MASTER

Improving the dynamic response of Catheter-Manometer Systems : automatic correction of distorted blood pressure measurements

Habets, R.J.E.

Award date:
1997

Eindhoven University of Technology
Faculty of Electrical Engineering
Division of Medical Electrical Engineering

Improving the Dynamic Response of Catheter-Manometer Systems

Automatic Correction of Distorted Blood Pressure Measurements

by R.J.E. Habets

This report was submitted in partial fulfilment of the requirements for the degree of Master of Electrical Engineering at the Eindhoven University of Technology.
The work was carried out from march 1994 until february 1995 under responsibility of Professor dr. ir. J.E.W. Beneken and under supervision of dr. ir. J.A. Blom at the division of Medical Electrical Engineering.

Summary

The invasive measurement of arterial blood pressure is an important procedure in the care of the critically ill. The most widely used method of invasive blood pressure measurement is the use of a fluid-filled catheter with an external pressure transducer (catheter-manometer system). A problem with these catheter-manometer systems (CMSes) is that they have a poor dynamic response. Because of this poor dynamic response, the high frequencies in the blood pressure signal can be distorted. The CMS can be modelled as a second order system (mostly underdamped). A second order system can be characterized by the natural frequency and the damping factor. These parameters are not constants, but they can change because of air-bubbles and blood clots in the CMS. With the help of a digital adaptive filter the distorted blood pressure signal will be corrected.

The parameters of the CMS can be determined *in vivo* with the fast-flush test described by Gardner. If the correction system is to be accepted in the clinical setting, it must operate fully automatic. To make the parameter estimation process automatic a device has been made, that enables computer-controlled flushing. To make sure the parameters are determined correctly a combination of, at least two, different parameter estimation algorithms are used. For underdamped systems the logarithmic decrement method will be used, in combination with a dynamic parameter estimation algorithm that uses an iterative minimum square error optimisation process. For moderate and heavily damped systems Karnopp's algorithm will be used, also in combination with a dynamic parameter estimation method.

The adaptive correction filter consists of the inverse of the CMS and a low-pass filter. The characteristics of the inverse are determined by the natural frequency and the damping factor whereas the characteristics of the low-pass filter (cutoff frequency) are determined by the frequency content of the blood pressure signal. The correction can be implemented as a single filter, using the bilinear transformation, or as a modular filter. An example of a modular implementation of the filter is the Natural Observation System (NOS). Single filters are more difficult to construct, but they are easier to use once implemented.

Besides the parameter estimation and the correction routines, also a blood pressure analysis algorithm, two filter design routines and a program shell to test all the algorithms on pre-recorded blood pressure signal (from a blood pressure simulator and pigs) have been implemented. All algorithms were tested using these pre-recorded signals, but further *in vitro* and *in vivo* tests will be necessary before this correction system will be suitable for the clinical setting.

Contents

1 Introduction

1.1 General introduction

The invasive measurement of arterial blood pressure is an important procedure in the care of the critically ill. The most widely used method of invasive blood pressure measurement is the use of a fluid-filled catheter with an external pressure transducer (catheter-manometer system). The development of the tipped-catheters eliminated the need for fluid-filled pressure lines. With these tipped-catheters the pressure transducer is inserted into the blood vessel of the patient, which enables more accurate measurements. Tipped-catheters are quite expensive and so they are used several times. Due to the higher cost and the need for sterilisation of tipped-catheters, disposable catheter-manometer systems are more frequently used in the clinical setting. A review of catheter-manometer systems can be found in [Wesseling & van Vollenhoven, 1969].

Based on the work of Otto Frank, who established the initial criteria for adequate pressure recordings [Frank, 1903], Hansen published one of the classic studies on dynamic response and its theoretical and practical expressions [Hansen, 1949]. Hansen recognized that the elastic fluid-filled catheter introduces errors in the blood pressure measurement [Hansen & Warburg, 1950]. It is shown that the dynamic response of catheter-manometer system resembles the dynamic response of a resonant second order system [Vierhout, 1959]. An example of such a system is a mass connected to one end of a spring. If a slow mechanical oscillation is applied to the other end of the spring, the mass will follow this oscillation. When the frequency of the oscillation is increased, it can be noticed that, above a certain frequency, the oscillation amplitude will increase. The amplitude will be maximal at the natural frequency of the spring-mass system. For frequencies above the natural frequency, the amplitude will decrease rapidly and eventually the oscillation stops completely.

A second order system can be characterized by two parameters. The first is the natural frequency. The natural frequency is the oscillation frequency of the system when moving freely (undamped). The second parameter is the damping ratio. This is a measure of how quickly the system comes to rest.

1.2 Project

In the clinical setting most catheter-manometer systems (CMS) are underdamped. If the measured blood pressure signal contains frequencies about and above the natural frequency, the signal will be distorted. This distortion can cause large errors in the blood pressure data (especially the systolic value). The dynamic properties of the measurement system can be improved by adding a digital compensating filter to the CMS. Because the dynamic properties of the CMS can change while using it (due to e.g. air bubbles), the digital compensating filter must be adaptive. The dynamic properties of the CMS (natural frequency and damping ratio) can be determined with the fast flush test [Gardner, 1981].

A compensating scheme can only be successful if it operates without extra attention from the medical personnel. This implies that the parameter extraction must be fully automatic. The goal of our research is to develop a clinically useful correction mechanism that improves the dynamic properties of the CMS. A computer will be used to control the flush device and analyze the flush response, so the parameter extraction will be fully automatic. The automatic parameter extraction and the digital compensation filter will be implemented in a computer program together with a blood pressure monitoring program.

Instead of connecting the manometer to a blood pressure monitor, it will be connected to the computer. The flush device will be connected to an electromagnet that is controlled by the computer. The extra time needed to install the system will depend on the realisation of the auto-flush device. While using the system, no extra attention will be needed.

1.3 About this report

In chapter two the basics of catheter-manometer systems will be explained. Then a model for the CMS will be explained. Finally, the dynamic requirements will be discussed along with the system parameters (the natural frequency and the damping ratio).

In chapter three the estimation of the system parameters will be explained. To begin, a few basic methods of parameter estimation will be explained. Next, *in situ* parameter estimation will be discussed. Then methods to eliminate the influence of the blood pressure signal on the parameter estimation process will be presented. After this, several methods for the actual parameter estimation will be presented. Finally, the accuracy of the estimation and the corresponding estimation strategy are explained.

In chapter four the methods that can be used to correct the dynamic response of the CMS are explained. First, mechanical damping will be explained. Then several electrical compensating schemes will be introduced.

In chapter five all necessary signal processing tools needed in our research are explained. First two digital filter types will be explained. Then two methods for the development of digital filters will be explained. Finally we will discuss the filters needed for our project like differentiators, low-pass filters, high-pass filters and smoothing filters.

In chapter six a blood pressure analysis algorithm is presented. The demands for this algorithm, as well as the implementation and the testing will be described.

In chapter seven the program structure for the blood pressure correction program (MONITOR) will be explained. First the menu structure will be discussed. Then the off-line and the on-line (real-time) model of the program will be presented. Finally, the functional unit structure will be discussed.

In chapter eight the results of the tests with the algorithms are presented. For these tests prerecorded blood pressure signals of a blood pressure simulator were used. First the tests with the parameter estimation and correction routines are discussed. After that, the complete procedure is explained (distortion and correction) with an example. Finally, the 'noise amplification' of the correction process is explained.

Finally, in chapter nine the conclusions and the recommendations will be presented.

2 The Catheter-Manometer System

2.1 Layout of the catheter-manometer system

In Figure 1 a catheter-manometer system (CMS) can be seen, as it can be found in the clinical setting.

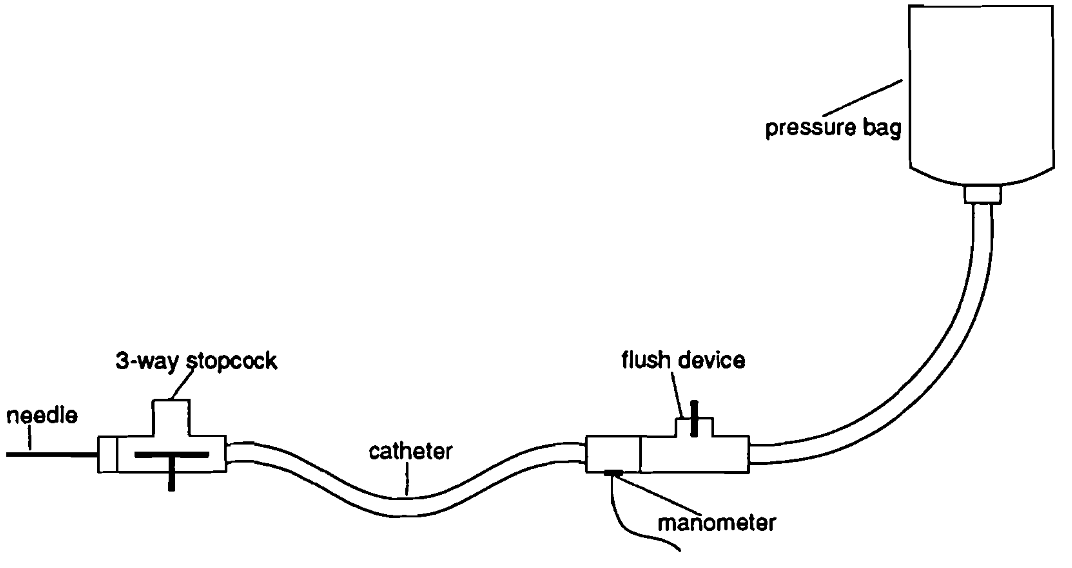


Figure 1 *Schematic representation of a catheter-manometer system.*

This CMS consists of the following parts:

Needle (or cannula) A needle can be made of metal or plastic. The tip of the needle is inserted into the patient's blood vessel. Instead of a needle, a cannula can be used. This is a thin catheter that can be inserted into the patient to measure blood pressure deeper in the vessel.

3-way stopcock A stopcock is used to make the connections between the parts of the monitoring system. Stopcocks can also be used to take blood samples, medicate the patient, and flush the system into open air.

Catheter A catheter is a flexible, transparent, plastic tube filled with saline (an iso-osmotic solution). The length can vary from 10 cm up to 2 metres. Catheters are available with different diameters and stiffness. The stiffness depends on the catheter wall diameter and the catheter material.

Manometer A manometer is a membrane type pressure transducer used to measure the blood pressure. The electrical signal from the manometer can be amplified and displayed on a monitor.

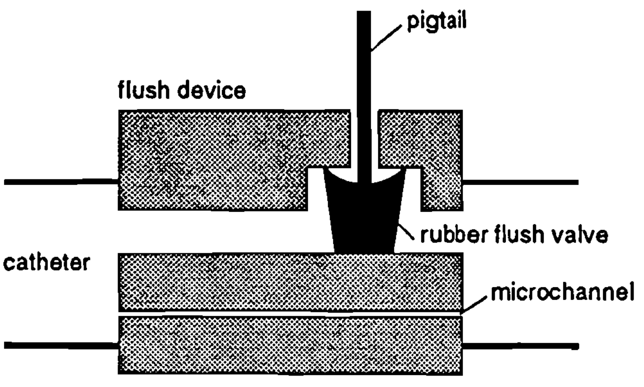


Figure 2 *Cross section of the flush device.*

Flush device A cross section of a flush device can be seen in Figure 2. It has an open microchannel to provide a small continuous flow. This continuous flow is necessary to prevent blood from entering the CMS. The device also has a valve that, when opened, enables a large flow to flush out all blood clots and air bubbles present in the CMS (a so-called fast flush). The valve is opened by pulling the pigtail (see Figure 2). Before a fast flush, the 3-way stopcock (Figure 1) can be turned to prevent that large air bubbles are flushed into the patient.

Pressure bag This bag contains a pressurized volume of saline solution used for the continuous flushing and the fast flush. The pressure in the bag must exceed the maximum blood pressure of the vessel, to ensure a flow directed into the patient.

2.2 Electrical model of the CMS

For the model of the CMS the following analogy will be used:

Electrical		Hydrodynamic	
v	voltage	p	pressure
i	current	q	flow
Q	charge	V	volume
t	time	t	time
R	resistance	R	flow resistance
L	inductance	L	inertance
C	capacitance	C	compliance

A frequently used model for the CMS is based on the π-filter model [Vierhout, 1959]. The model for the complete CMS can be seen in Figure 3. This catheter model is explained in the M.Sc. report by Jos Alofs [Alofs, 1993].

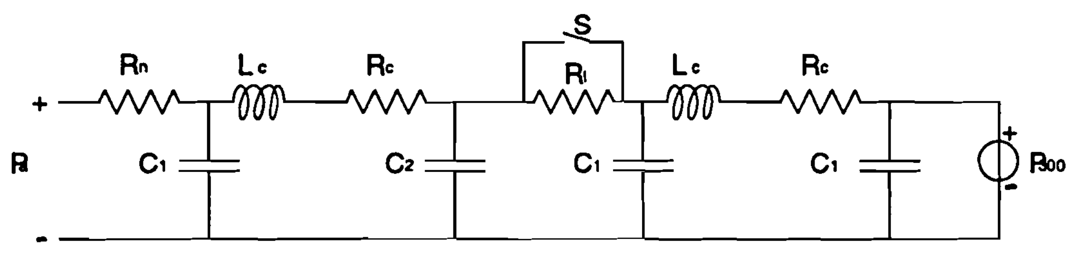


Figure 3 Electrical model of the CMS.

with:

$C_1 = C_c/2$

$C_2 = C_c/2 + C_m$

C_c catheter compliance

C_m manometer compliance

L_c catheter inertance

R_c catheter flow resistance

R_n needle resistance

A simplified model for the needle, catheter and the manometer can be seen in Figure 4.

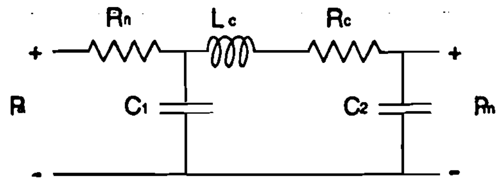


Figure 4 *Third order model of needle, catheter and manometer.*

The transfer function for this model is:

$$H_{cms}(s) = \frac{1}{s^3 R_n C_1 L_c C_2 + s^2 [L_c C_2 + R_n C_1 R_c C_2] + s[(R_n + R_c) C_2 + R_n C_1] + 1} \tag{1}$$

This third order transfer function can be simplified even further, under the condition that only frequencies up to twice the natural frequency are of importance. The simplified transfer function is:

$$H_{cms}(s) = \frac{1}{s^2 [L_c C_2 + R_n C_1 R_c C_2] + s[(R_n + R_c) C_2 + R_n C_1] + 1} \tag{2}$$

The transfer function $H_{cms}(s)$ in the standard notation for a second order system is:

$$H_{cms}(s) = \frac{\omega_n^2}{s^2 + 2\beta \omega_n s + \omega_n^2} \tag{3}$$

with:

$$\omega_n = 2\pi f_n = \sqrt{\frac{1}{L_c C_2 + R_n C_1 R_c C_2}} \tag{4}$$

$$\beta = \frac{[(R_n + R_c) C_2 + R_n C_1]}{2\sqrt{L_c C_2 + R_n C_1 R_c C_2}} \tag{5}$$

The damping factor (β) and the natural frequency (f_n) are the parameters that will be used to characterize the CMS. The relative amplitude (magnitude) and phase of $H_{cms}(s)$, for β values from 0.1 to 1.0, are presented in Figure 5.

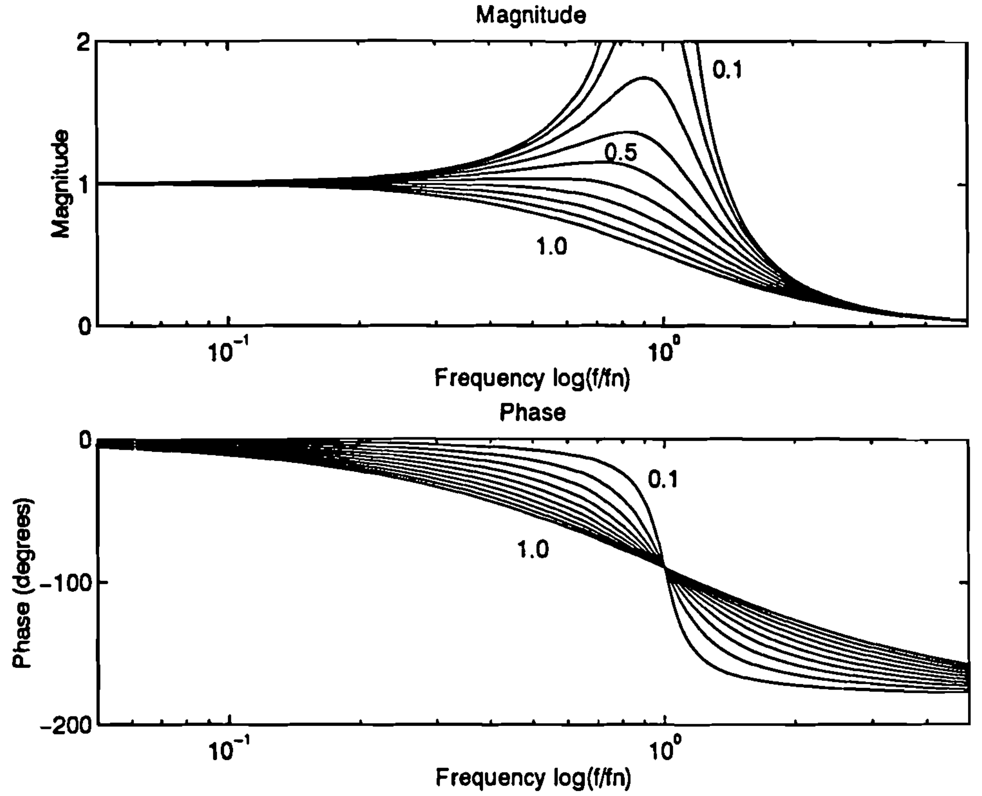


Figure 5 *Magnitude and phase of the transfer functions of a second order system for β from 0.1 to 1.0.*

Yeomanson described a more detailed model [Yeomanson & Evans, 1983]. He experimentally determined transducer compliance, manometer line compliance and hysteresis losses and calculated viscous resistance and inertance components. This model was shown to be adequate to at least 100 Hz. For our project the second order model will suffice.

2.3 Dynamic requirements

Most catheter-manometer systems used in the clinical setting can be modelled by an underdamped second order system. It was experimentally shown that the CMS follows this second order model for frequencies up to about 50 Hz [Gardner, 1981]. If the measured blood pressure signal contains frequencies around and above the natural frequency, the signal will be distorted. This distortion can cause large errors [Rothe & Kim, 1980]. The dynamic requirements for a catheter-manometer system can be expressed as a frequency bandwidth [Geddes, 1970; Paulsen, 1993] or by specification of the natural frequency and damping coefficient [Gardner, 1981; de Haas et al., 1994].

The dynamic requirements for a CMS depend on the frequency contents of the arterial blood pressure signal. Below an overview will be presented of the dynamic properties of the arterial blood pressure signal (at a peripheral measuring site) as found in the available literature.

- Frank stated that BP signal contains frequencies up to 30 Hz [Frank, 1903].

- Geddes stated that a uniform frequency response was needed of at least 6 and preferably to 15 times the heart rate. For a heart rate of 180 BPM, the BP signal would contain frequencies up to 45 Hz [Geddes, 1970, 1988].

- Paulsen did an extensive worst case study of several blood pressure waveforms in relation to the measurements that were done [Paulsen, 1993]. The maximum error he accepted was 5% or 1 mmHg whichever was greater. When measuring SBP and DBP at a peripheral measuring site, 2 harmonics and appropriate low-pass filtering are required. Near the heart 5 harmonics are needed to determine SBP and DBP. For the measurement of dP/dt at a peripheral measuring site, 20 harmonics are needed. Near the heart, 22 harmonics are necessary for this measurement. For a heart rate of 180 BPM the BP signal would contain frequencies up to 60 Hz at peripheral measuring sites and up to 66 Hz near the heart.

Given all this information, it can be concluded that the dynamic requirements are not constant but depend on the heart rate, the measuring site and the measurements done. Considering these dynamic requirements and the fact that the second order model holds for frequencies up to 50 Hz, we will use 50 Hz as the maximum meaningful blood pressure frequency.

For each situation it is possible to give a frequency up to which the frequency response of the CMS must be 'flat'. This condition can also be expressed by means of a pair (f_n, β). It can be easily understood, that there are more pairs, that meet the demand that the frequency response is 'flat' up to a certain frequency. For example, a critically damped system (β=0.7) the frequency response is 'flat' (within 5%) up to 60% of the natural frequency. With an underdamped system with β=0.2 the frequency response is 'flat' (within 5%) only up to 22,5% of the natural frequency. The optimal situation is reached at β=0.59, which has a 'flat' frequency response up to 87% of f_n [Hipkins et al., 1989].

Gardner [Gardner, 1981] developed a (β, f_n) diagram in which a triangular area denotes the area of adequate dynamic response of the CMS for a given situation (heart rate, measuring site and the type of measurement). If the β and f_n of the CMS are determined, the diagram will tell whether the measurement is reliable (undistorted) or not.

2.4 System parameters

As stated earlier, the CMS can be modelled as a second order system and thus it can be characterized by the natural frequency and the damping factor.

The following factors increase the natural frequency (the magnitude of change is indicated in parentheses) [Brunner, 1978]:

- increase of the tubing radius (proportional)
- reduction of capacitance by reduction of the transducer compliance or by an increase of tubing stiffness (as the square root)
- lower fluid density in the tubing (as the square root)
- shorter length of tubing (as the square root)

The following factors increase the damping:

- increase in fluid viscosity (proportional)
- decrease of the tubing radius (as the third power)
- increase of tubing length (as the square root)
- increase of system capacitance (as the square root)

The tradeoff between a practically usable system and the parameters is made by the CMS manufacturer. More about model parameters and their relation to β and f_n can be found in [Shapiro & Krovetz, 1970; Taylor et al., 1986, Taylor & Donovan, 1992].

One of the major problems with a CMS is that the parameters are not constant. Blood clots in the needle and air bubbles in the system cause the parameters to change. To prevent clotting of the needle a small, constant flow is maintained into the patient. Air bubbles are formed when temperature variations cause gas to come out of solution. Air bubbles come in different sizes. Besides the visible bubbles, there also exist micro-bubbles, which are invisible to the human eye. Micro-bubbles are no threat to the patient because they dissolve in the patients' blood, which has a higher temperature than the saline. Large air bubbles can be removed with the flush unit of the CMS by opening the 3-way stopcock and then opening the flush-valve until the air bubbles are flushed out.

An air bubble causes an increase of capacitance. The effects of this increase in capacitance are an increase of damping and a decrease of natural frequency. An increase of damping would be preferable, but the decrease of natural frequency causes the high frequency response to deteriorate significantly.

Recent tests of CMS [de Haas, 1994; van Langen et al.,1993; Heimann & Murray, 1993] have shown that available CMSes, in laboratory settings, can have natural frequencies from 12 to 64 Hz and damping factors from 0.1 to 0.43. *In situ* tests of CMSes have shown natural frequencies from 8.5 to 41 Hz and damping factors of 0.14 to 0.73. Van Langen also found four CMSes, which had no oscillations in the step response. This shows that, *in situ*, overdamped systems can occur.

3 Estimation of the System Parameters

From the previous chapter it can be concluded that it is very important to know the frequency response of the CMS. The most commonly used methods of frequency response testing for second order systems will be discussed. Methods will be presented that deal with underdamped as well as critically damped or overdamped systems. First some basic methods will be explained. Next *in situ* response testing will be looked upon. After that, the influence of the blood pressure on the response testing is investigated. Then a set of methods will be introduced. The usefulness of these methods (for a real-time and on-line system) will be evaluated. Finally, a parameter estimation strategy will be presented.

3.1 Basic methods

3.1.1 Frequency-sweep testing

With this method (which is also known as the standing wave analysis) a variable frequency oscillator is connected to the input of the CMS (at the side of the needle). This oscillator generates a sinusoidal frequency sweep from zero to some maximum frequency (greater than the natural frequency of the system under consideration). At the system output (manometer) the frequency transfer function can be measured [Gardner, 1981]. The input and output signals can be seen in Figure 6. From this frequency transfer function the damping and the natural frequency can be determined.

The frequency sweep method can be used to produce transfer functions of systems of any order. For a second order system the following formulas can be used to calculate the β and F_n [Taylor, 1990]:

$$\beta = \sqrt{\frac{1 - \sqrt{1 - \frac{1}{A^2}}}{2}} \qquad (6)$$

$$F_n = \frac{F_r}{\sqrt{1 - 2\beta^2}} \tag{7}$$

In which A is the gain at resonance and F_r is the damped resonant frequency.

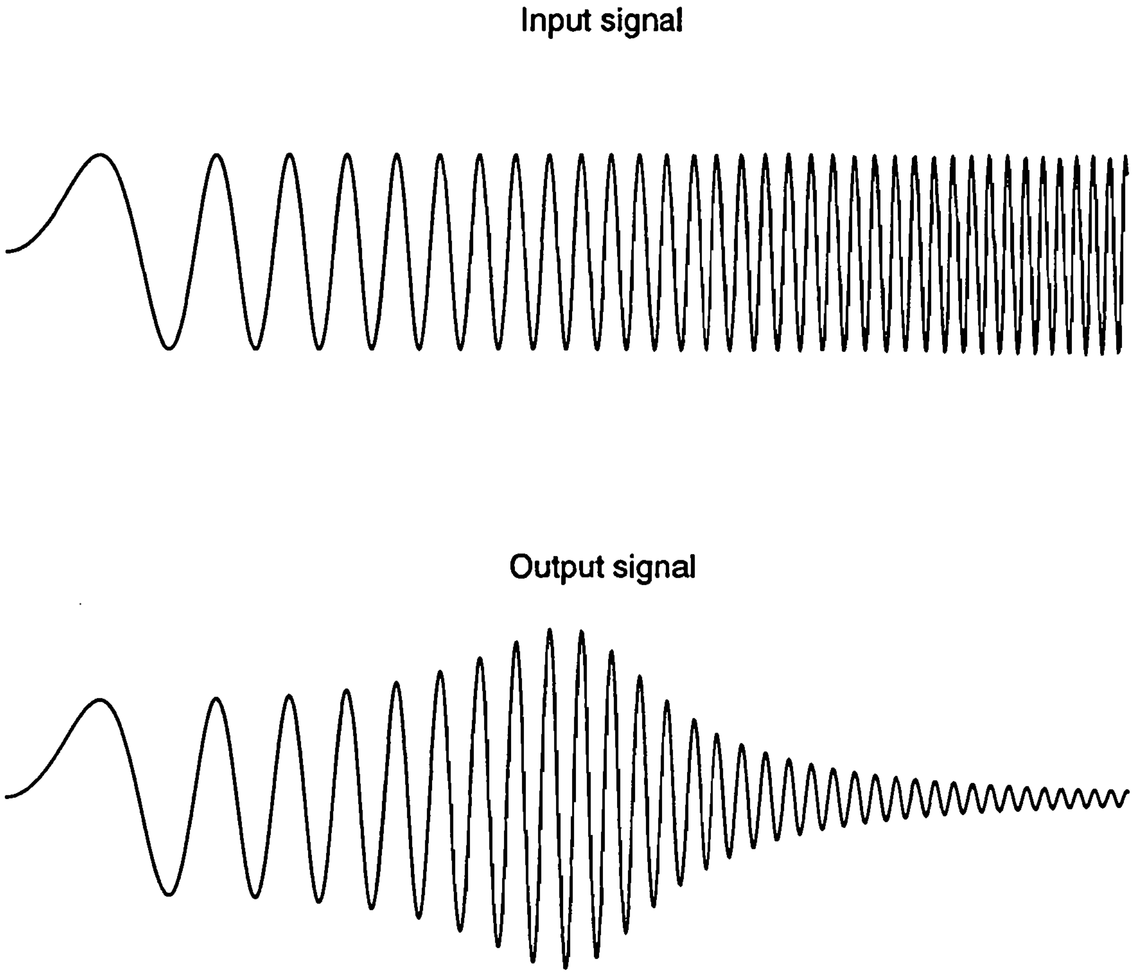


Figure 6 *Frequency sweep method (signals at the input and output of the CMS)*

The pressure oscillator used in this method is a rather complicated instrument. The frequency sweep method is very suited for laboratory experiments, because it produces the transfer function, of any system, directly. An example of an oscillator for these purposes can be found in the thesis of Vierhout [Vierhout, 1966] (also see [Vierhout & Vendrik, 1965]).

A method that resembles this standing-wave analysis technique was developed by Latimer. Latimer studied the problem of measuring the transmission characteristics of liquid-filled tubes at sub-audio and audio frequencies. The method enables the measurement of attenuation and phase shift

[K. Latimer & R. Latimer, 1969]. Latimer's method cannot be used for our research because it cannot be done automatically. Besides that, we are only interested in frequencies up to about 50 Hz (the information in the blood pressure signal).

3.1.2 Step response and impulse response testing

When a pressure step is presented to the input of the catheter, the output will show a similar pressure step with a damped oscillation on top of it (see Figure 7). In case the system is not underdamped, a response like in Figure 8 can occur also. From the step response the system parameters (β and f_0) can be extracted.

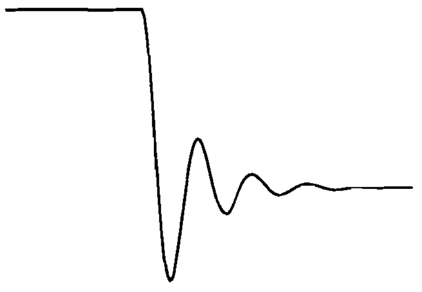


Figure 7 *Step response with* $\beta = 0.2$.

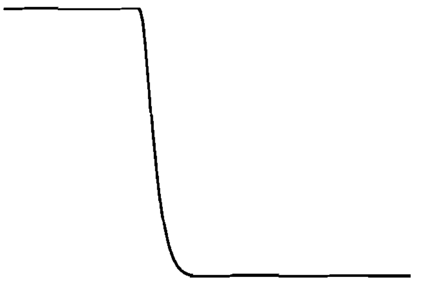


Figure 8 *Step response with* $\beta = 0.9$.

An ideal pressure impulse should be infinitely narrow (dirac pulse), but this is not possible in practice. A practical impulse will have a finite width. This impulse response can also be used to extract the system parameters. Several ways in which the parameters can be extracted from the step-response or the impulse-response will be discussed in paragraph 3.5

3.2 Response testing *in situ*

3.2.1 Methods

In the clinical setting the frequency sweep method cannot be used because the input of the CMS (needle) is inside the patient. Frequency generators that can be inserted into a CMS are not yet available (because there is no widespread need for them).

Step testing can be realized more easily. In laboratory settings Gabe [Gabe, 1972] developed the pop-test (also used by Boonzaier [Boonzaier, 1978]). In this method a liquid filled chamber is sealed with a balloon and a rubber ring. Via a catheter the chamber is connected to a pressure transducer. With a pump (for example a syringe) the balloon is inflated and the pressure in the chamber increases. Then the balloon is popped with a hot needle. This causes a pressure step in the chamber, and thus at the input of the catheter. This method cannot be used *in situ* measurements.

In 1970 Gardner described the use of a Catheter-Flush system for continuous monitoring of central arterial waveforms [Gardner et al., 1970]. Eleven years after this research, Gardner introduced the Fast-Flush test [Gardner, 1981]. This test makes use of the flush device that connects the pressure bag to the catheter. When the valve of the flush device is opened, the measurement system is exposed to high pressure. When the valve snaps shut, a pressure step is applied to the system. The response to the pressure step can be used to extract the system parameters. The pressure step generated this way is not applied to the system input, but at the valve, which is close to the pressure transducer (system output). Gardner did not prove that his test activated the whole system, and that the parameters extracted were valid to describe the whole CMS. Many have challenged the assumption that the fast flush test is a valid step response test [Rothe, 1980; Hipkins, 1989; Taylor, 1990; Sheahan et al., 1991; Billiet & Colardyn, 1992].

