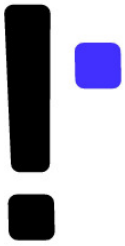


Fachhochschule
Südwestfalen
Hochschule für
Technik und Wirtschaft
University of Applied Sciences

Fachbereich Informatik und Naturwissenschaften
- Computer Vision based on Computational Intelligence –

Master's Thesis

Optimization of the Vagus Nerve Stimulation Parameters by the Means of Computational Intelligence

Author:	Aleksander Lodwich
Matriculation Number:	10002111
Address:	Friedensstr. 40 58097 Hagen Germany
E-Mail:	aleksander@lodwich.homeip.net
Supervisor:	Prof. Dr. J. Krone
Co-Supervisor:	Prof. Dipl.-Ing. Dipl.-Ing. U. Lehmann
Begin:	May 9th 2007
End:	Aug. 29th 2007

Declaration

I have written this thesis independently, solely based on the literature and tools mentioned in the chapters and the appendix. This document – in the present or a similar form – has not and will not be submitted to any other institution apart from the University of Applied Sciences Southwestfalia to receive an academical grade.
Iserlohn, 28th of August, 2007

(Aleksander Lodwich)

Table of Contents

1 Abstract

There are many kinds of epilepsy. Some of them cannot be treated neither with medicine nor can be treated with ablative or resective surgery [1-7]. Such forms of epilepsy are called refractory. A new method of treatment is tested. This method involves the electric stimulation of the Vagus Nerve with a device implanted into the patient's body, very much like the cardiac pacemaker. The efficacy of this method varies over time and is different for each patient [8][9]. It is little known about the cause of these variations but it was suggested by Zaaimi that long term Vagus Nerve Stimulation and some specific type of cardio-respiratory influence may cause neuro-protective effects [10]. In order to improve the Vagus Nerve Stimulation (VNS) the project aims at investigating statistical feature exposition for certain parameter vectors. To the author's knowledge no comparably complex nor complete studies were undertaken so far because of the challenges in data acquisition and data evaluation. Investigated vectors are used in real cases. Immediate tests on humans are not possible but rats are adequate substitute models for the operational aspect of the Vagus Nerve. Hence rats are used in order to provide biological response data upon stimulus. In theory the response will be different for each rat, but it is also being assumed that for some vectors similar response will also occur. If only enough rats were recorded then a statistically safe prediction could be made about what parameter vectors will result in the effects most aimed at by the therapist. It is the primary goal of the project to find similarities in different response cases for some of the typical vectors and eventually provide deeper insights on the stimulation. The project faces extremely large data amounts from data acquisition. Methods of Computational Intelligence (CI) are used for the conversion of data to knowledge [11]. After conventional data preprocessing evolutionarily designed artificial neural networks are used in order to evaluate available data and to help medical staff to verify their medical thesis. [12]

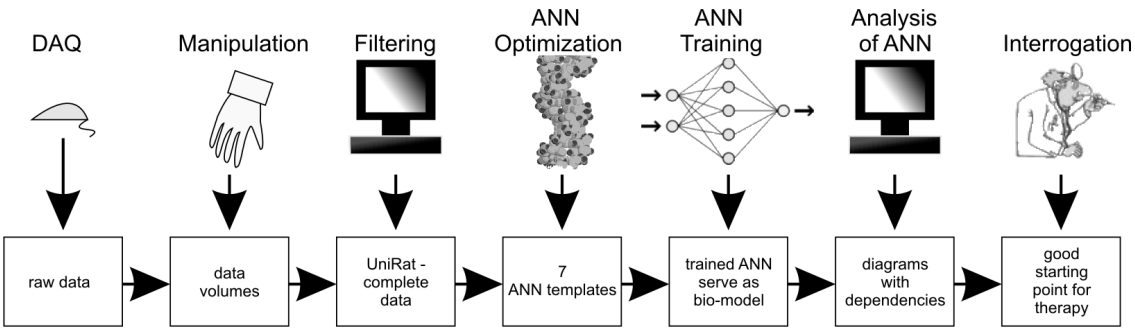


Image 1: Overview over the project steps.

2 Introduction

2.1 Document Conventions

There are some conventions used in this work used in the formula. Any vector is signed by a superior arrow and any matrix is signed by a superior double cross (#).

$$\begin{aligned} \vec{V} &: Vector \\ \overset{\#}{M} &: Matrix \end{aligned} \tag{1}$$

Letter case or italic/bold formatting is not used in order to indicate vectors or matrices.

$$\overset{\#}{A} \subset \overset{\#}{B} \tag{2}$$

Equation (2) indicates that matrix A is submatrix B. The exact orientation of the cut is context dependent.

(lat.) matrix = (en.) mother animal
(lat.) vector =(en.) carrier

2.2 Vagus Nerve Stimulation (VNS) and its Efficacy

The VNS is the only electric therapy method approved by the FDA (Food and Drugs Administration/USA 1998) in order to minimize the frequency of seizures in refractory epilepsies [13]. But its efficacy varies greatly and some of the reasons are well known whereas others are not. The efficacy of the stimulation improves over time and becomes best roughly after a year of application. The efficacy of the stimulation also degrades over time and becomes worst after many years when the Vagus Nerve's surface suffers from electric cicatrices [8-9]. These long term effects are accompanied by some short term deviations that seem to be affected by the choice of the parameters. The question arises whether the method can be improved by a more intelligent choice of the parameters than the use of thumb rules suggested in [53-55].

2.3 Subject of this Thesis

The subject of this thesis is to investigate the problem of finding the best parameter vectors (individual sets of values for the stimulator) for the VNS-Therapy from physical records made on rats. A feasible use of neural networks for data mining is attempted. The specified task is a classic domain of data mining and its methods and in many cases similar to a research done by Perner and Trautsch for the in vitro fertilization therapy in 1997 [14-16]. The work has provided new insights on the therapy but could not revolutionize it. As the authors state this is due the lack of detailed diagnostic measurements. This work has to deal with diagnostic measurements that are not solely depending on the input provided and discussed in due course. Hence a realistic purpose of this research is to provide new insights on the Vagus Nerve Stimulation but it will probably not be able to revolutionize the treatment of epilepsy.

In this thesis the performance of some CI-Methods is compared to more classical approaches of data mining and modeling according to applicability, performance and prediction capabilities. Not only the methods are compared but the obtained results are interpreted and a suggestion is made on how to use them in clinic situations or how better stimulators could be built.

2.4 Scope of the use of Data Mining Methods in this Work

Data Mining Methods either serve prediction or knowledge discovery. Prediction is the ultimate goal but in order to obtain it knowledge discovery must precede it. This work has a strong part in knowledge discovery and a smaller part in prediction. The goal is to understand how the parameter vectors influence an animal model population. The prediction is then done in the sense that any therapy will be specified under some criteria and then a predicted best starting point is predicted. As the reader already deems to understand it is not the goal to predict the precise behavior of an individual human or rat. Instead suggestions are made how to manipulate the vector in order to get the best result in the shortest amount of time.

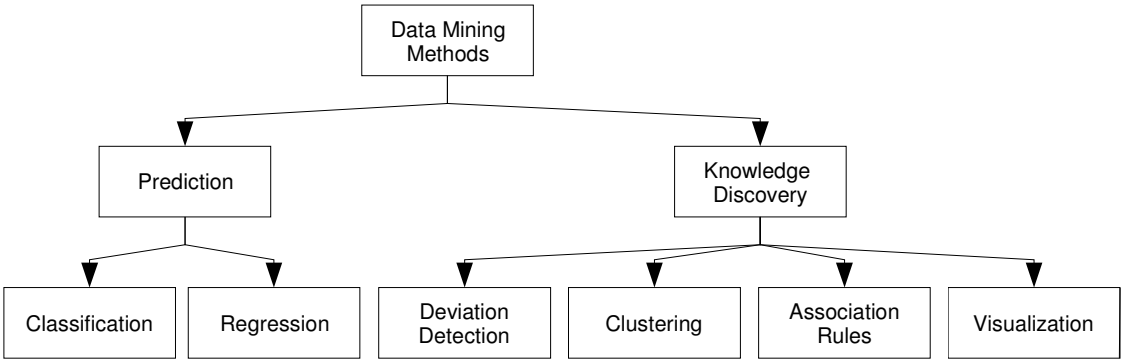


Image 2: Types of Data Mining Methods (refer to [15, p. 6-8])

- Prediction
    - Classification
        - K-NN Method can be used to predict a specific feature behavior. This method does not require the discovery of knowledge from data. The model is the data itself.
    - Regression
        - This work wants to predict feature intensity and direction according to numerical input values. Regression explains variability of one or more input variables. The applicable method is the creation of regression trees. [17-18]
- Knowledge Discovery
    - Deviation Detection

- Statistical method for search of unusual data behavior in correlation with specific parameters. Not applied to the presented research. *[15]*

- Clustering

  - Clustering is a process known to group n-dimensional attribute vectors into meaningful sets. Each set is then as similar as possible whereas members of a set should distinguish well from the other items in other sets. There are many different clustering methods in existence:

    - K-Means Cluster [15, p. 64]
    - Hierarchical or Agglomerate Clustering [15, p.62]
    - Fuzzy C-Means
    - SOTA [19]

- Association Rules

  - A field of knowledge discovery where semantically independent information is identified to exist at coincidentally with other packets of information. Because the number of retrievable information sources is small this technique has no importance in this research. [20-22]

- Visualization

  - Visualization techniques can be used in order to simplify the recognition of features in data [23]. In this sense visualization by itself is not a knowledge discovery method because the knowledge discovery is done by the human observer. Visualization is pretty much a data transformation into a suitable form from where the visual intelligence of the human brain can do the clustering or the detection of co-dependencies.

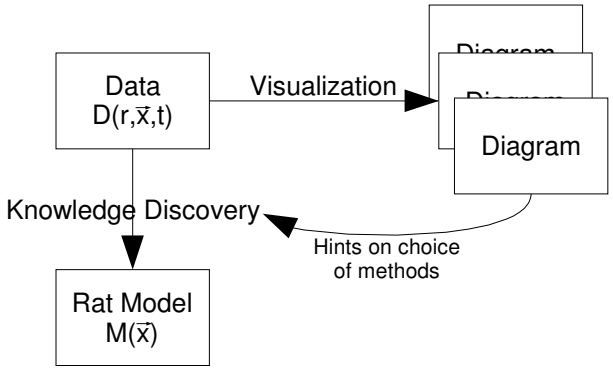


Image 3: The aim of Data Mining is to discover a rat-model that is only depending on the stimulation parameter vector x and is independent of absolute time or the rat's index.

2.5 Scope of the use of the Optimization Theory in this Work

Some parts of the research presented in this thesis require the use of optimization techniques. In order to better understand the individual choice a general overview over this field shall be presented in this place.

The optimization theory deals with two major types of problems. The first type is static whereas the second one is dynamic. Static optimization deals with problems that do not evolve over time and require a one time minimization. Dynamic optimization deals with systems evolving through time and takes care of minimizing the effects of disturbance on the optimized system. After such optimization process the dynamic system has the ability to heal itself from unwished states in as few steps and with as little deviation from the goal as possible. Therefore Control Theory and Cybernetics have a common mathematical framework with the dynamic section of the Optimization Theory. [24]

The best characterization of the difference between static and dynamic problems is the ability to constrain the problem to a known set of states. If it is possible to know all the possible states, then one is confronted with a static problem where one can just pick the best element from this set. Hence, a simple and effective but little time efficient method is to test every possible element of this known domain. Of course, Optimization Theory has developed methods much more efficient than that because some domains are still infinitely large. Contrary, in dynamic systems the description of this set is impossible because some of the variables will be assigned after the optimization process has been accomplished and very often such reassignments occur many times. A dynamic optimization method has to deal with a system that has some constant features on the first hand and some unknown features on the other hand. Ideally such unknown features should have a value of zero. It is the goal of the Control Theory which utilizes the dynamic optimization techniques to provide means of canceling out those unknown features also called disturbance variables. These mechanisms are called controllers.

Table 1: Static types of problems and examples of algorithms used to solve it.

	LINEAR	NON LINEAR
UNCONSTRAINED	Analytical solution	Quadratic Programming (GA)
CONSTRAINED	SIMPLEX	GA (Quadratic Programming)

For more information refer to the literature section [24-27]

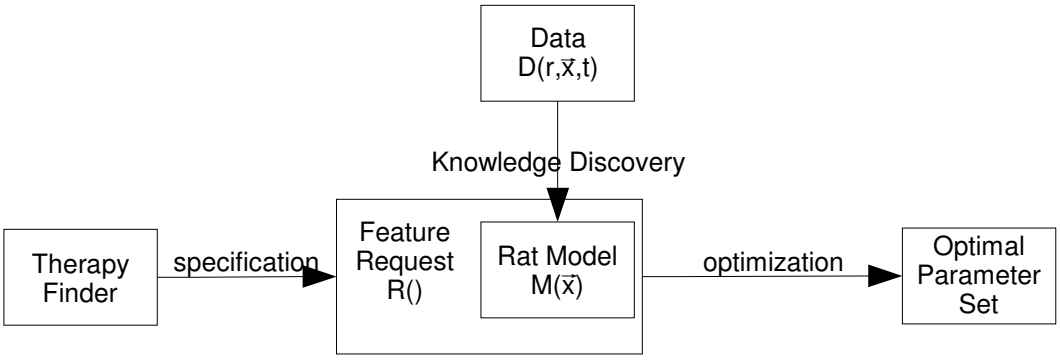


Image 4: The rat-model function M() is transformed via a feature-request function R() into a function that can be subject to min or max search.

Application of optimization methods can be discovered in a few places in this work. First, there is an optimization process in form of a minimum search in the combined function out of R() and M(). Second, for neural models 'training' means nothing else but to find an optimal state of weights, where a network can approximate the target function as close as possible. Interestingly this process is part of the knowledge discovery process that aims at providing the optimal model. Third, the ability of a network to achieve specific performance is depending on some structural parameters like learning rate, training method or the number and the size of layers. This optimization is implemented in form of a genetic algorithm. In other cases optimization can be discovered in decision or regression trees which separate data entries into optimal clusters or in interpolation methods or during Input Pattern Detection (IPD) where an optimal set of input dimensions is being searched for.

The main attention is focused on the title of this thesis: the optimal parameter vector to be used in conjunction with the VNS for decreasing ventilation rate and steady heart rate. This includes the minimization of the feature function R() in connection with the model function M().

$$y = R\left(M\left(\vec{x}\right)\right) \rightarrow 0 \tag{3}$$

Read chapters 3.15 and 3.16 in order to get more detailed information about the used hyperspace inference method.

2.6 Device Parameters

There are five parameters, which can be set on the (cybernics 104) stimulator. The first parameter defines the stimulation duration (T_{on}). The second parameter defines the pause between stimulations (T_{off}). The stimulation consists of square signals that occur periodically at the frequency (f). The temporal width of the square pulses is defined by the Pulse-width parameter (P). The current output is defined by the current-strength parameter (A). The stimulators today offer dozens of choices for each of the parameters. A combinatory experimenting is not possible due to the large amount of combinations. Assuming that each of the device parameters had just 20 possibilities, then this would result in 3.2 million combinations.

$$c = 20^5 = 3200000 \tag{4}$$

It is an exhaustible amount when considering that the current project focuses on a recording protocol of just 81 parameter vectors which already are difficult to record and to analyze. Appendix 7.1 references used parameters.

2.7 Electrode Interaction

In human body there are two Vagus Nerves [28]. The stimulation electrode is wrapped around one of them and then the attached stimulator can send electric impulses to the nerve in a manner specified by the parameter vector. The parameters are programmed with a handheld device and they are transmitted via an antenna to the stimulator residing beneath the patient's skin.

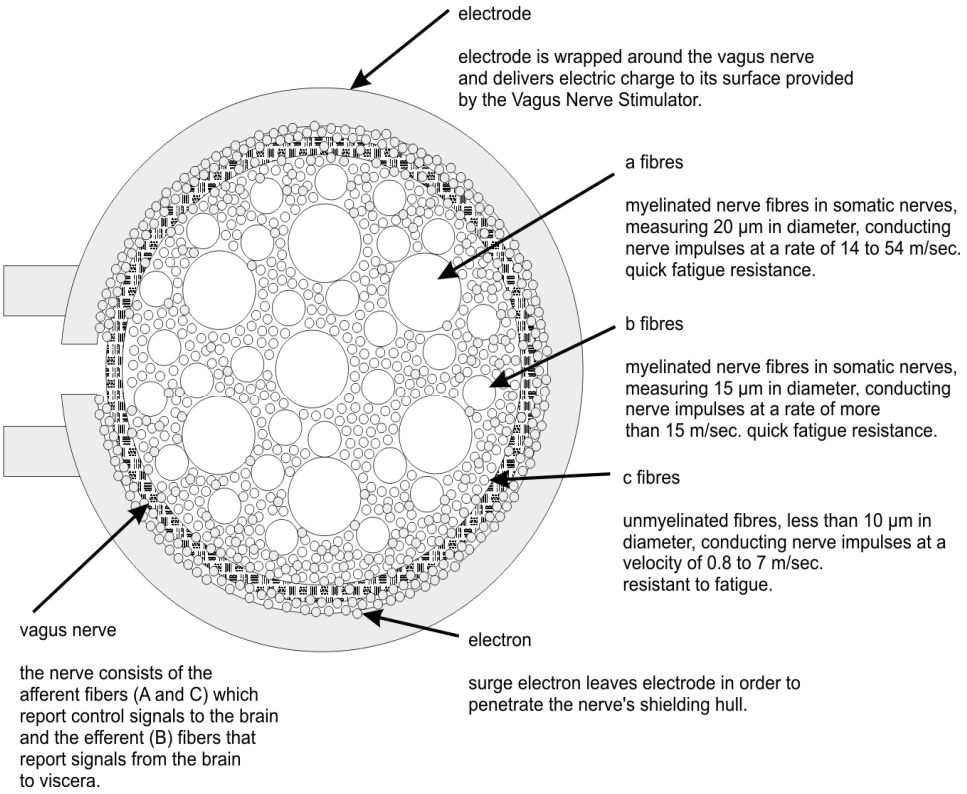


Image 5: The electrode is wrapped around the Vagus Nerve. Electric impulses inject electrons into the nerve. [29][30]

The idea behind the stimulation is to interrupt the afferent and efferent communication of the organs. It is expected that such an interrupt will enforce a change in oxygen absorption and distribution. Electrons saturate the nerves' surface suppressing eventual potentials for a neural spike. By far not all of the fibers can be blocked this way and the electrons can cause a signal themselves. Such signals can be harmful and painful. Parameters of that kind are associated with high currents and are excluded from the project.

The interactions between the electrode and the Vagus are not fully understood, but schematically look like image 5.

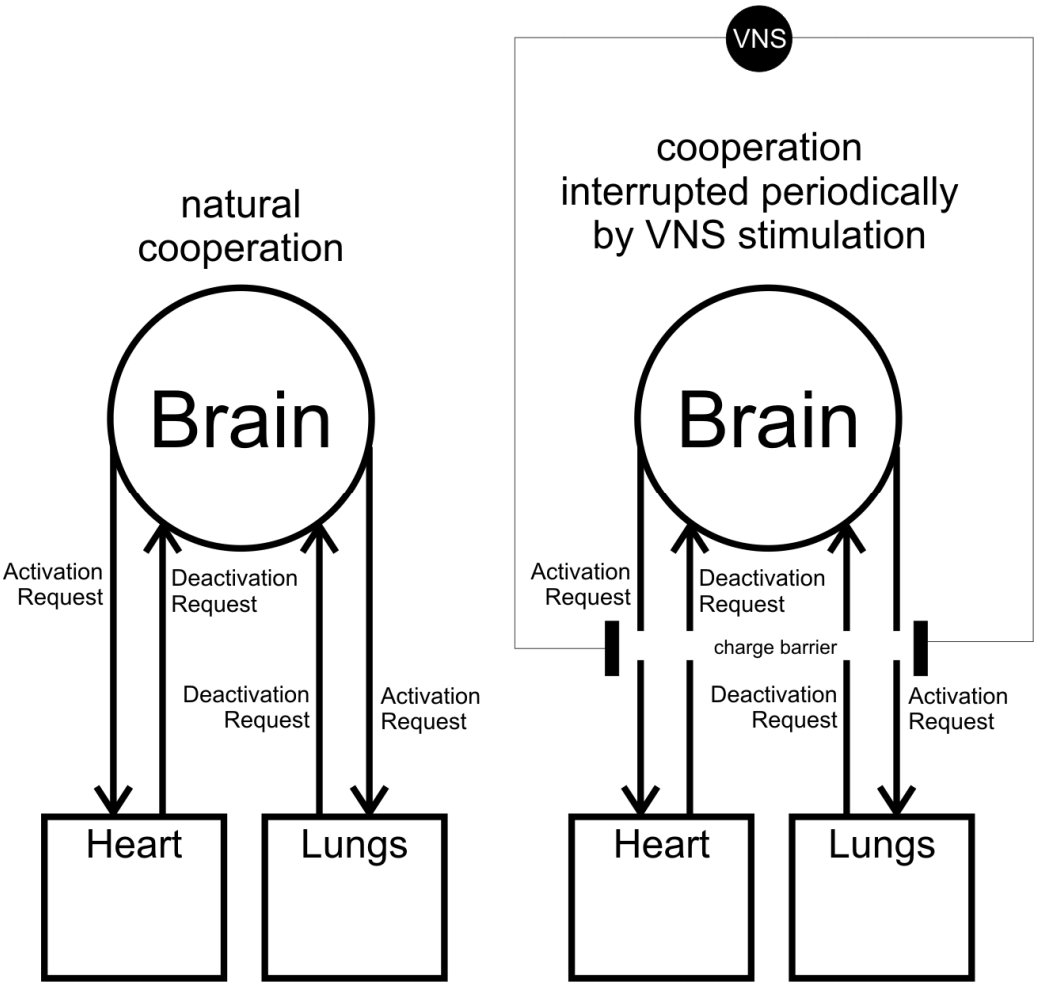


Image 6: Superficial operation scheme of the vagus nerve stimulator

The stimulator is adaptive in that sense, that it stimulates the Vagus Nerve (VN) with always the same amount of current by adapting its voltage level. The parameters P, A, f, T_{on} and T_{off} could be as well expressed as a single value: as a fraction of maximum power. This value is not representative enough for the problem, though. Variations in stimulus caused by the stimulation time and the pause cause a fluctuation of the oxygen saturation in the blood. [10] The nerve reacts differently to charging during the charging pulse and discharging at different absolute levels. Ignoring such important issues by combining the parameters into a single measure would make the analysis much simpler but the computer would also sway over islands of good parameter vectors without notice.

2.8 Medical Hypothesis

Medical thesis predicts that best therapy is given when strong changes in the ventilation rate (VR) are observed whereas heart rate (HR) changes occur as little as possible. It cannot be said for sure, that the assumed medical thesis is correct but it is guiding line during the development. The project is laid out in a way allowing other theories to be tested. This happens in the final stage of the project where medical staff can interrogate the system for parameters that seem to meet some specified criteria statistically. Thanks to the methods of computational intelligence many parameters can be tested and identified as best without having applied them on a real model before. This saves time and a lot of tests on animals. It is assumed that the recorded data does not provide the relevant features of the rat straight away; therefore five effects were extracted from the recorded signals:

1. Ventilation Rate (VR)
2. Ventilation Amplitude (VA)
3. Ventilation Slope (VS)
4. Inhalation Time (VI)
5. Exhalation Time (VX)
6. Minute Ventilation (VT)
7. Heart Rate (HR)

By adding additional features like Ventilation Amplitude or Inhalation Time a better parameter-set separation is possible. In clinic testing it might become obvious that heart rate or ventilation rate is irrelevant or just partially relevant and other features would gain more importance. Such conclusions cannot be done by the computer since the data does not contain any hints on the effects on the course of the illness.

2.9 Systematic Dependency between VNS Parameters and Rat Response

There exist hints, that the rat response to the Vagus Nerve stimulation is systematic to some extent. This hint is drawn from the data generated in the feature extraction phase:

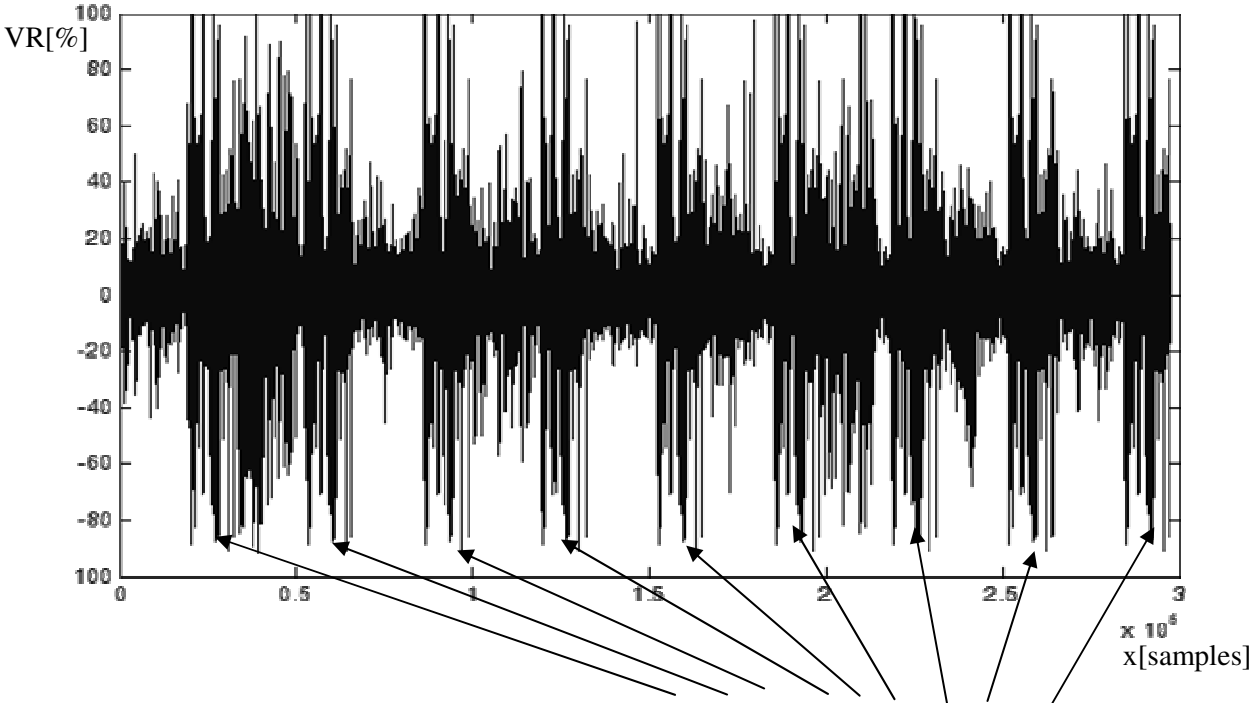


Image 7: Ventilation Rate pasted together from rats 16, 17, 19, 21, 22, 23, 24, 25, 26 shows some repeating patterns. Every rat was subject to the same protocol of stimulations.

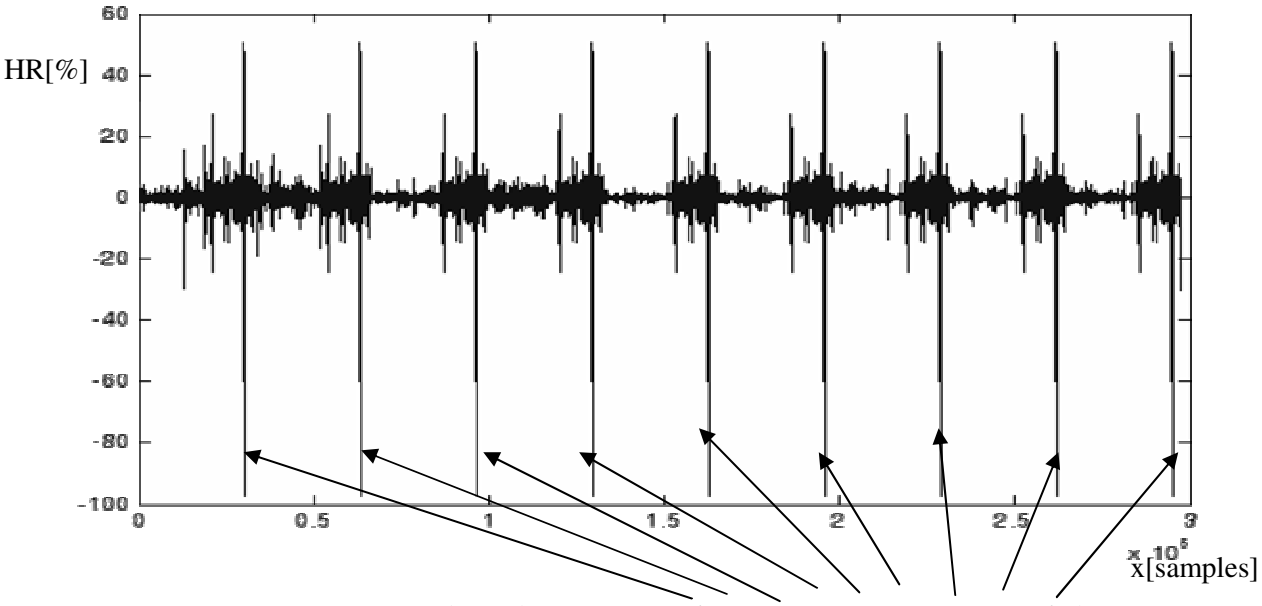


Image 8: Heart Rate (HR) pasted together from rats 16, 17, 19, 21, 22, 23, 24, 25, 26 shows some repeating patterns. Every rat was subject to the same protocol of stimulations.

2.10 Computational Challenges and Potential Risks

The OptiVaNeS project faces different computational challenges. The large data and the computational methods used require large computing power and memory, which sometimes cannot be easily provided. Generally, the project is threatened by different hazards. On the side of computing the following dangers are prominent:

- Large RAM required and alternatives might not be provided quickly enough before funding ends.
- Large computing power required and results might not be provided before funding ends.
- Data records are contradictory and similarities or systematic behavior are not obvious.
- Artificial Neural Networks can handle contradictions, but a wrong type of network can be chosen and relevant correlations might not become observable.
- Suboptimal training parameters which are a compromise between time pressure, available memory and available computing power.
- Programming errors.
- Various filters could be ineffective because they were built around some wrong assumptions.
- Other methods can deliver different results. Which one is right is difficult to decide.

3 Methods

3.1 Tools

Matlab was defined to be the project's main tool. Many parts of the software were created for the special purpose of this project in Matlab. Data recording is done with Spike2 [31].

3.2 Data Acquisition

The data used in the project is recorded from healthy rats. Epileptic rats' illness interferes with the recording process. Before recording, a rat must be sleeping or one cannot separate stimulation effects from effects occurring during consciousness. Epileptic rats suffer seizures relatively frequently which are making the rats run through various levels of wake state. A day of recording is not unusual until all 81 parameter vectors were recorded. Hence long natural sleep periods are required and healthy rats fulfill this requirement since they are active at night.

At first, the rat must be equipped with sensors. During a surgery the animal is prepared for the Vagus Nerve Stimulation. The electrode is wrapped around the nerve manually. During this surgery also the electrodes for the electrocardiogram (ECG) and the electric artifact from the stimulator are implanted. After quick recovery the rat's response to the stimulation can be recorded. The ventilation data is recorded from a barometric box (Plethysmography). The rat is feeling comfortable at all times. Any noise is avoided, only the stimulation changes interfere with the rat's sleep. This way the rat likes to sleep most of the time, which is a necessary requirement for the recording process.

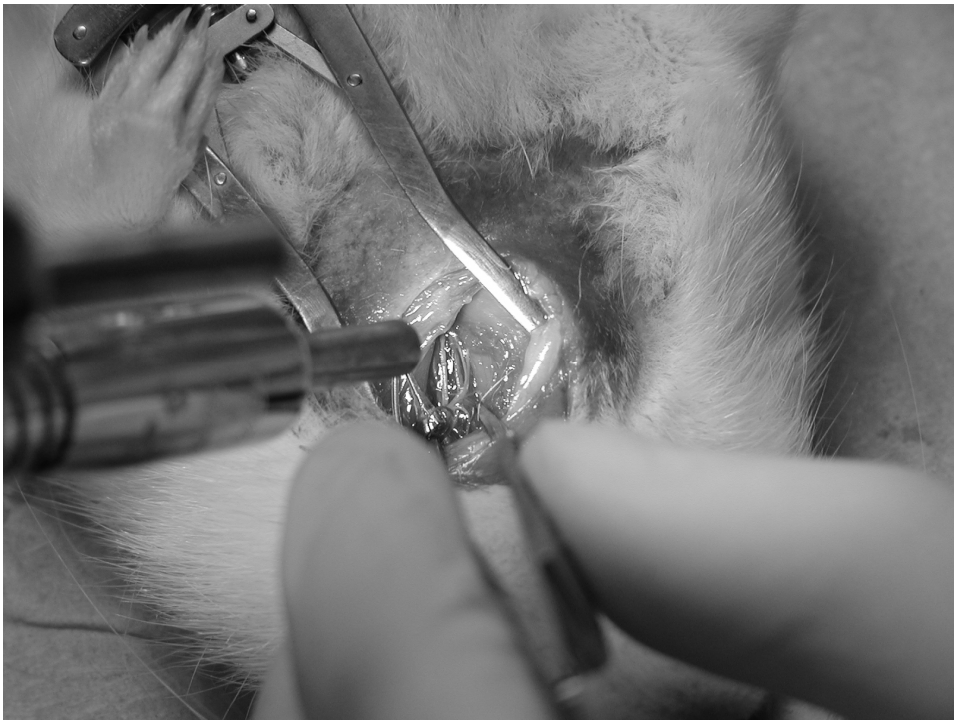


Image 9: Vagus Nerve of a rat used to acquire data

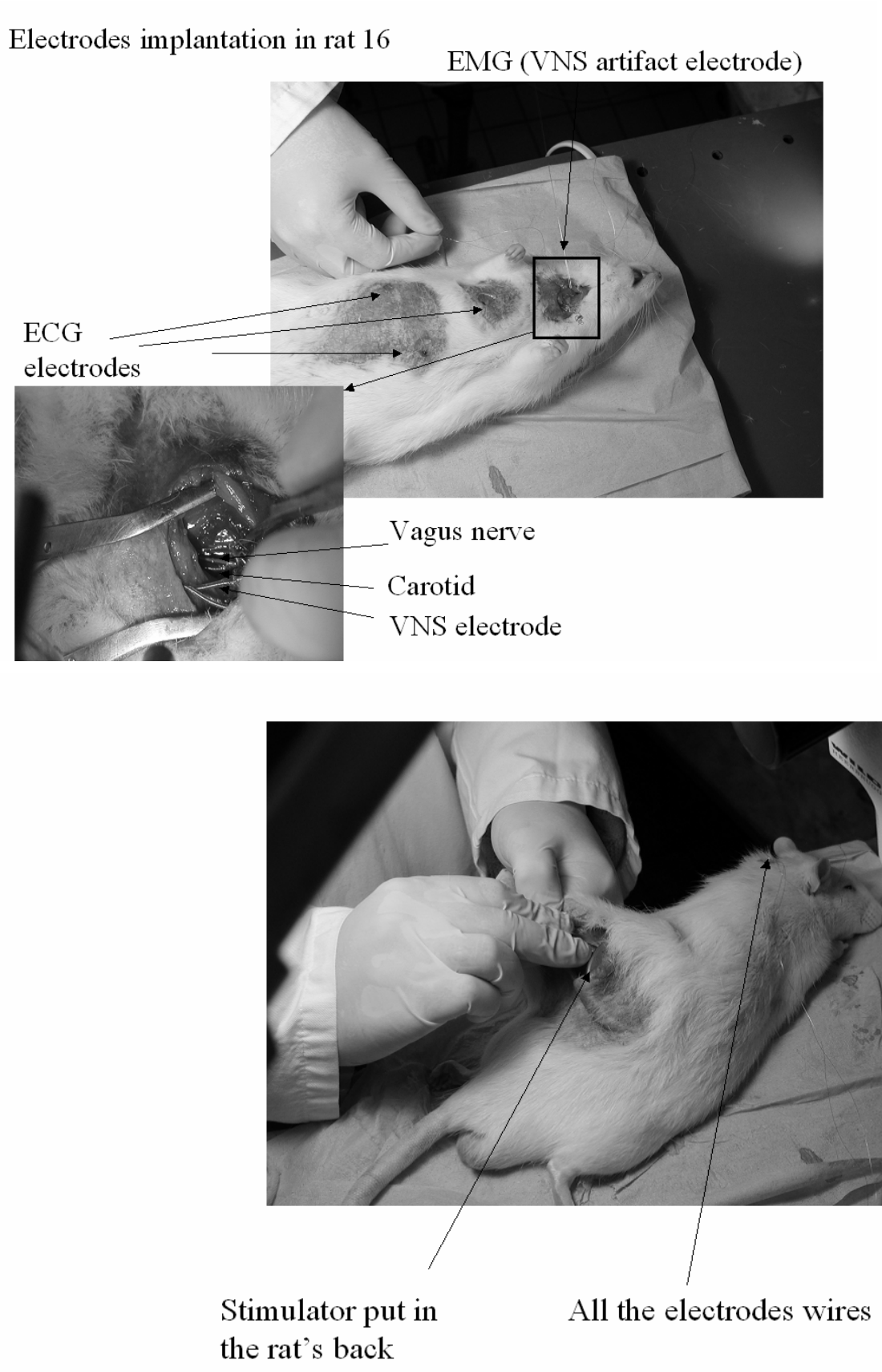


Image 10: Process of implantation

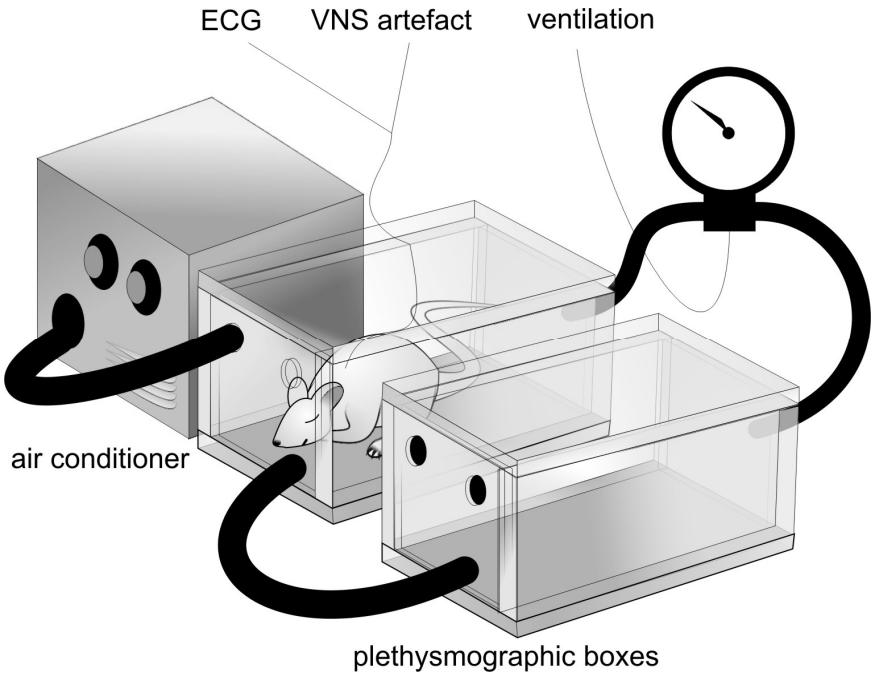


Image 11: The recording environment during data acquisition; three signals are being recorded: ECG, VNS artifact and the ventilation volume changes.

In order to complete a record each rat is stimulated with a set of 81 different parameter vectors. A total time of seven to nine hours of recording is easily achieved. The recording is done with 19" rack mounted amplifying A/D converters. The three signals electrocardiography (ECG), Vagus Nerve Stimulation artifact (VNS) and the barometric changes in the box due to ventilation (V) are attached to that device. The device transfers data to the computer through a USB communication channel where it is recorded with the Spike2 data acquisition software. This software doesn't allow interrupting the record and then continue in the same file. Each break would result in a new file. Stopping, saving the file, waiting and starting anew is an awfully error prone and time consuming process which would stretch the total time of work easily by two hours. Overwriting a file would produce errors that could not be recognized neither by human nor by computer. Hence the data acquisition is done in only three parts where the necessary time for the operation of the computer is low and errors can be mostly avoided. The resulting spike files typically contain 2½ hours of recorded data each and are roughly the size of 200 MB.

For each rat there is typically more than 500 MB of data recorded. For the project 11 rats were recorded of which two were removed again because the electrodes' fitting was bad. The total amount of raw data is billing up to 5GB. Roughly 5000 stimulations were recorded.

3.3 Data Pre-Processing

The recorded data is not free of errors and is not suitable for computational analysis directly. Therefore manual filtering is done in order to distill the relevant features. The features are corrupted by bad contacts on the electrodes, electrode misplacement, cross talking on the lines or simply by the rat toying around with the wires.

The filtering is not trivial and requires human's creative capabilities and a priori knowledge about the experiment in order to recognize parasitic effects and take proper action. After filter-

ing the features of the specific channels are recognizable enough for automatic analysis. The files thus far are Spike2 data files, which cannot be processed by Matlab directly. The export function in Spike is used to produce three comma separated values (CSV) text files from any single Spike file. Each file represents a channel. Due to the fact that CSV files have headers of different style, the data sections cannot be separated easily from the headers and the footers by the Matlab software. It was decided upon that removal by hand is easier than writing sophisticated data section recognition. With the headers and footers in the text file Matlab sometimes even denies importing. The text files are extremely large and can reach many hundred Megabytes in size. A simple text editing tool is not appropriate for removal of the headers. Instead a freeware hex editor XVI32 by Christian Maas is used [32]. The advantage is that large files load quickly and that block modifications are performed immediately.

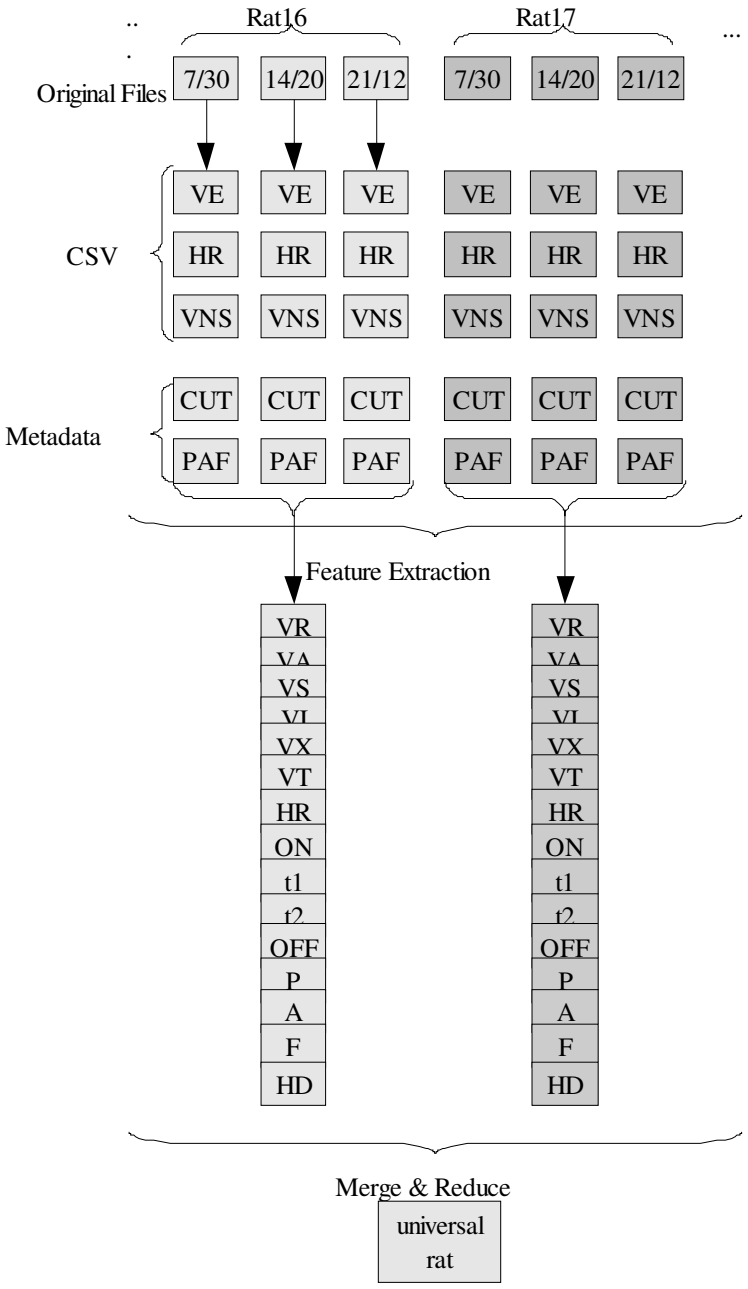


Image 12: In this scheme the transformations of the data are shown.

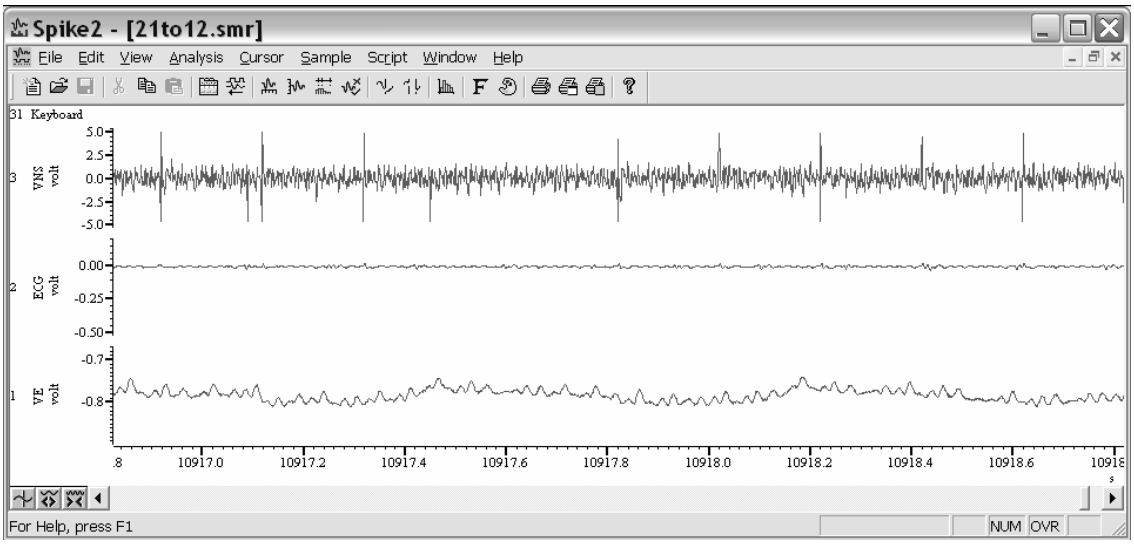


Image 13: Typical problems during recording are strong noise on the VNS channel, dangling wires with hardly any signal and cross talking on the lines.

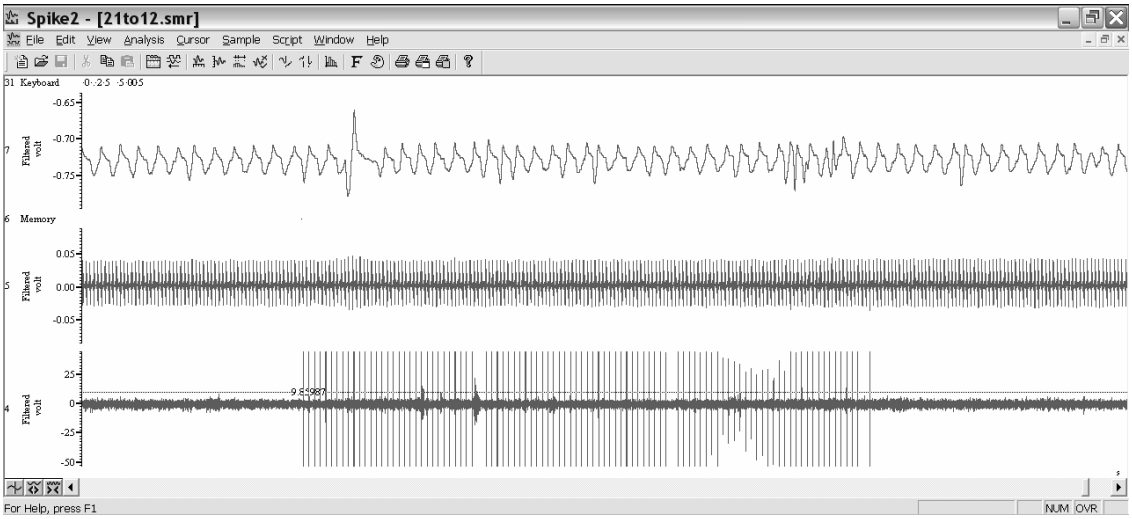


Image 14: Manual filtering provides signals analyzable by a computer.

After manipulation the files must be saved in a directory specially created for each rat. The data requires some additional information which is best described as metadata. This metadata contains the following information:

- *Sleep state and garbage separation times.* The data channels are reviewed manually in order to recognize a wake state or the opening of the barometric box (PG). This step requires the knowledge of how the PG signal is modified when the box is opened and requires a careful read of the keyboard channel in which sleep and wake states are held on record. Sometimes it is needed to mark other parts of the data for deletion, because no filtering can help to make a good signal out of the record. The files describing the parts for deletion are referred to as the CUT-files.

- *Manual description of when which parameter vectors were used.* This information is typed into Spike2 during record. The files' keyboard channel has to be reviewed manually in order to extract the typed information about the parameter vector. The files resulting from this process are referred to as the PAF-files.

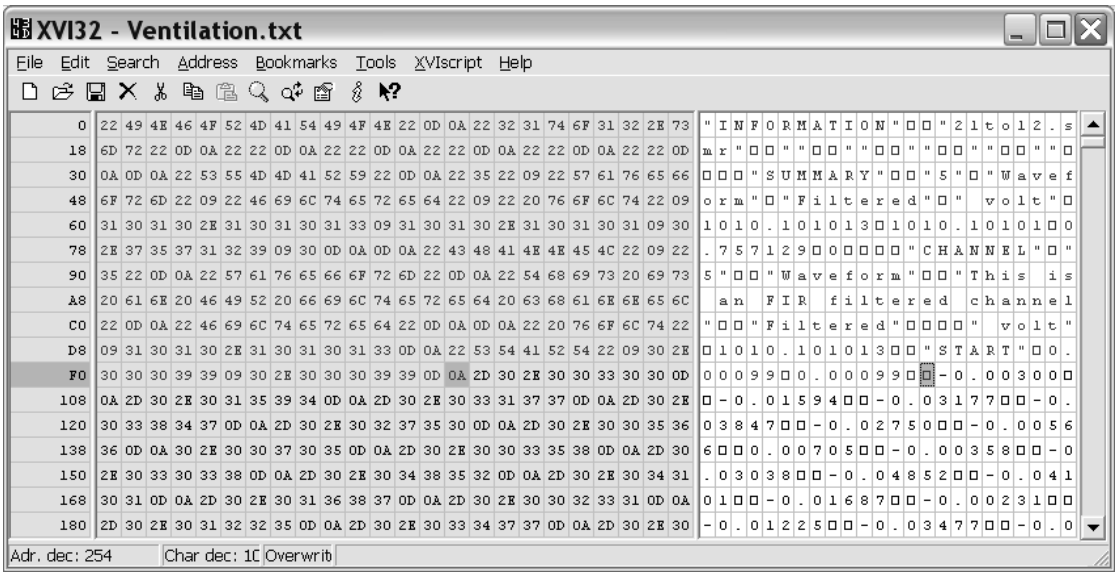


Image 15: XVI32 Hex Editor is used in order to remove headers and footers from the file.

As the files are standing relatively independently from each other in any of the rats' directories, a configuration file is needed. This files supplies all the information required to fabricate a virtual rat.

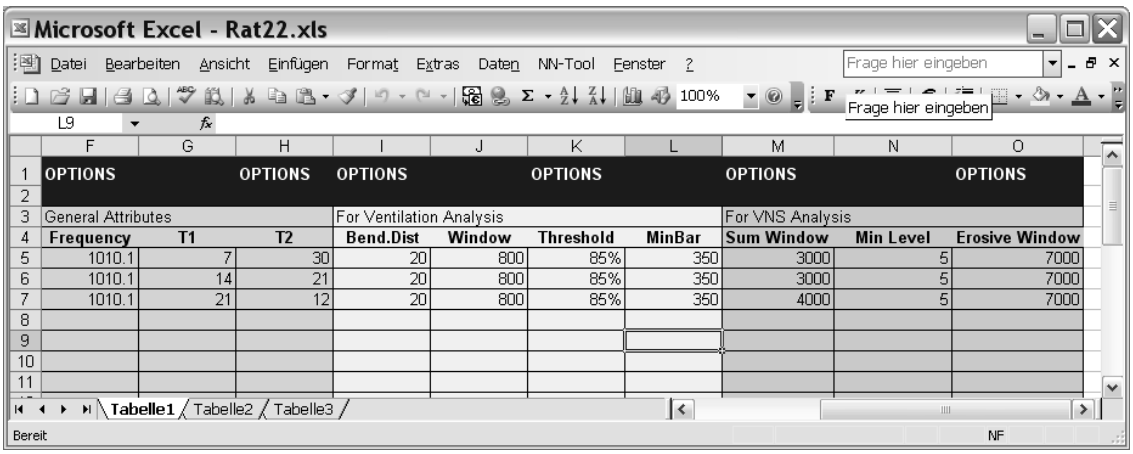


Image 16: In any rat directory there is a configuration Excel file where each row defines a volume with its data, metadata and options.

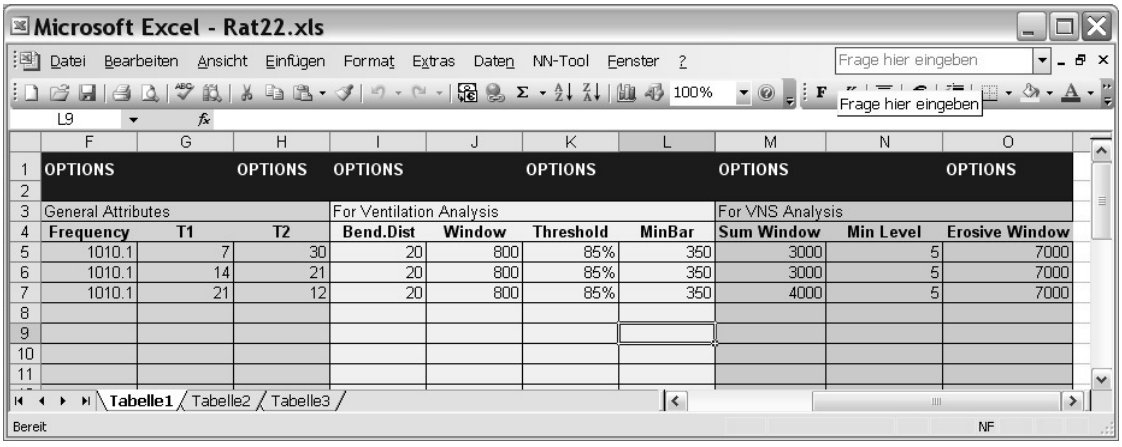


Image 17: Options for each of the volumes: Many filters require to know the volume's sampling rate.

Each volume was recorded at a different T_{on}/T_{off} (T1/T2) relationship. For better recognition of the ventilation signal or the VNS artifact the filter parameters (Bending Distance, Window, Threshold, MinBar, Sum Window, Min Level and Erosive Window size can be altered.

3.4 Artificial VNS

Sometimes VNS cannot be filtered out of the record up to a satisfactory level. In this case human help is required. A table of stimulations is created specifying when and how long stimulation occurred. With the help of a Matlab program this data is converted into an artificial VNS artifact that substitutes original data. The substitute is a high quality square signal.

3.5 The Extraction Process

3.5.1 Configuration

After configuration of the rat Excel file the extraction process can take place. The extraction process has 15 parameters which can be generated with defaults.

The defaults include no heart or ventilation visualization and a reduction level of 8. This level can be changed to whatever level is required (See next chapter for more information). The ConfigFile parameter is mandatory. All other parameters serve debugging purposes and can be left as are.

3.5.2 Extraction

In this process the three channels representing ventilation, heart and stimulator operation are transformed into eight channels 'ventilation rate' (VR), 'ventilation amplitude' (VA), 'ventilation slope' (VS), 'inhalation time' (VI), 'exhalation time' (VX), 'minute ventilation' (VT), 'heart rate' (HR), 'heart deviation' (HD). The HD is connected to HR and represents deviations from average in percent.

First, the features must be extracted. For each channel a Matlab program is applied that generates related information. After feature extraction, some channels must be generated from metadata. Finally the sections marked for deletion are removed. This happens quite late because the discarded parts of the data cannot be deleted right away. If they were then tracks generated later on would have complicated shifts against original data. The simplest way avoiding this is to operate on full data (including discarded sections) until all data tracks are generated and later to perform delete operations on all tracks simultaneously.

At that point a raw binary file was created including all of the rats data. A statistical component is run over that file in order to display the final product. This includes graphs for averages of any occurring parameter vector and graphs with the data base of that average. Excel files are generated with the data shown in the graphs with the averages. From this Excel data custom diagrams can be made.

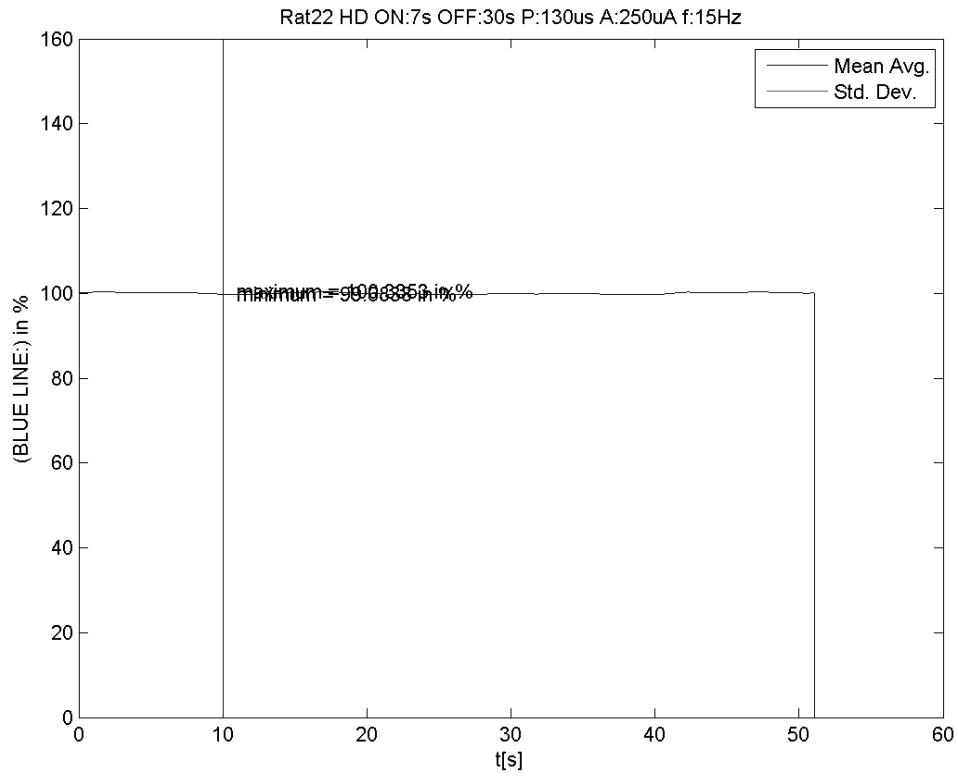


Image 18: A graph showing heart deviations occurring during the vector ON:7s OFF:30s P:130µs A:250µA f:15Hz in Rat22.

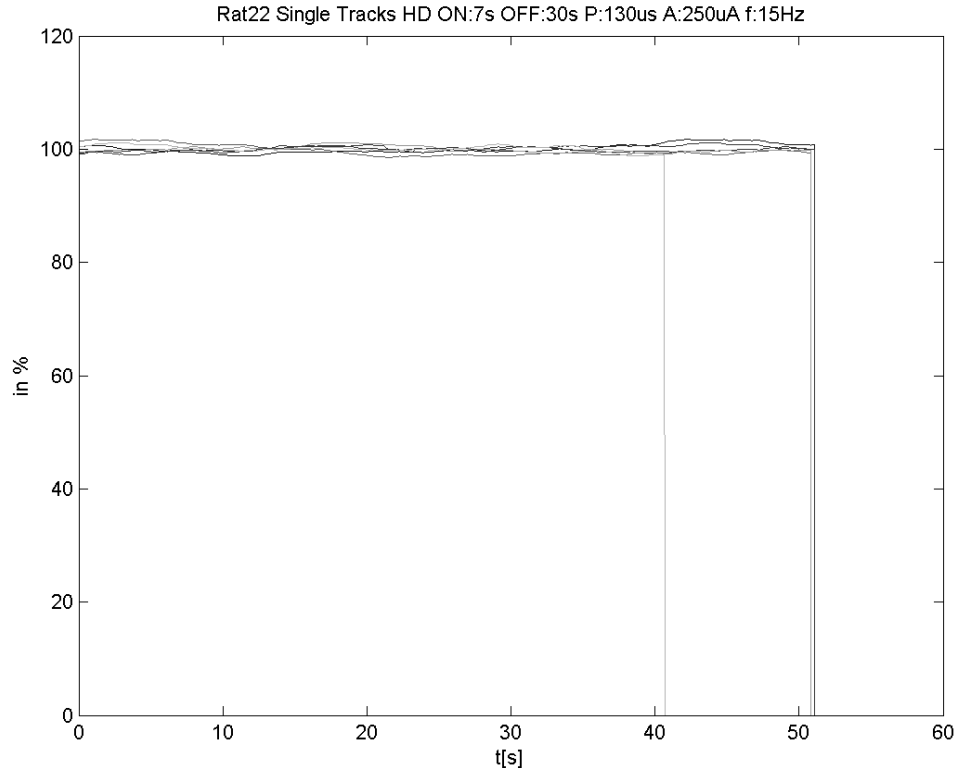


Image 19: The same rat and the same parameter vector as in image 8, but this time the data for the construction of the average is shown. It can be noted that the records are not all the same length. The program takes care of that problem by taking averages only from available samples.

3.6 Data Reduction and Redundancy

During the extraction process data can be reduced in a way that doesn't harm the information level. The channel with the finest final features is the heart rate. The heart beat occurs at no more than 600 beats per minute. For compare: the other features occur at rates as low as maximum 200 and 2. Therefore after heart rate extraction each feature is consuming minimum 101 samples to represent exactly the same event.

Table 2: Example of how data is used for the generation of the diagrams.

Feature	***Max. no. of occurrences per minute***
Heart Cycle	600
Ventilative Cycle	200
Stimulation Cycle	2

$$r = \frac{60\frac{s}{Minute} * SamplingFrequency}{OccurenceRate} = \frac{60 * 1010.1}{600} = 101.1$$

According to the same formula ventilation events occupy at least 303 samples and stimulation events not less than 30303 samples. The total amount of records can safely be reduced by a factor of 100. In that case still every heart beat is represented by at least one record. Every breath is represented by at least 3 records.

The so reduced data is still too large for further analysis. Especially computer memory puts strains on what size data can become. During application of CI-Methods memory shortages are common. Successive optimization of the software and a steady increase of the reduction factor led to a state where computing is possible. At that point of reduction an error level of 4,18% was reached.

The method for assessing this error is:

f_1: original function
f_2: subsampled function
r: reduction factor
n: number of records after subsampling

$$d = \sum_{t=0}^{n-1} f_1(t \cdot r) - f_1(t) \tag{5}$$

$$e = \frac{100 \cdot d}{\sum_{t=0}^{rn-r} f_1(t)} - 100 \tag{6}$$

When applied to data the following table was obtained:

Error	Downsampling Factor
0.82%	50
1.68%	100
2.79%	200
4.28%	400
5.86%	800

A very interesting observation is that a reduction of 100 causes an error of 1.68% although no information went lost. This is due to phase shifts that can always occur during sampling and is also true for subsampling since it cannot be guaranteed that the subsampled value will not cover also some of its neighbors in its source function.

