: *A patient during a hemodialysis treatment with hypotension oc-curing at the dotted lines (157 & 176 min). (a) Systolic and diastolic blood pressure, BP (mmHg). (b) $RMCP_{norm}$, $x_{med}(m)$ (solid line) and $G(x(m))$ (dashed line). (c) Oxygen saturation, O_2-Sat (%). Reproduced from [1].*

19

Chapter 4

Results

In our database the first thing we notice is the unstable nature of the RMCP signal, which still persists after a 5 min median filtration as can be seen in Figure 4.1. The raw RMCP signal is displayed in green with the 5 min median filtration in blue. In the Figure both the systolic (red dashed line) and the diastolic (blue dashed line) blood pressure is displayed in 1/100 th of the mmHg value (e.g. in Figure 4.1 the blood pressure is about 150/60 mmHg). The start and end of the treatments are indicated by vertical purple lines. In the lower part of the Figure the number of missing data points can be seen. With the 5 min median filtration up to 22000 data points can be missing before we notice an effect. The reason for this is the data length of the median filter. With a sample rate of 75 Hz, 5 minutes is equal to 22500 data points. In Figure 4.2 we can see the effect of too much missing data at the 200 and 300 minute mark.

These two Figures are typical for all treatments. Some have large parts of missing data and almost none are nice and smooth like the results that was produced in [1]. The problem is to explain why we can not replicate those results. However, some of irregularities can be explained. In Figure 4.3 we notice a big reduction in the RMCP signal 7 or 8 minutes after the start of the treatment. At this time, according to the treatment notes, the PPG sensor is moved from the index finger to the middle finger. But if the amplitude would be normalized again at this time we would still get a drop below 0.54 at the 50 minutes time mark. Also there are reductions in other treatments without an explanation in the treatment notes.

The first thing that could explain these results is the patient control. In [1] the patient's arm was held in place and the whole treatment was supervised. In this study the patients were free to move around with their

arms. Sometimes resting, reading a book, scratching etc.. These activities could all influence the RMCP signal. The second thing is the sensor type. Although both this and the study in [1] uses Nonin PPG sensors, the sensors were of different types. In [1] a Nonin Medair Life Sense sensor was used while the database in this study used the wireless Nonin 4100.

To test how body position, arm activity and type of sensor influence the RMCP signal, two simple pulse oximeter comparison tests were done.