De Beer applied system analysis theory to show that the system transfer function of the CMS can be found from the parameters β and f_n, which are determined with the fast flush test [de Beer, 1984]. Kleinman concluded from his research that the fast flush test activates the whole monitoring system and that f_n and β are the same throughout the system including the distal catheter [Kleinman et al., 1992]. The fast flush test (FT) now is generally accepted for dynamic testing of a CMS. An important advantage of the FT is that no extra equipment is needed to do the test.

The flush device can also be used to create a pressure impulse. For an impulse the valve should be opened very briefly. Because the flush device is operated manually, a pressure impulse generated this way would not be ideal nor reproducible. Opening the valve takes longer than letting it snap shut, so the impulse would not even be a symmetric pulse. Taylor examined the possibility to create an impulse with the pressure transducer itself. He showed that this method was not reliable [Taylor, 1990].

3.2.2 Automatic flushing

As stated in the first chapter, the response testing must be automatic (performed under control of a computer). For this purpose, a device has been made that can operate the valve of a flush device automatically [Alofs, 1993]. The pigtail (of the flush device) is connected to an electromagnet, that can be activated by the computer. This auto-flush device also has a button which enables the flush-valve to be opened manually.

Computer-controlled flushing has some important advantages, but also a few disadvantages. The advantages are:

- Automatic response testing.
- The ability of rapidly opening and closing the flush-valve introduces the possible use of a pressure impulse instead of a pressure step.
- Reproducibility of the excitation (step or impulse).
- There is no need for an algorithm that detects a flush (and its response) in the blood pressure signal, because the timing of the flush is known.
- The ability to control the time the flushing takes place, enables us to control the position of the flush-response relative to the blood pressure signal.

The disadvantages are:

- Extra time is needed to install the auto-flush device. This time needs to be minimized.
- Connecting the auto-flush device to the flush device will be difficult, because there are many very different types of flush devices used in the clinical setting. Focusing on one specific type might be a solution.

In a flush-response test the saline is flushed into the patient. Therefore, the flush valve must be opened as briefly as possible. The flush-time will be chosen so that the oscillation of the rising flank of the flush just died out. For a worst case of f_n=5 Hz and β=0.1 a flush-time of 0.3 second is adequate. If it is not possible to flush into the patient, the three-way stopcock must be turned. To keep the parameter estimation automatic, the stopcock must be turned automatically. Marks suggested a method for computer control of stopcocks [Marks et al., 1988]. Because the amount of saline flushed into the patient is very small with a flush-time of 0.3 seconds, no automatic stopcocks are probably needed for our project, except maybe for infants.

3.3 Blood pressure influence

With *in situ* measurements the flush response is superimposed on the blood pressure signal. The blood pressure signal will distort the response and thus introduce errors in the parameter estimation. In the next paragraphs two methods will be introduced that diminish the blood pressure influence on the parameter estimation.

3.3.1 Placement method

The simplest way to reduce the influence of the blood pressure on the step-response, is to place the response in the diastolic downslope of the blood pressure signal ('stable' part). For correct placement we need to know when the 'stable' part of the blood pressure signal begins. This information is provided by the blood pressure analysis algorithm. This algorithm, which is described in chapter 6, can produce trigger points at 25%, 50%, and 75% of the upstroke during the systolic part of the blood pressure. The position of the dicrotic notch relative to the start of the upstroke can be estimated as (360-HR) ms. The duration of the upstroke is fairly independent of the heart rate and is about 100ms. The position of the dicrotic notch relative to the 75% upstroke now is (285-HR)ms. This point is chosen as the start of the 'stable' region. An other approach to determine the 'stable' part of the signal is to look at the 25% upstroke trigger. The 'stable' part can be chosen to be the part, in which the blood pressure signal is below the 25% level. This situation corresponds to a 'state' of the blood pressure analysis algorithm.

The 'stable' part of the blood pressure signal can be modelled by three possible approaches:

- Constant approach: the diastolic part is considered constant.
- Linear approach: the slope of the diastolic part is considered constant.
- Quadratic approach: the second derivative of the diastolic part is considered constant.

Remarks

- The algorithm used to extract the parameters from the response must be able to deal with all three approaches.
- The length of the diastolic part must be long enough to contain the response (at least a part of the response that is long enough to extract the parameters from).
- All algorithms must be real-time because placement only works when there is no delay between actual and processed blood pressure signals.

- To keep the differentiation process stable the sample frequency must be high enough. To get a reliable derivative with the selected differentiation filter five or more samples per period would be needed. In the case of a maximum natural frequency of 64 Hz this would mean a minimal sample frequency of about 320 Hz. In the clinical setting (natural frequencies up to about 40 Hz) a minimal sample frequency of 200 Hz would be recommendable. For more information about the sample frequency see paragraph 3.4.

3.3.2 Subtraction method

The placement method only functions properly if the 'stable' part in the blood pressure signal is long enough to hold the response. If the heart-rate is high and the response is long (low natural frequencies), the response will not fit into the 'stable' part. In the subtraction method a BP signal part without a flush-response (before the flush) is subtracted from a BP signal part with a flush-response. The result will be the flush-response if, and only if, the blood pressure is periodical for the two BP parts used. The subtraction method is explained in Figure 9.

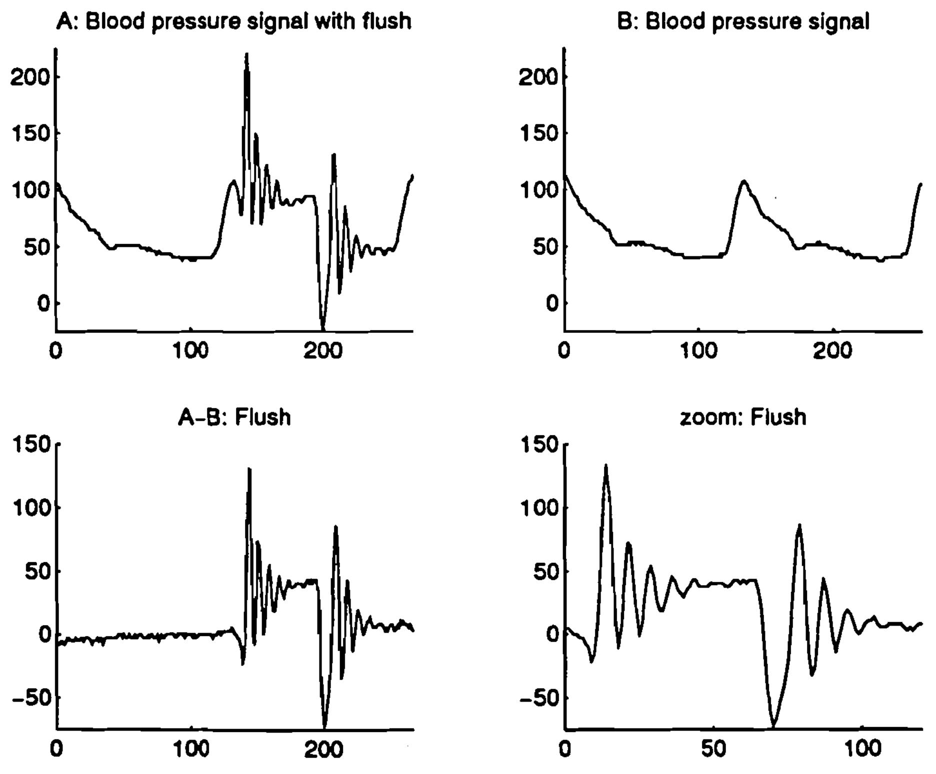


Figure 9 *Subtraction method. BP without flush-response (B) is subtracted from a BP with a flush-response (A). The result (A-B) can be seen as well as the actual flush-response (enlarged).*

Remarks

- The blood pressure must be periodical for some time (around the time of the flush test). Information about the period of the blood pressure signal (heart rate) can be obtained with the blood pressure analysis procedure. Besides the period also the shape of the pressure curve must be 'stable'. Information about the shape can be obtained via the systolic, diastolic and mean pressure values.
- If the flush response does fit in the 'stable' part of the blood pressure signal the subtraction method can be used in combination with the placement method constant approach. If the subtraction process is not optimal and trends (like respiratory effects) would remain present in the subtracted signal, the linear or quadratic approach can be used.
- The two blood pressure signal parts must 'fit'. For the alignment of the two signals the trigger points of the BP analysis algorithm can be used.
- The length of the signal parts will depend on the length of the flush-response. Make sure to use a whole number of BP periods (for easy fit).
- Instead of using two BP signal parts in the subtraction process, also three parts can be used. In this case parts without a flush are used from before and after the flush. The periodicity can now be tested for each of these periods and then be compared. If the periods of the two parts match (stable period) the average is subtracted from the BP signal part with a flush-response.

3.4 Sample frequency

The Nyquist theorem requires that the sample frequency is at least twice as high as the highest signal frequency. In this way it is guaranteed that we do not miss any periods of the highest signal frequency (no aliasing). As discussed earlier the blood pressure signal contains frequencies up to 50 Hz. For a blood pressure monitor a sample-frequency of 100 Hz would suffice. If a maximal heart-rate of 240 BPM (4 Hz) is assumed, every blood pressure period contains at least 50 samples. For a blood pressure analysis algorithm these 50 samples are enough to determine the maximum and minimum pressure values (and thus the systolic and diastolic pressure).

In the response tests a maximum natural frequency of 64 Hz (see paragraph 2.4) was found. To sample the flush response a sample frequency of 128 Hz would suffice. For the analysis of the response, algorithms are needed that determine the minima and maxima of the flush response (a decaying oscillation). For the determination of the extrema two samples per period (Nyquist) will not suffice.

The maximum error in the detection of an extreme of a sampled sinusoidal signal occurs when the extreme lies exactly between two samples see Figure 10. The extreme is located between the samples i and i+1. The determined extreme in this case would be the sample-value at sample i and thus an error e_{max} would be made.

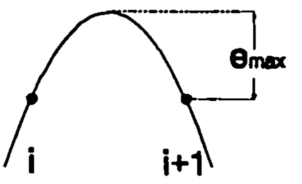


Figure 10 *Definition of the maximum error in the determination of the extreme.*

If the signal is sampled with f_s Hz, the percentile error in the extreme is:

$$e_{max}\,(\%) = \left(1 - \sin\left(\frac{\pi}{2} + \frac{T}{2}\right)\right) \cdot 100\% \qquad (8)$$

with $T = f_s^{-1}$.

The maximum error is depicted in Figure 11 (for f_s = 100 to 500 Hz).

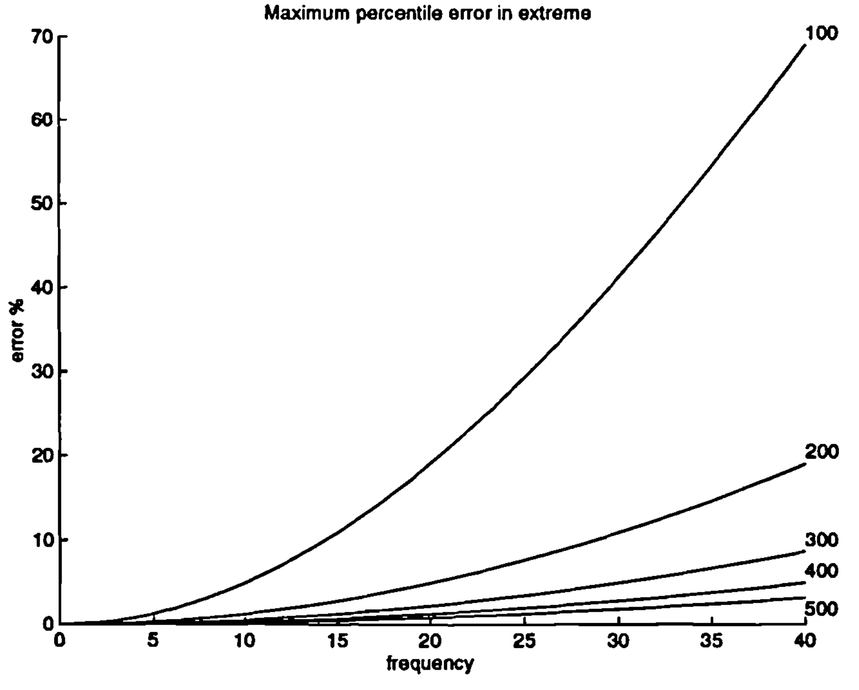


Figure 11 *Percentile error in amplitude for sample frequencies from 100 to 500 Hz.*

If natural frequencies up to 40 Hz are considered, the maximum error in the extrema detection will be limited to 5% if a sample frequency of 400 Hz is used.

Besides the extrema detection also differentiation is needed for the linear and quadratic approach. Two samples in each period are not enough to determine the derivative! Differentiation filters have odd filter coefficients. The differentiation process requires that there are at least as many samples in each period as there are filter coefficients.

The effects of a badly chosen differentiation filter and sample frequency can lead to very strange effects. Consider a filter with coefficients [-a, 0, 0, 0, a]. If we sample a 50 Hz sinusoidal signal with a sample frequency of 200 Hz, the result would be all zero!

All digital filter designs depend on the sample frequency. Filters designed for a certain sample frequency will behave differently (not according to the design specifications) when used at another sample frequency. Differentiation filters will generally not work correctly for frequencies up to half the sample frequency. Make sure to remember these things when using filter designs from books or other people or when adjusting things in programs in which digital filters are used.

3.5 Parameter estimation methods

3.5.1 Logarithmic decrement method

This method was first used by Gardner [Gardner, 1981] and later also by many others. The damping is calculated from the logarithmic decrement of the oscillating signal. The ratio of the amplitudes of adjacent resonance peaks is used to calculate the decrement (see Figure 12). The damped natural frequency f_d can be measured directly as the reciprocal of the oscillating period T_d.

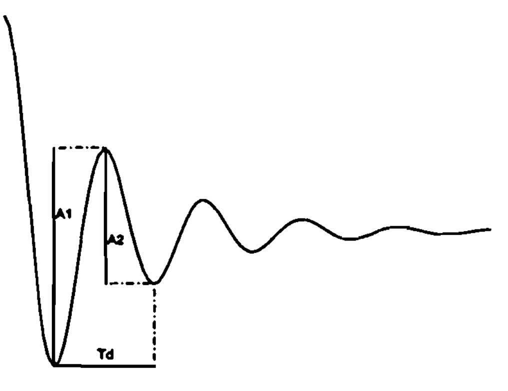


Figure 12 Parameters used in the logarithmic decrement method.

The undamped natural frequency f_n for a freely vibrating system is calculated as:

$$f_n = \frac{f_d}{\sqrt{1 - \beta^2}} \qquad for\ \beta < 1.0 \tag{9}$$

The damping coefficient and the natural frequency can then be calculated as:

$$\beta = \frac{-\ln\left(\frac{A2}{A1}\right)}{\sqrt{\pi^2 + \left[\ln\left(\frac{A2}{A1}\right)\right]^2}} \tag{10}$$

$$f_n = \frac{1}{T_d \sqrt{1 - \beta^2}} \tag{11}$$

Remarks

- This method can only be used for underdamped systems.
- At least two relative amplitudes are needed to estimate β and f_n. Because of this, at least 1.5 oscillation periods of the flush response are needed. If more periods are available, β and f_n can be determined more accurately.
- This method can be used with the placement method as well as the subtraction method. The first (and second) time derivatives of the exponentially decaying oscillation are characterized by the same β and f_n as the original signal. Proof for this can be found in the appendix of van Langen's article [van Langen, 1993].
- The sample frequency must be high enough to analyse the amplitudes of the oscillating signal. To get reliable values (less than 5% error) about 10 samples per period would be preferable (see paragraph 3.4).

3.5.2 Frequency domain method

In this method the exponentially decaying, oscillating signal (impulse response) is transferred to the frequency domain (using an FFT-algorithm). We then obtain the transfer function. If a step is used, the response is first differentiated to provide results equivalent to the impulse response. The peak in the spectrum is caused by the oscillating signal. The natural frequency and the damping are then calculated from the position and amplitude of this peak (see paragraph 3.1.1).

Remarks

- In a practical situation the impulse- or step response will be superimposed on the blood pressure signal. With high heart rates and low natural frequencies, it is possible that the blood pressure spectrum and the spectrum of the oscillating signal overlap. If this situation occurs, it is not correct to determine β and f_n from the peak in the total spectrum.
- Eliminating the blood pressure influence with the placement or the subtraction method will eliminate the problem described above (in case of a stable, periodic signal). The placement and subtraction methods can only be used in the constant approach, because differentiating the signal will increase the peak amplitude (proportional to the signal frequency).
- This method can only be used for underdamped systems.
- This method takes more time than the logarithmic decrement method because of the FFT-algorithm needed.

3.5.3 Dynamic parameter estimation

Micheletti presented a new method of characterizing the time domain response of a second order system to a step function by dynamic parameter estimation [Micheletti, 1990]. The method uses digitized samples of the input signal and does not require any *a priori* information. The parameters β, f_n and A (the step amplitude), are obtained by comparing the input signal to a mathematical model using an iterative optimization procedure. The model parameters are updated each iteration until the input signal adequately matches the model.

The mathematical model for an exponentially decaying oscillation as shown in Figure 7. is:

$$P(t) = \frac{Ae^{-2\pi f_n t}}{\sqrt{1-\beta^2}} \sin\left(2\pi f_n \sqrt{(1-\beta^2)}\, t + \arctan\frac{\sqrt{1-\beta^2}}{\beta}\right) \tag{12}$$

Equation (12) can be expressed as:

$$P(t) = P(f_n, \beta, A, t) \tag{13}$$

Expanding P(t) in a Taylor series in the neighbourhood of given value $\alpha_0 = (f_{n0}, \beta_0, A_0)$ of the parameter set $\alpha = (f_n, \beta, A)$, while ignoring the higher order terms, gives:

$$P(t) = P(t)_{\alpha 0} + \left(\frac{\delta P(t)}{\delta f_n}\right)_{\alpha 0} \Delta f_n + \left(\frac{\delta P(t)}{\delta \beta}\right)_{\alpha 0} \Delta\beta + \left(\frac{\delta P(t)}{\delta A}\right)_{\alpha 0} \Delta A \tag{14}$$

Now the error between the model and the measured signal will be minimized using a minimum square error criterion. The total square error is given by:

$$E = \sum_{i=1}^{N}\left[P_r(t_i) - P(t_i)_{\alpha 0} - \left(\frac{\delta P(t_i)}{\delta f_n}\right)_{\alpha 0} \Delta f_n + \left(\frac{\delta P(t_i)}{\delta \beta}\right)_{\alpha 0} \Delta \beta + \left(\frac{\delta P(t_i)}{\delta A}\right)_{\alpha 0} \Delta A \right]^2 \qquad (15)$$

With:

- $P_r(t_i)$ the real (measured) pressure value
- $P(t_i)_{\alpha 0}$ the pressure value calculated with the model using parameter set α_0
- N the number of samples of the measured signal

The total square error is minimized by solving the partial derivatives of (15) relative to f_n, β, A evaluated at α_0:

$$\frac{\delta E}{\delta f_n} = \frac{\delta E}{\delta \beta} = \frac{\delta E}{\delta A} = 0 \qquad (16)$$

The solution of (16) gives the corrections Δf_n, $\Delta\beta$, ΔA necessary for updating f_n, β, A for each iteration step. The initial parameters α_0 can be determined with the logarithmic decrement method as described in the previous paragraph. The initial value of the step function amplitude A can be calculated as (see Figure 13):

$$A = \frac{A_1 + A_2 + 2A_3}{4} \qquad (17)$$

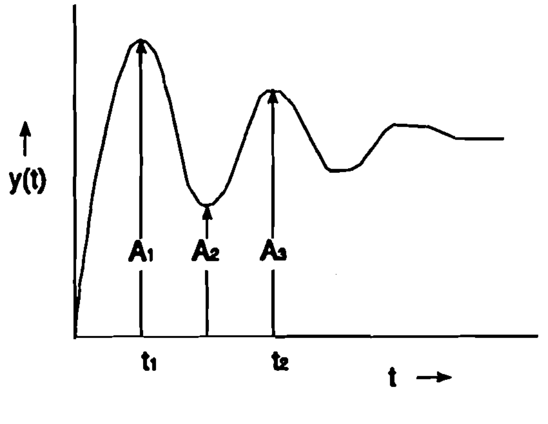


Figure 13 *Parameters used in the estimation of A.*

Remarks

- This method is intended to be used with underdamped systems. The initial set of parameters is determined with the logarithmic decrement method. The method can also be used with critically and overdamped systems (with $\beta<1$), but then a different method must be used for the initial parameters (see next paragraphs). For overdamped systems with $\beta\geq1$ another model must be used.
- The minimum square error criterion in combination with the placement method constant approach will not work because the blood pressure influence is not incorporated in the model. The subtraction method (constant approach) will produce accurate parameters. To use the dynamic parameter estimation method with the linear or quadratic approach (with or without subtraction method), the reference model must be differentiated as well (once respectively twice).

  As discussed with the logarithmic decrement method, the first (and second) time derivatives of the exponentially decaying oscillation are characterized by the same β and f_n as the original signal [van Langen, 1993]. The differentiation process introduces an extra phase shift and an additional constant. The additional constant can be incorporated in the initial guess for the amplitude. The extra phase shift can be eliminated in the matching process that fits the measured (sampled) data to the model. The dynamic parameter estimation method can thus be used for the linear and quadratic approaches (with or without the use of the subtraction method) with the same model as described in (12).
- The method takes more time than the other methods due to the iteration process and the minimum square error criterion. On the other hand the parameters can be determined more accurately (also in the presence of noise). The necessary accuracy for the parameters is discussed in paragraph 3.6.1.
- Micheletti uses a sample frequency of about 47 Hz to analyse a signal with an oscillation period of 0.85 seconds (1.18 Hz). The demands concerning the sample frequency are not as strict as with the logarithmic decrement method because no specific points of the oscillation need to be determined.

3.5.4 Karnopp's methods

A CMS cannot always be modelled by an underdamped second order system. Also critically damped and overdamped CMS can occur. The logarithmic decrement method cannot be used then, because no oscillations are present in the step response. The step response of critically damped and overdamped systems show an exponential decay. Warburg developed a curve-fitting method, in which β and f_n are determined by fitting the step response to an exponentially decaying mathematical model [Warburg, 1949].

Recently van Langen showed in his *in situ* measurements that in the clinical setting critically damped and overdamped systems do exist [van Langen, 1993]. Karnopp suggested dividing systems into the following ranges [Karnopp & Fisher, 1990a]:

$0.0 < \beta < 0.2$	light damping
$0.2 \leq \beta \leq 1.2$	moderate damping
$1.2 < \beta$	heavy damping

For systems with light damping Karnopp suggests the logarithmic decrement method. For the other two situations he developed the methods described in the next two paragraphs.

3.5.4.1 Moderately damped systems

The parameters for moderately damped systems can be determined from the impulse response of the system. The differential equation for the free motion system is:

$$\ddot{x} + 2\beta\omega_n\dot{x} + \omega_n^2 x = 0 \tag{18}$$

If we take the initial condition x(0)=0 the solutions of (18) are:

$$\begin{aligned} x &= \left(\frac{v_0}{\omega_d}\right)e^{-\omega_n\beta t}\sin(\omega_d t) \qquad & 0<\beta<1 \\ x &= v_0 t e^{-\omega_n t} \qquad & \beta=1 \\ x &= \left(\frac{v_0}{\omega_d}\right)e^{-\omega_n\beta t}\sinh(\omega_d t) \qquad & 1<\beta \end{aligned} \tag{19}$$

where:

$$\begin{aligned} \omega_d &= \omega_n\sqrt{|1 - \beta^2|} \\ v_0 &= \dot{x}(0) \end{aligned} \tag{20}$$

If a pressure impulse is applied to the CMS the response will look like Figure 14 (plot of eq. 18). The parameters used for the computation of β and f_n can also be found in Figure 14.

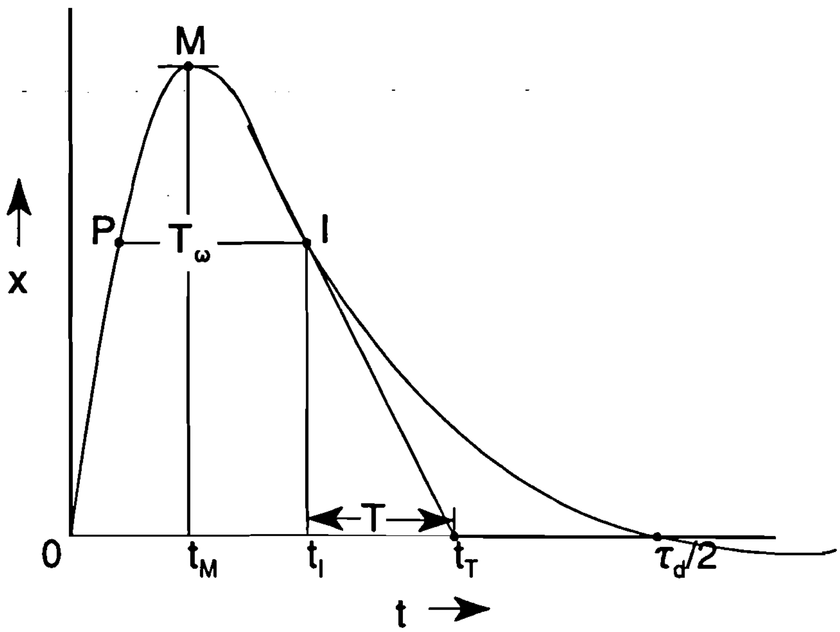


Figure 14 *Parameters used in Karnopp's method for moderate damped systems.*

Parameters used in the method:

M	a maximum point of the curve ($dx/dt = 0$)
x_M	the displacement at the point M
t_M	the time of the point M
I	the inflexion point after M ($d^2x/dt^2 = 0$)
x_I	the displacement at the point I
t_I	the time of the point I
t_T	the time at which a tangent line through I crosses the time axis
T	the time from t_I until t_T
P	a point preceding I at the same displacement as I
T_ω	the time from P to I
$\tau_d/2$	the half period (for underdamped systems only)

The method is based upon measurements of x_I and x_M and the time T. Let:

$$R = \frac{x_I}{x_M} \tag{21}$$

then:

$$\beta = \frac{0.25}{1 - R} + 0.50R - 0.32 \tag{22}$$

Once β is determined, f_n can be obtained in one of two ways (three ways if the system is underdamped).

First:

$$f_n = \frac{\beta}{\pi T} \tag{23}$$

A second method results from the measurement of T_ω:

$$f_n = \frac{\frac{3.0}{T_\omega}}{2\pi(1 + 0.9\beta)} \tag{24}$$

Finally, if the system is underdamped and oscillations are visible:

$$f_n = \frac{1}{\tau_d\sqrt{1 - \beta^2}} \tag{25}$$

The proof of this method can be found in [Karnopp, 1990a].

Remarks

- This method can be used for a large range of damping ($0.2 \leq \beta \leq 1.2$). The third variant of this method is equivalent to the decrement method.
- The method is based upon an impulse response test.
- The method does not deal with the influence of the blood pressure signal, so the placement or the subtraction method must be used. The mathematical model used in this method only holds for the constant approach. For the linear and quadratic approaches the model does not hold.

3.5.4.2 Heavily damped systems

For a heavily damped system the impulse response will look like Figure 15. The parameters used for the computation of β and f_n can also be found in Figure 15. The method is based upon measurements of the points M, x_A and x_B as well as the times Δt and T. It can be shown that $t_I = 2t_M$. The points A and B are selected as far as to the right of t_I as the data are meaningful. Then Δt is calculated. Let:

$$\omega_1 = \frac{1}{\Delta t} \ln \left(\frac{x_A}{x_B} \right) \tag{26}$$

then:

$$\beta = \frac{T\omega_1}{2\sqrt{T\omega_1 - 1}} \tag{27}$$

and:

$$f_n = \frac{\beta}{\pi T} \tag{28}$$

The proof of this method can be found in [Karnopp & Fisher, 1990b].

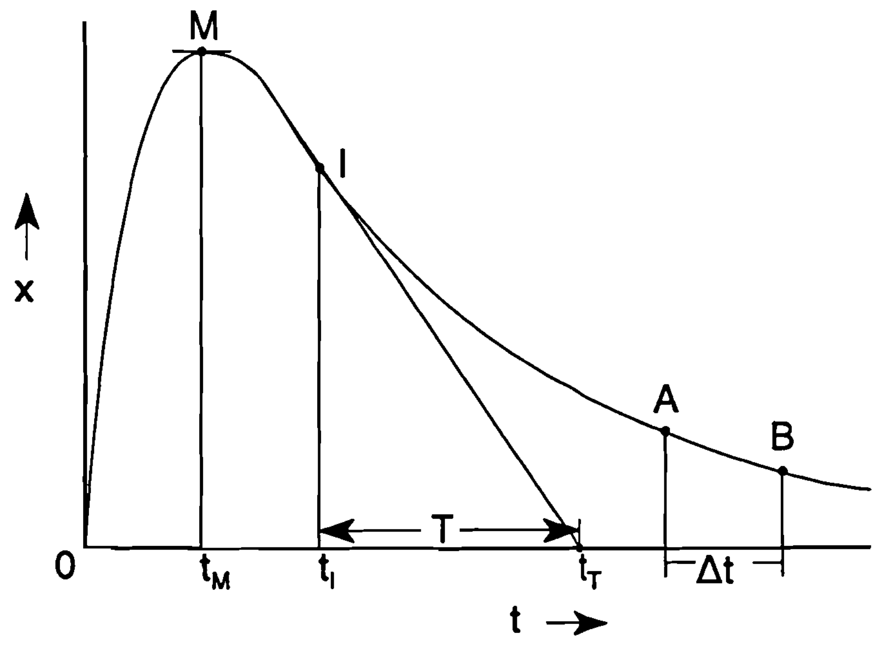


Figure 15 Parameters used in Karnopp's method for heavily damped systems.

Parameters used in the method:

A, B	two points at large values of t
x_A	the displacement at the point A
t_A	the time of the point A
x_B	the displacement at the point B
t_B	the time of the point B
Δt	the time from t_A until t_B

The other parameters are the same as in Figure 14.

Remarks

- This method can be used for very heavy damping ($\beta \geq 1.2$).
- The method is based upon an impulse response test.
- The method does not deal with the influence of the blood pressure signal, so the placement or the subtraction method must be used. The mathematical model used in this method only holds for the constant approach. For the linear and quadratic approaches the model does not hold.

3.5.5 Neural networks

Neural networks can also be used to determine the damping and the natural frequency of the catheter-manometer system. Johnson described a neural network for this purpose [Johnson et al., 1991]. From the blood pressure eleven parameters are extracted (systolic pressure, diastolic pressure, mean pressure, heart rate etc.). These parameters are used as input for the neural network. The output of the neural network will be estimations for β and f_n.

The network is trained with distorted blood pressures, of which the CMS parameters are known. After the network is trained, blood pressure signals (parameters) can be presented to the network. The network will then produce β and f_n. To reduce the complexity of the network, the number of inputs must be reduced. The number of inputs can be reduced by omitting those inputs that have little effect on the outputs.

The prediction error of this neural network for f_n was -0.56 Hz +/- 4.66 Hz over a range of 1 to 30 Hz. The prediction error for β was -0.03 +/- 0.19 over a range of 0.05 to 0.60. The sensitivity and specificity for detection of damped waveforms were 0.67 respectively 0.92. The sensitivity and specificity for detection of resonant waveforms were 0.74 respectively 0.86.

A similar application of neural networks is described by A. Prentza [Prentza & Wesseling, 1994]. As inputs for the network, the same blood pressure parameters are used that were described by Johnson. The output however in this case will be a measure whether the presented blood pressure signal is distorted or not. In this case the number of inputs can also be reduced. Two methods for the reduction of the number of inputs are described in [Prentza, 1994].

Remarks

- The neural network's performance is very poor when compared to the other methods.
- A big advantage of this method is that no step or impulse test is needed to determine the parameters of the CMS.
- A disadvantage is that the neural network must be trained. Training the network will require many different blood pressure signals with the corresponding CMS parameters. Training the network will be done in laboratory settings (because there is no time for this when using the device in the clinical setting).
- The blood pressure parameters are extracted (real time) with the blood pressure analysis algorithm. The neural network can determine the parameters at any time because no flush is needed.