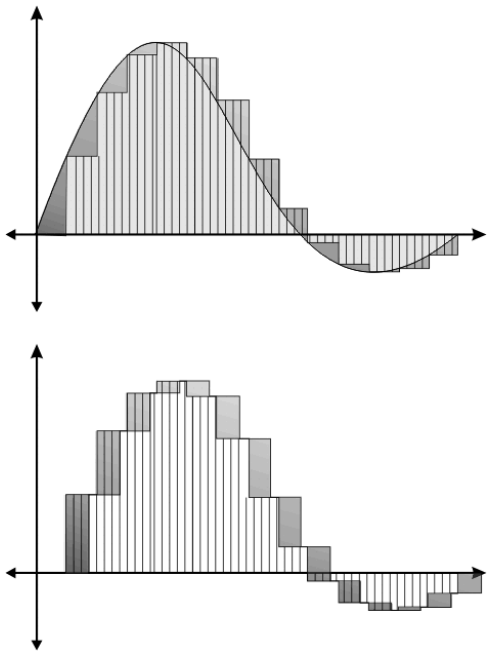


Image 20: Phase shifts during digitalization also occur when source function is digital itself.

3.7 Building UniRat

In the final step as shown in image 8 the many single files are merged into one single binary file with the name “universal.double”. The file is a raw binary file with a sequence of records formed of 15 double values per record. Each such record represents a sample for all of the 15 values. The file is loaded by invoking the LoadUniversal() function.

3.8 Extracted Features

The features a (VA), b (VR), c (VI), d (VX) and e (HR) were extracted. The other features like ventilation slope (VS) were calculated.

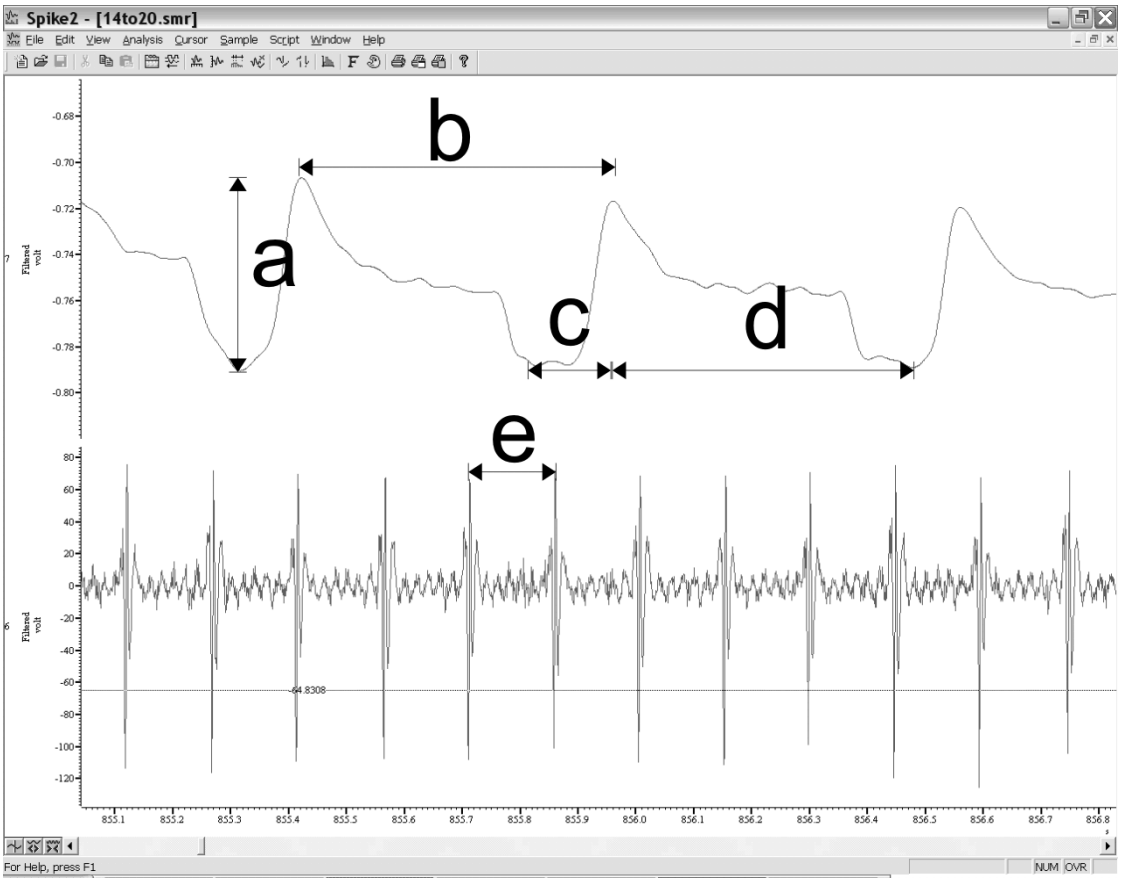


Image 17: (a) Amplitude (b) Inverse of ventilation rate (c) Inhilation time (d) Exhilation time (e) Inverse of heart rate

Calculated Features

Ventilation slope:

$$VS = \frac{a}{c} \tag{7}$$

Minute Ventilation:

$$VT = VA \cdot VR \tag{8}$$

Heart rate deviation: (whereas w is the size of the window and t is the sample index)

$$HD(t) = \frac{HR(t)}{\frac{1}{w}\sum_{i=\frac{-w}{2}}^{\frac{w}{2}} HR(t+i)} \tag{9}$$

3.9 Extracting Ventilation Related Features

3.9.1 Examples Of Signals

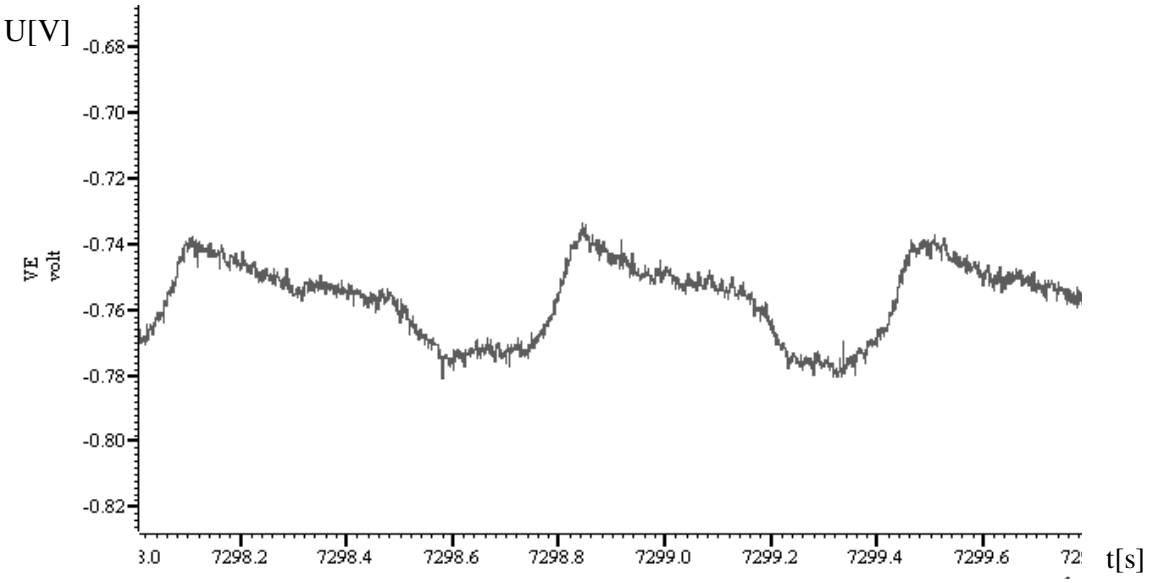


Image 21: Typical signal from the barometric box (Plethysmography) indicates ventilation operation.

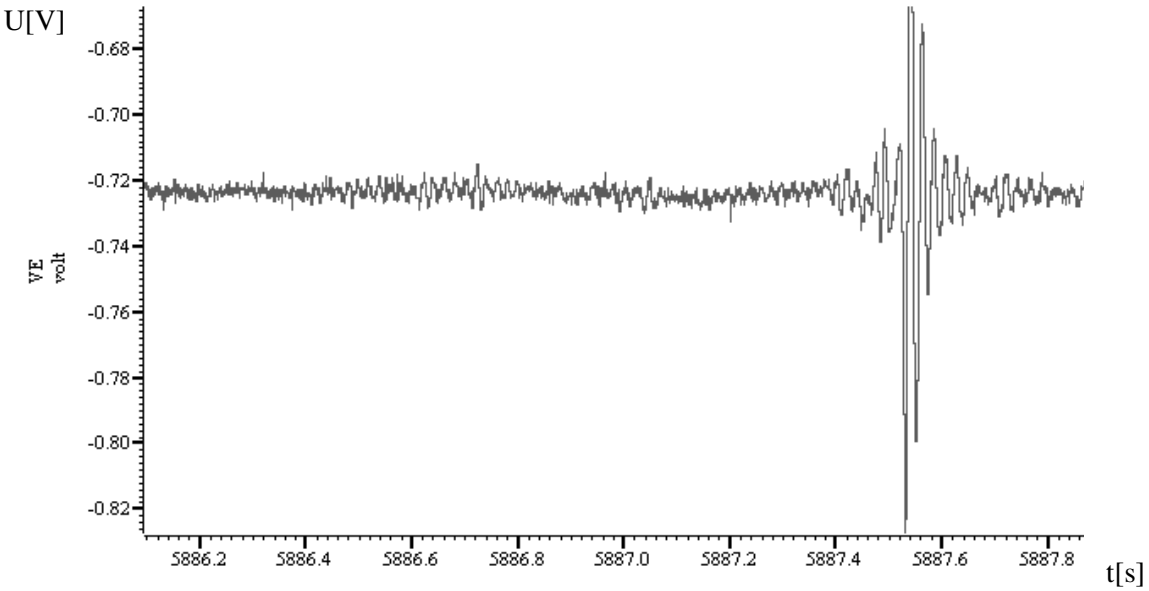


Image 22: Same channel as Image 20: Ventilation signal is gone. Noise is within size of regular operation.

3.9.2 Detecting Ventilation Cycles

The ventilation is recorded from a barometric setup like shown in image 6. Measuring the volume of the air inhaled is not easy with humans and is even more complicated with animals.

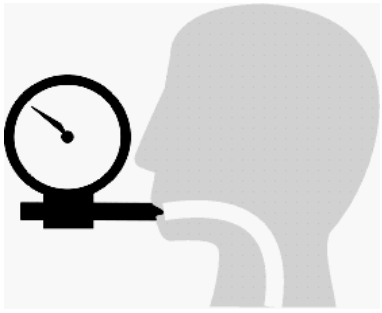


Image 23: Principle behind measuring volume achieved during a breath stroke

Although the idea behind measuring is as trivial as depicted in image 20 no animal will voluntarily allow such an invasive method. Plethysmography is not invasive and is used instead.

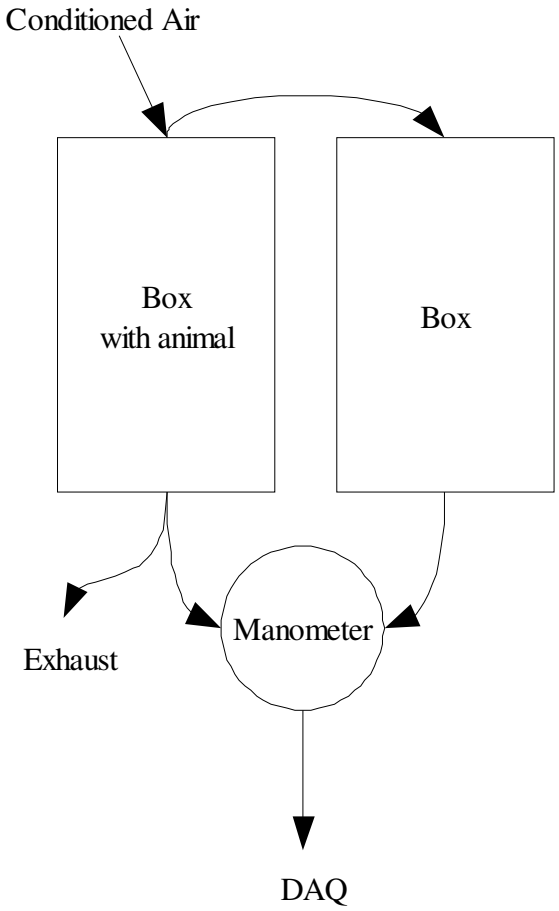


Image 24: Plethysmography measures changes in pressure between the box with the animal and the reference box.

Although Plethysmography is a differential measurement method a linear relationship between animal's lung volume and the recorded signal is assumed. In order to clarify that relationship a Matlab/Simulink model was developed. Some of the parameters are guessed, but they were chosen close to reality.

The model begins with an envelope generator that generates a typical respiratory stroke. The generator is designed to produce 120 ventilation cycles per minute. In order to simulate the

effects of the lung volume on the pressure of the manometer sensor it is important to convert volume into air flow. Air flow is expressed in the unit mol/s. According the law of ideal gases there is a relationship between pressure P, volume V, the number of moles n, the universal gas constant R, the temperature T, the number of molecules N and the Boltzmann constant k:

$$P \cdot V = n \cdot R \cdot T = N \cdot k \cdot T \tag{10}$$

$$n = \frac{P \cdot V}{R \cdot T} \tag{11}$$

$$V = 0.1^3 m^3 \tag{12}$$

$$T = 20°C \tag{13}$$

$$P = 1013hPa = 101300 \frac{N}{m^2} \tag{14}$$

$$R = 8.3145 \frac{J}{mol \cdot K} \tag{15}$$

$$n = \frac{101300 \frac{N}{m^2} \cdot 10^{-3} m^3}{8.3145 \frac{J}{mol \cdot K} \cdot 293K} \lim_{x \to \infty} \approx 41.58mol \tag{16}$$

Equivalent to one liter@stp (at standard pressure) where it is irrelevant what gas it is, although it is favorably used for monatomic gases at high temperatures and low pressures.

The volume of the lung is expressed in mol by multiplying n with the volume of the rat's lung. As there is resistance to the lungs the rat a P-T_1 transfer function was used to model air flow delays. Assuming that a rat's lung volume was

$$V_L = 25cm^3 = 25\mu m^3 \tag{17}$$

$$C_L = \frac{n_{stp} \cdot V_L}{STP} = \frac{41.58 \frac{mol}{m^3} \cdot 0.000025m^3}{101300 \frac{N}{m^2}} = 10.26 \cdot 10^{-9} \frac{mol \cdot m^2}{N} \tag{18}$$

C_L describes the capacitance of the rat's lung. RL is a (guessed) friction constant that causes some time delay before lung is full.

$$R_L = 1 \cdot 10^6 \frac{N \cdot s}{mol \cdot m^2} \tag{19}$$

The differential equation for a P-T_1 transfer function:

$$R_L \cdot C_L \cdot \dot{x}(t) + x(t) = K \cdot u(t) \tag{20}$$

u(t) : Volume of lungs x(t) : air mass in lungs

The Laplace transformed form the equation is:

$$F_L(\mathbf{s}) = \frac{K}{T \cdot \mathbf{s} + 1} = \frac{n \cdot V_L}{R_L \cdot C_L \cdot \mathbf{s} + 1} = \frac{103.95 \cdot 10^{-6} mol \cdot s}{0.01s \cdot \mathbf{s} + 1} \tag{21}$$

The equation (21) describes the animal's lung fill in Laplace domain.

$$V_L(\mathbf{s}) = \frac{K \cdot \mathbf{s}}{T \cdot \mathbf{s} + 1} = \frac{n \cdot V_L \cdot \mathbf{s}}{R_L \cdot C_L \cdot \mathbf{s} + 1} = \frac{103.95 \cdot 10^{-6} mol \cdot \mathbf{s}}{0.01s \cdot \mathbf{s} + 1} \tag{22}$$

"Lung's air flow" obtained by differentiation of equation(22).

There are two boxes in the process. The one box hosts the animal and the other box serves as a pressure reference. The rat's box is collecting air blown in the air conditioner. The flow of the conditioned air is increased and diminished by the animal's breathing activity. As more and more air is pressed into the box the pressure raises in the box. But any of the boxes has a small outlet and there is a hose connecting the two boxes where air can elude. This way pressure does not rise into infinity.

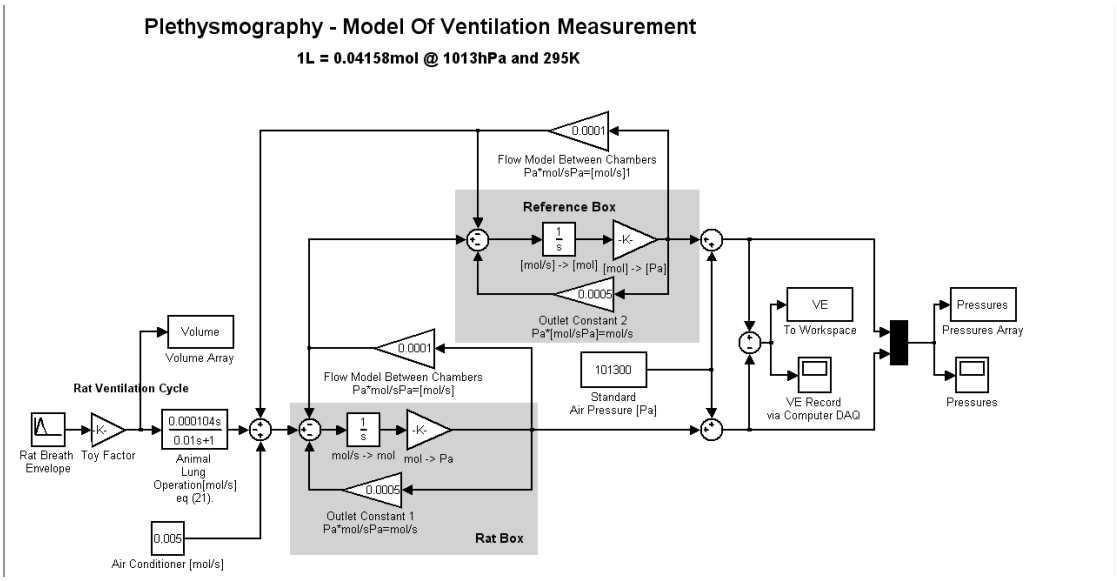


Image 25. Model of a plethysmographic session in Matlab/Simulink

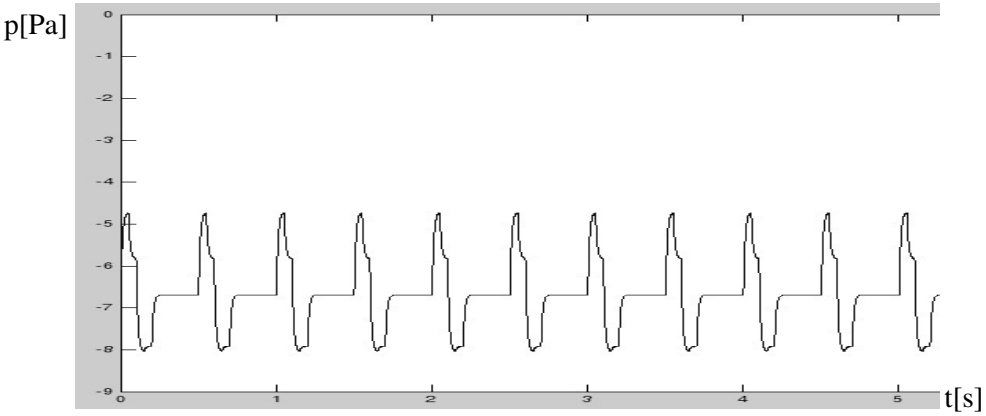


Image 26: Obtained signal reminds of original data which is also negative.

As pressure rises the elusion through the holes increases as well. The strength of leakage can be adjusted by the "Outlet Constants" in [mol/sPa]. The factor for converting the unit mol to Pa is taken from equation(10).

$$V = 0.0025m^3 \tag{23}$$

$$T = 295K \tag{24}$$

$$R = 8.3145\frac{J}{mol \cdot K} \tag{25}$$

$$P = \frac{n \cdot R \cdot T}{V} = n \cdot c \Rightarrow c = \frac{R \cdot T}{V} = \frac{8.3145\frac{J}{mol \cdot K} \cdot 295K}{0.0025m^3} = 981111\frac{J}{mol \cdot m^3}, \left[\frac{Pa}{mol}\right] \tag{26}$$

P describes the pressure that would be created by 1 mol of ideal gas at the specified temperature in the specified volume V.

Because the boxes are connected by a hose the difference in pressure between the boxes will cancel out, ultimately. The flow of air between the boxes is controlled by the "Flow Model Between Chambers" variable. There are two of these variables - one for each direction of the flow. Although asymmetric flow is possible symmetry was chosen in order to best reflect a simple hose.

Finally, a "Toy Factor" is introduced in order to investigate dependency between the actual lung volume and measured pressure differences between the two boxes. The result is very linear.

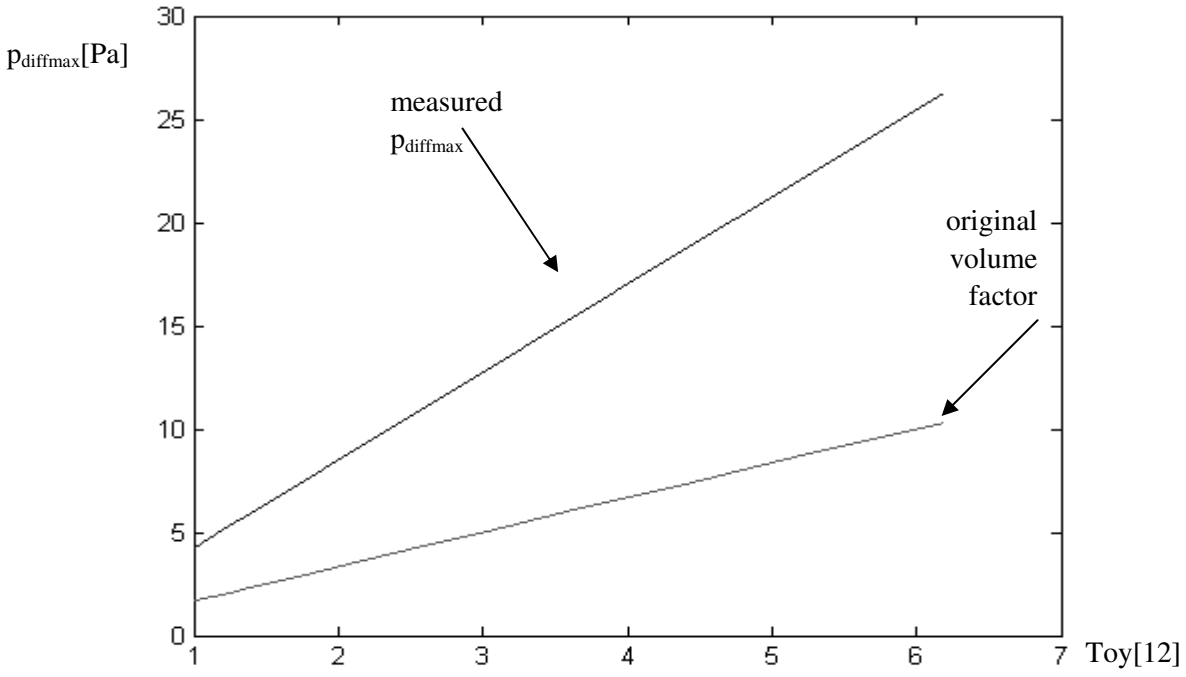


Image 27: Sensor's response to increased ventilation volume.

If the pressure to voltage factor is known then the Simulink model can be adjusted to match real data.

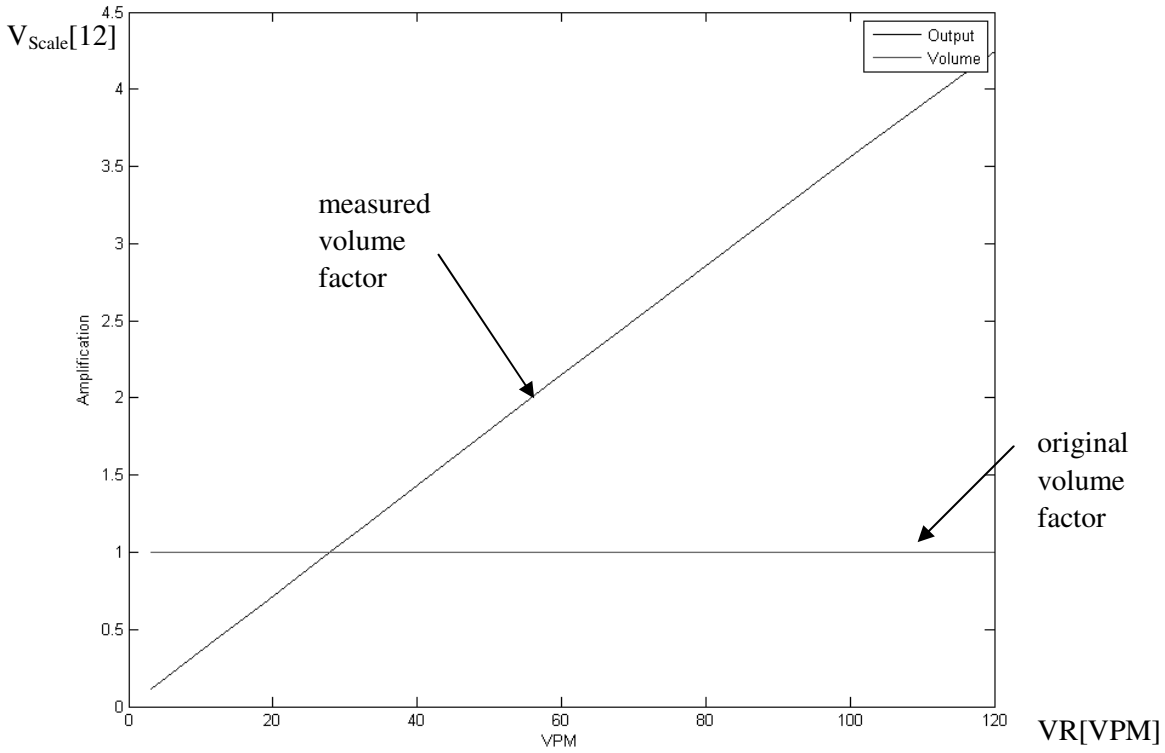


Image 28: Sensor's response to constant volume but increasing ventilation rate.

The diagram shown in image 23 suggests that amplitude of the signal should be normalized by the detected ventilation rate. This suggestion seems to be mandatory although it is not done in the project. Amplification in real data seems not to be depending too heavily on ventilation rate. This issue should be investigated later.

The inherent dependency of measured ventilation amplitude on ventilation rate might be considered negligible since the observed changes in ventilation rate are rather small and thus the amplitude interference.

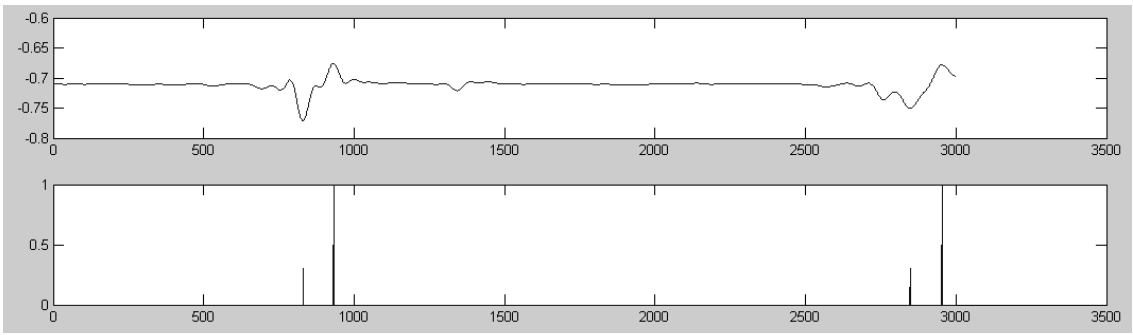


Image 29: Plethysmography is a differential measurement method. When volume changes stop then signal returns back to normal.

3.10 Extracting Heart Related Features

3.10.1 Quality

There are different levels of quality obtained during data acquisition.

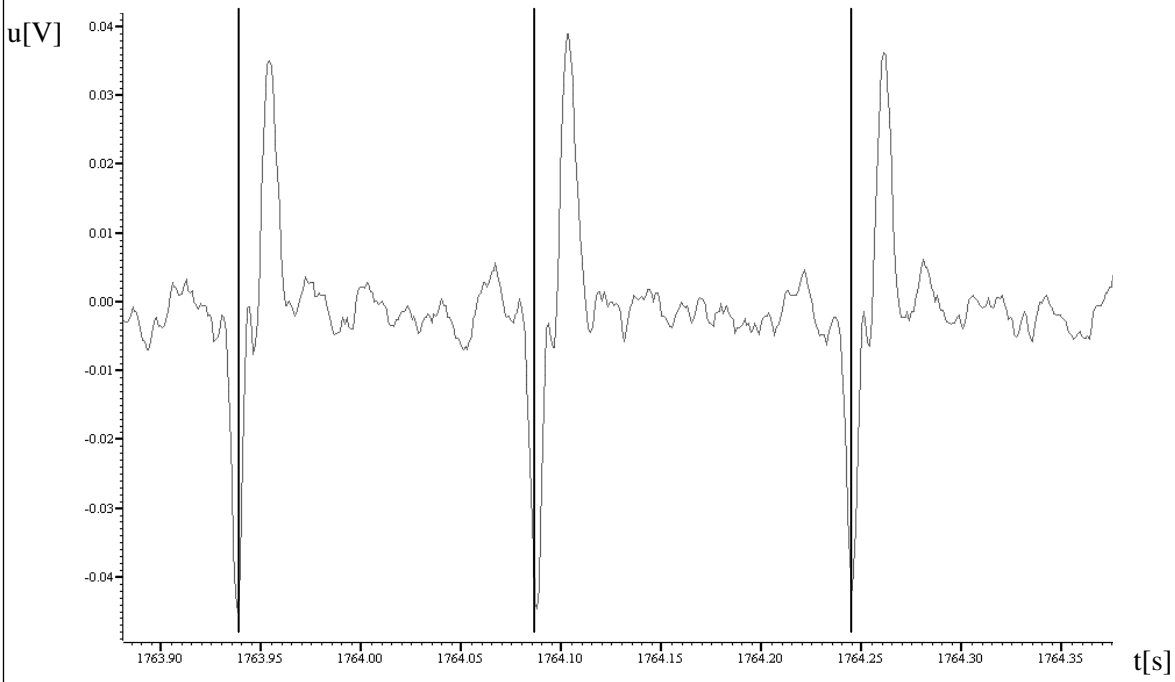


Image 30: ECG signal of good quality: Black marks indicate heart beats.

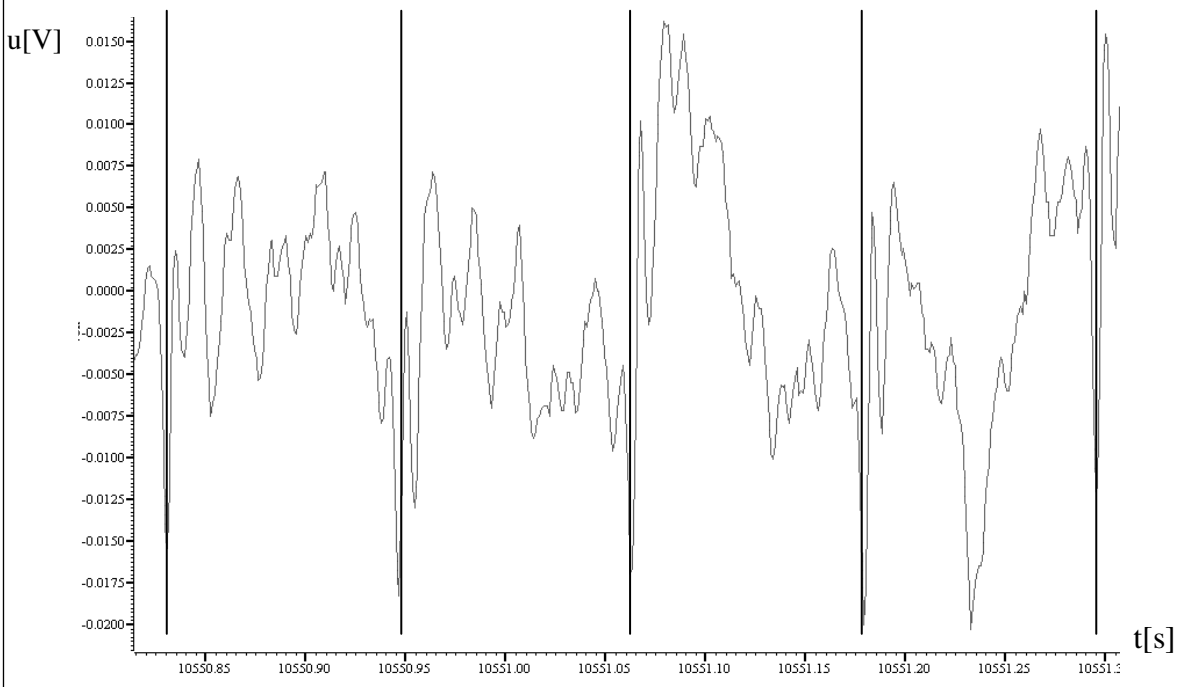


Image 31: ECG signal of bad quality but still recognizable. Black marks indicate heart beats.

3.10.2 General Properties of the ECG-signal

The ECG-Signal is recorded at a sampling frequency of 1010.1Hz. Even though it is saved in a 64 bit floating point format it is recorded at depth resolution of 16 bits.

3.10.3 Fidelity of the Threshold Method

During the project a discussion arose about the quality of heart beat detection with a threshold. Hitherto beat detection was done with threshold-marks-method..

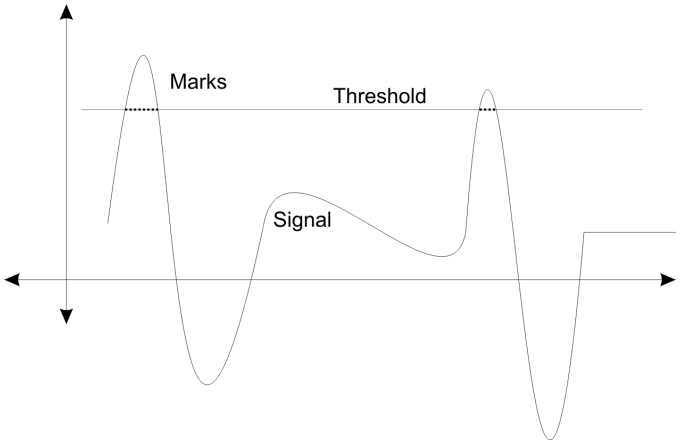


Image 32: The detection of heart beats is done by setting marks (black spots) whenever the threshold is crossed. The spots must satisfy a specific distance between each other and therefore do not sit infinitelly close to each other. In order to detect a beat a previously defined number of marks must stick together. If this number was defined 5 then the second spike would be discarded as noise but if this number was defined to be 3 then both spikes would be identified as heart beats.

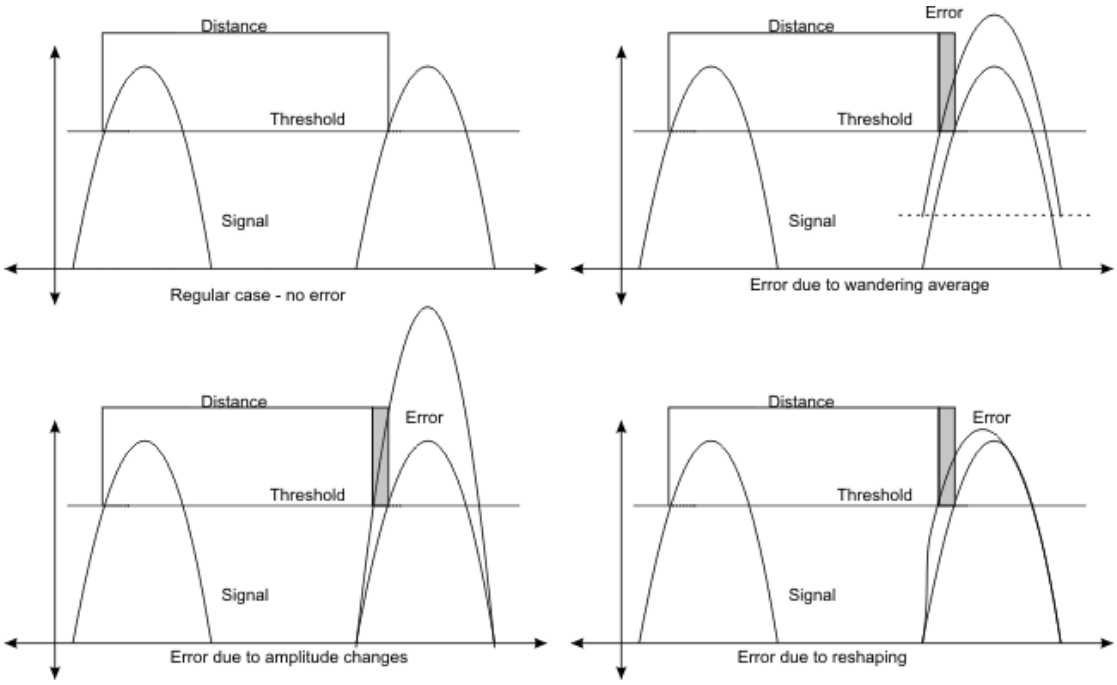


Image 33: Parasite effect causing fake changes in heart rate

The problem with the threshold is that deformations, shifts in signal average or changes of amplitude do influence the distance between the intersections between the signal and the threshold (image 11). Because heart rate is calculated from those distances some heart rate changes recognized until then might have been fake and only due to the parasite effects named before. Before trying to find a better method it is important to understand the influence of errors on the results. If those errors are small then the better results might not be worth the effort of improving the heart beat detection mechanisms.

3.10.4 Tolerance on Beat Recognition Marks

The heart rate cannot be assessed with infinite precision with aliased data. The sampled format has an uncertainty of 1 sample just by the nature of digitalization.

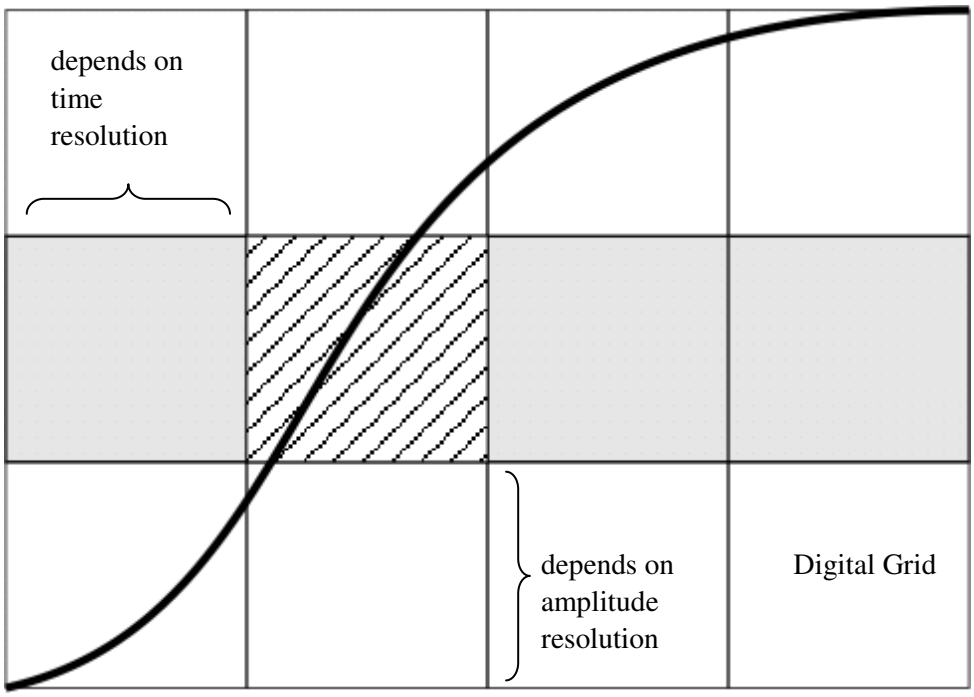


Image 34: The grey line is the threshold. Depending on the horizontal activation function the hatched square will be marked or not, but after threshold it cannot be investigated where the original function was going. Because the line could have crossed the hatched square full left or full right as well, the minimal uncertainty of the difference is 1.

The effects of a detection error of 1 sample can be seen in the following table where sampling frequency is 1.01 kHz and typical heart rates of a rat are 505 BPM to 374 BPM:

Table 3: Table of measured sample distances and the effect of the mismeasurement of the distance by 1 sample.

Tolerance of the method used to detect beats:

1 Samples

Length	BPM	BPM Tolerance	Required change for significance: Sig. Inc/Dec
120	505	4	0.8%
122	497	4	0.8%

124	489	4		0.8%
126	481	4		0.8%
128	473	4		0.8%
130	466	4		0.8%
132	459	3		0.8%
134	452	3		0.7%
136	446	3		0.7%
138	439	3		0.7%
140	433	3		0.7%
142	427	3		0.7%
144	421	3		0.7%
146	415	3		0.7%
148	410	3		0.7%
150	404	3		0.7%
152	399	3		0.7%
154	394	3		0.6%
156	389	2		0.6%
158	384	2		0.6%
160	379	2		0.6%

A beat detection by a threshold assumes, though, that the signal comes regularly at the same strength and shape, which is truly not the case and a short excerpt from the rat16 data clearly shows this:

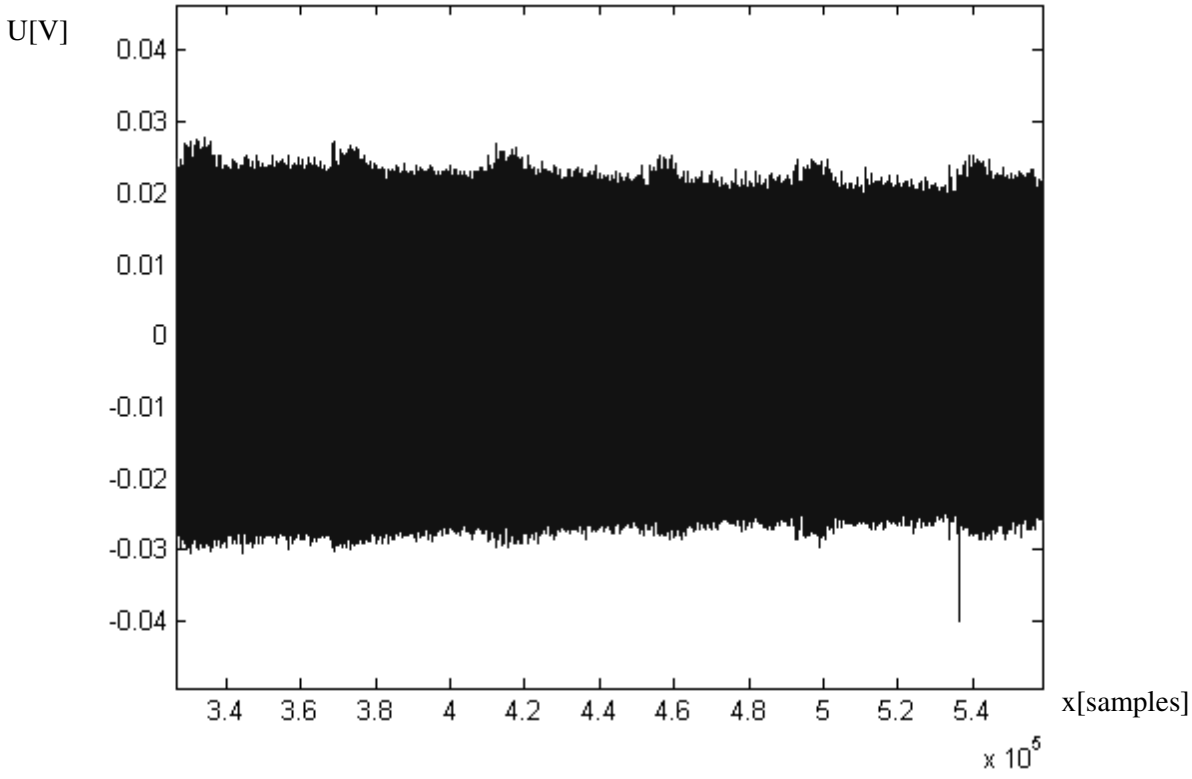


Image 35: Even in this good data one can clearly recognize the amplitude changes during VNS in the ECG.

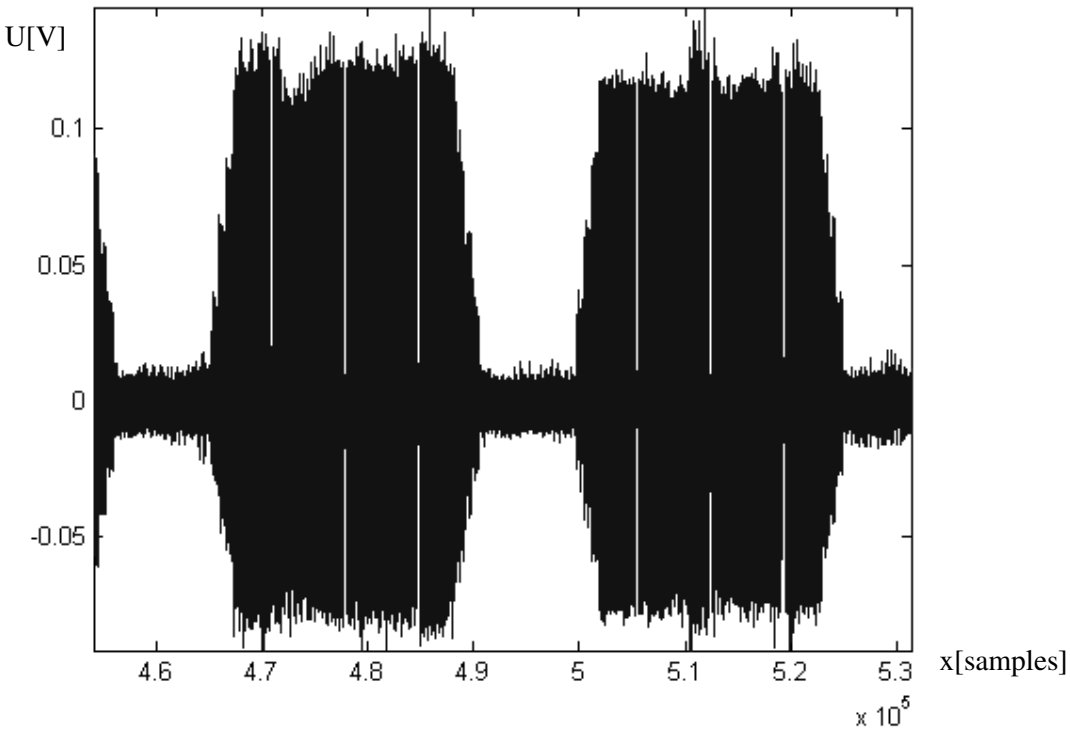


Image 36: In bad cases VNS can cause ECG to go into extreme values. Such effects, as it'll be shown later can account for frequency shifts either over exaggerating them or annihilating them.

Deformations, low frequency noise and shape blowups can account for additional inaccuracy or even systematic error in heart rate. In order to demonstrate these effects a test has been developed in order to estimate precision of the threshold method.

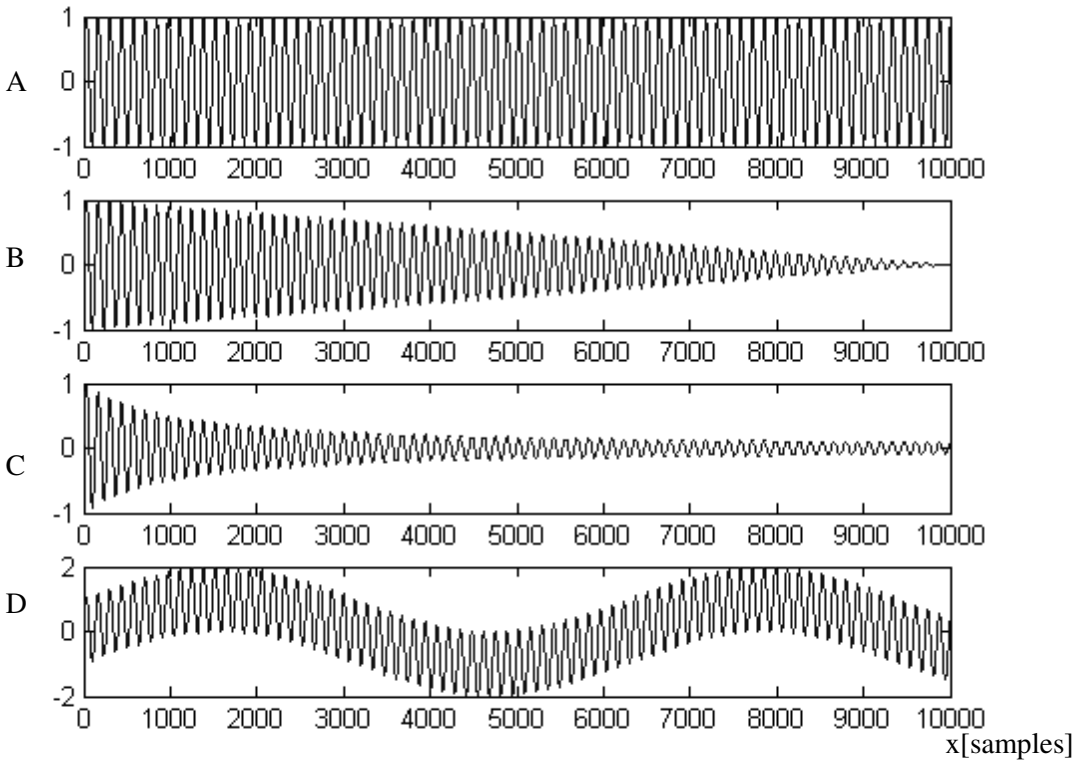


Image 37: The sine wave represents a 449 BPM heart rate at 1010.1 Hz sampling rate. Total time is roughly 10 s.

The first case is a very regular sine wave where each sine represents a single heart beat. The example is constructed at a rate of 449 BPM. The other three cases are examples of malicious deformation through deamplification and low frequency noise. The ramps of the curves were created in Matlab in the following manner:

```
FRQ  = 1010.1;   %sampling rate
Time = 10;       %Time frame
HR   = 466;      %per minute
Pulses = 466/60; %per second

for i=1:Time*FRQ %generate 10s of diagramm
    Test(i,1) = sin( i*2*pi*Pulses/FRQ ); %regular signal (A)
    Test(i,2) = ( 1-(0.0001*i))*Test(i,1); %linear decreasing (B)
    Test(i,3) = ( 2000 / ( 1000+i))*Test(i,1); %nonlinear dec.(C)
    Test(i,4)= sin(i*0.001)*0.3+Test(i,1); %wobble (D)
end;

R=[];
Threshold=0.5;

for n=1:4
    Last = 1;
    f = 0;
    t = InRange( Threshold, Test(:,n), 1 );
    for i=1:length(t)-1
        if t(i)== Threshold & t(i+1)>Threshold
            dT = (i+1-Last)/FRQ;
            f=60/dT;
            Last = i+1;
        end;
        R(i,n)=f;
    end;
end;
```

After thresholding at 0.5 (50%) the heart rate was calculated this resulted in the following heart rate flow:

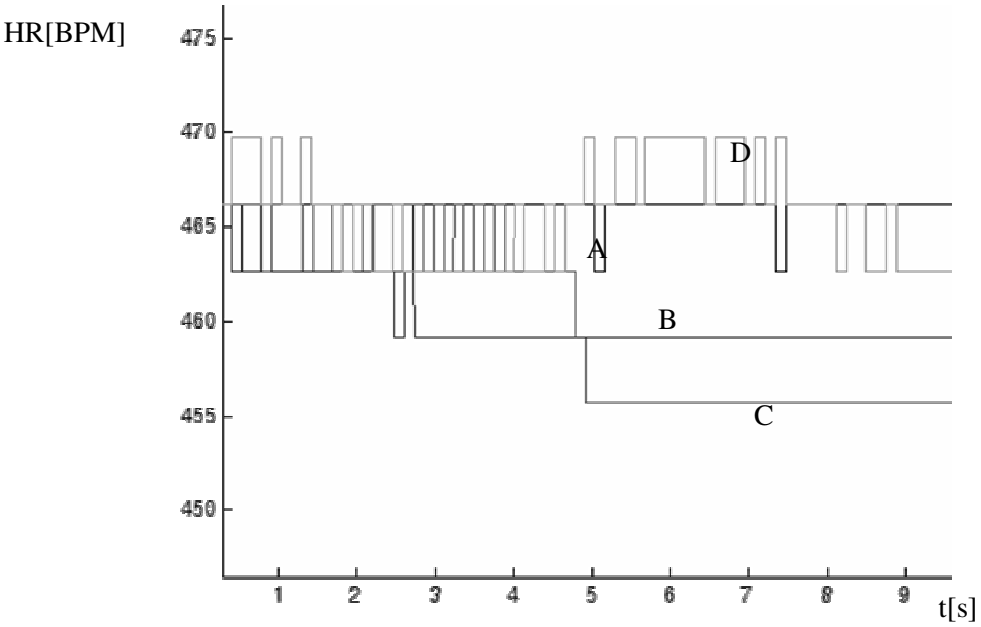


Image 38: Variations in detected frequency can be caused by amplitude changes alone.

The minimum of the test was 455 BPM and the maximum was 470 BPM. This means, those amplitude variations alone can cause an error of as much as 2.4%. This again has significant implication on the interpretation of the results. Changes in heart rate deviations of less than that value can not be taken into account as evidence for any special HR feature.

If the method is applied to a different threshold levels then it can be noticed that this method becomes ever more vulnerable to amplitude changes the higher the threshold.

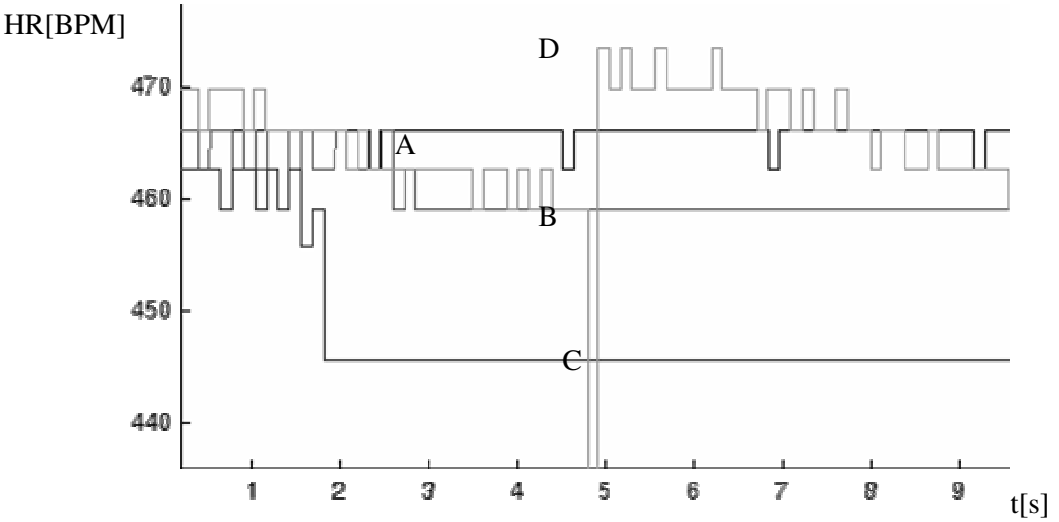


Image 39: Threshold at 70% - Measurements are between 445BPM and 475BPM – max. error is 4.5%

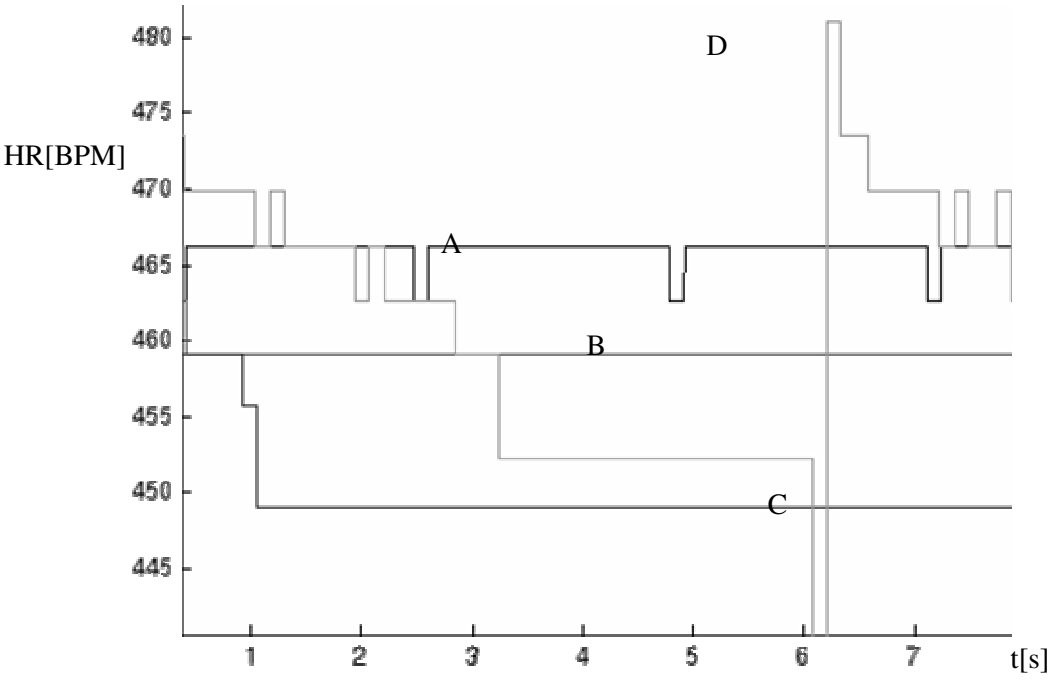


Image 40: Threshold at 95% - Measurements are between 448BPM and 482BPM – max. error is 3.8%

3.10.5 Adaptive Detection with two Fuzzy Criteria

Since the precision of the threshold is not sufficient to claim a rise or fall in heart rate frequency a new method had to be developed. Please note that measured changes in heart rate assessed with any threshold often lie within the range of the error!

A dual indicator operator provides the solution (Beat3.m):

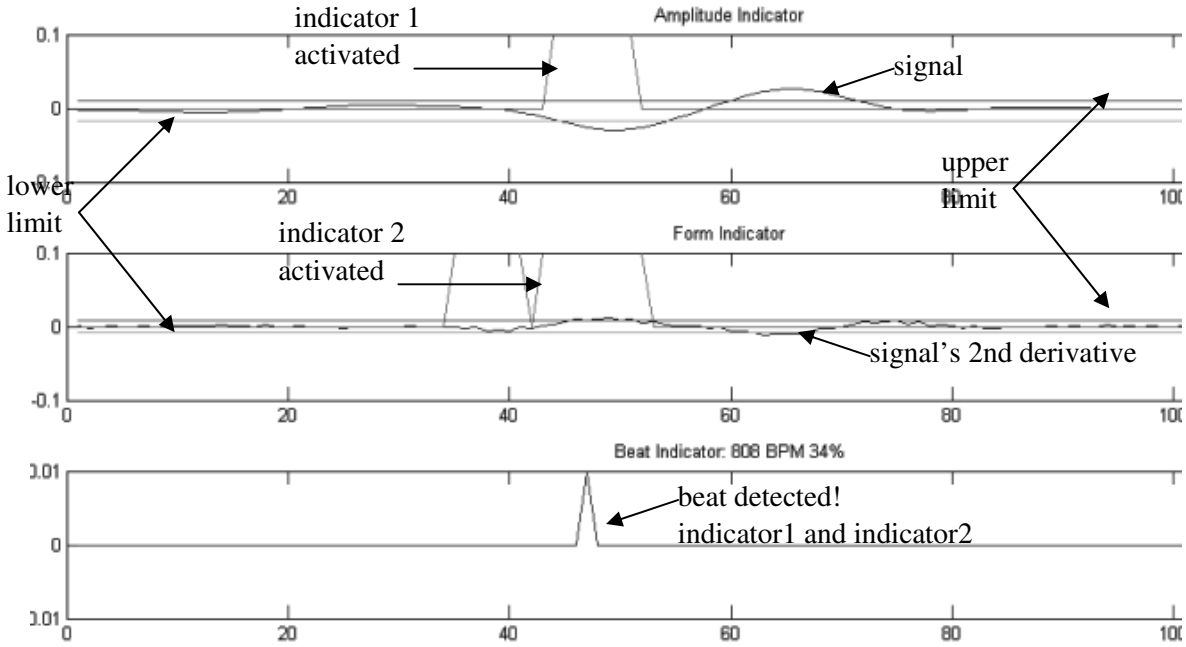


Image 41: Heart beat was recognized because of exceptional amplitude and exceptional 2nd degree derivative.

The ECG is the signal line in the in the topmost diagram. The indicator 1 line indicates significant amplitudes that cannot be considered a norm amount of noise. The parallel limiter lines mark the upper and the lower limit for what is considered normal noise. These lines are adapted repetitively during the process. The middle diagram consists of the 2nd derivation of the ECG shape and is called the shape indicator. The 'beat's shape' is indicated with the indicator 2 line in the middle diagram when it's up. Again, the two limiter lines mark the normal amount of noise. When the 2nd derivation exceeds one of these limits it sets the form indicator (indicator 2) up. From long observations it was taken for granted that a raise in amplitude AND a raise in curvature signalize a heart's beat. And indeed the heart beat can be detected by combining the two indicators together. This method has shown relatively robust against noise. Nevertheless, the heart beat signals are sometimes very close to the noise signal. Improvement was achieved by adding some fuzziness to the limits. The detection of a feature is depending on an Activity variable. This variable exists within a real range of value 10 and 60. These limits were set base upon experience. The Activity variable adapts to the noise level over time. A situation specific score is calculated from a data window and the limits. If this score is greater than the Activity value, then the appropriate indicator is set.

"Initial activity level" $A := 50$ (27)

"Initial upper limit" $U := 0.01$ (28)

„Initial lower limit“ $$L := -0.1 \tag{29}$$

„Adapt 5 times faster to lower noise“ $$\begin{aligned} a^+ &= 0.001 \\ a^- &= 0.005 \end{aligned} \tag{30}$$

The constants a^+ and a^- are adapted to sampling frequency.

„n samples wide window“ $$W(1..n) = D(i...i+n) \tag{31}$$

„m heart rate velocities“ $$H \subset \mathbb{R}^m \tag{32}$$

H is also called History. It contains as many entries as heart beats were detected. Every beat detection increases m.

„Indicates sample of last heart detection“ $$j \in \mathbb{Z} \tag{33}$$

„Minimal distance between beats“ $$B = 100 \tag{34}$$

„Used sampling frequency“ $$Frq = 1010.1Hz \tag{35}$$

Algorithm:

- Get window (31)
- Then adapt the limits:

$$\begin{aligned} U_i &= U_{i-1} + a^+ \cdot \left(\max(W) - U_i\right) \mid \max(W) \geq U_i \\ U_i &= U_{i-1} + a^- \cdot \left(\max(W) - U_i\right) \mid \max(W) < U_i \end{aligned} \tag{36}$$

$$\begin{aligned} L_i &= L_{i-1} + a^- \cdot \left(\min(W) - L_i\right) \mid \min(W) \geq L_i \\ L_i &= L_{i-1} + a^+ \cdot \left(\min(W) - L_i\right) \mid \min(W) < L_i \end{aligned} \tag{37}$$

- Calculate limit crossing indicator Z:

$$\begin{aligned} I_{0..5} &= 0 \\ I_0 &= 0.1 \mid W(n) \cdot 0.1 > U \vee W(n) \cdot 0.1 < L \\ I_1 &= 0.2 \mid W(n) \cdot 0.25 > U \vee W(n) \cdot 0.25 < L \\ I_2 &= 0.5 \mid W(n) \cdot 0.5 > U \vee W(n) \cdot 0.5 < L \\ I_3 &= 0.9 \mid W(n) \cdot 0.75 > U \vee W(n) \cdot 0.75 < L \\ I_4 &= 1 \mid W(n) \cdot 1 > U \vee W(n) \cdot 1 < L \\ I_5 &= 10 \mid W(n) \cdot 2 > U \vee W(n) \cdot 2 < L \\ Z &= \sum_{i=0}^{5} I_i \end{aligned} \tag{38}$$

- Calculate new activity level:

$$d = 100 \cdot \frac{H(i)}{\frac{1}{h}\sum_{x=1}^{h} H(i)} \tag{39}$$

$$\begin{aligned} A_i &= A_{i-1} \mid d \le 10 \wedge d \ge -20 \\ A_i &= A_{i-1} + 2 \mid d > 10 \\ A_i &= A_{i-1} - 2 \mid d < -20 \\ A_i &= A_{i-1} - 5 \mid d < -40 \end{aligned} \tag{40}$$

- Set indicator x (1 or 2) to true, if Z exceeds activity A:

$$Ind_x(i) = true \mid Z_i > A_i \tag{41}$$

- If Ind_1 and Ind_2 and i-j>B then calculate new heart rate and add this value to history H.

$$H(m+1) = \frac{60s \cdot Frq}{i - j} \tag{42}$$

This way even much distorted beats which are hard to detect are being detected. Beats can be detected that sometimes have unusual shape and do not cross the lower or upper limit clearly. When signal becomes too chaotic, of course this doesn't work either. In order to demonstrate the superiority of the new method it was applied to the same signals from image 33.