Figure 4.1: *A patient during a hemodialysis treatment. (top) $RMCP_{norm}$ (green) as a function of time, with the 5 min $RMCP_{norm}$ median (blue) on-top. The Systolic (red dashed line) and diastolic blood pressure (blue dashed line) is displayed as 1/100 th of mmHg. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the treatment (in this treatment the end time was never noted).*
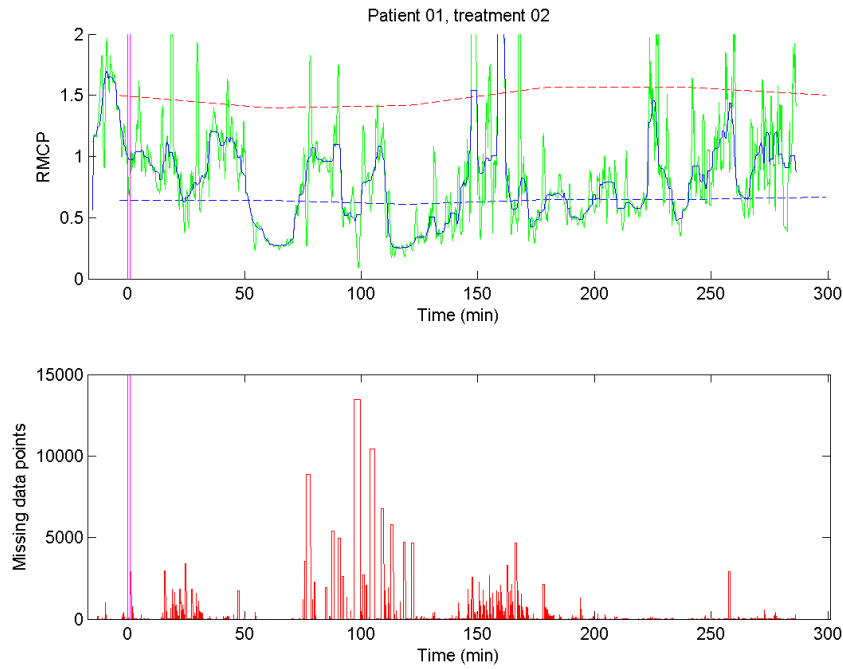
Figure 4.2: *A patient during a hemodialysis treatment. (top) RMCP$_{norm}$ (green) as a function of time, with the 5 min RMCP$_{norm}$ median (blue) on-top. The Systolic (red dashed line) and diastolic blood pressure (blue dashed line) is displayed as 1/100 th of mmHg. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the treatment.*
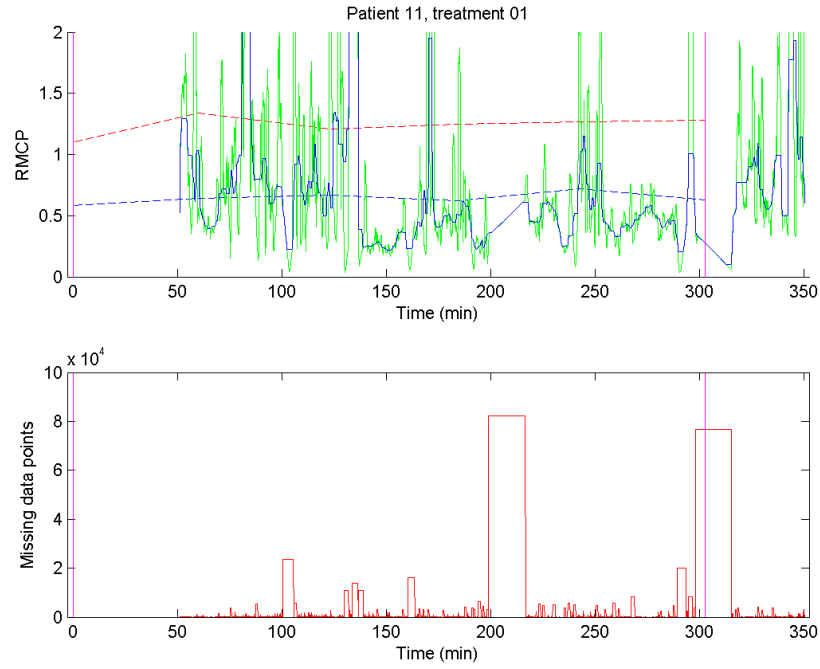
Figure 4.3: *A patient during a hemodialysis treatment. (top) RMCP$_{norm}$ (green) as a function of time, with the 5 min RMCP$_{norm}$ median (blue) on-top. The Systolic (red dashed line) and diastolic blood pressure (blue dashed line) is displayed as 1/100 th of mmHg. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indica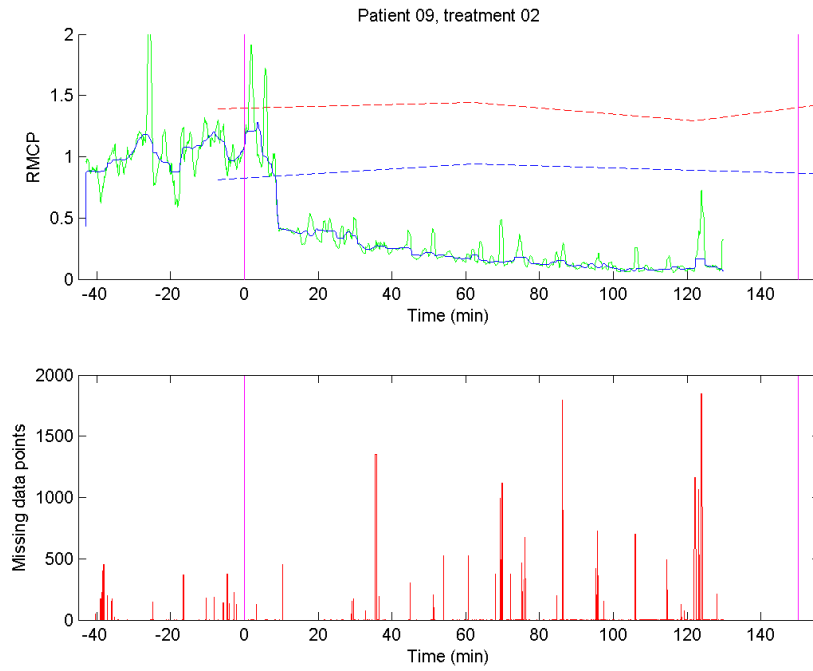te the start and end of the treatment. 8 minutes after treatment start the sensor is moved from the index to the middle finger.*

Chapter 5

Comparing Pulse Oximeters

5.1 Nonin Medair Life Sense vs. Nonin 4100

5.1.1 Method and Equipment

The PPG sensor from the study in [1], a Nonin Medair Life Sense (se Figure 5.1), with a sample rate of 22 Hz as well as the sensor used in this study, the wireless Nonin 4100, were attached to the index finger and middle finger of my left arm. While keeping this arm and myself as still as possible the following activities were donefor 20 minutes each in order:

- sitting (hand on armrest)
- sitting (holding hand high)
- lying down (hand on armrest)

Then with moving arms:

- sitting (restless)
- sitting (scratching on sensors)
- sitting (tinkering with mobile)

The position of the sensors was then swapped to eliminate the effect of finger placement and the whole sequence repeated.

The data was processed and analysed in the same way as described in Chapter 3.