3.5.6 Natural Observation System (NOS)

The natural observation method is a combined parameter extraction and correction method. The parameter estimation part uses a minimum square error technique to determine the NOS parameters. These NOS parameters can be directly related to the damping and the natural frequency. The NOS will be discussed in the next chapter.

3.6 Parameter estimation strategy

Before the strategy of the parameter estimation process is discussed, it must be known how accurately the parameters must be determined. It would be unnecessary to choose a very accurate method for the parameter estimation if the effect of this accuracy would be negligible in the correction process.

3.6.1 Parameter accuracy

Research of the accuracy needed in the determination of the parameters is presented in the report by Alofs in chapter 3 [Alofs, 1993]. The effect of errors from +50% to -50% in β and f_n on systolic, diastolic and mean pressure were investigated. The effects of errors in the parameter estimation (f_n and β) can be explained by looking at the effects on the amplitude of the transfer function.

In Figure 16(b) the optimal case is shown (β and f_n correct). The total transfer function is 'flat'. In a practical situation the inverse filter will be cascaded with a low pass filter that passes all meaningful information (blood pressure signal) and removes the noise. In Figure 16(a) the natural frequency is estimated correctly, but the damping is underestimated (25%). The effect of underestimating β is a transfer function that looks like the transfer function of a band-stop filter at a centre-frequency f_n. If the selected cascade low-pass filter has its cutoff frequency around f_n the effect of underestimating β will be negligible. In Figure 16(c) the natural frequency is estimated correctly, but the damping is overestimated (25%). In this case the total transfer function still shows some amplification for frequencies around f_n. Together with the cascade low-pass filter the overall effect will still be an improvement in the blood pressure frequency range.

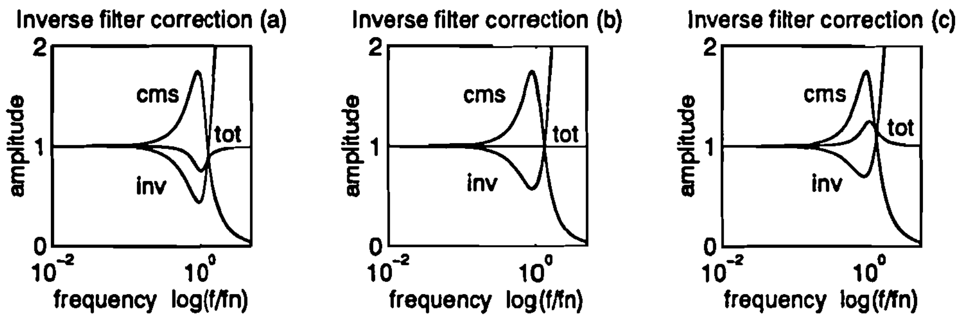


Figure 16 *Plot of the transfer function with f_n determined correctly. With: (a) β too small; (b) β correct; (c) β too large.*

In Figure 17(a) the natural frequency is overestimated (25%) and the damping is underestimated (25%). The resulting transfer function still shows some amplification around f_n. For frequencies higher than f_n the resulting transfer function behaves like a low-pass filter at cutoff frequency f_n. If the low-pass filter has its cutoff frequency around f_n this effect can even improve the low-pass filtering. In Figure 17(b) the natural frequency is overestimated (25%) and the damping is estimated correctly. The resulting transfer function looks like Figure 17(a) but there is more amplification around f_n.

In Figure 17(c) the natural frequency is overestimated (25%) and the damping is overestimated (25%). The resulting transfer function looks like Figure 17(b) but there is more amplification around f_n.

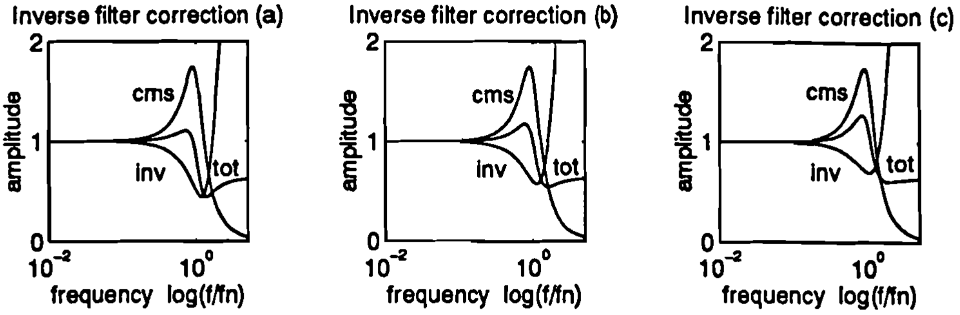


Figure 17 *Plot of the transfer function with f_n determined too large.*
With: (a) β too small; (b) β correct; (c) β too large.

In Figure 18(a) the natural frequency is underestimated (25%) and the damping is underestimated (25%). The resulting transfer function behaves as a low-pass filter up to the falsely estimated f_n. For frequencies above this f_n the amplification increases rapidly until it stabilizes at a high value. The cascade low-pass filter must be sharp and it must have a cutoff frequency equal to or below this f_n to prevent a 'noise explosion' in the higher frequencies. In Figure 18(b) the natural frequency is underestimated (25%) and the damping is estimated correctly. The effect is generally the same as in Figure 18(a), but the low pass behaviour up to the falsely estimated f_n is weaker. In Figure 18(c) the natural frequency is underestimated (25%) and the damping is overestimated (25%). The effect is generally the same as in Figure 18(b), but the low pass behaviour up to the falsely estimated f_n is weaker.

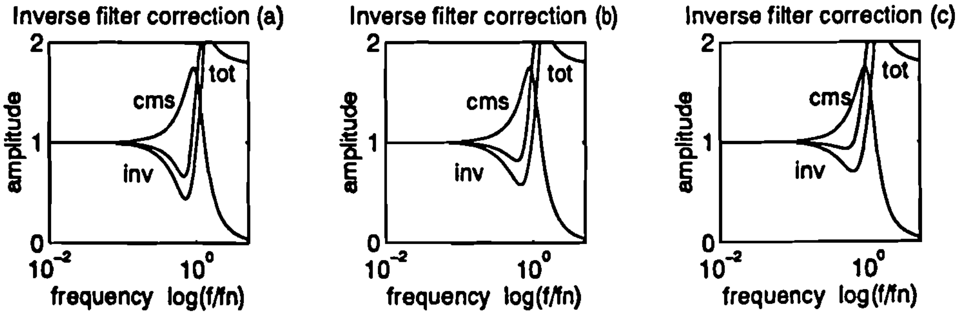


Figure 18 *Plot of the transfer function with f_n determined too small.*
With: (a) β too small; (b) β correct; (c) β too large.

The most important conclusion is that the effect of overestimating the parameters is not as dangerous as underestimating the parameters. When the natural frequency is overestimated, the overall transfer function will look like the transfer function of the CMS itself. When the natural frequency is underestimated, the overall transfer function will look like the inverse transfer function. This will deteriorate the blood pressure signal even more.

To get an impression of the actual errors made in the blood pressure parameters (systolic and diastolic pressure) these errors must be related to the errors made in the parameter estimation process. This relation however depends not only on the percentile error made in the estimation process, but also on the shape of the blood pressure signal, the 'real' damping and natural frequency and the cascade low-pass filter used.

Define the following errors:

Error in systolic blood pressure:	$SBP_a - SBP_c$
Error in diastolic pressure:	$DBP_a - DBP_c$
Mean error per period:	$\frac{1}{N}\sum_1^N \lvert p_a - p_c \rvert$
RMS error per period:	$\sqrt{\frac{1}{N}\sum_1^N (p_a - p_c)^2}$

with:

SBP	systolic blood pressure
DBP	diastolic blood pressure
p_a	actual pressure
p_c	corrected pressure
N	number of samples in one blood pressure period

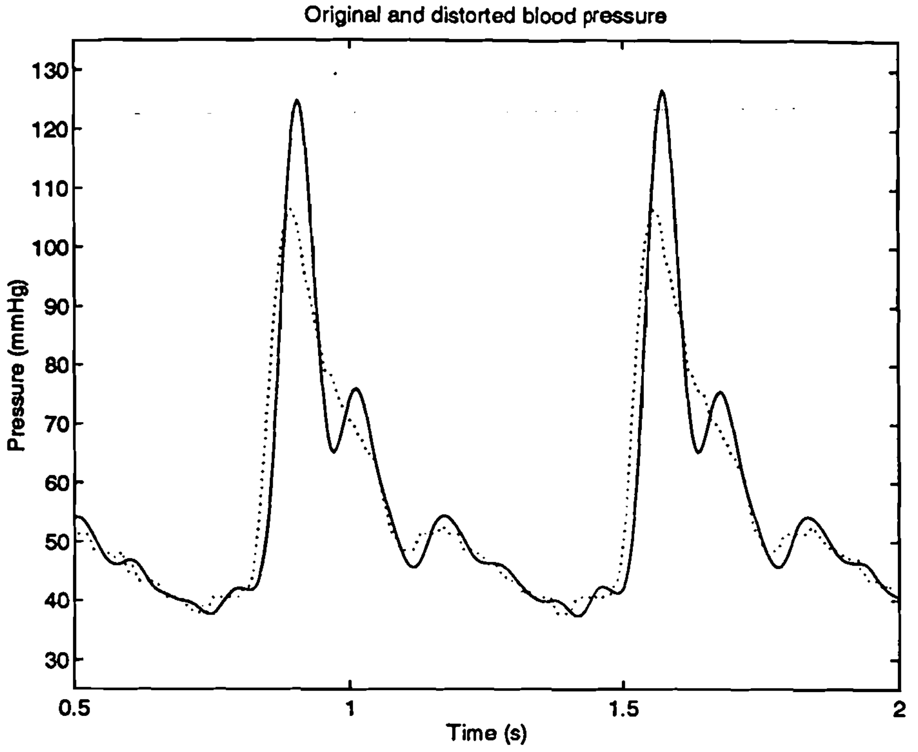


Figure 19 *Original (dotted line) and distorted (solid line) blood pressure signal. CMS parameters: β = 0.2 ; f_n = 10 Hz.*

The prerecorded blood pressure signal (of a blood pressure simulator) in Figure 19 is distorted with a CMS that has a natural frequency f_{na} of 10 Hz and a damping coefficient β_a of 0.2.

Original blood pressure signal:	SBP_a = 106 mmHg and DBP_a = 38 mmHg.
Distorted blood pressure signal:	SBP_d = 127 mmHg and DBP_d = 37 mmHg.

The errors caused by the CMS distortion:

SBP = +21 mmHg (20%)	MEAN = 4 mmHg
DBP = -1 mmHg (3%)	RMS = 7 mmHg.

The distorted signal is corrected with a correction filter with parameters f_{nc} and β_c. This will be done for $f_{nc} = 0.5 \cdot f_{na}$ up to $f_{nc} = 1.5 \cdot f_{na}$ (errors in f_n from -50% to +50%) and for $\beta_c = 0.5 \cdot \beta_a$ up to $\beta_c = 1.5 \cdot \beta_a$ (errors in β from -50% to +50%) in steps of 5%. For each combination f_{nc} and β_c the four errors will be calculated. The correction filter consists of the inverse-CMS filter cascaded with a second order low-pass filter with cutoff frequency f_c = 30 Hz (see paragraph 4.4.3). The resulting error curves can be found in Figure 20.

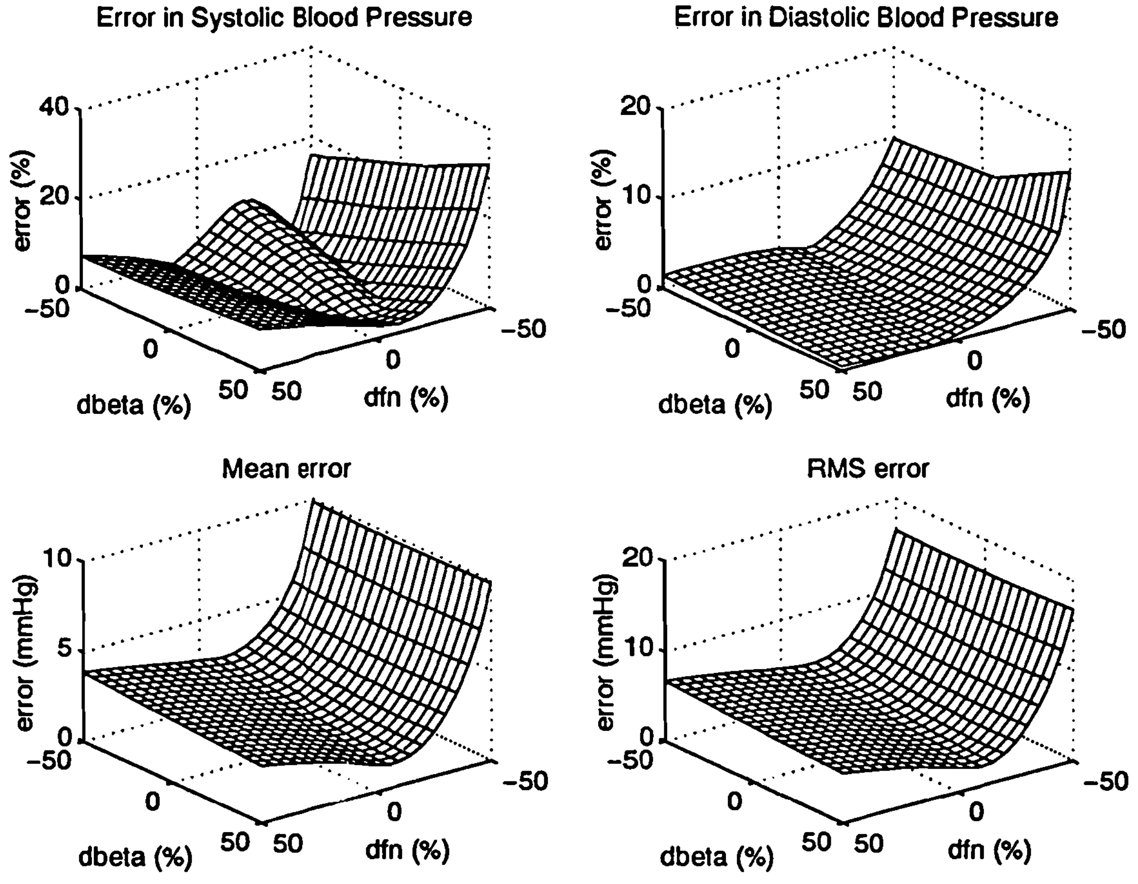


Figure 20 Error curves (correction parameters f_n and β range from -50% to +50%).

In this paragraph the relation between errors in the estimated parameters β_c and f_{nc} (for the correction filter) and the blood pressure errors were discussed, for a certain blood pressure signal and CMS (with parameters β_a and f_{na}). If the parameters β_c and f_{nc} are determined with the logarithmic decrement method, we need to know what causes the errors in the estimation. The parameters were calculated with formulas (10) and (11). Errors can occur if the relative amplitudes (A_1 and A_2) or the oscillation period (T_d) are estimated incorrectly. As discussed in this paragraph underestimating the natural frequency causes the largest errors. The natural frequency is estimated too small if T_d is estimated too large and if β is estimated too small (ratio A_2/A_1 is near 1).

Errors in A_1, A_2 and T_d occur if the sample frequency is chosen too low. In paragraph 3.4 the relation between the maximum error in the determination of an extreme (e_{max}) of an oscillating signal and the sample frequency (f_s) was discussed. The lowest estimate for β occurs when A_1 is estimated too low ($A_{1\_ESTIMATED} = A_1 - e_{max}$) and A_2 is estimated correctly. The lowest estimate for f_n then occurs when T_d is overestimated (maximal one sample period). The largest estimate for T_d thus is $T_{d\_ESTIMATED} = T_d + 1/f_s$.

The errors in the estimation of A_1, A_2 and T_d are thus related to f_s. If the real values for A_1, A_2 and T_d (and thus the real β and f_n) are known, the relation between the sample frequency and the maximum errors in the estimation of β and f_n can be found. With this relation and the corresponding error curves, the minimal sample frequency can be determined from the maximal tolerated error in the blood pressure parameters (for a certain blood pressure signal and CMS).

3.6.2 Patient safety

Another problem with the parameter estimation is the reliability of the determined parameters. The measurement of the system parameters itself can influence the system characteristics. During the flush test air bubbles can be moved (if there are any) or flushed out. This can cause the transfer function to change. The parameters extracted are then no longer representative for the system and the corrected blood pressure would be invalid.

To overcome this problem, and make to sure that the extracted parameters are representative for the system, the flush test can be done twice if there are indications that air bubbles or blood clots are present in the system. If the values extracted in the second test are close to the results of the first test, the system is stable and the values can be used for correction. In case the results do not match, a third flush test may be necessary. If the results of the second and the third test match, the parameters can be used. If not there are many air bubbles (or large ones) in the system and flushing is necessary.

As stated, this extra flushing is only necessary if there are indications that air bubbles or blood clots are present in the system. An indication for air bubbles or blood clots is a lower natural frequency and a higher value for the damping than usual. But what are the usual values for a CMS? The simplest way to determine the 'usual' parameters would be to use the parameter history (if this is stable). A problem with this approach is that the old parameters can be stable and wrong at the same time. If a large air bubble is stuck in the CMS the parameters will be (fairly) constant, but it would be a mistake to use these parameters as 'usual' values.

A possibility to determine the 'usual' parameter values, is to do a reference flush test after the CMS has been installed and before it is connected to the patient. A disadvantage of this method would be that with every change to the system the patient must be disconnected, to do a new

reference measurement. In the clinical setting this will take too much time.
The use of *a priori* knowledge about the selected CMS would be a more elegant solution. In this method the user must enter the components used to construct the CMS. With a mathematical model or a component library the 'usual' parameter values are determined. Every change to the CMS must be reported to the correction system to determine the new 'usual' parameters. An advantage would be that the possible margins for the CMS parameters (and the transducer parameters) could also be included in this library as well.

3.6.3 Advised strategy

Besides the problems of accuracy and reliability the parameter estimation technique itself must be considered. In the previous paragraphs we introduced several methods of parameter extraction. To make sure that the parameters extracted from a flush response are reliable (are extracted correctly) it is recommendable to use more than one method to estimate the parameters. This will work even better if the methods are based on different techniques.

To eliminate the effect of the blood pressure the response must always be placed in the diastolic downslope of the blood pressure signal. From the blood pressure analysis program the heart rate is used to calculate whether the diastolic downslope is long enough to hold the response. If so, the placement method can be used, if not the subtraction method must be used. In case of the subtraction method placement of the response will help determine the size of the signal part that will be used.

The simplest and most reliable method is the logarithmic decrement method, provided the sample frequency is high enough. It can be used in underdamped cases and with all three approaches of the placement method. This test is chosen to be the standard test that will always be done. Micheletti's test (dynamic parameter estimation) is very accurate, but this is not the reason to use it. Two advantages of this method are that it is more stable in case of noise and that it does not need a high sample frequency. The dynamic parameter estimation method has these advantages because it uses a minimum square error estimation. Because the initial parameters are available (the logarithmic decrement test is done always) Micheletti's test is a good verification.

Another test that can be helpful in the verification is the Natural Observation System (NOS). The NOS uses a completely different strategy from the other two. The CMS parameters can be calculated from the NOS parameters (if the CMS behaves exactly like a noise free second order system). These three tests provide all that is needed to determine the parameters.

parameter estimation:

- determine placement or subtraction method
- start flush test
- record flush response
- start logarithmic decrement method
- in case of oscillations:
    - verify using Micheletti's method
    - (verify using NOS)
    - if parameters are not representative:
        - repeat parameter estimation
        - if representative:
            - update parameters
        - else:
            - repeat parameter estimation
        - if representative:
            - update parameters
        - else:
            - WARNING FLUSHING NEEDED
    - else:
        - update parameters
- in case no oscillations:
    - use Karnopp's method
    - verify using dynamic parameter estimation
    - (verify using NOS)
    - if parameters are not representative:
        - repeat parameter estimation
        - if representative:
            - update parameters
        - else:
            - repeat parameter estimation
        - if representative:
            - update parameters
        - else:
            - WARNING FLUSHING NEEDED
    - else:
        - update parameters

4 Improving the Response of CMSes

4.1 Increasing the damping

The response of a CMS can be improved by expanding the frequency range with `flat' response. This can be done by increasing the damping coefficient to near the optimum value of 0.59. The CMS will then have a `flat' (within 5%) response up to 87% of the natural frequency. How can damping be increased? If the CMS is considered as a transmission line, the transmission will be optimal if the line is terminated at either end or both by its characteristic impedance. At the input end of the catheter a matching impedance can be arranged quite easily, using a capillary of the correct dimensions [Latimer, 1968]. These correct dimensions depend on the β and the f_n of the CMS that is used. With the CMS (second order resonant system) damping can also be included at the manometer end. Three types of damping at the manometer end will be discussed below.

4.1.1 Series damping

In the case of series damping a capillary with resistance R_s is series-connected between the catheter and the manometer (see Figure 21). The value for R_s can be approximately calculated, but depends on the catheter length. The series damping, however, introduces an extra low pass filter into the system that limits the frequency response even further. For practical use series damping is not suited.

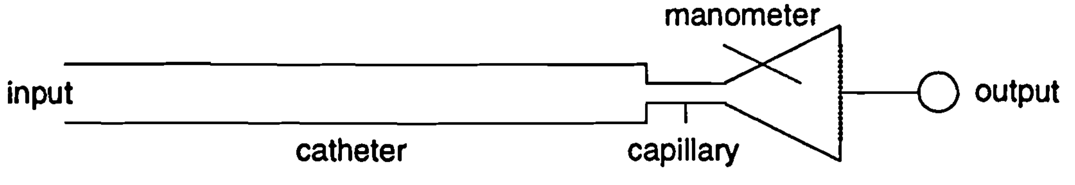


Figure 21 Series damping: capillary of correct dimensions in series with catheter.

4.1.2 Parallel damping

In the case of parallel damping a parallel resistance R_p is introduced between the catheter and the manometer (see Figure 22). Parallel damping does not limit the linear frequency response as series damping does. If the required parallel resistance becomes comparable to the catheter resistance, an error in the DC pressure indication occurs. The manometer then indicates a too low value and thus re-calibration or some other compensation scheme is necessary. Because of the re-calibration necessary, parallel damping is not suited for practical use.

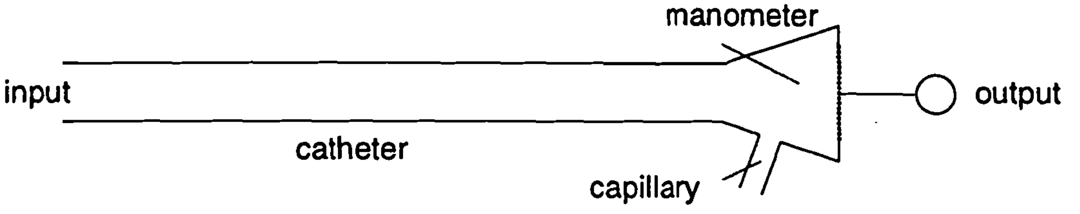


Figure 22 Parallel damping: capillary of correct dimensions parallel with catheter.

4.1.3 Manometer damping

The damping now is incorporated in a viscous manometer membrane (see Figure 23). It would have the advantages of parallel damping, without the disadvantages of re-calibration. The damping should however be adjusted to the type of catheter being used. Manometer damping has never been practically used.

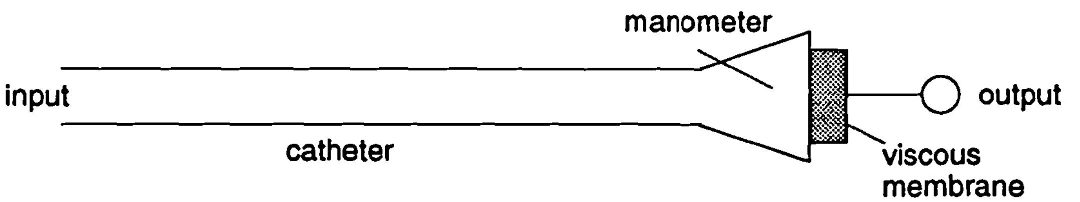


Figure 23 Manometer damping: Viscous membrane provides correct damping.

All three mechanical damping schemes require some sort of calibration or adjustment before they can be used. In the clinical setting the catheter properties can change because of blood clots and air bubbles. With every change of the system, the damping schemes require adjustment. Because mechanical damping requires constant attention from OR or ICU personnel (who are busy enough as it is) these basic techniques have never been used in practice.

4.2 Mechanical damping devices

Two practical mechanical damping devices (resonance elimination devices) that were developed are:

- The Rose® (Resonance OverShoot Eliminator, Gould Instrument Inc., Cleveland, OH). This device contains a small air bubble isolated from the pressure system by a flexible diaphragm. The device is not adjustable.

- The Accudynamic® (Sorensen, ADN-02). This device has a 0.1 ml captive air bubble isolated from the pressure system via an adjustable needle valve. By turning this valve the mechanical damping can be adjusted. A similar adjustable needle valve was used in [Lapointe & Roberge, 1974].

These devices are not frequently used in the clinical setting because they need extra attention from the clinical personnel (install them and optionally calibrate or adjust the device). Tests of the latter two devices were presented in articles by Hipkins [Hipkins, 1989] and Kleinman [Kleinman, 1992].

4.3 Electrical compensation in hardware

In the last three decades several researchers have investigated the possibility of electrical compensation of the frequency response of a CMS. With these techniques the resonant character of the CMS is accepted and attempts are made to compensate the output of the manometer with some kind of electrical system.

Electrical compensation tries to expand the range of 'flat' frequency response. The first way to do this is to use a filter that flattens the hump in the transfer function of the CMS around the natural frequency (overshoot). The second, and more elegant way, is to implement an inverse filter, which gives a completely 'flat' transfer function for the total system. Just like the parameter estimation process, the correction process must be fully automatic. Several correction methods implemented in hardware will be discussed in this paragraph.

4.3.1 Compensating amplifiers

Noble developed an amplifier that had a built-in correction circuit [Noble & Barnett, 1963]. The circuit used in his amplifier was an analog low pass filter that could be tuned with potentiometers. The amplifier provided a uniform frequency response and a linear phase response.

Remarks

- This amplifier cannot tune the filter automatically and therefore it is not suited for our purpose.
- Special hardware is necessary in this case that replaces the standard amplifiers used in the clinical monitor.

4.3.2 Automatic notch filter

A fully automatic device for compensating for artifacts in CMS pressure recordings was developed by Brower [Brower et al., 1975]. The heart of this device is a parameter detector. Brower assumed that the natural frequency was in the range from 15 to 88 Hz. The preconditioner of the parameter detector consists of a band-pass filter (15-100 Hz) and a differentiator. The preconditioned signal is then transferred to the frequency domain and analyzed (see paragraph 3.5.2). The differentiator is used to produce a more uniform spectrum (differentiating causes an amplification of the weaker, high frequencies).

The extracted β and f_0 are then used to tune an equalisation filter that consists of an inverse filter and a compensation filter (notch filter). The whole device was implemented in hardware, which makes it rather expensive. Nowadays Brower's device could be implemented on a computer.

Remarks

- The assumption that the natural frequency is above 15 Hz is not correct. The spectra of the blood pressure signal and the resonance effects generally overlap (see first remark in paragraph 3.5.2).
- The differentiator used to produce a more uniform spectrum will also amplify the high frequency noise.
- Compensation techniques that use special hardware are not suited for our purposes.

4.4 Electrical compensation in software

As discussed in paragraph 4.3 the correction process must be fully automatic. The hardware solutions found are not flexible enough to satisfy this demand. In this paragraph several correction methods implemented in software will be discussed. The methods discussed in paragraph 4.4.1 and 4.4.2 use a filter that flattens the hump in the transfer function of the CMS around the natural frequency (overshoot). The methods in paragraphs 4.4.3 to 4.4.4 implement an inverse filter, that gives a completely 'flat' transfer function for the total system.

4.4.1 Automatic low-pass or Butterworth filter

Plasman described the use of first order low-pass and two-pole Butterworth low-pass filters to improve the frequency characteristics of a CMS [Plasman & Timmers, 1981]. These filters were implemented as digital adaptive filters. The system parameters were estimated with the logarithmic decrement method, in case of an underdamped system, and with Warburg's method, in case of an overdamped system.

Plasman's calculations show that for β between 0.05 and 0.23 the Butterworth filters are best and that for β between 0.24 and 0.50 the low-pass filter is best. The filters improve the bandwidth over which the transfer function will be tolerably flat (6%) and they reduce the overshoot to acceptable limits ($< 50\%$ for $0.1 \leq \beta < 0.26$, negligible for $\beta \geq 0.26$).

Remark

- The performance could be improved by using these filters in combination with an inverse filter like all recent methods do.

4.4.2 Adaptive spline filter

In this method a spline filter is used to estimate the resonant frequency [Takeuchi et al., 1988]. The design specifications for the spline filter are:

(a) The filter must remove any oscillations that appear directly after the rising edge and the falling edge of the waveforms.

(b) The filter must reproduce the original waveform in the segment where the noise does not appear.

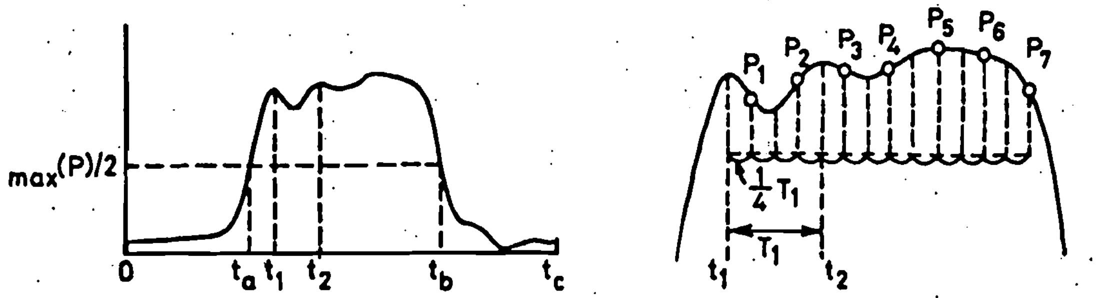


Figure 24 *Selected samples in spline filter method.*

The spline filter consists of a noise detector and a spline function generator. The noise detector divides a blood pressure period into three parts. The first part $(0, t_a)$ is from the start of the period to the 50% maximum pressure level on the rising edge, the second part (t_a, t_b) between the 50% levels of the rising edge and the falling edge, and the third part (t_b, t_c) between the 50% level of the falling edge and the end of the blood pressure period (see Figure 24 on the left).

The noise detector analyzes the blood pressure signal directly after the rising and the falling edge. The two periods of the oscillations following an edge are extracted. The samples in the first part will be evenly spaced (fixed sample rate). In the second part the first sample is chosen to be between the first local maximum and first local minimum after the rising edge. The other samples taken from the second part will be located at $n \cdot$ (half an oscillation period) from the first sample (see Figure 24 on the right). According to the same principle samples will be selected in the third part. The sample rate in these last two parts is thus determined adaptively according to the frequency of the oscillations present in the signal.

After all samples are collected the output signal is constructed by fitting a spline through all the samples. The effect of this would be the same as applying a digital filter that changes its characteristics for each part of the blood pressure period.

Remarks

- This method is not suited for real-time signal processing.
- No separate parameter estimation algorithm is needed with this method.
- In case of high values for β (for example due to air bubbles) no oscillations will be visible. Using a spline curve in this case (if the noise detector works in this situation) will `flatten' the signal even more. The method will not work correctly in case no oscillations are visible.

4.4.3 Digital adaptive inverse filters

As discussed in chapter 2, the CMS can be modelled as a second order system. The general transfer function for the CMS, H(jω), was stated in formula (3). The transfer function of the inverse filter will be:

$$H_{inv}(s) = \frac{1}{H_{cms}(s)} = \frac{s^2 + 2\beta\omega_n s + \omega_n^2}{\omega_n^2} \qquad (33)$$

The numerator of this transfer function has a higher order than the denominator. A filter with such a transfer function is not physically realisable because it has an infinite amplification for infinitely high frequencies. The transfer functions of the CMS and the inverse filter for this theoretical case can be seen in Figure 25.