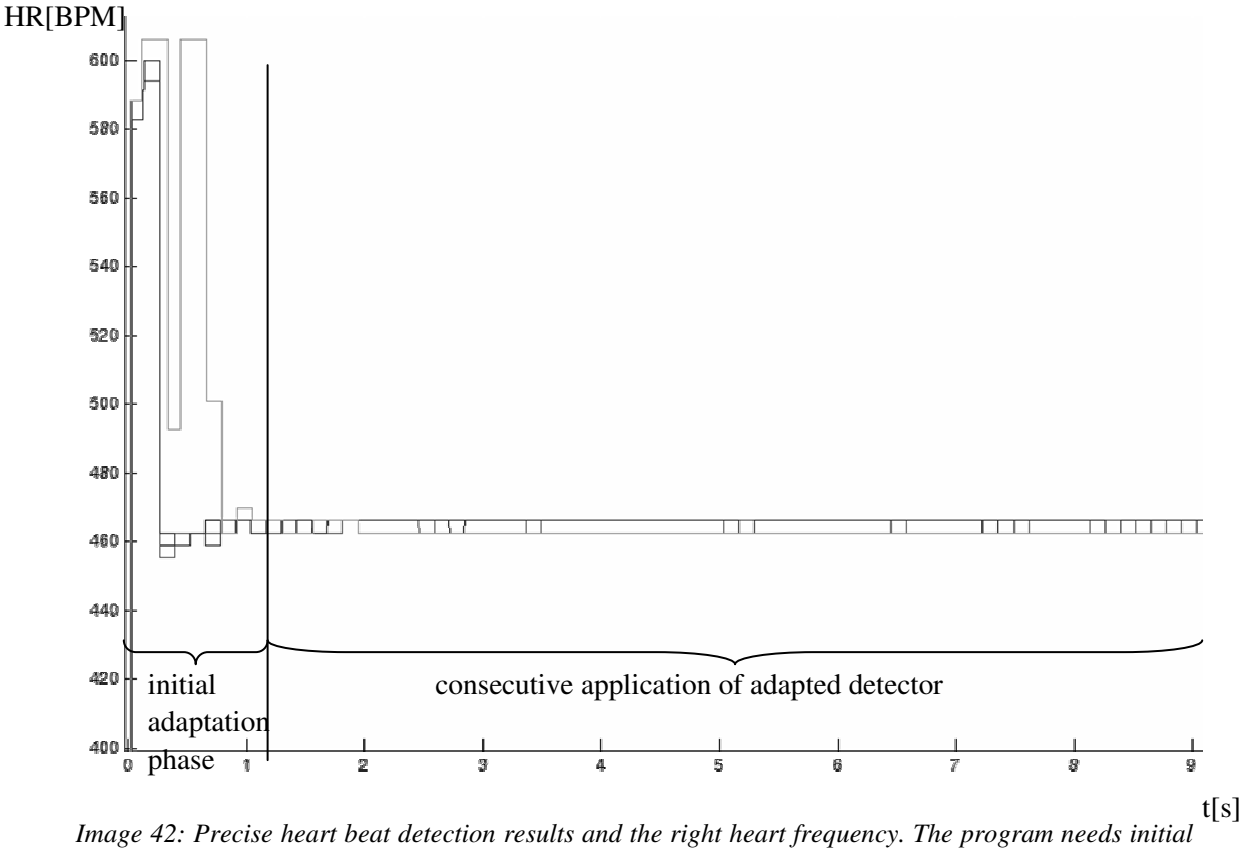


Image 42: Precise heart beat detection results and the right heart frequency. The program needs initial time to adapt before it produces correct results.

3.10.6 Conclusion

The multi criteria method for heart beat (and ventilation rate) detection is sufficiently good and reliable and hence the heart rate curves can be trusted. Additionally it delivers heart rate in BPM and deviations from normal in percent. Observations made for frequency changes become relevant with this method. Finally, let compare the BPM curves from the ecg.txt which was retrieved conventionally and the BPM curves created by the program Beat3.m [24pendix B]:

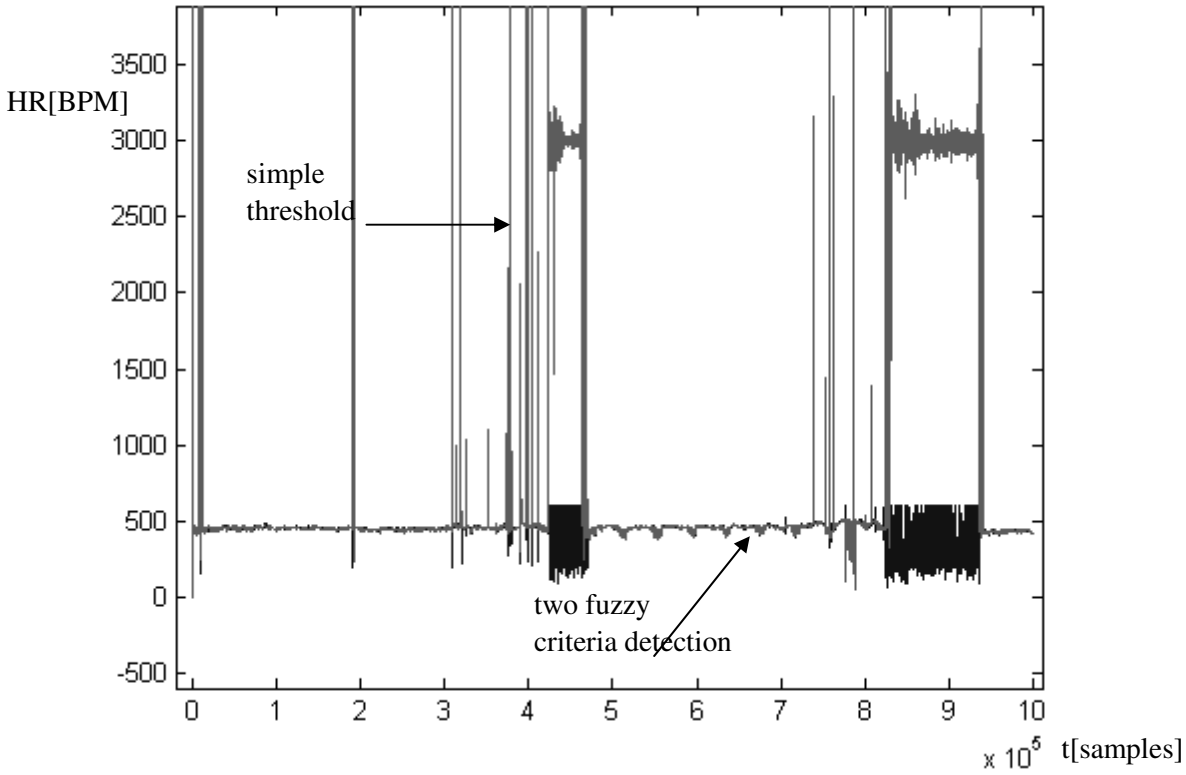


Image 43: compare of new and original method. The two fuzzy criteria detector detects heart beats in places where the threshold is not capable of doing so.

The good results of ecg.txt are obviously founded in a more complex method than the threshold method alone. Still, the line from the new heart rate classifier does not suffer from erroneous peaks. The previous program onlyforH.m relies on peak marks given by the Spike2 program. The heart rate from this peak file had been reconstructed (green line). This means that the new method used for heart rate generation can be trusted on and performs comparatively or better than the methods that were used before.

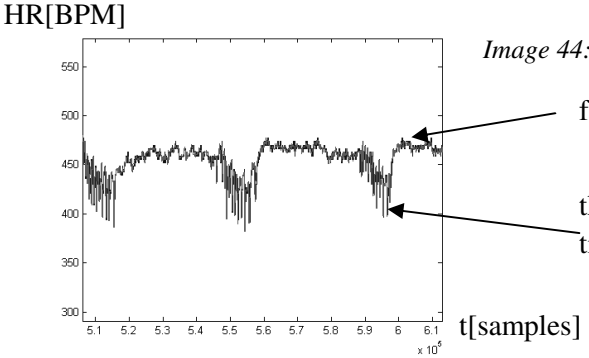


Image 44: The fuzzy method is smoother!

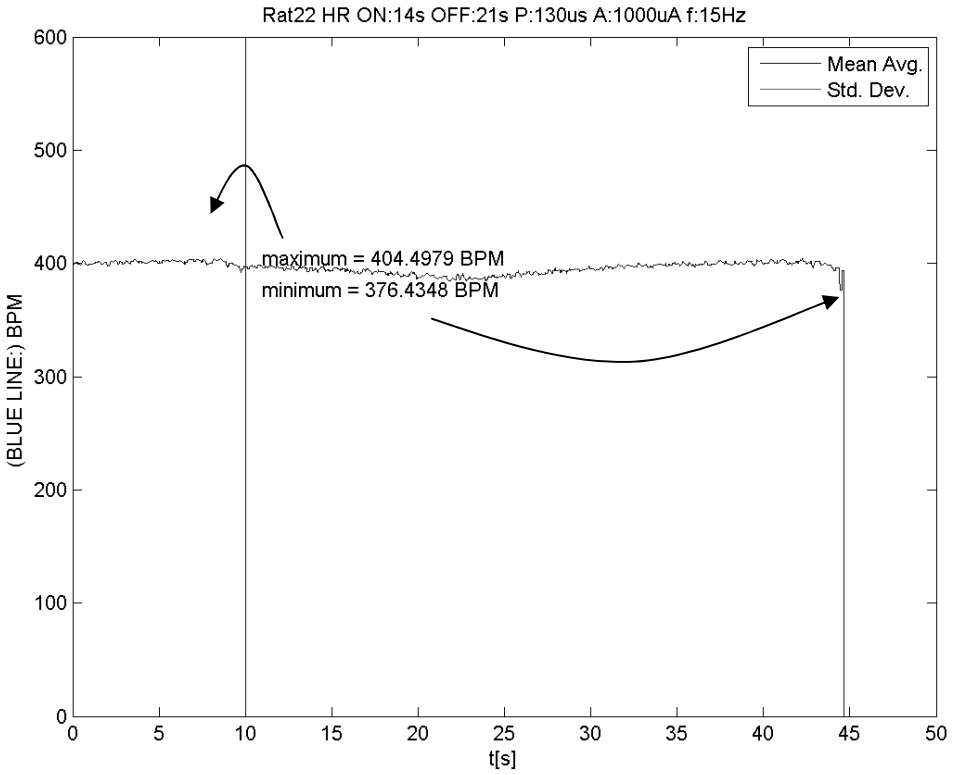


Image 45: Stimulation starts at vertical line. Even a faint decrease in heart rate can be observed.

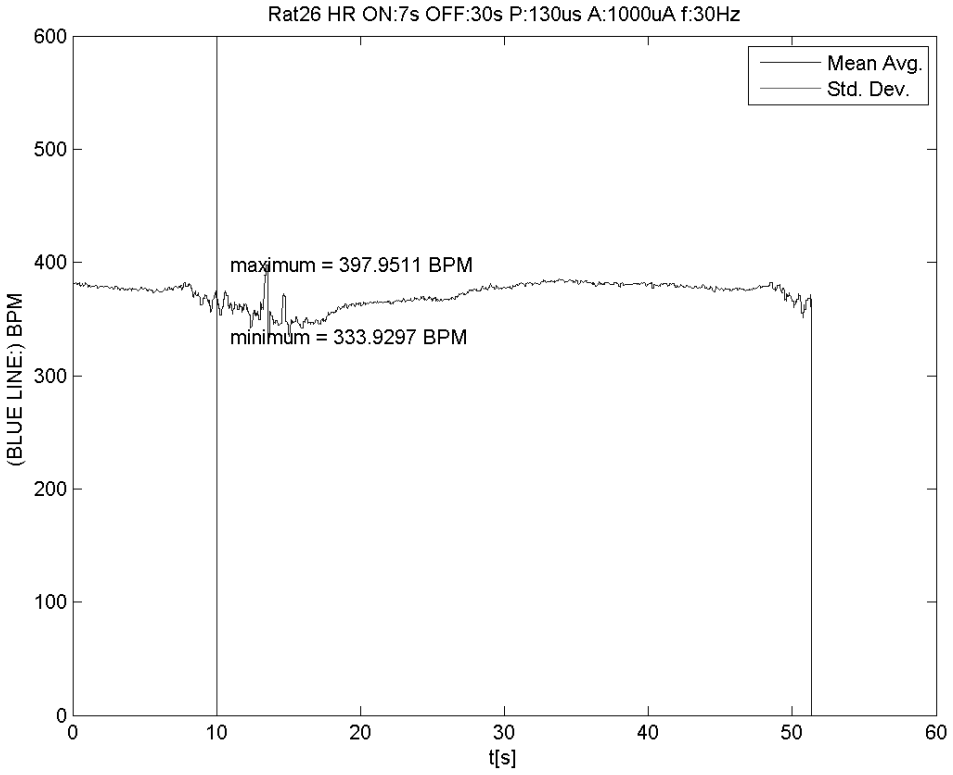


Image 46: The heart detection algorithm detects even very rapid changes in heart rate.

3.11 Feature and Parameter Association

The data shown in chapter 2.5 seems to be associated with a specific group of parameters of the VNS. A possible method to perform such association is to sort stimulations by the value of its features and to divide the associated parameters into groups. These groups can be compared in the sense of what parameter values seem to dominate the group. This simple method would convey relatively simple parameter-to-feature association, provided that such simple relationship exists.

The first problem to solve is that any stimulation cycle consists of many feature values which are irrelevant. Relevant are maximum-effect descriptions during one of the three phases of any stimulation cycle.

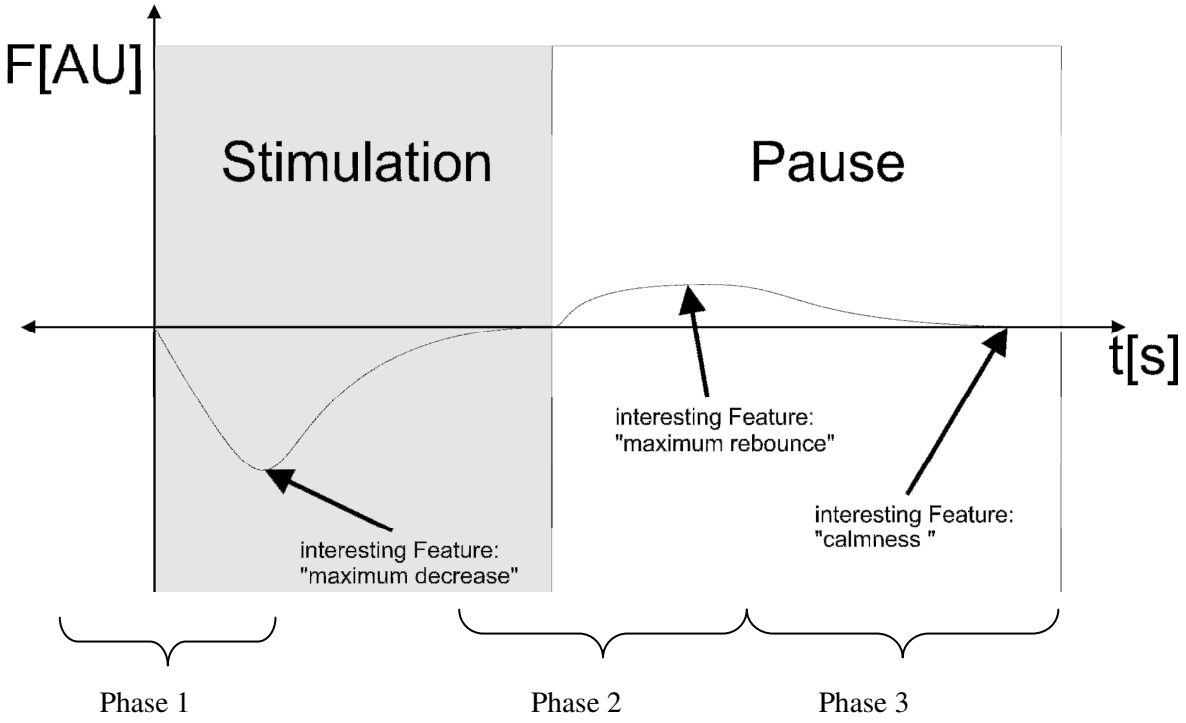


Image 47: Each phase has one specific value of interest which can be expressed as the maximum absolute value during that phase.

The three phases were defined as such:

Phase 1: -80% to 5% of ON+OFF

Phase 1 is the time frame where maximum effects occur caused by VNS.

Phase 2: 15% to 40% of ON+OFF

Phase 2 is the time frame where rebound effects occur.

Phase 3: 40% to 70% of ON+OFF

Phase 3 is the time frame where organism can reach calm state again.

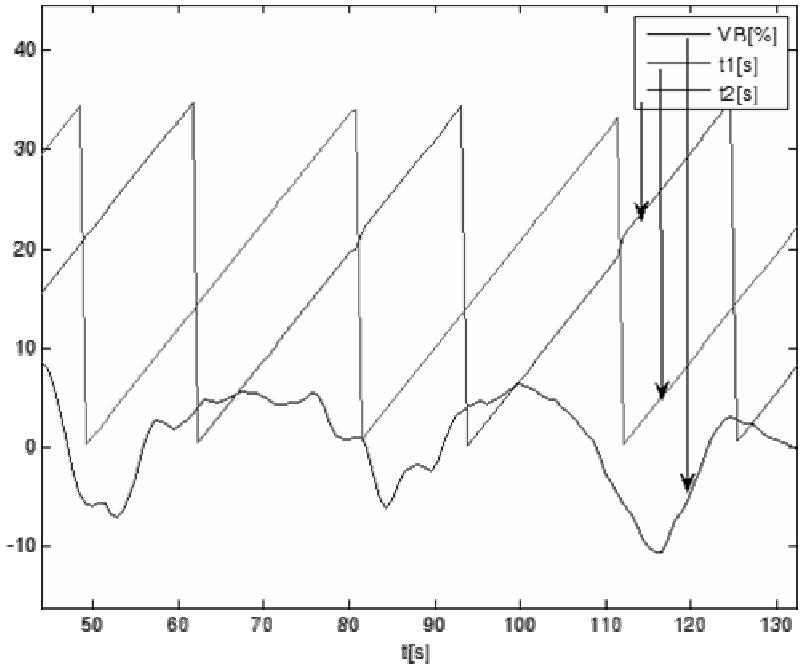


Image 48: This is a real data plot showing the best case situation where t1 and t2 are the timers indicating time since start or stop of the stimulation. VR is the ventilation rate increase/decrease in percent. In worse cases the timers jump because of cut joints or because of failed VNS detection and VR sometimes never becomes positive during a cycle.

The complete data is sectioned into phases. Each phase is reduced to a single value. This value is the maximal absolute value from the section. The variable N contains the number of samples associated with a Phase $P_{1,2,3}$ which is a vector of values, essentially.

„Phase Data Vector“ $$\vec{P}_x \in \mathbb{R}^N \tag{43}$$

„Phase Value“ $$v = \max\left(\left|\vec{P}_x\right|\right) \cdot sign\left(\vec{P}_x\left(\max pos\left(\vec{P}_x\right)\right)\right) \tag{44}$$

The result of this process is three matrixes $\overset{\#}{M}_{1..3}$ of data containing a tuple of stimulator parameters and the associated phase value. Happily, this method brings a large data reduction without changing the overall look of the data. This is important because it was shown in chapter 2.7, that the diagram shows repeating features indicating that parameters can be associated to specific features.

$$\overset{\#}{M} = \begin{bmatrix} VR_1 & VR_2 & \dots & VR_m \\ VA_1 & VA_2 & \dots & VA_m \\ VS_1 & VS_2 & \dots & VS_m \\ VI_1 & VI_2 & \dots & VI_m \\ VX_1 & VX_2 & \dots & VX_m \\ VT_1 & VT_2 & \dots & VT_m \\ HR_1 & HR_2 & \dots & HR_m \\ ON_1 & ON_2 & \dots & ON_m \\ OFF_1 & OFF_2 & \dots & OFF_m \\ P_1 & P_2 & \dots & P_m \\ A_1 & A_2 & \dots & A_m \\ F_1 & F_2 & \dots & F_m \end{bmatrix} \tag{45}$$

Note, that the compressed data in $\overset{\#}{M}_{1..3}$ does not contain timers anymore.

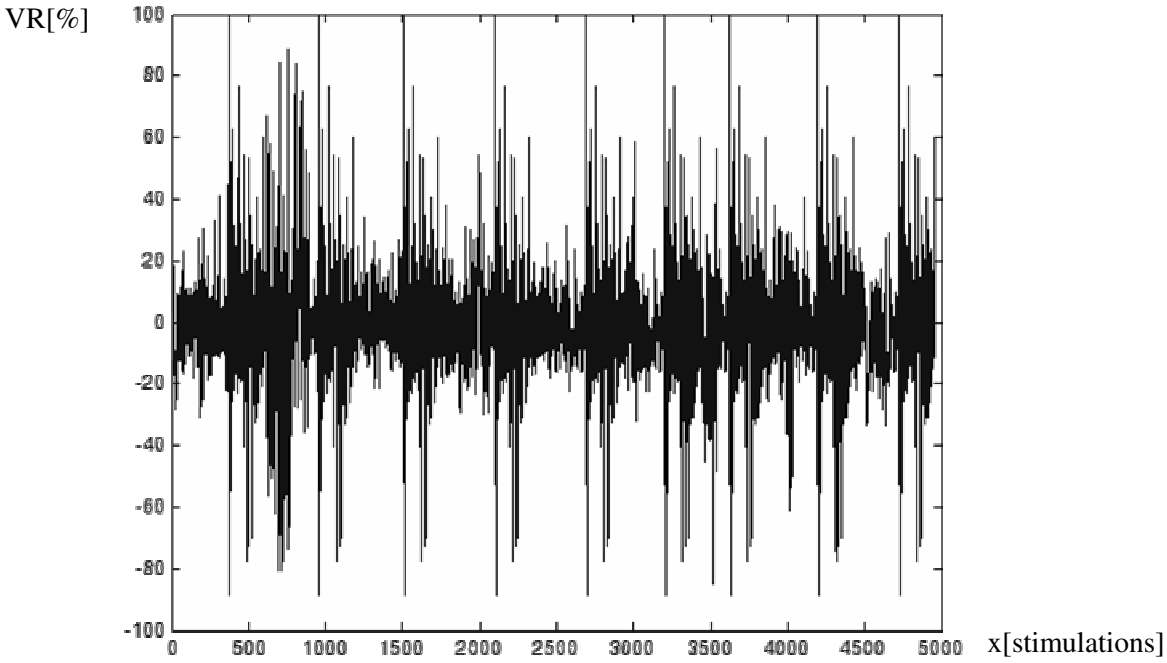


Image 49:Phase 1 data: After the separation of the UniRat data to the three phases and after the reduction to a single value per stimulation the repeating features still remained.

In a compactation step uniqueness of parameters must be achieved. Right until now $\overset{\#}{M}_{1..3}$ contain 9xN[1] entries for each parameter vector used in the recording protocol. The effects are mean averaged which leads to a new reduced data set $K_{1..3}$ containing less than hundred entries per phase. This is an amount of data which can be handled really well.

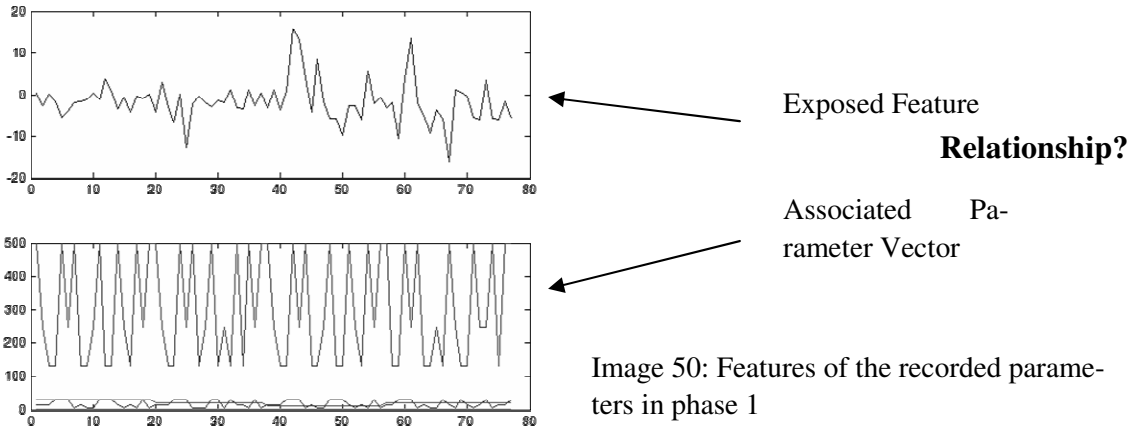


Image 50: Features of the recorded parameters in phase 1

[1] If we multiply 9 rats with N=7.153 stimulations/per vector then it leads us to a stimulation redundancy factor of 64.3766. If we take the 4957 stimulation values and divide them by this factor, then we obtain the 77 entries for phase 1 after compactation.

3.11.1 Visualization of compacted data

One way to extract information about the dependencies between the different parameters and the physiological response is a way to visualize the data set. With original UniRat data this is quite difficult. But with the compacted data set all kinds of visualizations are possible. A non standard Matlab function from the [33] with the name gridfit() was used for visualization. [56]

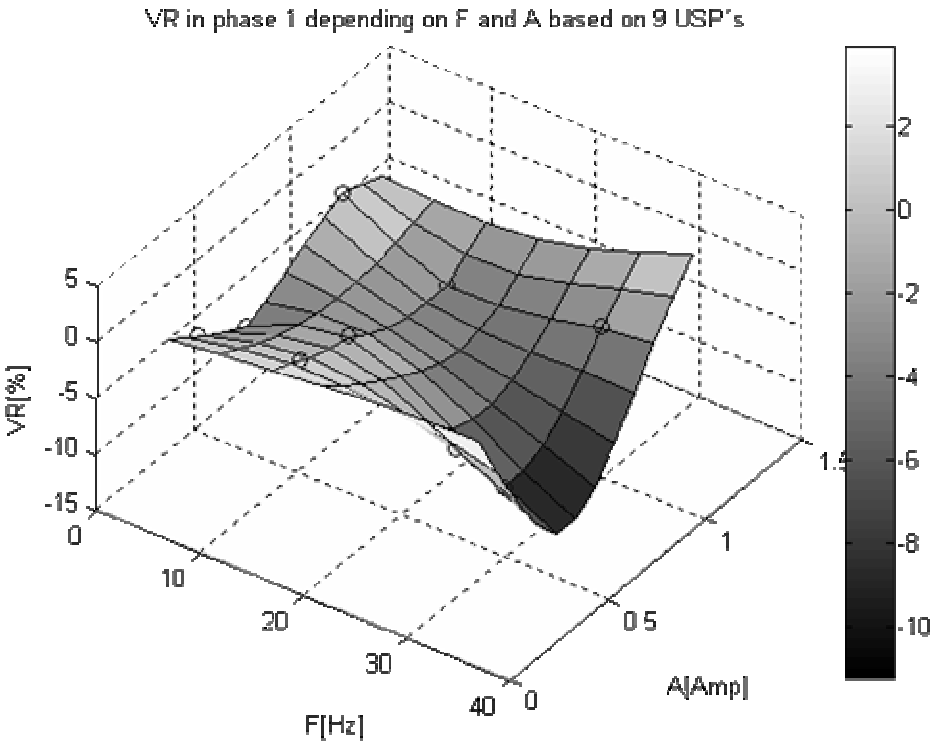


Image 51: statistical ventilation rate changes depending on A and F as obtained by a surface fitting mechanism. Surface is supported by 9 unique vectors.

The surface in image 50 is supported by 77 vectors where many of them fall into the same position in the diagram. It is one of the outstanding features of the gridfit() function to do an interpolation between those vectors.

Table 4: Example of how data is used for the generation of the diagrams.

VR	VA	VS	VI	VX	VT	HR	ON	OFF	P	A	F
0,48	0,67	8,24	1,97	2,70	5,24	0,78	7,00	30,00	500,00	1,00	15,00
-2,72	-12,70	-3,04	7,56	3,75	-0,98	-0,61	7,00	30,00	250,00	0,50	15,00
0,07	9,17	-3,12	-2,66	3,38	-1,63	-0,10	7,00	30,00	130,00	1,00	15,00
-1,64	-9,93	-1,41	10,28	5,09	0,27	0,08	7,00	30,00	130,00	1,00	30,00
-5,45	13,00	-6,84	-0,86	15,48	-3,77	-0,47	7,00	30,00	500,00	1,00	30,00
-3,86	13,72	-1,43	0,49	15,94	-2,00	-0,36	7,00	30,00	250,00	0,50	30,00
-1,82	-11,88	-5,20	5,70	13,23	-2,86	-0,26	7,00	30,00	500,00	0,50	5,00
-1,75	-11,27	-6,07	-5,96	4,01	-3,14	0,31	7,00	30,00	130,00	0,25	15,00
-1,17	-4,58	0,98	0,89	2,68	0,11	-0,10	7,00	30,00	130,00	0,50	5,00
0,27	-0,45	-2,65	-3,31	5,27	-2,69	0,01	7,00	30,00	250,00	1,00	5,00
-1,03	8,88	-1,62	-3,21	6,72	-1,78	0,28	7,00	30,00	500,00	0,25	30,00
3,59	5,65	2,97	-7,93	4,31	-0,82	-0,18	7,00	30,00	130,00	0,50	30,00
1,01	10,68	-2,18	0,82	-2,21	1,62	0,08	7,00	30,00	130,00	0,25	30,00
-3,32	14,02	0,74	-0,75	8,46	1,54	-0,34	7,00	30,00	500,00	0,25	15,00

-0,63	0,55	-1,01	-0,49	6,29	-0,51	0,02	7,00	30,00	250,00	0,25	5,00
-4,22	-5,43	-0,43	-2,32	6,87	-2,52	-0,23	7,00	30,00	130,00	0,50	15,00
-0,43	5,14	0,22	-2,00	4,16	-1,04	0,09	7,00	30,00	500,00	1,00	5,00
-0,76	-0,60	-4,43	-1,24	1,45	0,39	0,19	7,00	30,00	250,00	0,25	30,00
0,05	1,67	12,76	-4,63	-1,45	15,17	0,16	7,00	30,00	500,00	0,25	5,00
-4,21	-17,88	0,21	17,71	-1,70	1,25	-0,35	14,00	21,00	500,00	1,00	15,00
2,87	-18,72	-6,74	13,43	-4,22	-0,50	-0,14	14,00	21,00	250,00	0,50	15,00
-2,92	-24,93	-5,65	18,25	1,36	-1,40	-0,65	14,00	21,00	130,00	1,00	15,00
-6,82	-29,11	-7,55	20,52	4,05	-2,77	0,27	14,00	21,00	130,00	1,00	30,00
0,06	-23,42	-6,13	14,30	1,48	-2,19	0,13	14,00	21,00	500,00	1,00	30,00
-12,71	-39,29	-18,72	15,45	13,26	-6,43	0,71	14,00	21,00	250,00	0,50	30,00
-2,27	-39,05	-11,99	24,17	0,11	-6,10	0,50	14,00	21,00	500,00	0,50	5,00
-0,43	-4,56	-1,34	5,41	1,20	-0,43	-0,11	14,00	21,00	130,00	0,50	5,00
-1,88	-18,14	-3,39	11,48	-1,97	-0,32	-0,02	14,00	21,00	250,00	1,00	5,00
-3,00	-16,32	-5,26	13,12	-0,20	-2,79	0,35	14,00	21,00	500,00	0,25	30,00
-1,30	-12,81	-3,78	8,82	0,49	-1,45	0,18	14,00	21,00	130,00	0,50	30,00
...	...	...	...	...	...	...	...	...	...	...	...

1 feature selected

6 features omitted

3 omitted columns

2 selected columns

The number of unique supporting vectors for a diagram is an indicator for the quality of prediction and could be used to weigh the result in the segmentation process described in chapter 3.15.1.

```
size( Unique([Compact1(:,12) Compact1(:,14) ]), 2 )
ans =
     9
```

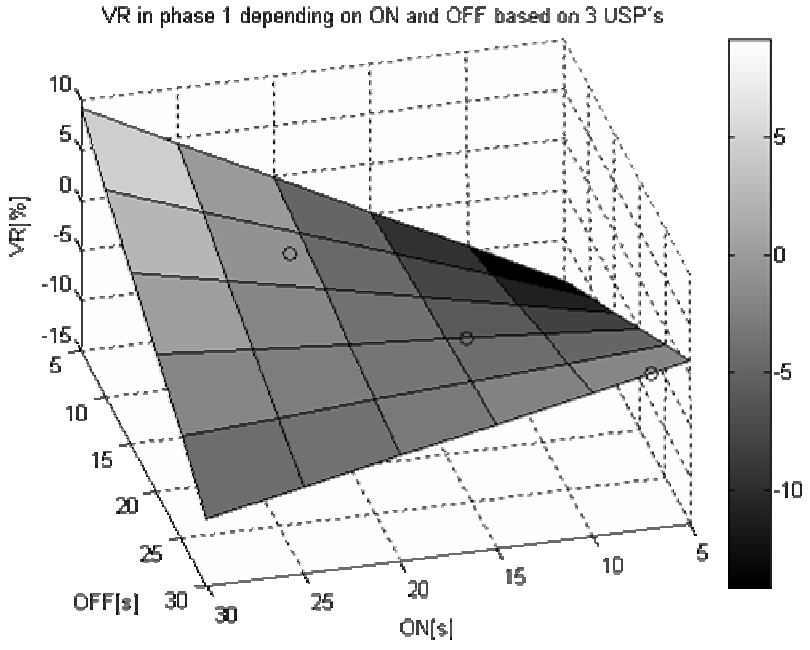


Image 52: statistical ventilation rate changes depending on TON and TOFF as obtained by a surface fitting mechanism. Supported by 4 unique vectors.

3.11.2 Template Cluster Vector

The data obtained this far can be divided up into at least three ranges of interest ‚increase', ‚constant' and ‚decrease'. These three groups may have some distinct average parameter vectors that could be utilized as classification templates. The use of them is simple as it would be enough to find the closest template to an unknown vector and assign a physical response of the template to the questioned vector. The simplest idea is to use the geometric distance metric d_2 which would divide the parameter space into a Voronoi Diagram [34].

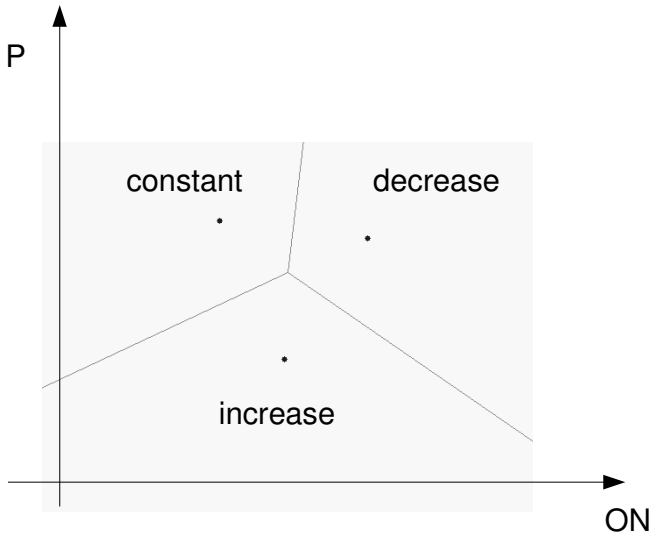


Image 53: Idea of this method is to find well separable templates for classification.

Example: Data $K_{1..3}$ is separated into three sets. After separation into subsets $G_{j,x}$ the templates are extracted and assessed for their usefulness.

record k

$$\vec{k}_{i,x} \in \overset{\#}{K}_x, \overset{\#}{G}_x \subset \mathbb{R}^{14} \tag{46}$$

$$\overset{\#}{K}_x = \begin{bmatrix} k_{1,x,1} & k_{1,x,\dots} & k_{1,x,14} \\ k_{\dots,x,1} & k_{\dots,x,\dots} & k_{\dots,x,14} \\ k_{n,x,1} & k_{n,x,\dots} & k_{n,1,14} \end{bmatrix} = \begin{bmatrix} \vec{k}_{1,x} \\ \vec{k}_{\dots,x} \\ \vec{k}_{n,x} \end{bmatrix} \tag{47}$$

feature extractor from vector k

$$Feature\left(\vec{k}_{j,x}\right) \subset \mathbb{R} \tag{48}$$

decrease

$$\overset{\#}{G}_{D,x} \subset \overset{\#}{K}_x \mid Feature(\vec{k}_{i,x}) < -1 \tag{49}$$

constant

$$\overset{\#}{G}_{C,x} \subset \overset{\#}{K}_x \mid Feature(\vec{k}_{i,x}) > -1 \wedge Feature(\vec{k}_{i,x}) < +1 \tag{50}$$

increase

$$\overset{\#}{G}_{I,x} \subset \overset{\#}{K}_x \mid Feature(\vec{k}_{i,x}) > +1 \tag{51}$$

More generally those sets can be described as:

$$\overset{\#}{G}_{j,x} \subset \overset{\#}{K}_x \quad | Feature(\vec{k}_{i,x}) >= low_{j,x} \wedge Feature(\vec{k}_{i,x}) < high_{j,x} \tag{52}$$

where $low_{j,x}$ and $high_{j,x}$ are range limits for each of the phases x for each of the ranges j.

The parameters T_{ON}(ON) and T_{OFF}(OFF) haven't been recorded in all possible combinations. Therefore those parameters do not have enough importance by themselves, because they are not independent of each other. The following table shows the problem:

Table 5: Availability of time parameters for analysis

T_{ON}	T_{OFF}	T_{ON}/T_{OFF}	DAQ
7	30	0.233	used
7	21	0.333	not available
7	12	0.588	not available
14	30	0.466	not available
14	21	0.666	used
14	12	1.166	not available
21	30	0.7	not available
21	21	1	not available
21	12	1.75	used

It is not the same thing to use T_{ON} and T_{OFF} or the proportion of them both. It is also hard to say which one is better for use in classification. One might argue that because T_{ON} only occurs in connection with specific T_{OFF}, T_{OFF} is redundant. This would clearly counterfeit practically attained experience. In fact, there is no clear answer. From personal point of view the T_{ON}/T_{OFF} proportion is more universal but an unanswerable question is whether it holds true at all scales. A hint might be coming from the limit finding algorithm which is used in order to find the most relevant feature-limits for what can be considered a low, a constant or a high value. The idea behind it is to start of with limits as close to zero as possible and to increase the limit as long as the separation of the template vectors increases. At some larger values the limits cause the template vectors to come closer together because enough of the decrease/increase records are sorted into the constant category.

number of slices $$S = 20 \tag{53}$$

limit, try i for in phase x: $$l_{i,x} = \frac{\max\left(\left|\overset{\#}{K}_x\right|\right) \cdot i}{S} \quad | i \in \{1..S-1\} \subset \mathbb{N} \tag{54}$$

sort out decrease rows: $$\overset{\#}{G}_{D,i,x} \subset \overset{\#}{K}_x \quad | Feature(\vec{k}_{j,x}) < -l_{i,x} \tag{55}$$

separate const: $$\overset{\#}{G}_{C,i,x} \subset \overset{\#}{K}_x \quad | Feature(\vec{k}_{j,x}) >= -l_{i,x} \wedge Feature(\vec{k}_{j,x}) < l_{i,x} \tag{56}$$

sort out increase rows: $$\overset{\#}{G}_{I,i,x} \subset \overset{\#}{K}_x \quad | Feature(\vec{k}_{j,x}) >= l_{i,x} \tag{57}$$

number of elements in $G_{y,i,x}$ $$n_{y,i,x} \in \mathbb{N} \tag{58}$$

template generation $$\vec{g}_{y,i,x} = \frac{1}{n_{y,i,x}} \sum_{j=1}^{n_{y,i,x}} \vec{G}_{y,i,x,j} \quad | \vec{G}_{y,i,x,j} \in \overset{\#}{G}_{y,i,x} \tag{59}$$

there is a distance i in phase x... $$d_{i,x} \in \mathbb{R} \tag{60}$$

...that calculates: $$d_{i,x} = \sum_{y=2}^{4} \left\| \vec{g}_{y,i,x} - \vec{g}_{y-1,i,x} \right\|_2 \quad | \vec{g}_{1,i,x} = \vec{g}_{4,i,x} \tag{61}$$

Equations (61) only holds true for 3 segments because the perimeter in a triangle is the sum of distances between all points. If more than 3 segements are used than please use instead:

$$d_{i,x} = \sum_{b=1}^{h} \sum_{y=b}^{h} \left\| \vec{g}_{y,i,x} - \vec{g}_{b,i,x} \right\|_2$$

Start with i=1 and continue increasing i until $d_{i,x} > d_{i+1,x}$. Use according $g_{1..3,i,x}$ for classification purposes.

In the following tables the dimensionality of K and G has been varied in order to accomplish a fair comparison of T_{ON} and T_{OFF} with T_{ON}/T_{OFF}.

Table 6: This table shows the limit selection algorithm applied to a vector with T_{ON}/T_{OFF}.

i	$l_{i,x}$[%]	$d_{i,x}$
1	0,79	77,34
2	1,58	74,17
3	2,37	71,02
4	3,15	88,96
5	3,94	168,83
6	4,73	113,09
7	5,52	123,76
8	6,31	235,28
9	7,10	202,58
10	7,88	202,58
11	8,67	224,13
12	9,46	175,65
13	10,25	95,05

Table 7: Resulting vectors template vectors for classification of phase 1 in VR cycle.

VR Phase 1	T_{ON}/T_{OFF}	P	A	F
decrease	0,9	228	0,5	22
constant	0,52	328	0,59	13
increase	1,4	333	0,66	30

Table 8: This table shows the limit selection algorithm applied to a vector with T_{ON} and separate T_{OFF}.

i	$l_{i,x}$[%]	$d_{i,x}$
1	0,79	79,64
2	1,58	79,57
3	2,37	76,50
4	3,15	94,05
5	3,94	172,35
6	4,73	122,84
7	5,52	133,43
8	6,31	237,79
9	7,10	205,21
10	7,88	205,21
11	8,67	228,82
12	9,46	180,50
13	10,25	100,39

Table 9: Resulting vectors template vectors for classification of phase 1 in VR cycle.

VR Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	15	19	212	0,58	23
constant	10,7	25	328	0,59	13
increase	19,25	14	282	0,63	24

Table 10: *This table shows the limit selection algorithm applied to a vector with T_{ON} alone.*

i	$l_{i,x}$[%]	$d_{i,x}$
1	0,78	78,20
2	1,57	76,39
3	2,36	73,26
4	3,15	90,99
5	3,94	170,15
6	4,73	117,52
7	5,51	128,16
8	6,30	236,21
9	7,09	203,56
10	7,88	203,56
11	8,67	225,95
12	9,46	177,52
13	10,25	97,11

Table 11: Resulting vectors template vectors for classification of phase 1 in VR cycle.

VR Phase 1	T_{ON}	P	A	F
decrease	15	211	0,58	23
constant	10	328	0,59	13
increase	19	282	0,62	24

It is important to emphasize the difference in results when making a choice to separate times, single time or time proportion.

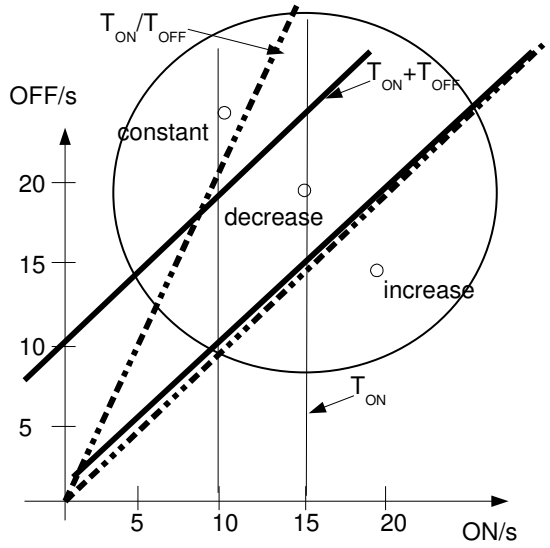


Image 54: Parameter vectors will be classified in different manner when choosing one of the three methods.

If tested vectors are close to the original vectors (circle) the classification seems to be best with T_{on} and T_{off}. Values beyond the circular can also be classified using the proportions method. From personal observation and intuition the difference between T_{on} and T_{off} seems to be the relevant feature to look here for. If ON is at least as long as OFF, then a ventilation rate increase is most probable in phase 1. If ON is half as long as OFF rather a decrease in ventilation rate is to expected. In other cases ON becomes to short and ventilation shows no clear increase or decrease. Please note that all three methods have resulted in a limit of 6.3%. That means that the diagram segments changes of at least 6.3%. Changes below this value are considered constant.

Please note, that the current plays no relevant role in the tables. In all three cases the averages stick together closely. This implicates that current strength is not associated with increases or decreases in ventilation rate in this example.

3.11.3 Learning Vector Quantization (LVQ)

Learning vector quantization (LVQ) is a dual layer network first presented by [35] and [36] which are related to self organizing maps (SOM). LVQ type networks can be used to learn classification functions from a training set that is consisting of an input vector and a classification result [37]. There are two ways of using LVQ networks for the good of the project: either the tables in chapter 4.1 can be taken to set up an LVQ manually, or alternatively, the LVQ can be trained with pre-classified data. For training set classification the separation levels from chapter 4.1 can be used. The pre-classified data can be found in chapter 7.2. Original compacted data is presented in chapter 4.4.

3.11.4 Decision-, Classification- and Regression Trees

Decision Trees (DT) are decision finding structures or expressed in differently: rules for classification. DT's can be designed manually in order to reflect expert knowledge or can be induced. According to the computational learning theory [38, p.668] the creation algorithms for a decision tree are classified as PAC-algorithms (probably approximately correct) that are example based and must be distinguished from explanation based learning. Rule induction in DT's is mainly based on the concept of entropy reduction. Such a task can be performed with algorithms like the famous ID3, C4.5, CN2 or FOIL. [38, p.709]

The most prominently used algorithm is ID3 that works by generating subtrees recursively trying to find the most dominant input attributes that offer improved classification over the case not using them. i.e.: A possible result of such rule induction is that the dominant feature is the T_{ON} time and second important attributes are P, T_{OFF} and F.

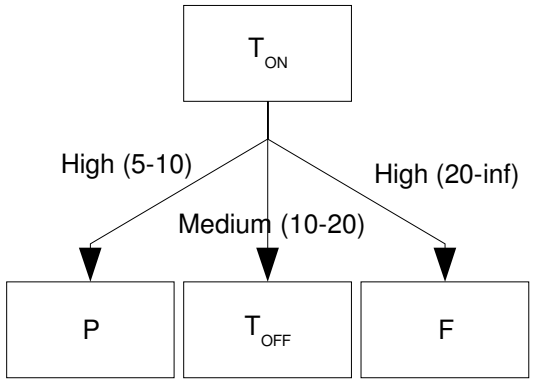


Image 55: Two top level of a sample tree. Note that this is a hypothetical example and cannot be used for practical conclusions.

Table 12: Excerpt from compacted Phase 1 data.

VR[%]	VA[%]	VS[%]	VI[%]	VX[%]	VT[%]	HR[%]	ON[s]	OFF[s]	P[µs]	A[A]	F[Hz]
-4,2	-5,4	-0,4	-2,3	6,9	-2,5	-0,2	7	30	130	0,5	15
-0,4	5,1	0,2	-2,0	4,2	-1,0	0,1	7	30	500	1	5
-0,8	-0,6	-4,4	-1,2	1,5	0,4	0,2	7	30	250	0,25	30
0,0	1,7	12,8	-4,6	-1,4	15,2	0,2	7	30	500	0,25	5
-4,2	-17,9	0,2	17,7	-1,7	1,2	-0,3	14	21	500	1	15
2,9	-18,7	-6,7	13,4	-4,2	-0,5	-0,1	14	21	250	0,5	15

Table 13: The same data converted into discrete symbols. Classification could be done with the separation level in chapter 4.1 for the features and some intuition for the VNS parameters.

VR	VA	VS	VI	VX	VT	HR	ON	OFF	P	A	F
LOW	LOW	CONST	LOW	HIGH	LOW	LOW	LOW	HIGH	SHORT	MEDIUM	MEDIUM
CONST	HIGH	CONST	LOW	HIGH	CONST	CONST	LOW	HIGH	LONG	HIGH	LOW
CONST	CONST	LOW	LOW	CONST	CONST	HIGH	LOW	HIGH	MEDIUM	LOW	HIGH
CONST	CONST	HIGH	LOW	CONST	HIGH	HIGH	LOW	HIGH	LONG	LOW	LOW
LOW	LOW	CONST	HIGH	CONST	CONST	LOW	MEDIUM	MEDIUM	LONG	HIGH	MEDIUM
HIGH	LOW	LOW	HIGH	LOW	CONST	CONST	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM

The data has to be prepared for the use with decision trees. Typically, numerical values are not being processed with a DT. Decision Trees work with rules for discrete attributes in contrast to regression trees that have the task to deal with continuous domains.

Unfortunately, the number of features to be classified is 7 but the ID3 algorithm works with one feature alone. This is not a real problem, though, because any combination of the features can be considered a CASE that can replace any occurring combination of attribute values.

Table 14: The same data converted into discrete symbols. Classification could be done with the separation level in chapter 4.1 for the features and some intuition for the VNS parameters.

VR	VA	VS	VI	VX	VT	HR	CASE
LOW	LOW	CONST	LOW	HIGH	LOW	LOW	CASE1
CONST	HIGH	CONST	LOW	HIGH	CONST	CONST	CASE2
CONST	CONST	LOW	LOW	CONST	CONST	HIGH	CASE3
CONST	CONST	HIGH	LOW	CONST	HIGH	HIGH	CASE4
LOW	LOW	CONST	HIGH	CONST	CONST	LOW	CASE5
HIGH	LOW	LOW	HIGH	LOW	CONST	CONST	CASE6

Table 15: The final data for induction: In full data some or all cases will repeat.

CASE	HR	ON	OFF	P	A	F
CASE1	LOW	LOW	HIGH	SHORT	MEDIUM	MEDIUM
CASE2	CONST	LOW	HIGH	LONG	HIGH	LOW
CASE3	HIGH	LOW	HIGH	MEDIUM	LOW	HIGH
CASE4	HIGH	LOW	HIGH	LONG	LOW	LOW
CASE5	LOW	MEDIUM	MEDIUM	LONG	HIGH	MEDIUM
CASE6	CONST	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM

The decision tree induction only is feasible if the number of different cases is not larger than 25%. This means that the homogeneity factor f should be less than 25 when calculated from the number of unique cases c and the number of records k.

$$f = 100 \cdot \frac{c}{k} \tag{62}$$

For higher numbers of unique cases the tree becomes more and more a trivial decomposition of the table. If there are only unique cases, then there is absolutely no informational gain if compared to the original table. The choice of value ranges that belong to an attribute value, the number of attribute values per attribute and the total number of attributes influence the number of different cases that can be classified. The maximum number of theoretically possible cases can be calculated:

Attribute A (63)

Number of values per attribute $n_A \in \mathbb{N}$ (64)

Number of attributes $N \in \mathbb{N}$ (65)

Number of possible cases $$c = \prod_{A=1}^{N} n_A \tag{66}$$

For two attributes (VR and HR) á three values (LOW, MEDIUM, HIGH) there are 9 possible cases. This is already a homogeneity factor of 11.7%. That is roughly 9 parameter vectors per case.

At each level of the rule recognition process each input attribute (T_{ON}, T_{OFF}, P, A, F) is investigated separately for an information gain when used at this level of a tree. In the Information Theory the average number of decisions to classify a number of symbols is referred to as the information content. For binary decision problems the informational content of a set of ten symbols is ld(10)=3.32.

$$2^{3.32} = 10 \tag{67}$$

The entropy of binary decision trees can be calculated as weighted information content of the two subsets that it would split up [39, p.305].

Entropy level $$I(p,n) = -\frac{p}{p+n} \cdot ld\left(\frac{p}{p+n}\right) - \frac{n}{p+n} \cdot ld\left(\frac{n}{p+n}\right) \tag{68}$$

A more general form is required when a different number of values exist for the target attribute (non binary classification).

$$S = \sum_{j=1}^{n} x_j \tag{69}$$

$$I\left(x_{1..n}\right) = -\sum_{i=1}^{n} \frac{x_i}{S} \cdot \frac{\ln\left(\frac{x_i}{S}\right)}{\ln\left(n\right)} \tag{70}$$

One way to decide what attribute to take is to take is to take the attribute with the highest I. In literature it is very often spoken of informational gain that is calculated. In order to calculate a gain it is necessary to calculate the initial information level E_A. The gain is defined then:

$$G_A = I_A\left(x_{1..n}\right) - E_A \tag{71}$$

For binary trees the initial value E_A is calculated: [39, p.307]

$$E_A = \sum_{i=1}^{n_A} \frac{t_i + \mathrm{f_i}}{t + \mathrm{f}} \cdot I\left(t_i, \mathrm{f_i}\right) \tag{72}$$

The variables t_i (TRUE values in conjunction with attribute value i) and f_i (FALSE values in conjunction with attribute value i) are sums of occurrences. The variables t and f are sum of global occurrences.

For more general setup the following calculation is to be used:

$$E_A = \sum_{i=1}^{n_i} \frac{\sum_{k=1}^{d} x_{k,i}}{S} \cdot I\left(x_{1..d}\right) = \sum_{i=1}^{n_i} \frac{\sum_{k=1}^{d} x_{k,i}}{\sum_{l=1}^{d} x_l} \cdot I\left(x_{1..d}\right) \tag{73}$$

The constant d is 2 in binary systems and represents the number of distinct classification types. The variable x_y is the number of class $y = \{1..d\} \subset \mathbb{N}$ entries and $x_{y,i}$ is the number of class y entries in a specific set belonging to the attribute value i.

Classification Trees are a special variant of decision trees because they accept continuous input data [40]. This new ability is paid with the restriction to two leafs for each node. The node is making a decision based on an inequality like P > 200 or F<15. The method for finding those splitting points is called binary partitioning. The training samples are sorted and the information gain is calculated at every point as if it was a splitting point. The splitting point with the highest information gain is compared to all the other gains for the other investigated attributes. At any branch the attribute with the highest information gain is taken together with the splitting point that applied to it [41]. Because entropy and information level are inversely related to each other this method is called the entropy discretization [42]. There are other methods for discretization for both binary and multi-interval trees. There exists a binary discretizations based on inter- and intra class variance [15, p.33]. There also exist multi-interval algorithms that either have static or dynamic intervals for discretization. Those are brute force attempts in order to calculate the best cut-points [15, p.35]. But there is other work to mention that has dealt with finding more effective branching like [42] or [43] which are based on MLD (Minimum Length Description) criteria. But there are other discretizations based on LVQ by [44], [45] , histogram based discretizations [46],[43] and Chi-Merge discretization [37],[15, p.38].

Regression Trees are special variants of classification trees. They allow for continuous output.

Pruning. No matter of what tree could be used for knowledge discovery there is a branch level at which the tree is overtrained (overfitting to training data [15, p.42]). At this level the branching algorithms try to homogenize noisy sets of data generating waste branches that decrease the performance of a tree first in runtime performance and second in classification performance because typically, data presented to the tree inducer does not satisfy the expected level of redundancy, consistency, completeness and efficiency. Therefore a pruning mechanism like the χ^2 algorithm is required. If test data cannot be provided for the pruning, then an abortion criterion can be set from where E_A is high enough or G_A is low enough.

DT's can be used in two ways. They can either be used for classification of a result or they can be used for classification input parameters that cause a requested specific feature. Applied to VNS it means that we can either induce a tree for input parameters or for stimulation effects.

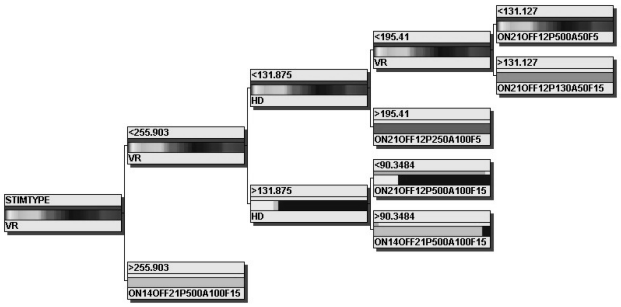


Image 56: Typical look of a classification (decision) tree with continuous inputs

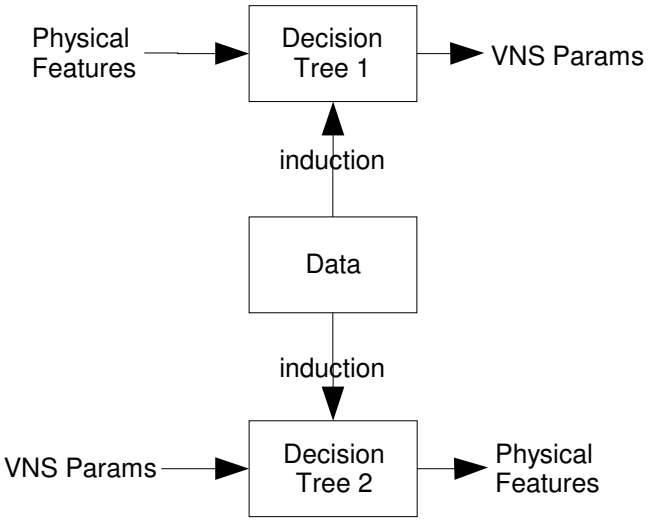


Image 57: The induction can be done in two ways.

Both variants can be useful. The tree no. 1 has the advantage that by answering a series of simple questions about the effects one of the initial VNS parameter vectors will be returned quickly. But this only holds true if the questioned combination of features exists at all. The disadvantages are quickly recognized: It is up to the user to check whether all relevant features occurred during the tree scan and the results can never lie outside the original input vectors. A generalization is therefore impossible. This kind of tree is a dead end if ones goal is an optimized result.

The tree no. 2 has the advantage that if it is laid out as a classification tree also unknown parameter vectors can be tested. But this solution also has its disadvantages. A tree like this can only know of the features that had been presented to it before. Thus it is unable to find a better solution than those who were available in the data, originally. If there are no satisfactory combinations of features in the original data, then no better solutions than that can be found with a decision/classification tree. A solution here is a regression tree that operates on continuous data.

3.11.5 Sensitivity- and Uncertainty Analysis

When for some reasons the predicted parameters do not result in optimal response then the question arise where to go? This is an issue of informed decision making in heuristic systems [47]. Sensivity Analysis (SA) classically deals with the question "what is the parameter that has the strongest influence on the outcome." Sensitivity Analysis is often used in conjunction with Uncertainty Analysis (UA) which deals with the random character of input to output relationships.

In linear systems i.e. the parameter dimension with the highest value of the 1st derivative is the most influencing factor on the outcome. In more complicated systems the question flattens to "how important is it to fine tune a specific parameter in order to achieve the effect or an effect at all?" One idea to give a hint is to compare this problem to blind darting.

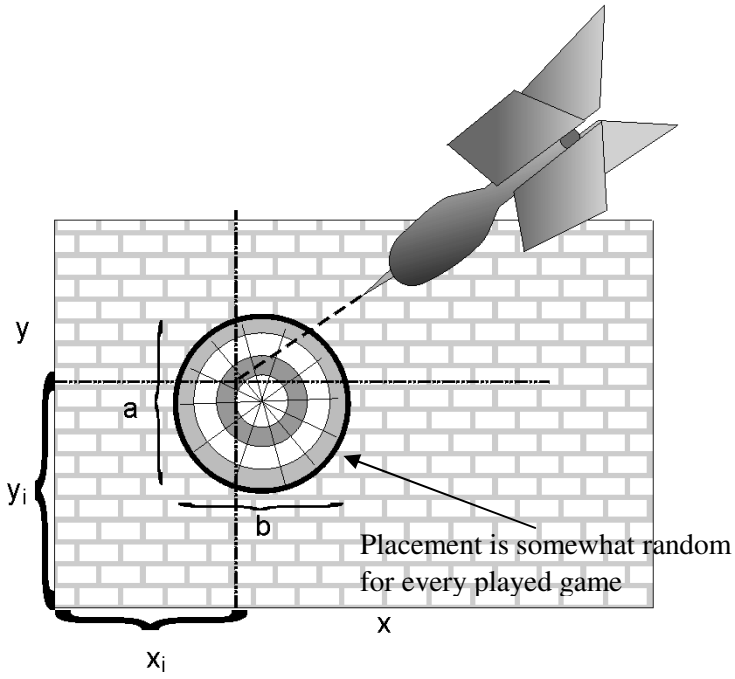


Image 58: Blind Darting. Where to aim if the initial guess failed because of individual feature placement? a and b are the dimensions of the feature target area, where we expect the biggest effect or an effect at all. Statistically, this area is placed at xi,yi coordinates (analog: optimal paramter vector) and one would begin each game by aiming at this point with a dart every time. X and Y are the parameter space. In darts this parameter space is only 2D – in the VNS discussion it is 5D.

In UA it is practice to rerun the experiment for a known set of parameters a couple of times. This was done during data acquisition, when the same class of rat was tested with the same protocol but with different individuals bringing in some randomness. It is taken for granted, that the VNS parameters are not the only effectors in place and thus the uncertainty is high with every factor. This type of analysis is often used to identify irrelevant factors or to initiate a robustification measure.

Table 16: UA: Average randomness in % in the relationship between input and output. The smaller the numbers the more important is the input parameter quantitatively.

Phase 1	VR	VA	VS	VI	VX	VT	HR
T_{ON}	4,67%	11,63%	5,21%	6,39%	6,24%	3,42%	0,98%
T_{OFF}	4,67%	11,63%	5,21%	6,39%	6,24%	3,42%	0,98%

P	5,10%	13,73%	5,29%	8,84%	7,09%	3,47%	1,10%
A	2,89%	10,70%	5,64%	7,37%	3,58%	4,20%	0,45%
F	3,27%	10,27%	6,30%	6,86%	6,20%	4,75%	0,63%

The upper table shows the influence of the parameter vectors on the effects observed, quantitatively - that means how it does influence effect strength alone. The values in the table are expectedly high but can give advice what parameters to change first. If the patient suffers from snappy breathing you would probably like to reduce the effects of VS. Because of the values in the table you would change T_{ON} and T_{OFF} values first or at least more decisively than the parameter P, A or F.

If the patient does not show the qualitative changes of response the upper table is not helping much.

One could start to systematically try all parameters around the current one if the current parameter vector has already achieved some of the goals. In the situation where the body is not responding well at all the problem increases extremely. Unfortunately, the time required to try parameters along every of the five dimensions is quite troublesome for the patient because of combinatorial explosion.

A standard way to look for optima is to get the gradient in a place and move along the steepest path. It is probably not possible to measure two close parameter vectors and create a gradient to the optimal effect because too close parameters do not offer the necessary safety of measured feature differences, too far away parameter vectors do not offer the necessary locality and optimality in therapy sense is not specified in terms of maximum or minimum features. Maximum and minimum can be applied to scalars but not to the typically requested vectors (combinations of effects).

If we think of darts again, there can be some knowledge discovered from the setup. If we assume that the target board is always the same size then if we obviously do not hit the center that's because it is hanging somewhere else. If we know that the width of the wall is twice the height of the wall but a equals to b then we can deduce that the horizontal aiming position is twice as important for success as vertical one. This can be expressed in such terms that more radical movement along the horizontal axis is required for quicker board detection. The size of the wall is known but how about the targeting board. The size of the board is not known and therefore the problem becomes a problem of shape detection. What is needed, though, is the ratio between a and y, b and x making the knowledge of the precise "shape" of the targeting area somewhat less important because what the practitioner needs is the knowledge along what axis to move more radically – what direction to go in order to hit the "board."

There are three ways how such a direction advice can be given. The first method is the one used in conjunction with optimal parameter tables - the other is used on data. The third way is to use the diagrams from chapter 4.3 in order find a direction towards a singular effect.