Figure 5.1: *A Nonin Medair Life Sense PPG sensor.*

5.1.2   Results

The resulting RMCP signal for both sensors can be seen in Figures 5.2-5.5. In the first sequence (Figures 5.2-5.3) there is a clear difference between the RMCP signals. The Nonin Life Sense sensor gives a much smoother and expected result (compare with Figure 3.3). The only time there is a reduction in amplitude is after 20 minutes when the arm is raised. Not even tinkering and fiddling with the sensor gives an amplitude change in the RMCP signal. On the other hand the Nonin 4100 sensor's result is anything but expected. It starts quite smooth but as the arm is raised the amplitude increases fast. In the second sequence when the sensors had swapped fingers (Figures 5.4-5.5) we notice that the Nonine Life Sense sensor gives almost the same result as in the first sequence, as expected. The Nonin 4100 sensor however does not. This might indicate that the Nonin 4100 sensor lacks consistency. Also worth noticing is that in both sequences the Nonin 4100 has very few and short periods of missing data, which shows that connection errors are not the cause.

Figure 5.2: *First sequence. (top) $RMCP_{norm}$ (green) as a function of time, with the 5 min $RMCP_{norm}$ median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the test.*

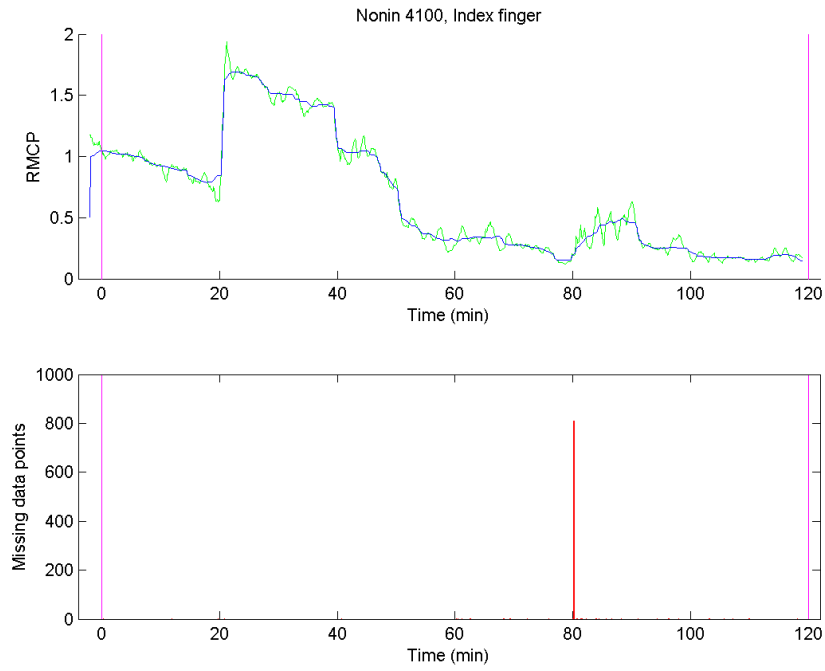Figure 5.3: *First sequence. (top) $RMCP_{norm}$ (green) as a function of time, with the 5 min $RMCP_{norm}$ median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the test. The only major change in amplitude occurs at 20-40 min where the arm is raised.*
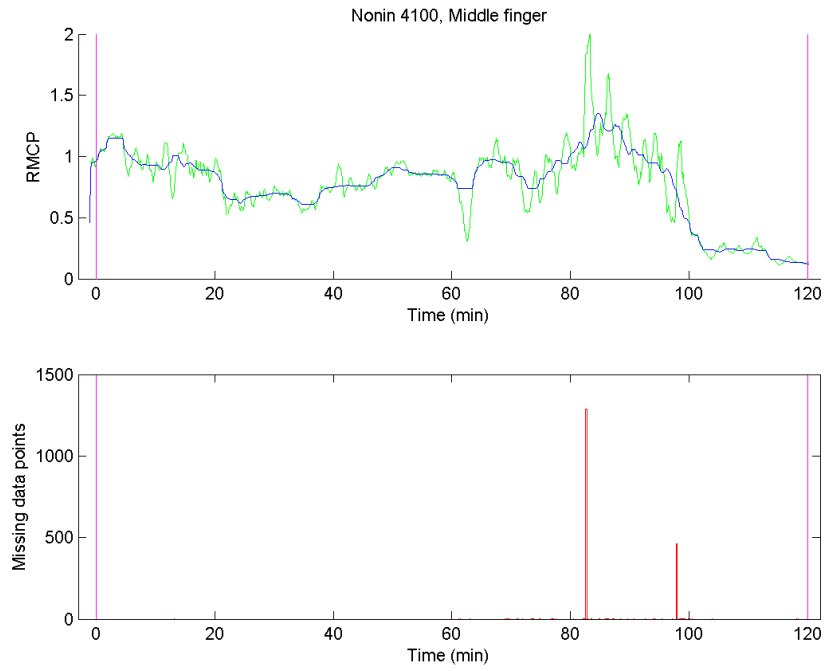
Figure 5.4:   *Second sequence. (top) RMCP$_{norm}$ (green) as a function of time, with the 5 min RMCP$_{norm}$ median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the test.*