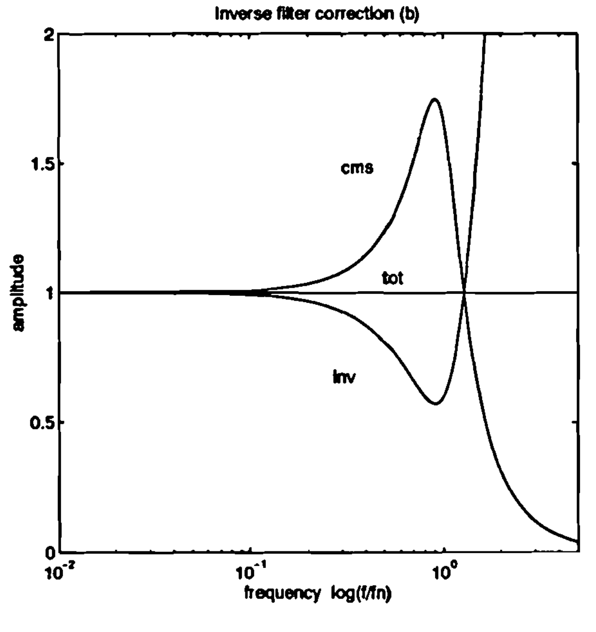


Figure 25 Correction of the CMS transfer function with the inverse filter.

A 'flat' transfer function is only needed for the part of the frequency spectrum that holds the blood pressure signal. For higher frequencies (noise) the optimal system behaviour is that of a sharp low-pass filter. To realize this we add a low-pass filter to the inverse filter. The combination of the inverse filter and the low-pass filter will be called correction filter from now on. The cutoff frequency of the low-pass filter will depend on the bandwidth of the blood pressure signal. If the order of this low-pass filter is two or higher, the correction filter will be physically realisable. The transfer function of the most elemental second order low-pass filter is:

$$H_{lpf}(s) = \left(\frac{\omega_c}{s + \omega_c}\right)^2 \tag{34}$$

with ω_c the cutoff frequency of the filter.

The total transfer function of CMS plus correction now is:

$$H_{tot} = H_{cms} \cdot H_{inv} \cdot H_{lpf} = \left(\frac{\omega_c}{s + \omega_c}\right)^2 \tag{35}$$

The correction process with the correction filter is visualized in Figure 26.

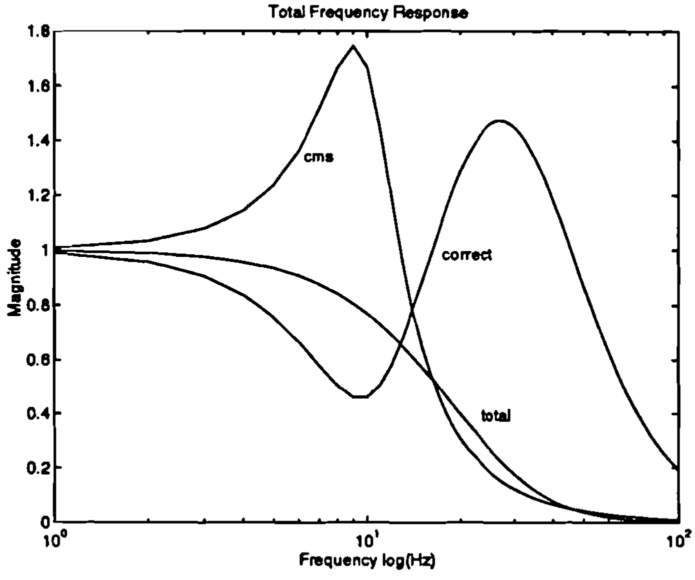


Figure 26 Correction of the CMS transfer function with the correction filter.

Instead of the elemental second order low-pass filter (34) any low-pass filter with an order of two or higher can be used. The only demand is that the correction filter (inverse plus low-pass filter) is physically realisable.

The correction filter can be implemented in many ways. The transfer function of the correction filter is known in the time-continuous frequency (S) domain. The most direct approach is to translate this transfer function to the discrete-time (Z) domain. The correction filter will then be implemented as a single digital filter. For this translation the digital filter theory provides the bilinear transformation. The bilinear transformation is discussed in paragraph 5.3.1.

An advantage of a single filter is that the corrected blood pressure sample values can be calculated simply and fast. A disadvantage is that the construction of the single filter is rather complex (because of the bilinear transformation).

Another approach to construct the correction filter is to construct a modular filter. With this method the (complex) transfer function in the S domain is `decomposed' into smaller and simpler parts. The simple parts are then implemented digitally. An advantage of a modular filter is that its parts are simple digital filters. A disadvantage is that the calculation of the corrected blood pressure samples is more complex. In paragraphs 4.4.3.1 and 4.4.3.2 two examples of modular filters will be presented. A modular approach with integrated parameter estimation will be presented in paragraph 4.4.4.

4.4.3.1 Time domain approach

In this approach the transfer function is translated back to the time domain. Let $p_A(t)$ be the actual blood pressure and $p_M(t)$ the measured blood pressure. The corresponding frequency domain blood pressures are $P_A(s)$ and $P_M(s)$. In the frequency domain the actual blood pressure can be calculated by:

$$P_A(s) = \frac{s^2 + 2\beta\omega_n s + \omega_n^2}{\omega_n^2} P_M(s) \tag{36}$$

and in the time domain:

$$p_A(t) = \frac{1}{\omega_n^2} \cdot \frac{d^2 p_M(t)}{dt^2} + \frac{2\beta}{\omega_n} \cdot \frac{dp_M(t)}{dt} + p_M(t) \tag{37}$$

A block diagram of this correction filter can be seen in Figure 27.

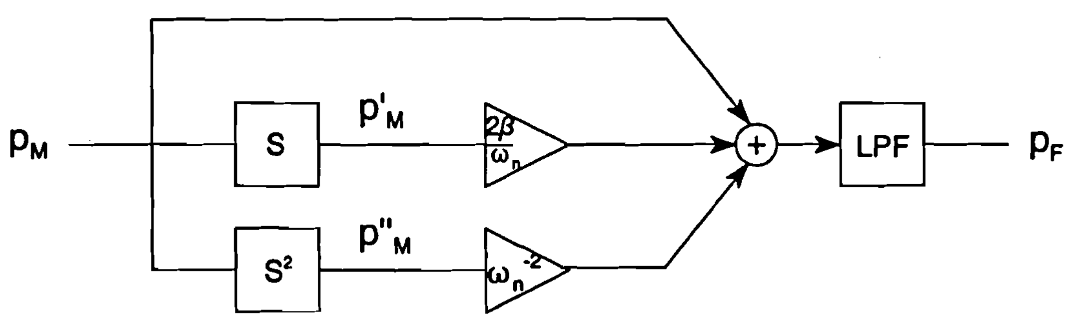


Figure 27 Block diagram of correction filter.

The differentiators S and S^2 are 'ideal' time-continuous differentiators with an amplification that is proportional to the signal frequency. In a practical situation this 'ideal' differentiator will amplify the noise (the high frequency part) in such a way the signal derivative is useless. An 'ideal' discrete-time differentiator will have the same frequency response as the time-continuous differentiator up to $0.5 \cdot f_s$ and the frequency response will be zero for frequencies above $0.5 \cdot f_s$. A practical discrete-time differentiator must have the same frequency response as the 'ideal' one up to the highest frequency in the signal that is being differentiated. For frequencies above this highest signal frequency, the frequency response must be as small as possible.

If practical differentiators are used in Figure 27, the resulting transfer function will not be (33) but the frequency response will be correct for all frequencies that are present in the blood pressure signal. In fact, a practical differentiator is an 'ideal' differentiator in cascade with a low-pass filter. The low-pass filter in Figure 27 can then be omitted.

Remarks

- Because of the differentiation in this method, the noise is amplified. Before differentiating a signal it is wise to get rid of all (high) frequencies that do not contain information.
- To keep the differentiation process stable, the sample frequency must be high enough.
- Remember to use the correct delays for p_M, p'_M and p''_M so that a correct addition is obtained.

Schwid used a method that resembles the method in this paragraph [Schwid, 1989]. The difference is that he uses an even more simplified electrical model for the CMS than that of Figure 4. In Schwid's method the catheter parameters (β and f_0) are expressed in the R, L and C of his model. The corrected pressure P_n is calculated from the measured pressure P_m and the values R, L and C by: $P_n = P_m + C \cdot R \cdot dP_m/dt + L \cdot C \cdot d^2P_m/dt^2$

4.4.3.2 Laplace transform approach

Boonzaier used another way of implementing a modular inverse filter [Boonzaier, 1978]. He described the behaviour of the CMS with the Laplace transform I(S) of its impulse response I(t):

$$I(S) = \frac{1}{(S + \beta_0)^2 + \omega_0^2} \qquad \textit{with roots: } S = -\beta_0 \pm j\omega_0 \tag{38}$$

The Laplace transform of the inverse filter $R(S)=I(S)^{-1}$ in exponential form is

$$\frac{y(s)}{x(s)} = 1 - 2e^{-\beta\tau}\cos(\omega\tau)\,e^{-S\tau} + e^{-2\beta\tau}e^{-2S\tau} \tag{39}$$

with:

- β: damping
- ω: angular resonant frequency
- τ: one sampling interval
- $e^{-nS\tau}$: n time intervals earlier

formula (39) can also be written as the recurrent relation:

$$y_n = x_n - 2e^{-\beta\tau}\cos(\omega\tau)\,x_{n-1} + e^{-2\beta\tau}x_{n-2} \tag{40}$$

The signal flow path is depicted in Figure 28.

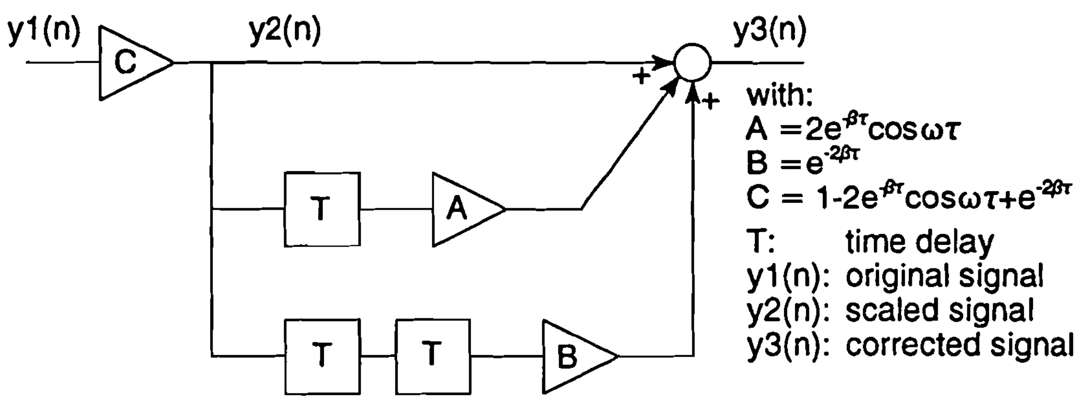


Figure 28 Block diagram of correction filter (Boonzaier).

Remarks

- In this method is necessary to filter out high frequency noise before and after the correction, because it is essentially multiplied by the process.
- Boonzaier used a sample frequency of 1 kHz because this was needed by the pattern recognition subroutine that extracts the system parameters (logarithmic decrement method).

4.4.4 Natural Observation System (NOS)

This method has an integrated parameter estimation mechanism [Saitoh et al.,1988; Hori et al., 1993]. The NOS implements the inverse filter and the low-pass filter and allows real-time correction of the blood pressure signal. Figure 29 shows a block diagram of the method based on the NOS.

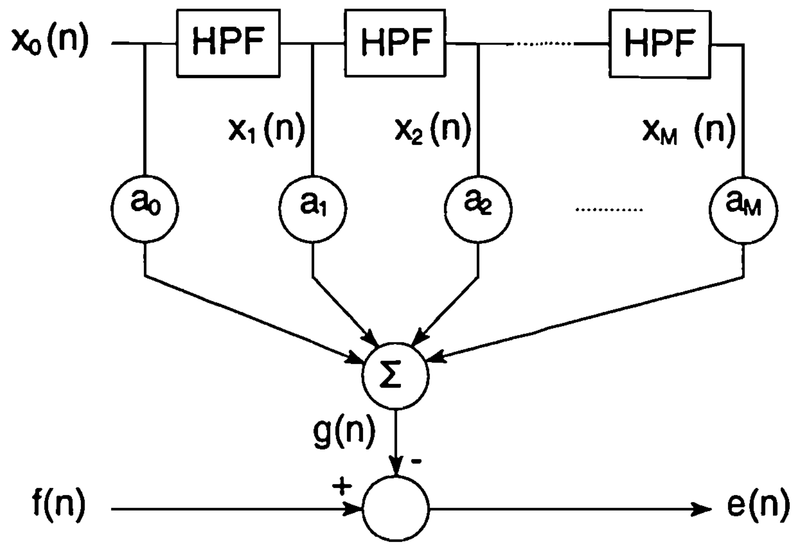


Figure 29 Block diagram of the NOS filter.

In Figure 29:

$x_0(n)$ is the discrete blood pressure signal.

$x_i(n)$ is the blood pressure signal after i-th order high-pass filtering.

HPF is a standard high-pass filter with cutoff frequency ω_c and transfer function.

$$H_{HPF} = \frac{s}{s + \omega_c} \tag{41}$$

$a_0, a_1, ..., a_M$ are the weighting parameters for each of the signals $x_i(n)$.

g(n) is the discrete corrected blood pressure signal.

f(n) is a discrete noise free reference signal.

e(n) is the discrete error signal.

The transfer function of the M th-order NOS is given by:

$$
\begin{aligned}
H_n(s) &= \sum_{i=0}^{M} a_i \left(\frac{s}{s+\omega_c}\right)^i \\
&= \left(\frac{\omega_c}{s+\omega_c}\right)^M \cdot \sum_{i=0}^{M} \frac{a_i s^i (s+\omega_c)^{M-i}}{\omega_c^M}
\end{aligned}
\tag{42}
$$

where ω_c is the cutoff frequency of the low-pass filter, and a_0, a_1, ..., a_M are the weighting parameters. The weighting parameters must be adjusted so that the Σ term in formula (42) is identical to the denominator of formula (3) (to make this possible M must be greater than, or equal to 2). The transfer function of the total system, $H_t(s)$, is given by:

$$
H_t(s) = H_{cms}(s) \cdot H_n(s) = \left(\frac{\omega_c}{s+\omega_c}\right)^M \tag{43}
$$

If the order of the CMS becomes more than two (because extra extension tubes are used) a higher order of the correction system M is required. The order of the correction system can also be greater than the order of the CMS. In that case the weighting parameters must be chosen in such a way, that the extra higher order terms are eliminated.

The high pass filters in the implementation of the NOS can be designed using the bilinear z-transformation. It is also possible to use a digital filter design routine that calculates the filter parameters directly from the design specifications.

The corrected output, g(n), described in Figure 29 is given by:

$$
g(n) = \sum_{i=0}^{M} a_i x_i(n), \quad n = 1, 2, ..., N \tag{44}
$$

with $x_0(n)$ the input signal and N the processing interval (number of samples). Assuming f(n) is a noise-free original signal, e(n) is given by:

$$
\begin{aligned}
e(n) &= f(n) - g(n) \\
&= f(n) - x_0(n) - \sum_{i=1}^{M} a_i x_i, \quad n = 1, 2, ..., N
\end{aligned}
\tag{45}
$$

N must be large enough to cover the time needed for the transient part of the observed signal.

If $H_{cms}(s)$ is a noise free signal, the NOS parameter vector $a=[a_1, a_2, ..., a_M]^T$ can be calculated theoretically. In this case the equation $H_n(s) \cdot H_{cms}(s) = 1$ must be solved. For a second order NOS the parameters are:

$$
\begin{aligned}
a_1 &= 1 \\
a_2 &= 2\beta\frac{\omega_c}{\omega_n} - 2 \\
a_3 &= \frac{\omega_c^2}{\omega_n^2} - 2\beta\frac{\omega_c}{\omega_n} + 1
\end{aligned}
\tag{46}
$$

In practical situations $H_{cms}(s)$ will not be a noise-free system. Then the optimum NOS parameter vector $a^* = [a_1^*, a_2^*, ..., a_M^*]^T$ can be determined using a least square error estimation by the following method:

$$
\begin{aligned}
f &= [f(1), f(2), ..., f(N)]^T \\
x_i &= [x_i(1), x_i(2), ..., x_i(N)]^T \\
e &= [e(1), e(2), ..., e(N)]^T \\
X &= [x_1, x_2, ..., x_M]
\end{aligned}
\tag{47}
$$

Then Eq. (45) is rewritten as:

$$e = f - x_0 - Xa \tag{48}$$

The evaluation function J is given by:

$$J = (f - x_0 - Xa)^T(f - x_0 - Xa) \tag{49}$$

Through $\delta J/\delta a = 0$, the normal equation is:

$$X^TXa = X^T(f - x_0) \tag{50}$$

The optimum NOS vector, a^*, is calculated as:

$$a^* = (X^TX)^{-1}X^T(f - x_0) \tag{51}$$

Remarks

- As reference signal f(n) a step function is chosen. The corresponding observed signal is the distorted step function. This distorted step function is obtained by using the subtraction method.
- The NOS is a simple real-time correction method with built-in parameter estimation. This parameter estimation behaves well in the presence of noise because a least squares error estimation is used.
- If the built-in parameter estimation is used, the NOS can deal with high-order effects of the CMS. If the NOS order is chosen to be i then i-th order effects can be corrected.
- Remember to use the correct delays for the signals $x_i(n)$ so that a correct summation is obtained.
- To guarantee real-time correction, the parameter estimation part must be implemented in a different program branch than the correction part.
- The optimal NOS order can be determined via the error signal e(n). The order, M, which provides the smallest e(n) can be used for the correction.
- Hori tested the NOS correction with a sample frequency of 500 Hz. The NOS order was set at 6 and the cutoff frequency for the high-pass filters was set at 50 Hz.
- Using the NOS should be considered in our project.

The NOS method resembles the method used by Bart [Bart & van Vollenhoven, 1985], who used a polynomial prediction filter to correct the blood pressure signal (measured with the CMS). The output of the prediction filter was subtracted from the blood pressure signal, and so an error signal was obtained. The coefficients of the prediction filter are determined by minimizing the error signal with a minimum square error criterion.

5 Digital Signal Processing

5.1 Introduction

As discussed before, the correction system must be fully automatic. To correct the distorted blood pressure signal with a computer, adaptive digital filters are needed. The signal processing starts with the digitalization of the blood pressure signal. The sample frequency for the A/D converter module f_s must be at least twice as high as the highest signal frequency of the sampled signal (Nyquist theorem). For the blood pressure signal with a frequency content of about 50 Hz this would mean a sample frequency of 100 Hz would suffice. In paragraph 3.4 it was explained that a sample frequency of 100 Hz was not enough for our purposes.

5.2 Digital filter types

The simplest kind of digital filters are the non-recursive filters. A non-recursive or FIR filter is defined by the linear formula:

$$y_n = \sum_{k=0}^{N} c_k u_{n-k} \tag{52}$$

The coefficients c_k are constant for the filter, the u_{n-k} are the inputs, and the y_n is the output. N+1 is the number of input samples used in the calculation of the output sample.

A digital filter that also uses old values of the output y_n in the calculation of the new output value is called a recursive or IIR filter. A recursive or IIR filter is defined by the linear formula:

$$y_n = \sum_{k=0}^{N} c_k u_{n-k} + \sum_{k=1}^{M} d_k y_{n-k} \tag{53}$$

The coefficients c_k and d_k are constants for the filter, the u_{n-k} and y_{n-k} are the inputs, and the y_n is the output. N+1 is the number of input samples and M+1 the number of old output samples used in the calculation of the new output sample.If the coefficients (c_k and d_k) vary in time the filter is called a time-variant or adaptive filter.

When a digital filter is first used there are no values u_{n-k} and y_{n-k} to calculate the new output value. These values, the initial values for the filters, are often chosen to be zero. In most cases this will lead to a sharp discontinuity in the input signal. With a FIR filter the effect of the initial values on the output will be gone after N samples. With an IIR filter, which remembers its own past, the effect of the initial values on the output can be present much longer. IIR filters thus have longer settling times than FIR filters. On the other hand IIR filter implementations can provide sharper filters with fewer coefficients.

5.3 Developing digital filters

There are generally two ways of designing a digital filter. The first way is to make a time-continuous design and then transform this design to a digital filter. The transformation that is most frequently used for this purpose is the bilinear transformation. The second, and more direct way, is to design the digital filter directly from the filter specifications.

5.3.1 The bilinear transformation

With a continuous-time design method a filter, with transfer function $H_c(s)$, is designed. This transfer function can be transferred to a discrete-time design with transfer function H(z) by substituting $s = 2 \cdot f_s \cdot (z-1)/(z+1)$. This bilinear transformation is algebraic, one-to-one and thus invertible. As required for stability, the bilinear transformation maps the left-half s plane into the interior of the unit circle of the z plane. The continuous-time frequency axis (left-half s plane) is compressed into the discrete-time frequency axis (within unit circle in z plane). The continuous-time frequency ω and the discrete-time frequency ω' are related by:

$$\omega' = 2f_s \tan^{-1}\left(\frac{\omega}{2f_s}\right) \tag{54}$$

This non-linear relationship between ω and ω' is know as frequency warping.

A design $H_c(j\omega)$ is compressed in frequency by the bilinear transformation, but otherwise, the characteristics of $H_c(j\omega)$ are preserved in $H'(\omega')$. To design a discrete-time design low pass filter with cutoff frequency f'_{lpf}, this value must first be prewarped according to:

$$\omega = 2f_s \cdot \tan\left(\frac{\omega'}{2f_s}\right) \tag{55}$$

in order to determine the appropriate value f_{lpf} for the corresponding continuous-time design.

The bilinear transformation is not suited to use with a time-continuous differentiator. The linear magnitude response of the differentiator is transformed into a nonlinear (tangent) magnitude response and an acceptable differentiator is achieved only for small ω'. The same problem also arises with continuous-time Bessel designs; its linear phase response is also distorted.

The bilinear transformation was defined as a substitution process, but this is a tedious procedure in practice. The zeros and poles together with a gain factor are all we need to specify H(z). So the easiest thing to do, is map the time-continuous zeros and poles to the discrete-time zeros and poles using the bilinear transformation. This procedure can also be easily implemented stepwise on a computer.

Step 1: Write $H_c(s)$ in its pole-zero form

$$H_c(s) = K_c \frac{\prod_{i=1}^{M} (s - z_m)}{\prod_{k=1}^{N} (s - p_k)} \tag{56}$$

As discussed earlier the relationship between the time-continuous frequency (ω) and the discrete-time frequency (ω') was nonlinear (tangent). There is one point where $H(j\omega) = H(\omega')$. It is possible to choose this point, $\omega_p = 2\pi f_p$, ourselves by prewarping the sample frequency according to:

$$f_s' = \frac{\omega_p}{\tan\left(\frac{2\omega_p}{f_s}\right)} \tag{57}$$

Step 2: Use the bilinear transformation to transform the zeros and poles:

$$zd_m = \frac{1 + \dfrac{z_m}{2 \cdot f_s'}}{1 - \dfrac{z_m}{2 \cdot f_s'}} \qquad pd_k = \frac{1 + \dfrac{p_k}{2 \cdot f_s'}}{1 - \dfrac{p_k}{2 \cdot f_s'}} \tag{58}$$

Step 3: Determine the discrete-time gain factor:

$$Kd = RE\left(K_c \frac{\prod_{i=1}^{M} (f_s' - z_m)}{\prod_{k=1}^{N} (f_s' - p_k)} \right) \tag{59}$$

Now (N-M) discrete zeros are added at z = -1. These are the zeros of $H_c(s)$ at infinity mapped to z = -1 by the bilinear transformation. We now have the discrete-time transfer function in zero-pole form:

$$H(z) = Kd(1 + z^{-1})^{N-M} \frac{\prod_{m=1}^{M} (1 - zd_m z^{-1})}{\prod_{k=1}^{N} (1 - pd_k z^{-1})} \tag{60}$$

Step 4: To get the coefficients for the digital filter (IIR) the transfer function must be written as:

$$H(z) = \frac{Kd + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{(N-1)} z^{-(N-1)}}{1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{(N-1)} z^{-(N-1)}} \tag{61}$$

These coefficients can directly be used as coefficients for the IIR filter.

First the filter coefficients c_k are determined from the zeros zd_m and Kd. We start with expanding the product term and combining the terms of the same order. If there are n zeros (the numerator is of order n) we have n+1 coefficients. The k-th coefficient of order n is called $c'_{k,n}$. The coefficients can be determined with an iterative process.

The initial coefficients $c'_{k,n}$ for a numerator of order 1 (only one zero):

$c'_{0,1} = 1$

$c'_{1,1} = -dz_1$

The following iterative formula can be used to calculate the coefficients $c'_{k,n}$ of order n from those of order n-1:

$c'_{k,n} = 1$	for $k = 0$
$c'_{k,n} = c'_{k,n-1} - zd_n \cdot c'_{k-1,n-1}$	for $k = 1$ to n
$c'_{k,n} = 0$	for $k > n$

When the coefficients c'_k for a certain order n are determined, the filter coefficients c_k can be calculated as:

$c_k = Kd \cdot RE(c'_k)$

The denominator filter coefficients d_k are determined from the poles pd_k in a similar way as the nominator coefficients c_k. First the product-term is expanded and combined (replace zd_i with pd_i). This provides the coefficient d'_k. The filter coefficients d_k can then be calculated as:

$d_k = RE(d'_k)$

Remark

- If the bilinear transformation is used to create the correction filter, the natural frequency is chosen as the prewarp frequency. This is necessary to match the 'humps' in the frequency spectra of the CMS (time-continuous frequency) and the inverse filter (discrete-time frequency).

5.3.2 Digital filter design

Digital filters can also be designed directly by choosing a transfer function in the discrete-time domain and then calculating H(z) and thus the coefficients for the filter. This method will now be explained with an example in which a low pass FIR filter with cutoff frequency f_p is designed. In the example the normalized frequency $f = f/f_s$ will be used.

Let us select the transfer function:

$$H(f) = 1 \quad \text{for } 0 < f < f_p$$
$$H(f) = 0 \quad \text{for } f_p < f < 0.5$$
$$H(-f) = H(f)$$

The function H(f) is even, so the fourier coefficients of the sine terms are all zero. The fourier coefficients for the cosine terms are:

$$a_k = \frac{2}{\pi} \int_0^{\infty} H(\omega) \cdot \cos(k\omega)\, d\omega = 4 \int_0^{\frac{1}{2}} H(f) \cdot \cos(2\pi k f)\, df \qquad (62)$$

Substituting our H(f) gives:

$$a_k = 4 \int_0^{f_p} \cos(2\pi k f)\, df = \frac{2}{\pi k} \sin(2\pi k f_p) \qquad (63)$$

The corresponding Discrete Fourier series (truncated at the fifth term) is:

$$\tilde{H} = 2f_p + \sum_{k=1}^{4} \left[\frac{2}{\pi k} \sin(2\pi k f_p) \right] \cos(2\pi k f_p) \qquad (64)$$

Truncating the Fourier series introduces the Gibbs phenomenon (ripples in the pass and stop-band of the filter). To reduce the size of these ripples a window can be used. The fourier coefficients must be multiplied with the window coefficients. If a rectangular window is chosen, the window coefficients (sigma factors) are:

$$\sigma(5,k) = \frac{\sin(\frac{\pi k}{5})}{\frac{\pi k}{5}} \qquad for\ k = 0...4 \qquad (65)$$

Thus for the modified Discrete Fourier series we have:

$$\tilde{H} = 2f_p + \sum_{k=1}^{4} \left(\frac{\sin(\frac{\pi k}{5})}{\frac{\pi k}{5}} \right) \left[\frac{2}{\pi k} \sin(2\pi k f_p) \right] \cos(2\pi k f_p) \qquad (66)$$

The digital filter coefficients c_k are half those in the cosine expansion except for the constant term ($c_0 = 2f_p$). The filter was even so $c_{-k} = c_k$ (for all k).
Instead of the rectangular window any window can be used. More about filters and windows can be found in [Hamming, 1989] and [Jackson, 1989]. Hamming also described a filter design routine using the Kaiser window (see paragraph 5.5).

Another way of designing digital filters is the use of a general filter design tool. The math-tool MATLAB, for example, has a powerful signal processing toolbox that contains a great diversity of digital filter designs. Also features like the bilinear transformation, the plotting of the magnitude and phase of a transfer function and spectral transformations are included in this library.

The digital filter design methods provided in MATLAB are:

IIR digital filter design

butter	Butterworth filter design.
cheby1	Chebyshev type I filter design.
cheby2	Chebyshev type II filter design.
ellip	Elliptic filter design.
yulewalk	Yule-Walker filter design.

FIR filter design

fir1	Window based FIR filter design - low, high, band, stop.
fir2	Window based FIR filter design - arbitrary response.
remez	Parks-McClellan optimal FIR filter design.

5.4 Filters used for our project

In this paragraph the filters used in the parameter estimation and the correction process will be discussed. The filter coefficients and plots of the magnitude and phase of the transfer functions will be provided.

5.4.1 Differentiators

In the logarithmic decrement parameter estimation the first, second and third derivatives of the signal are needed. As discussed in paragraph 4.4.3.1 a practical digital differentiator must have the same frequency response as the 'ideal' one up to the highest frequency in the signal that is being differentiated. For frequencies above this highest signal frequency, the frequency response must be as small as possible. This will prevent that the derivative is corrupted with noise.

The second and third derivative filter can be replaced by a cascade of two or three first derivative filters. This cascade implementation has two advantages. First the noise amplification is suppressed after each differentiation, which provides good derivatives. Secondly the implementation of the cascade variant is easier, because we only need one filter.

The most commonly used differentiator is the super Lanczos low-noise differentiator. Differentiators have an odd transfer function, so the Discrete Fourier series will only contain sine terms (and no constant term). The Discrete fourier series of the Lanczos differentiators with N = 2,3 and 4 are:

$$\tilde{H}_2 = \frac{8\sin(\omega) - \sin(2\omega)}{6}$$

$$\tilde{H}_3 = \frac{58\sin(\omega) + 67\sin(2\omega) - 22\sin(3\omega)}{126} \tag{67}$$

$$\tilde{H}_4 = \frac{126\sin(\omega) + 193\sin(2\omega) + 142\sin(3\omega) - 86\sin(4\omega)}{594}$$

The super Lanczos low-noise differentiator of order 4 (N = 2) is calculated as:

$$u'(n) = \frac{1}{12h}[u(n-2) - 8u(n-1) + 8u(n+1) - u(n+2)] \tag{68}$$

with:

$u'(n)$	n th sample of derivative signal
$u\,(n+i)$	n+i th sample of differentiated signal ($-N \le i \le N$)
$h = 1/f_s$	

A plot of the magnitude and phase of the transfer functions of the super Lanczos low-noise differentiators for N = 2, 3 and 4 can be seen in Figure 30.

As mentioned earlier a practical digital differentiator must have the same frequency response as the 'ideal' one up to the highest frequency in the signal that is being differentiated. A practical flush response signal contains an oscillation with a maximum natural frequency of about 40 Hz. Looking at Figure 30 it can be seen that, with a sample frequency of 200 Hz, a super low-noise Lanczos differentiator with N = 3 would suffice.