3.11.5.1 Parameter Importance Factors

A Parameter Importance Factor (PIF) can be calculated from a table of optimal vectors. If the dimensions of each of the parameters are fine grained enough and if the total range covers all

practically available values then it can be stated that if there are only few specific values from a parameter domain, then this parameter is more sensitive to appropriate choice. If in the table of optimal vectors all values of some domain show up, then the choice of the parameter seems to be uncritical. Simplified this can be expressed as a quotient between v - the count of different values in any column - and D – the total count of investigated values. Ideally, the values used for the table should be evenly spaced and D should be roughly on the same order of magnitude or the comparability of PIF values will suffer.

$$pif = \frac{D}{v} \tag{74}$$

Table 17: Different parameters with exemplary domains used for the calculation of a PIF indicator:

P1	P2	P3	P4	P5
valid values: 1,2,3,4,5,6,7,8,9,10	valid values: -2,-1,0,1,2	valid values: 0.0,0.2,0.4, 0.6,0.8,1.0	valid values: 1,3,5,7,9	valid values: 50,100,150,200,250

Table 18: The parameter importance factor is calculated in exemplum:

P1	P2	P3	P4	P5
3	-2	0.8	1	150
3	-2	0.6	1	150
4	-2	0.8	3	150
4	-1	0.6	3	150
4	-2	0.4	5	150
3	-2	0.6	5	150
5	-1	0.2	7	150
4	-1	0.2	7	150
3	-1	0.0	9	150
PIF_1=11/3	PIF_2=5/2	PIF_3=6/5	PIF_4=5/5	PIF_5=5/1

PIF's with highest values indicate that this parameter has to be set most precisely. In turn that means that if the requested effect did not occur then the parameter with the highest PIF was set to a wrong value most probably.

3.11.5.2 Effect Sorting

A different way to get parameter importance estimation is by sorting the data. If the investigated data is sorted by effect then data for a diagram can be produced where vectors associated with the same kind of response level are howled together into a neighbor relationship.

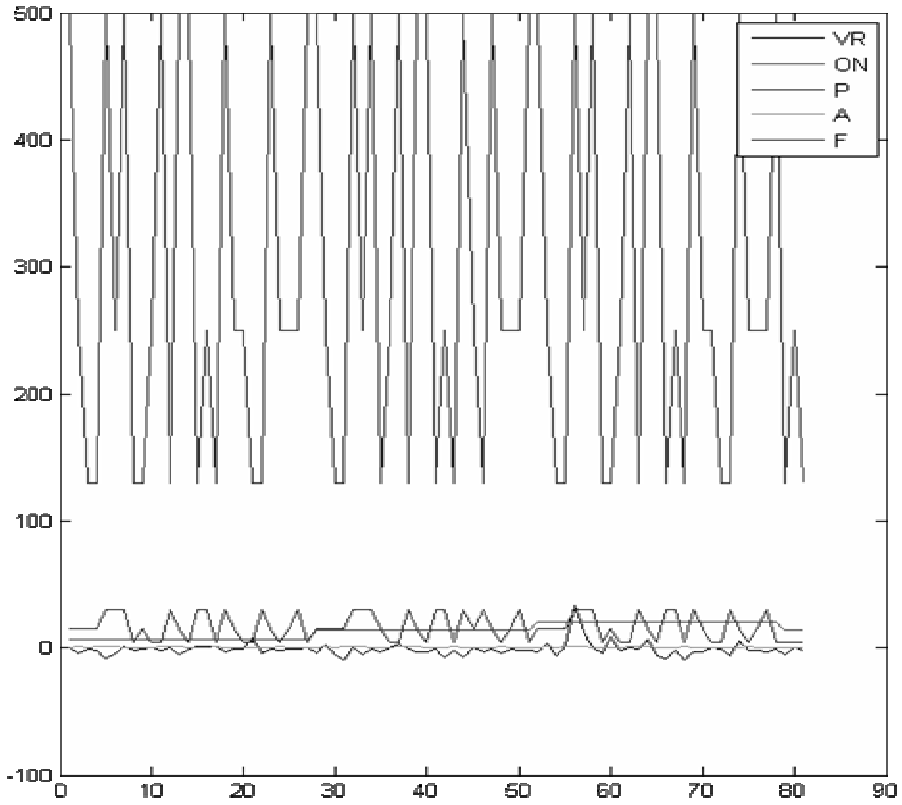


Image 59: not normalized, unsorted data from phase 1

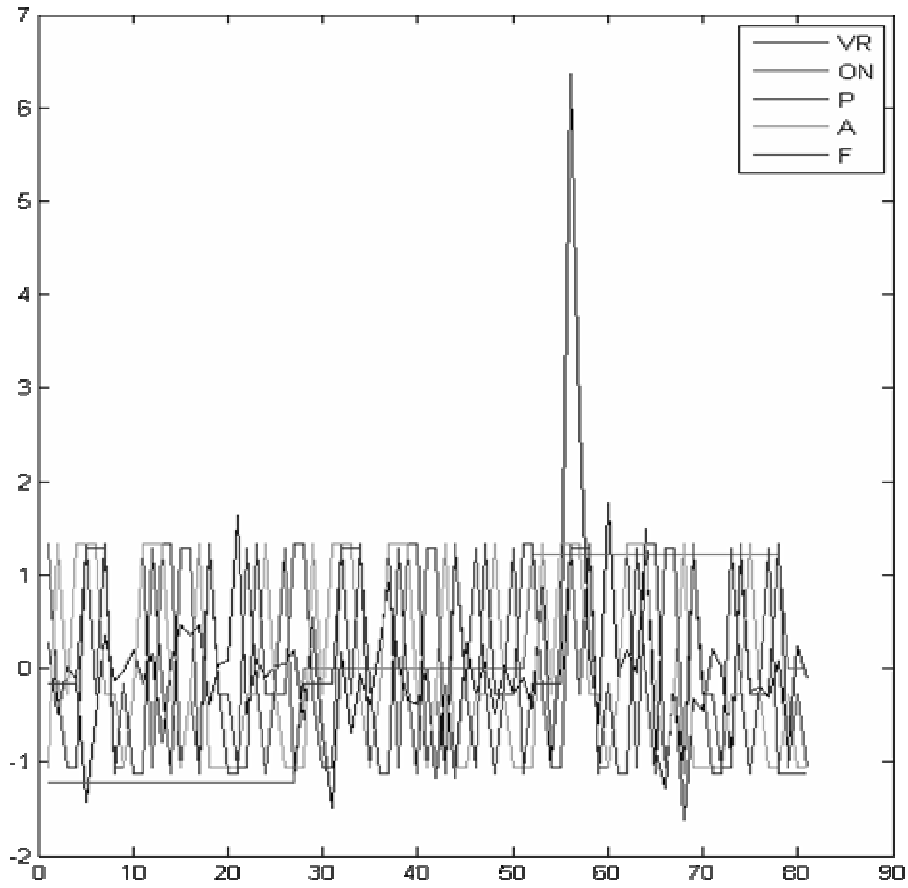


Image 60: normalized, unsorted data from phase 1

The better the data sorted by effect the more complicated are the clusters of data belonging together because none of the parameters is clearly responsive for any of the effects by itself alone (as shown in table 16). Once the data is sorted by the feature no more sorting is possible and therefore the ability to gain knowledge out of the data is limited.

The process can be reversed, though. Data can be sorted recursively four times. The channel T_{OFF} is left out because it only occurs in relationship with specific values for T_{ON} and does not offer additional information. The idea is: The better the data is structured the better the correlation between the parameter sorted at the end.

The function implemented this way is BestColumnOrder(). It is a recursive algorithm exploring all branches of a tree where each leaf is a fully sorted data in a specific column order. There are as many leafs as there are combinatory possibilities to sort the columns. The possibilities are evaluated through a disorder function. The more up and down in the function, the higher the indicator g for the feature i.

(un-) sorted feature data

$$\vec{S}_i \in \mathbb{R}^n \tag{75}$$

disorder indicator

$$g_i = \sum_{j=2}^{n} \left| \vec{S}_{i,j} - \vec{S}_{i,j-1} \right| \tag{76}$$

It is very interesting to note, that the double normalized, sigma limited function comparation between offers almost the same result table. Thus shall g-factor be justified for this purpose.

IPD indicator

$$p_i = \frac{\left| \sum (F_i - F_{i-1}) \cdot (S_i - S_{i-1}) \right|}{\sum \left| (F_i - F_{i-1}) \right| \cdot \left| (S_i - S_{i-1}) \right|} \tag{77}$$

Table 19: The columns and their numbers

Column 1: VR
Column 2: ON
Column 3: P
Column 4: A
Column 5: F

Table 20: The column sorting results are compared to each other (g-factor)

Phase 1	1st column	2nd column	3rd column	4th column	g_{VR}
worst	5	2	4	3	55,18
23	2	3	4	5	55,33
22	3	2	4	5	55,36
21	2	4	5	3	55,96
20	2	5	4	3	56,46
19	2	4	3	5	56,61
18	5	4	2	3	57,50
17	3	4	2	5	57,55
16	4	3	2	5	57,73
15	4	2	5	3	58,34

14	4	5	2	3	58,47
13	4	2	3	5	58,98
12	3	5	2	4	61,05
11	5	3	2	4	61,11
10	3	4	5	2	61,84
9	4	3	5	2	62,02
8	3	5	4	2	62,84
7	5	3	4	2	62,89
6	5	4	3	2	63,73
5	5	2	3	4	63,83
4	4	5	3	2	64,71
3	2	5	3	4	65,10
2	2	3	5	4	65,38
best	3	2	5	4	65,42

Table 21: The column sorting results are compared to each other (IPD)

Phase 1	1st column	2nd column	3rd column	4th column	p_{VR}
worst	5	2	4	3	0,0119
23	2	4	5	3	0,0121
22	2	3	4	5	0,0121
21	3	2	4	5	0,0121
20	2	5	4	3	0,0122
19	2	4	3	5	0,0123
18	5	4	2	3	0,0124
17	4	2	5	3	0,0125
16	3	4	2	5	0,0125
15	4	3	2	5	0,0125
14	4	5	2	3	0,0125
13	3	5	2	4	0,0125
12	5	3	2	4	0,0125
11	3	4	5	2	0,0126
10	4	3	5	2	0,0126
9	4	2	3	5	0,0127
8	3	5	4	2	0,0127
7	5	3	4	2	0,0127
6	5	4	3	2	0,0128
5	5	2	3	4	0,0129
4	4	5	3	2	0,0130
3	2	5	3	4	0,0130
2	2	3	5	4	0,0131
best	3	2	5	4	0,0131

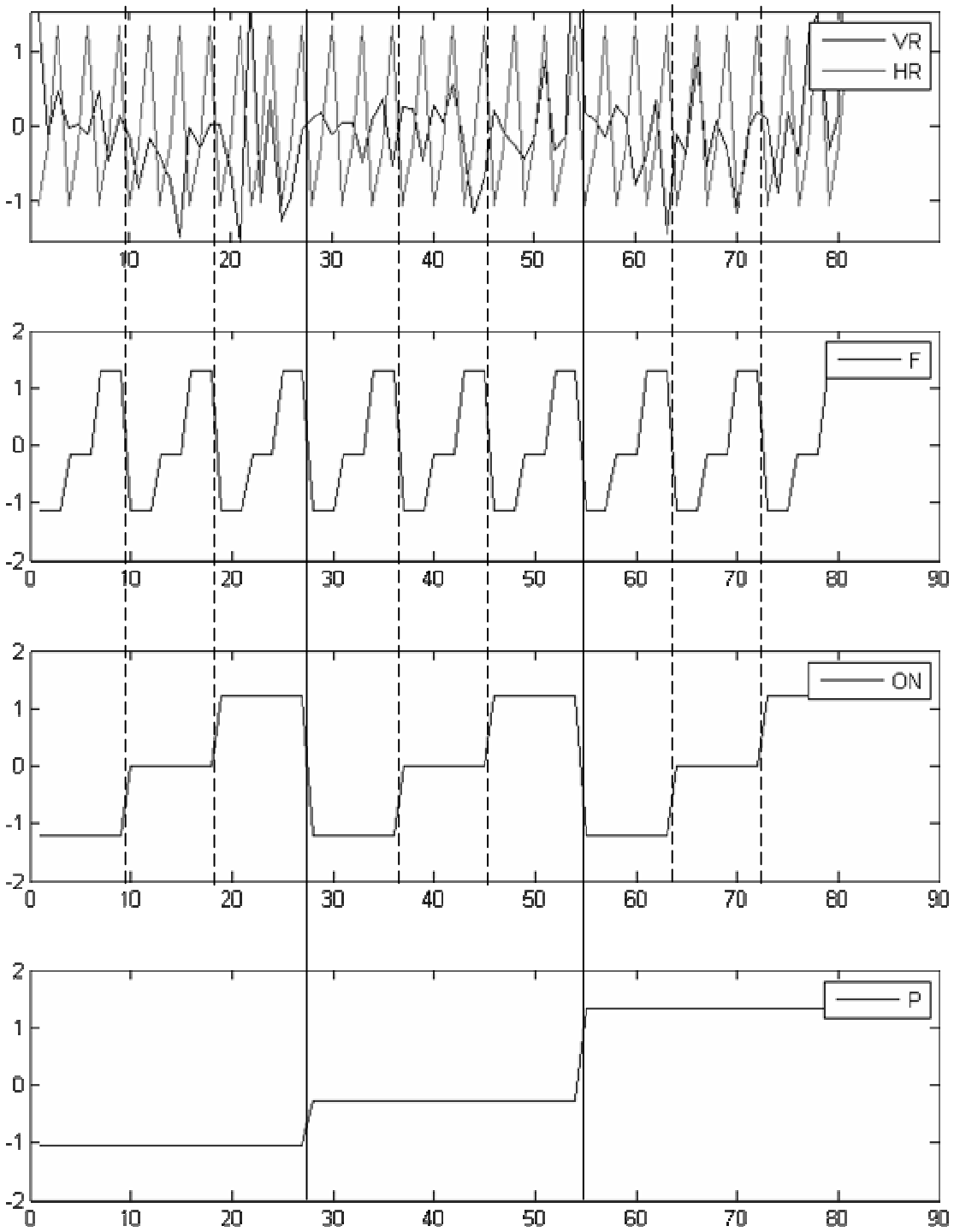


Image 61: An informative correlation between VR and A is developed when first sorted by P, then by ON (ON/OFF) and then by F.

This kind of diagram can be used to retrieve export knowledge in simple means like in this example:

For any values of pulse width the VR response is inversely related to the current A and starts with almost no effect for low values of ON (ON/OFF) and ends up with high VR suppression effect with longer ON (ON/OFF) values. Generally, during low and medium long ON (ON/OFF ratios) values the suppression effect dominates. For high values of ON (ON/OFF

ratio) VR rather increases. High frequency seems to be one important factor for the VR alteration strength. Therefore stimulation frequency seems to be the second most important factor in controlling VR alterations after A. The third most important factor seems to be the time because of the upper observation. The softest control over the VR effects comes from the parameter P which is gently intensifying the effects described above.

In conclusion to this description can be stated, that the order of the columns also specifies their immediate influence on the outcome. In this example the current A is strongly responsive for the effect, but can go both ways – up or down depending on the parameter F and time ratio.

3.12 K-NN Classification

The "k nearest neighbors" classifier has been first mentioned by Fix and Hodges [48] in 1951 and has received repeating attention in order to make it computationally efficient [49]. K-NN is considered an optimal classifier because for $k \to \infty$ it does have a failure ratio not higher than the level of randomness in the data.

For the compacted UniRat and pre-classified data there exists the possibility to apply the k-NN algorithm in order to find the direction of the physical feature. Pre-classification can be done like portrayed in chapter 3.11.4. Refer to the appended tables in chapter 7.2.

Although the k-NN classifier is unable to obtain results beyond the original scope it offers the ability to generate a result space similar to the one produced by the neural networks. It is in so far different as it would not have any time conception.

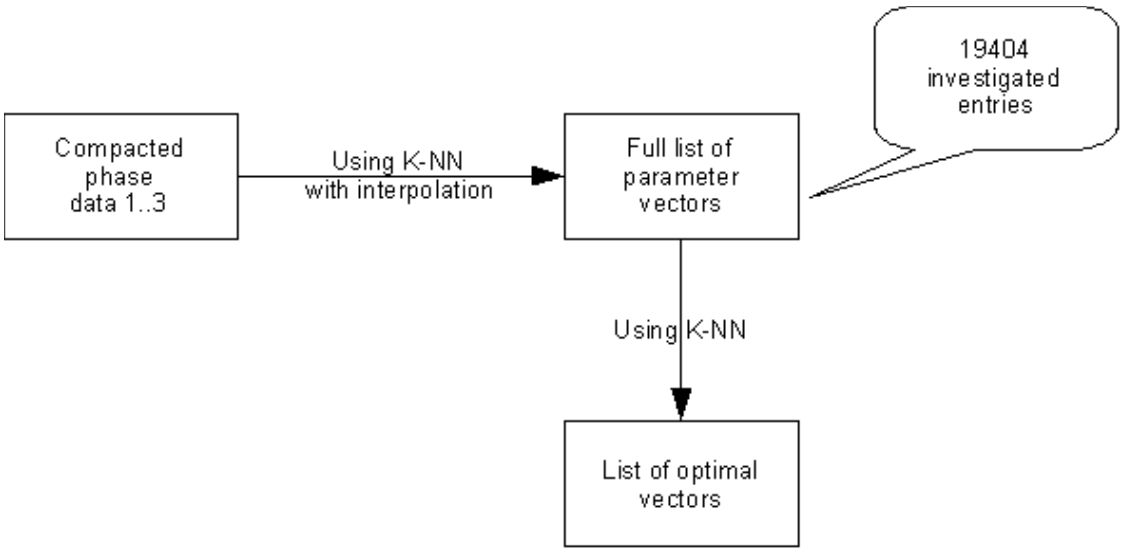


Image 62: The K-NN algorithm can be used for result space generation and for the evaluation of your theory.

The generation of the full parameter list is a straight forward idea. The compacted phase data as presented in chapter 4.4 and is taken and processed in such a way that artificially generated parameter vectors are classified with that data

```
function L = List_from_K_NN( Compacted );

L(1:19404,1:14) = 0;
i = 1;
CountOfBest     = 3;
InterpolateBest = 1;

for TON = 5:5:30 %6
    for TOFF = 5:5:30 %6
        for P = 100:40:500 %11
            for F = 5:5:35 %7
                for A = 0.1:0.2:1.3 %7
                    TestVec = [TON TOFF P A F];

                    V = K_NN( Compacted( :,[ 8 11:14] ), Compacted(
                    :,[1:7] ), TestVec, CountOfBest, InterpolateBest
                    );

                    L(i, : ) = [ V TON 0 0 TOFF P A F ];
                    i = i + 1;
                    if mod(i,100)==0
                        disp( '.' );
                    end;
                    if mod(i,1000)==0
                        disp( fix(i/1000) );
                    end;
                end;
            end;
        end;
    end;
end;
```

The K_NN() function can be given any parameters for the first parameter and any results as the second parameter and returns an interpolated results according to how close it comes to the test vector. This flexibility of this function quite relevant, because the function can be used again in order to serve as an interrogation facility for the retrieved list L. This is a good alternative to the TherpyFinder module which takes a long time to compute the results. The list L contains a wide range of parameter vectors and physical responses that pretend to be recorded from a rat, directly.

```
function V = K_NN( Params, Result, TestVec, Interp, Mode );

if ~exist( 'Interp', 'var' )
    Interp = 1;
end;

if ~exist( 'Mode', 'var' )
    Mode = 1; %means interpolation - 0 means dump the first Interp
end;


S = std( Params );
O = mean( Params );
c = size( Params, 1 );
```

```
w = size( Params, 2 );
h = size( Result, 2 );

for i = 1:w
if S(i) == 0
    S(i) = 1;
end;
end;

for i = 1:c
Params(i,:)  = ( Params(i,:)  - O ) ./ S;
end;
TestVec = ( TestVec - O ) ./ S;

Params( 1, w+1 ) = 0;

Base = [Params Result];
for i = 1:c
    Base( i, w+1 ) = norm( Params( i, 1:w ) - TestVec );
end;

Base = sortrows( Base, w+1 );
V = [];
if ( Mode == 1 )
    Base(1:Interp,w+1)=max( Base(1:Interp,w+1), 0.0001 );
    Base(1:Interp,w+1)=1./Base(1:Interp,w+1);
    F = sum( Base(1:Interp,w+1) );

    V(1:h) = 0;
    for i=1:Interp
        V = V + Base(i,w+2:w+1+h) * ( Base(i,w+1)/F );
    end;
    return;
end;

if ( Mode == 0 )
    V = Base( 1:Interp, w+2:w+1+h );
    return;
end;

return;
```

Examples of usage:

Generate Lists:

```
L1 = List_from_K_NN( Compacted1 );
L2 = List_from_K_NN( Compacted2 );
L3 = List_from_K_NN( Compacted3 );
```

Specify Theory:

```
VR=1; HR=7; %columns

%We want to know: TON, TOFF,P,A,F
ParamVecs =[8 11:14];
```

```
%Hypothesis is -5% in VR and 0% in HR are good for treatment
Hypothesis =[-5 0];

BestParamVex = K_NN( L(:,[VR HR] ), L(:,[ ParamVecs]), [ -5 0 ], 10, 0 )
```

See chapter 4.5 for examples of results.

3.12.1 Interpolation Methods

A classical method used during optimization is called interpolation. Interpolation is always then needed, when no analytically evaluable function is present. Instead singular points are given which can have any dimensionality. From these original points intermediate points can be deduced with the help interpolation. The simplest interpolation method is linear interpolation but it is hardly any help when you have only few points to put up. Better methods have been developed like polynomial, trigonometric (Fourier Transform), spline, hermite or bezier interpolations. Such interpolations better approximate natural systems and can reach values beyond original scope. At joints hermite interpolations can offer high depth levels of smoothness.

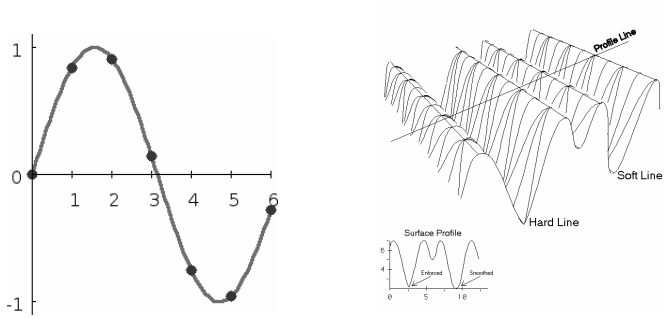


Image 63: 1D spline interpolation and 2d spline interpolation

Unfortunatelly, there are some restrictions to interpolation that excludes available data from the application of interpolation. Most algorithms supplied by Matlab require uniform spacing. These algorithms offer quite a few methods that could be investigated but the questioned physical response function is supported only by a low number of non-uniformly spaced vectors. Additionally, interpolation algorithms require that there is only one result value per vector. Original UniRat does not satisfy this requirement. Only the compacted data offers a basis for interpolation.

There are three ways how an interpolation can be achieved anyway.

The first is to use Matlab‘s griddatan() function which can work in two modes: linear and nearest. Essentially, the nearest mode does a Voronoi parceling similar to the one done in chapter 3.11.2 but offers numeric results. The linear mode of operation interpolates between the supporting vectors by the Delaunay tessellation method also used in the QHull [50][51] algorithm but cannot extrapolate and is therefore of a very limited use in this case. It cannot even calculate interpolations for all parameter vectors that clearly lie inside the input range because they lie outside of the triangulation area span by the example vectors. The nearest variant works,

always, and is applicable to parameter vectors outside the input range. The results are more precise than just an 'increase', 'decrease' or a 'constant' statement but may mislead the reader to expect such exact values in experiments. The resulting hypermatrix or hypercube of the so obtained values is only a rough prediction of the feature strength change.

The second method is to use the Radial Basis Function (RBF) neural networks. RBF networks can easily (and quickly) learn from the example data but is often over specific and too bumpy. If the supporting vectors are spaced too far away, then artificial extreme points may be introduced. If the supporting vectors are spaced too close to each other then fine feature might get lost.

In both cases the immediate result is a five dimensional hypermatrix that can be graded according to a given hypothesis. The graded results are added and form a limited search space that is small enough for linear search method. Phase data is presented in chapter 4.4.

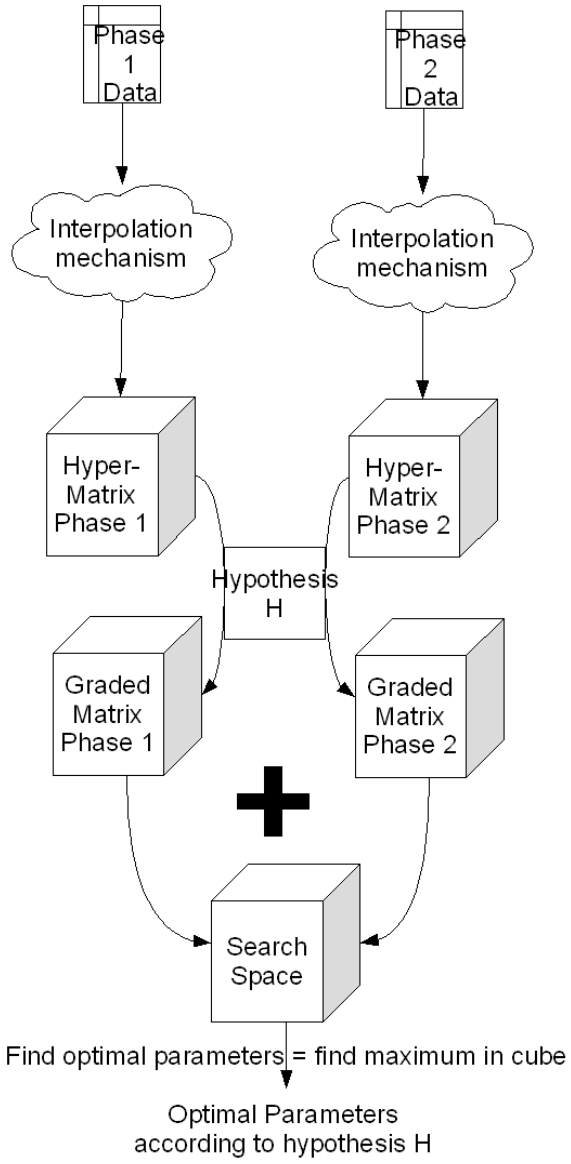


Image 64: Data transformation according to the interpolation scheme

The third method of interpolation is not quite obvious. The gridfit() method can be used to generate smooth planes that go through the mean averaged points. Through inference (chapter 3.16) those planes can be put together to form a hypermatrix that can be used as shown in the scheme image 58.

3.12.2 Input Pattern Detection

Input Pattern Detection (IPD) is a method that was developed during the science seminar for the analysis of the Airbus mounting facility data in order to suggest data transformations that could improve the generalization performance of the artificial networks [52]. It was applied to the original UniRat data. It did a trivial but correct discovery that the development of the biological curves was mainly related to the timer t1 and timer t2. One other suggestion among the operations matrix φ was the use of an T_{ON}/T_{OFF} ratio.

3.13 Network Design by Genetic Algorithms (GA)

Each of the seven dimensions (VR, VA, VS, VI, VX, VT and HR) depends on a set of input parameters, namely ON, OFF, t1, t2, P, A, F which are referred to as the parameter vector. The purpose of the artificial neural network is to serve as a statistical model of the specified rat population's cardio-respiratory response to VNS parameters. In other words the artificial neural network serves as an advanced, highly non linear vector field. For simplicity and because of good experience with singular outputs the function is separated into seven separate scalra functions.

It is supposed that a neural network is suitable to represent such a scalar function.

feature of channel i

$$f_i \in \mathbb{R} \tag{78}$$

neural function

$$n_i\left(\mathbb{R}^7\right) \to \mathbb{R} \tag{79}$$

$$f_i = n_i\begin{pmatrix} T_{ON} \\ \dots \\ F \end{pmatrix} \tag{80}$$

The goal of genetic algorthms is to find an appropriate network design that is able to represent the biological function n_i as closely to reality as possible. As the result of the genetic (evolutionary) design process a repeating familiar architecture is expected that can be trained with available tools. It is not the aim to produce new kinds of architectures along with the often required architecture-specific training algorithm.

The number of architectures is large, but not limitless. The following table gives an overview over the type of network architectures that could or could not be used with the available data. The list is due to the frequent questions why other network types were not used. The neural network is a tool and not the main focus of the work. Hence it was concentrated upon working solutions implemented in Matlab because results can be reused easily in other components of Matlab and thus gives faster a usable product.

Table 22: Table of supported training algorithms by Matlab 7.1.

Name	Description	Comment	Used
Short connected Feed Forward	Neurons are a function of a sum of weighted input vector. Neurons are arranged to layers. Lower layers connect up to all higher layers.	Extensively investigated type of network with many applications like the function approximation and classification. Many training types back-propa-gation methods: - Gradient descent - Momentum descent - Adaptive descent - Levenberg-Marquardt - Quasi-Newton(BFGS) - Online and Batch variants - Bayesian regulated descent - Powell-Beale - Fletcher-Powell - Polak-Ribiere - One-Step-Secant descent - Resilient - In-order / Out-of-order updating	ALL
Cascade Correlation Feed Forward	Same as above, but only one neuron per layer.	Powerful creationist variant, but not supported by Matlab. The same training methods applicable as above.	NOT USED
Feed Forward	Neurons are a function of a sum of weighted input vector. Neurons are arranged to layers. Any layer of neurons is connected just to the next layer of neurons.	Classical variant to which also belongs the Perceptron. Special case of short connected feed forward neurons. Equivalent to short connected weights are zero.	INCLUDED
Elman Networks	Forward directed networks without short connections. Output is redirected to input.	Network type is well suitable for streamed data. Not suitable for data with time holes or no order data.	NOT SUITABLE
Jordan Networks	Forward directed networks with virtual memory layers for each of the layers.	Network type is well suitable for streamed data. Not suitable for data with time holes or no order data.	NOT SUITABLE
Learning Vector Quantization (LVQ)	Winner takes all principle applied to vectors.	This type of network is designed for classification purposes.	NOT SUITABLE

Hopfield Networks	Fully feedbacked networks. Neurons most often with a binary activation function.	Suited for complicated, context based pattern transformation or optimization under constrains. Mostly used in its binary form.	NOT SUITABLE
Boltzmann Machine	Like Hopfield, but different kind of activation function for neurons.	Statistical search for global minimum using simulated annealing. Similar application as Hopfield.	NOT SUITABLE
Bidirectional Associative Memory (BAM)	Two layer, fully feedbacked neural architecture. Related to Hopfield.	Application in pattern transformation.	NOT SUITABLE
Radial Basis Function Feed Forward (RBF)	Short range activated neurons arranged in feed forward manner.	Local optimization helps to speed up the training of this feed forward type network. Suited for ad-hoc interpolations but great danger exists for creating bumpy functions.	POSSIBLE, BUT NOT EVALUATED
Probabilistic Networks	Also called Bayesian Networks. Feed forward type of network with risk calculation.	Mainly used for classification.	POSSIBLE, BUT NOT EVALUATED
Adaptive Resonance (ART)	A family of related neural models that preserve knowledge but allow construction of new neural clusters.	This kind of networks allows the recognition of new classes of inputs without erasing old ones. Suited for classification problems where not all classes can be defined in advance.	NOT SUITABLE
Neocognitrons	Multi stage, multi cluster, 2 type neuron kind of neural network.	The application of this network is in the field of image recognition (pattern classification)	NOT SUITABLE
Time Delay Networks (TDNN)	Networks with wandering inputs.	Classification problems in phonetics. Require streamed data without time holes.	NOT SUITABLE
Adaptive Logic Networks (ALN)	Neurons perform logical operations.	Synthesis of optimal logical expressions.	NOT SUITABLE
Self Organizing Maps (SOM)	Geometrically arranged LVQ for non supervied learning	Best suited for hard clustering problems and class definition.	NOT SUITABLE

The decision easily falls on feed-forward networks with shortcuts. Networks without shortcuts are special cases of same architecture. Feed-forward networks with short connections implemented in Matlab's neural network toolbox are used. Training Methods have been limited later

to some few that were really working like Leveberg-Marquardt (TRAINLM) One Step Secant Backpropagation (TRAINOSS) and Resilient Backpropagation (TRAINRP). Other training algorithms have consequently been sorted out. In order to improve speed in repetitive optimization runs the search space has been restricted to those three.

The number of layers and the number of neurons in each layer is subject to an optimization process based on evolutionary algorithm implemented in Matlab's evolutionary toolbox. The toolbox has shown limits, though. The floating-point nature of the genes made it difficult to have a quick optimization process. The bit-wise alternative generated too many invalid states. Based on these facts a discrete special purpose discrete GA algorithm was written (Evolution()). This special purpose function has significantly sped up the optimization process as it was not retrying similar networks for very long time. The use of the architecture history was reduced significantly. Architecture History is a facility within the fitness function CheckNetQuality() that helps avoid the costly retraining of the same types of networks.

Another parameter being optimized was the learning rate. It was limited between $0.0001<\mu<0.3$. The resulting μ resulting from the optimization process were on the scale of $\mu \sim 10^{-1}$

Further restrictions are coming from the data itself: The input and the output neurons are always linear; the hidden neurons are always sigmoid (TANSIG) as this proves to be the practically best function for practical problems. The attempt to use the LOGSIG alternative quickly reveals greater error rates. The maximum number of hidden layers was first limited to 6, later to 8 layers. The increase in depth did not offer an improvement of the final results. The optimizations process has sorted too deep networks out anyway because of slight depth penalty. This means that deeper networks had to be significantly better. The level of significance was put very low (<<1%) but this has already been enough to sort them out.

The fitness function performs multiple training attempts for each candidate with a small fraction of the data in order to cancel out the influence of the initialization. A one time run process finds the most sophisticated and complex data for a given window size with the FindMostSignificantData() function. In the end the network with the lowest mean square error is selected. In order to improve overall optimization performance the network is not retraining networks that are far off the field and skips a network if some kind of error occurred (this happened occasionally).

The genetic optimization functions have all been using the score minimization for sorting of their population members. The cross-over has been implemented in scattered manner in ga() and Evolution(). The default offers no better/worse performance than the alternatives. Therefore Evolution() uses single-point method which is easy to implement.

3.14 Training

The training of the neural network was done with all the data using the training algorithm that has been found by the genetic algorithm. For each cardio-respiratory feature one neural network was trained. The data is trained in blocks out of order in order to minimize the memorization/forgetting effect. The favorable batch processing was not possible. Instead a hybrid online/offline method was used in order to reduce RAM requirements.

3.15 Network Analysis

After an artificial network has been trained it contains a description of the dependencies between the input and the output neurons. In other words: at some local point it can be precisely said what inputs have the strongest, weakest or some kind of conditioned relationships. In order to learn from the network it is important to extract and to visualize this information. This is done during the network analysis phase. During this analysis each feature is investigated. For each of the features all combinations of the input parameters are tested. For each of the combinations many working points are evaluated. The structure describing the test is defined in Matlab as follows:

Table 23. VNS parameters recorded on rats and range of VNS parameters investigated by the neural network

			Investigated ranges			
parameter	Units	Recorded values	L *lower end*	U *upper end*	S *step size*	T Total number of values
Ton	Seconds (s)	7,14,21	10	30	4	6
Toff	Seconds (s)	12,20, 30	10	30	4	6
P	Microseconds (µs)	130,250,500	100	500	50	9
A	Milliamps (mA)	0.25,0.5,1	0.1	1	0.3	4
F	Hertz (Hz)	5,15,30	3	30	3	10

The total number of combinations is:

$$c = \prod_{i=1}^{7} \frac{Max_i - Min_i}{Step_i} + 1 \tag{81}$$

For the upper configuration c = 6 * 6 * 6 * 6 * 9 * 4 * 11 = 513216

Because it is not feasible to produce 513216 diagrams a method for reduction was chosen that works in such a way that it ...

1. ... generates a diagram only when it had a greater height span effect than the previously generated diagram for the same pair of tested parameters.

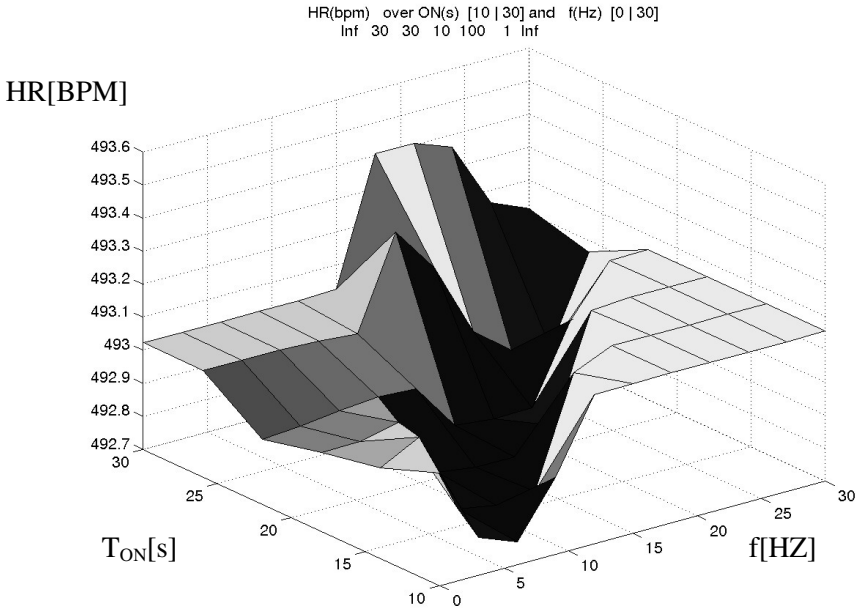


Image 65*: Across the diagram a valley can be seen.*

2. ... forms an arithmetic average over all of the generated data in order to produce a generic diagram for a specified pair of parameters in a manner that's independent of the other five parameters. These diagrams stand out through rather smooth surfaces. The lower example has a net change of roughly 0.04 BPM.

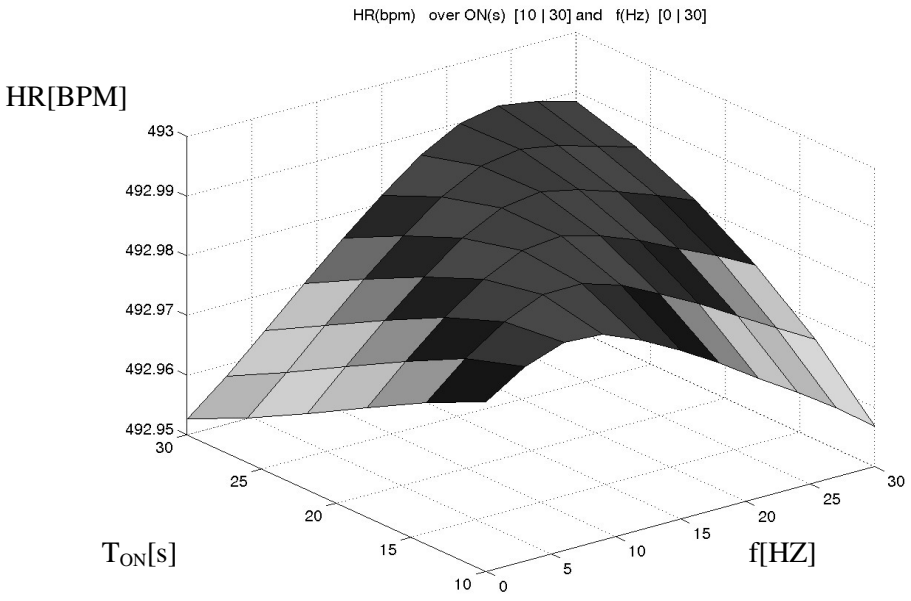


Image 66*: There where the valley has been seen before now emerges a hilly back.*

As it can be clearly seen, strong differences between the single working point diagram and the overall average occur. In the case of the heart rate across ON-time and the frequency the diagrams seem to have inverse relationships. This is not a conflict but shows that it cannot be safely concluded that individual setups always look like the average in their effect. Nevertheless the tendency diagram shows that along the hilly back in image 42 rather an increase than a decrease should be expected. Additionally it can be read from the diagram that heart rate depends on both, the frequency and how long the VNS was applied.

3.15.1 The Segmentation Process

The segmentation process is performed for every plane obtained by the analysis step. In this step it is strongly assumed, that normal levels are also the most average levels.

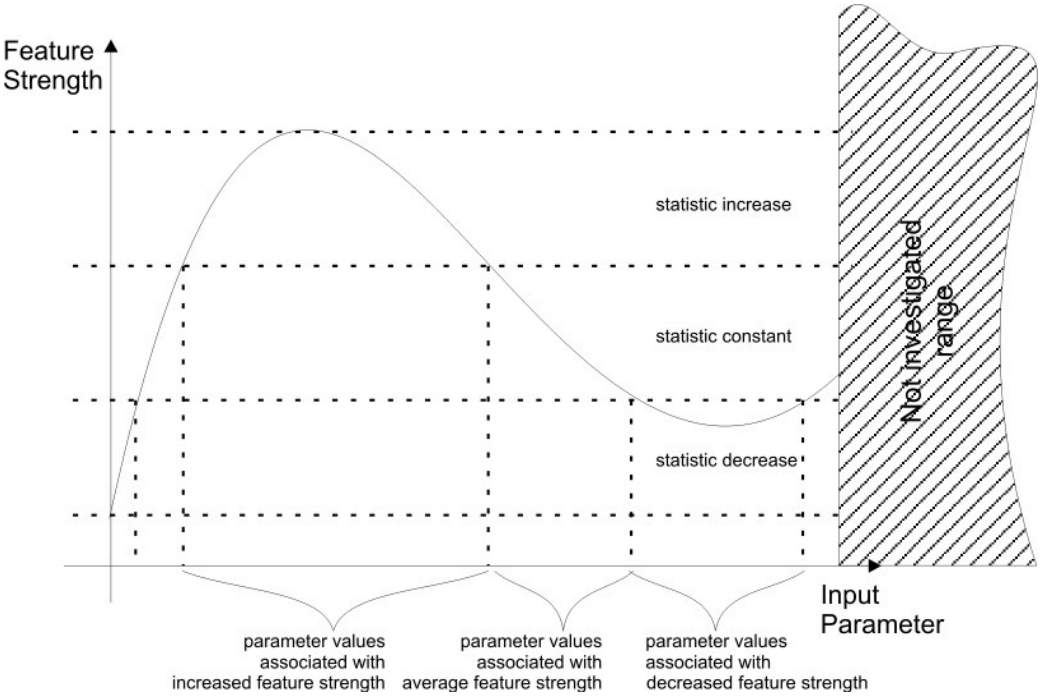


Image 67:: Idea behind the segmentation: The values in the plane (here only one slice) are subdivided into three layers: the topmost layer is the layer associated with statistical increase in feature strength while the lowermost layer is associated with statistical decrease. The layer in between is associated with no remarkable influence.

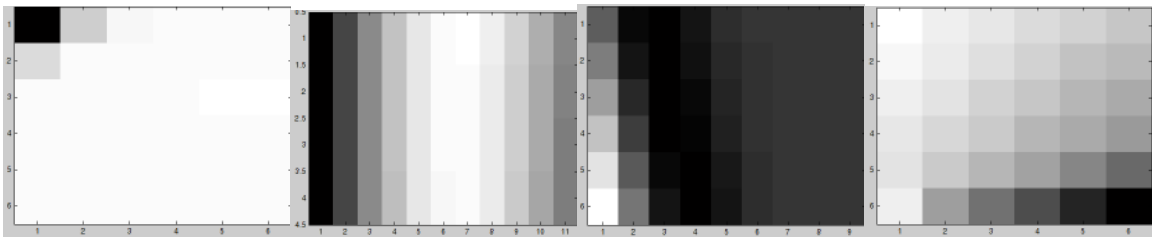


Image 68: Different examples of segmented planes. The segmentation is graded. Feature values most strongly associated with a specified demand are white.

Segmentation of features being marked as to be ignored are always empty (black). After segmentation the planes are used to reconstruct the 7 dimensional parameter space, vividly portrayed in image 44. During this process quality results are generated for far many more than the original 81 parameter vectors which is the main advantage of this method. The mathematical expression of the performed segmentation and the inference can be expressed as:

dimension sizes: $$D \in \mathbb{N}^7 = \begin{pmatrix} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \end{pmatrix}^T \quad (82)$$

index of feature: $$i \in \mathbb{N}\{1...7\} \quad (83)$$

index of input dimension 1: $$A \in \mathbb{N}\{1..7\} \tag{84}$$

index of input dimension 2: $$B \in \mathbb{N}\{1...7\} / \{A\} \tag{85}$$

set of dimensions for WP's: $$C = \mathbb{N}\{1...7\} / \{A \cup B\} \tag{86}$$

element n of set C $$C_n \in C \tag{87}$$

highest value for element n $$\hat{C}_N := \max(C) \tag{88}$$

highest value for dimension 1 $$\hat{A} := 7 \tag{89}$$

highest value for dimension 2 $$\hat{B} := 6 \tag{90}$$

Vector to parameter dereferencing is done by one of the vectors $\vec{H}_?$. See table 1 for help.

$$\begin{aligned}
\vec{H}_1 &= \{10,14,18,22,26,30\} \\
\vec{H}_2 &= \{0,6,12,18,24,30\} \\
\vec{H}_3 &= \{0,6,12,18,24,30\} \\
\vec{H}_4 &= \{10,14,18,22,26,30\} \\
\vec{H}_5 &= \{100,150,200,250,300,350,400,450,500\} \\
\vec{H}_6 &= \{0.1,0.4,0.7,1\} \\
\vec{H}_7 &= \{0,3,6,9,12,15,18,21,24,27,30\}
\end{aligned} \tag{91}$$

hypermatrix of result space $$\overset{\#}{R}_i \subset \mathbb{R}^{D_1 \times D_2 \times D_3 \times D_4 \times D_5 \times D_6 \times D_7} \tag{92}$$

$$\begin{aligned}
&\overset{\#}{R}_i = r_i\left(H_1(p), H_2(q), H_3(w), H_4(j), H_5(k), H_6(l), H_7(m)\right) | \\
&p \in \mathbb{N}\{1...D_1\}, q \in \mathbb{N}\{1...D_2\}, w \in \mathbb{N}\{1...D_3\}, \\
&j \in \mathbb{N}\{1...D_4\}, k \in \mathbb{N}\{1...D_5\}, l \in \mathbb{N}\{1...D_6\}, m \in \mathbb{N}\{1...D_7\}
\end{aligned} \tag{93}$$

Effectively, the function r_i is the artificial neural network that contains the information how the i^{th} feature depends on the input parameters. The hypermatrix $\overset{\#}{R}_i$ is a seven dimensional structure serving as a function cache.

Plane of feature i: $$\overset{\#}{P}_i(A,B)=\left\{\begin{matrix}p=1..\hat{A}\\q=1..\hat{B}\end{matrix}\right\}\frac{1}{7!}\sum_{w=1}^{\hat{C}_1}\sum_{j=1}^{\hat{C}_2}\sum_{k=1}^{\hat{C}_3}\sum_{l=1}^{\hat{C}_4}\sum_{m=1}^{\hat{C}_5}\overset{\#}{R}_i\left(\left\{\begin{matrix}p\to A\\q\to B\\w\to C_1\\j\to C_2\\k\to C_3\\l\to C_4\\m\to C_5\end{matrix}\right\}\right) \tag{94}$$

Segmentation function for feature i $$f_i\left(\overset{\#}{v}\right)\to\overset{\#}{s} \tag{95}$$

The matrices v and s have the same number of dimensions. The function f_i can be implemented in one of the following variants:

$$f_i()\in\left\{f_L(),f_C(),f_H()\right\} \tag{96}$$

normed content of a plane $$\overset{\#}{X}=\frac{\overset{\#}{P}_i-\min\left(\overset{\#}{P}_i\right)}{\max\left(\overset{\#}{P}_i\right)-\min\left(\overset{\#}{P}_i\right)} \tag{97}$$

decrease detector: $$f_L\left(\overset{\#}{X}\in\mathbb{R}^{m\times n}\right)=\left\{\begin{matrix}i=1..n\\j=1..m\end{matrix}\right\}\max\left(1-3\cdot\overset{\#}{X}(i,j),0\right) \tag{98}$$

const detector: $$f_C\left(\overset{\#}{X}\in\mathbb{R}^{m\times n}\right)=\left\{\begin{matrix}i=1..n\\j=1..m\end{matrix}\right\}1-3\cdot\min\left(\left|1-2\cdot\overset{\#}{X}(i,j)\right|,\frac{1}{3}\right) \tag{99}$$

increase detector: $$f_H\left(\overset{\#}{X}\in\mathbb{R}^{m\times n}\right)=\left\{\begin{matrix}i=1..n\\j=1..m\end{matrix}\right\}\max\left(3\cdot\overset{\#}{X}(i,j)-2,0\right) \tag{100}$$

segmentation result $$\overset{\#}{S}(A,B)=f_i\left(\overset{\#}{P}(A,B)\right) \tag{101}$$

The hypermatrix R_i is not available right away but is stored inside the artificial network. It is equal to evaluate the hypermatrix or the neural network at the same parameter vector. The method using planes is more favorable than the immediate evaluation of the hypermatrix R, because the grading of the vectors is finer. The sum of the seven hypermatrices can only total up to seven points whereas the plane method allows additional six evaluations in different contexts that bill up to a total maximum score of 49.

3.16 Interrogation Module

In order to evaluate the many diagrams a special interrogation module was developed. The idea behind it was the ability to express one's own assumptions about a good therapy closely to

the human language. The machine then segments all the diagrams according to the fulfillment of the request. These are partial results that take into account only the two dimensions in the diagram and ignore the other five. After segmentation the many 2D data arrays form quality maps that must be retrofitted into the original 7 dimensional results space. In such a space each coordinate vector is the parameter vector and the value in that 7D grid is the fitness of the parameter vector. The higher the value the better it suits the initially stated requirements.

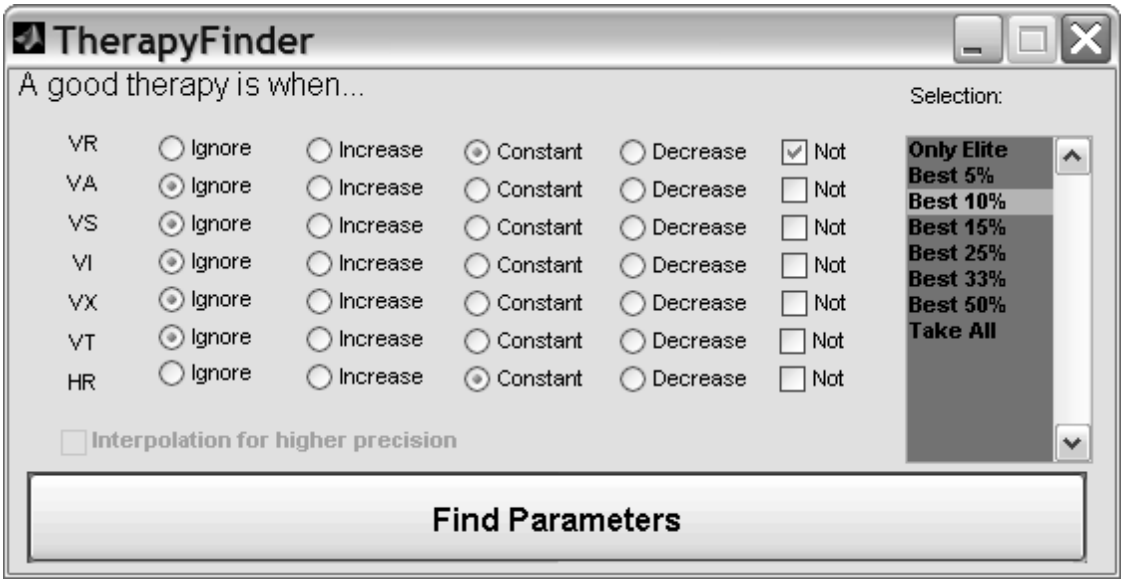


Image 69: Interface to the therapy finder module.

Often it is not important to see the many thousands of parameter vectors all sorted by the final score but it is only needed to obtain the best few that can be tested on rats. Before segmentation and before inference it is possible to specify the fraction of the maximal score that will appear in the final list.

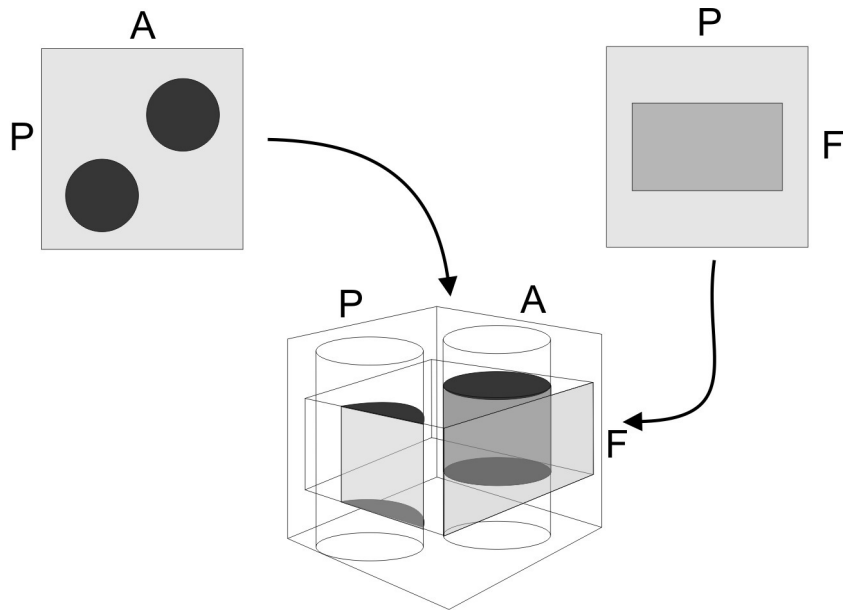


Image 70: Example of inference: two segmented planes form sets that are overlapping in a space of higher dimension. The intersections of the planes are the parameter vectors of interest (red).

4 Results

In this section the immediate results including tables and diagrams are presented. Because so many different methods were used this chapter is subdivided. The synergetic discussion of the results takes place in chapter 5.

4.1 Cluster Vectors

The following tables have been done with the T_{ON+} T_{OFF} method. Refer to chapter 3.11.1.

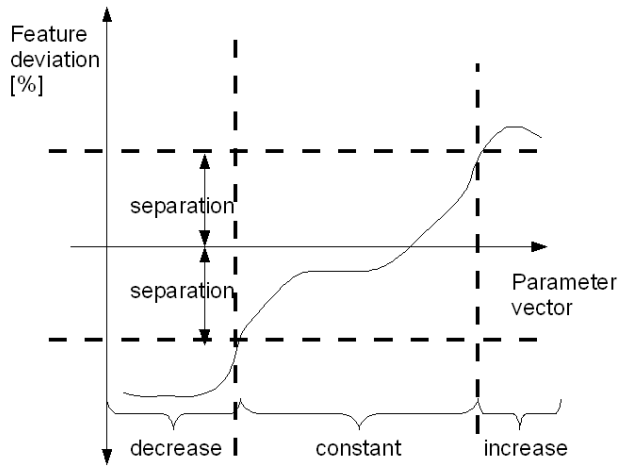


Image 71: Interpretation of the separation level

4.1.1 Ventilation Rate

Table 24: Ventilation Rate, Phase 1, separation 6.3%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	15	19	212	0,58	23
constant	10,7	25	328	0,59	13
increase	19,25	14	282	0,63	24

Table 25: Ventilation Rate, Phase 2, separation 4,2%

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	15,1	19,1	310	0,60	19, 5
constant	15,8	18,6	330	0,5	15
increase	13,1	21,8	206	0,40	18,8

Table 26: Ventilation Rate, Phase 3, separation 3.25%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	14	19	397	0,58	11,6
constant	13	22	252	0,55	17,86
increase	16	18,5	320	0,73	18,79

4.1.2 Ventilation Amplitude

Table 27: Ventilation Amplitude, Phase 1, separation 4%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	14,9	19,54	283	0,61	17
constant	12	23,43	340	0,61	13,6
increase	10, 2	25,82	309	0,52	21,36

Table 28: Ventilation Amplitude, Phase 2, separation 1%

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	10,5	25	500	0,25	5
constant	17,5	16,25	407	0,50	7,5
increase	13,9	20,89	288	0,57	18,2

Table 29: Ventilation Amplitude, Phase 3, separation 6.75%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	13,6	21,3	322	0,63	16,7
constant	14	20,8	198	0,28	17,8
increase	17,5	16,5	160	0,3125	16,25

4.1.3 Ventilation Slope

Table 30: Ventilation Slope, Phase 1, separation 1.8%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	14,5	20,0	293	0,57	18,2
constant	11,2	24,6	376	0,75	11
increase	14	20,9	376	0,38	13,75

Table 31: Ventilation Slope, Phase 2, separation 1.4%

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	18,4	15	391	0,50	10,6
constant	14	21	376	0,42	25
increase	12,9	22	297	0,59	17,8

Table 32: Ventilation Slope, Phase 3, separation 4.4%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	12,7	22,9	381	0,67	13,6
constant	14,3	20,3	258	0,58	17,7
increase	14,875	19,5	345	0,46875	18,125

4.1.4 Ventilation Inhalation Time

Table 33: Ventilation Inhalation Time, Phase 1, separation 0%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	10	26	358	0,39	20,7
constant	no cluster	no cluster	no cluster	no cluster	no cluster
increase	15,85	18	301	0,59	16,4

Table 34: Ventilation Inhalation Time, Phase 2

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	14	20,7	300	0,56	17,34
constant	no cluster	no cluster	no cluster	no cluster	no cluster
increase	no cluster	no cluster	no cluster	no cluster	no cluster

Table 35: Ventilation Inhalation Time, Phase 3, separation 4.8%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	17,5	21	375	0,31	16,3
constant	18,7	21	293	1	21,7
increase	12,2	23	280	0,57	18,7

4.1.5 Ventilation Exhalation Time

Table 36: Ventilation Exhalation Time, Phase 1, separation 3.6%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	17,2	16,7	274	0,64	16,4
constant	16,6	17,6	313	0,38	16,9
increase	12,2	23	280	0,57	18,7

Table 37: Ventilation Exhalation Time, Phase 2

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	17,5	16,3	191	0,41	12,5
constant	no cluster	no cluster	no cluster	no cluster	no cluster
increase	13,7	21,1	309	0,58	17,97

Table 38: Ventilation Exhalation Time, Phase 3, separation 3.3%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	14	20.6	307	0,61	17,8
constant	11,6	23,3	253,3	0,3	8,3
increase	19,6	13,6	326	0,7	14

4.1.6 Minute Ventilation

Table 39: Minute Ventilation, Phase 1, separation 2.8%

Phase 1	T_{ON}	T_{OFF}	P	A	F
decrease	15.0	19,4	351	0,5	19,8
constant	12.4	22,8	223	0,6	16,1
increase	15.2	19,3	438	0,6	12,5

Table 40: Minute Ventilation, Phase 2, separation 1.5%

Phase 2	T_{ON}	T_{OFF}	P	A	F
decrease	17,8	15,7	353,6	0,56	13,6
constant	13	22	340	0,57	15
increase	12,1	23,1	297	0,59	14,2

Table 41: Minute Ventilation, Phase 3, separation 12.8%

Phase 3	T_{ON}	T_{OFF}	P	A	F
decrease	14	25,5	500	1	10
constant	13,5	21,4	263	0,59	17,1
increase	17,5	16,5	190	0,625	17,5

4.1.7 Heart Rate

Table 42: Heart Rate, Phase 1, separation 0.2%

Phase1	T_{ON}	T_{OFF}	P	A	F
decrease	13,5	21,5	330	0,64	16,9
constant	13,1	21,9	328	0,53	11,9
increase	15,5	18,7	273	0,59	20

Table 43: Heart Rate, Phase 2, separation 0.45%

Phase1	T_{ON}	T_{OFF}	P	A	F
decrease	15,6	18,5	300	0,63	21,2
constant	12,9	22,1	299	0,5	16,1
increase	18,2	15,6	450	0,6	9

Table 44: Heart Rate, Phase 3, separation 0.7%

Phase1	T_{ON}	T_{OFF}	P	A	F
decrease	16,3	19,5	335	0,625	18,3
constant	13,5	21,3	271	0,54	16,2
increase	16,1	19,9	338	0,65	22

4.2 Visualization Diagrams (VNS Effects Map)

The following diagrams are sorted by phases and are generated from the three compacted data sets. Although they haven't been used with TherapyFinder, yet, they are predestined for this application because the data is minimally modified and shows more detail than the tendency diagrams generated by the artificial neural network. Hitherto the presented material is the most complete VNS effects map.