Figure 5.5: *Second sequence. (top)* $RMCP_{norm}$ *(green) as a function of time, with the 5 min* $RMCP_{norm}$ *median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the v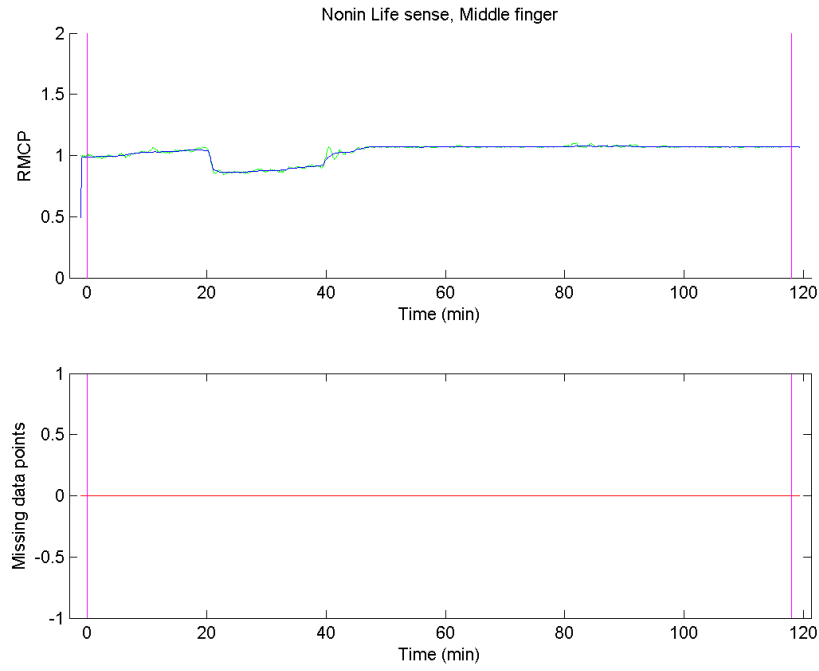ertical purple lines indicate the start and end of the test. The only major change in amplitude occurs at 20 min where the arm is raised.*

30

5.2   Nonin 4100 Consistency

5.2.1   Method and Equipment

Two wireless Nonin 4100 PPG sensors, numbered 1 and 3, were tested for consistency. Sensor number 1 was attached to my index finger and number 3 to my middle finger. While making sure I kept this arm and myself as still as possible I did the following activities in this order:

- sitting 20 min (hand on armrest)

- sitting 20 min (holding hand high)

- sitting 10 min (hand on armrest)

The sensor placement was then swapped and the procedure repeated:

- sitting 20 min (hand on armrest)

- sitting 20 min (holding hand high)

- sitting 10 min (hand on armrest)

The data was processed and analysed the same way as described in Chapter 3.

5.2.2   Results

The RMCP signal for both sensors can be seen in Figures 5.6-5.7. At the 50 minute mark we notice the missing data as the sensor positions are swapped. Appart from amplitude differences in certain areas, the shape of the RMCP signal is clearly consistent (see Figure 5.8) as they both show an increase at the same points. We also notice that the signal amplitude is higher for both sensors when they are attached to the middle finger compared to the index finger. As we noted before, there may be amplitude differences between fingers.