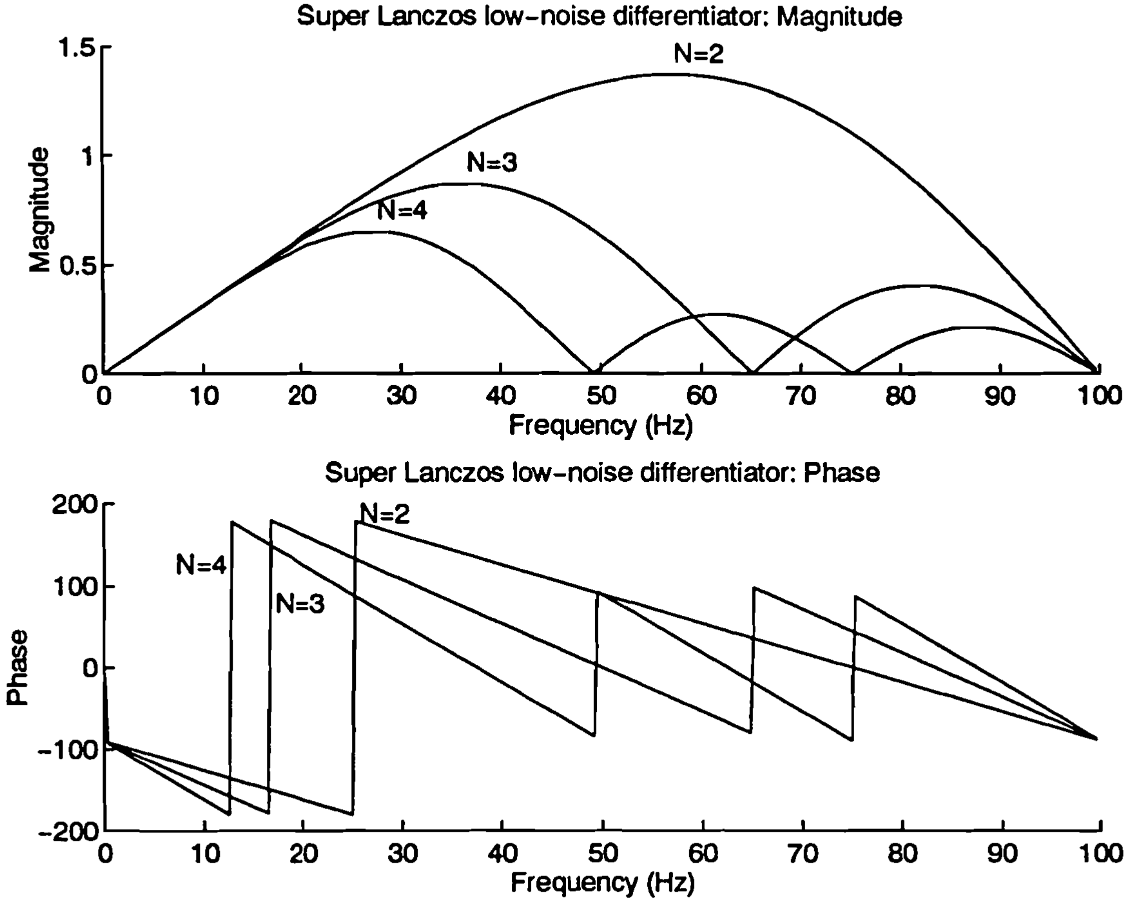


Figure 30 *Magnitude and phase of low-noise Lanczos differentiators of order 2,3 and 4.*

A differentiator designed with the Parks-McClellan optimal FIR filter design (MATLAB: remez) is

$$u'(n) = 0.0424\,[u(n-2) - u(n+2)]$$

This filter has order 4 and a cutoff frequency of 30 Hz. A plot of the magnitude and phase response of this filter can be seen in Figure 31.

By choosing the cutoff frequency for the differentiating filter we can make a specific design based upon the frequency contents of the differentiated signal. Remez designs of order 4 with cutoff frequencies of 20, 30, 40 and 50 Hz can be seen in Figure 32.

The noise suppression in the stop-band of the differentiator depends on the filter order used. Remez designs with a cutoff frequency of 30 Hz and order 4, 8 and 32 can be seen in Figure 33.

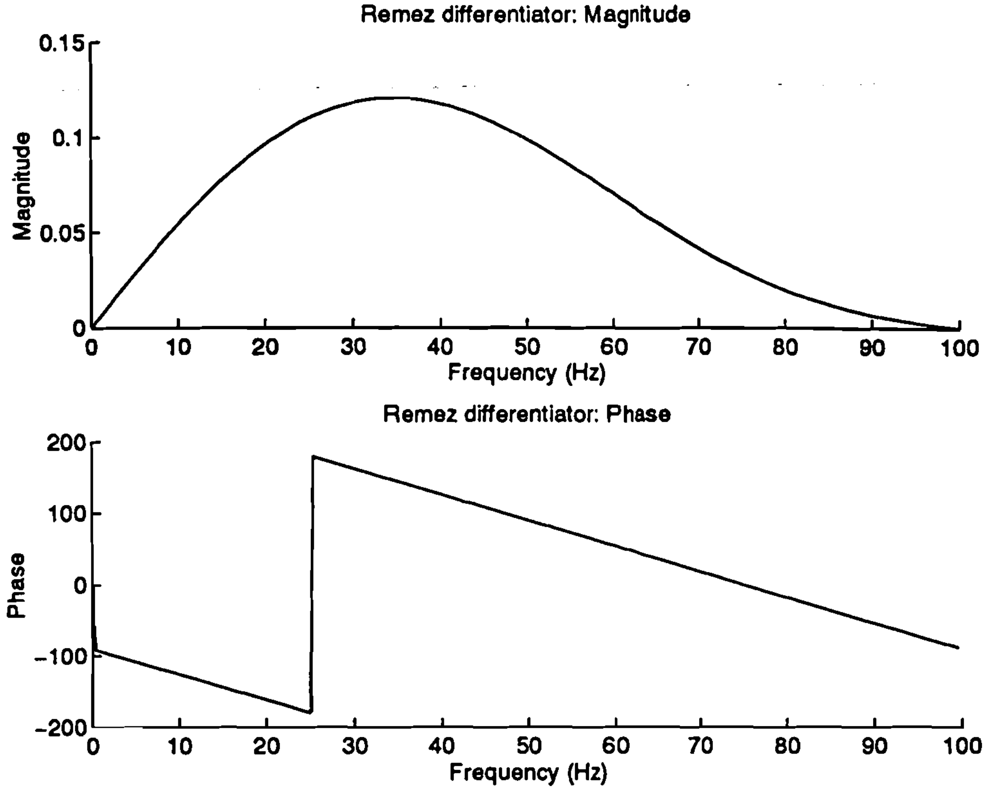


Figure 31 *Remez differentiator with cut-off frequency f_c=30 Hz and order 4.*

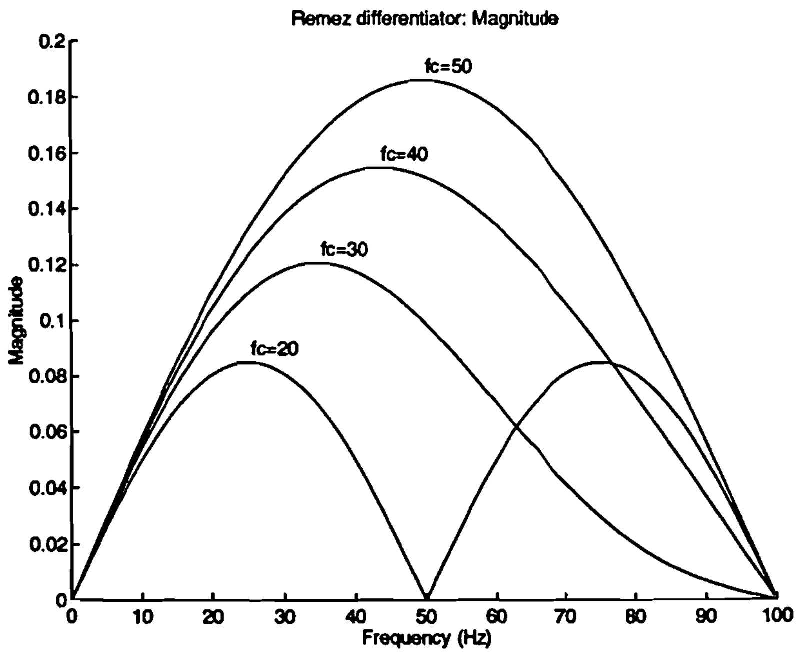


Figure 32 *Remez designs with order 4 and f_c = 20, 30 40 and 50 Hz.*

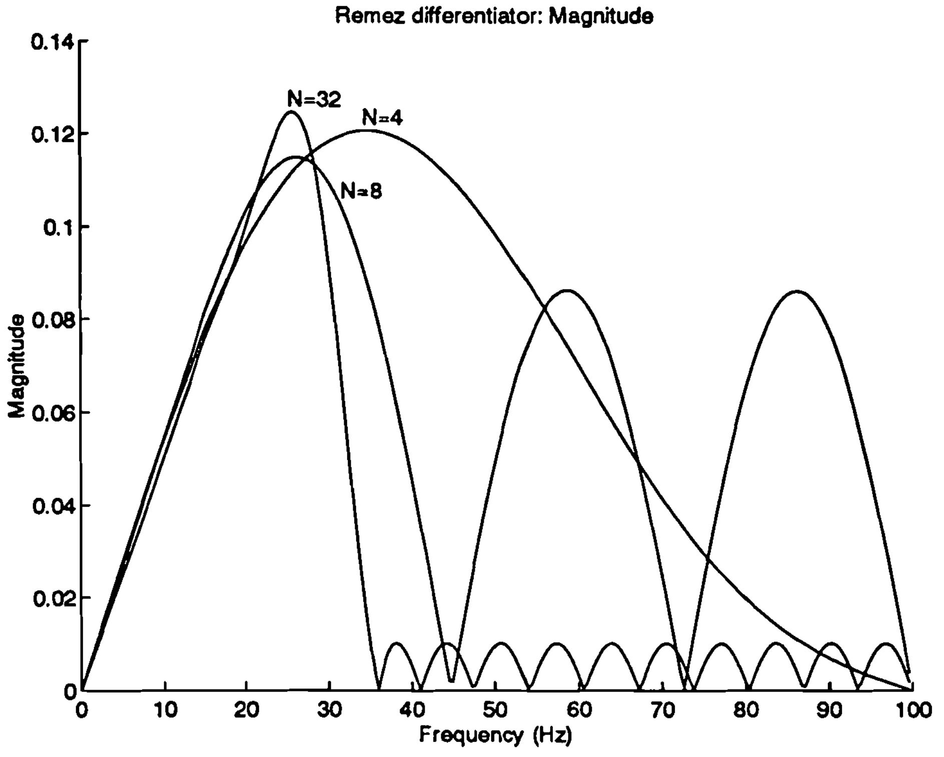


Figure 33 *Remez designs with cut-off frequency f_c = 30 Hz and order 4, 8 and 32.*

5.4.2 Low-pass filters

In the correction process a low-pass filter is cascaded with the inverse filter to obtain a realizable (stable) filter. The first filter used was the 'standard' second order low-pass filter (34) in paragraph 4.4.3. A plot of this time-continuous filter can be seen in Figure 34 (solid line).

Using the bilinear transformation to transform this filter to a discrete-time filter, with the cutoff frequency as prewarp frequency provides the filter coefficients c_k and d_k:

$$c_k = [\ 0.1139 \quad 0.2279 \quad 0.1139\ ] \quad \text{for } k = 0..2$$
$$d_k = [\ 1.0000 \quad -0.6498 \quad 0.1056\ ] \quad \text{for } k = 0..2$$

A plot of this discrete-time filter can also be seen in Figure 34 (dotted line).

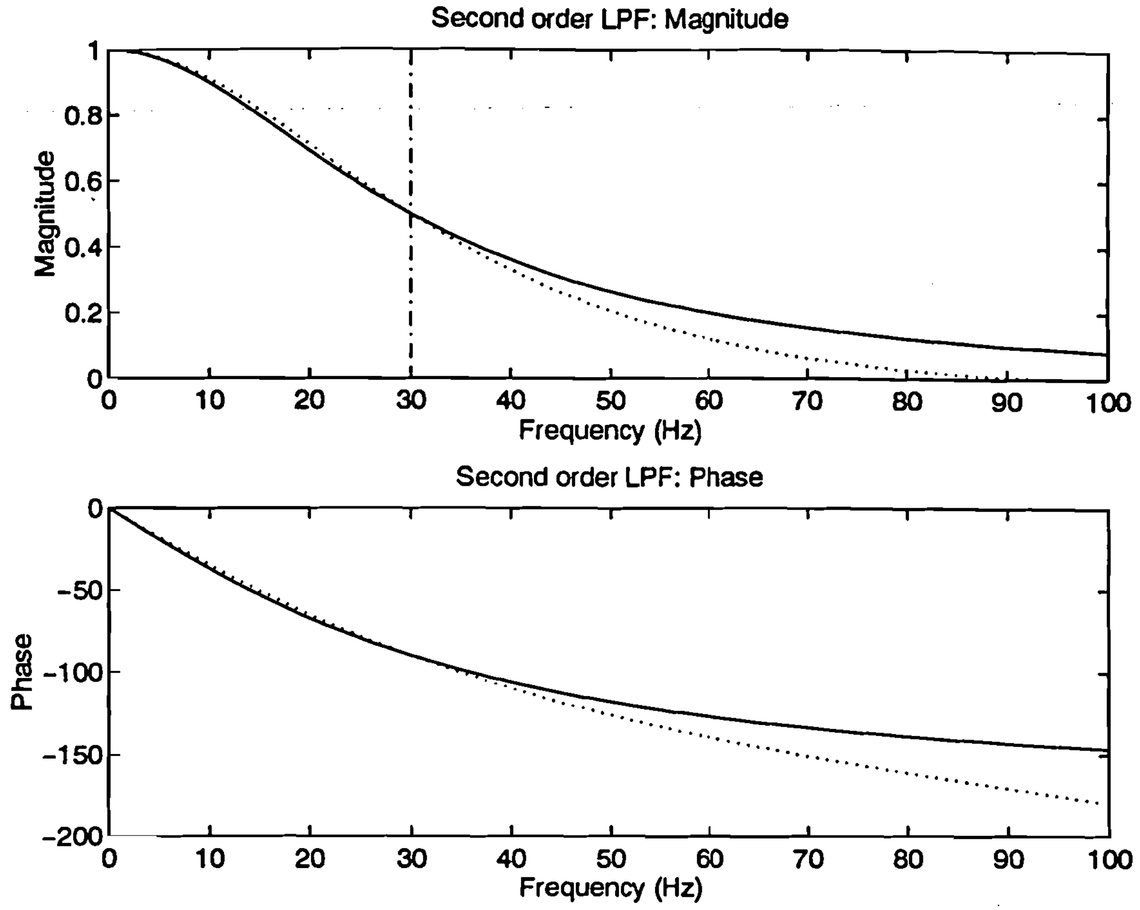


Figure 34 *Second order low-pass filter design with f_c = 30 Hz; continuous-time (solid line) vs. discrete-time (dotted line).*

The magnitude plot of the total transfer function of CMS with the correction filter using this low-pass filter (Figure 26) showed not the ideal 'flat' frequency response. If a low-pass filter with a flat frequency response in its pass-band is used, the ideal 'flat' frequency response would be achieved. Besides the 'flat' response also a linear phase of the total system transfer is very important. A linear phase causes a uniform delay for all frequencies. If we do not have this linear phase all frequencies in our signal would get a different delay and the signal reconstruction would fail.

A filter that has a flat frequency response in its pass-band and a linear phase response, is a chebyshev type 2 filter. A second order chebyshev type 2 filter with a cutoff frequency of 30 Hz and a maximal ripple of 10 dB in the stop-band has the filter coefficients:

$$c_k = [\ 0.2933 \quad -0.1856 \quad 0.2933\ ] \qquad \text{for } k = 0..2$$
$$d_k = [\ 1.0000 \quad -1.0205 \quad 0.4214\ ] \qquad \text{for } k = 0..2$$

A plot of this filter can be seen in Figure 35.

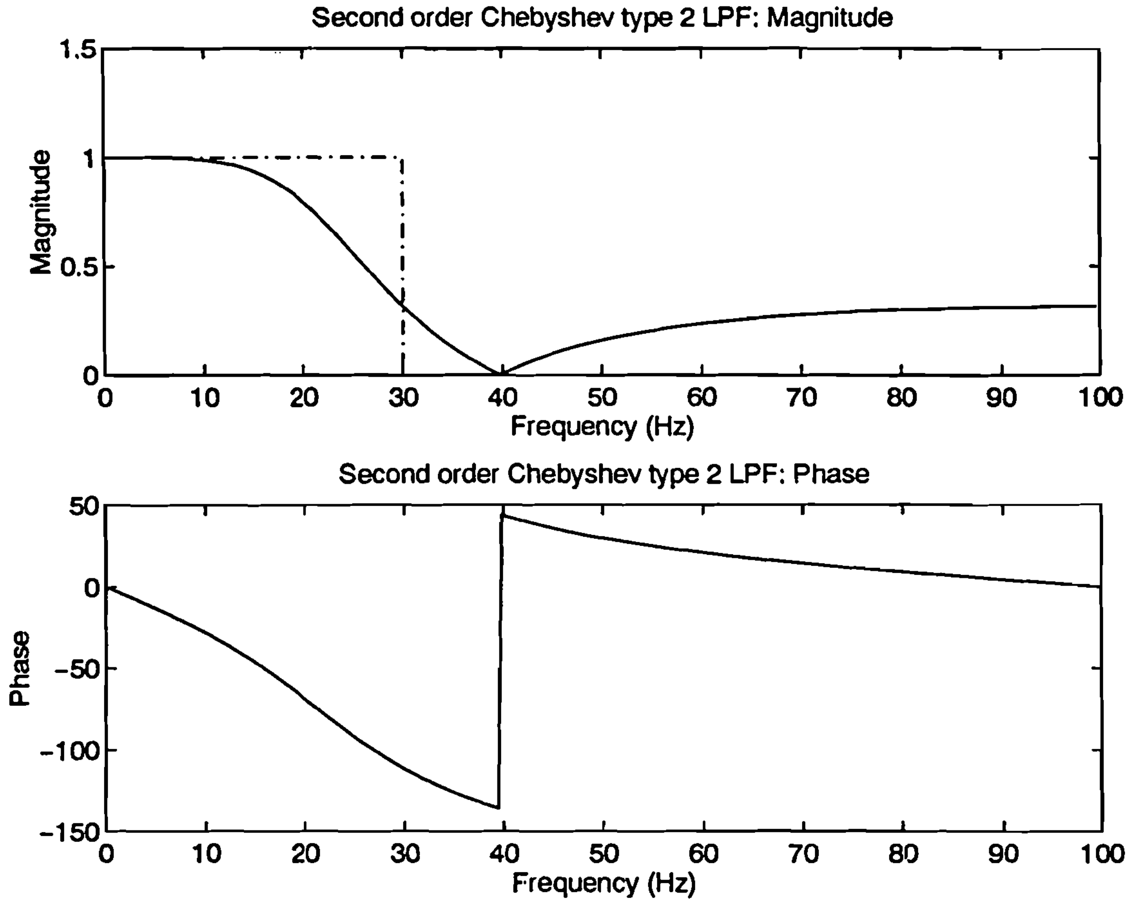


Figure 35 *Second order chebyshev type 2 design low-pass filter with f_c=30 Hz and ripple=10 dB.*

The design of chebyshev filters is a trade-off between filter sharpness and allowed ripple size. If the maximal allowed ripple is increased, the filter will cutoff sharper. Chebyshev type 1 and 2 designs of fourth order can be seen in Appendix 3. An alternative for the type 2 design is the type 1 design, which allows a ripple in the pass-band. If we make this allowable ripple small enough a type 1 design can best be used in our correction filter. The type 1 design has the advantage that the magnitude of the transfer function is zero in the pass-band (and this is necessary to suppress the high frequency amplification of the inverse). A type 1 design with f_c = 30 Hz and an allowable ripple of 0.1 can be seen in Figure 36.

The total correction filter with a chebyshev type one filter with f_c = 30 Hz and an allowable ripple of 0.01 (completely flat in the pass-band and stop-band) can be seen in Figure 37. This correction filter does have the ideal 'flat' frequency response in the band that contains the blood pressure information. Also, the phase is fairly linear.

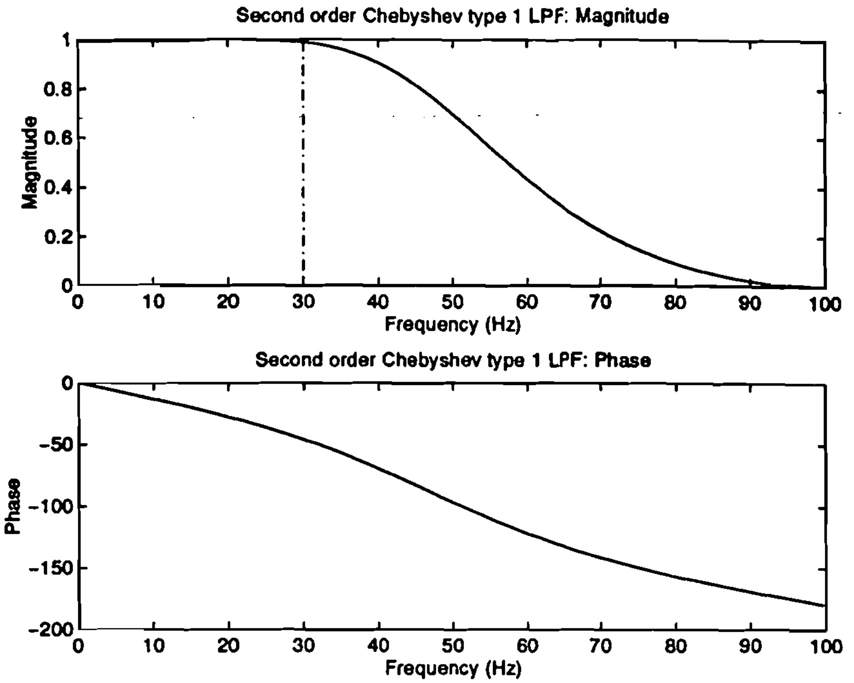


Figure 36 *Second order chebyshev type 1 design low-pass filter with f_c=30 Hz and ripple=0.1 dB.*

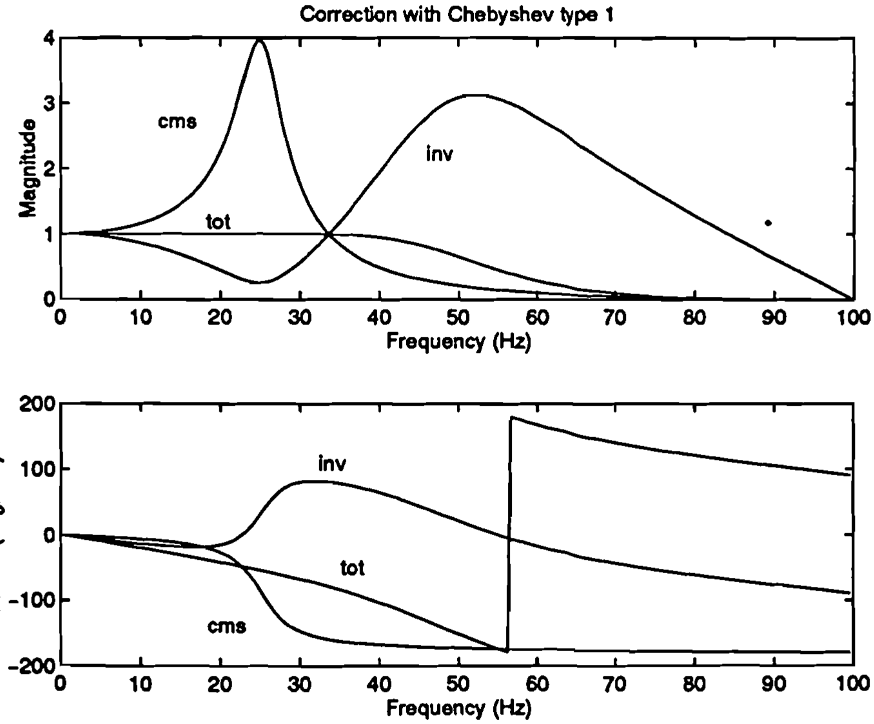


Figure 37 *Magnitude and phase of CMS, correction and total transfer function. Correction filter uses chebyshev type 1 low-pass filter with f_c=30 Hz and ripple = 0.01 dB.*

5.4.3 High-pass filters

In the implementation of the NOS correction a first order high-pass filter (41) is needed. This filter is a time-continuous filter and so the bilinear transformation must be used to transform the filter to a discrete-time filter with the pass-frequency as prewarp frequency. The filter coefficients c_k and d_k for this filter are:

$$c_k = [\ 0.6625 \quad -0.6625\ ] \qquad \text{for } k = 0,1$$
$$d_k = [\ 1.0000 \quad -0.3249\ ] \qquad \text{for } k = 0,1$$

A plot of the time-continuous and the discrete-time filter can be seen in Figure 38.

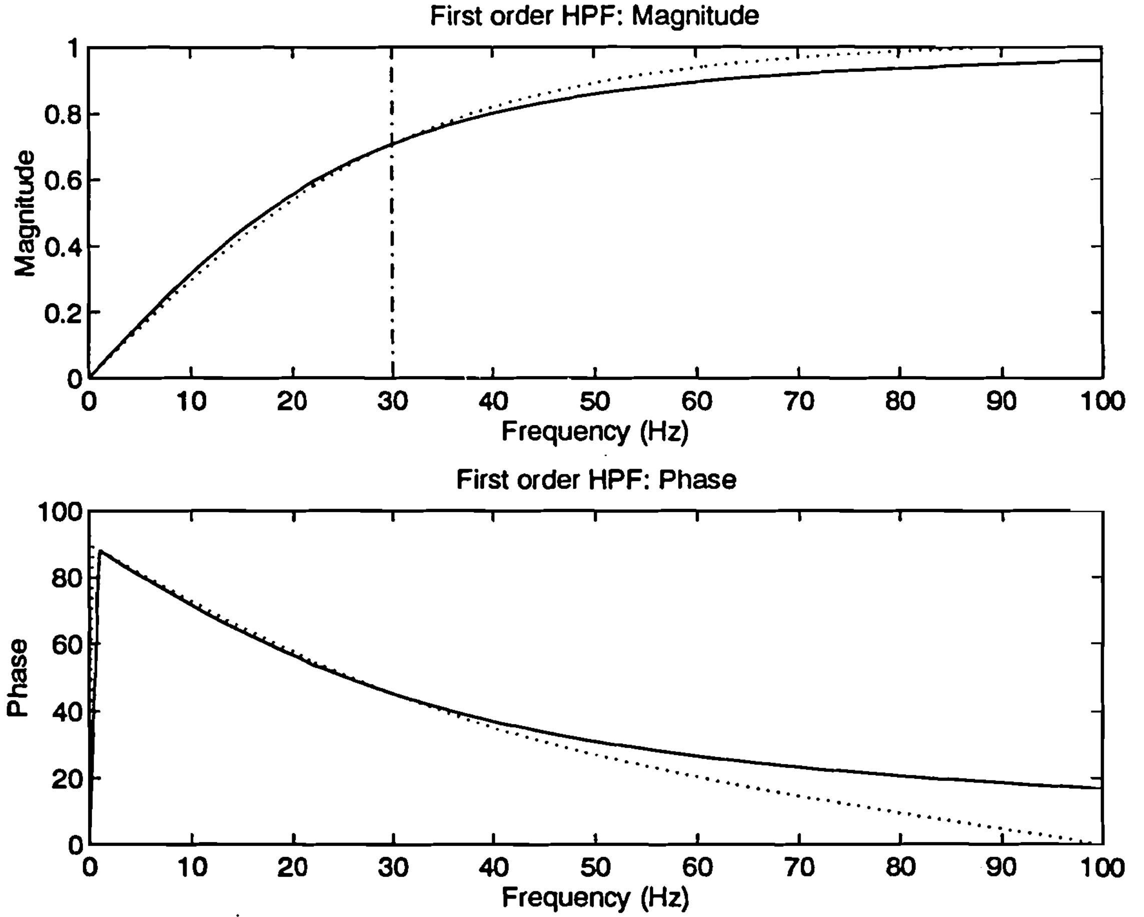


Figure 38 *First order high-pass filter design with f_c = 30 Hz; continuous-time (solid line) vs. discrete-time (dotted line).*

5.4.4 Smoothing filters

If the blood pressure signal contains a lot of noise, the parameter estimation and the correction procedures may not work properly. For the logarithmic decrement procedure the natural frequency is the only frequency of interest. A smoothing filter (or low-pass filter) will eliminate high frequencies in the signal (mostly noise) and make it smooth. If the frequency component at f_n remains strong enough, the effect of smoothing on the logarithmic decrement process (and the differentiators used in this process) will be positive.

For the correction process only the band that contains the blood pressure information is of importance. All higher frequencies can be eliminated. This pre-filtering will have a positive effect, because the inverse filter tends to amplify high frequencies (and thus the noise).

5.5 Digital filter design routines

Two design programs for digital filters have been implemented in Turbo Pascal. These filter design routines can also be implemented as procedures. In this way programs that use these procedures can design filters during run-time. The first is the low-pass filter design routine described in paragraph 5.3.2. This routine is implemented in the program FILDES.PAS. The program asks the user to input the filter order (N) and the cutoff frequency (f_c) for the low-pass filter. The sample frequency is implemented as a program constant and is set at 200 Hz. The program then outputs the low-pass filter coefficients to the screen and to a file. The name of this file is constructed from the filter parameters as:

Filename = N<order>Fc<cutoff frequency>.ftr

The terms <order> and <cutoff frequency> are replaced with the corresponding numbers entered by the user. Besides the coefficients for the low-pass filters the program also outputs the corresponding coefficients for a high-pass filter with f_c as cutoff frequency. These coefficients can be found with use of the spectral transformation.

The second routine is the kaiser filter design routine mentioned in paragraph 5.3.2. This routine was described in [Hamming, 1989] on page 197. This routine is implemented in the program KAISER.PAS.

The program asks the user to input:

δ	maximum allowed ripple in the pass- and stop band ($0 < \delta \leq 1$)
ΔF	transition width in Hz ($\Delta F <$ abs(f_s-f_p))
f_p	filter pass frequency in Hz ($0 \leq f_p \leq f_{sample}$)
f_s	filter stop frequency in Hz ($0 \leq f_s \leq f_{sample}$)

The function of these parameters is depicted in Figure 39.

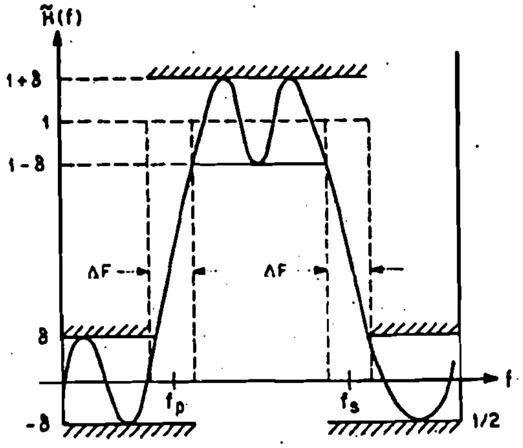


Figure 39 Parameters in the kaiser filter design.

The two parameters f_p and f_s define the type of filter.

- If we choose $f_p = 0$ we create a low-pass filter with cutoff frequency f_s.
- If we choose $f_s = 0.5 \cdot f_{sample}$ we create a high-pass filter with cutoff frequency f_p.
- If we choose f_p and f_s between 0 and f_{sample} with $f_p<f_s$ we create a band-pass filter with a pass band from f_p to f_s.
- If we choose f_p and f_s between 0 and f_{sample} with $f_p>f_s$ we create a band-stop filter with a stop band from f_s to f_p.

The sample frequency is implemented as a program constant and is set at 200 Hz. The program then outputs the filter coefficients to the screen and to a file. The name of this file is constructed from the filter parameters as:

Filename = Fp<f_p>Fs<f_s>.ftr

The terms <f_p> and <f_s> are replaced with the corresponding numbers entered by the user.

6 Blood pressure analysis

6.1 Demands

A suitable blood pressure algorithm for an analyzing arterial blood pressure signal must have the following properties:

- It must extract the following features from the blood pressure signal:

SBP:	Systolic blood pressure
DBP:	Diastolic blood pressure
MEAN:	Average blood pressure over one period
HR:	Heart-rate in beats per minute

- It must be suited for real-time blood pressure analysis of several blood pressure signals simultaneous.
- It must provide a trigger point that can be used to synchronize other processes with the blood pressure signal (for example the data-output and the flush-positioning process).
- It must be a 'safe' algorithm. This means that if the tracking of the blood pressure signal is lost, the algorithm must be able to restart itself.