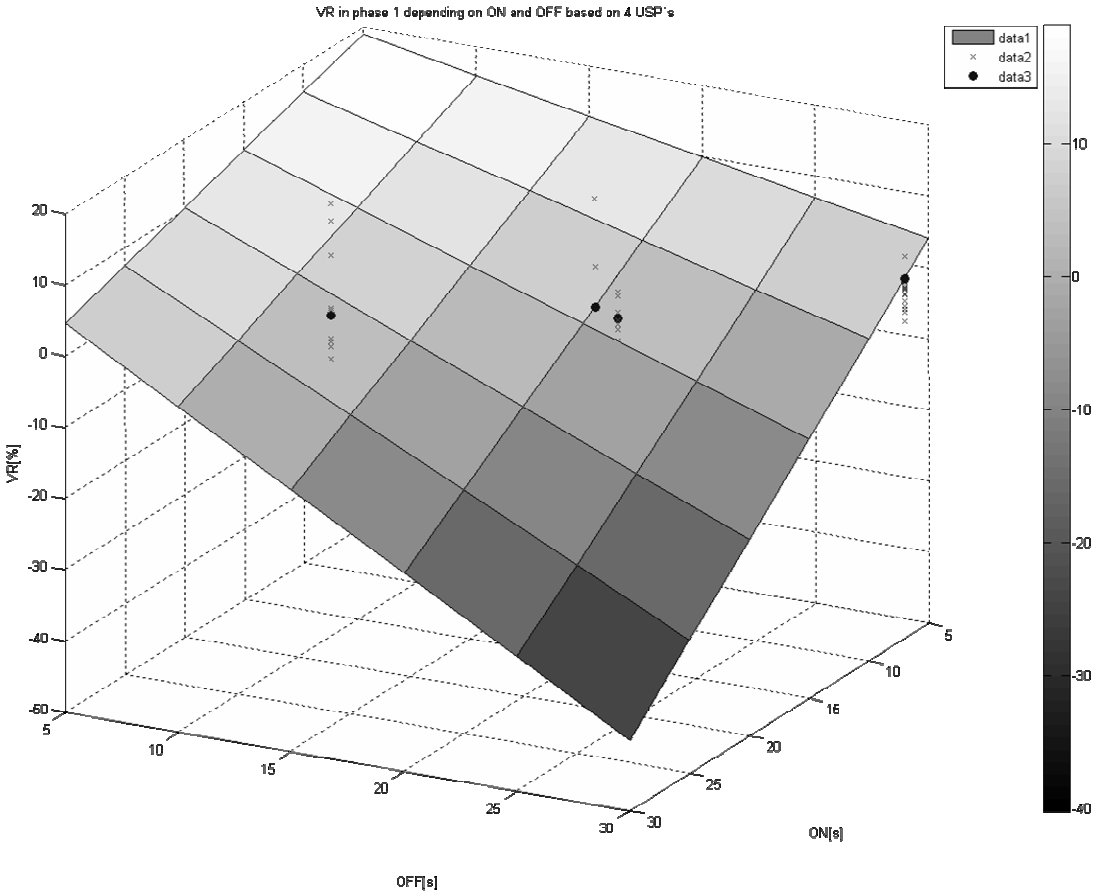
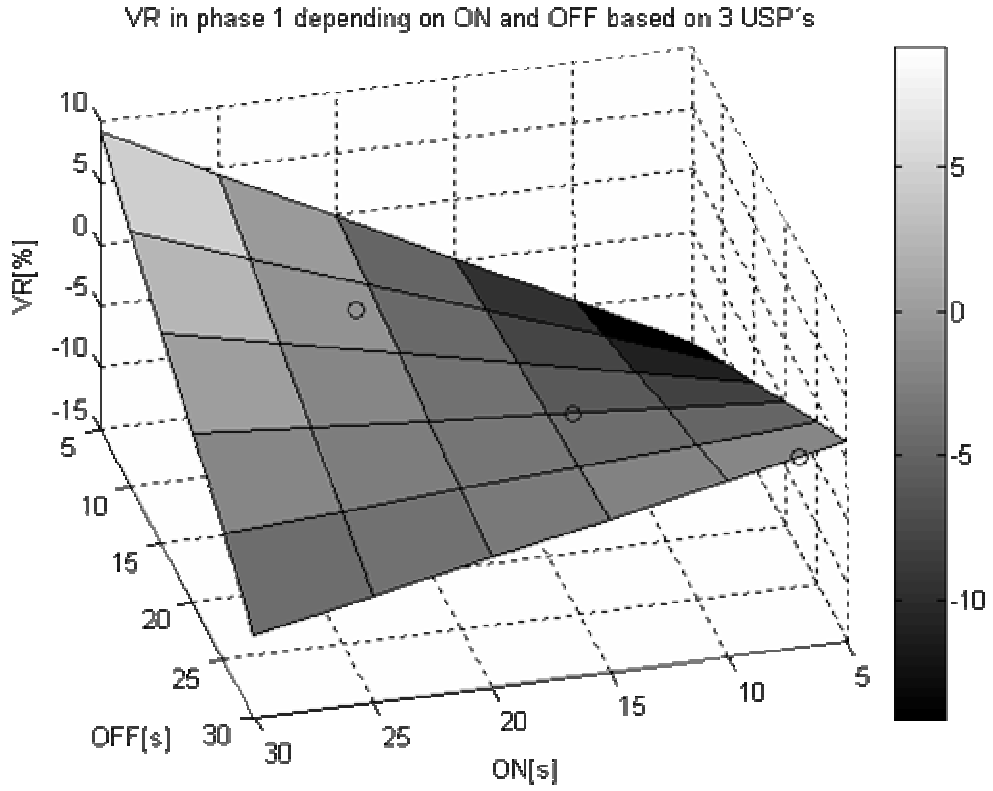
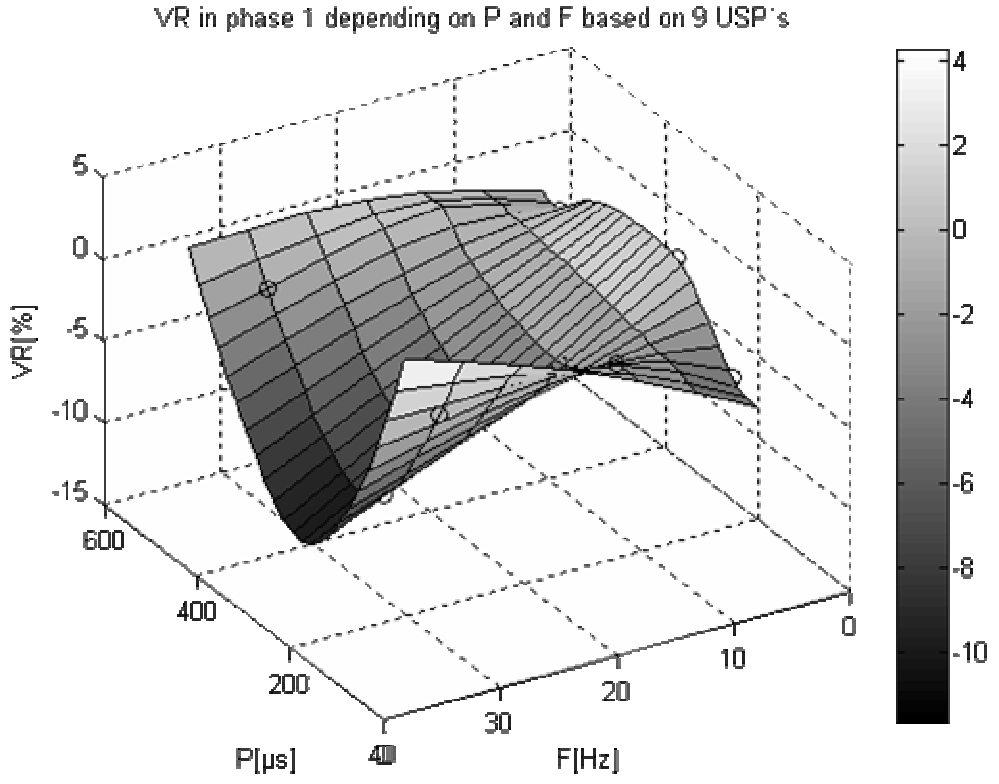


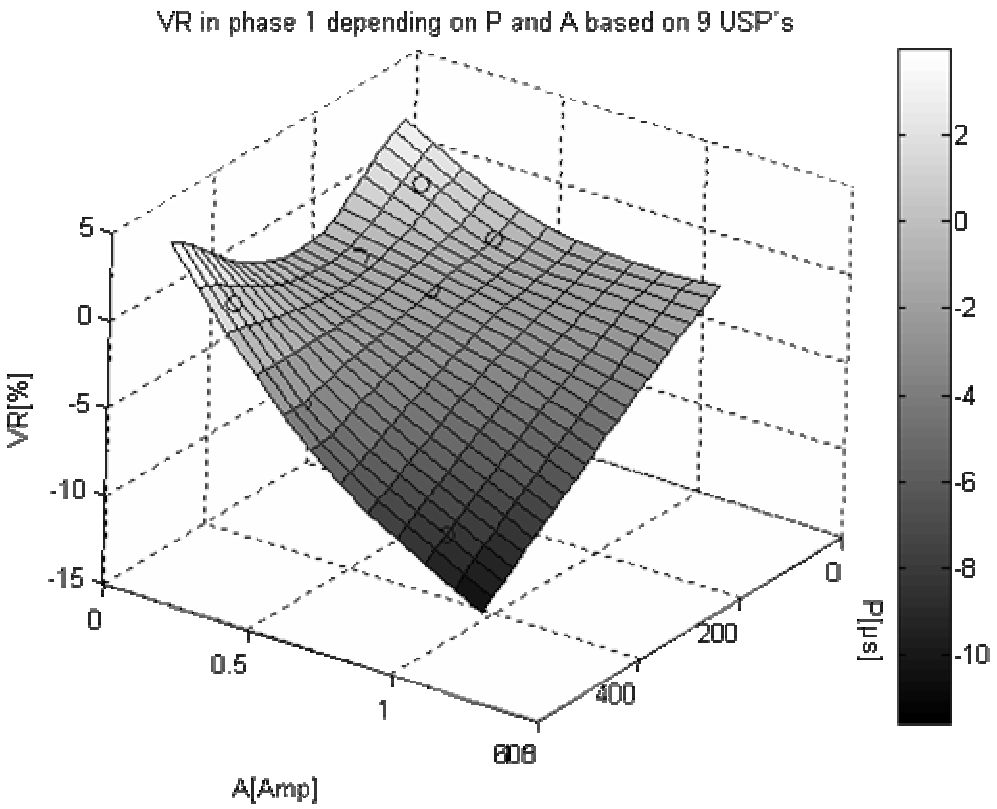
Image 72: If only two of the five parameters are used for drawing the dependency diagram, then there is a surplus of data at the available coordinates (crosses). This data is mean-averaged and then used for the surface generation (circles). In this example there are only four unique supporting vectors for the surface which are placed along an axis. This largely means, that the steady slope portrayed in this diagram is probably untrue but the data does not support a better prediction.

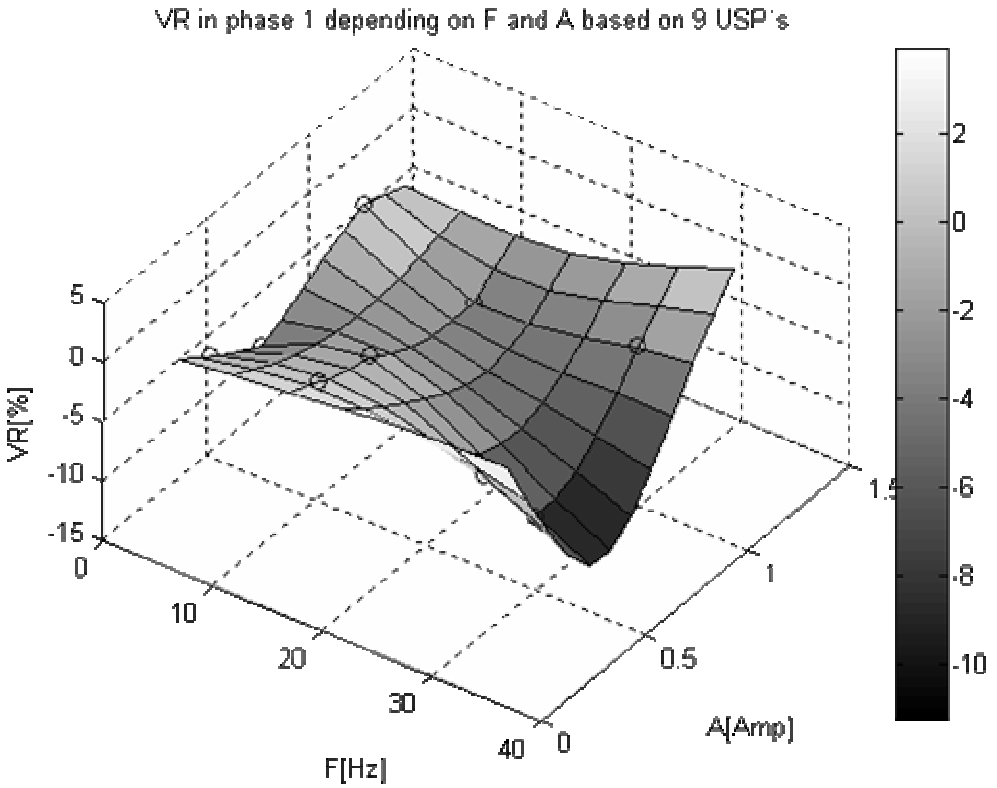
4.2.1 Phase 1 (Effects during stimulation)

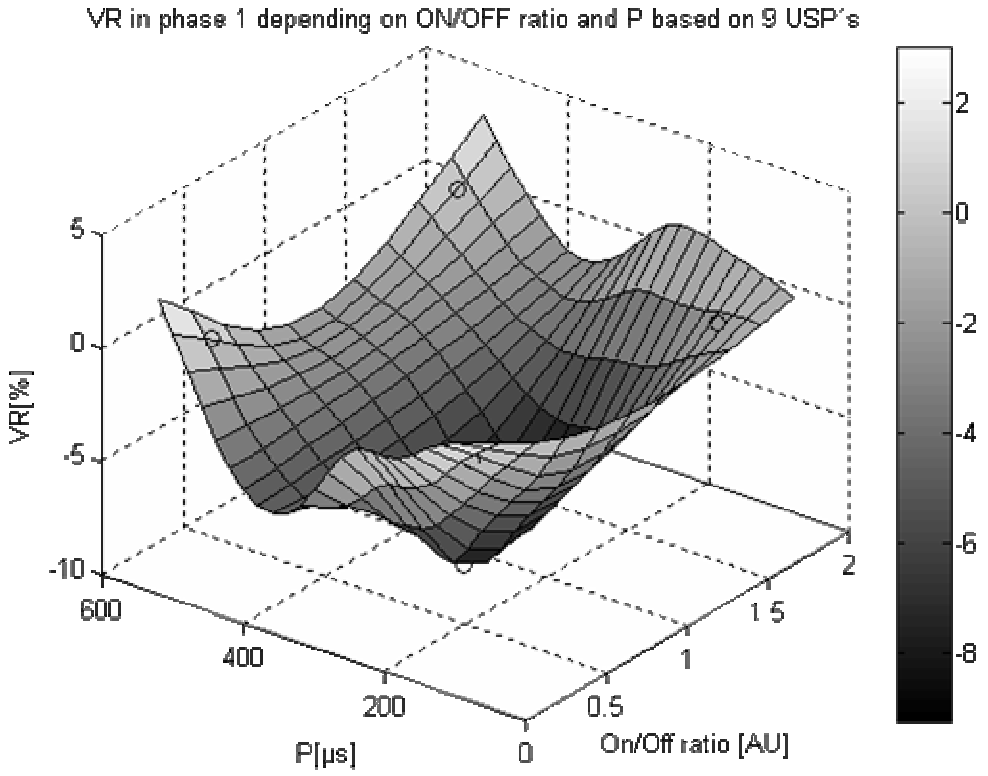
4.2.1.1 Ventilation Rate

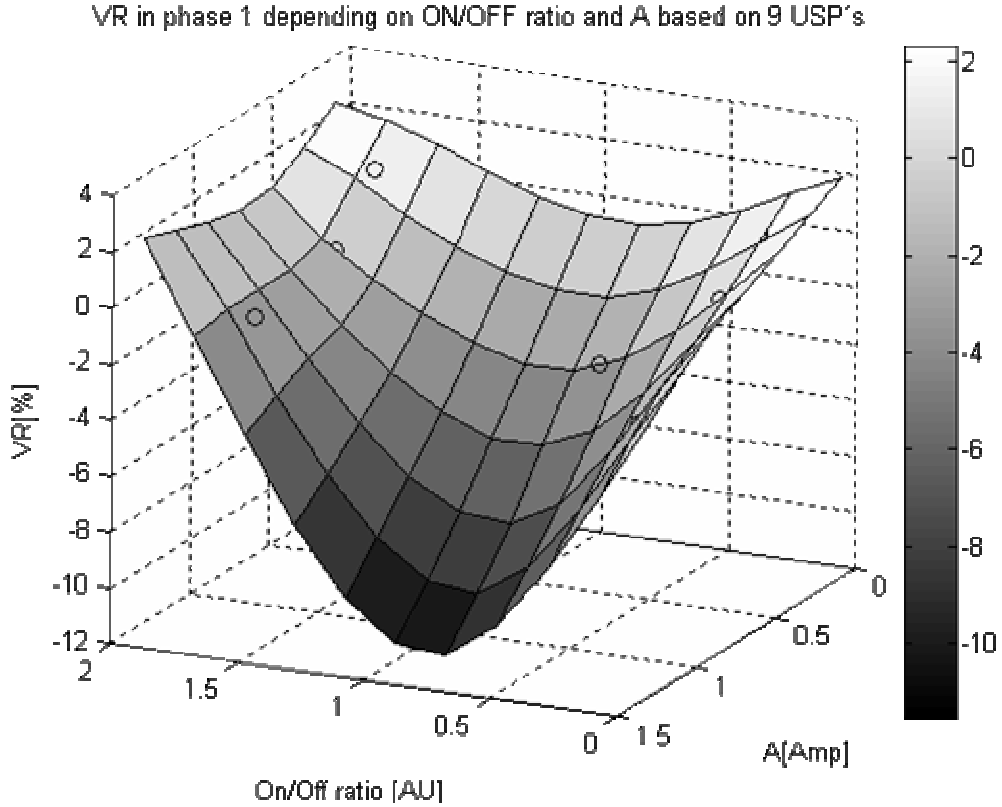


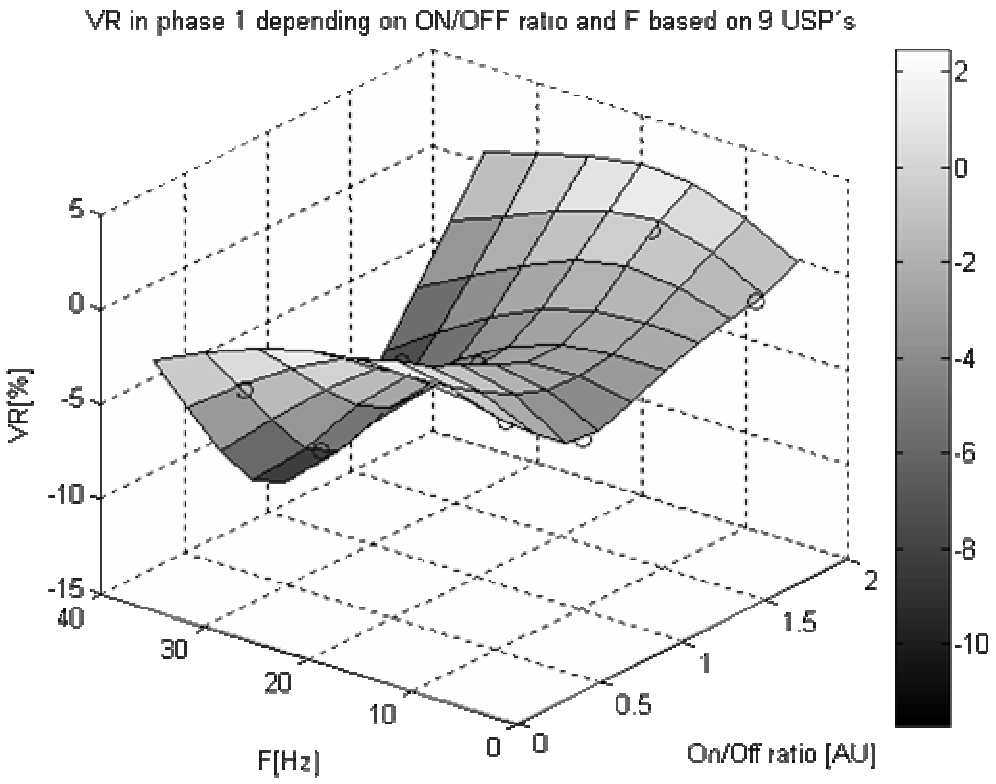



VR in phase 1 depending on P and A based on 9 USP's
VR[%]
5
0
-5
-10
-15
0
0.5
1
A[Amp]
0
200
400
600
P[µs]
2
0
-2
-4
-6
-8
-10

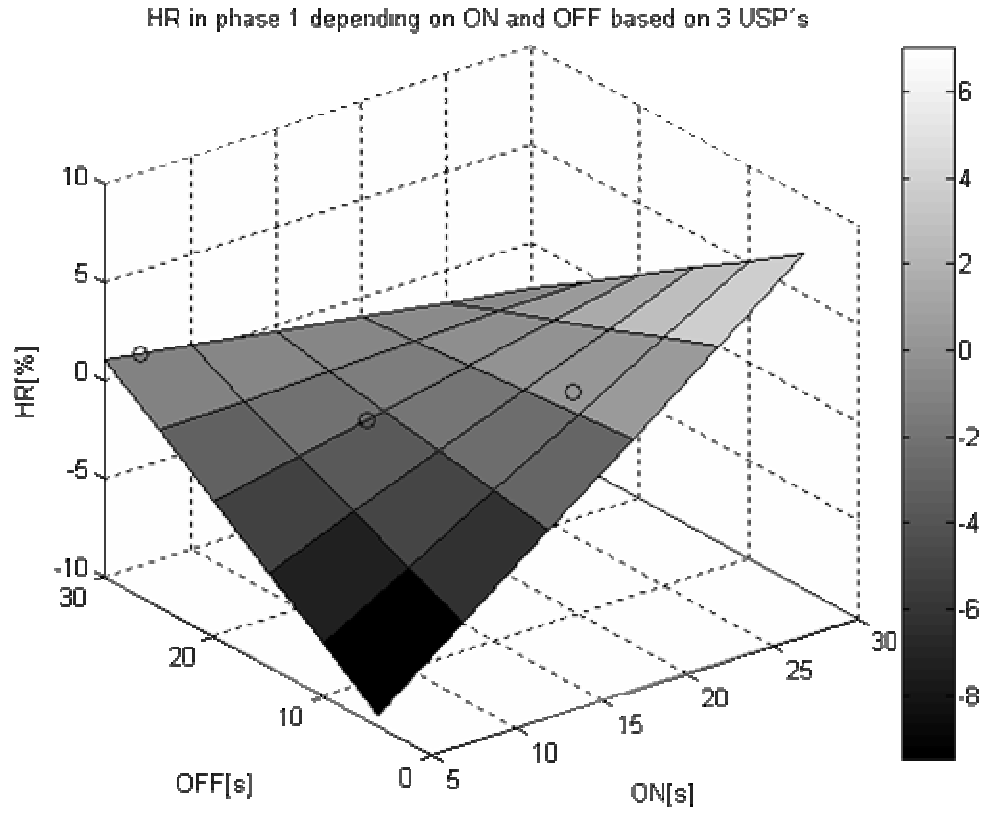

VR in phase 1 depending on F and A based on 9 USP's
VR[%]
5
0
-5
-10
-15
0
10
20
30
40
F[Hz]
0
0.5
1
A[Amp]
2
0
-2
-4
-6
-8
-10

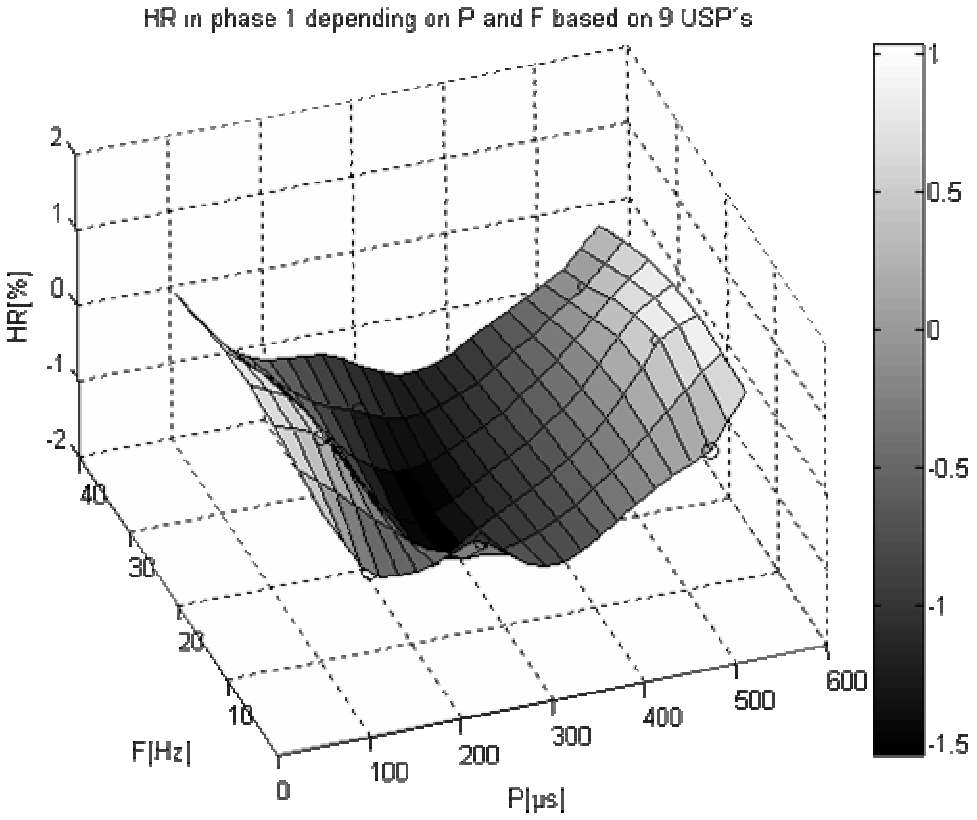

VR in phase 1 depending on ON/OFF ratio and P based on 9 USP´s
VR[%]
5
0
-5
-10
600
400
200
0
P[µs]
2
1.5
1
0.5
0
On/Off ratio [AU]
2
0
-2
-4
-6
-8

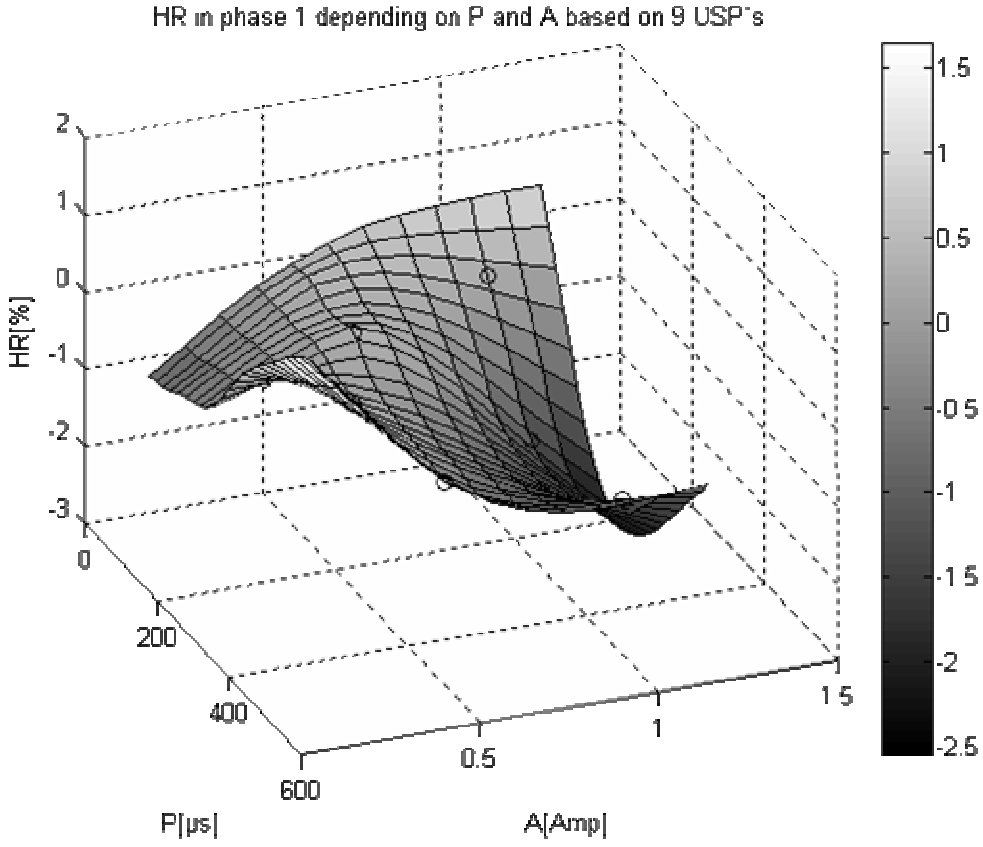

VR in phase 1 depending on ON/OFF ratio and A based on 9 USP´s
VR[%]
4
2
0
-2
-4
-6
-8
-10
-12
2
1.5
1
0.5
0
On/Off ratio [AU]
0
0.5
1
1.5
A[Amp]
2
0
-2
-4
-6
-8
-10

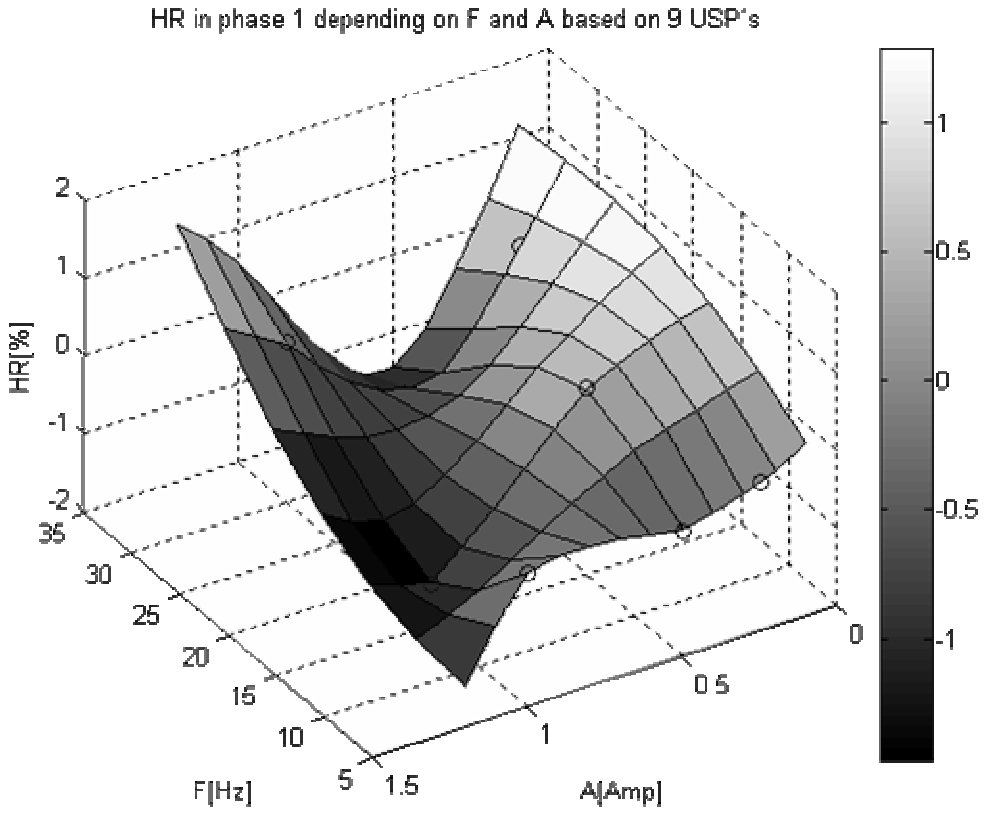


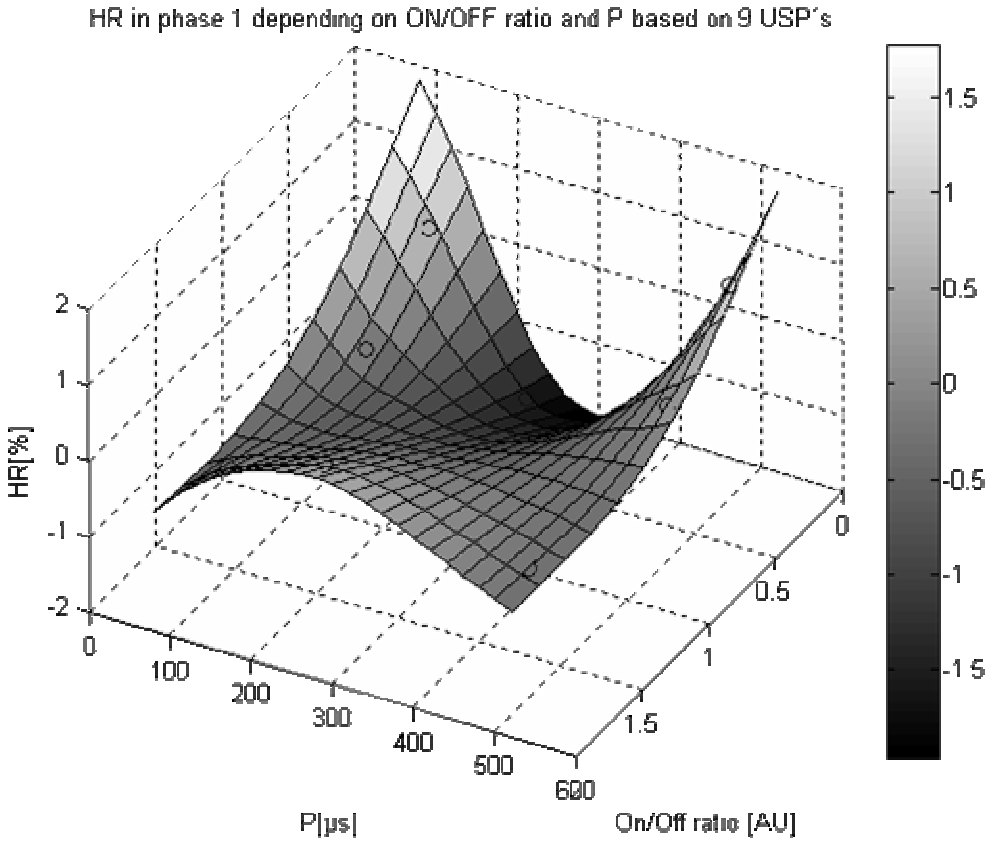
4.2.1.2 Heart Rate

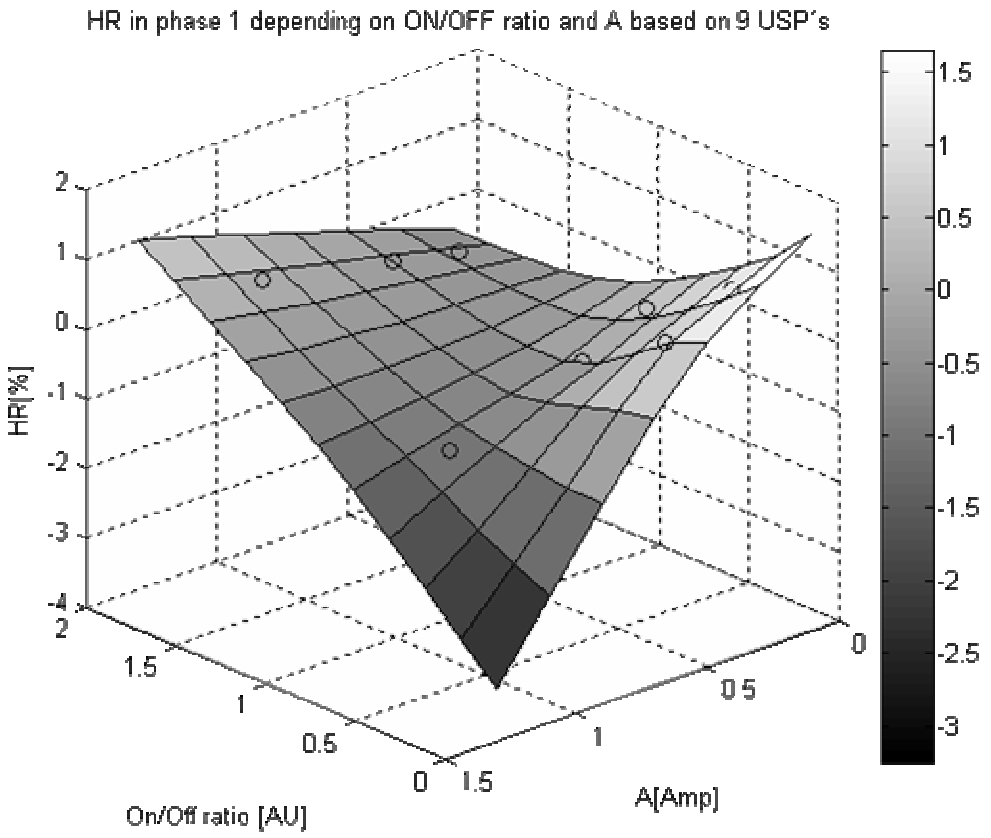


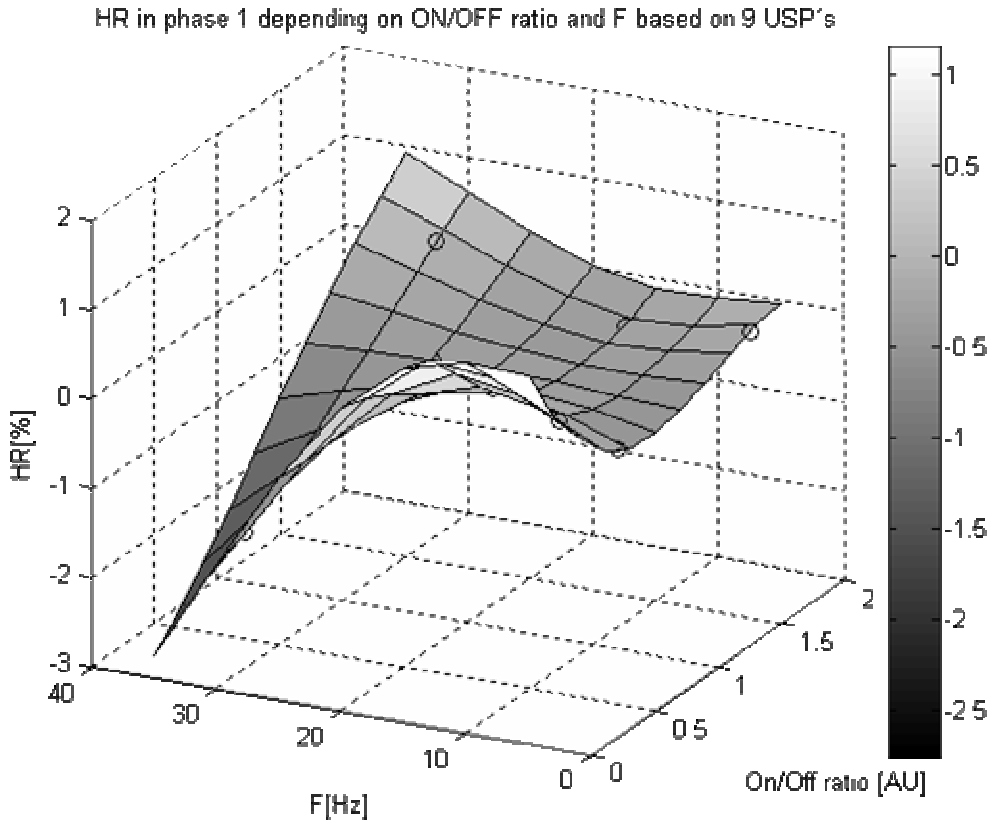

HR in phase 1 depending on P and F based on 9 USP´s
HR[%]
2
1
0
-1
-2
40
30
20
10
F[Hz]
0
100
200
300
400
500
600
P[µs]
1
0.5
0
-0.5
-1
-1.5


HR in phase 1 depending on P and A based on 9 USP´s
HR[%]
2
1
0
-1
-2
-3
0
200
400
600
P[µs]
0.5
1
1.5
A[Amp]
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5

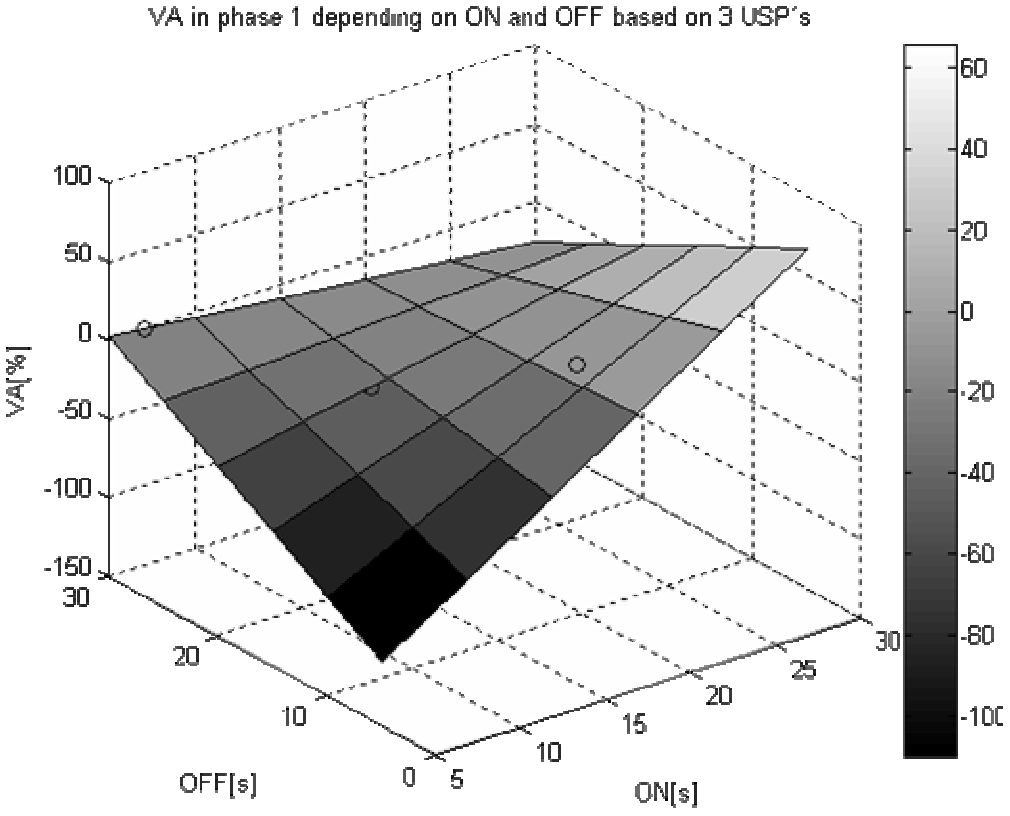

HR in phase 1 depending on F and A based on 9 USP's
HR[%]
2
1
0
-1
-2
35
30
25
20
15
10
5
F[Hz]
1.5
1
0.5
0
A[Amp]
1
0.5
0
-0.5
-1

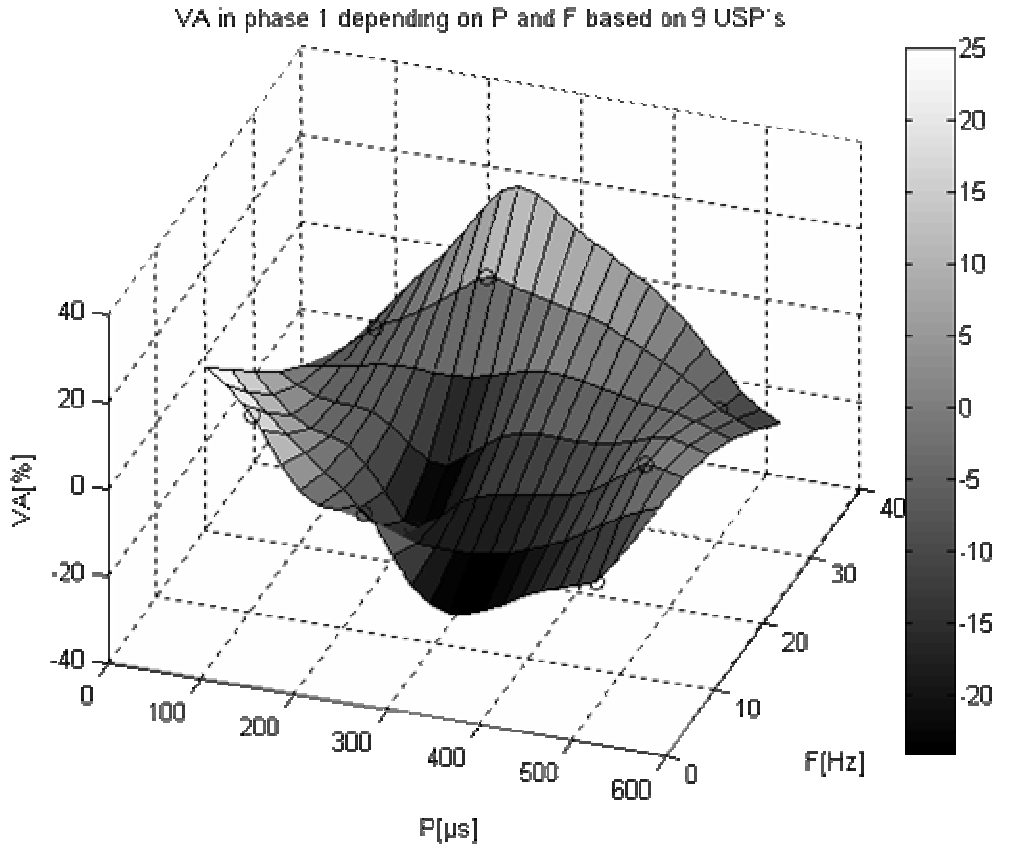

HR in phase 1 depending on ON/OFF ratio and P based on 9 USP´s
HR[%]
2
1
0
-1
-2
0
100
200
300
400
500
600
P[µs]
0
0.5
1
1.5
On/Off ratio [AU]
1.5
1
0.5
0
-0.5
-1
-1.5

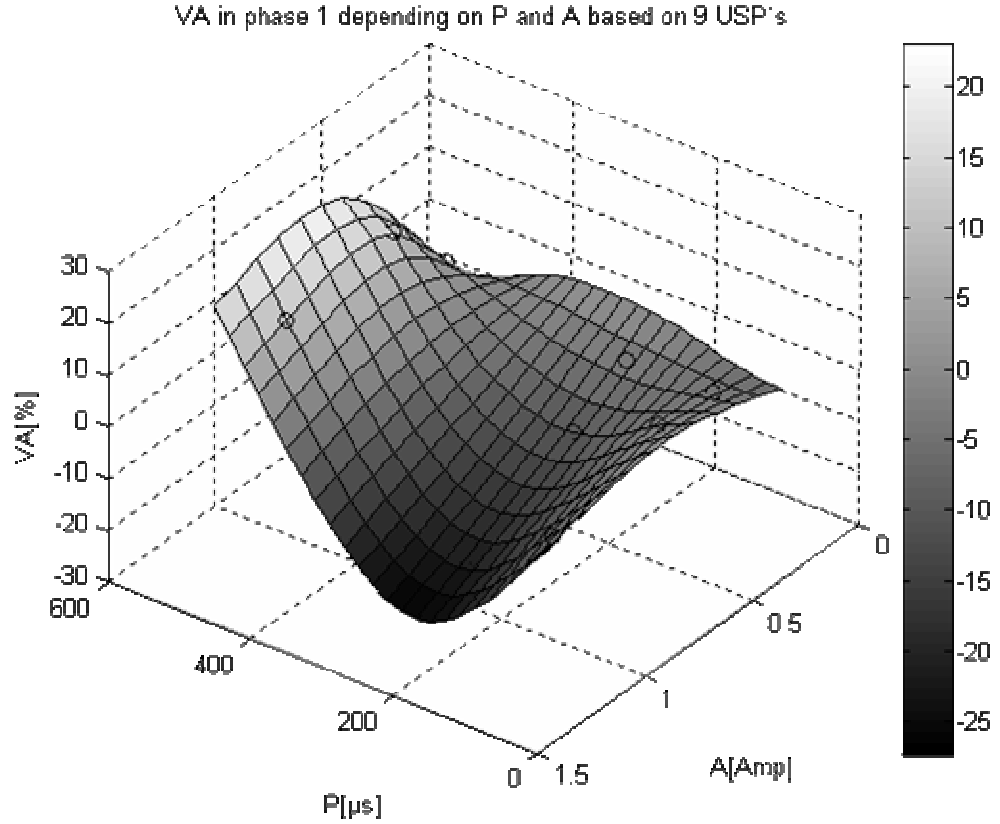

HR in phase 1 depending on ON/OFF ratio and A based on 9 USP´s
HR[%]
2
1
0
-1
-2
-3
-4
2
1.5
1
0.5
0
On/Off ratio [AU]
0
0.5
1
1.5
A[Amp]
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3

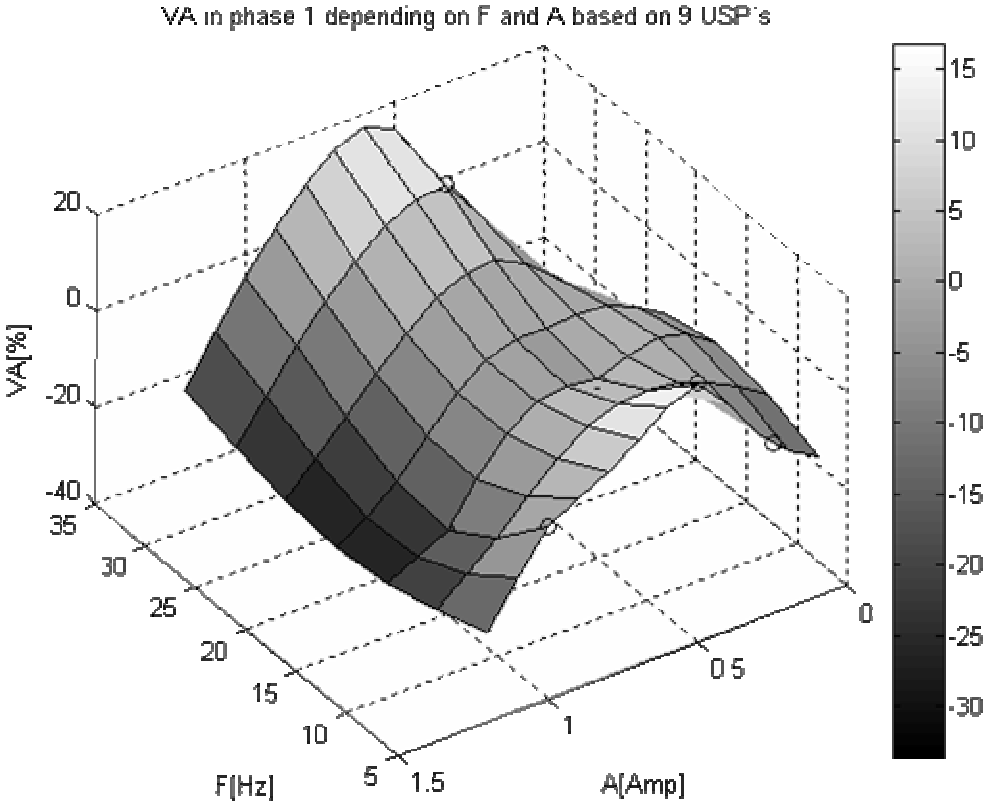

HR in phase 1 depending on ON/OFF ratio and F based on 9 USP´s
HR[%]
2
1
0
-1
-2
-3
40
30
20
10
0
F[Hz]
0
0.5
1
1.5
2
On/Off ratio [AU]
1
0.5
0
-0.5
-1
-1.5
-2
-2.5

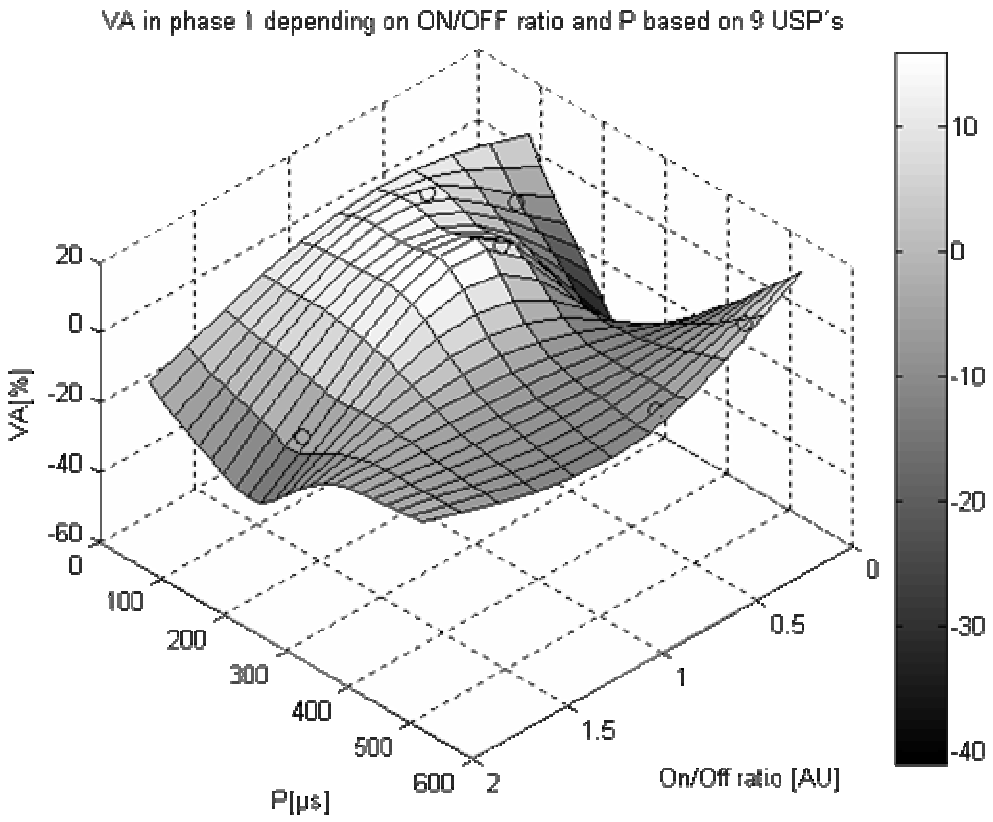
4.2.1.3 Ventilation Amplitude

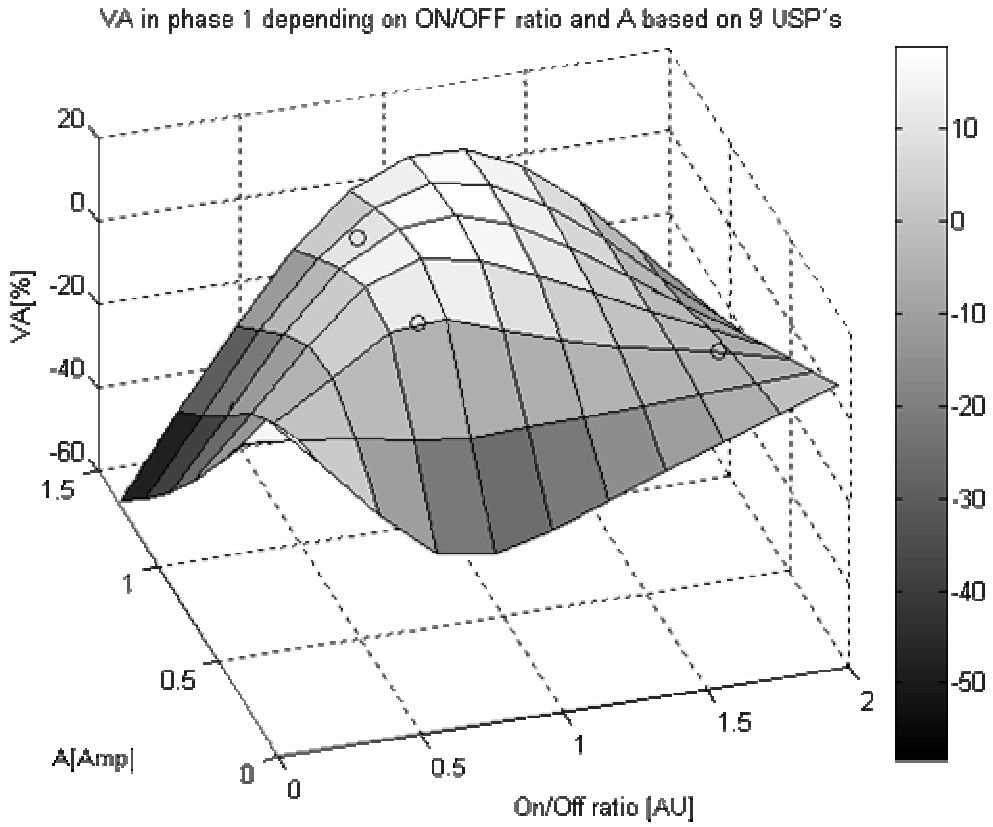


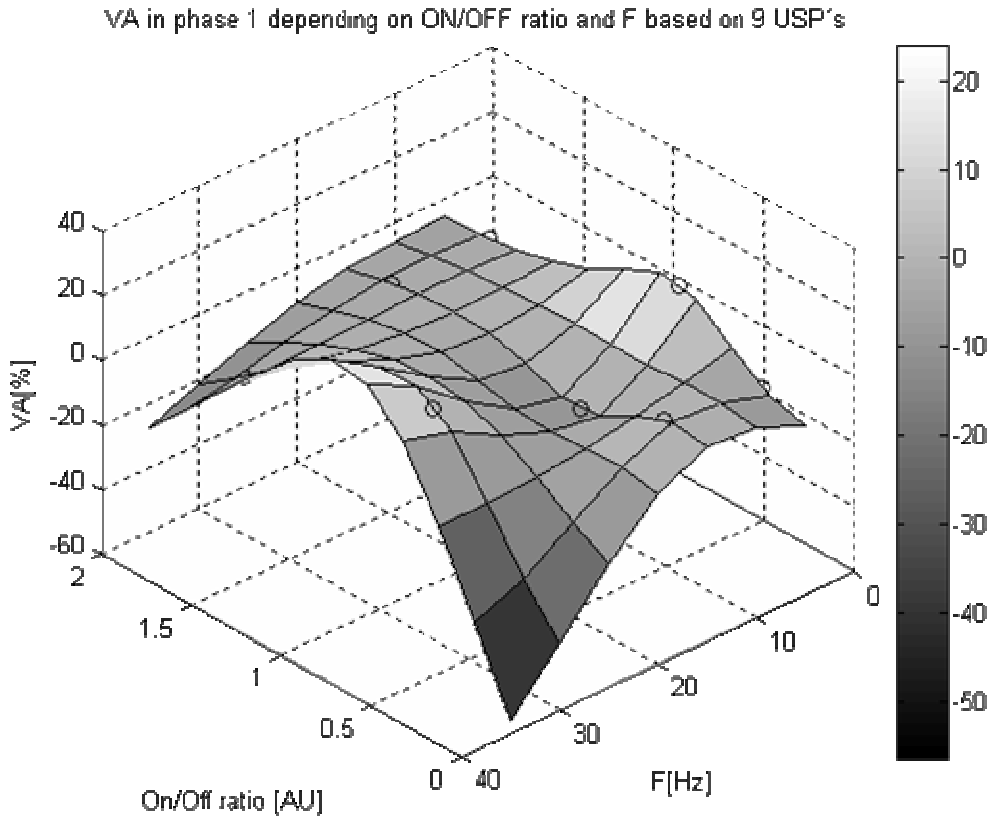



VA in phase 1 depending on P and A based on 9 USP's
VA[%]
30
20
10
0
-10
-20
-30
600
400
200
0
P[µs]
1.5
1
0.5
0
A[Amp]
20
15
10
5
0
-5
-10
-15
-20
-25

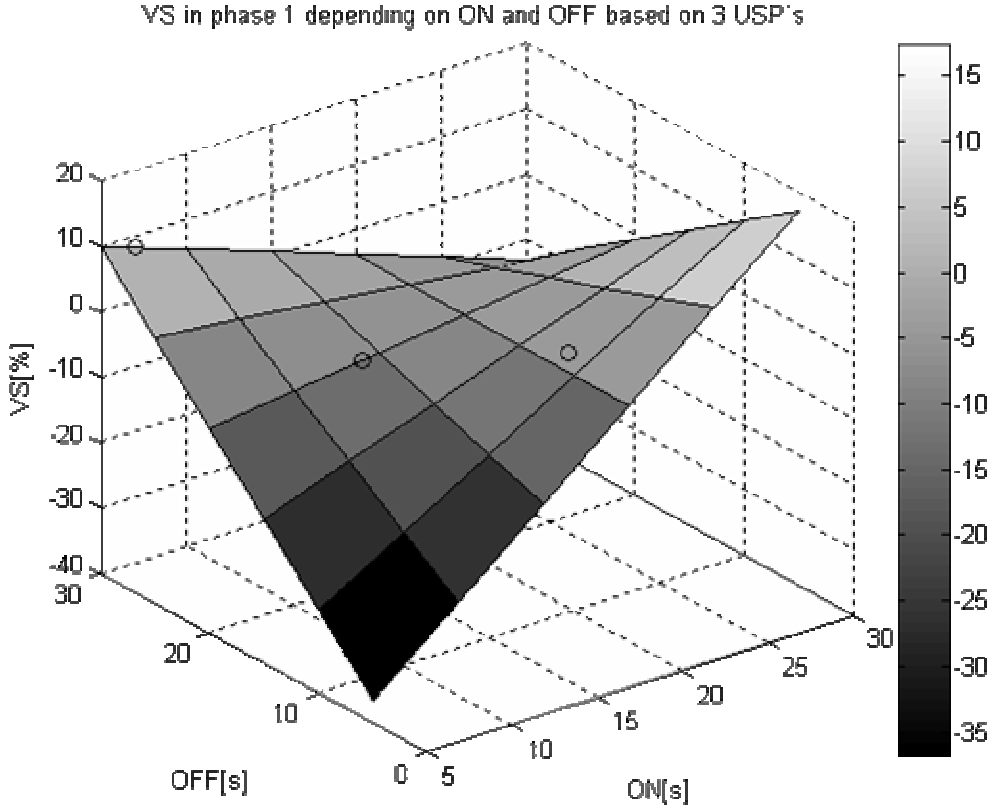

VA in phase 1 depending on F and A based on 9 USP's
VA[%]
20
0
-20
-40
35
30
25
20
15
10
5
F[Hz]
1.5
1
0.5
0
A[Amp]
15
10
5
0
-5
-10
-15
-20
-25
-30

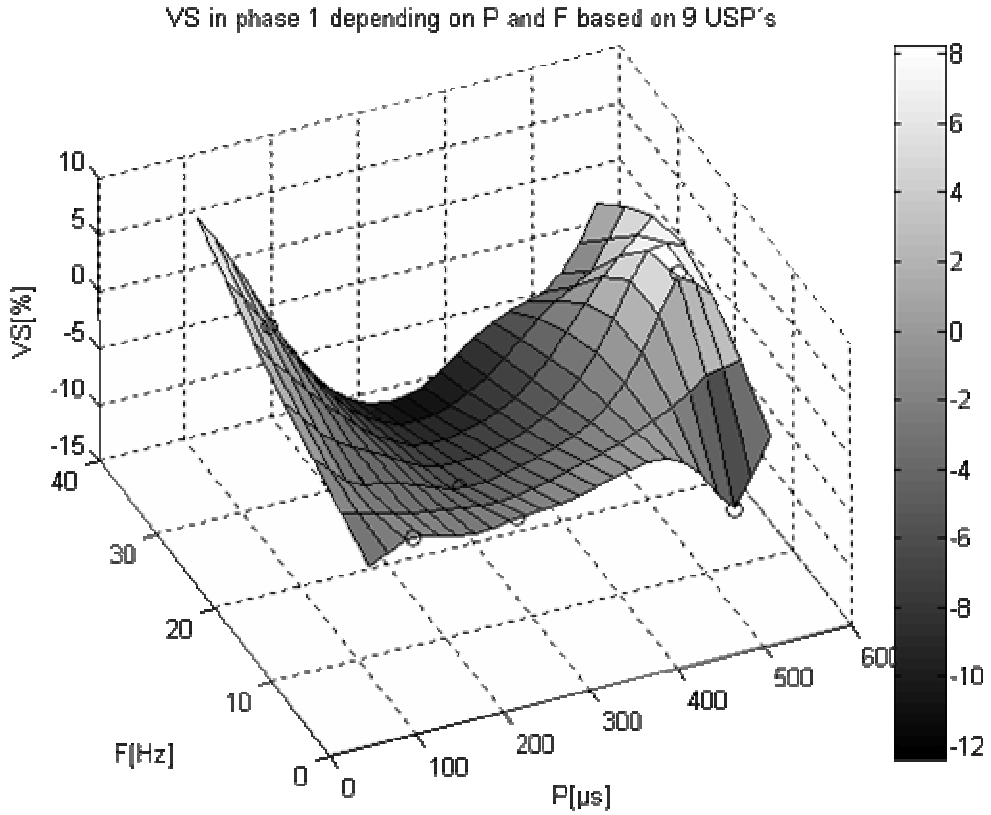

VA in phase 1 depending on ON/OFF ratio and P based on 9 USP´s
VA[%]
20
0
-20
-40
-60
0
100
200
300
400
500
600
P[µs]
0
0.5
1
1.5
2
On/Off ratio [AU]
10
0
-10
-20
-30
-40

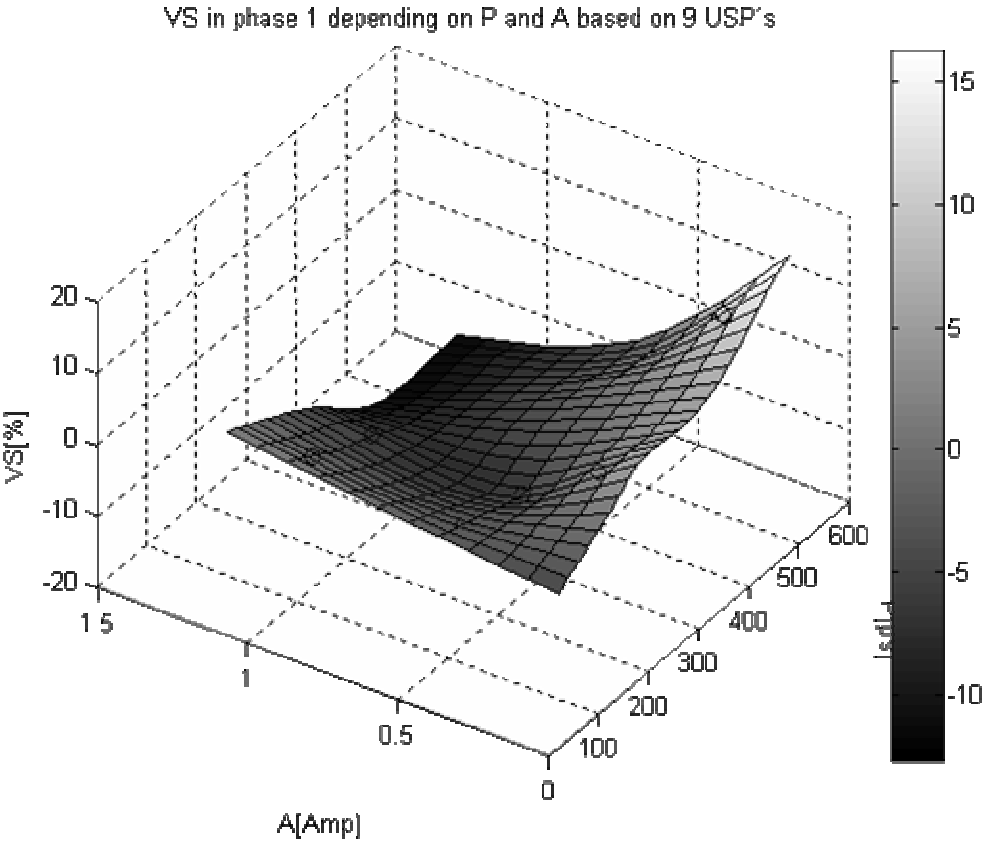

VA in phase 1 depending on ON/OFF ratio and A based on 9 USP´s
VA[%]
20
0
-20
-40
-60
1.5
1
0.5
0
A[Amp]
0
0.5
1
1.5
2
On/Off ratio [AU]
10
0
-10
-20
-30
-40
-50