However, the main problem with an unstable RMCP signal can still be seen in both sensors.
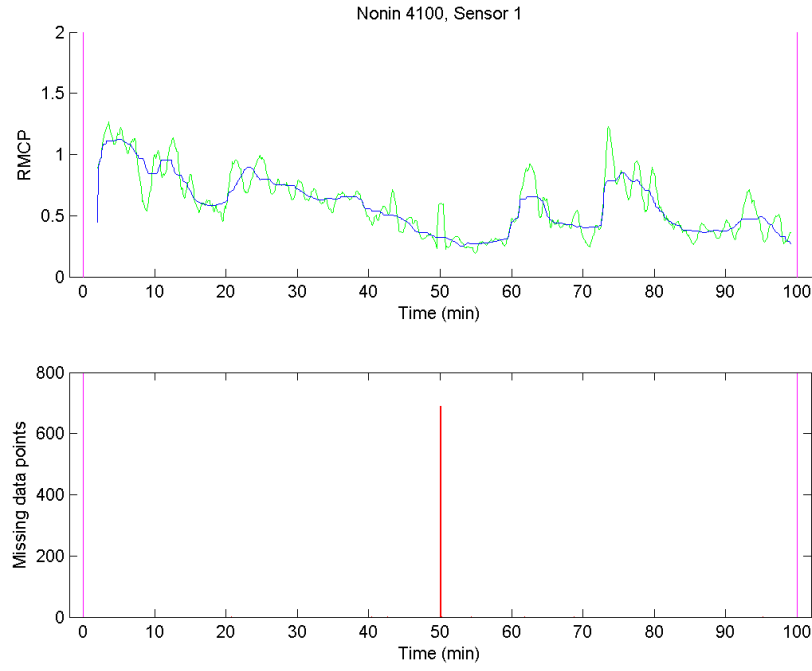
Figure 5.6: *Sensor number 1. (top) RMCP_norm (green) as a function of time, with the 5 min RMCP_norm median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the test.*
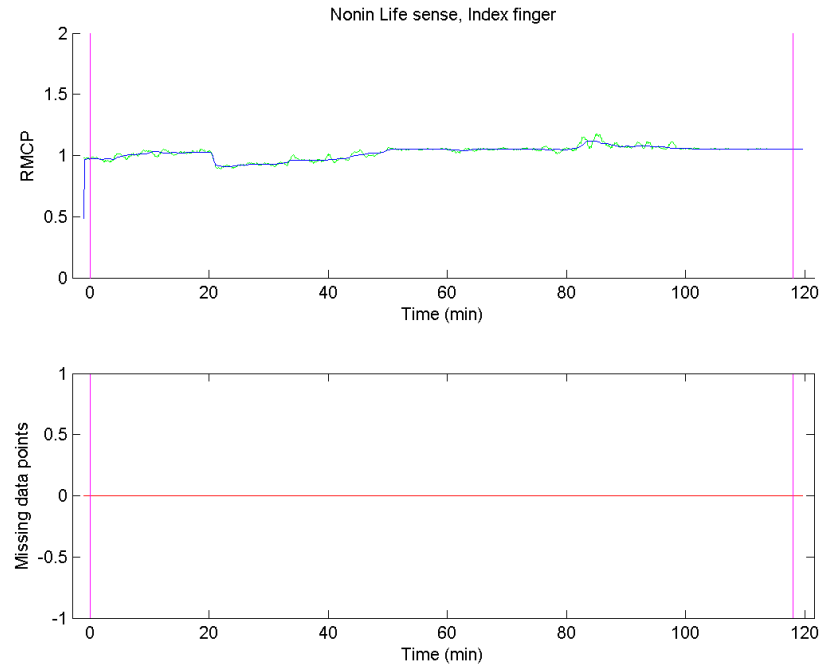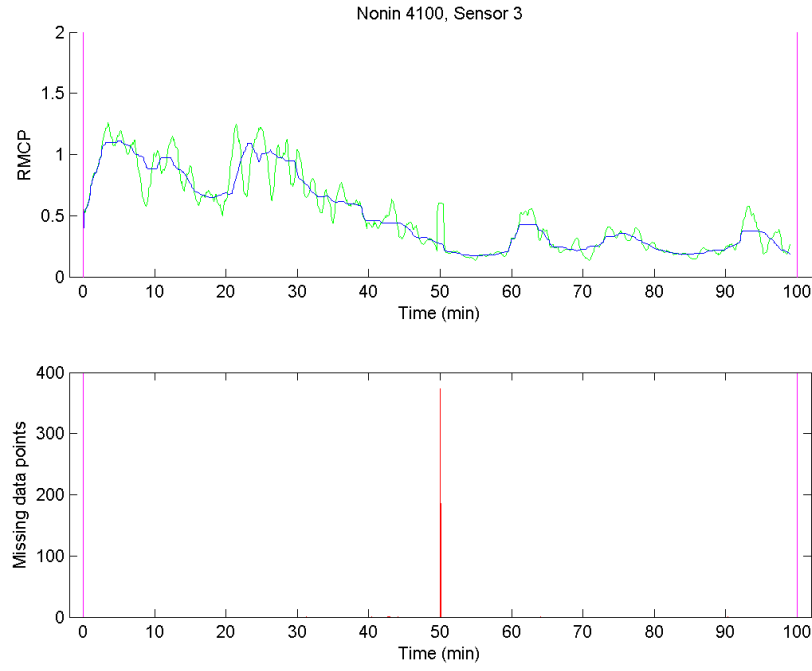
Figure 5.7: *Sensor number 3. (top) RMCP$_{norm}$ (green) as a function of time, with the 5 min RMCP$_{norm}$ median (blue) ontop. (bottom) The number of removed data points in a row as a function of time. In both Figures the vertical purple lines indicate the start and end of the test.*

Figure 5.8: *The RMCP signal from sensor number 1 (blue, dashed) and sensor 3 (red) as a function of time. The vertical purple lines indicate the start and end of the test. The first 50 min, sensor 1 (blue, dashed) is attached to the index finger and sensor 3 (red) to the middle finger. At the 50 min mark the placements are swapped.*