6.2 Blood pressure simulator

The blood pressure signals that are used to test the algorithm were recorded with several types of CMSes (see [Alofs, 1993] paragraph 7.2). Instead of a human patient, a blood pressure simulator was connected to the CMSes. This blood pressure simulator is a hydraulic pressure generator that produces several pressure signals that resemble human blood pressures. These test signals contain flushes, so they can be used to test the parameter estimation algorithms. Because these signals were distorted with 'real' CMSes they can also be used to test the correction algorithms.

6.3 Algorithm

A blood pressure algorithm that satisfies our demands was described by J.A. Blom in [Blom, 1990]. A representative period of an arterial blood pressure of a blood pressure simulator, can be seen in Figure 40.

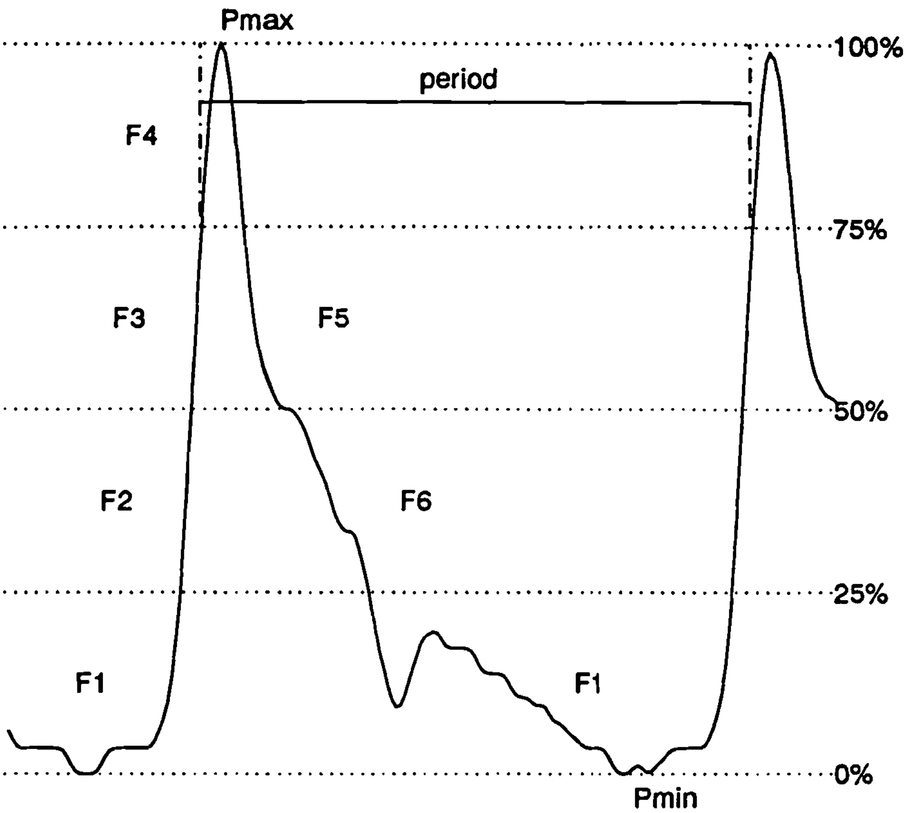


Figure 40 *A typical period of the arterial blood pressure with auxiliary levels and the states of the blood pressure algorithm.*

The blood pressure signal is segmented by a set of auxiliary lines at 0%, 25%, 50%, 75% and 100% level. These segments are then named F1 to F6. The feature extraction process can then be described as a state-machine (see Figure 41).

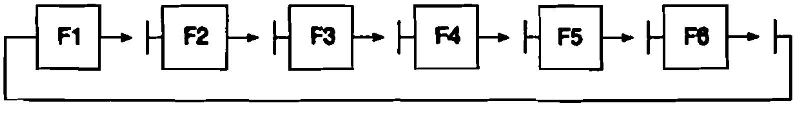


Figure 41 *State-machine representation of the arterial pressure feature extraction process.*

The states represent the signal segments (F1-F6) and the state-transitions represent the crossing of the levels. Besides the transitions F1->F2, F2->F3, F3->F4, F4->F5, F5->F6 and F6->F1 also the transitions F2->F1 and F3->F2 were needed to cope with a possible extra peak in the downstroke.

The heart-rate (HR) will be determined at the 75% level to provide good noise-immunity. The 75% level will also be used to update the model parameters and produce the trigger point. In the state F4 we search for the signal maximum. In the states F6 and F1 we search for the signal minimum.

The auxiliary levels are determined from the maximum (systolic value) and minimum (diastolic value) of the blood pressure signal. The blood pressure parameters (SBP, DBP, MEAN and HR) and the auxiliary level derived from these parameters form the model for the blood pressure signal. Before the auxiliary levels can be calculated we need values for the systolic and diastolic pressure. To get initial values for the systolic and diastolic value we add a learning state to the state-machine. In this learning state L we search for the absolute minimum and maximum of the blood pressure signal, and use them as SBP (maximum) and DBP (minimum). This leaning period must be long enough to hold one blood pressure period. If we assume a minimum heart-rate of 30 BPM we will need a learning period of two seconds. After the learning period we need to synchronise, that is enter state F1 at the correct blood pressure level.

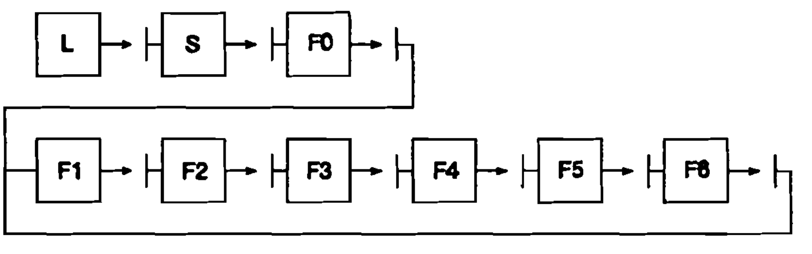


Figure 42 *A state-machine representation of the arterial pressure feature extraction process, including start-up and synchronisation states.*

For the synchronisation we add two extra states S and F0 (see Figure 42). When the learning is finished (after two second) the results are verified against limits. If the results are not within these limits we re-start the learning process. If the results are within these limits we enter state S, the first synchronisation state. The transition between state S and F0 is taken as soon as a sample > 75% is found. The transition from F0 to F1 is made when the blood pressure signal falls below the 25% level.

At some point in the cycle all features are determined. The 75% level crossing was chosen for this point. The results can then be validated against the limits and against the model. In the validation the running-average of the parameters SBP, DBP, MEAN and HR are used as model.

$$\text{model SBP}(n) = w \cdot \text{feature } P_{max}(n) + (1-w) \cdot \text{model SBP}(n-1)$$
$$\text{model DBP}(n) = w \cdot \text{feature } P_{min}(n) + (1-w) \cdot \text{model DBP}(n-1)$$
$$\text{model MEAN}(n) = w \cdot \text{feature } P_{mean}(n) + (1-w) \cdot \text{model MEAN}(n-1)$$
$$\text{model HR}(n) = w \cdot \text{feature HR}(n) + (1-w) \cdot \text{model HR}(n-1)$$

in which n is a counter that is increased each time new valid features are found ($n \geq 1$). The factor w defines a 'time window' for averaging. If $w = 0$ the new feature has no influence on the model (not adaptive model). If $w = 1$ the model is very adaptive, no old model values are used. Depending upon the maximum possible change of the feature, a value for w can be chosen between 0 and 1. Blom stated that a value of 0.2 generally works well [Blom, 1990]. The initial values for the model are:

model SBP(0) = feature P_{max} determined in learning state
model DBP(0) = feature P_{min} determined in learning state
model MEAN(0) = feature P_{mean} determined in learning state
model HR(0) = feature HR(0)

The limits for the features are presented in Table 1

Table 1 Acceptable limits for the features of the arterial blood pressure

unit	mmHg			ms
feature	P_{max}	P_{min}	P_{mean}	HR
minimum	40	20	30	200
maximum	270	200	230	2000
difference	10	10	10	200

The table should be read as follows: A feature is accepted if it is between the minimum and the maximum and if it differs less than difference from the model value. If all features are valid the model parameters can be updated, if not the features will be ignored (the presence of an artifact is assumed). Because the 75% level is used as the validation (and trigger) point, an extra peak higher than 75% in the blood pressure signal will cause problems. After this peak, which will be seen as an invalid 75% level, the feature extraction starts all over (all temporary parameters are reset). In case of an extra peak we want to continue until the valid 75% level is found (the end of the BP period). For an extra peak the tests for the P_{max} feature will pass, but all other tests can fail. If the feature HR is still smaller than the model HR and the P_{max} feature is valid, we continue the feature extraction without a reset. Else the feature extraction will be reset.

If the synchronisation is lost for a short time (for example due to an artifact), we need to re-synchronise. If the synchronisation is lost for a longer time, the model parameters may not be valid anymore. Then re-learning is necessary. To enable re-synchronisation and re-learning two timers are needed.

The first timer is the state-timeout counter. This counter is increased each sample. If the process remains in a state for longer than the state-timeout, re-synchronisation will take place (the process will be forced into state S). The state-timeout is set at two seconds (equal to the maximum period time). The state-timeout counter is reset on every state transition.
The second timer is the data-valid-timeout counter. The blood pressure model is updated at every 75% level unless the features were found to be invalid. The counter is increased each time the 75% level is reached. If the counter is greater than data-valid-timeout, something is definitely wrong and re-learning is necessary (the process will be forced into state L). Data-valid-timeout is set at 5 (maximal five periods with no update allowed). The counter is reset after every update of the model (valid features at 75% level).

6.4 Implementation

Elements needed in the implementation:

variables

state	state variable with value from (L,S,F0,F1,F2,F3,F4,F5,F6)
model	variable that holds the model (SBP,DBP,MEAN,HR,25%,50%,75%,100%)
features	variable that holds all features (P_{max},P_{min},P_{mean},HR)
sample	variable that holds the latest BP sample value

procedures

init	initialisation procedure that resets all variables (state will be set to L and all other variables are set to 0).

analyze	procedure that contains the analysis algorithm
extract_data	procedure that calls the analysis procedure and outputs the results when the trigger is set.

special tasks

re-synchronize	sets state to S
re-learn	call init procedure

For each blood pressure signal that must be analyzed, a set of variables (state, model and features) is needed together with a procedure extract_data. The blood pressure sample is passed to the extract_data procedure that calls the analysis procedure with the correct set of variables. The blood pressure analysis algorithm is implemented in Turbo Pascal. The implementation can be found in the unit BEAT.

6.5 Tests

The blood pressure algorithm was tested with pre-recorded blood pressure signals from a blood pressure simulator (see paragraph 6.2). These blood pressure signals were recorded with different types of catheter-manometer systems. The algorithm was tested with four blood pressure files from each of the eight sets of measurements. These blood pressure files also contain flushes. A flush response should be interpreted as an artifact by the algorithm. One set also contained a blood pressure signal with an extra 75% peak. Another blood pressure recording had several flush responses at the beginning of the file (close together).

In these tests the algorithm worked properly. The algorithm detected each flush, and re-synchronized properly. The extra 75% peak was correctly ignored by the algorithm. For the file with the several flushes at the beginning of the file, the algorithm re-started the learning phase as it should.

The blood pressure signals from the pressure simulator are very 'constant' (stable values for SBP, DBP, MEAN and HR). To get a complete impression about the blood pressure algorithm, we also need to test it with 'real' blood pressure signals. In the set pre-recorded signals with flushes, also one blood pressure recordings of a pig were available. In this animal blood pressure recording, the respiratory pressure influence on the blood pressure is visible. The pig's blood pressure with this respiratory influence can be seen in Figure 43.

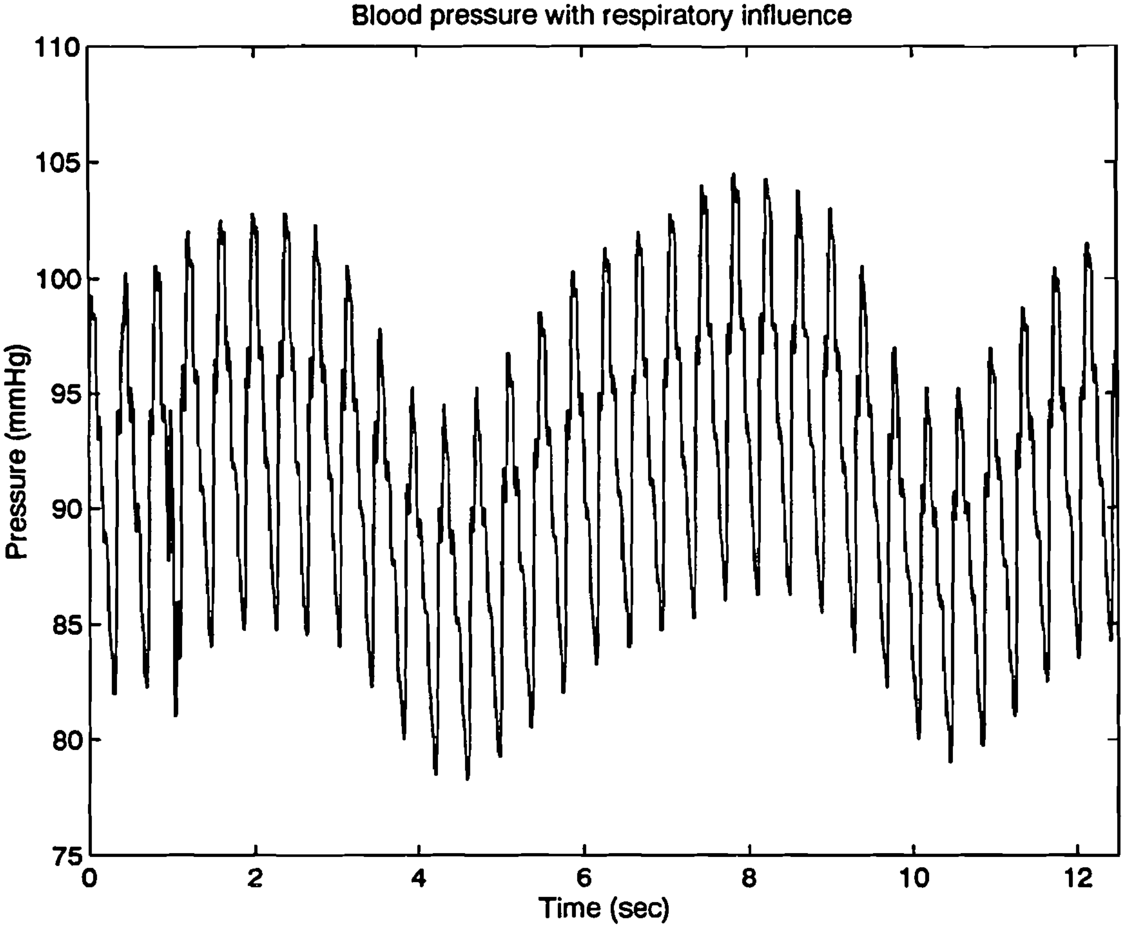


Figure 43 *Blood pressure signal of a pig with strong respiratory effect.*

The blood pressure algorithm lost tracking with this blood pressure file. The tracking is lost due to this respiratory effect in combination with a small pulse pressure. The model used in the blood pressure algorithm uses the moving average values of the features as a model. The actual features of this blood pressure recording vary faster than the moving average model is updated. The tests in the algorithms against the model values cause the error also because the pulse pressure is very low.

A solution to this problem is a faster update of the blood pressure model (choose a larger w). With this blood pressure however, the error remains even if the model is updated every period (w = 1). Whether extreme situations like this one also exist for human blood pressures or not, must be tested (with prerecorded human blood pressure signals) before the clinical tests begin.

The algorithm as described in paragraph 6.3 is not equipped to deal with strong respiratory effects as noticed in the pig's blood pressure signal. To deal with these effects the arterial pressure model must be changed. A blood pressure analysis algorithm with a more complex arterial pressure model was described by Plasman [Plasman, 1981]. More about blood pressure fluctuations and heart-rate variability can be found in the thesis by de Boer [de Boer, 1985].

The results are promising. The only test that failed was the blood pressure measurement of the pig with the strong respiratory effect. Tests with 'real' human blood pressure measurements must be used to fine-tune all the parameters (time-outs, boundaries etc.) of the algorithm.

7 Software

7.1 Program specifications

To test all the algorithms a computer program is needed. This program must be able to perform the following tasks:

- Sample blood pressure signals with an A/D converter board.
- Display and analyze blood pressure signals.
- Control the automatic flush device.
- Use the parameter estimation methods to determine the CMS parameters.
- Use the correction methods to correct the blood pressure signal.
- Display and analyze the corrected blood pressure signal.

During the further development of the algorithms it is very important to have access to all intermediate results and to control all algorithm options from the program menu. In this way the algorithms can be easily evaluated during tests. Problems and errors can be located and corrected quickly. When the program is ready for use, the program must be as simple and fail-safe as possible. In this case the program menu should only contain the basic program functions.

7.2 Menu structure

In this paragraph the menu structure of the program, which will be called MONITOR, in the development phase will be discussed. The menu's described are displayed at the bottom of the screen in a statusbar. Menu items can be chosen using the function keys (F1, F2 etc.).

Main menu:

F1 Quit	Exit program.
F2 Off-line Analysis	Off-line analysis menu (play, analyze and correct prerecorded blood pressure signals from disk).

F3 On-line Analysis	On-line analysis menu (real-time blood pressure analysis and correction). The on-line part of the program is not implemented yet. All remarks about it should be seen as implementation advice.
F4 Display Options	Set the display options for the blood pressure windows.
F5 Test Filter	Filter test menu to test the characteristics of the filter used on the blood pressure signal.

Off-line menu:

F1 Main Menu	Return to main menu.
F2 Load	Load a blood pressure data-file from disk.
F3 Play	Display, analyze and correct the blood pressure signal. If a flush response is found, the parameter estimation routines will be called to determine the CMS parameters and the correction filter will be updated.
F4 Display Options	Set the display options for the blood pressure windows.
F5 Data Directory	Select the directory for the blood pressure data files.
F6 Set Parameters	Manually set the system parameters (β and f_0).
F7 DPE On/Off	Enable/Disable the dynamic parameter estimation algorithm.

On-line menu:

F1 Main Menu	Return to main menu.
F2 Display Options	Set the display options for the blood pressure windows.
F3 Data Directory	Select the directory for the blood pressure data files.
F4 Store On/Off	Enable/Disable the storage to disk. If enabled a blood pressure signal part around the flush response (with a length of 25 seconds) will be saved to disk.
F5 Set Parameters	Manually set the system parameters (β and f_0).
F6 DPE On/Off	Enable/Disable the dynamic parameter estimation algorithm.
F7 Flush	Manually start flush response test.

Display options menu:

F1 Done	Return to main menu.
F2 Windows	Toggle between a single blood pressure window (for measured and corrected BP) and two windows (one for the measured BP and one for the corrected BP).
F3 Zoom in	Zoom in for the time axis.
F4 Zoom out	Zoom out for the time axis.

F5 mmHg Range	Set the axis values for minimum and maximum blood pressure (in mmHg).

In the Off-line and On-line menus the option DPE On/Off can be replaced by an algorithm menu, which enables the user to choose the algorithms that are used in the parameter estimation and the correction process. This menu could look like this:

Algorithm menu:

F1 LogDec mode	Set the mode for the logarithmic decrement method. Possible values can be C(onstant), L(inear), Q(uadratic) and A(ll).
F2 Subtract On/Off	Enable/Disable the subtraction method.
F3 DPE On/Off	Enable/Disable the dynamic parameter estimation algorithm.
F4 Karnopp mode	In case the system is not underdamped the karnopp method must be used. Possible values for Karnopp mode can be M(oderate damping) and H(eavy damping).
F5 Inv&LPF On/Off	Enable/Disable the single filter correction. The low-pass filter can be a standard LPF or the chebyshev filter. The order of the LPF and its cutoff frequency can be specified.
F6 NOS On/Off	Enable/Disable the NOS correction (with or without built-in parameter estimation).

7.3 Off-line operation

With Off-line operation the program part is meant that plays back the pre-recorded blood pressure files from disk. In this mode we do not need any real-time processes. For reasons of efficient programming we will try to implement the off-line and the on-line mode in such a way, that they use the same procedures for the parameter estimation and the correction processes.

7.3.1 Storage

The blood pressures will be stored on disk in the ascii format. Choosing the ascii format has the advantage that the blood pressure files can be read with a simple editor and that they can be imported into many other programs (such as for example MATLAB). The structure of the data-files is described in the M.Sc. report of Jos Alofs [Alofs, 1993]. In these files each line contains the measured sample value, the corrected sample value, a boolean to indicate the position of the flush-valve and a code for the used method.

A more efficient structure for the data-files would be:

% comment	
% header	
n	number of samples in the file
f	number of flushes in the file
f pairs (o, c)	'o' is the index of the sample that the flush-valve was opened, 'c' is the index of the sample the flush-valve was closed
% data	
samples	list with n samples

Textblock 1 Data-file structure.

Like the blood pressure analyze procedure, the off-line procedure is implemented on a sample to sample basis. If each sample must be read from disk separately, the procedure would be very slow. Therefore, all the samples will be read into a memory buffer together with the position of the flushes. This buffer will be implemented as an array.

7.3.2 Structure of Off-line

The off-line procedure is implemented as a state machine within a loop. The loop is executed for each sample, as long as there are samples available. The states symbolise the status of the analysis process. The three possible states are: MONITORING, FLUSHING and RESPONSE. Tasks that must be performed each sample are placed outside the state machine. Tasks that are specific to a program status are placed in the corresponding state. An example of such a status specific part is the analysis of the blood pressure signal. It makes no sense to pass samples to the analysis procedure during the flush period (or response period) because the algorithm would then detect an artefact and restart (try to re-synchronise). It is better to stop passing the samples to the analysis procedure and then force the analysis procedure in the synchronise state after the response is over. In this way we save the computing time needed for the analysis process during the flush and the response. The analysis task will thus be placed in the MONITOR state.

The other tasks are:

- get next sample, correct sample, draw samples (always)
- analyze samples, store indices of 75% Upstroke level crossings (state MONITOR)
- call the parameter estimation routine (state RESPONSE)

7.4 On-line operation

With On-line operation the program part is meant that performs the real-time correction of the distorted blood pressure signal. In this mode we need to guarantee that the sampling, drawing, analyzing and correction processes operate real-time. Whether the parameter estimation process can be done real-time or not, depends on the type and the number of parameter estimation methods that are selected. In the development phase, all algorithms generate file output to support easy evaluation and improvement of the implementations. Because file-IO is very time-consuming the parameter estimation part will not be included in the list of real-time tasks.

7.4.1 A/D conversion and storage

The A/D and D/A conversions are performed using the LabMaster board (any type of A/D converter board for a personal computer can be used). With the use of a timer the sample frequency of the board can be set (see manual). If a conversion is completed, the board will generate an interrupt. The user is supposed to write an interrupt-service routine to service the interrupt and get the sample. The interrupt-service routine collects the sample (two bytes) and places it in a circular buffer. The program that uses the samples can call the get_sample routine to get a sample from the buffer. The use of an interrupt-service routine and a buffer guarantees the real-time collection of samples. All necessary routines to use the LabMaster board are implemented in the Turbo Pascal Unit ADCAL.TPU.

If the user has enabled the Store Option the contents of the circular buffer (25 seconds of data) will be written to disk. If the write operation is issued 12 seconds after the response has ended, the flush (and response) will be located in the middle of the data-file. The blood pressure samples written to disk must be in mmHg. The raw samples from the A/D convertor can best be translated to mmHg format before they are stored in the buffer. This translation process must then be included in the interrupt-service routine. For more information about the translation function the user is referred to the source code file NEW_WAVE.PAS of Alofs' work.

7.4.2 Structure of On-line

The real-time part of the off-line procedure is implemented as an interrupt-service routine. This interrupt-service routine is the routine that services the interrupt of the A/D convertor. The real-time tasks are: drawing the sample, correcting the sample and drawing the corrected sample. In this way the blood pressure signal will be displayed real-time (essential in patient monitoring systems).

The non time-critical part of the on-line procedure is implemented as a state machine within a loop (as in the off-line procedure). The loop is executed for each sample, as long as the user wants the process to continue. The states used in the state-machine are the same as in the off-line procedure, except for the extra state PLACE. The task performed in this state is the positioning of the flush in the stable part of the blood pressure signal (placement method). The position of the flush was calculated relative to the position of the 75% Upstroke level crossing (via TRIGGER of the blood pressure analysis procedure). The positioning will only work correctly if the analysis algorithm works real-time too, but we did not place the analysis process in the real-time part (interrupt service routine). The analysis algorithm was not included in the interrupt-service routine because the blood pressure signal need not be analyzed during the flush and the response (see off-line part). If the computer is fast enough, the non time-critical part will keep up with the real-time part (except while estimating the parameters). After the parameter estimation is finished, the blood pressure analysis (which is in the synchronisation state S at this time) will catch up with the real-time part quickly. The user will not even notice the delay, because the moving average of the blood pressure data is presented. A possible implementation is presented in Textblock 2 and Textblock 3.

tasks for real-time part (the interrupt service routine)

- get sample from D/A board
- convert sample to mmHg
- store sample in the circular buffer
- draw sample
- correct sample
- draw corrected sample

Textblock 3 On-line mode: real-time part.

A problem with this implementation is that the keyboard commands of the user are blocked during the parameter estimation procedure. In all other situations (when the non time-critical part keeps up with the real-time part) the keyboard will be serviced each sample. In our application this is acceptable but in a situation where this is unacceptable the keyboard input must be implemented as an interrupt-service routine.

Another problem is the menu options that require input from the user. The input routine in a real-time program cannot wait for characters (in a loop) until the user has pressed enter. In the on-line menu there are several options that require user input (e.g. Set Parameters). To support these functions, an input routine must be made, that collects one character per sample period. Another possibility is the interrupt driven interrupt input routine described in the previous problem.

tasks for the non time-critical part

repeat

- get next sample from circular buffer

case OnlineStatus of

MONITORING:
- analyze sample
- analyze corrected sample
- store indices of 75% Upstroke level crossings using the TRIGGER from the blood pressure analysis procedure (for subtraction method)
- if time to flush and 75% Upstroke crossing passed then set placement_counter and set OnlineStatus to PLACE

PLACE:
- decrease placement_counter
- if placement_counter equals zero then open flush valve, store sample index (o), set flush_counter and set OnlineStatus to FLUSHING

FLUSHING:
- decrease flush_counter
- if flush_counter equals zero then close flush valve, store sample index (c), set response_counter and set OnlineStatus to RESPONSE

RESPONSE:
- decrease response_counter
- if response_counter equals zero then determine the reference 75% level crossing for the subtraction method, call the parameter estimation routine, set the OnlineStatus to MONITORING and set the AnalysisProcedure Status to S (Synchronisation state 1).

end case
- check if the user pressed a valid key, if so service it, if not continue

until stop requested by user

Textblock 2 On-line mode: non time-critical part.

In the implementation the test 'time to flush' was used. Some of the criteria that can be used in the decision if it is 'time to flush' are:

- Flush at regular intervals.

  What a reasonable interval is must be tested in a clinical setting. A time from two to five minutes seems acceptable.

- Flush at user request.

  The user can request a flush via the on-line menu. Reasons to request a flush can be: the CMS setup is changed, a blood sample is taken from the patient, the patient is medicated or simply when an air bubble is located visually.

- **Flush if the parameters differ too much from preset values.**
  The preset values for the used CMS could be obtained from a library or calculated from the components installed in the CMS. In both cases the user must enter the components the CMS consists of. The use of these reference values was also discussed in paragraph 3.6.2.

- **Flush if there are signs that air bubbles or blood clots are present in the CMS.**
  The effects of air bubbles and blood clots are a decrease of natural frequency and an increase of damping. The effect on the blood pressure signal will be that of an extra low-pass filter (weakening of high-frequency components). Because the blood pressure parameters are presented as moving average values, they will not be affected much by the extra low-pass filter. The effect can be detected when monitoring the blood pressure derivative dP/dt.

7.5 Software structure

To keep the structure surveyable, all related program tasks (implemented as procedures and functions) are grouped in units. The procedures and functions in these units can be seen as building blocks for the program. A further advantage of this unit structure is that individual blocks can be changed or replaced without altering the complete software structure. The units of our program (MONITOR) will now be discussed. Routines that are not described in this paragraph, are implemented in the main program file MONITOR.PAS.

Unit SCREEN

This unit contains the graphical procedures of the blood-pressure monitor program. These functions can be used to create a graphical interface for the program MONITOR. The basis for the graphical screen is several windows in which the program data is presented. The graphical interface for MONITOR has two types of windows, the signal windows and the data windows. Signal windows are used to plot the blood pressure signals. The windows are defined by their four corner coordinates. The windows sizes are relative to the graphical screen driver used (Turbo Pascal BGI). The positions of the windows, relative to the complete screen, can be set in the procedure init.

For the user interaction a statusbar is introduced at the bottom of the screen. In this statusbar messages and errors can be displayed. Also, a function is provided to get user input via this statusbar. This input procedure is not yet suited for real-time programs. The complete graphical display is drawn by the procedure builddisplay. This procedure automatically switches to the graphical display mode. It also calls an initialisation procedure, which contains all window settings. Short descriptions of the global procedures of the unit SCREEN are presented in appendix D1. For detailed information about the implementation the user is referred to the comments in the source code of SCREEN.PAS.

Unit PAREST

This unit contains the parameter estimation routines. The parameter estimation process is controlled via the procedure ParameterEstimation. The inputs for this procedure are the sample indices of the reference and last 75% Upstroke level crossings together with the index of the sample the flush valve is opened. These three indices (positions in the buffer that contains the blood pressure samples) provide enough information for the parameter estimation procedures.

The parameter estimation routines for underdamped systems are implemented in the procedure LogDec. Via the parameter list of LogDec the logarithmic decrement method approach (Constant, Linear, Quadratic or All) can be selected. Also the Subtraction- and the Dynamic Parameter Estimation method can be enabled or disabled. The implementation for the LogDec procedure is presented in Textblock 4.

use the subtraction method (if selected)
determine the flush response and the necessary derivatives of the flush response

according to the selected approaches the following tasks are done:

- search the extrema of the decaying oscillation in the response
- check if the extrema are part of a decaying signal
- check if the extrema are part of an oscillation (stable period)
- if there are still valid extrema left after these tests we calculate the parameters β and f_n according the formulas of the Logarithmic Decrement method
- if the parameters could be determined and the Dynamic Parameter Estimation is enabled the parameters β and f_n are used as an initial guess for the Dynamic Parameter Estimation method

finally all determined parameters are returned

Textblock 4 Implementation for the LogDec procedure.

Short descriptions of the global procedures of the unit PAREST are presented in appendix D2. For detailed information the user is referred to the comments in the source code of PAREST.PAS.

Unit CORRECT

This unit contains two correction routines, which are implemented as modular filters are implemented. The first is the filter discussed in paragraph 4.4.3.1 (time domain implementation of the inverse filter), the second is the NOS correction filter discussed in paragraph 4.4.4.
Short descriptions of the global functions of the unit CORRECT are presented in appendix D3. For detailed information about the implementation the user is referred to the comments in the source code of CORRECT.PAS.

Unit FILTER

This unit contains all routines dealing with digital filters. Routines are provided for FIR filters and IIR filters. In the case of FIR filters a special implementation is provided for symmetric (even and odd) filters. This special implementation reduces the number of multiplications with a factor 2. To clear the history (old sample values) of a filter, the routine ClearHistory can be used. Clearing the history must be done when a filter is re-used for a new signal. This is necessary to avoid sharp discontinuities, which cause the filter not to settle as fast as it should. In this unit also a bilinear transformation procedure is provided (zero-pole format). To convert the zero-pole format filter description to the transfer function format (and thus the actual filter coefficients) a zero-pole to transfer function procedure is also provided.

Besides these basic tools this unit also contains a procedure that can create the inverse filter with help of the bilinear transformation and the zero-pole to transfer function procedure (Textblock 5). Because complex arithmetic is needed for the bilinear transformation, also procedures are provided for complex addition, subtraction, multiplication and division. Also, a function for the calculation of the complex radius is included.

- calculate the zeros, poles and constant for the CMS inverse in the time-continuous domain
- add the zeros, poles and constant for the cascade n-th order (standard) low-pass filter in the time-continuous domain
- use the bilinear transformation to calculate the zeros, poles and constant for the total correction filter in the discrete-time domain
- use the ZP2TF routine to convert the total correction filter from the pole-zero description to the transfer function description
- store the resulting filter coefficients in the variable InvFilter (IIR)

Textblock 5 Implementation for the CreateInvFilter procedure.