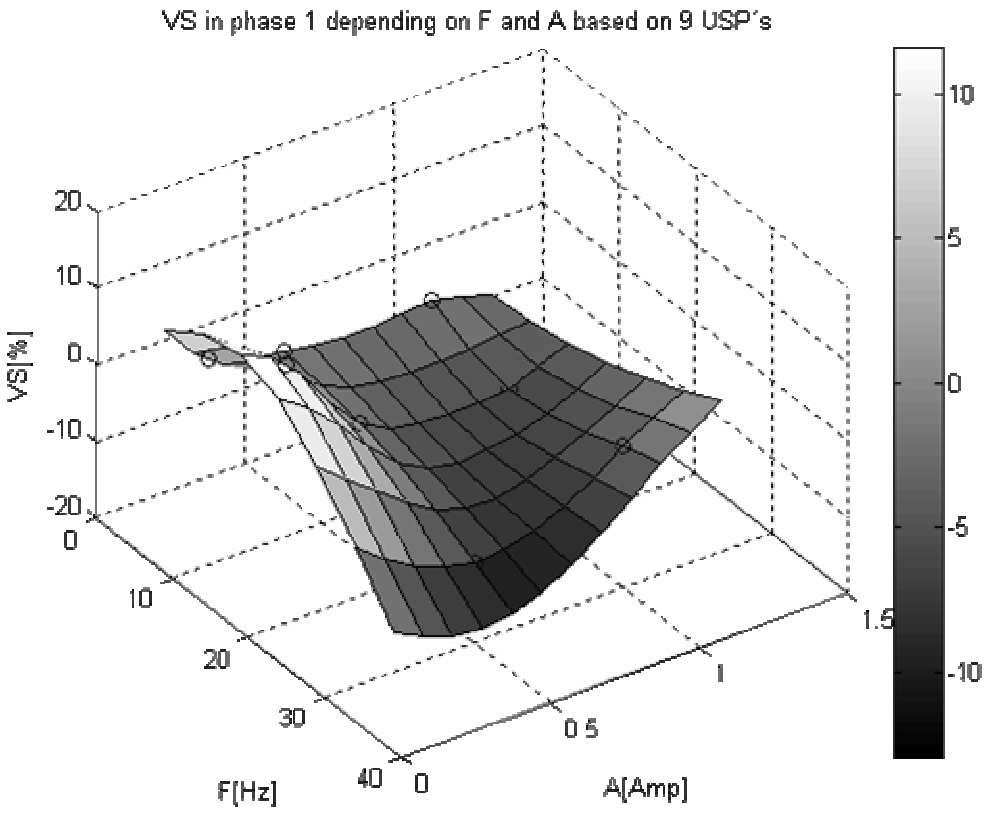


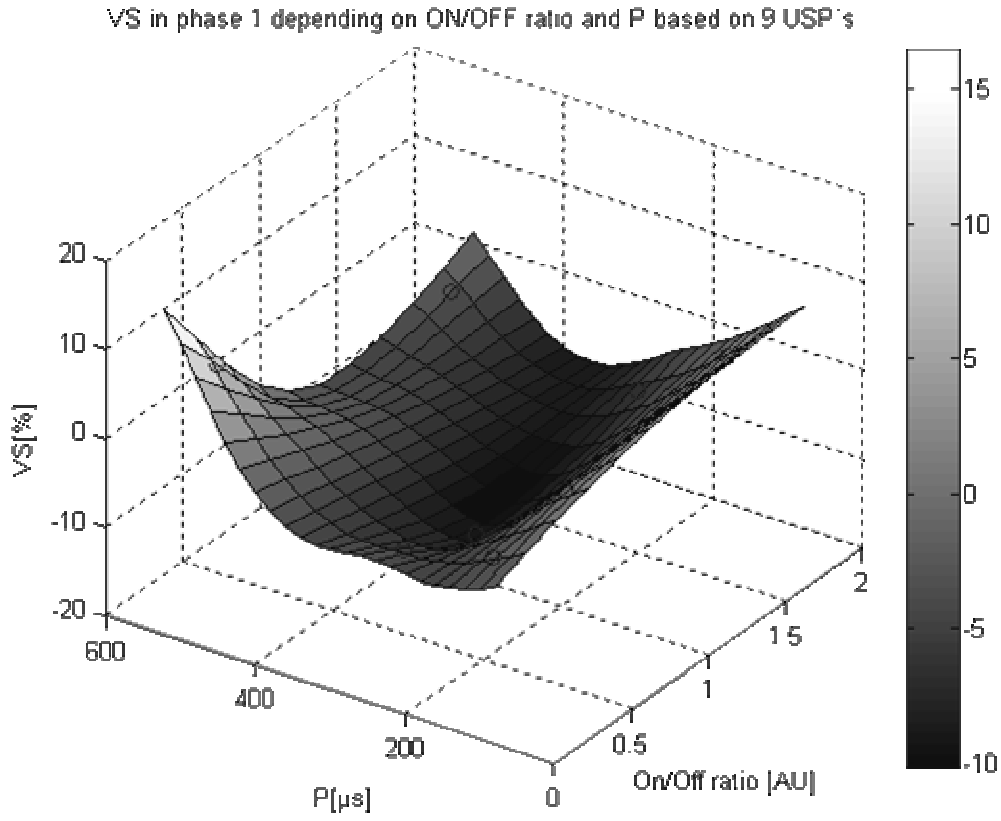
4.2.1.4 Ventilation Slope

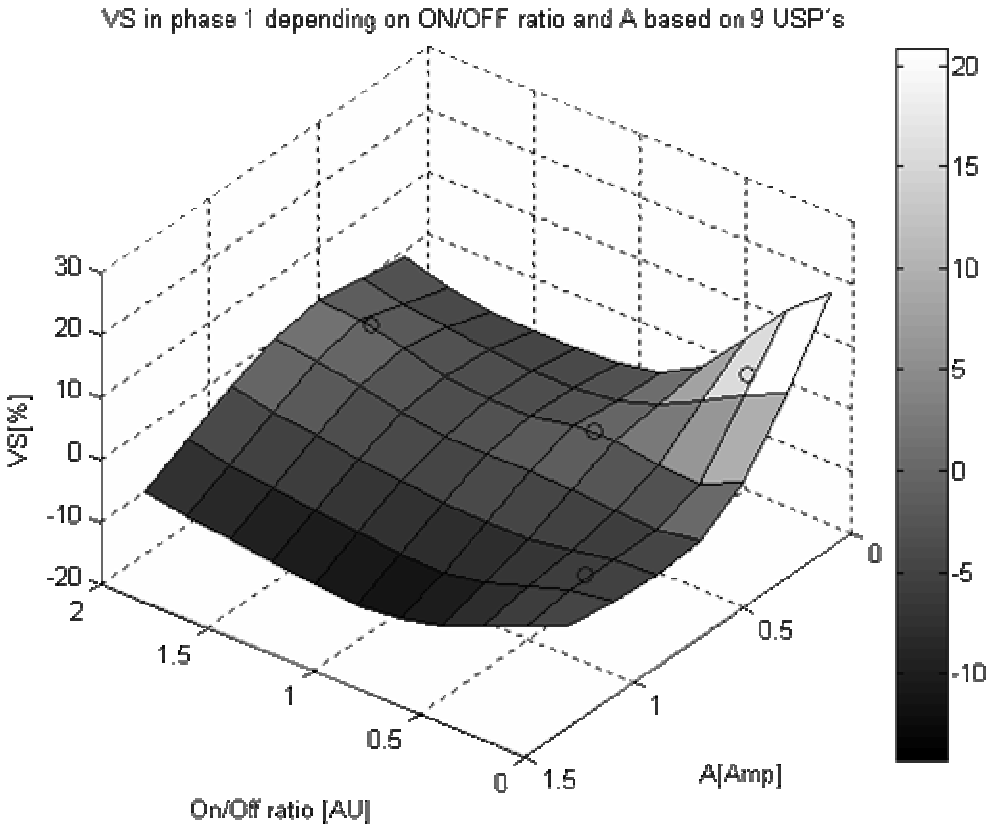


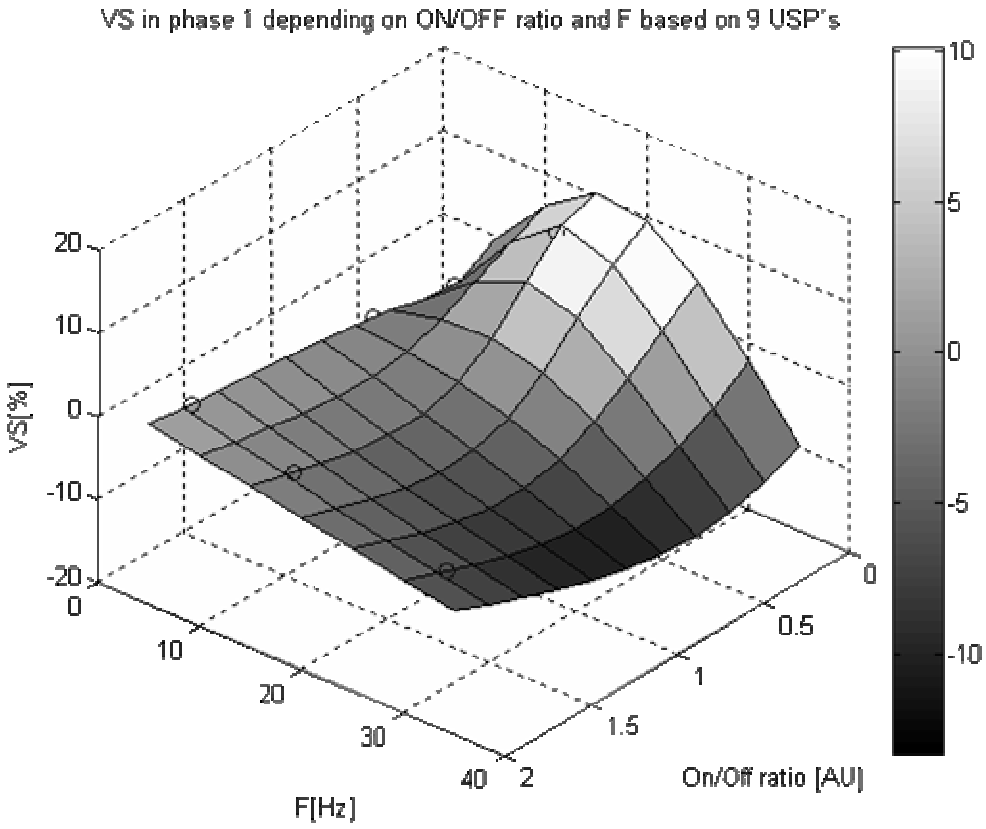

VS in phase 1 depending on P and F based on 9 USP´s
VS[%]
10
5
0
-5
-10
-15
40
30
20
10
0
F[Hz]
0
100
200
300
400
500
600
P[µs]
8
6
4
2
0
-2
-4
-6
-8
-10
-12


VS in phase 1 depending on P and A based on 9 USP´s
VS[%]
20
10
0
-10
-20
1.5
1
0.5
0
A[Amp]
0
100
200
300
400
500
600
15
10
5
0
-5
-10

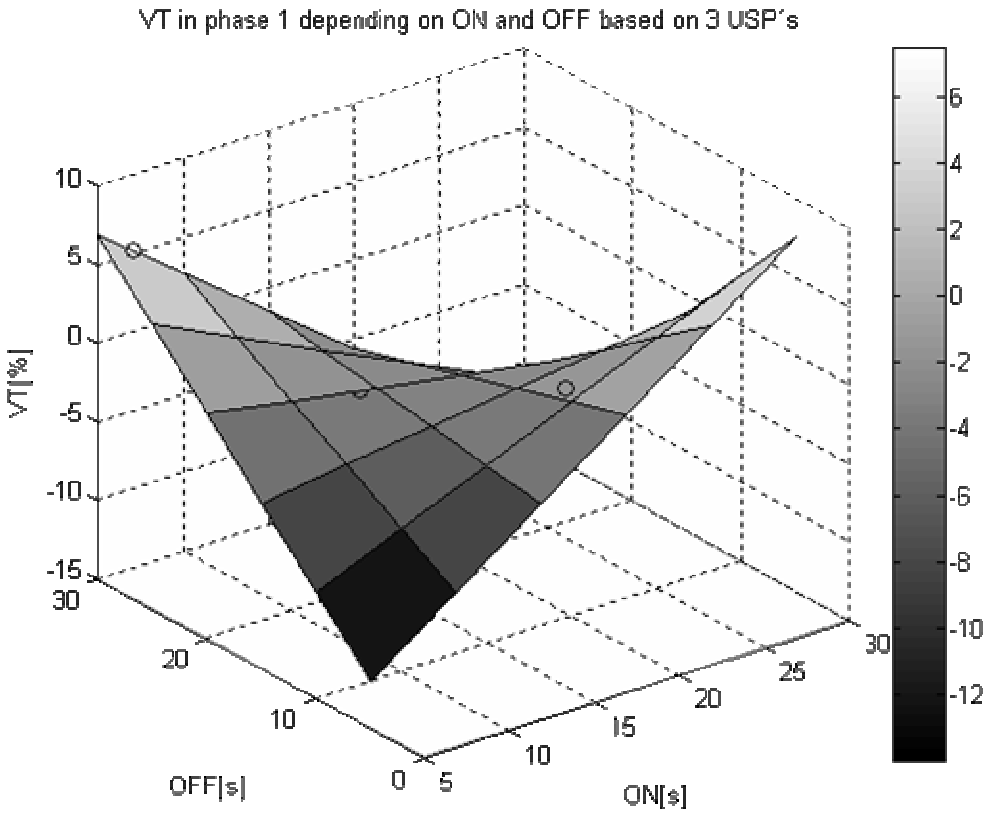

VS in phase 1 depending on F and A based on 9 USP´s
VS[%]
20
10
0
-10
-20
0
10
20
30
40
F[Hz]
0
0.5
1
1.5
A[Amp]
10
5
0
-5
-10

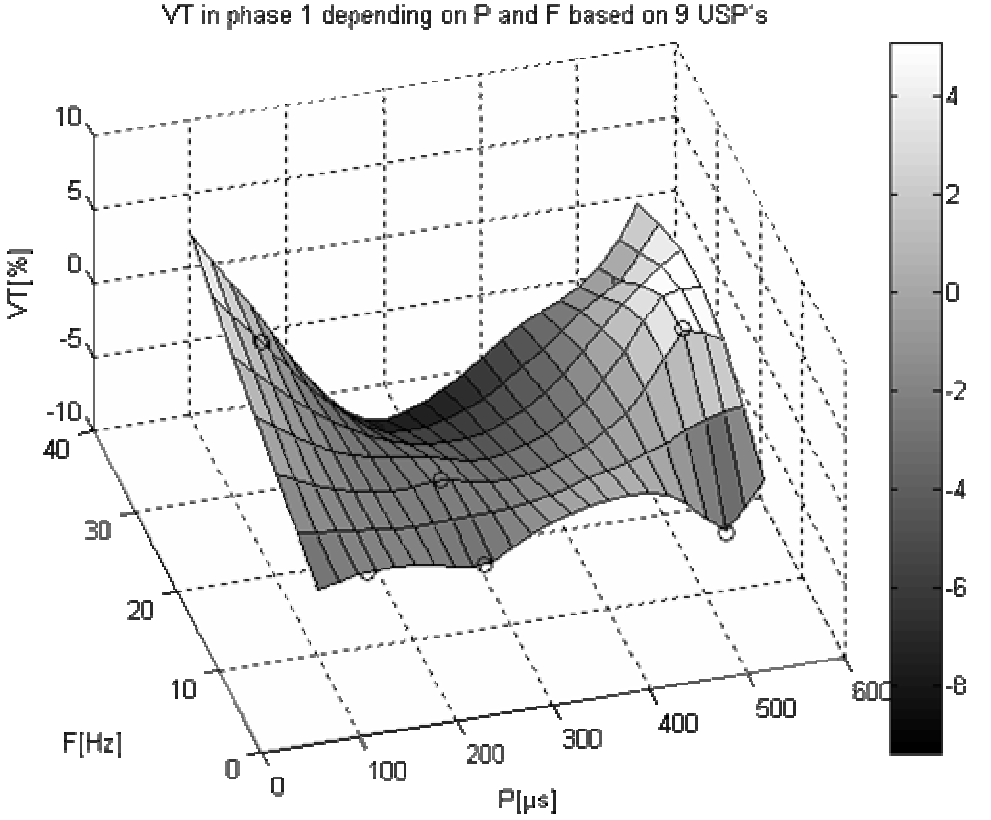

VS in phase 1 depending on ON/OFF ratio and P based on 9 USP´s
VS[%]
20
10
0
-10
-20
600
400
200
0
P[µs]
0.5
1
1.5
2
On/Off ratio [AU]
15
10
5
0
-5
-10

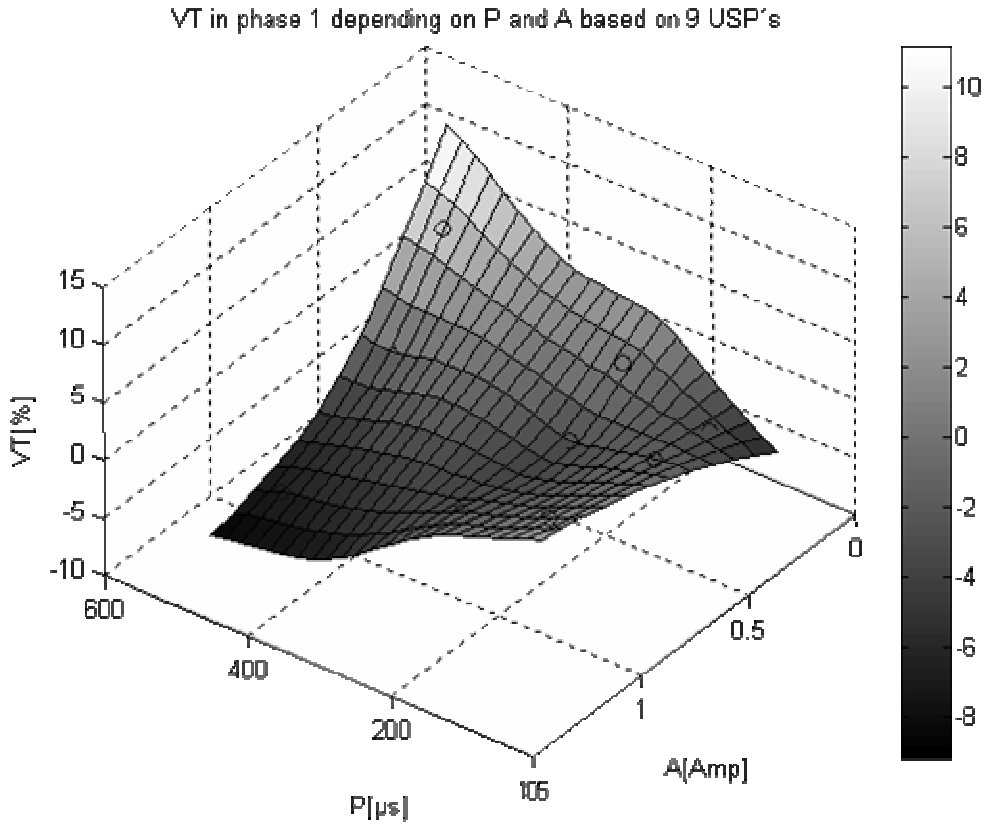

VS in phase 1 depending on ON/OFF ratio and A based on 9 USP's
VS[%]
30
20
10
0
-10
-20
2
1.5
1
0.5
0
On/Off ratio [AU]
0
0.5
1
1.5
A[Amp]
20
15
10
5
0
-5
-10

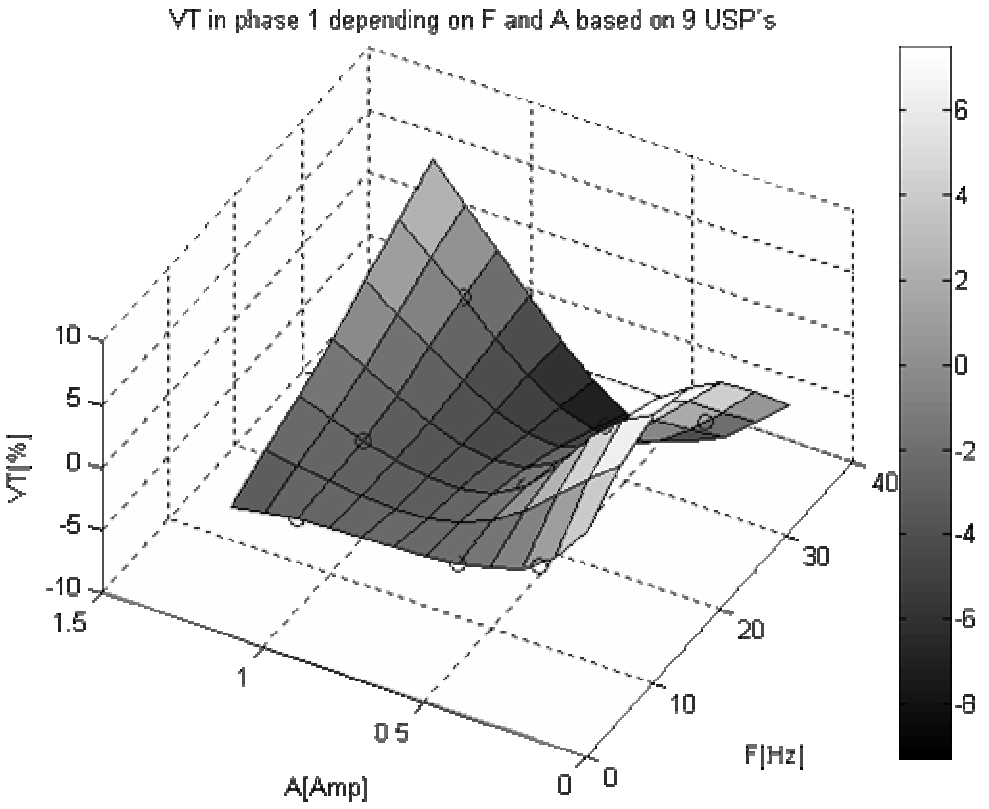

VS in phase 1 depending on ON/OFF ratio and F based on 9 USP's
VS[%]
20
10
0
-10
-20
0
10
20
30
40
F[Hz]
2
1.5
1
0.5
0
On/Off ratio [AU]
10
5
0
-5
-10

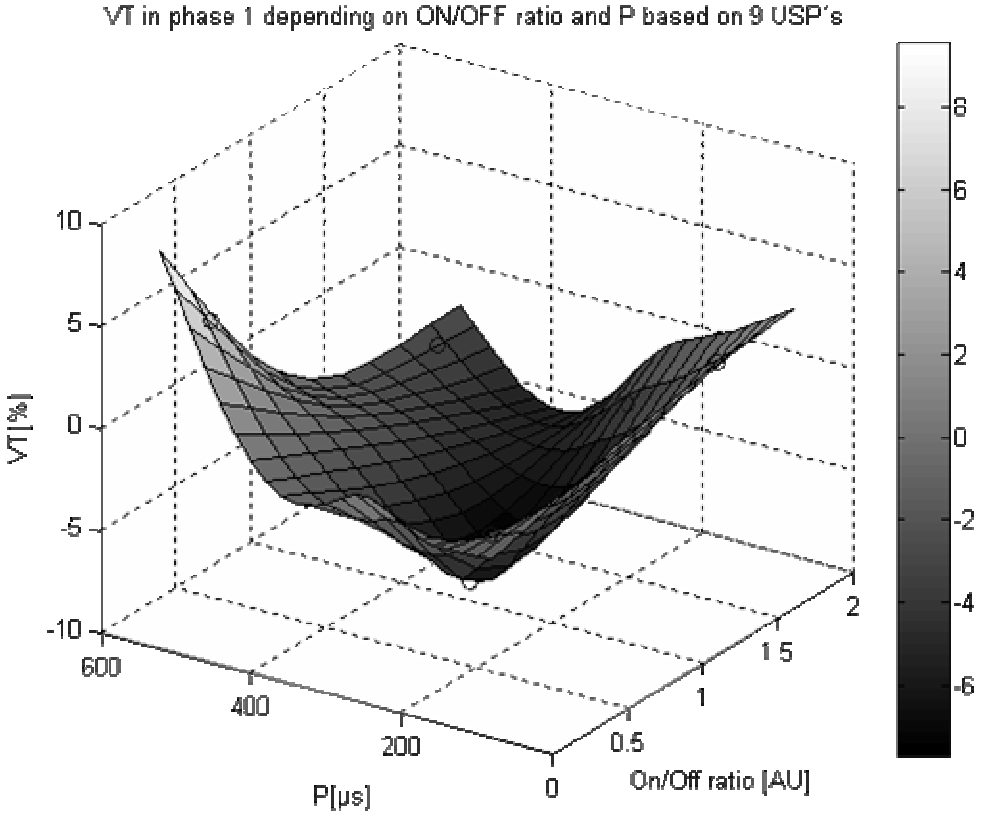
4.2.1.5 Minute Ventilation (Overall Ventilation Performance Indicator)

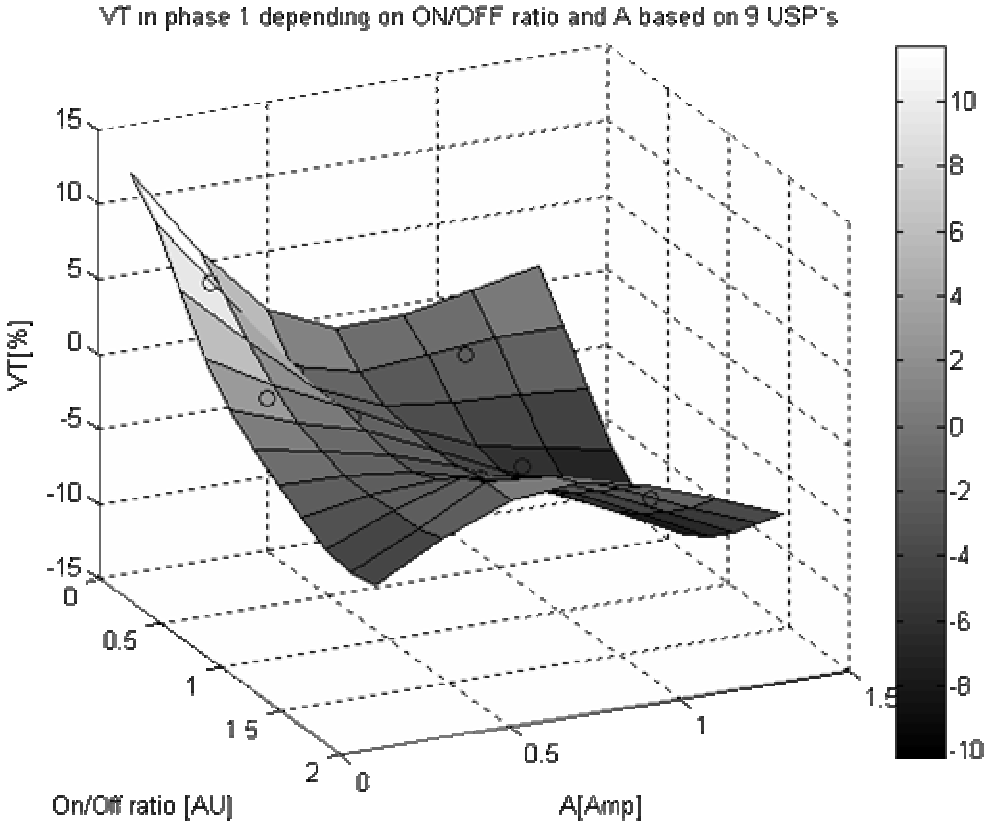


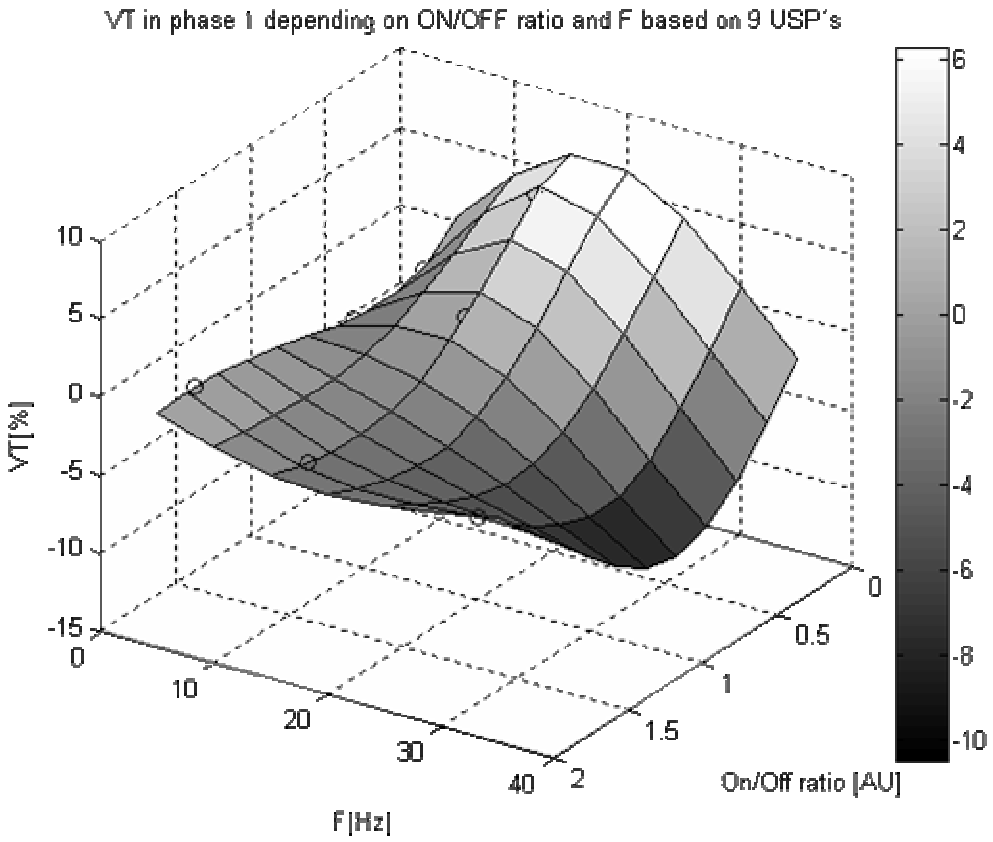



VT in phase 1 depending on P and A based on 9 USP´s
VT[%]
15
10
5
0
-5
-10
600
400
200
105
P[µs]
0
0.5
1
A[Amp]
10
8
6
4
2
0
-2
-4
-6
-8


VT in phase 1 depending on F and A based on 9 USP´s
VT[%]
10
5
0
-5
-10
1.5
1
0.5
0
A[Amp]
0
10
20
30
40
F[Hz]
6
4
2
0
-2
-4
-6
-8

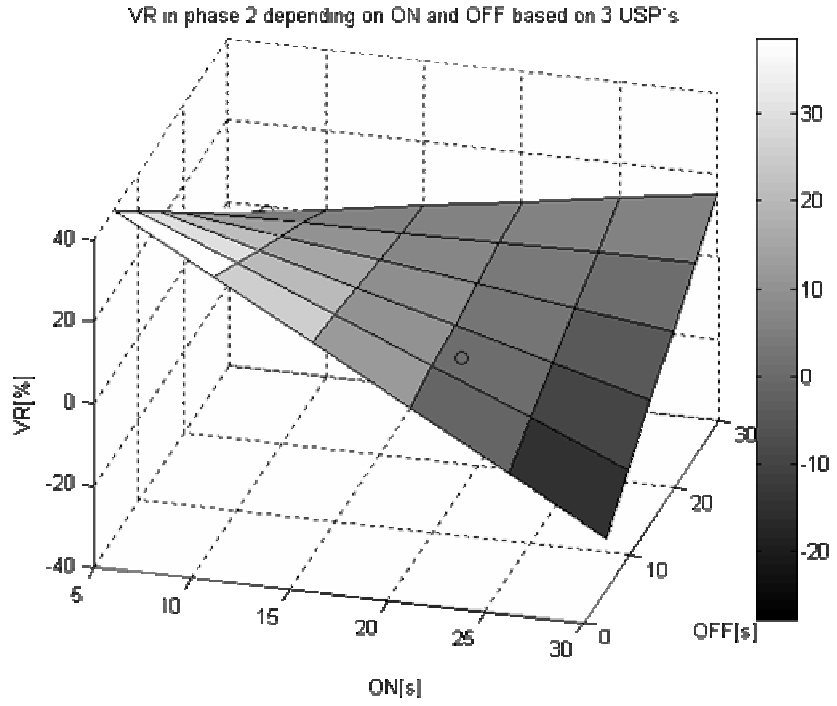

VT in phase 1 depending on ON/OFF ratio and P based on 9 USP´s
VT[%]
10
5
0
-5
-10
600
400
200
0
P[µs]
0.5
1
1 5
2
On/Off ratio [AU]
8
6
4
2
0
-2
-4
-6

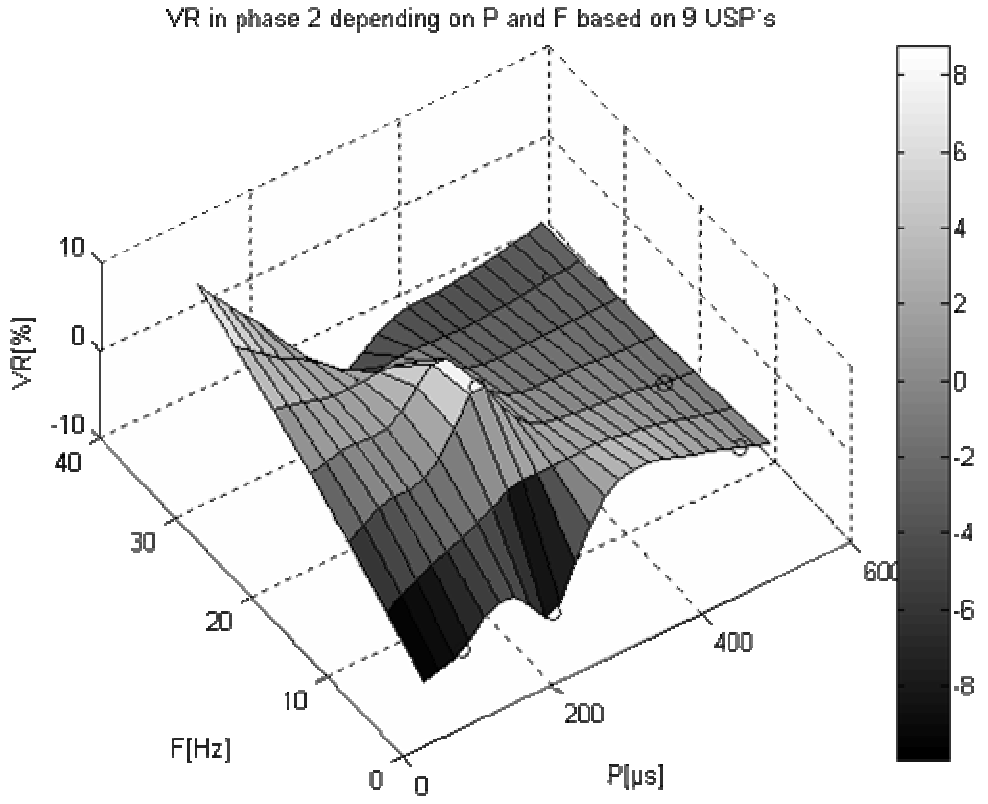

VT in phase 1 depending on ON/OFF ratio and A based on 9 USP´s
VT[%]
15
10
5
0
-5
-10
-15
0
0.5
1
1 5
2
On/Off ratio [AU]
0
0.5
1
1.5
A[Amp]
10
8
6
4
2
0
-2
-4
-6
-8
-10

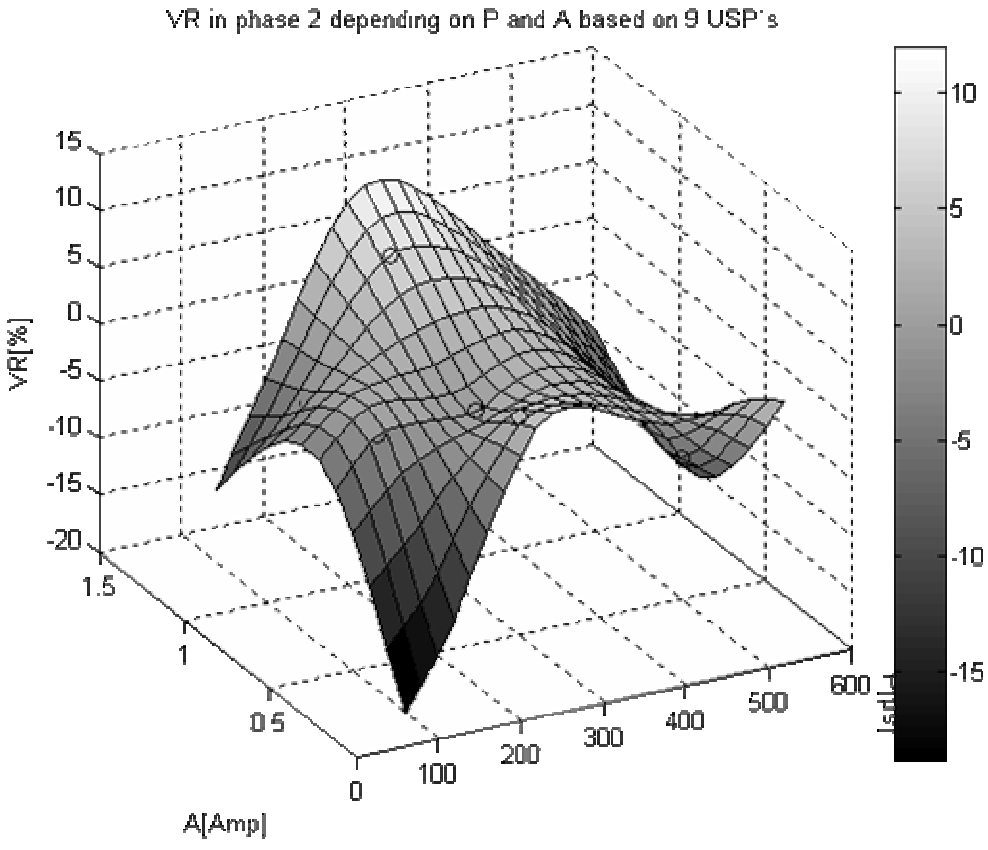


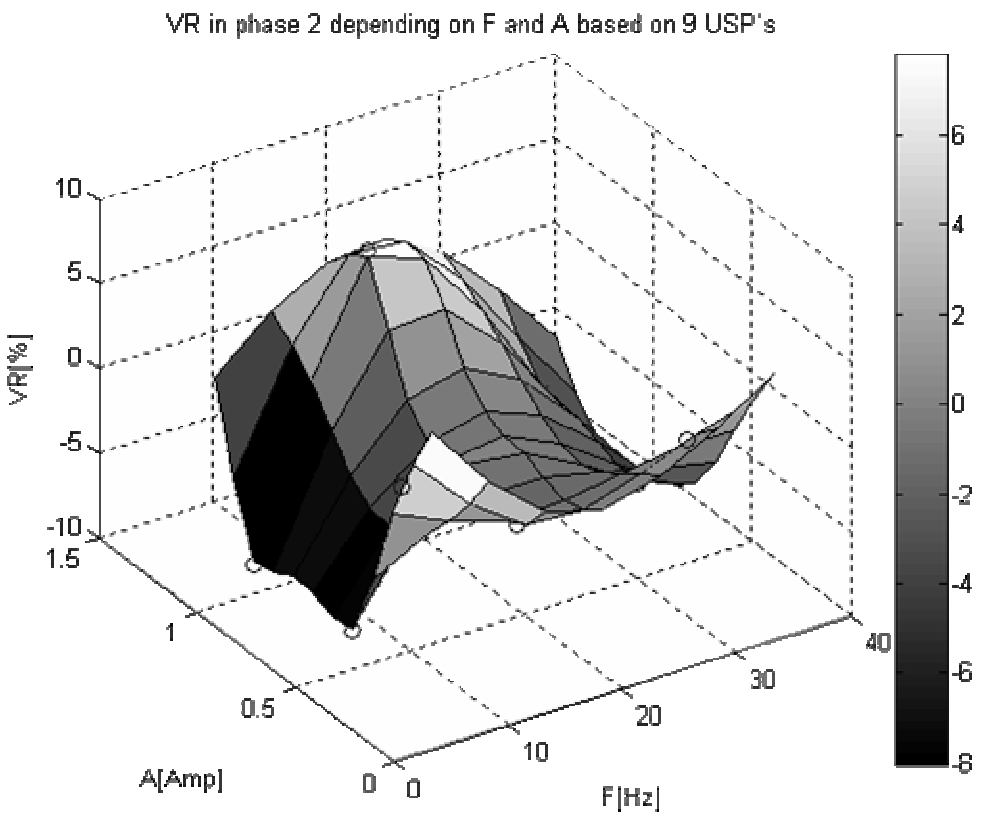
4.2.2 Phase 2 (Rebounce Effects)

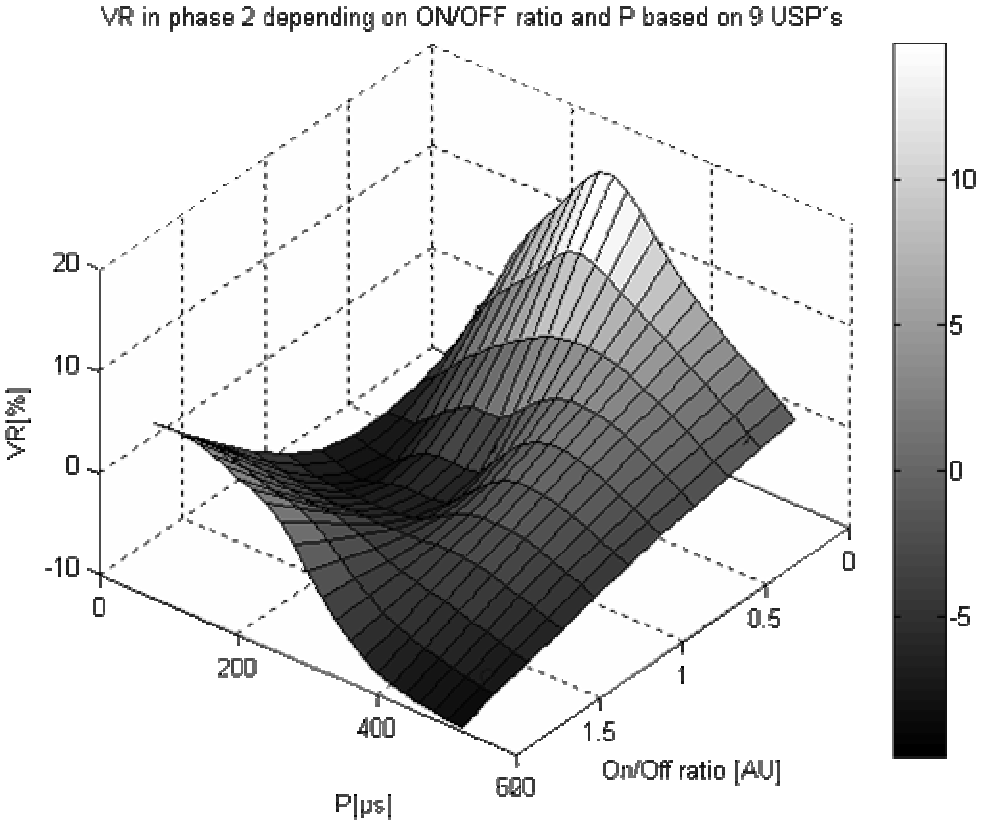
4.2.2.1 Ventilation Rate

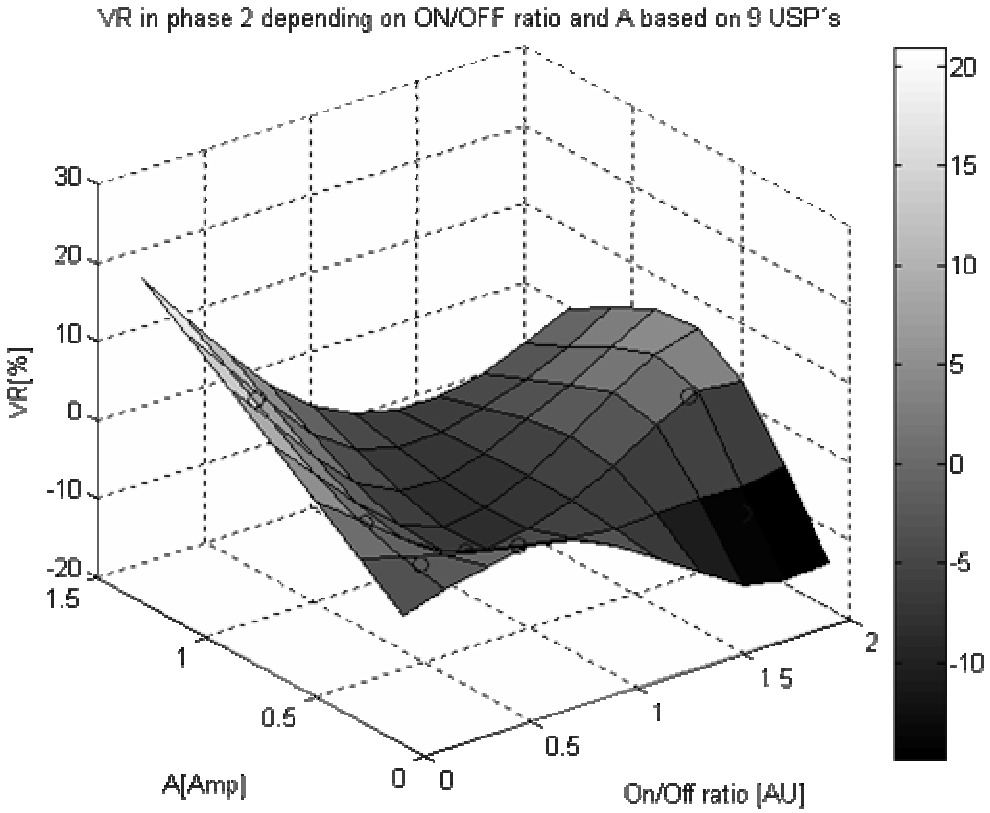


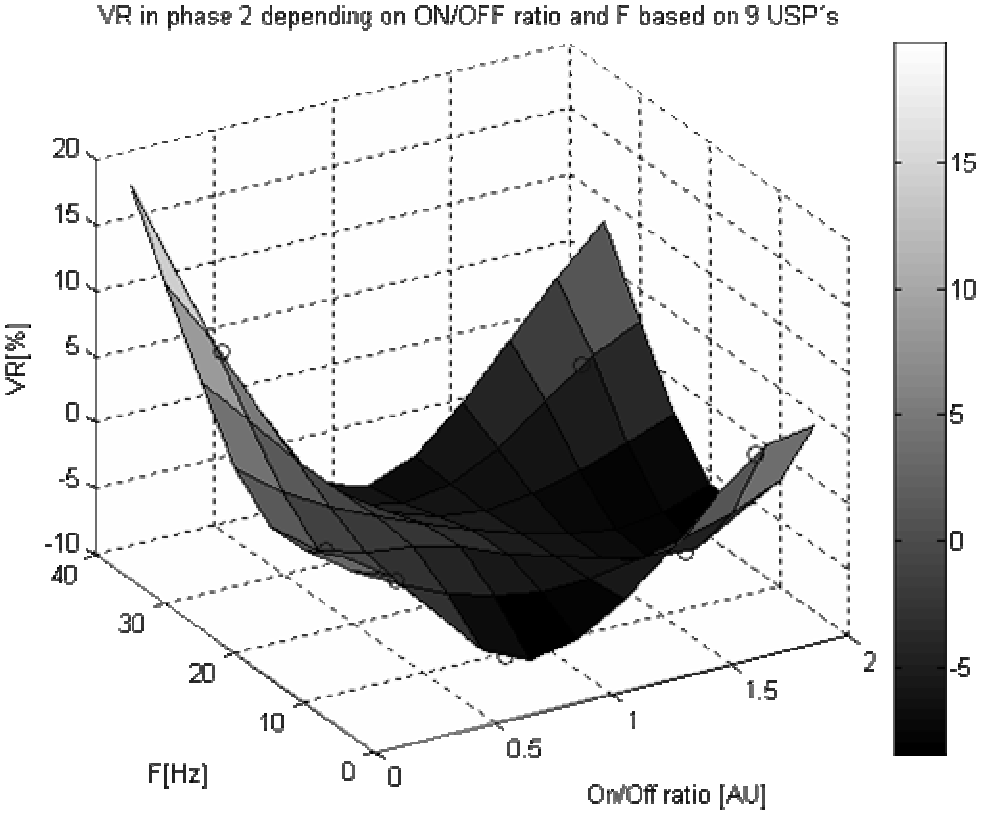

VR in phase 2 depending on P and F based on 9 USP's
VR[%]
10
0
-10
40
30
20
10
0
F[Hz]
0
200
400
600
P[µs]
8
6
4
2
0
-2
-4
-6
-8


VR in phase 2 depending on P and A based on 9 USP's
VR[%]
15
10
5
0
-5
-10
-15
-20
1.5
1
0.5
0
A[Amp]
0
100
200
300
400
500
600
10
5
0
-5
-10
-15

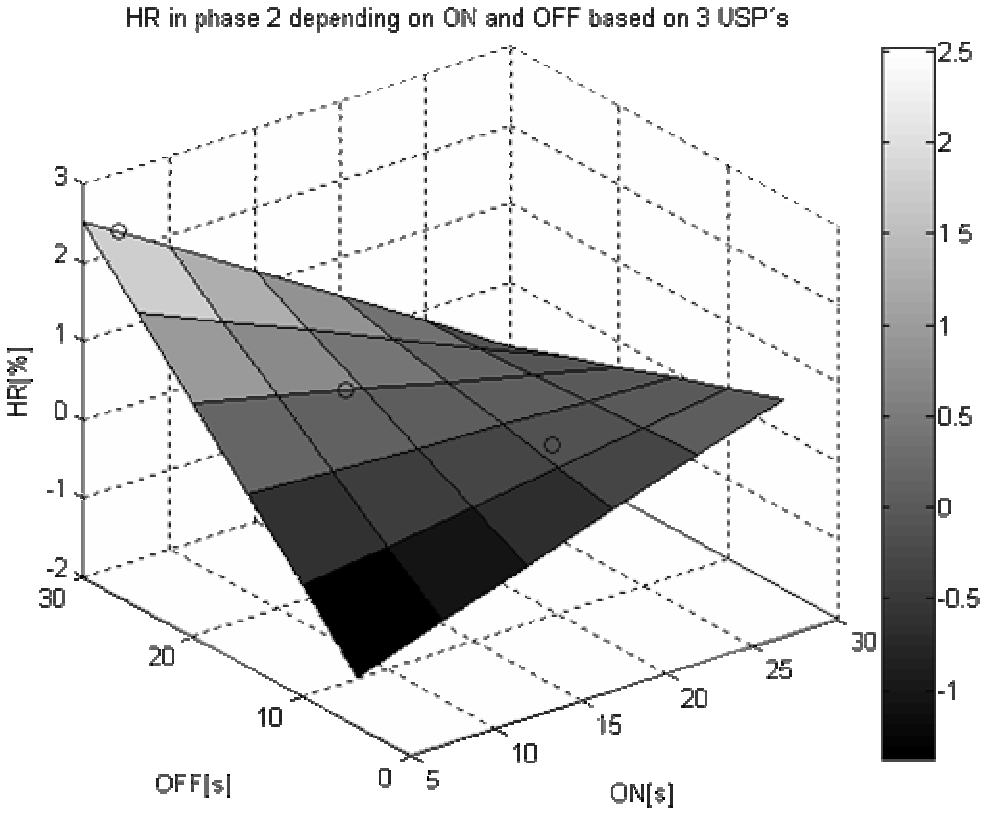

VR in phase 2 depending on F and A based on 9 USP's
VR[%]
10
5
0
-5
-10
1.5
1
0.5
0
A[Amp]
0
10
20
30
40
F[Hz]
6
4
2
0
-2
-4
-6
-8

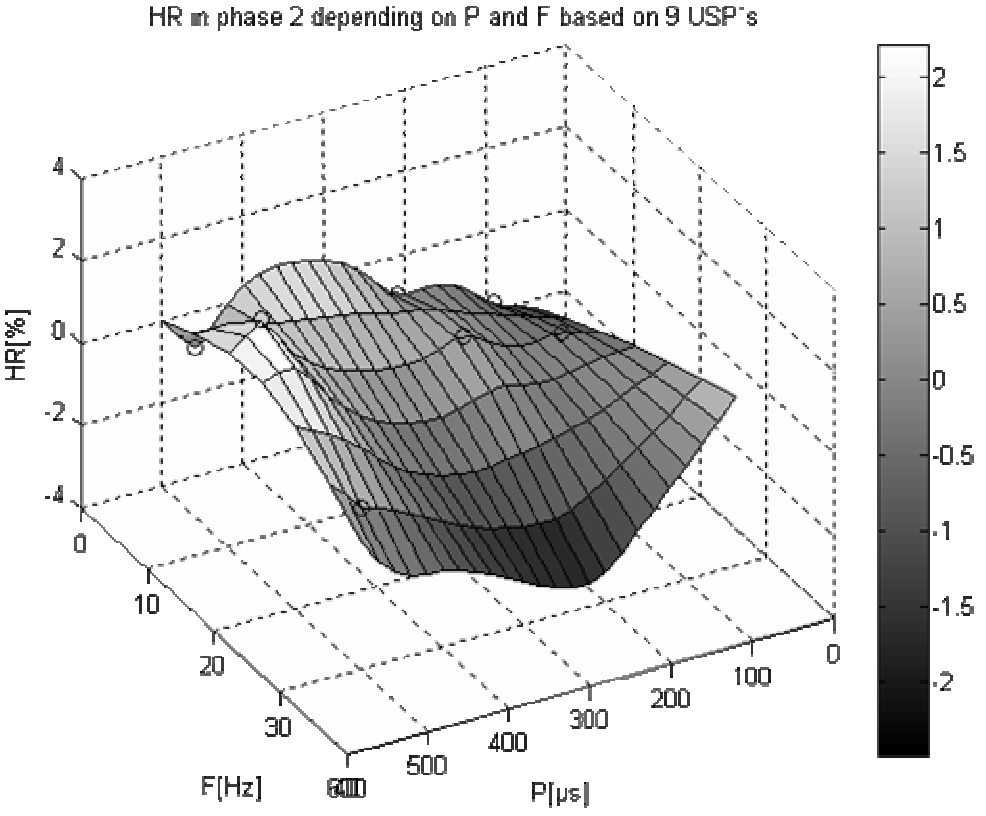

VR in phase 2 depending on ON/OFF ratio and P based on 9 USP's
VR[%]
20
10
0
-10
0
200
400
600
P[µs]
0
0.5
1
1.5
On/Off ratio [AU]
10
5
0
-5

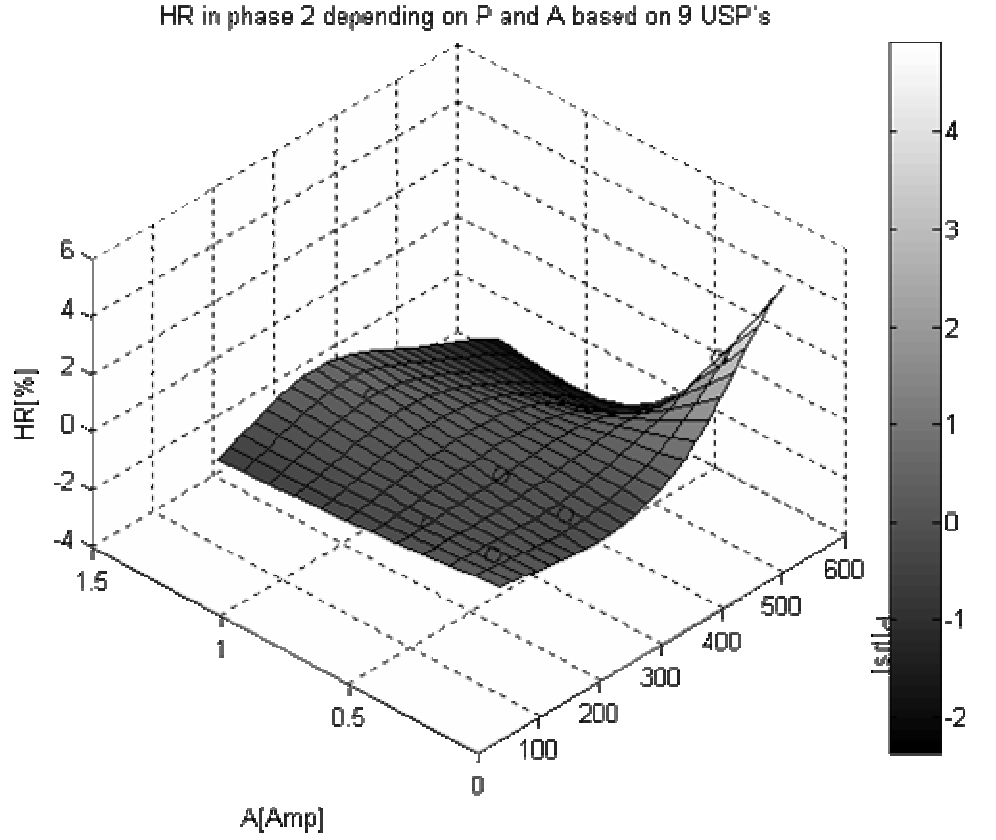

VR in phase 2 depending on ON/OFF ratio and A based on 9 USP´s
VR[%]
30
20
10
0
-10
-20
1.5
1
0.5
0
A[Amp]
0
0.5
1
1.5
2
On/Off ratio [AU]
20
15
10
5
0
-5
-10

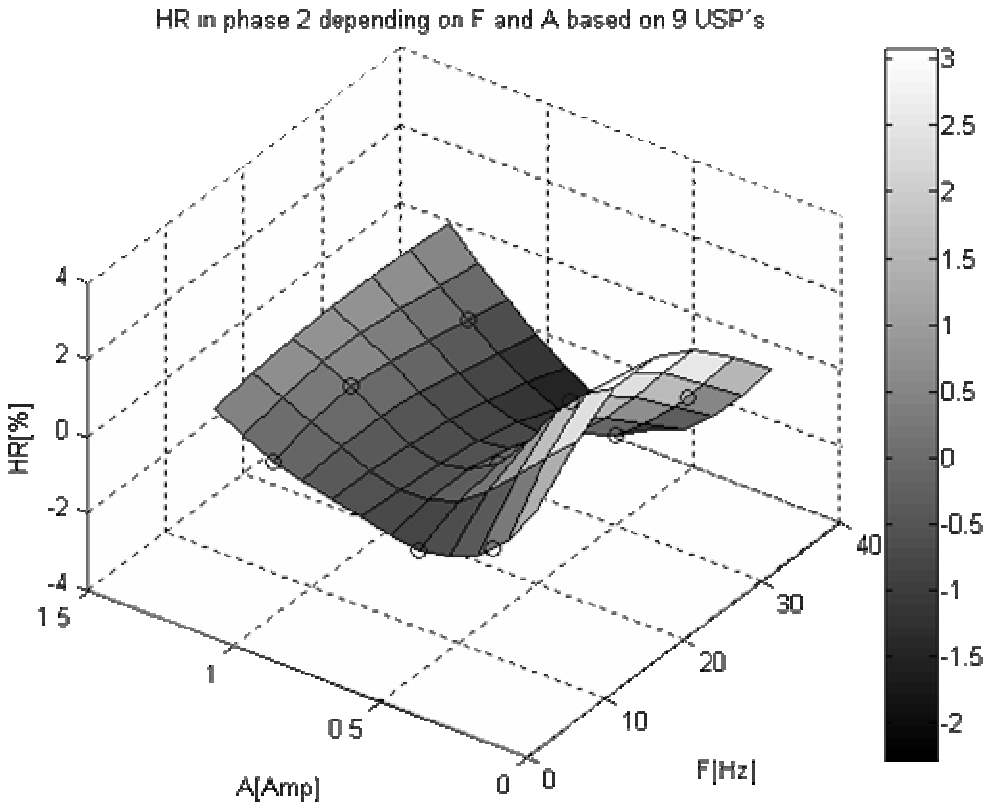

VR in phase 2 depending on ON/OFF ratio and F based on 9 USP´s
VR[%]
20
15
10
5
0
-5
-10
40
30
20
10
0
F[Hz]
0
0.5
1
1.5
2
On/Off ratio [AU]
15
10
5
0
-5

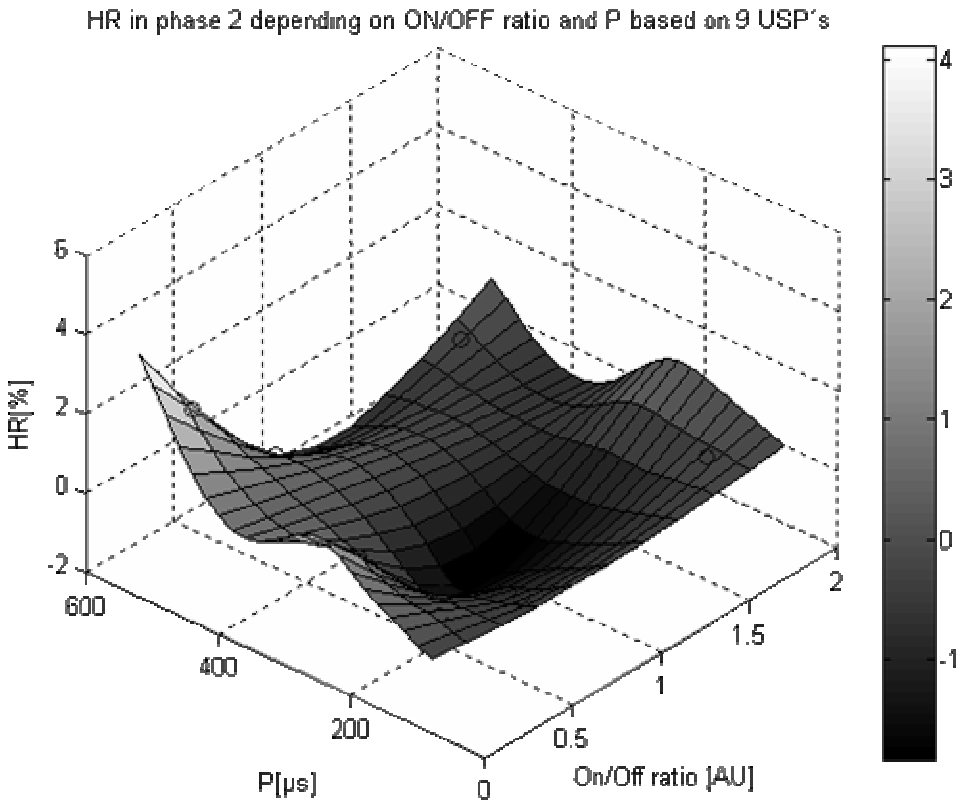
4.2.2.2 Heart Rate

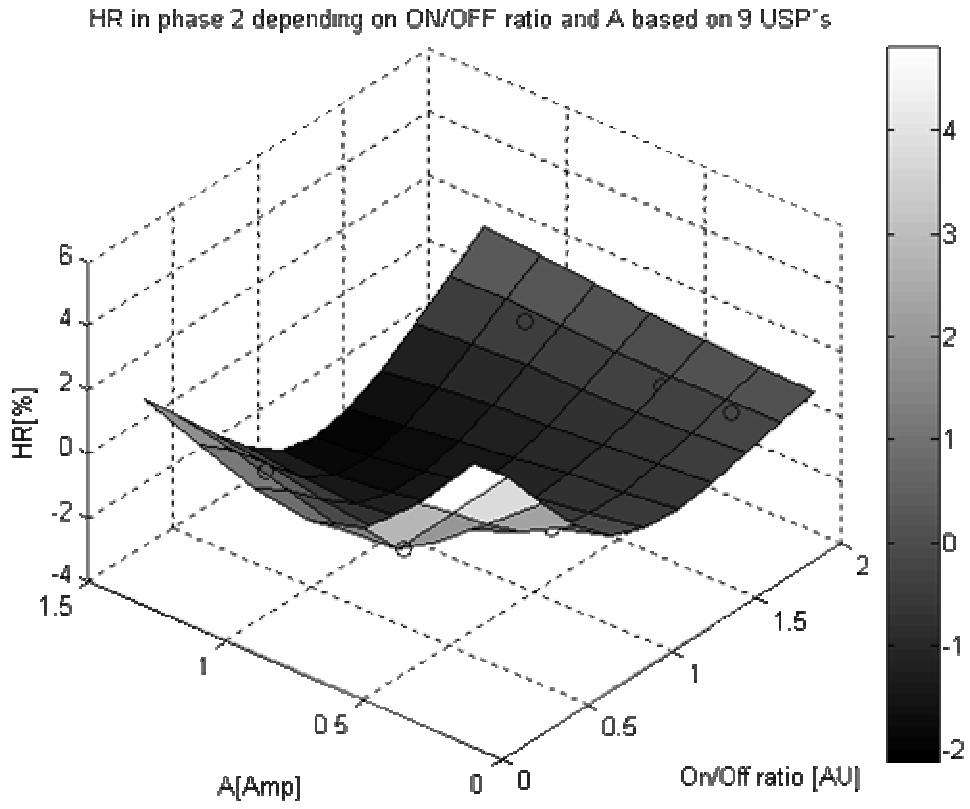


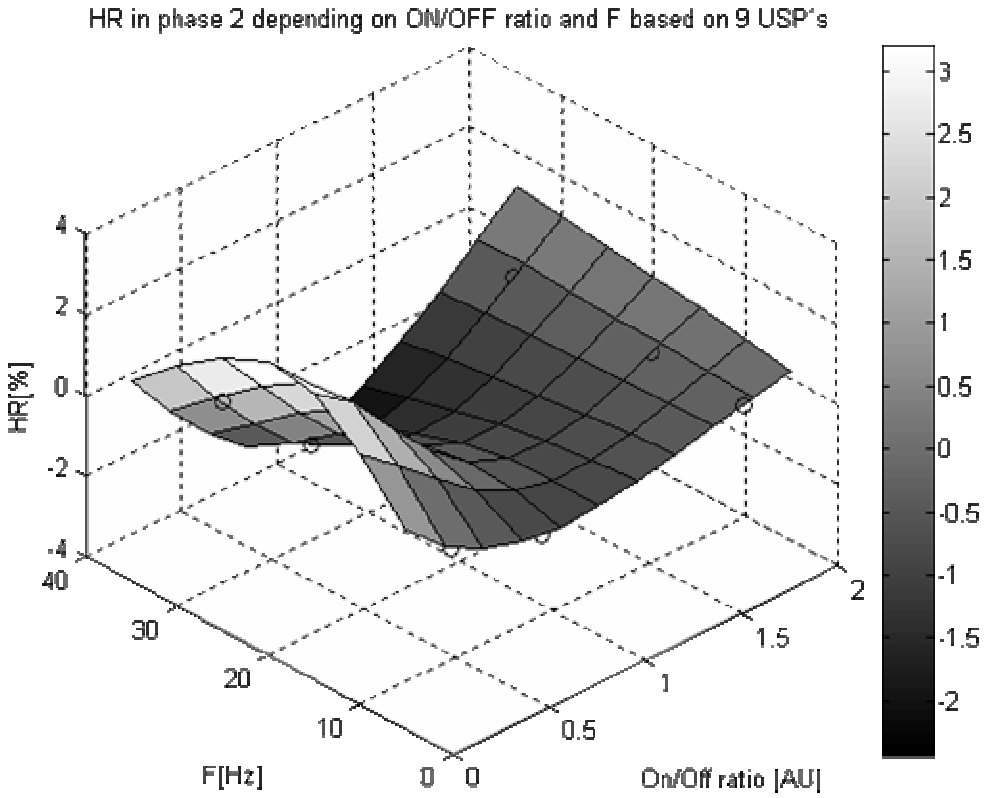



HR in phase 2 depending on P and A based on 9 USP's
HR[%]
6
4
2
0
-2
-4
1.5
1
0.5
0
A[Amp]
100
200
300
400
500
600
4
3
2
1
0
-1
-2