Chapter 6

Discussion and Conclusions

6.1 Discussion

During the work on this thesis it was first very surprising that the results in [1] could not be reproduced. It was hypothesized t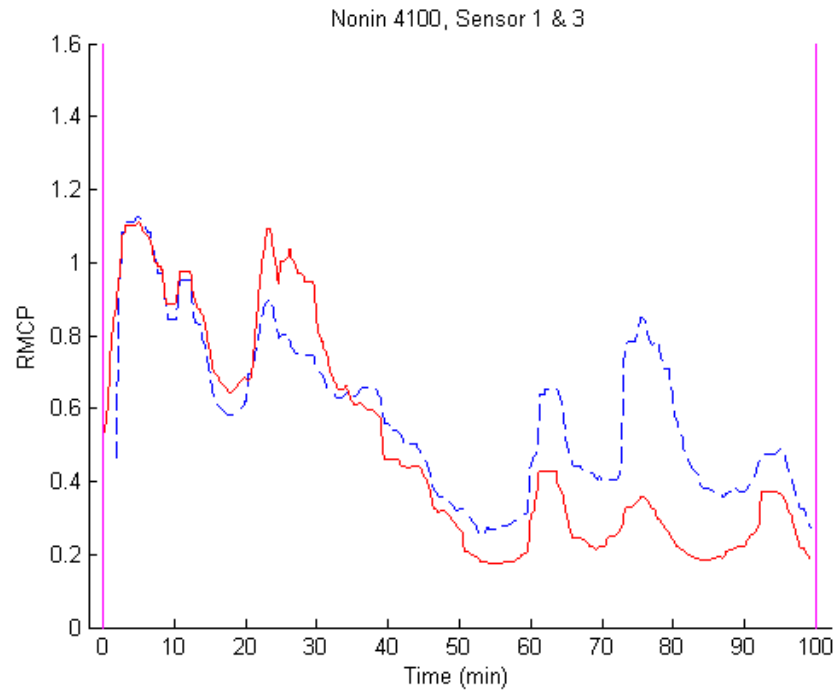hat patient control was the reason for this. As the patients in [1] had been supervised but the patients in this thesis had not, it was believed that the Nonin 4100 sensors were touched, moved and fiddled with too much, making the prediction algorithm unuseable during real treatments. One idea was to re-normalize the RMCP signal every time the sensor was moved, if this was done too often however, the purpose would be countered as amplitude changes in the RMCP signal over longer periods of time is what we are looking for.

After the two comparison tests (see chapter 5), it was clear that if the study had used the Nonin Medair Life Sense sensor the results might have been quite diffrent. With the current results the Nonin 4100 sensor can not be used for prediction of hypotension as it does not give a stable RMCP signal from a healthy patient, not in treatment, just sitting still. There are clear differences between the two sensors, though the nature of those diffrences is currently unknown. Nonin has been contacted regarding the sensor diffrences but no answear has been given yet.

Also, the database in this treatment does not contain a single occurance of acute intradialytic hypotension, although several of the patients were prone for hypotension. Even if the same sensor was used (as in [1]) there would be nothing to predict, making an good evaluation of the method impossible.

6.1.1 Further Study

Since it has been shown that the database used in this thesis could not be used for evaluation of the prediction algorithm, a new database must be gathered. The choice of PPG sensor is very important and the sensor must be doublechecked in advance, making sure it can be used for the intended purpose. Also, the database must contain as many hypotensive episodes as possible, for the method to be evaluated.

6.2 Conclusions

In this thesis it has been shown that the database gathered at Lund University Hospital in 2010 can not be used for the prediction of intradialytic hypotension with the method described in [1]. Because of this, the method itself could not be evaluated. It has also been shown that there are clear differences between the amplitude in the PPG signals from the Nonin Medair Life Sense and the wireless Nonin 4100 PPG sensors, although the cause of this is currently unknown.

References

[1] K, Solem. Signal Modeling and Detection in Nephrologic and Cardiac Application. Ph.D. thesis. Department of Electrical and Information Technology, LTH (2008).

[2] Gambro Education. Gambro (2001). Gambro Basics.

[3] G, Bjerneroth Lindström. Chock. Region Skåne (2010). http://www.1177.se/Skane/Fakta-och-rad/Sjukdomar/Chock/?ar=True. Loaded 111025.

[4] Hypotension. Medical Dictionary (2011). http://medical-dictionary.thefreedictionary.com/hypotension. Loaded 111025.

[5] J, Allen. Photoplethysmography and its application in clinical physiological measurement. Physiol. Meas., vol 28, R1-R38 (2007). (http://iopscience.iop.org/0967-3334/28/3/R01. Loaded 110906)

[6] 4100 Bluetooth Enabled Digital Pulse Oximeter OEM Specification and Technical Information. NONIN Medical Inc (2008).