New filters are defined like this:

1) Define a global variable for the filter.
   possible types: FIRFilterType, SYMFIRFilterType, IIRFilterType
2) Create local constants for the filter coefficients and filter order
   names: use filtername + Coef. respectively filtername + Order.
3) Add an initialisation part for the filter variable in the Init procedure.

Existing filters are changed like this:

Alter the local constants for the filter coefficients and the filter order.

Existing filters can also be changed dynamically (during run-time) like this:

Alter the characteristics of the global filter variables. This change will only be active for as long as the program runs (or the filter is redefined again).

Short descriptions of the global functions of the unit FILTER are presented in appendix D4. For detailed information about the implementation the user is referred to the comments in the source code of FILTER.PAS.

Unit BEAT

This unit contains the blood pressure analysis routine for the blood pressure waveforms (measured and corrected). The Algorithm implemented in this unit was described in chapter 6. With this implementation several blood pressure waveforms can be analyzed simultaneously on a sample to sample basis. The analysis process is implemented as a state machine, so it is easy to understand and change or expand. All values are in 0.25 mmHg. The timeout for each phase is set at two seconds. Also, a timeout is included for the validity of the global blood pressure data. This is included because else changes in the data could cause the relative checks (at 75% upstroke level crossings) to fail each time.

A Global variable TRIGGER is declared in this unit, that can be used to synchronise a process with the 75% Upstroke level crossing of the measured blood pressure. The variable TRIGGER (a boolean) is TRUE for the first sample that reaches the 75% upstroke and false for all other values (transition from state 3 to state 4 in the algorithm). This trigger can also be used to trigger the output of blood pressure data to the screen.

The main analysis routine (procedure analysis) is controlled via the procedures Extract...Data. For each blood pressure signal that must be analyzed, an Extract...Data procedure must be defined. In this way the output of the blood pressure data and the marks at the 75% Upstroke level crossings is located outside the analysis procedure. Feedback from the analysis procedure to the user however, must use the functions from the screen unit. The analysis procedure can easily be used for other programs when all this feedback is removed (then it is completely independent of other units).

Short descriptions of the global routines of the unit BEAT are presented in appendix D5. For detailed information about the implementation the user is referred to the comments in the source code of BEAT.PAS.

Unit SELDIR

This unit contains the file SELDIR.PAS that was copied from the stack BBS (ftp address terra.stack). The unit contains the function DIRSELEC, which can be used to select a file from a list of all files in a specified directory.

A Short description of the global routine of the unit SELDIR is presented in appendix D6. For detailed information about the implementation the user is referred to the comments in the source code of SELDIR.PAS.

Units SPE, SLE, IOM

These units are part of the Turbo Pascal Numerical Library. This Library contains procedures for all kinds of mathematical problems. Further information about the Numerical Library can be found in the Numerical Library manual. This manual can be obtained at the Computer Centre of the Eindhoven University of Technology. The software can also be obtained via the copy service facility on the Computer Centre Network.

From the unit SPE the function SPESGN is used, which returns the sign of a number.
From the unit SLE the procedure SLEGLS is used, which implements the minimum square error optimisation procedure.
From the unit IOM the procedure IOMWRV is used, which writes a matrix to file.

8 Test Results

In this chapter the tests of the algorithms will be discussed. Because the real-time program part is not finished yet, all algorithms are tested with pre-recorded blood pressure signals from a patient simulator. Because these signals are extremely periodic, they are very useful for the tests. All problems detected during these tests will probably be due to errors in the implementation and not due to uncontrollable effects in the blood pressure signal.

After these basic problems are solved and the real-time part is implemented, tests with real (human) blood pressures can be done. Problems that are detected in these test, probably arise from blood pressure signals the algorithms cannot handle. With these tests the algorithms themselves can be improved.

The implementations of the algorithms are tested using MATLAB. All intermediate program results are written to disk. To analyze these intermediate results MATLAB script files have been developed, that present the results in a numerical and graphical manner. In this way the results can be interpreted easily and changes can be made quickly.

8.1 Parameter Estimation Algorithms

The pre-recorded blood pressure signal used is from the file ea092244.dat. This is a signal from a blood pressure simulator (not a 'real' human pressure signal). The data-file was analyzed with the program MONITOR. The settings for the logarithmic decrement method was A(11). Runs were made with and without the subtraction method and with and without the Dynamic Parameter Estimation method.

In Figure 44 the results of the logarithmic decrement method without subtraction and without DPE are presented. The extrema that passed the three test (extreme with absolute minimum period size test, decay test and periodicity test) are marked with a vertical dotted line (sample position) and a small horizontal solid line (BP value).

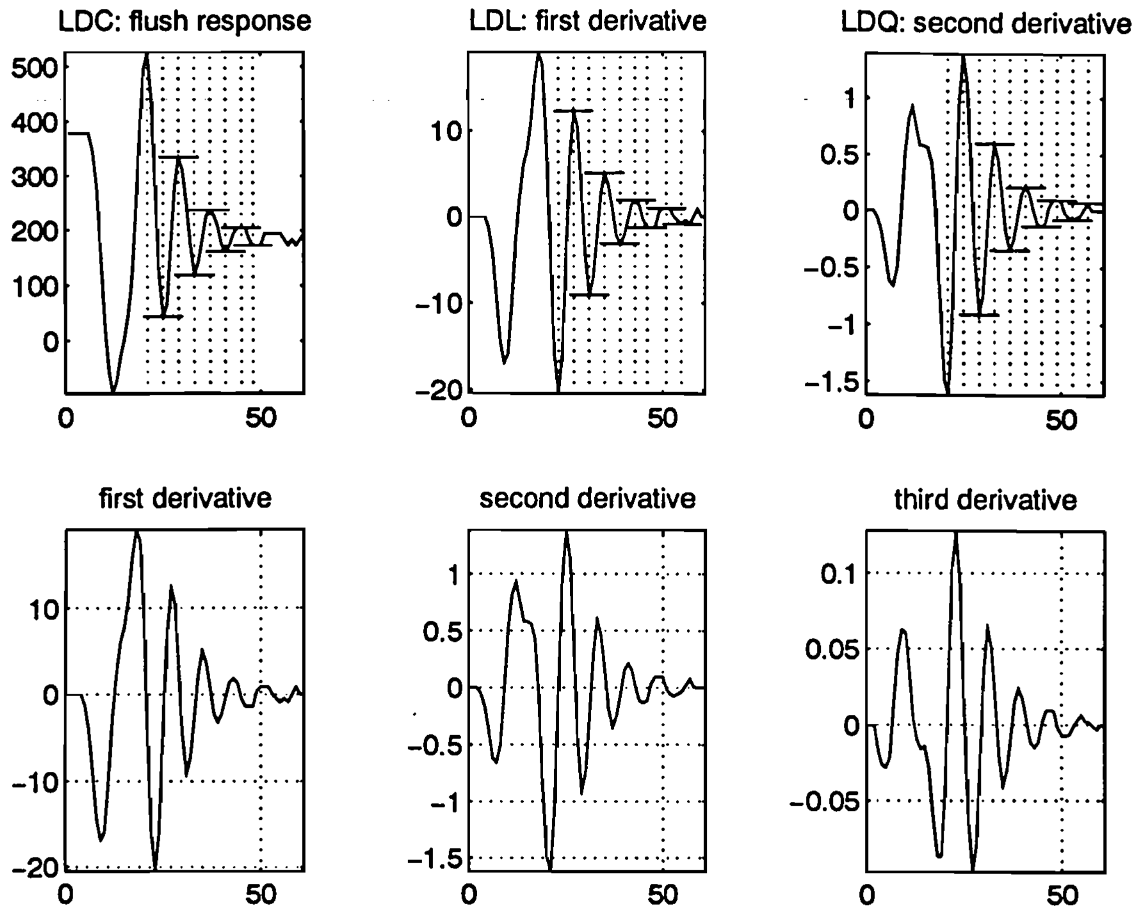


Figure 44 Results of the logarithmic decrement method for all three approaches.

From the arrays with the valid extrema, the parameters β and f_n were calculated for each approach. If more sets of parameters can be calculated (more than three valid extrema), an average set (β, f_n) is calculated. In this average a set (β, f_n) is only included, if the parameters satisfy the following demands: $0.1 \leq \beta < 1$ and $7.5 \leq f_n \leq 50$. The extracted sets of parameters (valid and invalid) are presented in Table 2. The average values are: $\beta = 0.17$ and $f_n = 25.36$ (Constant); $\beta = 0.14$ and $f_n = 25.86$ (Linear); $\beta = 0.14$ and $f_n = 25.27$ (Quadratic).

To verify the extracted parameters, a second order model with the average parameters is compared to the measured signal. A plot of the model signal (solid line) and the measured signal (dotted line) can be seen in Figure 45. The amplitude of the pressure step for the model signal is calculated as described in paragraph 3.5.3 in Figure 13. With the constant approach the model shows an offset from the measured signal, because the blood pressure influence is not constant for the test signal. The shape of the two curves, however, show a strong resemblance. From Figure 45 it can be concluded that the linear approach provides the best fit (estimate) in this example.

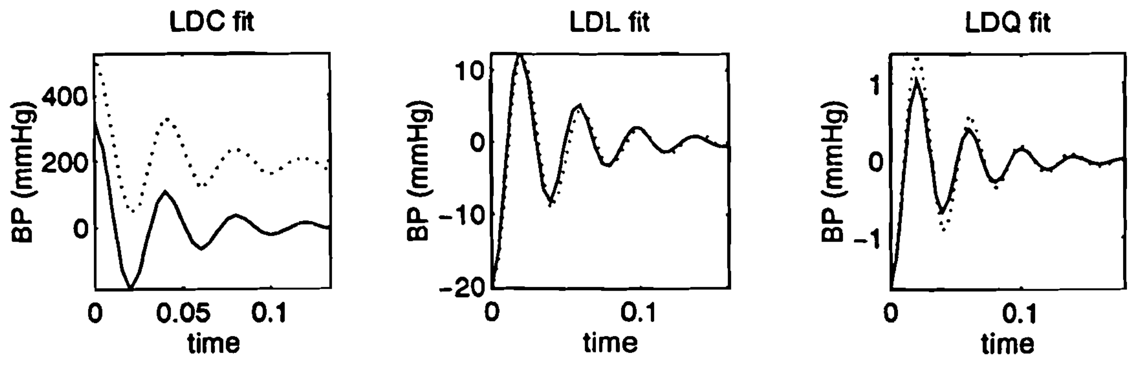


Figure 45 *Model (with the average of the extracted parameters) of the flush signal (solid line) compared to the measured flush signal (dotted line).*

Table 2 Results logarithmic decrement method (without subtraction, without DPE)

Constant approach			Linear approach			Quadratic approach		
β	f_0	valid	β	f_0	valid	β	f_0	valid
0.1603	25.33	yes	0.1343	25.23	yes	0.0858	25.09	no
0.0948	25.11	no	0.1314	25.22	yes	0.1331	25.22	yes
0.1876	25.45	yes	0.1711	25.37	yes	0.1466	25.27	yes
0.1428	25.26	yes	0.1547	25.30	yes	0.1650	25.34	yes
0.1744	25.38	yes	0.1428	28.87	yes	0.1501	25.29	yes
0.0936	28.69	no	0.1098	25.15	yes	0.1296	25.21	yes
			0.0738	22.28	no	0.0945	25.11	no
						0.0391	25.02	no

In Figure 46 the results of the logarithmic decrement method with subtraction and without DPE are presented. The extrema are marked in the same way as in the case without the subtraction method. The results are verified in the same way as without the subtraction method (Figure 47).

From the results of the logarithmic decrement method with subtraction and without DPE also an average set (β, f_0) is calculated. The average values are: β = 0.14 and f_0 = 24.69 (Constant); β = 0.14 and f_0 = 25.25 (Linear); β = 0.14 and f_0 = 25.23 (Quadratic).

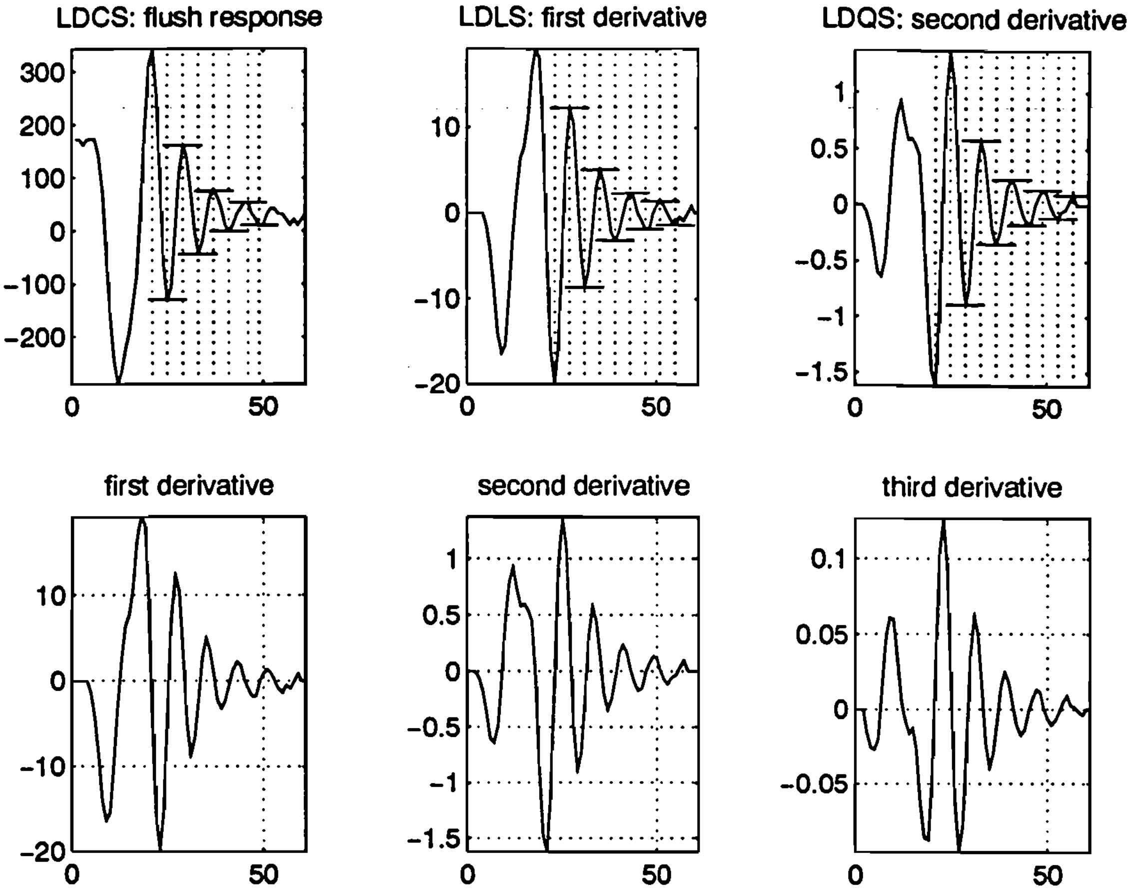


Figure 46 *Results of the logarithmic decrement method for all three approaches with use of the subtraction method.*

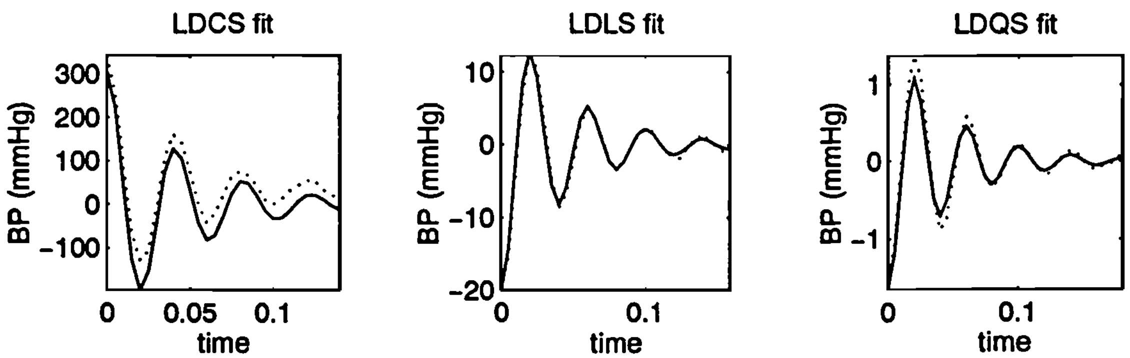


Figure 47 *Model (with the average of the extracted parameters) of the flush signal (solid line) compared to the measured flush signal (dotted line).*

The results (averages) from the logarithmic decrement method have also been used as input for the Dynamic Parameter Estimation method. The results from the DPE without the subtraction method are presented in Figure 48. The solid line is the signal model with the parameters from the DPE and the dotted line is the measured signal. The DPE method does not work for the constant approach, because the blood pressure influence is not incorporated in the model the DPE method uses.

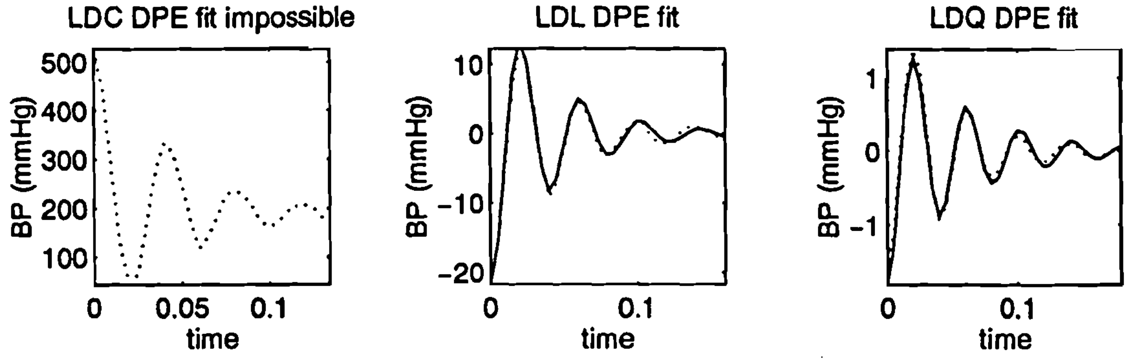


Figure 48 *Results of the DPE method without the use of the subtraction method.*

The results of the DPE method with the subtraction method are presented in Figure 49. The constant approach does work now, because the blood pressure influence is diminished by the subtraction method. The subtracted signal still shows a small trend, but this could be handled by the DPE method. Trends like this can occur because the subtraction algorithm never eliminates the blood pressure influence completely. With real 'human' blood pressure signals trends are also caused by the respiratory effects and changes in blood pressure period.

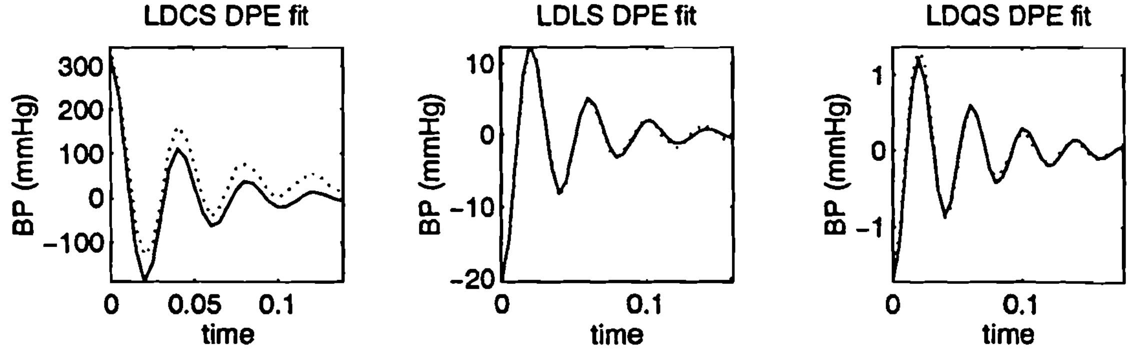


Figure 49 *Results of the DPE method with the use of the subtraction method.*

The results of the DPE methods are presented in Table 3. As can be seen in this table, all DPE estimates for f_n are equal (within 0.5%). With the DPE estimations for β (mode L and Q) the use of the subtraction method makes almost no difference. The DPE estimation for β in the Constant approach (with subtraction) is higher than the estimation for the Linear or Quadratic approach, because of the offset that was still present in the signal after subtraction.

Table 3 Results of the Dynamic Parameter Estimation method.

Without subtraction method		With subtraction method		LD Approach
β	f_0	β	f_0	
---	---	0.1684	24.88	Constant
0.1513	24.81	0.1450	24.85	Linear
0.1138	24.82	0.1111	24.86	Quadratic

The final set (β, f_0) produced by the parameter estimation process will be an average value of the sets found in all used methods. A set that differs significantly from the other sets, is not used in the averaging. If library parameter values were available, the sets could also be validated against the corresponding library set. In this case all values for β are within 7% of the mean β and all values for f_0 are within 1% of the mean f_0. The final set is thus formed from averaging all sets. The final set is: β = 0.14 ; f_0 = 24.98.

8.2 Correction Algorithms

We will start with a simulation of the whole process. The data-file used is the same as in paragraph 8.1. The process of distortion and correction is explained with the help of three signals. These signal are a flush signal, a blood pressure signal and a blood pressure signal with flush.

The undistorted signals can be seen in Figure 50.

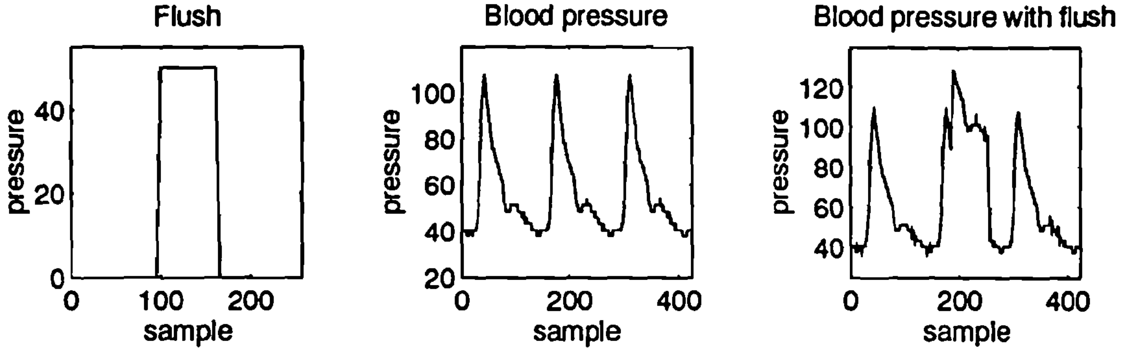


Figure 50 Signals used in the demonstration of the complete distortion and correction process.

To distort these signals we use a CMS model with $\beta = 0.2$ and $f_n = 10$ Hz. The distorted signals can be seen in Figure 51. The original signals are drawn with a dotted line, the distorted with a solid line.

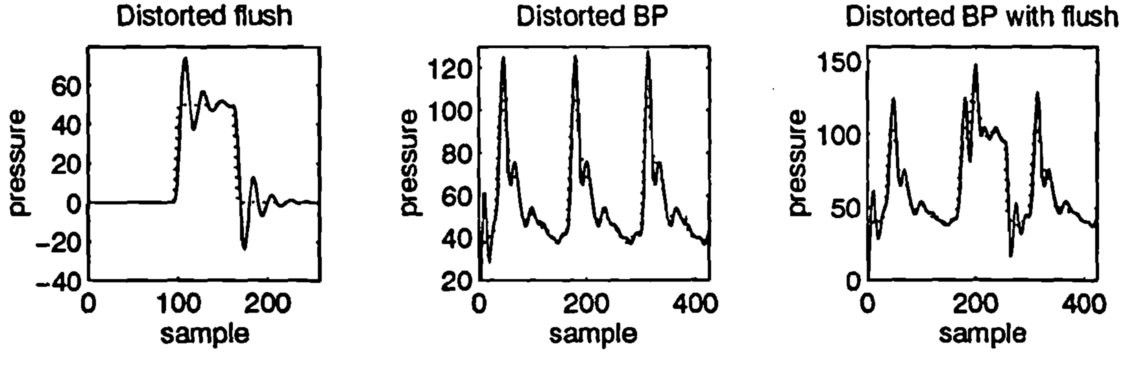


Figure 51 Distorted signals.

The signals will then be corrected with two types of correction filters. The first correction filter consists of the CMS inverse and the standard low-pass filter. The second consists of the CMS inverse and a Chebyshev type 1 low-pass filter. Both filters have a cutoff frequency of 20 Hz and are of the third order. The Chebyshev filter has an allowable ripple of 0.01 dB.

Plots of the magnitude and phase of the two correction filters can be seen in Figure 52.

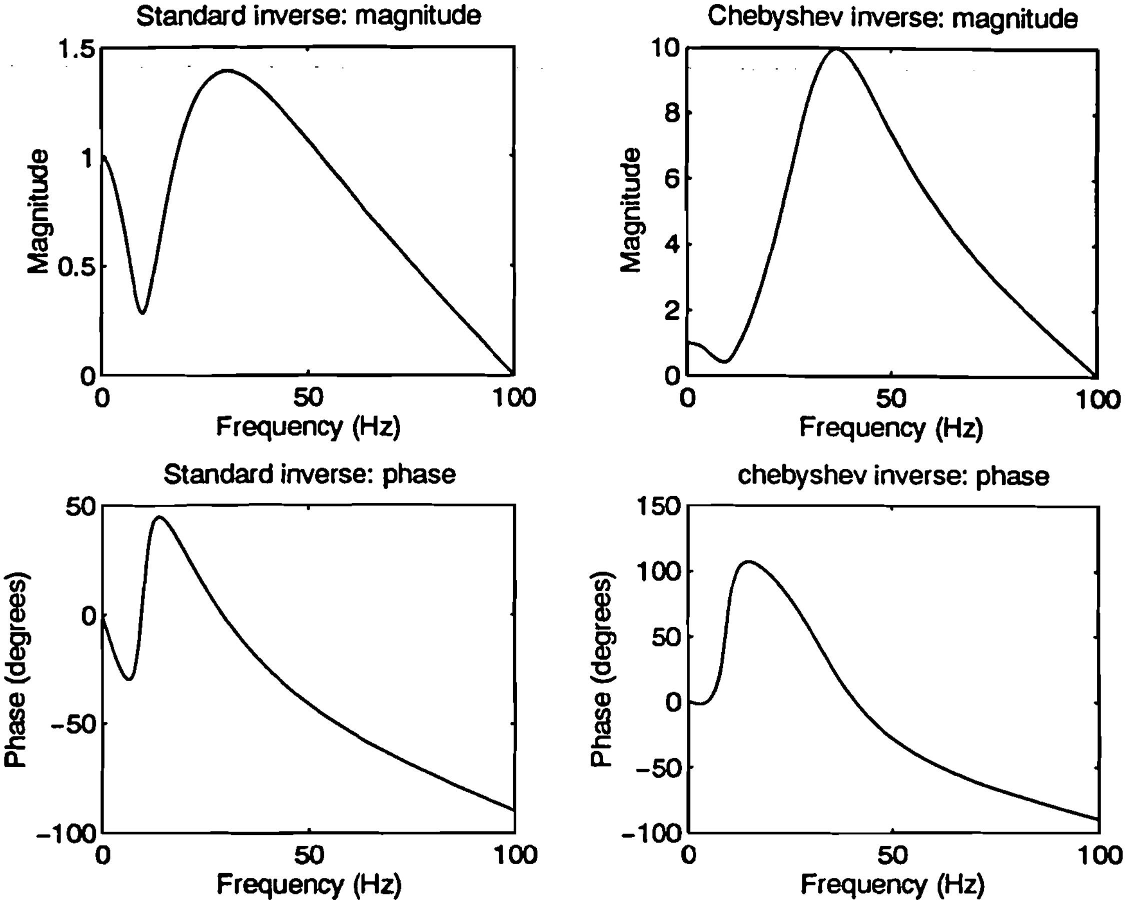


Figure 52 Correction filters.

The corrected signal with the standard low-pass filter can be seen in Figure 53. In this figure the dotted line is the undistorted signal and the solid line is the corrected signal. A zoom view is presented in Figure 54.

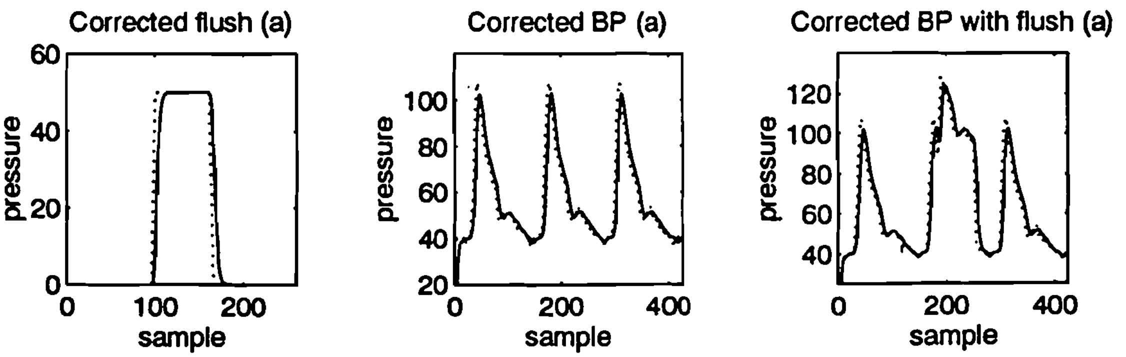


Figure 53 Corrected signals with standard low-pass filter.

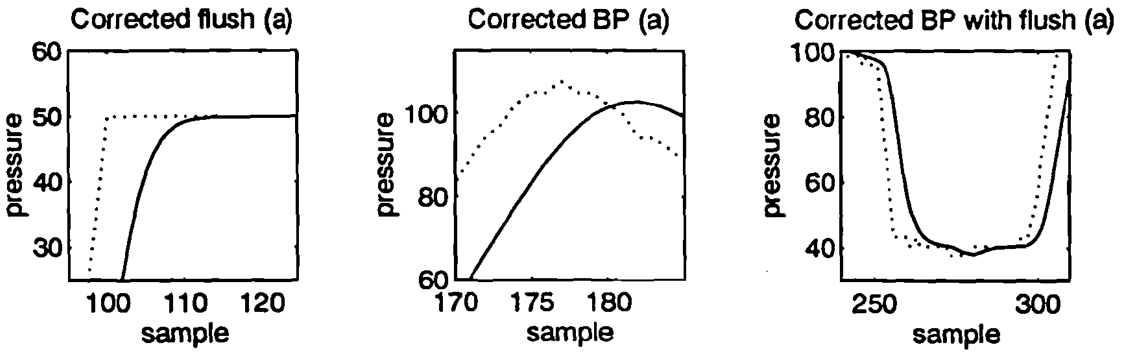


Figure 54 *Corrected signals with standard low-pass filter (zoom).*

The corrected signal with the Chebyshev low-pass filter can be seen in Figure 55. In this figure the dotted line is the undistorted signal and the solid line is the corrected signal. A zoom view is presented in Figure 56.