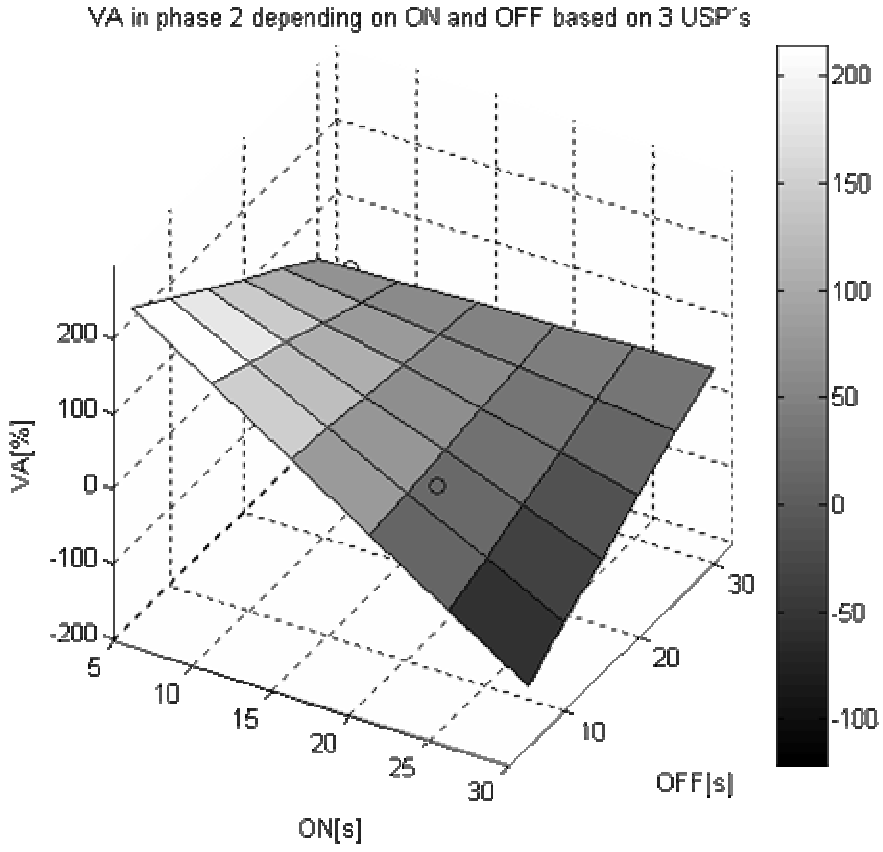

HR in phase 2 depending on F and A based on 9 USP's
HR[%]
4
2
0
-2
-4
1 5
1
0 5
0
A[Amp]
0
10
20
30
40
F|Hz|
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2

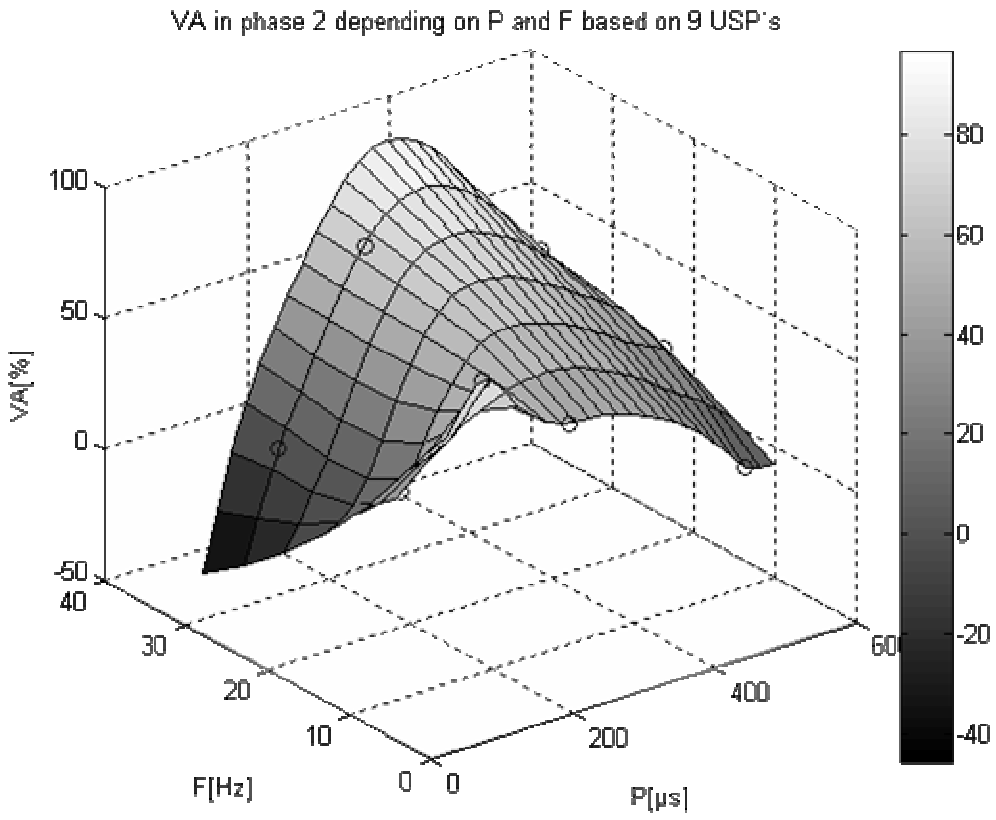

HR in phase 2 depending on ON/OFF ratio and P based on 9 USP´s
HR[%]
6
4
2
0
-2
600
400
200
0
P[µs]
0
0.5
1
1.5
2
On/Off ratio [AU]
4
3
2
1
0
-1

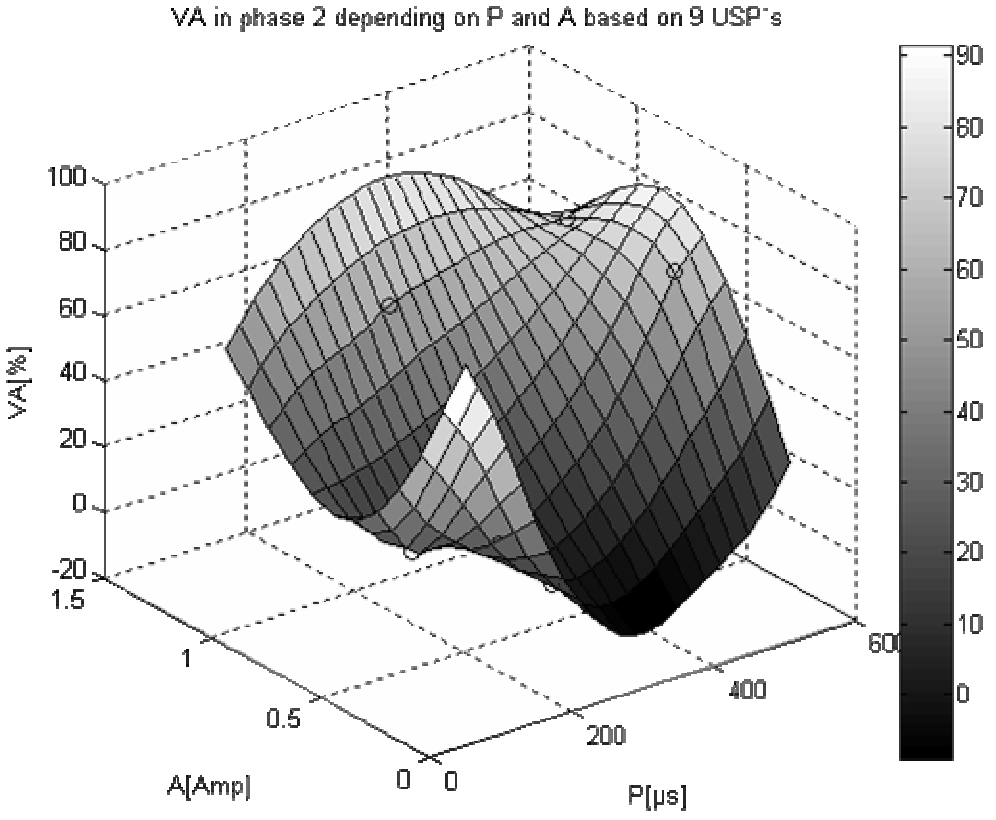

HR in phase 2 depending on ON/OFF ratio and A based on 9 USP´s
HR[%]
6
4
2
0
-2
-4
1.5
1
0.5
0
A[Amp]
0
0.5
1
1.5
2
On/Off ratio [AU]
4
3
2
1
0
-1
-2

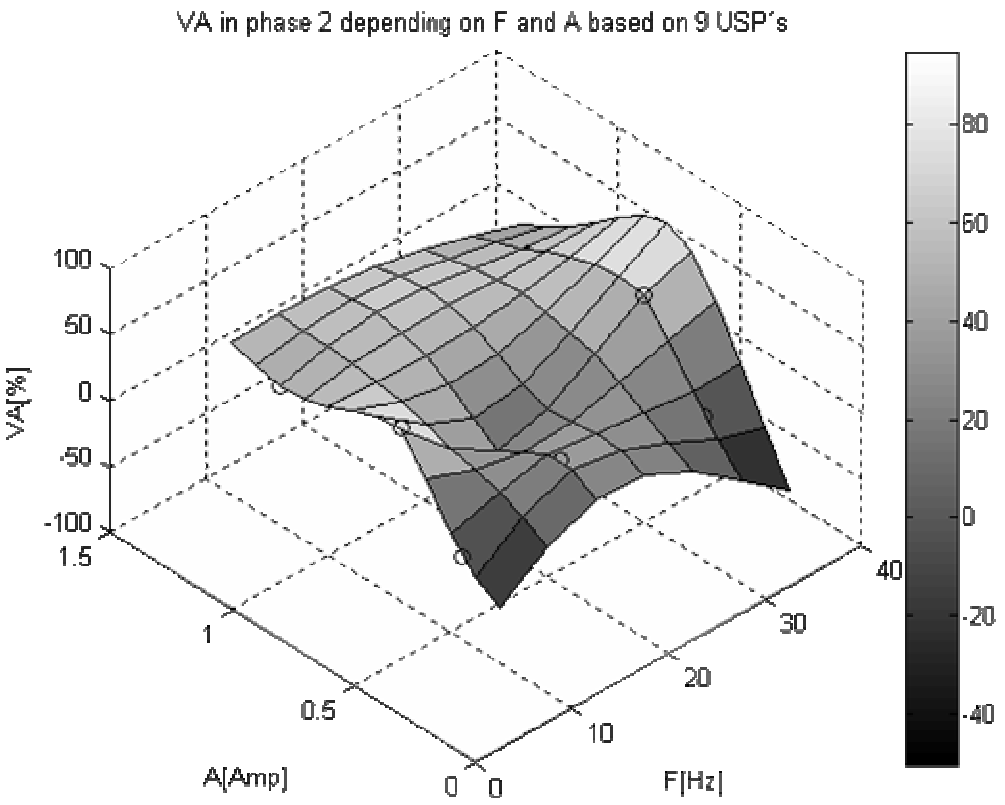


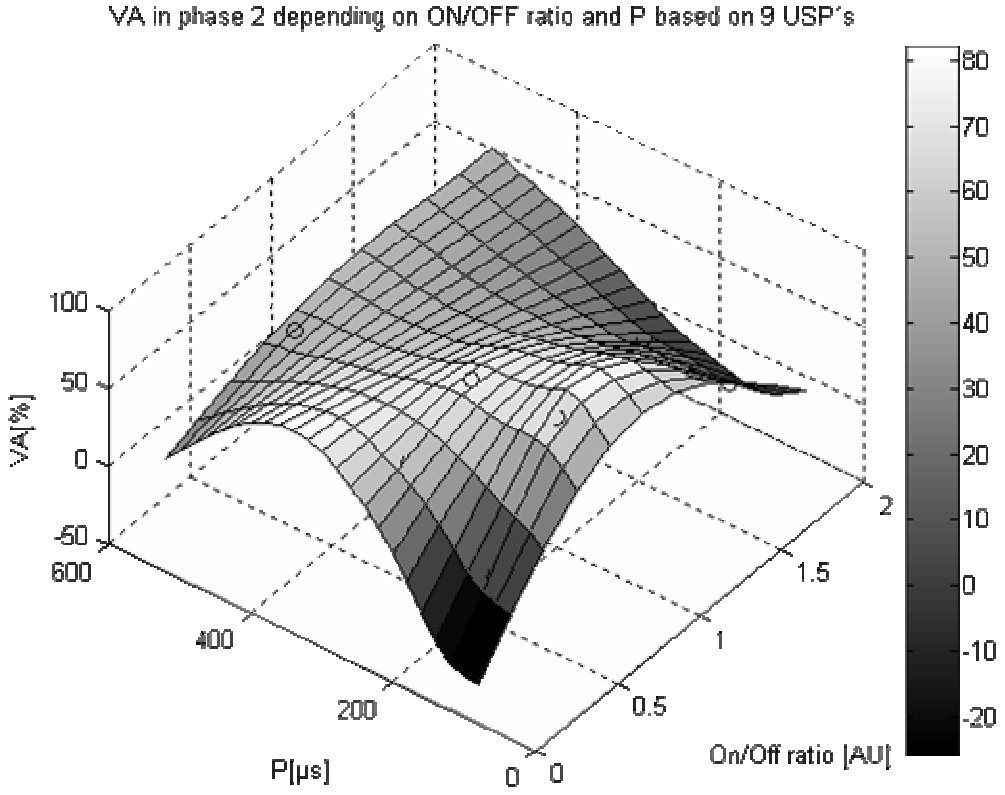
4.2.2.3 Ventilation Amplitude

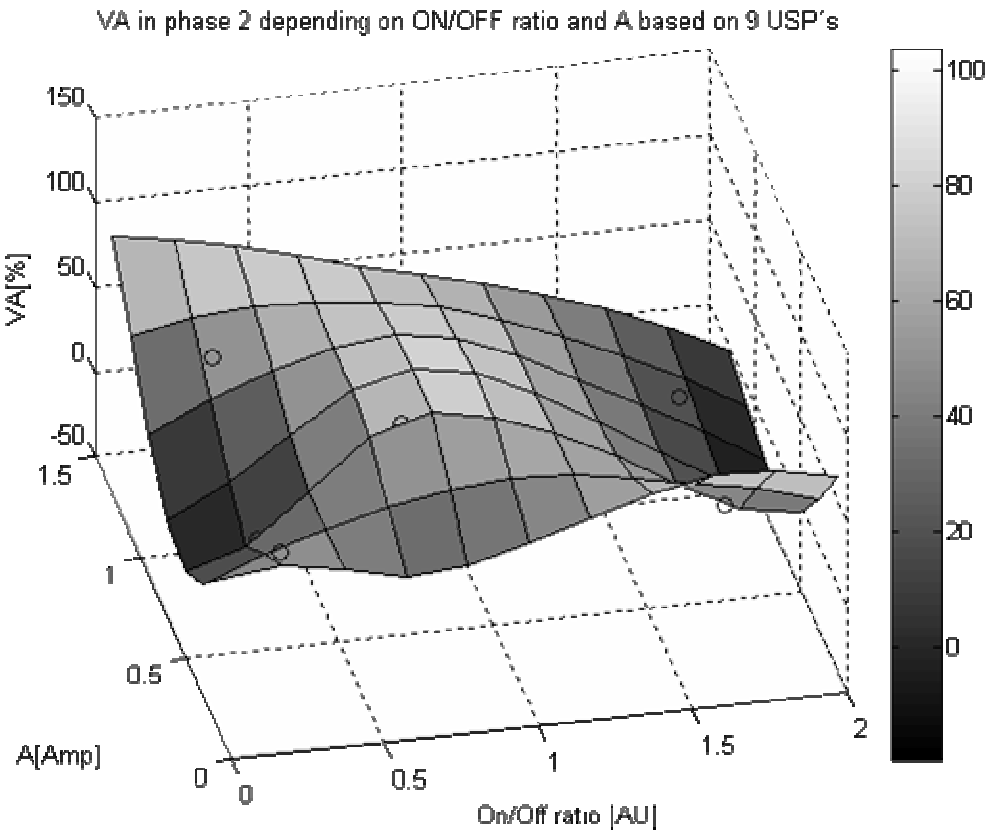


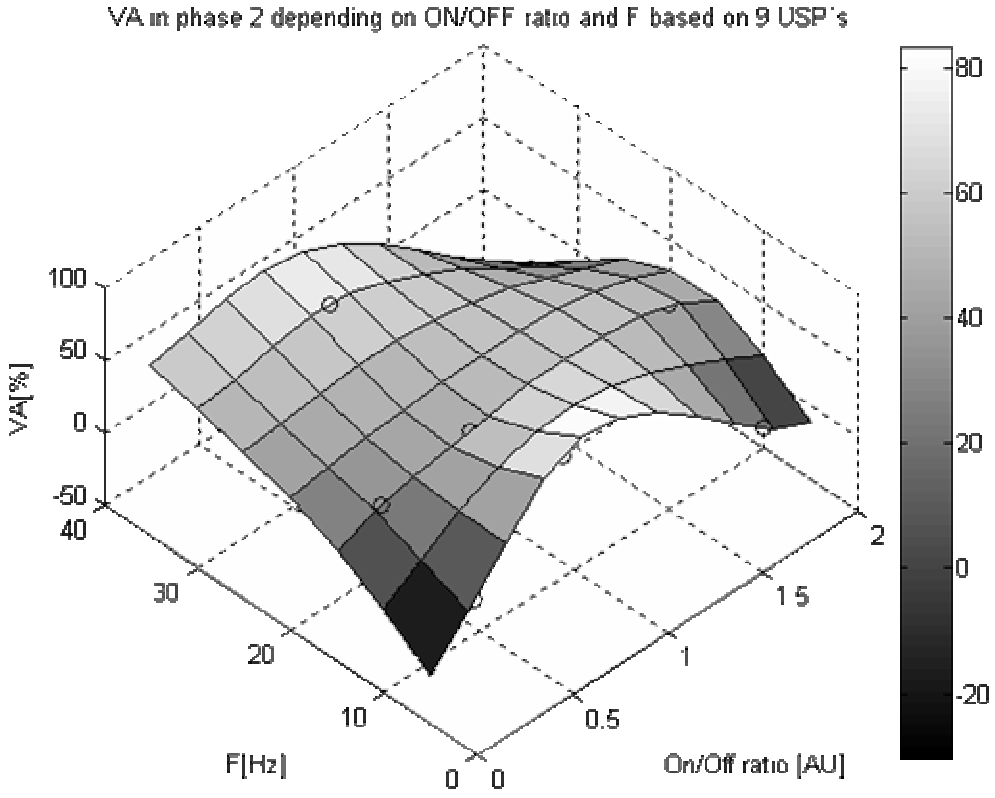

VA in phase 2 depending on P and F based on 9 USP's
VA[%]
100
50
0
-50
40
30
20
10
0
F[Hz]
0
200
400
600
P[µs]
80
60
40
20
0
-20
-40


VA in phase 2 depending on P and A based on 9 USP's
VA[%]
100
80
60
40
20
0
-20
1.5
1
0.5
0
A[Amp]
0
200
400
600
P[µs]
90
80
70
60
50
40
30
20
10
0

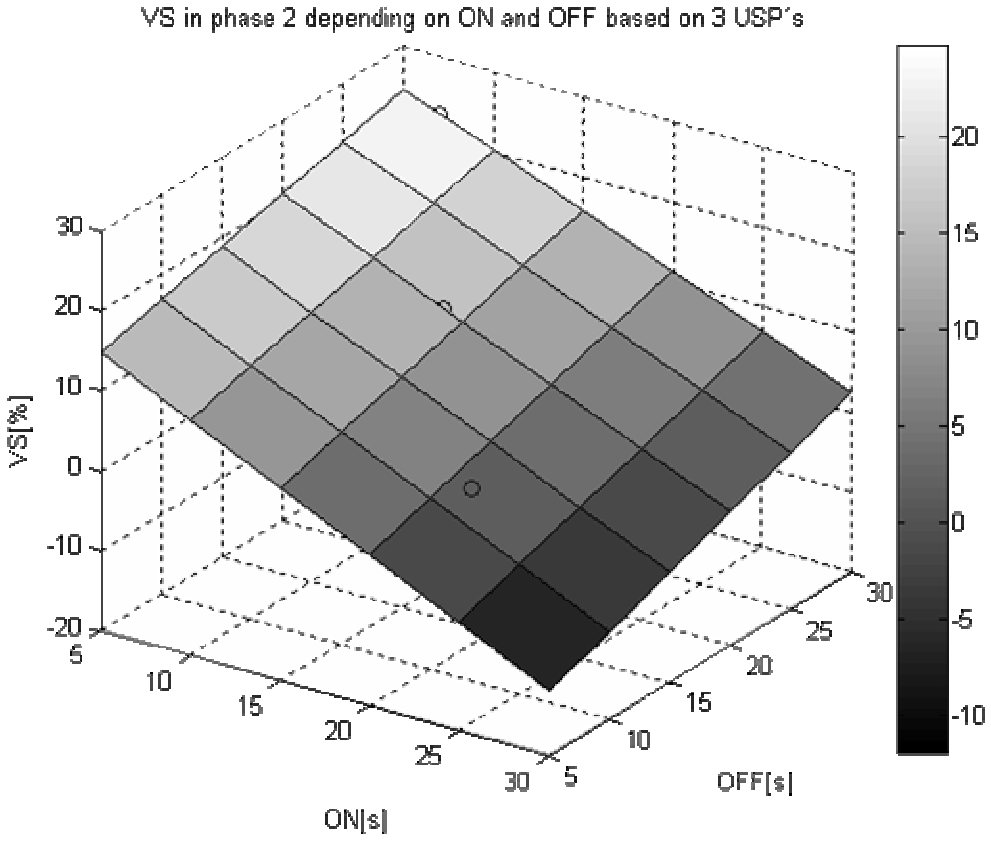

VA in phase 2 depending on F and A based on 9 USP´s
VA[%]
100
50
0
-50
-100
1.5
1
0.5
0
A[Amp]
0
10
20
30
40
F[Hz]
80
60
40
20
0
-20
-40

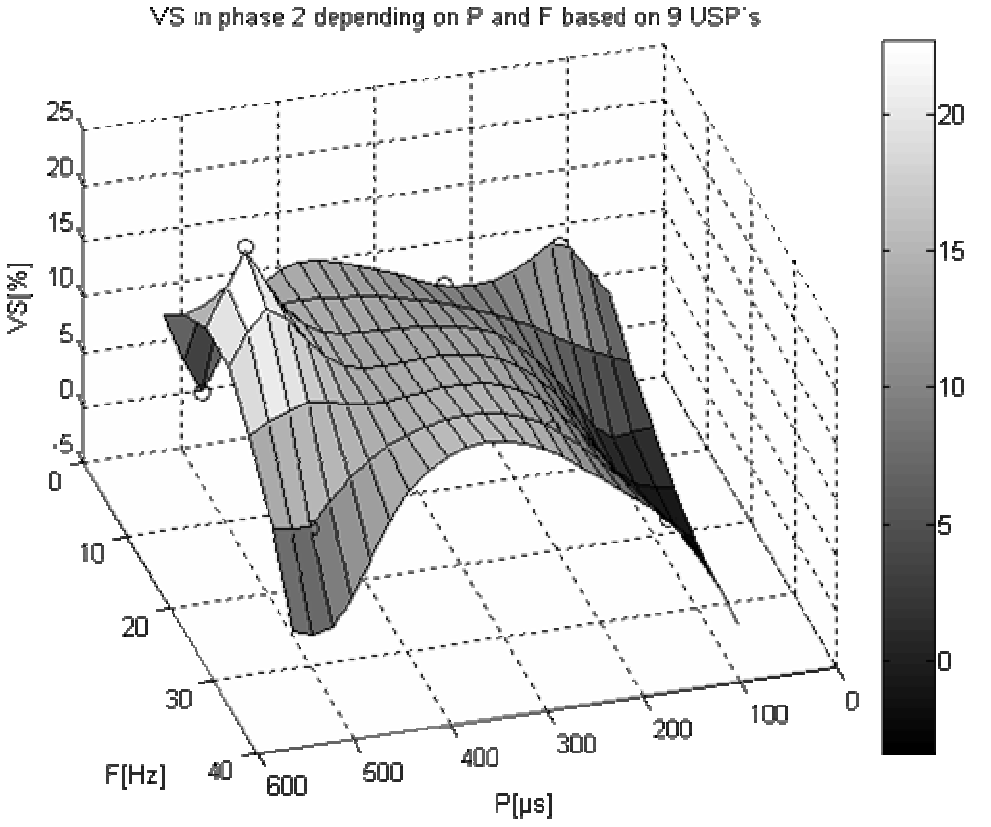

VA in phase 2 depending on ON/OFF ratio and P based on 9 USP´s
VA[%]
100
50
0
-50
600
400
200
0
P[µs]
0
0.5
1
1.5
2
On/Off ratio [AU]
80
70
60
50
40
30
20
10
0
-10
-20

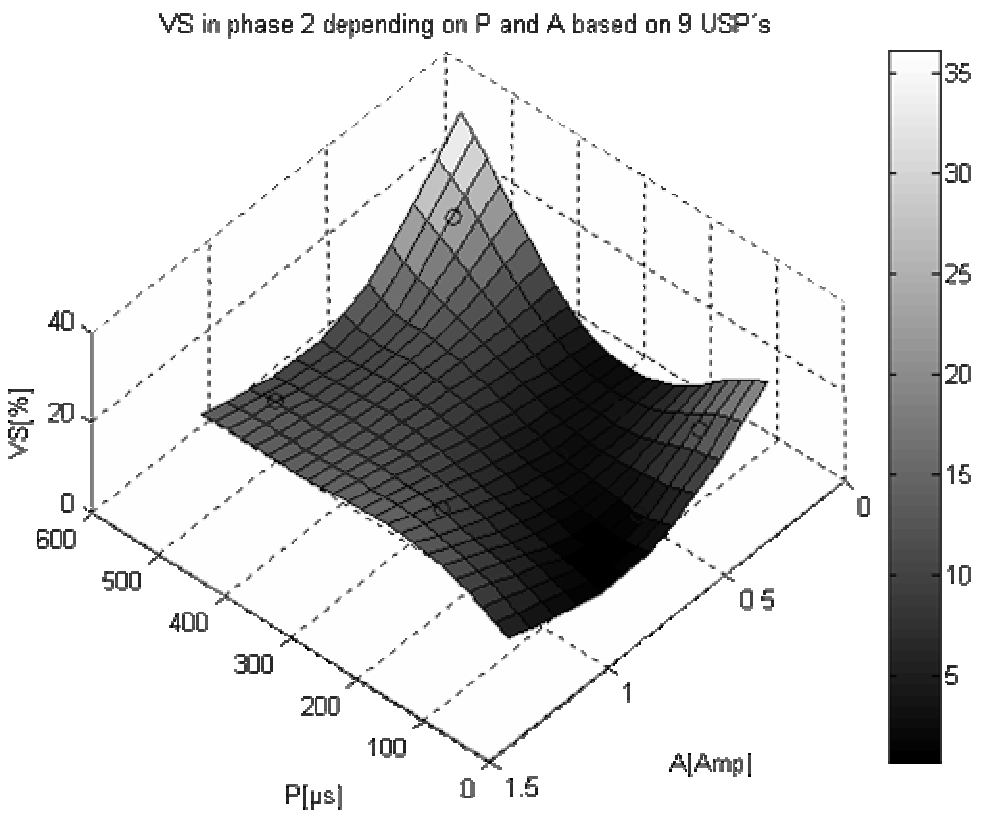

VA in phase 2 depending on ON/OFF ratio and A based on 9 USP´s
VA[%]
150
100
50
0
-50
A[Amp]
1.5
1
0.5
0
0
0.5
1
1.5
2
On/Off ratio [AU]
100
80
60
40
20
0

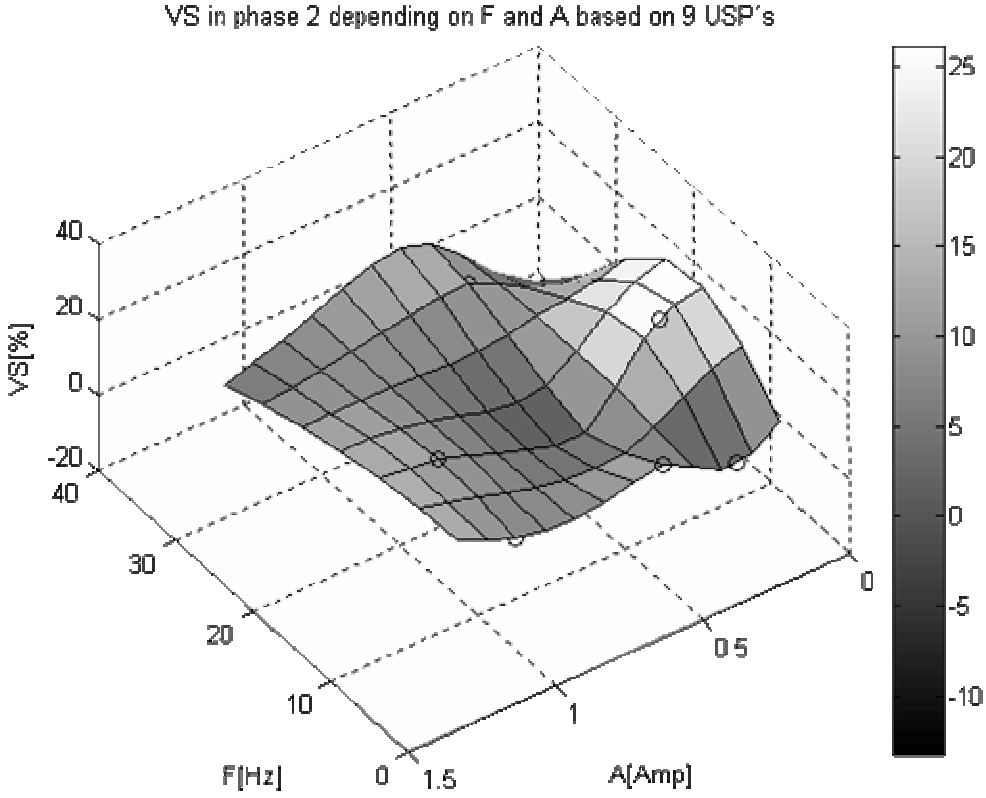

VA in phase 2 depending on ON/OFF ratio and F based on 9 USP´s
VA[%]
100
50
0
-50
F[Hz]
40
30
20
10
0
0
0.5
1
1.5
2
On/Off ratio [AU]
80
60
40
20
0
-20

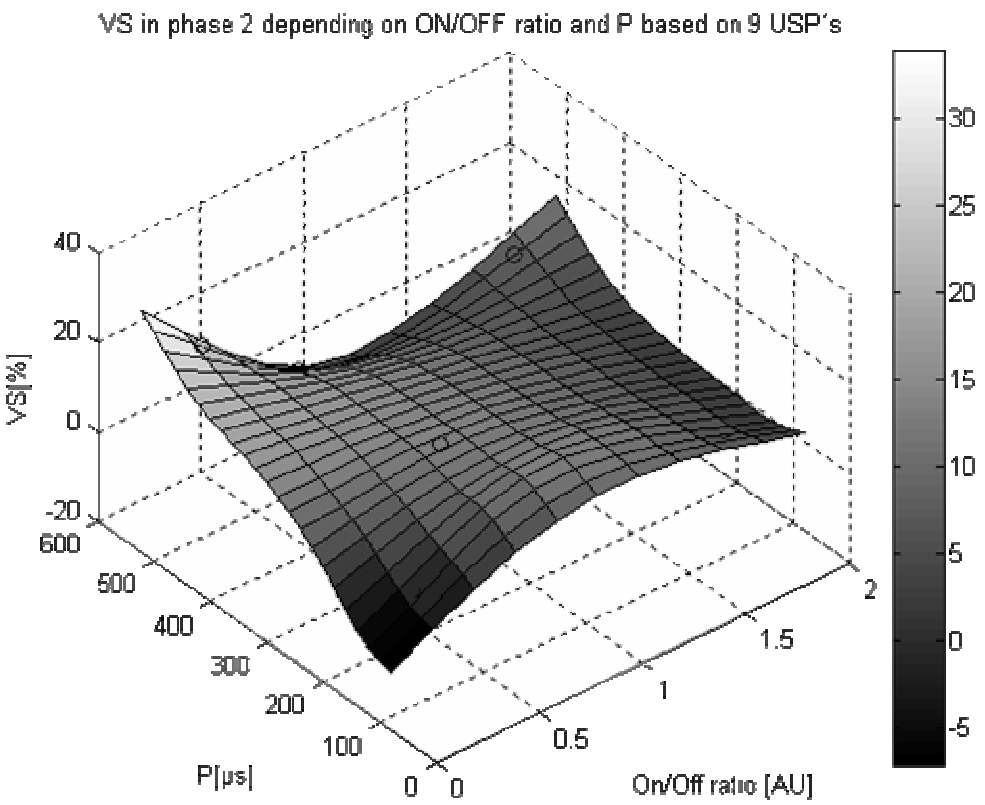
4.2.2.4 Ventilation Slope

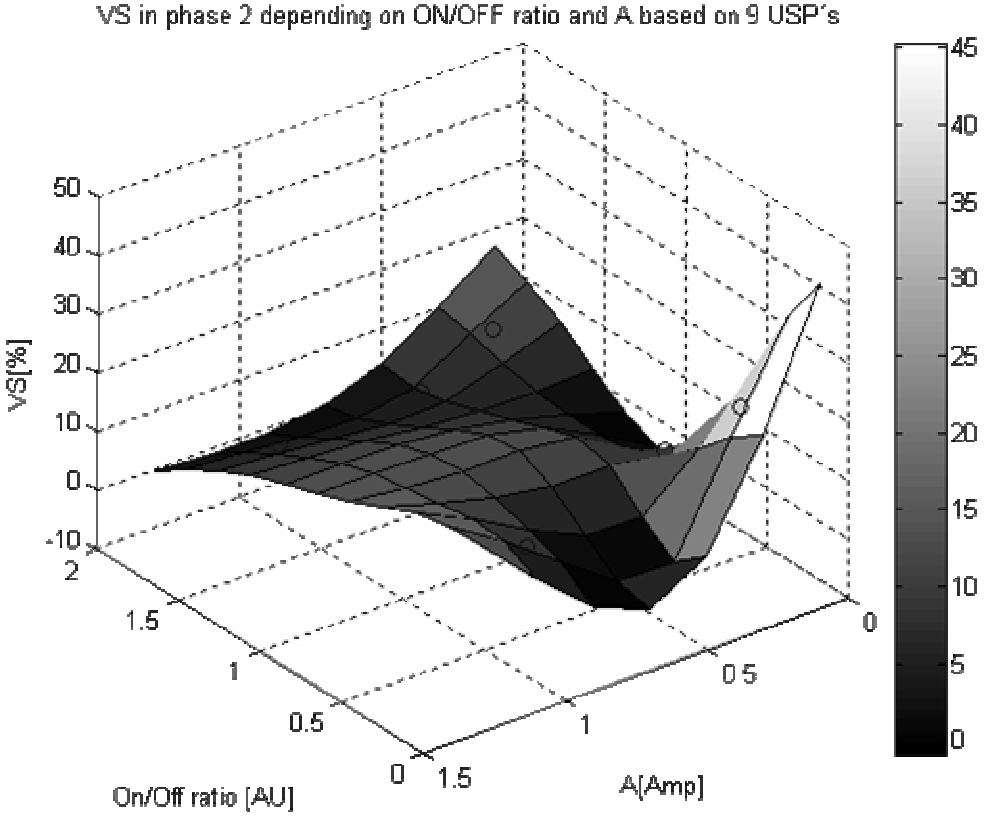


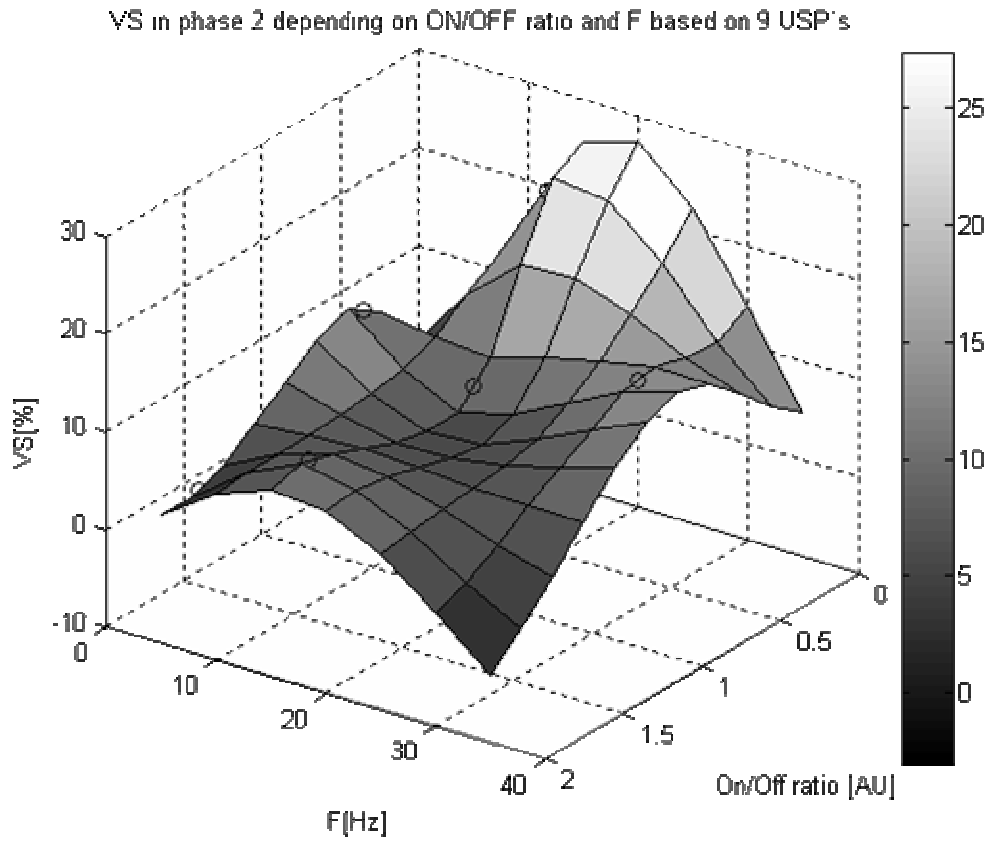



VS in phase 2 depending on P and A based on 9 USP´s
VS[%]
40
20
0
600
500
400
300
200
100
0
P[µs]
0
0.5
1
1.5
A[Amp]
35
30
25
20
15
10
5

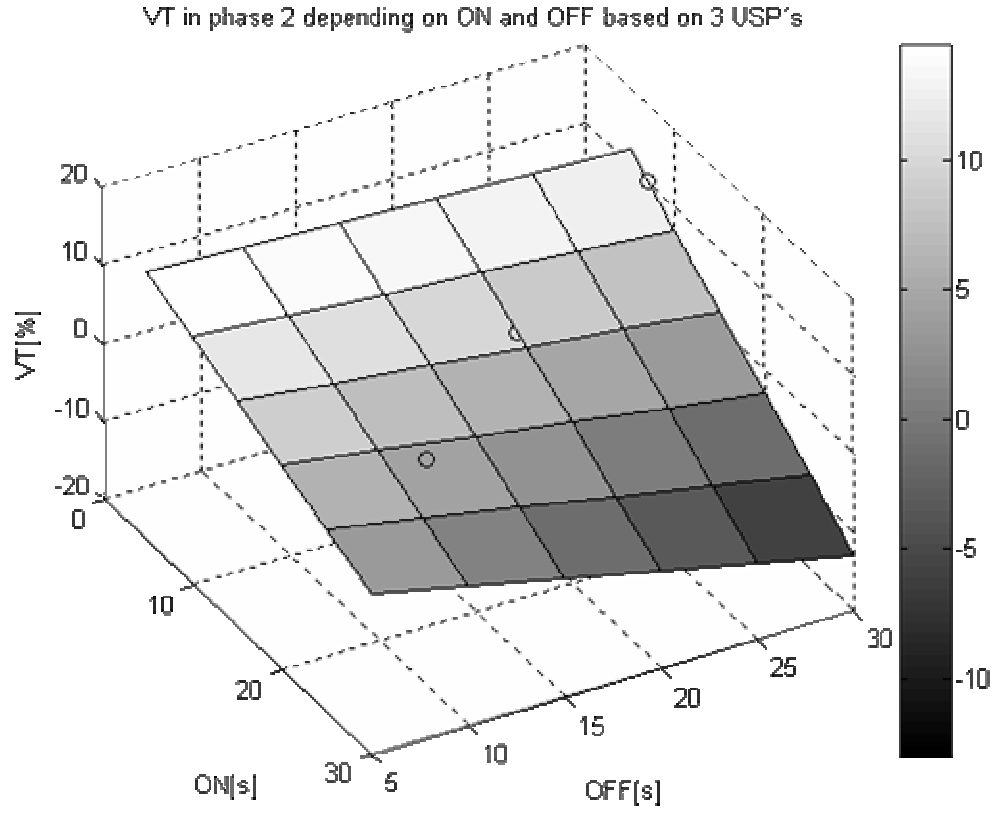

VS in phase 2 depending on F and A based on 9 USP´s
VS[%]
40
20
0
-20
40
30
20
10
0
F[Hz]
0
0.5
1
1.5
A[Amp]
25
20
15
10
5
0
-5
-10

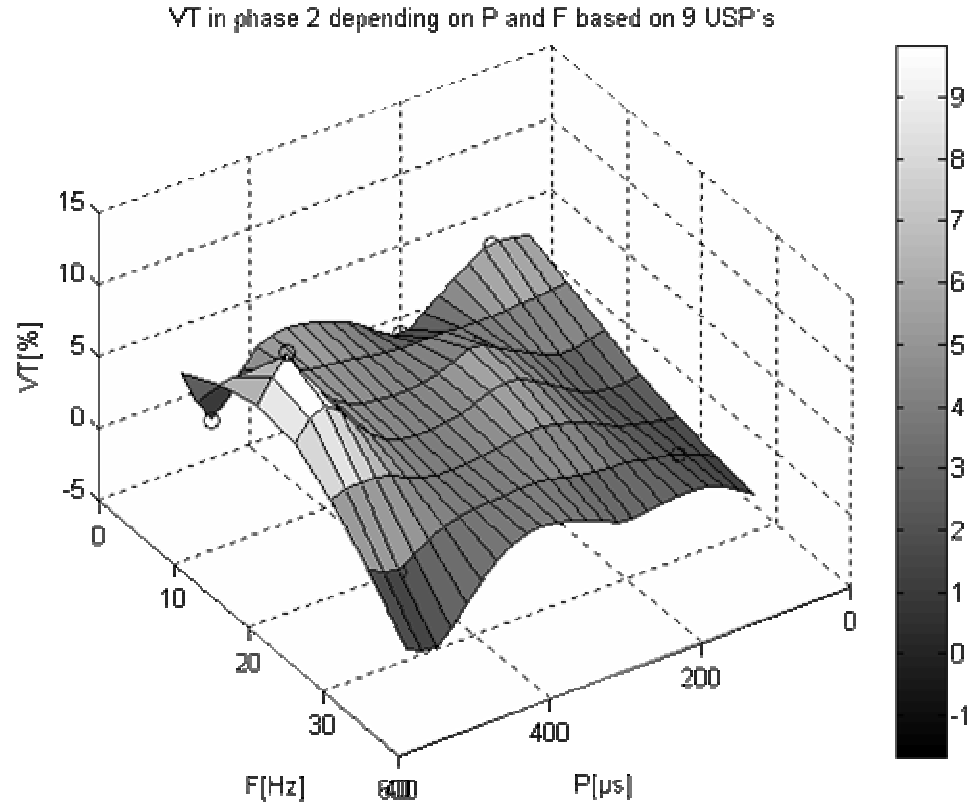

VS in phase 2 depending on ON/OFF ratio and P based on 9 USP´s
VS[%]
40
20
0
-20
600
500
400
300
200
100
0
P[µs]
0
0.5
1
1.5
2
On/Off ratio [AU]
30
25
20
15
10
5
0
-5

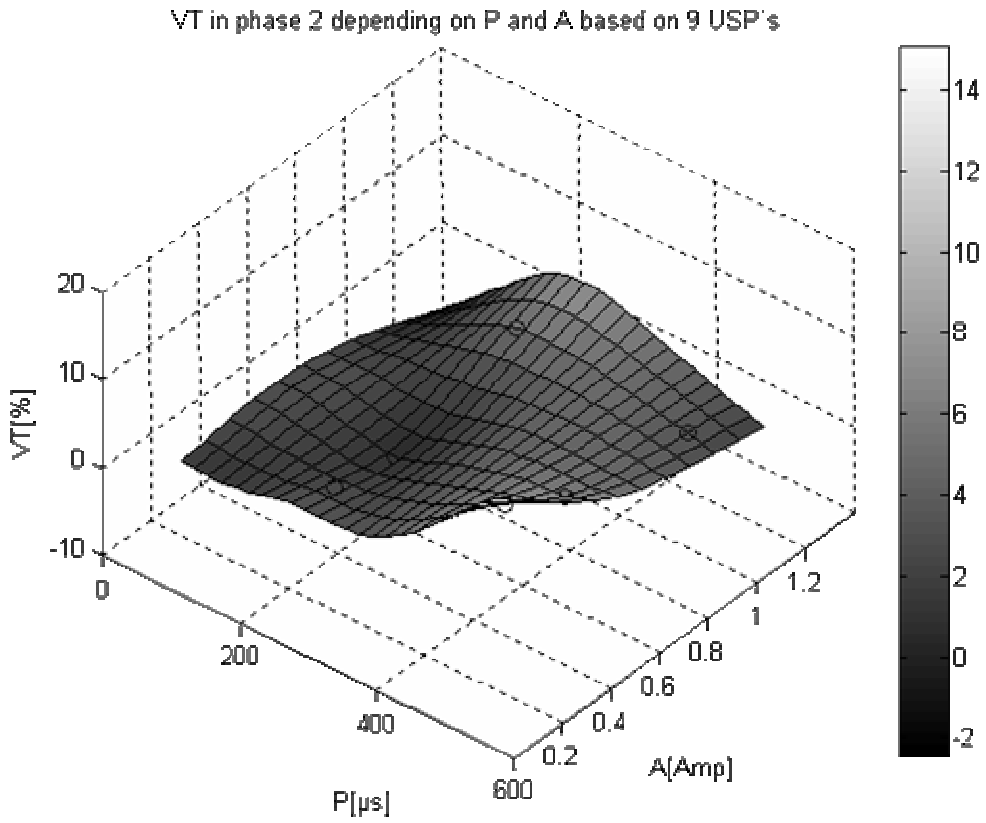

VS in phase 2 depending on ON/OFF ratio and A based on 9 USP´s
VS[%]
50
40
30
20
10
0
-10
2
1.5
1
0.5
0
On/Off ratio [AU]
0
0.5
1
1.5
A[Amp]
45
40
35
30
25
20
15
10
5
0

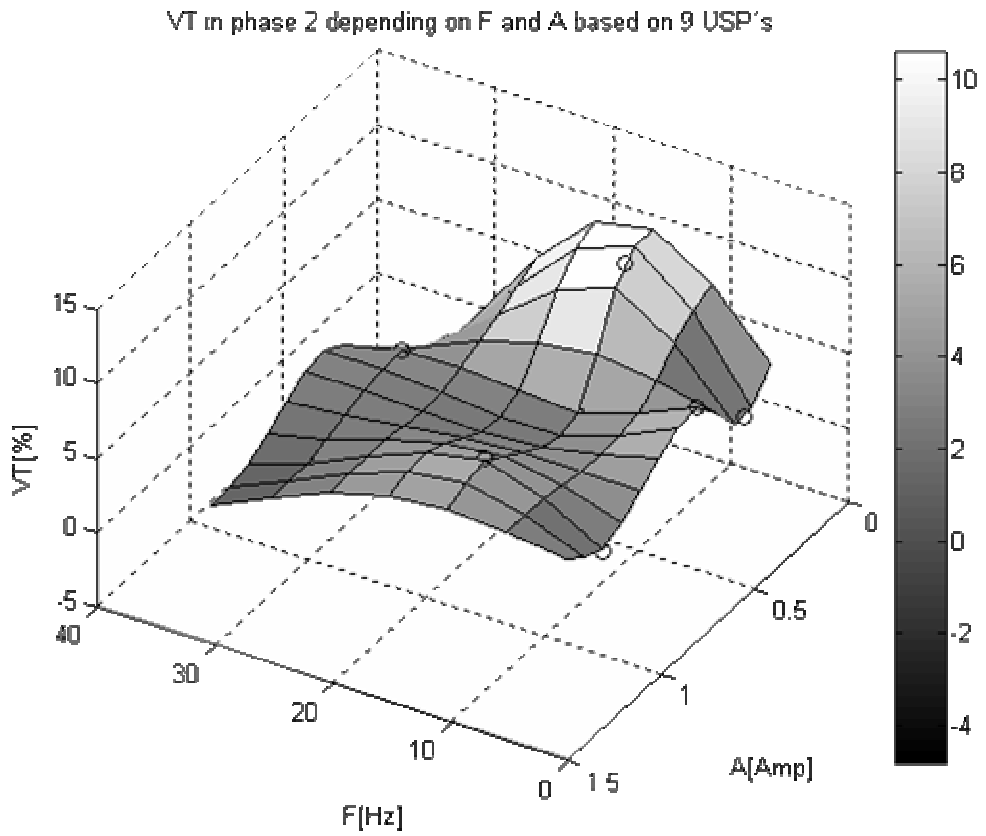


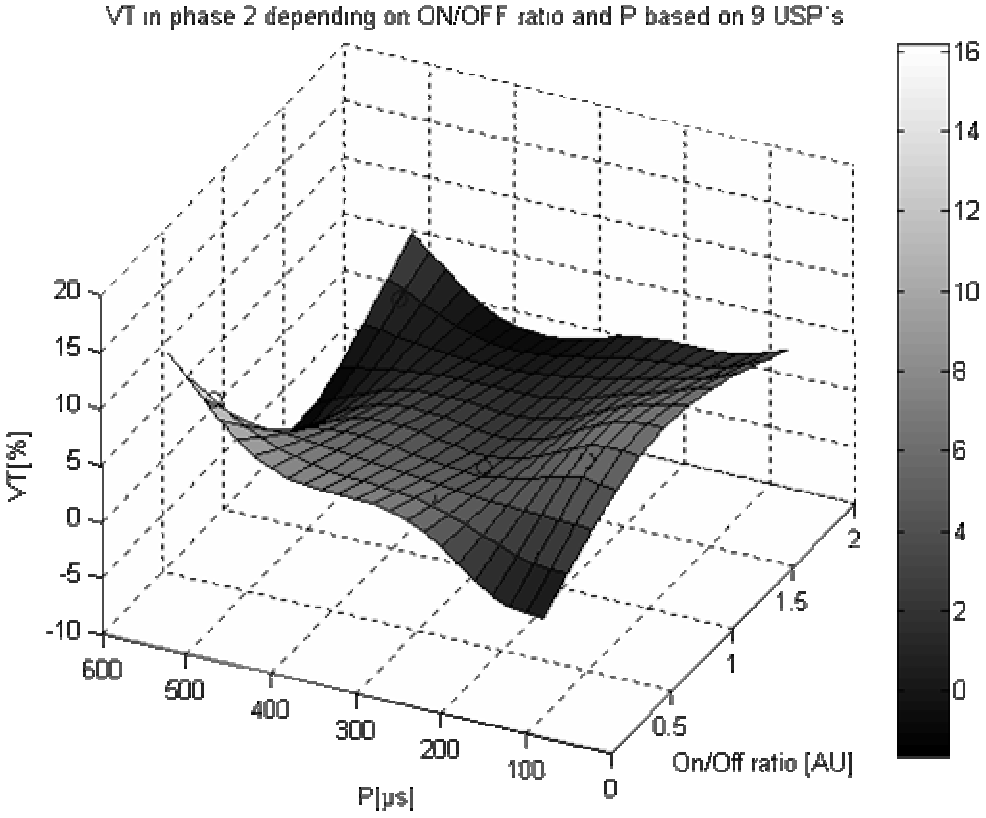
4.2.2.5 Minute Ventilation (Overall Ventilation Performance Indicator)

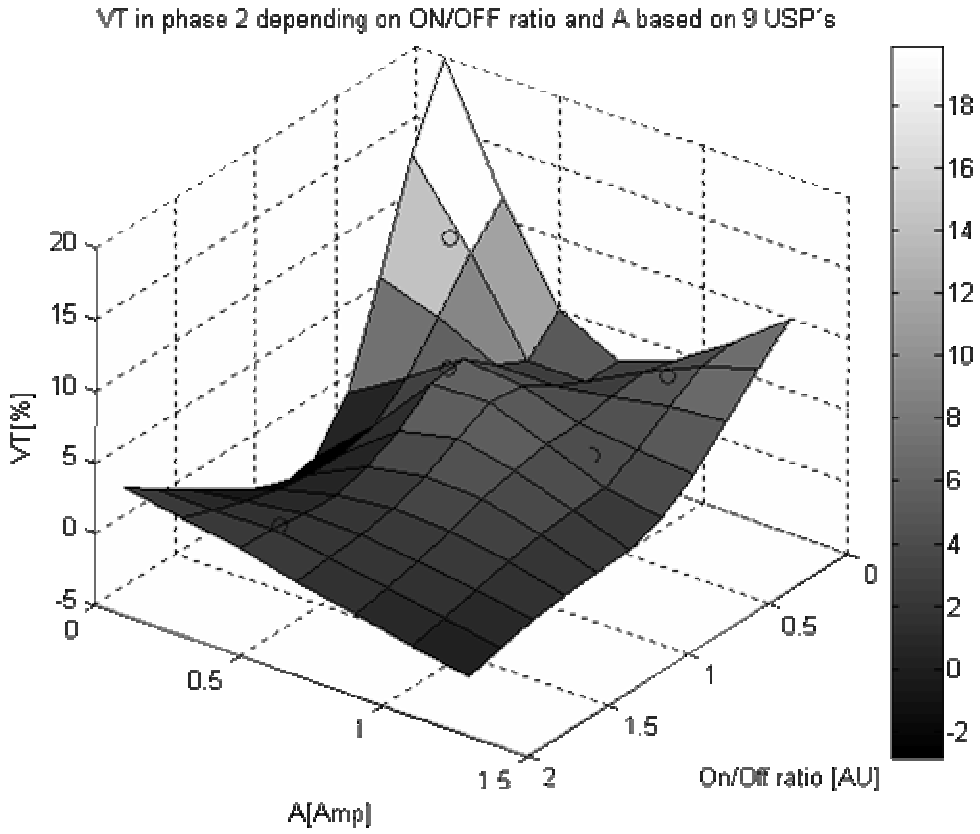


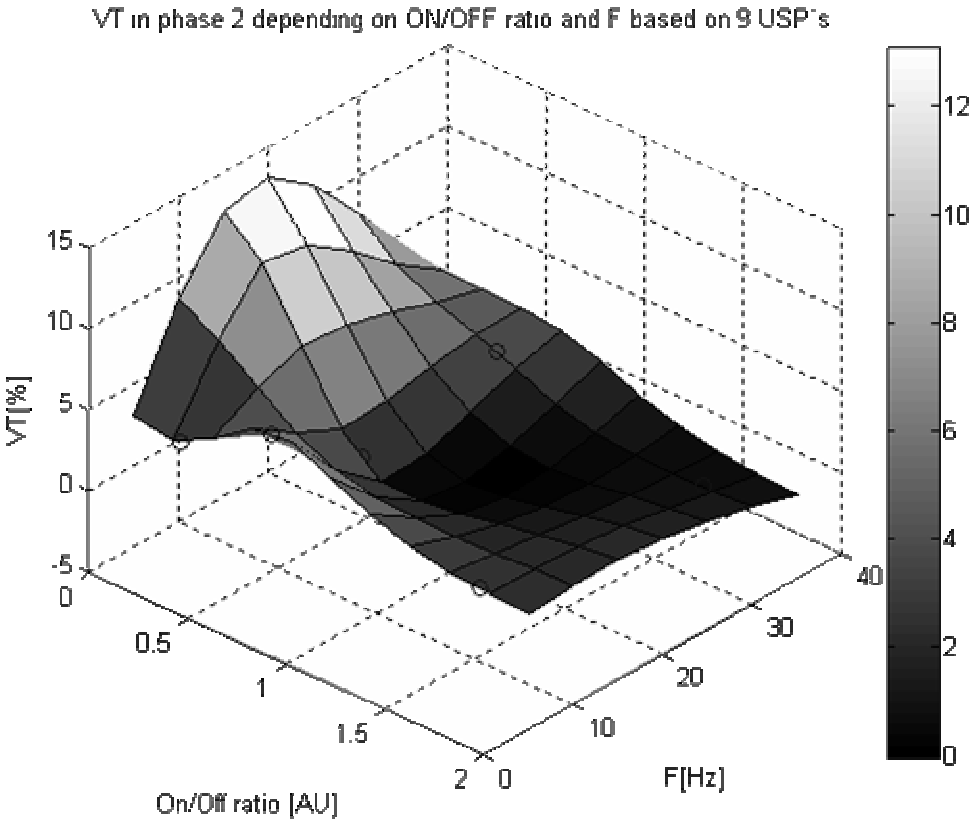

VT in phase 2 depending on P and F based on 9 USP's
VT[%]
15
10
5
0
-5
0
10
20
30
F[Hz]
0
200
400
P[µs]
9
8
7
6
5
4
3
2
1
0
-1


VT in phase 2 depending on P and A based on 9 USP's
VT[%]
20
10
0
-10
0
200
400
600
P[µs]
0.2
0.4
0.6
0.8
1
1.2
A[Amp]
14
12
10
8
6
4
2
0
-2


VT in phase 2 depending on F and A based on 9 USP´s
VT[%]
15
10
5
0
-5
40
30
20
10
0
F[Hz]
0
0.5
1
1.5
A[Amp]
10
8
6
4
2
0
-2
-4

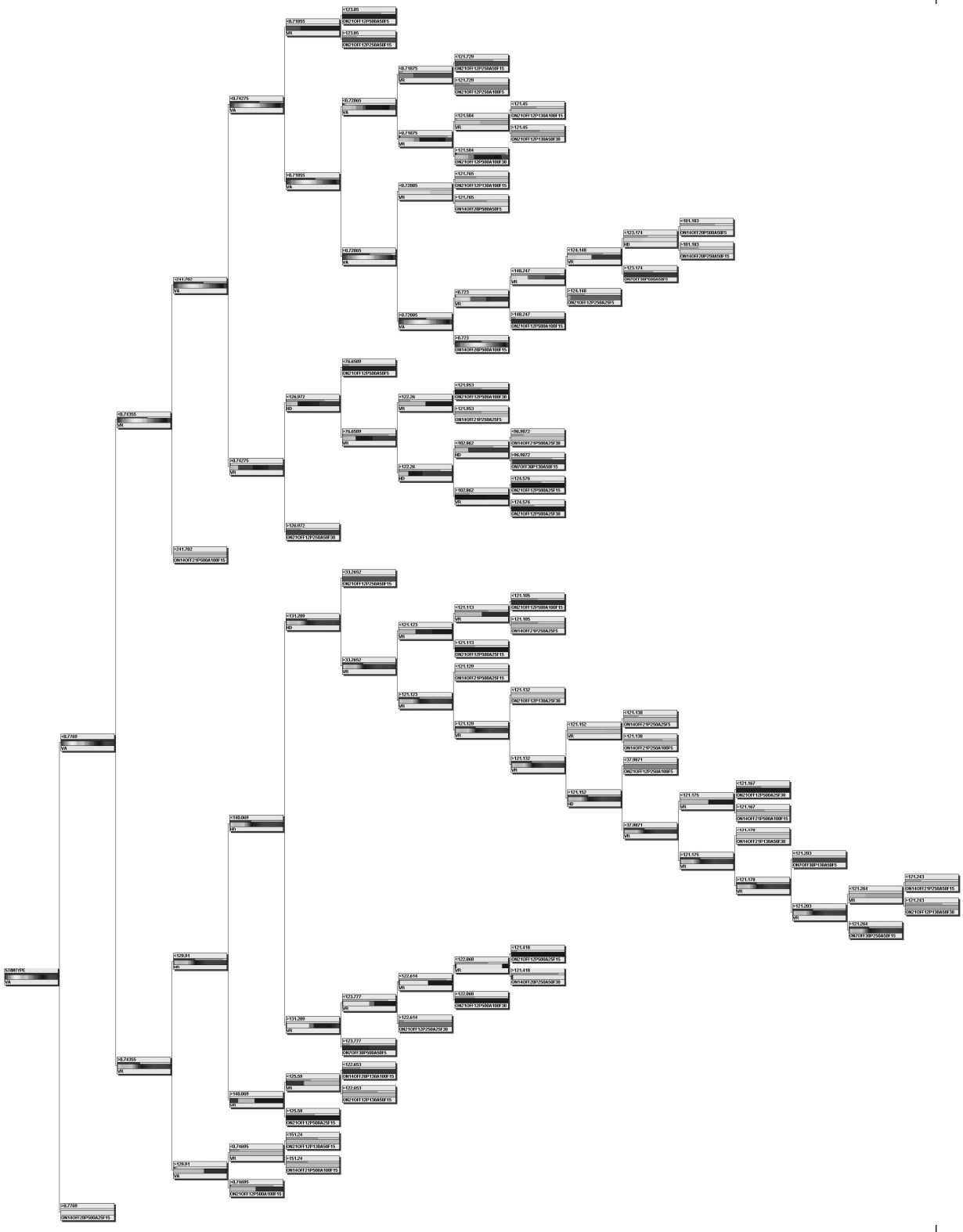

VT in phase 2 depending on ON/OFF ratio and P based on 9 USP´s
VT[%]
20
15
10
5
0
-5
-10
600
500
400
300
200
100
0
P[µs]
0
0.5
1
1.5
2
On/Off ratio [AU]
16
14
12
10
8
6
4
2
0


VT in phase 2 depending on ON/OFF ratio and A based on 9 USP´s
VT[%]
20
15
10
5
0
-5
0
0.5
1
1.5
2
A[Amp]
0
0.5
1
1.5
On/Off ratio [AU]
18
16
14
12
10
8
6
4
2
0
-2


VT in phase 2 depending on ON/OFF ratio and F based on 9 USP´s
VT[%]
15
10
5
0
-5
0
0.5
1
1.5
2
On/Off ratio [AU]
0
10
20
30
40
F[Hz]
12
10
8
6
4
2
0

4.3 Classification Tree Type 1

This tree is represented on the attached CD. The tree is included here in order to get an idea about the applicability of this method. The trees are rather large and offer a pinpoint knowledge but can be generated quickly with free software like InfoMiner or KNIME.