Appendix A

Matlab Scripts

A.1   extData.m

This script was used to extract the data from the binary files. It could then be saved as .mat files.

```
function dataStr = extData(fname)
% extData(fname) extracts the data from the file 'fname' and returns a data
% structure with the vectors STATUS, PPG, HR, SPO2, TMR and TMRPPG  data.
%
% STATUS, status bytes
% PPG, PPG signal
% HR, Heart rate (gives value 511, if sensor unable to calc)
% SPO2, Oxygen saturation (gives value 127, if sensor unable to calc.)
% TMR, Timer, 1 tick for every package (loops after 2^14 ticks).
% TMRPPG, Timer for PPG, 75 ticks per second (looping removed).
%
% // Erik Wallenborg 2011
%
% TODO: More info can be extracted.


% open file
fid = fopen(fname,'r');

% read data as uint8 (bytes)  (TODO: can malfunction if data length is not a
% multiple of 5...)
a = fread(fid,inf, 'uint8=>uint8');

% Go through all bytes to find each valid synk=1.
% inx will contain the indices to synk=1 in data a.
% Note: the reamining data will have the unsynced parts removed without gaps
% inbetween.
i=1;
k=1;
synkpuls=bitget(a,1);
inx=zeros(length(a),1);
```

```
while i<length(a)-124
    if a(i)>127 && synkpuls(i)==1    % if possible synkpulse found, test next
%25 statusbytes
        if all(a(i+5:5:i+125)>127) && all(~synkpuls(i+5:5:i+120)) &&
synkpuls(i+125)
            inx(k)=i;
            k=k+1;
            i=i+124;
        else
            i=i+1;
        end
    else
        i=i+1;
    end
end
inx = inx(1:k-1);

% Packet and frame indices
ind_packets = inx;
ind_frames = 1:(length(inx)*25);
for i = 1:length(inx)
    ind_frames((((i-1)*25)+1):i*25) = inx(i)+(0:5:124);
end

% create struct to store data in
dataStr = struct('INDX',[],'SNSD',[],'ARTF',[],'OOT',[],'SNSF',[],
'PERF',[],'SYNC',[],'PPG',[],'HR',[],'SPO2',[],'TMR',[],'TMRPPG',[],
'STAT2',[]);

% extract STATUS
STATUS  = a(ind_frames);
dataStr.SNSD   = bitget(STATUS,7);
dataStr.ARTF   = bitget(STATUS,6);
dataStr.OOT    = bitget(STATUS,5);
dataStr.SNSF   = bitget(STATUS,4);
dataStr.PERF   = bitget(STATUS,3)+bitget(STATUS,2);
dataStr.SYNC   = bitget(STATUS,1);

% indx
dataStr.INDX = inx;

% extract PPG waveform
dataStr.PPG = swapbytes(typecast(a(sort([ind_frames(:)+1;ind_frames(:)+2])),
'uint16'));

% extract HR
dataStr.HR = swapbytes(typecast(a(sort([ind_packets(:)+3;ind_packets(:)+8])),
'uint16'));
dataStr.HR = bitset(dataStr.HR,8,bitget(dataStr.HR,9));
dataStr.HR = bitset(dataStr.HR,9,bitget(dataStr.HR,10));
dataStr.HR = bitset(dataStr.HR,10,0);
dataStr.HR = bitset(dataStr.HR,11,0);
dataStr.HR = bitset(dataStr.HR,12,0);
dataStr.HR = bitset(dataStr.HR,13,0);
dataStr.HR = bitset(dataStr.HR,14,0);
dataStr.HR = bitset(dataStr.HR,15,0);
```

```
dataStr.HR = bitset(dataStr.HR,16,0);

% extract SPO2
dataStr.SPO2 = typecast(a(ind_packets(:)+13), 'uint8');

% extract TMR
dataStr.TMR = swapbytes(typecast(a(sort([ind_packets(:)+28;ind_packets(:)+33]))
, 'uint16'));
dataStr.TMR = bitset(dataStr.TMR,8,bitget(dataStr.TMR,9));
dataStr.TMR = bitset(dataStr.TMR,9,bitget(dataStr.TMR,10));
dataStr.TMR = bitset(dataStr.TMR,10,bitget(dataStr.TMR,11));
dataStr.TMR = bitset(dataStr.TMR,11,bitget(dataStr.TMR,12));
dataStr.TMR = bitset(dataStr.TMR,12,bitget(dataStr.TMR,13));
dataStr.TMR = bitset(dataStr.TMR,13,bitget(dataStr.TMR,14));
dataStr.TMR = bitset(dataStr.TMR,14,bitget(dataStr.TMR,15));
dataStr.TMR = bitset(dataStr.TMR,15,0);
dataStr.TMR = bitset(dataStr.TMR,16,0);

%Create TMR for all PPG samples with stamps in seconds, also removes the
%looping.
dataStr.TMRPPG = zeros(1,length(dataStr.PPG));
k=1;
l=0;
for i = 1:length(dataStr.TMR)
    if (i > 3), if ((dataStr.TMR(i-1)-dataStr.TMR(i)) > 10000),l=l+2^14;end
,end %Handle looping of TMR
    for j = 1:25
        dataStr.TMRPPG(k) = ((double(dataStr.TMR(i))-1+l)*25) + j;
        k=k+1;
    end
end
dataStr.TMRPPG = dataStr.TMRPPG/75; %(sample at 75 Hz)
dataStr.TMRPPG = dataStr.TMRPPG - dataStr.TMRPPG(1); % Sets t0 = 0
```