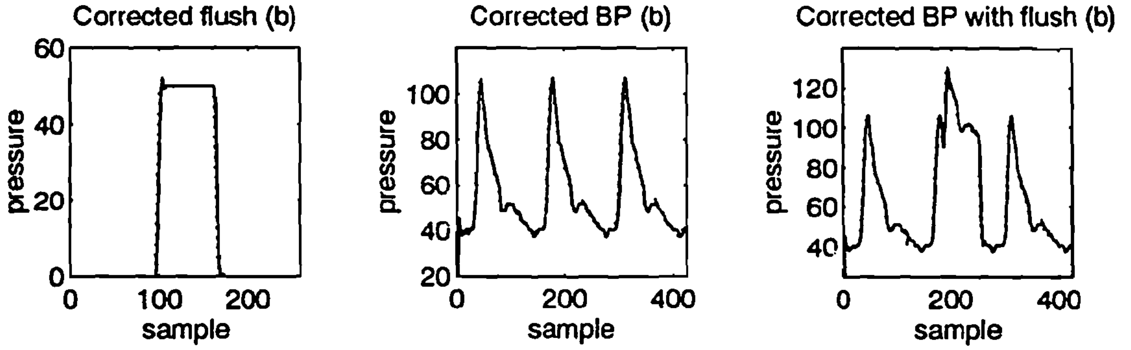


Figure 55 *Corrected signals with type 1 Chebyshev low-pass filter.*

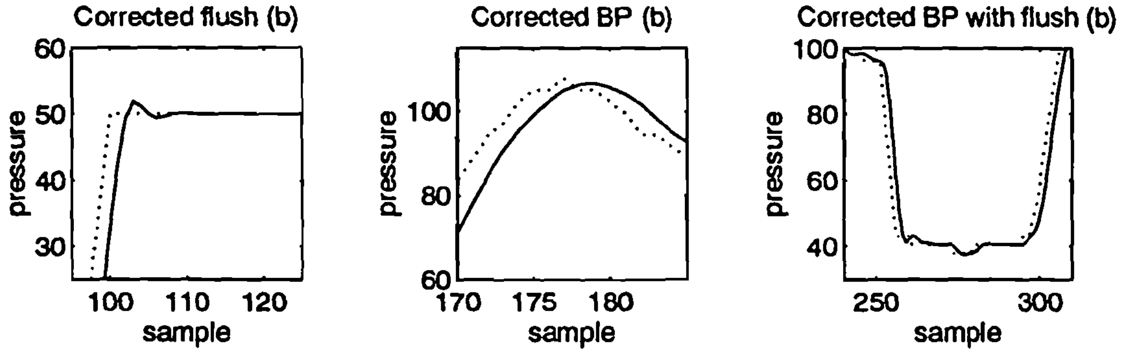


Figure 56 *Corrected signals with type 1 Chebyshev low-pass filter (zoom).*

In a practical situation the effects of noise must be considered when choosing the low-pass filter. In Figure 57 and Figure 58 the effects of a noise explosion on the BP signal and a flush signal are shown. In this situation the correction filter was made with a type 1 Chebyshev filter of order 3 with a cutoff frequency of 50 Hz.

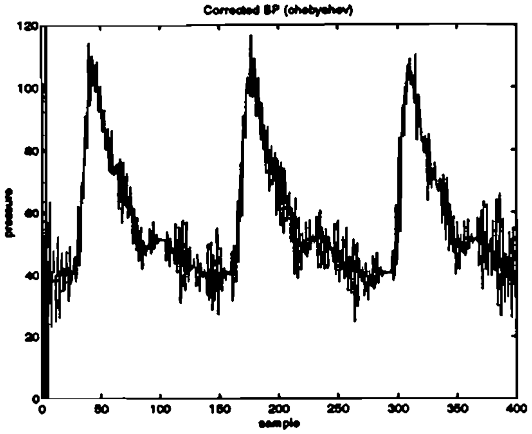


Figure 57 *Corrected BP signal with 'noise explosion'.*

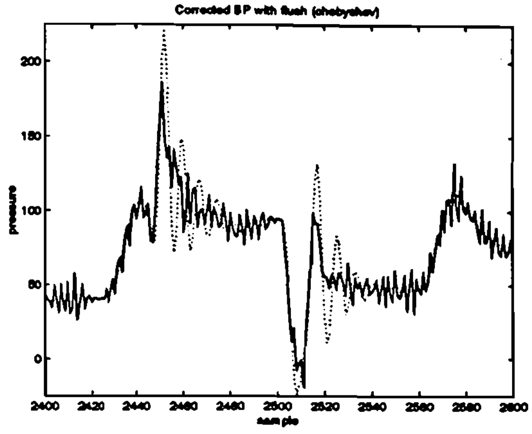


Figure 58 *Corrected flush signal with 'noise explosion'.*

A correction with adequately chosen low-pass filter can be seen in Figure 59 and Figure 60. In this case a type 1 Chebyshev low-pass filter with cutoff frequency 30 and order 4 was used.

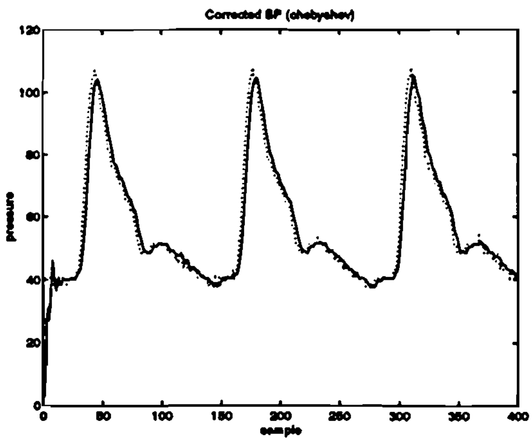


Figure 59 *Adequately corrected BP signal.*

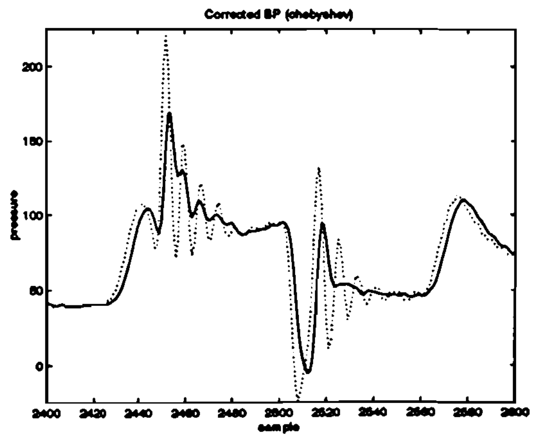


Figure 60 *Adequately corrected flush signal.*

9 Conclusions & Recommendations

Conclusions

Very recently it was shown that the dynamic response of catheter-manometer systems, used in the clinical setting, is still very poor [de Haas, 1994]. After several decades of research, still no generally accepted solution is found. From all this research it can be concluded that a solution must work automatically in order to be accepted in the clinical setting.

The fast flush test is the only reliable parameter estimation method that can be done with a standard component of a catheter-manometer system (flush device). This flush test can be done automatically by opening the valve with an electromagnet, which is controlled by a computer. The CMS parameters can be determined from the flush response with a computer program.

For a reliable correction it is important that the parameters are determined accurately. By using a combination of at least two different parameter estimation methods we can compare the extracted parameters and decide whether the algorithms work properly. If the extracted parameters are also verified against library (model) values or a second test, we can see if the parameters are representative for the used CMS.

The most frequently used parameter extraction method for underdamped systems is the logarithmic decrement method. This method performs well, but it can be very unreliable if the sample frequency is chosen too low. This method can be combined with a Dynamic Parameter Estimation (DPE) method. The DPE method also works well for low sample frequencies and it is more accurate. The DPE however is more sensitive to trends (like the respiratory component in the blood pressure signal) and it takes more time to compute.

From van Langens research it can be concluded that, *in situ*, overdamped systems can occur. To determine the parameters in this situation the algorithms by Karnopp or Warburg can be used. These methods can also be combined with a DPE method (based on the model for an overdamped system).

The goal of the correction process is to obtain a 'flat' frequency response for the frequency range in which the blood pressure signal contains information. The correction filter is a digital, adaptive filter that contains the inverse transfer function of the CMS and a low-pass filter. The low-pass filter part is added to make the total correction filter realisable and to prevent an amplification for the high frequencies (noise). If the low-pass filter is badly chosen the correction process will fail.

The correction filter can be implemented as a single- or a modular filter. The single filter implementation is easier to use, although it is more difficult to construct (via the bilinear transformation). If many extra extension tubes and stop-cocks are used in a CMS the second order model will no longer be valid. In this case the correction can be done with the Natural Observer System (NOS). The NOS is a modular correction filter with a built-in parameter estimation procedure.

All related program tasks of the program MONITOR are grouped in units. The procedures and functions in these units can be seen as building blocks for the program. Because of the unit-structure, the program is easy to survey and changes can be made quickly.

The real-time tasks need not all be implemented in an interrupt service routine. If the total time needed for all tasks is smaller than a sample period, the 'normal' program part will keep up with the real-time part. The parameter estimation process, however, takes more time than one sample period. Real-time tasks which must continue during the parameter estimation process, must be implemented in an interrupt service routine.

Finally it can be concluded that real-time blood pressure correction is possible. To get an acceptable system further tests, especially with human blood pressure signals, are necessary.

Recommendations

As mentioned before, the program MONITOR is not finished yet. The real-time part (with the A/D conversion) must be implemented before testing can start. Before continuing the research, it is advised to look at the software implemented sofar and to get acquainted with MATLAB.

The following tasks remain:

- Implement the real-time part. An implementation advice can be found in this report. The routines for the A/D conversion can be found in the files ADCAL.PAS and RTI815.PAS.

- Test the system and use MATLAB to verify all results. Find the errors in the implementation and correct them.
- If overdamped systems occur the method of Warburg or Karnopp can be implemented.
- If situations are found, in which the second order model does not work, the NOS method with built-in parameter estimation can be implemented.
- Complete the MONITOR program and prepare the system for clinical tests.

For the clinical test the following should be considered:

- To indicate how well a method works, the corrected blood pressure signal should be compared with an undistorted blood pressure signal. This undistorted signal can be obtained with a tip-catheter.
- During the clinical tests, extra knowledge can be collected about blood pressure periodicity, frequently occurring artifacts, respiratory effects and other effects. This knowledge can be used to improve the algorithms.
- Try to make the program 100% safe. This means that the algorithm will perform well in all situations. If the program does not know what to do, it must say so.
- The final program must be able to choose the appropriate parameter extraction algorithms itself. The program must be as easy to control as a blood pressure monitor.

References

Alofs, J.M.G.
Kwaliteitsverbetering van Invasieve Bloeddrukmetingen.
Eindhoven University of Technology, Department of Electrical Engineering, Division of Medical Electrical Engineering, 1993. M.Sc. report (no. 7078).

Bart, A.J.M. & E. van Vollenhoven.
Nauwkeurige Bloeddrukmeting met een Catheter-Manometer Systeem.
Ziekenhuis Techniek, june 1985, p. 163-165.

Beer, R.H.M. de.
Kwaliteitsbewaking van Invasieve Bloeddrukmetingen.
Eindhoven University of Technology, Department of Electrical Engineering, Division of Medical Electrical Engineering, 1984. M.Sc. report (BEER 8405).

Billiet, E. & F. Colardyn.
Hazardous Information from Bedside Fast-Flush Device Test for Fluid-Filled Pressure Monitoring Systems.
Angiology, Vol. 43 (1992), No. 12, p. 988-995.

Blom, J.A.
The Simplexys Experiment, Real Time Expert Systems in Patient Monitoring.
Ph. D. Thesis, Eindhoven University of Technology, 1990.

Boer, R.W de.
Beat-to-Beat Blood-Pressure Fluctuations and Heart-Rate Variability in Man: Physiological Relationships, Analysis Techniques and a Simple Model.
Ph. D. Thesis, University of Amsterdam, 1985.

Boonzaier, D.A.
Resonance Artefact in Intravascular Blood-Pressure Measuring Systems: a Technique for On-Line Digital Computer Correction.
South African Journal of Science, Vol. 74 (1978), Iss. july, p. 250-255.

Brower, R.W. & W. Spaans, P.A.M. Rewiersma, G.T. Meester.
A Fully Automatic Device for Compensating for Artifacts in Conventional Catheter-Manometer Pressure Recordings.
Biomedical Engineering, No. 10 (1975), p. 305-310.

Brunner, J.M.R.
Handbook of Blood Pressure Monitoring, Second Printing.
Littleton, Massachusetts: PSG, 1978.

Frank, O.
Kritik der Elastischen Manometer.
Z. Biol., Vol. 44 (1903), p. 445-613.

Gabe, I.T.
Pressure Measurement in Experimental Physiology.
In: Cardiovascular Fluid Dynamics, vol. 1, p 11-50.
Edited by D.H. Bergel
New York: Academic Press, 1972.

Gardner, R.M. & H.R. Warner, H.F. Toronto, W.D. Gaisford.
Catheter-Flush System for Continuous Monitoring of Central Arterial Waveform.
Journal of Applied Physiology, Vol. 29 (1970), No. 6, p.911-913.

Gardner, R.M.
Direct Blood Pressure Measurement - Dynamic Requirements.
Anesthesiology, Vol. 54 (1981), No. 3, p. 227-336.

Geddes, L.A.
The Direct and Indirect Measurement of Blood Pressure.
Chicago: Year Book Publishers, 1970.

Geddes, L. A. & J.D. Bourland.
Estimation of the Damping Coefficient of Fluid-Filled, Catheter-Transducer Pressure-Measuring Systems.
Journal of Clinical Engineering, Vol. 13 (1988), No. 1, p. 59-62.

Haas, D. de. & C. Kruidenier, K. v.d. Mijl, E. Oldenborg, O. Veltman.
Dynamische Eigenschappen van Verscheidene Disposable Invasieve Drukmeetsystemen.
TG, No. 2, February 1994, p. 7-10.

Hamming, R.W.
Digital Filters, Third Edition.
Englewood Cliffs, New Jersey: Prentice hall, 1989.

Hansen, A.T.
Pressure Measurement in the Human Organism.
Acta Physiol. Scand., Vol. 19 (1949), Suppl. 68, p. 1-227

Hansen, A.T & E. Warburg.
The Theory for Elastic Liquid-Containing Membrane Manometers.
Acta Physiol. Scand., Vol. 19 (1950), Suppl. 65, p. 306-332.

Heimann, P.A. & W.B. Murray.
Construction and Use of Catheter-Manometer Systems.
Journal of Clinical Monitoring, Vol. 9 (1993), No. 1, p. 45-53.

Hipkins, S.F. & A.J. Rutten, W.B. Runciman.
Experimental Analysis of Catheter-Manometer Systems in Vitro and in Vivo.
Anesthesiology, Vol. 71 (1989), No. 6, p. 893-906.

Hori, J. & Y. Saitoh, T. Kiryu, T. Iijima.
Automatic Correction of Left-Ventricular Pressure Waveform Using the Natural Observation Method.
IEICE Transactions on Information and Systems,
Vol. E75-D (1992), No. 6, p. 909-915.

Jackson, L.B.
Digital Filters and Signal Processing, Second Edition.
Boston: Kluwer Academic Publishers, 1989.

Johnson, R.W. & T.S. Scanlon, N. Smith, J.P Mulier.
A Neural Network for Identification of Damped or Resonant Blood Pressure Waveforms.
In: Proceedings of the 13th Annual Conference on Engineering in Medicine and Biology, Orlando, Florida, USA, October 31-Nov 3 1991.
Edited by Nagel, J.H. & W.M Smith.
Piscataway: IEEE Service Centre, 1991.
Vol. 13 (1991), part 3, p. 1413-1414.

Karnopp, B.H. & F.E. Fisher.
Determination of Vibration Parameters in Moderately Damped Systems.
Journal of the Franklin Institute, Vol. 327 (1990), No. 4, p. 611-620.

Karnopp, B.H. & F.E. Fisher.
On the Vibrations of Overdamped Systems.
Journal of the Franklin Institute, Vol. 327 (1990), No. 4, p. 601-609.

Kleinman, B. & S. Powell, P. Kumar, R.M. Gardner.
The Fast Flush Test Measures the Dynamic-Response of the Entire Blood-Pressure Monitoring-System.
Anesthesiology, Vol. 77 (1992), No. 6, p. 1215-1220.

Langen, H. van & P. Brienesse, K. Kopinga, P. Wijn.
Dynamic Response of a Neonatal Catheter-Manometer System in Situ.
Journal of Clinical Monitoring, Vol. 9 (1993), No. 5, p. 335-340.

Lapointe, A.C. & F.A. Roberge.
Mechanical Damping of the Manometric System Used in the Pressure Gradient Technique.
IEEE Transactions on Biomedical Engineering, January 1974, p. 76-78.

Latimer, K.E.
The Transmission of Sound Waves in Liquid-Filled Catheter Tubes Used for Intravascular Blood-Pressure Recording.
Medical & Biological Engineering, Vol. 6 (1968), p. 29-42.

Latimer, K.E. & R.D. Latimer.
Measurement of Pressure-Wave Transmission in Liquid-Filled Tubes Used for Intravascular Blood-Pressure Recording.
Medical & Biological Engineering, Vol. 7 (1969), p. 143-168.

Marks, L.A. & K.L Short, D. Hoffmann, A. Lew.
Microprocessor-Based Robotic System for Control of Fluid Connections in the Cardiac Catheterization Laboratory.
IEEE Transactions on Biomedical Engineering, Vol. 35 (1988), No. 2, p. 161-166.

Micheletti, R.
Dynamic Parameter Estimation of a Second-Order System to a Step Function.
IEEE Transactions on Instrumentation and Measurement,
Vol. 39 (1990), No. 5, p. 790-792.

Noble, F.W. & G.O. Barnett.
An Electronic Circuit for Improving the Dynamic Response of the Conventional Cardiac Catheter System.
Med. Electron. Biol. Engng., Vol. 1 (1963), p. 537-545.

Paulsen, A.W.
Implications for Clinical Monitoring of Intra-Arterial Blood Pressure Based on the Frequency Content of Worstcase Pressure Waveforms.
Biomedical Instrumentation & Technology, Vol. 27 (1993), No. 3, p. 217-234.

Plasman, J.L.C. & C.M.M. Timmers.
Direct Measurement of Blood Pressure by Liquid-Filled Catheter Manometer Systems.
Eindhoven: Eindhoven University of Technology, Department of Electrical Engineering, 1981. EUT rapport 81-E-121.

Plasman, J.L.C.
An Introduction to the Analysis of the Arterial and Venous Waves.
Eindhoven University of Technology, Department of Electrical Engineering, Division of Medical Electrical Engineering, 1981. M.Sc. report (4323).

Prentza, A. & Wesseling, K.H.
Catheter-Manometer System Damped Blood Pressures Detected by Neural Nets.
Internal report, Technical university of Eindhoven, Department of Electrical Engineering, Division of Medical Electrical Engineering, 1994.

Rothe, C.F. & K.C. Kim.
Measuring Systolic Arterial Blood Pressure: Possible Errors from Extension Tubes or Disposable Transducer Domes.
Critical Care Medicine, Vol. 8 (1980), No. 11, p. 683-689.

Saitoh, Y. & J. Hori, T. Kiryu, H. Makino, K. Tamura, T. Yoshizaki, T. Lijima.
Correction of Pressure Waveforms in a Catheter Manometer System by the Natural Observation Method.
Japanese Journal of Medical Electronics and Biological Engineering,
Vol. 26 (1988), No. 3, p.133-138, (language: japanese).

Schwid, H.A.
Semiautomatic Algorithm to Remove Resonance Artifacts from the Direct Radial Artery Pressure.
Biomedical Instrumentation & Technology, Vol. 23 (1989), No. 1, p. 40-43.

Shapiro, G.G. & L.J. Krovetz.
Damped and Undamped Frequency Responses of Underdamped Catheter Manometer Systems.
American Heart Journal, Vol. 80 (1970) No.2, p. 226-236.

Sheahan, N.F. & B. Tuohy, R. Kirkham, D. Coakley, J. Malone.
The Fast Flush Test: Evaluation in Radial Artery Catheter-Manometer Systems.
Clinical Physics and Physiological Measurement,
Vol. 12 (1991), No. 3, p. 247-252.

Takeuchi, H. & K. Toraichi, M. Kamada, F. Nagasaki, R. Mori.
Time-Varying Spline Filter and its Application to the Left Ventricular Pressure Measurement.
Medical & Biological Engineering & Computing, Vol. 26 (1988), No. 1, p. 88-91.

Taylor, B.C. & D.M. Ellis, J.M. Drew.
Quantification and Simulation of Fluid-Filled Catheter/Transducer Systems.
Medical Instrumentation, Vol. 20 (1986), No 3, p. 123-129.

Taylor, B.C.
Frequency Response Testing in Catheter-Transducer Systems.
Journal of Clinical Engineering, Vol. 15 (1990), No. 5, p. 395-406.

Taylor, B.C. & F.M. Donovan, Jr.
Hydraulic Resistance and Damping in Catheter-Transducer Systems.
IEEE Engineering in Medicine and Biology Magazine,
Vol. 11 (1992), No. 4, p. 72-78.

Vierhout, R.R.
Approximative Models for Transmission Lines and their Errors.
Electronic Engng, 31,94,95.

Vierhout, R.R. & Vendrik, A.J.H.
On Pressure Generators for Testing Catheter Manometer Systems,
Phys. Med. Biol. 10, p. 403-406.

Vierhout, R.R.
The Response of Catheter Manometer Systems Used for Direct Pressure Measurements.
Ph. D. Thesis, University of Nijmegen, 1966.

Warburg, E.
A Method of Determining the Undamped Natural Frequency and the Damping in Underdamped and Slightly Underdamped Systems of One Degree of Freedom by Means of a Square Wave Impact.
Acta Physiol. Scand., Vol. 19 (1949), p.344-349.

Wesseling, K.H. & E. van Vollenhoven.
Catheter-Manometer Systems: a Review and Some Measurement Results.
TNO-nieuws, Vol. 24 (1969), p. 603-616.

Yeomanson, C.W. & D.H. Evans.
The Frequency Response of External Transducer Blood Pressure Measurement Systems: a Theoretical and Experimental Study.
Clinical Physics and Physiological Measurements,
Vol. 4 (1983), No. 4, p. 435-449.

Appendix A List of Figures

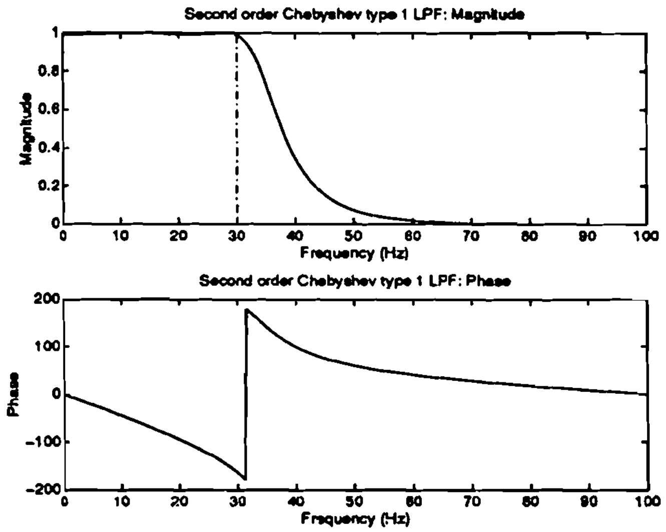
Appendix B List of MATLAB Script Files

bili.m	inverse + n-th order lpf(standard) via bilinear transformation
bili2.m	inverse + n-th order lpf(standard) via bilinear transformation (2nd version)
CMS.m	m-file description of the SIMULINK system named CMS
cmsinv.m	transfer function CMS (S,Z), INV (S,Z) and TOT (S,Z)
difsam.m	display the effects of fs on the differentiation
dpefit.m	plot fit-results of LD algorithm with DPE
dpesfit.m	plots fit-results of the logarithmic decrement with DPE and subtraction
erextr.m	maximum error in extrema extraction for sinusoidal signal
fderror.m	function that calculates the error made because of the chosen f_s in the first derivative around zero
fildes.m	filter design routine
freqresp.m	plot filter frequency response
freqsw.m	plot frequency sweep
h.m	magnitude of the CMS transfer function
hpf.m	create first order HPF with bilinear transformation
iir.m	plot file testcor.dat
impresp.m	function that calculates the impulse response
invfil.m	draw the transfer functions of CMS, INVERSE and TOTAL
jos.m	create a first order digital differentiator filter with the bilinear transformation
laplpf.m	plot standard lpf
ld.m	demo of logarithmic decrement procedure
lderror.m	calculates error in peak amplitude in LD process
ldfit.m	plot fit-results of LD method with placement method
ldres.m	plot result (signals) of LD method
ldsubfit.m	plot fit-result of LD with subtraction method
ldsubres.m	plot results (signals) of LD with subtraction
logdec.m	plot parameters used in LD method
lowpass.m	function that calculates magnitude of a LPF
michel.m	demonstration of micheletti's dynamic parameter estimation method
omegad.m	calculate ω_d for impulsresponse

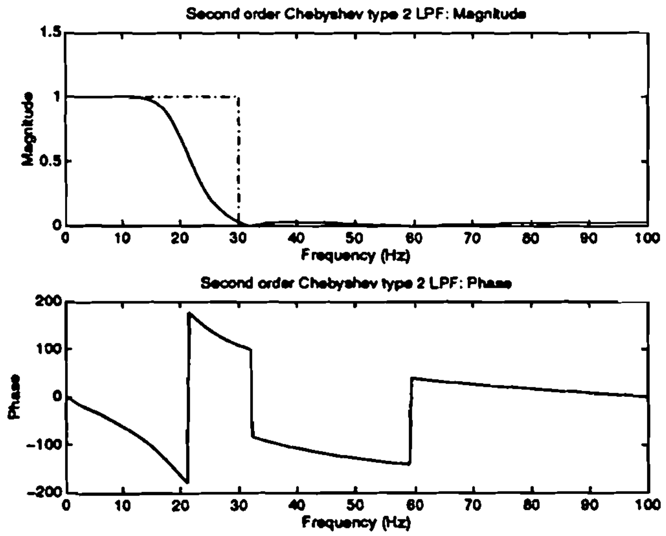
oscil.m	function that calculates second order system oscillation
parerr.m	function that returns errors in the blood pressure parameters given the error in the cms parameters
perror.m	calculate error in amplitude in LD method given f_s
pimp3.m	3d plot of impulsresponse
pimpuls.m	plot impulse-response
plotb.m	plot β graphs
plotbeat.m	plot of BP with levels and states
plotbeta.m	plot magnitude and phase of second order CMS
plotcheb.m	create a second order digital chebyshev low-pass filter
plotextr.m	plot results of LD algorithm (reads files from disk)
plothpf.m	create and plot a first order digital high-pass filter
plotlanc.m	plot magnitude and phase of lanczos differentiator
plotlpf.m	create and plot a second order digital low-pass filter
plotppe.m	function scriptfile that plots the results of a run of pparerr.m
plotrem.m	plot magnitude and phase of remez differentiator
plotrsp.m	plot blood pressure signal with respiratory effects
plotsubt.m	scriptfile that plots a picture explaining the subtraction process
pparerr.m	3-dimensional error curves (calls parerr.m function)
process.m	demonstration of the complete correction process
pvsperr.m	errors in parameters (β and f_n) versus errors in the blood pressure parameters
secord.m	function that calculates magnitude of second order transfer function
sfildes.m	design continuous-time filter
step.m	function that calculates step function
stepplot.m	plot step-function
subtract.m	plot subtraction process
test.m	plot results from test.dat
testcms.m	demonstration distortion & correction process
testfil.m	plot contents of file test.dat (filtered BP)

Appendix C Chebyshev Designs

Type 1 design: 4th order, f_c = 30 Hz, ripple = 0.1 dB



Type 2 design: 4th order, f_c = 30 Hz, ripple = 30 dB



Appendix D Contents of the Units;

D1 Global routines of Unit SCREEN

Builddisplay;

This procedure will construct the complete graphical screen including all viewports. All labels are relatively coupled to the size of the viewports. This procedure automatically switches to te graphical mode. It also calls the init procedure that contains all window settings.

ClearWindows;

This procedure clears the BP signal windows to black, draws the axes and also resets the BP signal display variables.

PlotMeasuredSample(BPValue:integer);

This procedure can be used to draw a blood pressure sample in the measured BP signal window.

PlotCorrectedSample(BPValue:integer);

This procedure can be used to draw a blood pressure sample in the corrected BP signal window.

PlotMeasuredBPData(SBP,DBP,MEAN,HR:integer);

This procedure can be used to draw the measured BP data in the corresponding window.

PlotCorrectedBPData(SBP,DBP:integer);

This procedure can be used to draw the corrected BP data in the corresponding window.

PlotParameters(B1,B2,B3,B4,B5,B6,F1,F2,F3,F4,F5,F6:real);

This procedure can be used to draw the extracted parameters in the corresponding window.
In case one of the parameters is zero '---' will be printed on that parameters' position.

PlotUsedParameters(Beta,Fn:real);

This procedure can be used to plot the Used parameters in the corresponding window.

PlotStatusBar(StatusBar:BarType;Text:String);

This procedure can be used for the user ineraction. The statusbar is located at the bottom of the screen. In this statusbar messages and errors can be displayed as well as the menu texts. To select the type of interaction the variable Statusbar of Bartype is introduced.

PlotTime;

This procedure can be used to plot the time. The position of the clock is the upper right corner of the screen in the programe name bar.

PlotMark(Mark:MarkType;Window:WindowType;Color:word);

This procedure can be used to plot a mark in one of the BP signal windows. The procedure can mark the last drawn sample with a vertical line or a small horizontal dash.

ReadFromStatusbar(Message:string;var UserInput:string);

This procedure can be used for the user ineraction (input).

SetDisplayInterval(Mode:DisplayIntervalType;Interval:real);

This procedure is used to control the scale of the time axis.

SetHgRange;

This procedure can be used to set the HGRange (y-axis) of the BP signal windows. The user input is done via the procedure ReadFromStatusBar.

SetParameters(var DAMPING,FNATURAL:real);

This procedure can be used to set the parameters β and f_0 used in the correction procedures manually. The user input is done via the procedure ReadFromStatusBar.

SetNumOfWindows(Number:integer);

This procedure can be used to change the number of BP signal windows. The measured and corrected blood pressure signal can be drawn in the same or in separate windows.

D2 Global routine of Unit PAREST

ParameterEstimation(Ref75Up,Last75Up,StartResponse:integer);

This procedure controls the parameter estimation strategy.

D3 Global routines of Unit CORRECT

CorrectSampleJos(Value:integer;beta,fn:real):real;

This function can be used to correct blood pressure samples with the method described by Jos Alofs [Alofs, 1993]

CorrectNOS(Value,beta,fn:real;order:integer):real;

This function can be used to correct blood pressure samples with the NOS method. The NOS parameters a_i for a second- or third order NOS are calculated theoretically from the CMS parameters β and f_n.

D4 Global routines of Unit FILTER

FIRFilter(Value:real;var SelectedFilter:FIRFilterType):real;

This function calculates the filtered sample value for the sample entered (N div 2) samples ago. The array with filter coeficients must start with the coeficient for the oldest sample and end with the coeficient for the most recent sample.

SYMFIRFilter(Value:real;var SelectedFilter:SYMFIRFilterType):real;

This function calculates the filtered sample value for the sample entered (N div 2) samples ago. The array with filter coeficients must start with the coeficient for the oldest sample and end with the coeficient for the most recent sample. The fact that the filter is symetric is used, and so time is saved because the number of multiplications is reduced with a factor two.

IIRFilter(Value:double;var SelectedFilter:IIRFilterType):double;

This function calculates the filtered sample value for the sample entered (N div 2) samples ago. The arrays with filter coeficients must start with the coeficient for the oldest sample and end with the coeficient for the most recent sample.

ClearFIRHistory(Filter:FIRFilterType);
ClearSYMFIRHistory(Filter:SYMFirFilterType);
ClearIIRHistory(Filter:IIRFilterType);

These procedures can be used to clear the history (old samples) of a filter. Clearing the history must be done when a filter is re-used for a new signal, else the filter will not settle as fast as it should.

CreateInvFilter(var InvFilter:IIRFilterType;beta,fn,fc:real;order:integer);

This procedure can be used to create the inverse filter (single filter type) that corrects the distorded blood pressure signal. The user has to provide a variable of the IIRFilterType and the parameters for the inverse filter (i.e. damping coeficient β, natural frequency f_n, low-pass filter cutoff frequency f_c and the low-pass filter order n). The correction filter is returned in the variable InvFilter.

D5 Global routines of Unit BEAT

ExtractMeasuredData(BPValue:integer);

This procedure can be used to pass a measured blood pressure sample to the analysis algorithm (via the variable BPValue).

ExtractCorrectedData(BPValue:integer);

This procedure can be used to pass a corrected blood pressure sample to the analysis algorithm (via the variable BPValue).

InitDataAnalysis(Window : AnalysisType);

This procedure resets all variables of the selected blood pressure signal window to the initial state.

D6 Global routine of Unit SELDIR

DIRSELECT(mask : strtype; attr : Integer) : strtype;

This function is used to select a blood pressure data file from the directory specified in mask. The variable attr is not used (use 0).