A.2 hypoAnalysis.m

This was the script that implemented the prediction algorithm.

```
clear all
close all

folder = 'PPG_mat';
data = dirr(folder); %(from mathworks, similar to dir(folder))
nbrOfFiles = length(data);

for m = 1:nbrOfFiles

    close all
    load([folder '\' data(m).name]);

    %=== REMOVE FAULTY DATA ===
    PPG2 = zeros(1,length(PPG));
    TMR2 = zeros(1,length(TMRPPG));
    j=1;
    for i = 1:length(PPG)
        stat = SNSD(i) + ARTF(i) + OOT(i) + SNSF(i);% + PERF(i);
        if stat < 1
            if (HR(ceil(i/75)) < 511 && SPO2(ceil(i/75)) < 127)
                PPG2(j) = PPG(i);
                TMR2(j) = TMRPPG(i);
                j=j+1;
            end
        end
    end
    PPG = PPG2(1:j-1);
    PPG = PPG - mean(PPG);
    TMR = TMR2(1:j-1);
    Fs = 75; % sample fq.
    %=====================

    %=== Interpolate gaps with line ===
    diffs = diff(TMR);
    newL = round(Fs*sum(diffs)+1);
    newTMR = zeros(1,newL);
    newPPG = zeros(1,newL);
    k=0;

    for i = 1:length(diffs)
        newPPG(i+k) = PPG(i);
        newTMR(i+k) = TMR(i);
        if diffs(i)*Fs > 1.5
            l = round(diffs(i)*Fs-1);
            for j = 1:l
                newPPG(i+k+j) = nan;
                newTMR(i+k+j) = nan;
            end
            k = k + l;
        end
    end
    newPPG(end) = PPG(end);
```

```
        newTMR(end) = TMR(end);

        %Interp1 gaps
        PPG = newPPG;
        TMR = newTMR;
        bd = isnan(newPPG);
        gd = find(~bd);
        PPG(bd) = interp1(gd,newPPG(gd),find(bd));
        TMR(bd) = interp1(gd,newTMR(gd),find(bd));
        %==========================

        %=== Downsample ===
        y = resample(PPG,3,Fs);
        Fs = 3;
        %==================

        %=== Butter ===
        Wn = 0.5/(Fs/2);  %wn = 0.5 med Fq = 2 Hz, cutoff freq = 0.5 Hz.
        [b,a] = butter(2,Wn,'low');
        y = filtfilt(b,a,y); % forward and backward
        %==============

        %=== Upsample ===
        y = resample(y,75,Fs);
        PPG = PPG - y(1:length(PPG));
        Fs = 75;
        %================

        %== Integral sumation and median ==
        timeHist = 60; %Time hist in s.
        skipL = 5; % skip in s.

        histL = timeHist*Fs; % Frame sample length
        skip = skipL*Fs;
        ampsum = zeros(1,length(PPG));

        j=1;
        for i = (histL):skip:length(PPG)
            ampsum(j) = sum(abs(PPG((i-histL+1):i))); % sum abs
            j=j+1;
        end
        ampsum = ampsum(1:j-1);
        %median
        mT = 5; %last N min.
        ampsummed = medfilt1(ampsum,60*mT/skipL);
        %=================================


        %== Plot Fix ===
        tend = (length(ampsummed)-1)*skipL/60;
        t1 = linspace(1,tend,length(ampsummed)) - treatStart;
        t2 = linspace(1,tend,length(bd)) - treatStart;

        if treatStart < 0 % if start is before recording first 10 is norm
%, else 10 min from start.
            norm = median(ampsummed(1:(10*60/skipL))); % median of first
```

42

```
% 10 min gives norm
    else
        norm = median(ampsummed((treatStart*60/skipL)+1:
((treatStart+10)*60/skipL)));
    end

    xLimits = [min([0 t1(1) bpTimes(1)])-2 max([(tend-treatStart)
treatEnd-treatStart bpTimes(end)])+2];

    fig1 = figure;
    subplot(2,1,1)
    hold on
    plot(bpTimes,bpSys/100,'r--')
    plot(bpTimes,bpDia/100,'b--')
    plot(t1,ampsum/norm,'g')
    plot(t1,ampsummed/norm)
    ylim([0 2]);
    plot([0 0],ylim,'m')
    plot([treatEnd-treatStart treatEnd-treatStart],ylim,'m')
    title([data(m).name '  PPG nbr: ' int2str(PPGNBR)])
    xlabel('Time (min)')
    ylabel('RMCP')
    xlim(xLimits);
    hold off

    j = 0;
    bdl = zeros(1,length(bd));
    for i = 1:length(bd) %fix baddata (bd) for plot
        if bd(i) == 1
            j = j + 1;
        else
            j = 0;
        end
        bdl(i) = j;
    end
    j = length(bdl);
    for i = 1:j-1
        if bdl(j-i) ~= 0 && bdl(j-i+1) ~= 0
            bdl(j-i) = bdl(j-i+1);
        end
    end

    subplot(2,1,2)
    plot(t2,bdl,'r')
    hold on
    plot([0 0],ylim,'m')
    plot([treatEnd-treatStart treatEnd-treatStart],ylim,'m')
    hold off
    ylabel('Missing data points')
    xlabel('Time (min)')
    xlim(xLimits);

    saveas(fig1,[data(m).name '.png']);

end
```