4.4 Results of phase compactation

Vector ID relates to the parameter vector in chapter 7.1.Whenever Phase 3 is not needed it is recommended to merge Phase 1 with Phase 3 in such a way, that features which are stronger in Phase 3 replace the weaker features in Phase 1.

4.4.1 Phase 1

Table 45: Average population effects observed during Phase 1.

Vector ID	VR[%]	VA[%]	VS[%]	VI[%]	VX[%]	VT[%]	HR[%]
1	0.48	0.67	8.24	1.97	2.7	5.24	0.82
2	-3.61	-22.95	-5.83	11.68	3.06	-2.37	-1.49
3	-0.95	-3.27	-3.18	-2.57	1.72	-1.65	0.39
4	-1.64	-9.93	-1.41	10.28	5.09	0.27	0.05
5	-8.75	11.42	-9.6	2.74	23.38	-7.15	-1
6	-3.86	13.72	-1.43	0.49	15.94	-2	-0.39
7	0.85	-4.19	-1.98	3.09	16.34	-2.7	-0.28
8	-1.75	-11.27	-6.07	-5.96	4.01	-3.14	0.31
9	-1.17	-4.58	0.98	0.89	2.68	0.11	-0.1
10	0.01	11.05	0.96	-4.89	2.53	-1.02	0.02
11	-1.85	14.72	-3.82	-4.79	6.85	-0.07	0.08
12	-0.19	22.49	-3.71	-4.37	6.5	-1.75	-0.34
13	-5.25	-23.2	-9.56	8.55	11.52	-3.93	-0.45
14	-0.67	10.48	-1.09	-3.72	6.98	-1.85	0.29
15	1.46	10.36	-1.33	-11.2	7.08	-3.05	-0.2
16	0.88	-22.69	-9.71	-0.8	10.7	-5.08	-0.02
17	1.46	12.12	-1.89	0.19	-2.41	1.92	0.11
18	-3.11	12.95	0.51	-0.09	8.09	1.23	-0.38
19	-0.78	-1.37	-4.88	0.16	6.03	-2.24	-0.15
20	-0.58	-4.97	-3.25	4.11	7.21	-2.26	0.15
21	7.83	-1.94	9.2	-4.3	4.34	2.31	-0.17
22	-3.58	-8.03	-0.62	-1.62	5.68	-2.59	-0.26
23	-0.51	2.18	-0.64	-1.14	3.23	-1.04	0.1
24	-1.55	0.74	0.61	2.16	4.67	0.84	0.36
25	-0.85	3.3	-5.19	-3.38	3.01	0.4	0.24
26	-0.72	8.85	5.3	-6.66	6.05	3.53	0.04
27	0.05	1.67	12.76	-4.63	-1.45	15.17	0.16
28	-3.93	-13.03	-1.2	11.14	0.37	-1.74	-0.23
29	1.97	-21.03	-6.41	14.51	-3.64	-0.41	-0.02
30	-4.85	-17.8	-3.27	15.76	2.13	-1.84	-0.09
31	-9.06	-17.27	-8.67	8.01	10.74	-3.44	0.01
32	-0.04	-31.15	-8.55	16.88	0.91	-2.72	0.17
33	-4.7	-35.22	-13.31	14.21	8.62	-4.29	0.54
34	-1.31	-22.98	-8.03	13.4	2.22	-3.54	0.73
35	-3.13	-5.05	-2.21	8.36	4.81	-0.37	0.02
36	0.26	-4.86	-1.26	4.94	3.03	-1.02	0.09

37	3.88	-9.57	-1.39	6.56	-1.59	0.61	0.21
38	-0.86	-26.31	-5.64	12.52	1.12	-0.82	-0.21
39	-2.84	-36.74	-14.6	20.75	0.48	-5.17	0.02
40	-3.1	-11.98	-6.76	7.83	4.36	-5.56	0.28
41	-1.09	-14.09	-4.29	11.2	0.3	-1.2	0.32
42	-7.29	-28.2	-4.33	20.13	-0.21	-2.14	0.43
43	-1.92	-4.57	-3.49	4.84	2.57	-1.06	-0.09
44	-7.32	-10.67	-8.17	9.98	5.44	-3.98	0.05
45	0.5	-1.1	-1.13	1.69	-1.44	-0.44	-0.05
46	-2.58	-16.67	-4.28	16.52	1.34	-0.12	0.26
47	-0.69	-12.28	-4.84	10.13	-0.96	-0.97	-0.04
48	-3.59	-9.34	-5.27	8.49	2.98	-0.86	-0.22
49	-0.83	-24.8	-3.54	13.86	-3.65	-0.01	0
50	-2.48	-10.39	-4.62	9.05	3.31	-0.62	0.29
51	-1.54	-12.3	7.51	9.1	-1.03	-4.44	-0.2
52	-3.22	-21.25	-3.93	16.18	-2.32	1.36	-0.26
53	3.7	-11.46	-2.91	10.28	-13.84	-5.12	-0.5
54	-6.45	-10.64	-7.36	10.11	8.36	-2.41	-0.38
55	0.83	-19.29	1.5	11.11	-5.58	6.19	-0.37
56	33.26	-43.74	-12.28	34.67	-1.95	-6.22	-0.34
57	13.33	-40.04	-12.48	20.68	0.06	-6.12	-0.18
58	-0.19	-19.19	-12.5	5.91	-4.87	-7.86	0.4
59	-4.25	11.87	4.93	0.51	23.86	1.81	-1.32
60	8.5	-0.64	-3.88	10.05	-15.4	-0.93	0.32
61	-1.47	-15.2	-0.33	10.19	10.69	-0.1	0.2
62	0	-7.66	-9.35	8.88	-1.89	0.38	1.59
63	-1.29	-19.45	-10.03	6.13	-16.13	0.42	1.62
64	7.06	-24.19	-4.31	16.89	-1.89	1.23	-0.73
65	-5.77	-0.05	-16.07	-5.82	11.43	-10.91	0.6
66	-7.89	14.2	3.12	5.49	19.38	2.19	0.13
67	-1.8	-11.34	2.35	29.03	6.71	1.35	0.95
68	-9.73	-21.53	-6.34	18.21	4.41	-8.12	0.25
69	-2.72	-24.7	-1.68	13.15	-0.82	4.88	-0.03
70	-3.47	-15.6	-6.81	13.24	-1.2	-2.61	0.13
71	0.13	-21.21	7.49	4.33	-2.92	6.4	1.72
72	-0.93	-5.19	-2.12	10.18	-1.54	0.15	2.26
73	-6.12	-23.53	-5.36	31.33	2.27	-3.55	-10.59
74	5.59	-10.07	-3.05	13.27	-11.56	3.25	-1.12
75	-2.4	-6.44	-1.85	7.37	3.26	-0.61	1.34
76	-2.07	-1.24	2.01	13.53	1.76	1.95	-0.13
77	-2.7	-11.94	-4.02	7.97	2.91	-1.84	-0.32
78	-0.61	-3.62	5.72	17.67	-2.82	1.49	0.18
79	-5.49	-13.96	-6.74	9.24	-9.1	-4.88	0.18
80	0.33	0.75	-1.81	-1.16	-2.7	-0.17	-0.02
81	-1.58	-14.22	-2.49	11	1.53	-0.3	-0.31

4.4.2 Phase 2

Table 46: Average population effects observed during Phase 2.

Vector ID	VR	VA	VS	VI	VX	VT	HR
1	-1.43	37.9	22.98	-9.62	3.09	10.02	2.26
2	8.12	63.79	12.12	-30.3	11.23	5.59	0.5
3	0.1	15.42	1.33	-9.61	2.14	2.82	-0.08
4	-3.04	49.27	7.33	-25.47	18.58	1.37	-0.09
5	-4.42	73.03	13.14	-32.58	21.06	3.24	-1.63
6	-7.68	75	11.82	-30.86	20.09	5.89	-0.83
7	-8.19	48.11	9.73	-19.2	14.28	2.02	0.04
8	0.37	9.5	3.31	-6.91	3.12	0.84	0.1
9	2.96	6.22	3.41	-5.98	-5.53	2.29	-0.06
10	-0.78	7.41	1.79	-1.49	3.73	0.41	0.01
11	-2.4	22.02	6.76	-13.02	8.55	1.29	-0.18
12	-4.07	48.83	9.28	-20.21	11.45	3.05	-0.44
13	-0.57	59.52	13.79	-29.35	12.82	6.21	-0.24
14	-0.67	10.07	2.74	-1.89	6.12	-0.44	-0.2
15	-0.34	21.19	5.23	-12.21	5.18	2.41	0.1
16	-3.29	39.53	7.83	-16.53	14	3.84	0.15
17	-0.13	20.57	3.9	-6.66	3.18	1.02	0.27
18	3.16	28.17	6.82	-20.02	2.55	2.94	-0.01
19	4.09	16.73	5.77	-8.98	9.89	2.96	-0.04
20	0.73	1.27	4.7	-1.51	5.01	1.98	0.04
21	-2.89	-12.13	-3.55	3.57	4.38	-4.44	-0.08
22	8.06	33	7.98	-23.83	7.73	4.05	0.16
23	0.76	8.24	3.69	-2.63	0.57	3.37	-0.05
24	2.34	20.94	3.11	-8.83	4.47	1.78	-0.01
25	2.08	21.72	1.19	-9.18	5.27	1.39	0.08
26	0.62	18.51	1.87	-7.48	3.47	1.64	0.19
27	-0.87	-26.75	5.08	-1.64	-0.16	1.8	-0.3
28	1.47	12.38	6.26	-11.72	6.32	8.59	0.17
29	-0.98	28.57	8.67	-16.36	8.01	0.78	-0.13
30	-2.07	23.38	3.02	-13.72	6.5	1.44	-0.14
31	-7.29	29.71	6.21	-11.63	15.02	-2.56	0.12
32	-11.24	45.12	8.45	-22.16	14.7	2.03	-1.52
33	-14.64	44.06	11.43	-17.61	24.22	1.06	-0.45
34	-14.03	32.71	8.86	-19.75	9.07	3.06	-0.49
35	-3.44	15.4	4.96	-7.34	2.93	0.98	-0.42
36	2.22	8.98	2.62	-9.49	1.95	1.15	0.12
37	-4.72	13.63	-2.15	-7.11	9.18	-1.87	-0.19
38	-2.13	26.28	2.35	-18.62	3.74	-0.2	0.41
39	-2.7	45.91	8.32	-23.13	12.57	0.76	-0.03
40	-7.08	37.22	11.33	-19.51	11.76	2.28	-0.39
41	-3.1	25.71	2.14	-13.2	11.35	-1.98	-0.12
42	-4.83	18.39	4.5	-9.64	7.55	1.32	-0.11
43	4.15	38.15	7.94	-19.64	7.23	3.53	-0.36

44	1.61	18.5	1.48	-14.41	6.75	0.67	0.13
45	-4.99	29.34	4.77	-20.7	13.9	-0.1	0
46	-1.5	5.85	2.09	-5.7	2.09	2.4	-0.03
47	-2.42	15.42	2.6	-10.94	4.45	0.96	0.31
48	-1.7	21.14	2.35	-17.59	8.79	2.41	-0.09
49	-0.48	23.39	4.84	-11.83	6.59	1.28	0.07
50	2.85	23.27	3.49	-16.62	4.19	0.22	0.32
51	0.41	26.25	5.19	-17.25	9.45	0.33	-0.1
52	-0.56	22.54	4.68	-24.57	8.07	2.3	-0.1
53	1.11	-0.47	2.59	-10.63	4.79	-8.05	0.08
54	2.79	0.47	2.16	-2.45	6.7	2.47	-0.57
55	-17.32	15.74	-8.88	-11.73	27.87	-3.57	1.12
56	0.21	20.78	3.82	-9.18	-7.74	1.21	0
57	-3.66	13.87	5.42	-6.86	15.33	-1.66	0.05
58	-23.46	35.2	5.52	-24.88	30.51	-3.39	0.77
59	-29.02	36.1	10.58	-8.1	25.33	-0.97	0.09
60	-0.11	-9.27	-6.88	-21.46	-17.38	-0.8	0.01
61	0.82	19.76	1.08	-5.52	-1.95	3.77	-0.31
62	-0.27	5.92	-7.78	-9.2	-5.26	-6.45	-0.1
63	3.37	14.06	2.7	-8.78	9.72	-4.33	-0.17
64	-0.86	16.38	7.68	-9.91	11.96	4.52	-0.81
65	-5.94	14.4	3.09	-29.05	13.78	-2.38	-0.96
66	-0.64	17.31	8.11	-1.45	1.56	2.23	1.51
67	0.17	8.78	0.14	-5.27	17.01	-1.46	-1.9
68	0.13	6.88	2.38	-9.78	-3.97	-0.06	0.08
69	-6.7	11.72	-9.03	-16.83	18.81	-9.99	-0.32
70	9.26	14.64	4.12	-11.4	-0.87	2.33	-0.38
71	-2.38	12.94	-2.17	-9.67	8.83	-1.42	0.8
72	1.69	14.78	2.45	-10.63	5.32	2.31	0.27
73	-1.79	0.44	-4.68	5.56	12.85	-6.13	1.97
74	2.81	3.01	-0.21	2.94	0.07	-2.15	0.62
75	2.36	19.47	0.83	-19.19	-1.2	2.05	-6.75
76	-4.26	23.16	-3.21	-10.85	1.77	-1.68	0.98
77	0.32	11.85	-5.86	-9.69	8.39	-5.73	-0.78
78	1.14	14.05	3.53	-8.51	4.57	-1.12	-1.03
79	6.66	16.5	5.59	-27.5	9.54	3.63	0
80	-1.62	-0.52	-6.2	-6.28	3.58	-5.61	-0.38
81	0.73	8.68	2.64	-9.35	2.83	1.34	0.1

4.4.3 Phase 3

Table 47: Average population effects observed during Phase 3.

Vector ID	VR	VA	VS	VI	VX	VT	HR
1	-0.29	-15.55	0.77	8.15	-3.67	-0.32	0.24
2	-0.21	-6.59	-2.03	3.79	-2.65	-0.58	-1.59
3	-1.47	-33.7	-7.17	17.51	-12.09	-5.26	0.28
4	-1.62	-5.11	0.47	6.16	-1.92	-1.26	-0.03
5	2.31	-21.56	-3.15	11.63	-13.16	0.22	0.39
6	3.02	-37.08	-7.37	15.25	-20.52	-0.21	1.95
7	2.31	-33.69	-7.77	12.98	-15.12	-1.62	0.8
8	1.84	-20.04	-6.24	9.14	-11.69	-0.08	0.29
9	0.93	5.3	0.45	4.2	0.13	0.15	-0.03
10	-3.02	-3.17	-3.46	5.49	2.15	-0.5	0
11	-0.65	-5.25	-1.73	6.42	-4.61	-1.51	-0.14
12	2.49	-14.53	-4.42	4.79	-6.49	-0.7	0.32
13	1.28	-24.15	-3.03	12	-11.14	0.51	0.71
14	1.83	-27.01	-4.15	12.68	-14.39	-2.11	0.37
15	0.85	-7.39	-0.48	1.05	-4.27	-1.05	-0.14
16	-1.01	-10.53	-1.32	7.77	-4.05	-2.35	-0.08
17	0.29	-25.05	-0.88	9.48	-10.28	-0.07	-0.04
18	-2.78	-3.26	-1.65	6.82	2.04	0.15	-0.23
19	-0.86	-13.41	-5.13	9.53	-3.93	-4.29	0.2
20	0.98	-0.45	-0.7	1.52	0.9	-1.31	-0.17
21	0.18	-8.61	-0.67	-0.72	4.57	0.42	0.04
22	-1.66	-24.92	-4.41	10.89	-8.69	-1.9	0.23
23	-0.6	-6.47	-0.87	3.58	-2.82	-1.69	-0.08
24	0.32	-1.6	-1.68	2.2	-2.54	-1.36	-0.08
25	-0.16	-14.29	0.25	7.24	-5.9	-0.44	-0.25
26	-0.62	-0.35	-0.06	4.51	-0.4	-1.54	0.19
27	0.59	-14.23	-6.15	4.7	-1.79	7	-0.05
28	0.32	-17.62	-0.52	13.49	-7.11	-2.45	0.06
29	3.86	-26.03	-2.5	14.86	-10.9	0.78	0.11
30	1.86	-20.41	-2.72	11.12	-5.07	0.18	0.14
31	1.5	-19.05	-3.22	14.98	-9.82	0.16	0.05
32	6.08	-26.55	-3.38	12.05	-10.76	0.66	0.4
33	4.82	-35.92	-6.72	18.5	-18.22	1.73	1.17
34	11.24	-43.79	-8	20.75	-27.65	0.93	0.62
35	11.31	-29.75	-7.52	19.62	-24.72	-0.25	0.6
36	3.12	-11.96	-1.23	11.07	-11.17	0.84	0.07
37	-1.54	-8.25	-2.09	4.83	-1.57	-0.03	0.07
38	3.17	-8.26	0.93	7.06	-6.79	0.83	0.02
39	1.48	-19.68	-2.78	13.04	-10.38	-0.04	0.28
40	2.96	-37.19	-5.86	19.28	-12.35	-0.27	-0.47
41	1.56	-21.52	-1.11	11.23	-10.04	0.81	-0.01
42	5.86	-14.53	2.66	7.15	-9.84	3.52	0.07
43	-2.34	-28.4	-3.96	18.29	-4.42	-1.08	0.05

44	0.15	-10.86	-1.16	7.57	-1.75	-1.5	0.03
45	0.7	-21.41	0.76	20.12	-11.68	0.68	-0.01
46	2.16	-27.16	-1.78	15.99	-13.86	1.66	-0.04
47	-0.56	-8.42	2.61	6.54	-3.08	2.81	0.07
48	1.85	-12.99	-0.74	8.46	-8.16	1.29	0.21
49	2.24	-19.44	-0.91	18.23	-9.62	-0.99	-0.12
50	0.77	-18.52	-3.62	13	-3.62	-1.23	0.22
51	-1.45	-17.44	-1.84	13.44	-6.07	-0.31	-0.07
52	0.81	-22.27	-3.26	14.84	-7.39	-0.7	0.08
53	-1	-7.25	-1.33	11.31	-3.36	-0.71	-0.09
54	-2.34	-16.88	3.7	11.07	-1.45	-2.09	0.08
55	4.22	-8.45	0.14	5.6	-4.94	0.72	0.18
56	1.34	-2.53	-0.58	8.7	-13.54	-1.37	-0.77
57	-1.68	-10.63	-5.07	9.78	9.8	-4.53	0.26
58	10.74	-4.04	1.77	0.94	-5.15	5.33	0.48
59	20.32	-27.5	2.93	17.63	-26.25	6.43	-0.5
60	19.32	-21.46	1.76	7.97	-28.06	4.37	0.01
61	3.6	-8.73	-12.14	-2.98	-16.87	-9.68	0.1
62	-2.44	4.52	4.08	1.85	11.23	0.61	-0.51
63	1.82	6.8	-0.45	5.15	-1	-2.01	-0.22
64	-0.17	-8.94	-2.88	5.52	4.25	-7.5	0.45
65	3.09	-3.07	1.03	6.57	-10.01	0.2	0.46
66	7.42	-10.68	-3.38	-0.84	-11.92	-0.47	0.99
67	2.83	-10	-1.6	0.05	-12.17	0.06	-0.5
68	0.25	-19.09	0.17	2.54	7.97	-2.11	0.76
69	6.19	-5.14	6.57	14.11	0.38	4.1	0.26
70	0.82	-4.41	2.7	9.61	3.17	0.9	0.42
71	0.47	4.59	2.91	2.21	-4.54	2	0.02
72	1.62	-8.21	-1.85	13.28	-8.97	-0.15	-0.35
73	3.24	-4.54	2.14	3.2	-2	2.23	-0.02
74	-5.07	1.06	-1.07	3.06	11.73	1.54	-0.11
75	-1.33	-3.88	2.57	2.91	2.92	1.25	-6.59
76	-0.46	-5.52	2.07	10.8	-12.14	0.69	-3.19
77	3.57	-8.75	-3.89	5.08	-5.36	1.38	-0.34
78	0.18	3	4.74	8.04	3.77	2.6	0.17
79	-2.07	-1.2	-4.84	6.72	3.16	-4.01	1.25
80	4.62	-7.63	4.24	-1.38	-6.2	2.07	-0.13
81	2.57	-4.39	2.99	8.37	-3.95	1.04	0.34

4.5 Results of the K-NN application

In contrast to the neural network a list is fairly easily generated (as presented in chapter 3.11.6). The usage of the classifier is also explained in that chapter. Here only examples of possible hypothesis are presented. The following hypotheses use L1:

Hypothesis 1:

VR	VA	VS	VI	VX	VT	HR
-5%	-	-	-	-	-	0%

Answer to Hypothesis 1:

ON[s]	OFF[s]	P[µs]	A[mA]	F[Hz]
20	20	140	1,1	5
20	20	140	0,9	5
15	20	100	1,1	15
15	20	100	0,9	15
20	20	140	1,3	5
25	30	100	1,3	5
25	30	100	0,9	5
25	30	100	1,1	5
20	20	180	0,9	5
20	20	180	1,1	5

Hypothesis 2:

VR	VA	VS	VI	VX	VT	HR
-2%	+1%	-	-	-	+3%	2%

Answer to Hypothesis 2:

ON[s]	OFF[s]	P[µs]	A[mA]	F[Hz]
20	10	140	0,3	5
20	10	180	0,3	5
20	15	140	0,3	5
20	15	180	0,3	5
20	10	100	0,3	5
15	10	220	0,1	5
25	25	220	0,1	5
20	15	100	0,3	5
15	10	260	0,1	5
25	25	260	0,1	5

Because there are no validated hypotheses up to this day it is not clear what an optimal parameter vector should look like!

4.6 Uncertainty Tables

The following tables show the average standard deviation within homogeneous groups of values for each parameter. The closer the value to zero the more relevant is the parameter for a given feature. The following tables are not normalized and are therefore difficult to compare. Because of that the tables were reproduced with values normalized to the min/max range of deviations.

Table 48: UA: Average randomness in % in the relationship between input and output. The smaller the numbers the more important is the input parameter quantitatively.

Phase 1	VR	VA	VS	VI	VX	VT	HR
T_{ON}	4,67%	11,63%	5,21%	6,39%	6,24%	3,42%	0,98%
T_{OFF}	4,67%	11,63%	5,21%	6,39%	6,24%	3,42%	0,98%
P	5,10%	13,73%	5,29%	8,84%	7,09%	3,47%	1,10%
A	2,89%	10,70%	5,64%	7,37%	3,58%	4,20%	0,45%
F	3,27%	10,27%	6,30%	6,86%	6,20%	4,75%	0,63%

Table 49: UA: Average randomness in % in the relationship between input and output. The smaller the numbers the more important is the input parameter quantitatively.

Phase 2	VR	VA	VS	VI	VX	VT	HR
T_{ON}	5,64%	15,52%	4,64%	7,96%	7,46%	2,97%	0,84%
T_{OFF}	5,64%	15,52%	4,64%	7,96%	7,46%	2,97%	0,84%
P	5,75%	17,47%	5,17%	8,39%	7,54%	3,33%	0,92%
A	5,15%	14,57%	5,06%	7,60%	6,79%	3,35%	0,89%
F	5,24%	15,49%	5,02%	6,67%	7,26%	3,24%	0,89%

Table 50: UA: Average randomness in % in the relationship between input and output. The smaller the numbers the more important is the input parameter quantitatively.

Phase 3	VR	VA	VS	VI	VX	VT	HR
T_{ON}	3,60%	9,48%	3,13%	4,67%	7,58%	2,33%	0,78%
T_{OFF}	3,60%	9,48%	3,13%	4,67%	7,58%	2,33%	0,78%
P	4,12%	10,82%	3,33%	5,67%	7,78%	2,50%	0,82%
A	3,66%	10,16%	3,23%	5,57%	7,13%	2,42%	0,87%
F	3,54%	9,80%	3,15%	5,09%	6,75%	2,51%	0,88%

The following three tables are based on the same information as the above three ones but are normalized to the min/max-range. Here a hint to read the tables properly: 50% means that the input is basically random to this parameter. 100% means, that the investigate parameter is the sole parameter to the feature function.

Table 51: Relevance of parameters for a specific outcome in Phase 1 (normalized)

Phase 1	VR	VA	VS	VI	VX	VT	HR
T_{ON}	78,27%	64,88%	63,86%	72,14%	68,79%	73,77%	84,74%
T_{OFF}	78,27%	64,88%	63,86%	72,14%	68,79%	73,77%	84,74%
P	76,27%	58,54%	63,31%	61,46%	64,54%	73,39%	82,87%
A	86,56%	67,69%	60,88%	67,87%	82,09%	67,79%	92,99%

F	84,79%	68,99%	56,30%	70,09%	68,99%	63,57%	90,19%

Table 52: Relevance of parameters for a specific outcome in Phase 2 (normalized)

Phase 2	VR	VA	VS	VI	VX	VT	HR
T_{ON}	70,53%	69,49%	70,98%	58,24%	68,83%	70,36%	81,42%
T_{OFF}	70,53%	69,49%	70,98%	58,24%	68,83%	70,36%	81,42%
P	69,96%	65,66%	67,67%	56,02%	68,53%	66,74%	79,47%
A	73,09%	71,36%	68,36%	60,17%	71,62%	66,48%	80,20%
F	72,62%	69,54%	68,64%	65,01%	69,68%	67,65%	80,32%

Table 53: Relevance of parameters for a specific outcome in Phase 3 (normalized)

Phase 3	VR	VA	VS	VI	VX	VT	HR
T_{ON}	71,64%	62,54%	66,49%	60,64%	61,92%	72,06%	81,69%
T_{OFF}	71,64%	62,54%	66,49%	60,64%	61,92%	72,06%	81,69%
P	67,54%	57,22%	64,38%	52,25%	60,89%	69,96%	80,87%
A	71,16%	59,85%	65,49%	53,06%	64,16%	71,00%	79,63%
F	72,11%	61,26%	66,28%	57,11%	66,08%	69,87%	79,35%

4.7 Sorted Diagrams

4.7.1 Ventilation Rate (Phase 1)

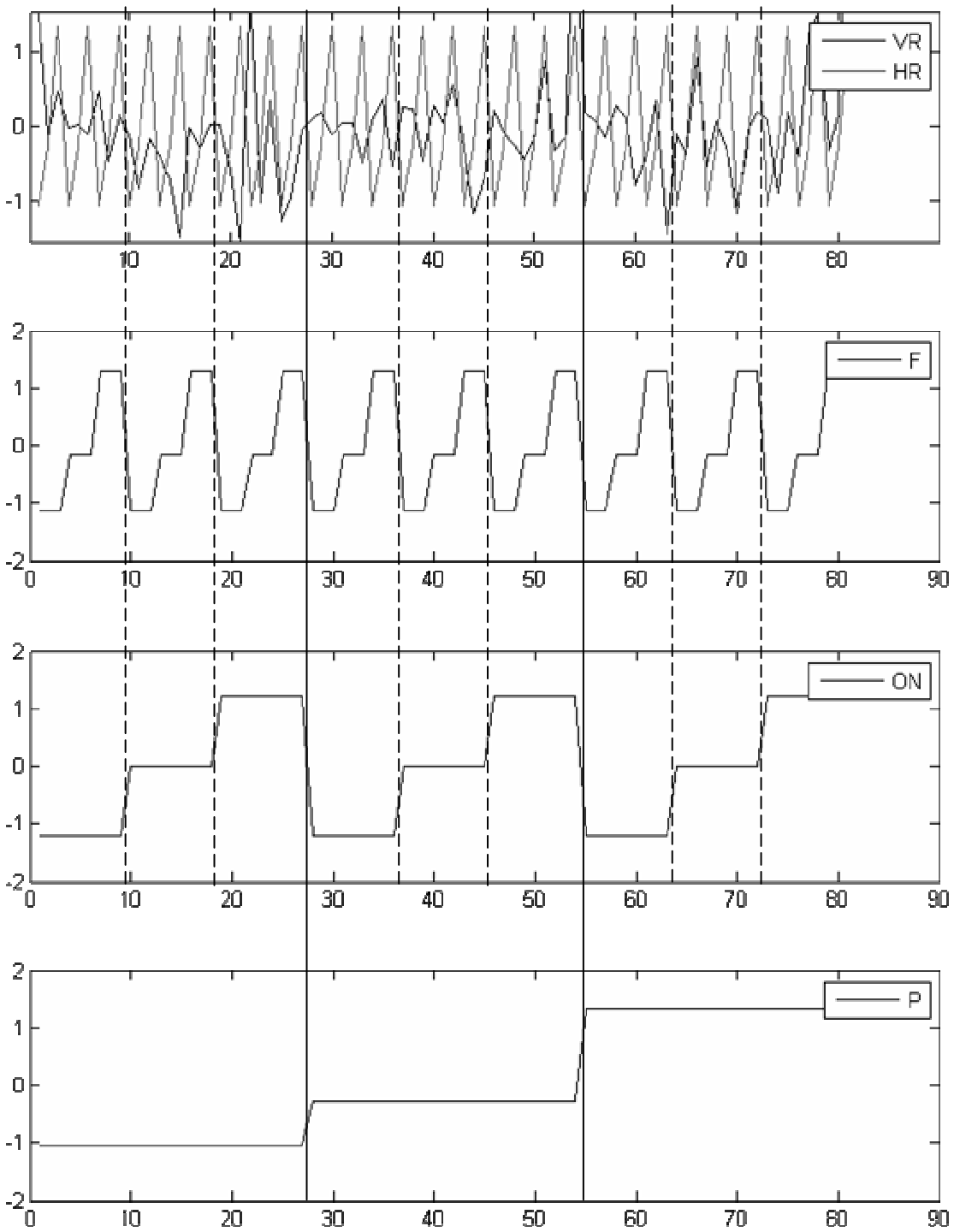


Image 73: An informative correlation between VR and A is developed when first sorted by P, then by ON (ON/OFF) and then by F.

The High frequency seems to be the dominating factor for the VR alteration strength. Therefore stimulation frequency seems to be the second most important factor in controlling VR

alterations. The third most important factor seems to be the time because with high values it activates a positive relationship between current and VR increase. The softest control over the VR effects comes from the parameter P gently intensifying the effects described above.

4.7.2 Ventilation Amplitude (Phase 1)

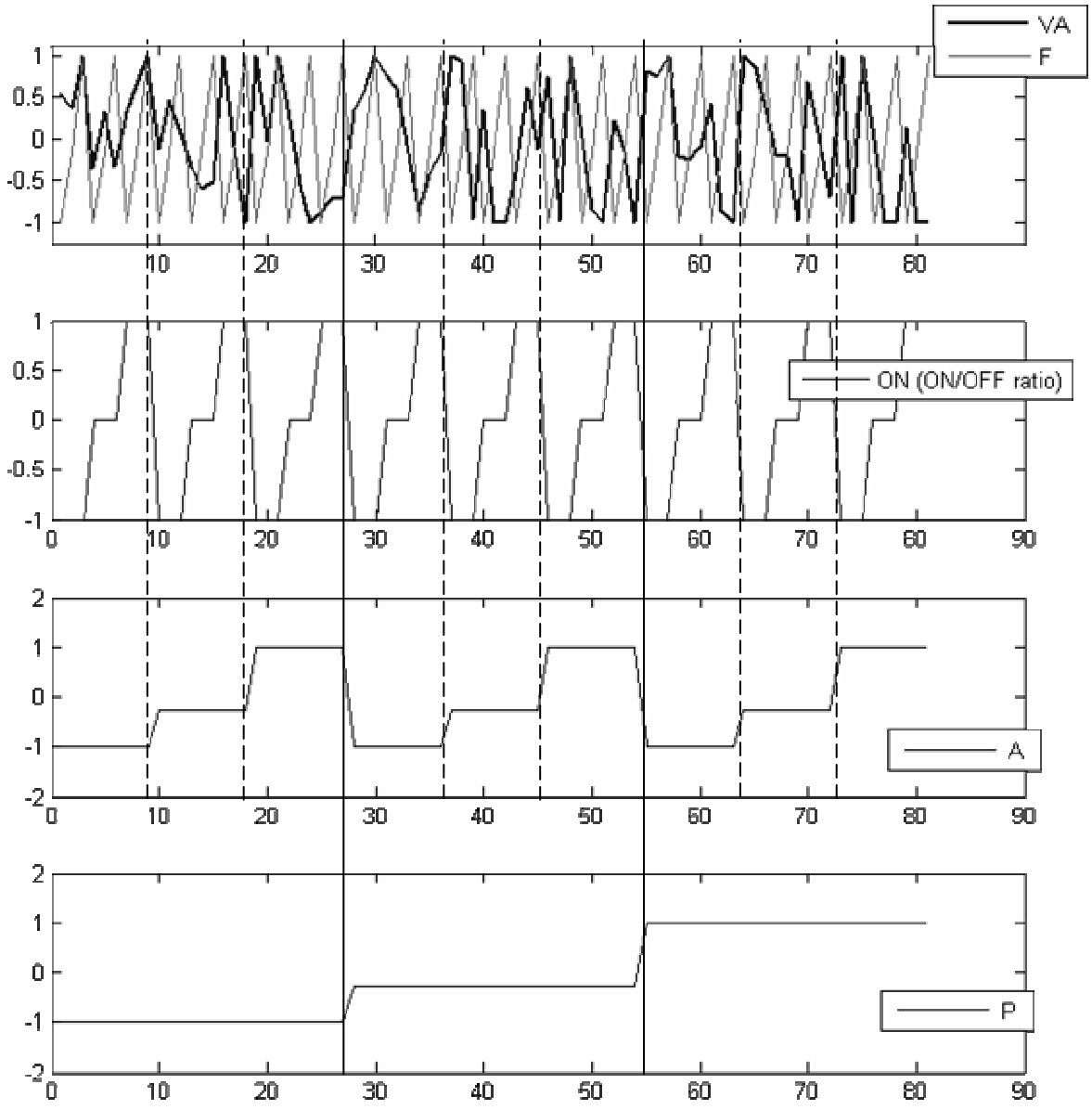


Image 74: An informative correlation between VR and A is developed when first sorted by P, then by ON (ON/OFF) and then by F.

The VA diagram is difficult to interprete because no verz clear coincidence can be seen between the frequency F and the feature occurrence. Conclusion over this diagram is that VA tendency cannot be predicted by some thumb rules.

4.7.3 Ventilation Slope (Phase 1)

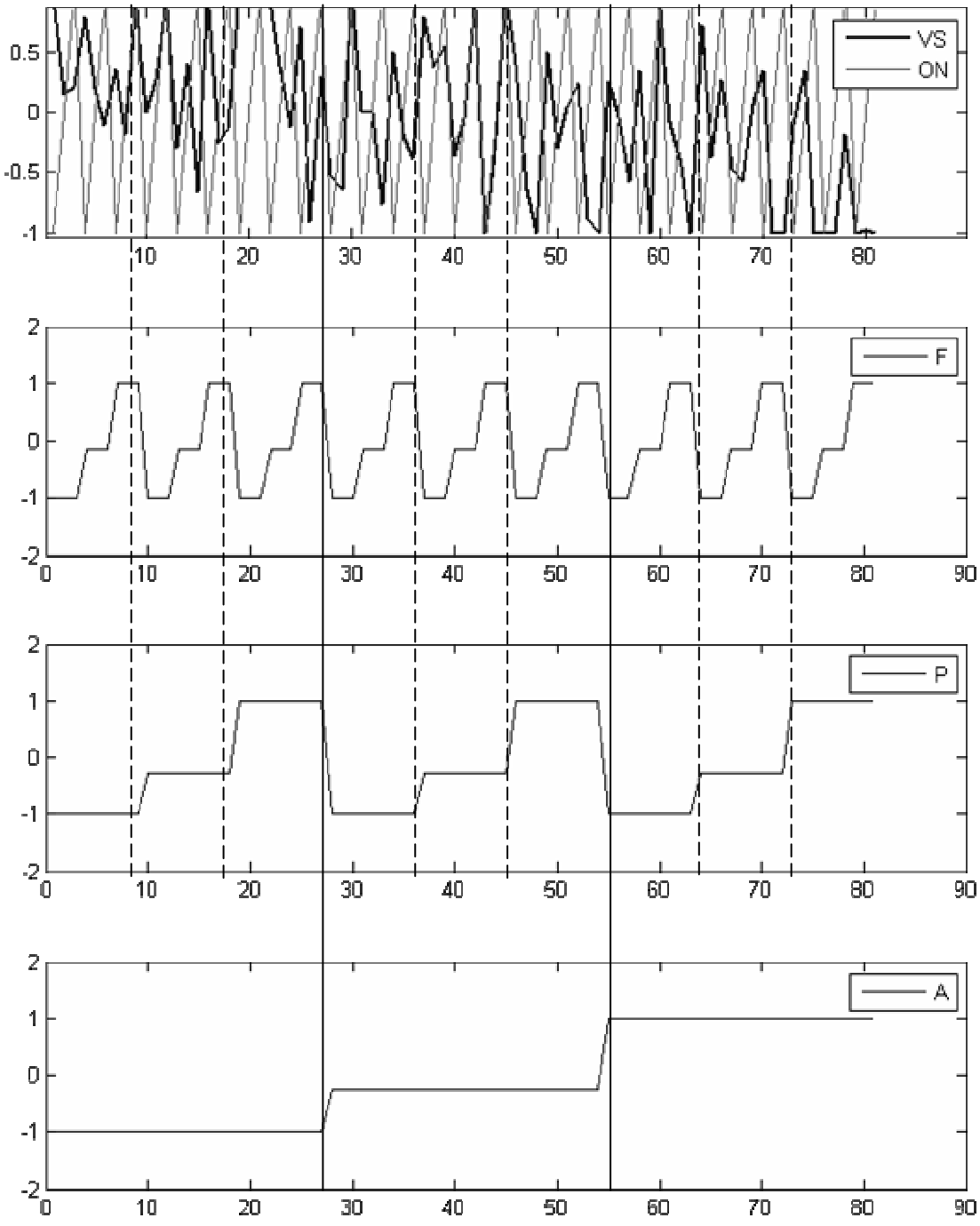


Image 75: An informative correlation between VS and A is developed when first sorted by P, then by ON (ON/OFF) and then by F.

The VS behavior begins to become systematic once a threshold current has been crossed. This seems to be somewhere around 0.5mA. For pulse widths wide enough (P>250µs) the ventilation slope changes pretty systematically with the stimulation time or stimulation/pause-ratio. For lower values of P rather inverted relationship seems to be observed where higher stimulation times cause a decrease in slope. It can be read from the diagram, that the direction of the

slope feature seems to be depending on both F and ON fairly equally. Higher values for F seem to influence the slope toward lower values.

4.7.4 Inhalation Time (Phase 1)

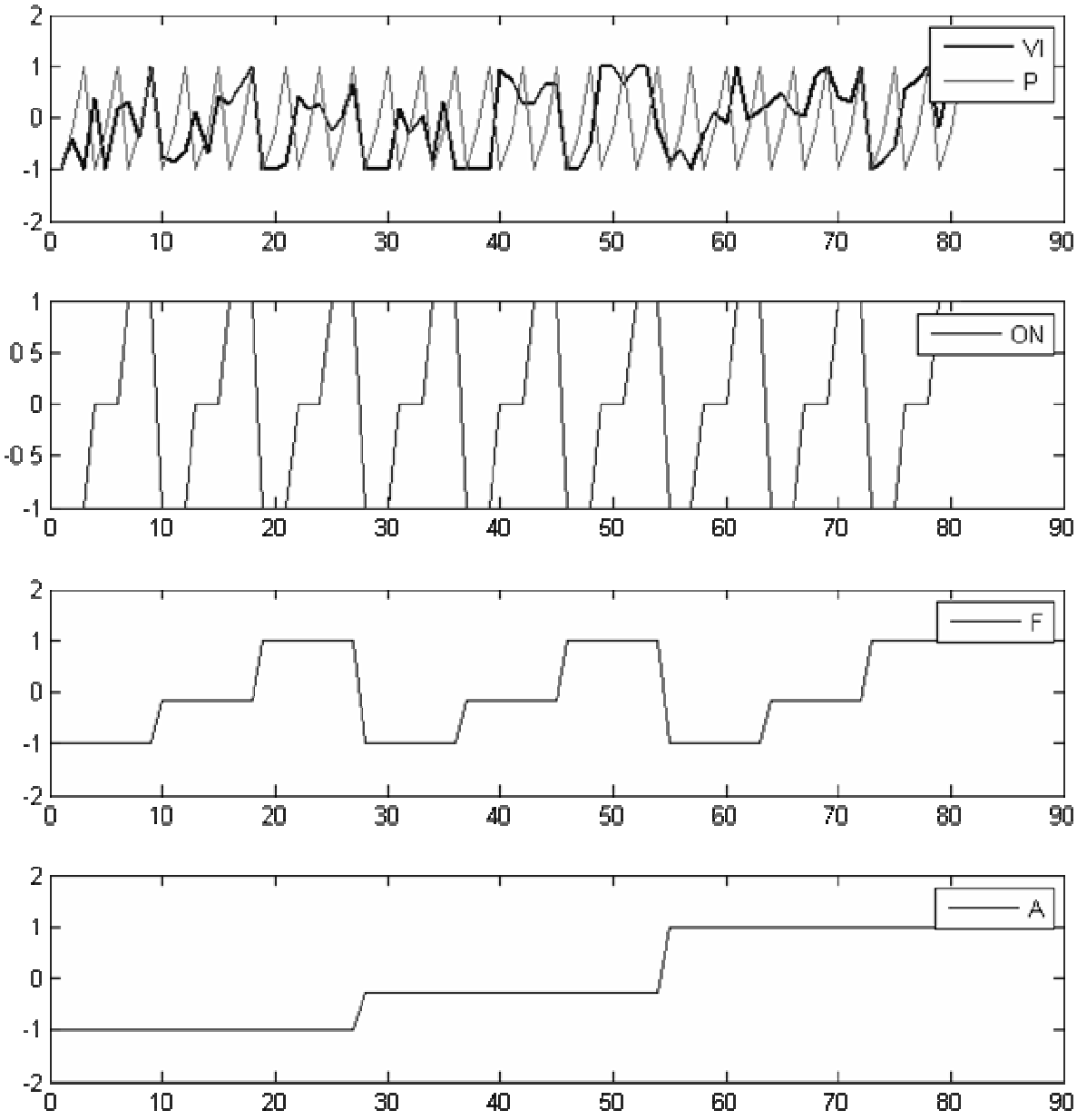


Image 76: An informative correlation between inhalation time, pulse width P and stimulation time ON is developed when first sorted by A, F, ON (ON/OFF) and then by P.

The inhalation time shows a rather systematic behavior in low and middle range current strengths (A<1mA). In that case inhalation times increase only when stimulation times are longer than 10s. If the stimulation time is shorter then inhalation time seem to decrease. Frequency plays a minor role but works like an amplifier factor where higher frequencies favor increase in inhalation time.

4.7.5 Exhalation Time (Phase 1)

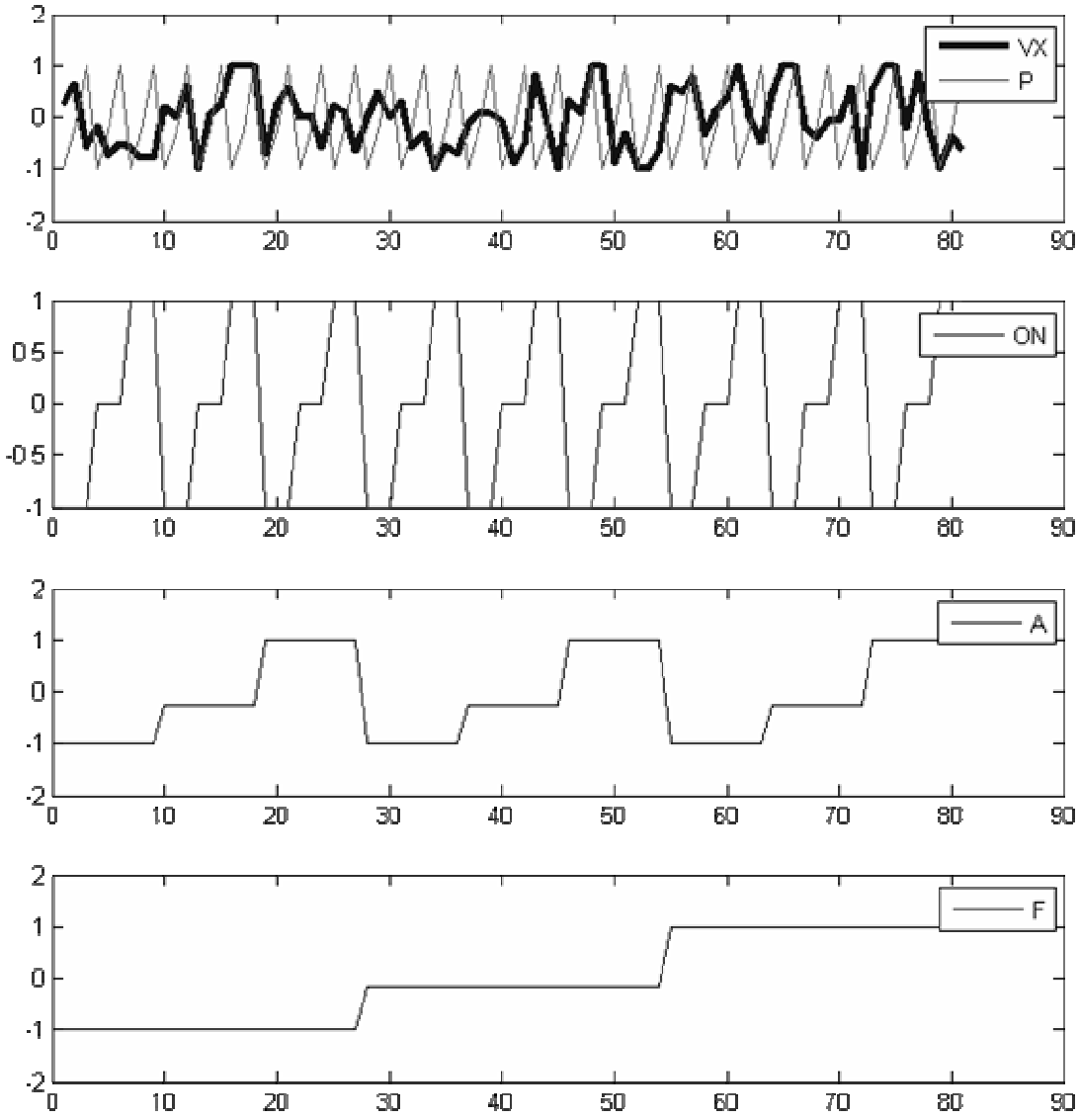


Image 77: An informative correlation between exhalation time, pulse width P and stimulation time ON. Sorted in the order F, A, ON, P.

The exhalation time has a fine structure that seems to be depending on the pulse width P. But this structure does not dominate the overall look of the VX-curve. It rather seems that there are overlaid structures in increasing strength. The stimulation duration seems to have a negative effect on exhalation time making it shorter the longer the stimulation. The overall shape seems rather to be affected by current strength. At low values exhalation time decreases whereas at higher current exhalation time increases. This thumb rules induced from reading the diagram fails at high frequencies where A has not the meaning anymore.

4.7.6 Minute Ventilation (Phase 1)

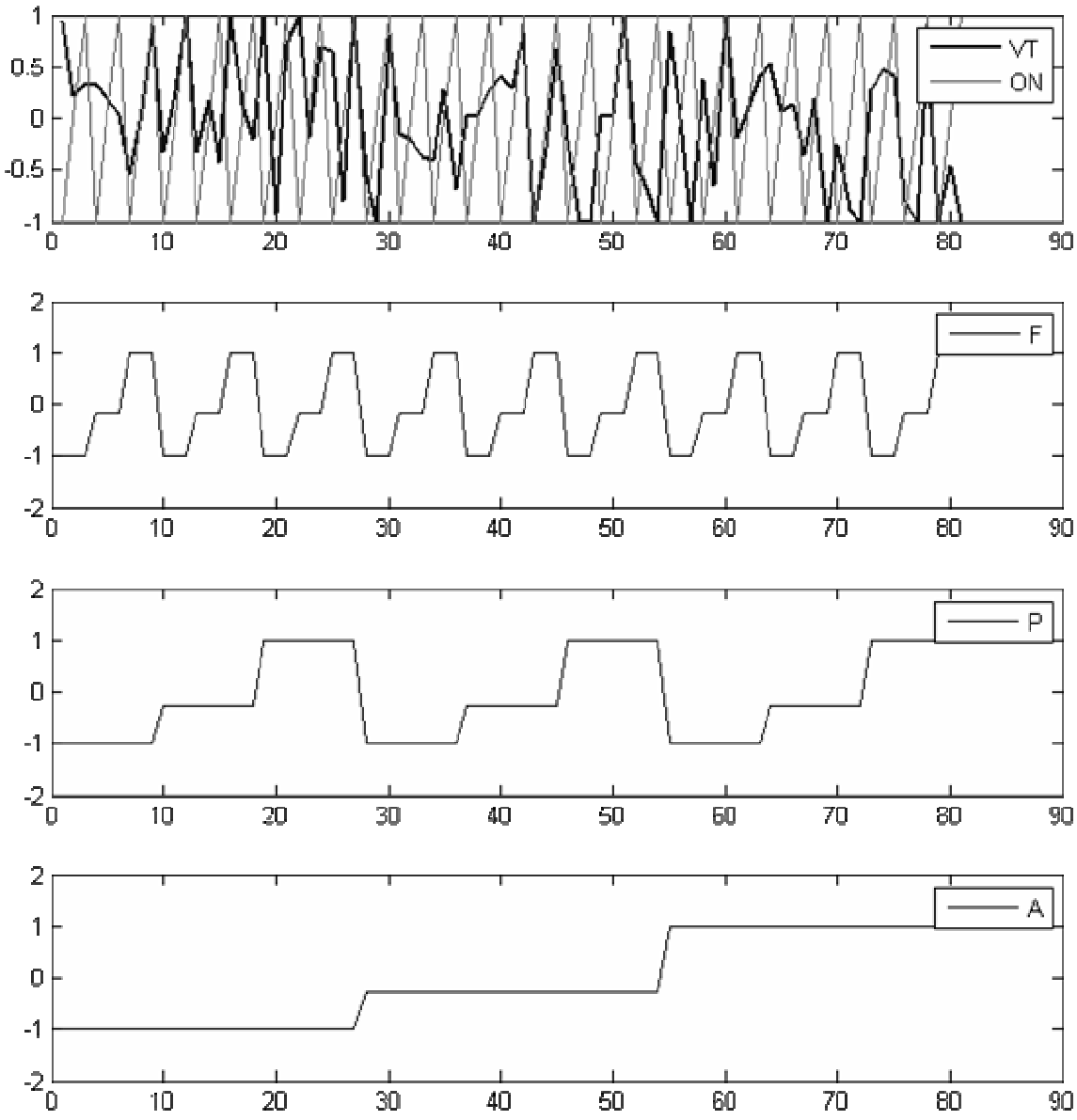


Image 78: The minute ventilation is a rough indicator of ventilation performance and has the greatest correlation with stimulation time for low currents.

The minute ventilation is not well understood in terms of dependencies on single parameters. Only for very low currents there seems to be a positive correlation between the ventilation performance an the stimulation time or the stimulation/pause-ratio.

4.7.7 Heart Rate Changes (Phase 1)

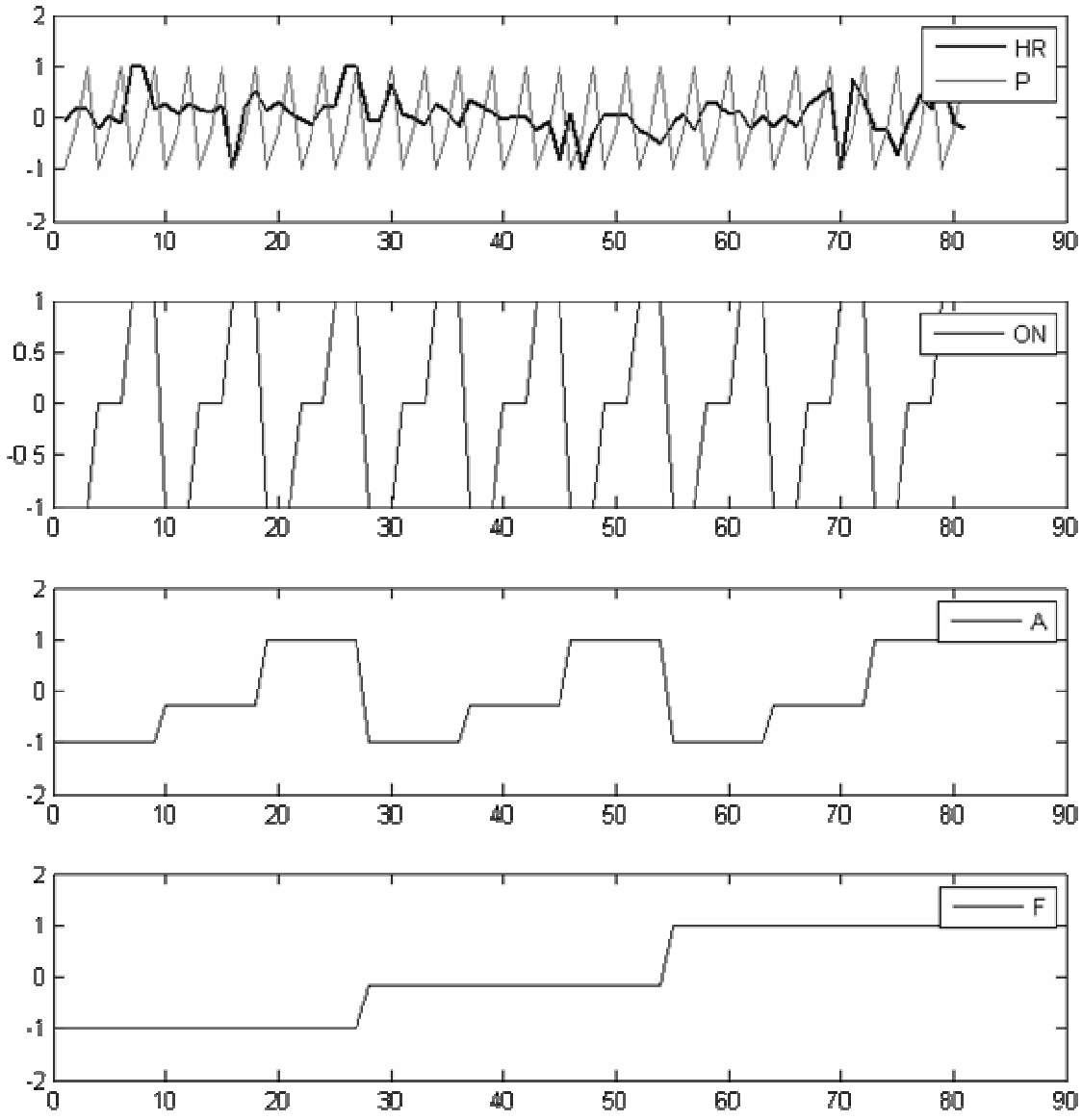


Image 79: Heart rate has been sorted first by F, A, ON and then by P. Heart rate seems to be rather independent of the stimulation parameters.

Heart rate seems to have rather constant behavior. Only at few places singular anomalies occur. It seems to be true for most of the cases where ON has a high value. Strong currents seem to support anomalies. But the direction of the feature (increase/decrease) seems to correlate with the pulse width. Stimulations with high pulse width cause an increase in heart rate and short pulse width cause the heart rate to decrease.

4.8 Results of the TherapyFinder

The original assumption which led to the investigation of the data was that changes to ventilation rate and minimal changes to heart rate offer the best therapy due to some possible neuroprotective mechanisms [12]. The TherapyFinder module was set to look for the right parameter vectors. The following table shows the result. Only results are shown that have a score better than 36 points.

Table 54: TherapyFinder's table of optimal parameters for ventilation decrease and constant heart rate

ON(s)	OFF(s)	P(us)	A(mA)	F(Hz)	Score(Points)
14	22	200	1	30	**40**
14	22	200	0,1	30	**39**
14	22	200	0,4	30	**39**
14	22	200	0,7	30	**39**
14	22	200	1	24	**39**
14	22	200	1	27	**39**
14	22	200	1	27	**38**
14	22	200	1	30	**38**
14	18	200	1	30	**38**
14	22	200	0,1	24	**38**
14	22	200	0,4	24	**38**
14	22	200	0,7	24	**38**
14	22	200	1	21	**38**
14	26	200	1	21	**38**
14	26	200	1	24	**38**
14	26	200	1	27	**38**
14	26	200	1	30	**38**
14	22	200	1	30	**38**
10	22	200	0,1	24	**37**
10	22	200	0,1	30	**37**
10	22	200	0,4	24	**37**
10	22	200	0,4	30	**37**
10	22	200	0,7	24	**37**
10	22	200	0,7	30	**37**
10	22	200	1	24	**37**
10	22	200	1	30	**37**
14	18	200	1	30	**37**
14	22	200	0,1	30	**37**
14	22	200	0,4	30	**37**
14	22	200	0,7	30	**37**
14	22	200	1	24	**37**
14	26	200	1	27	**37**
14	18	200	0,1	30	**37**
14	18	200	0,4	30	**37**
14	18	200	0,7	30	**37**

The first entries in the table represent the parameters that seem to best match the required biological response of a statistical rat population – thus are considered optimal for treatment.

4.9 Diagrams generated from the Artificial Neural Network

The following selected diagrams (4 from 147) depict the influence of two of the seven VNS parameters for a specific cardio-respiratory effect. The information from the other five parameters was averaged. Those diagrams are difficult to comapare to those presented in chapter 4.2 because of three reasons. The first reason is that the networks learned the absolute effects during a VNS cycle whereas the diagrams in 4.2 only show alterations from normal. Training with relative values did not succeed. The second reason is that the diagrams in 4.2 are generated from a set of dimensions including the timers t1 and t2. Therefore the artificial network is capable of reproducing a regular VNS cycle. This implies a strong smoothing mechanism over the other five dimensions. The third reason is that the T_{ON} and T_{OFF} times are always shown separately. This entails a much larger number of diagrams. All the three reason together result in a fundamental difficulty to understand the complexity of the inherent function. Also note that the diagrams presented below include values from all phases. That means that the performed network analysis shows average increases in the feature base, not the feature alteration during stimulation. Because the project aims at long term effect that might induce neuroprotective benefits such investigations of the feature base change seem to be desirable.

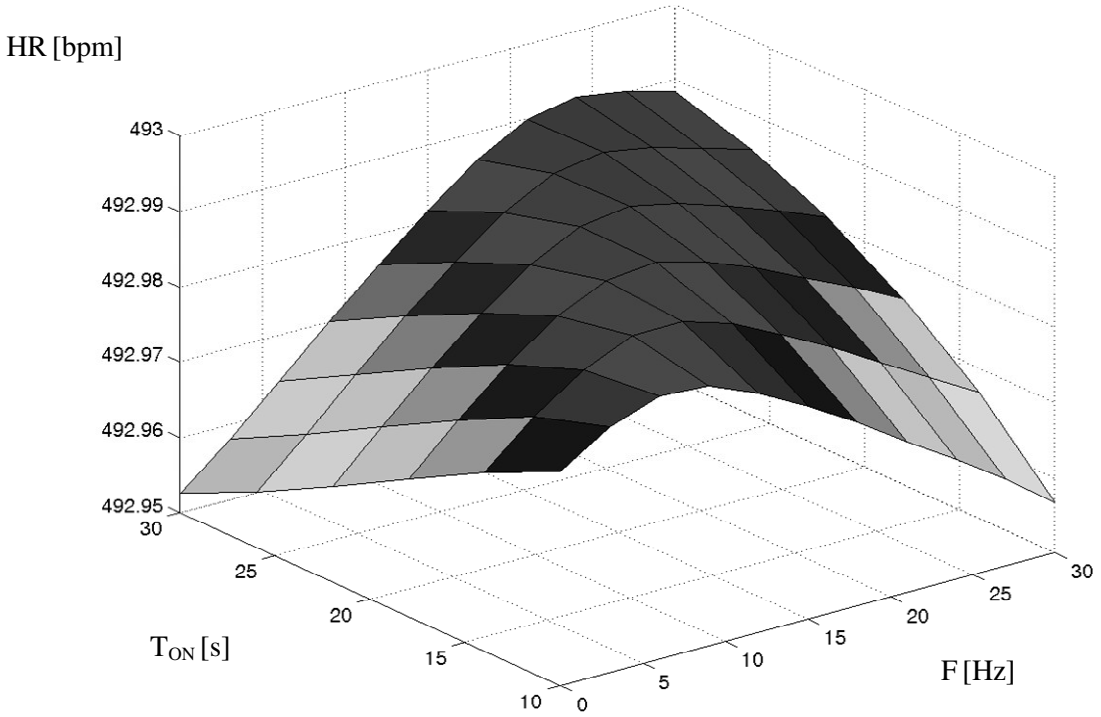


Image 80: VNS period duration's and VNS frequency's influences on heart rate.

The first diagram illustrates the influence of F and T_{ON} on the heart rate. A tendency to a decrease in heart can be obtained with a high value of T_{ON} and a small value of F or with a small value in Ton and a high value of F. A tendency for heart rate to increase can be observed when the ratio between F and Ton is roughly 1.

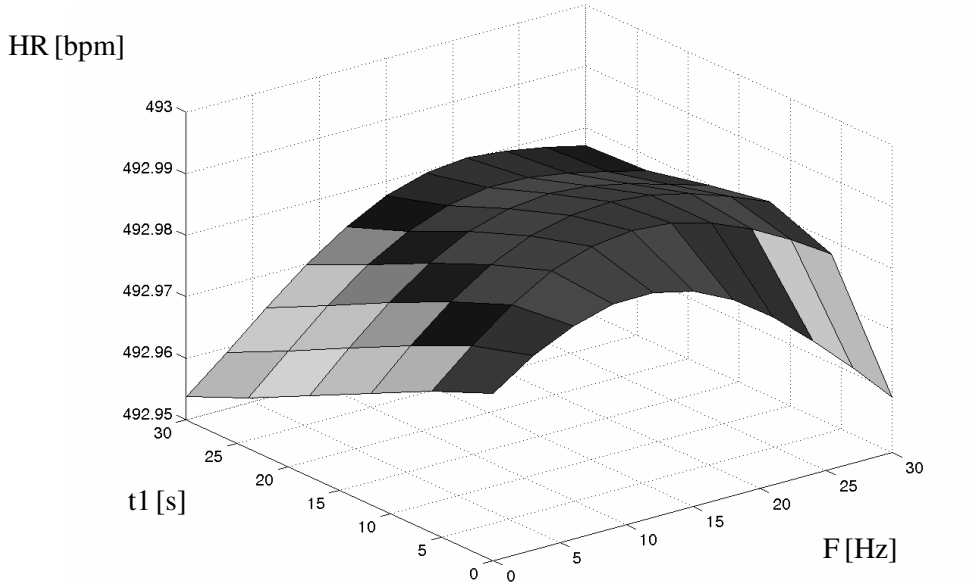


Image 81: dynamic evolution of heart rate during VNS period in function of VNS frequency

Effects on heart rate occur in the first 15s of stimulations. If frequency is higher than 10 and lower than 20 Hz, then there is a tendency for the heart rate to increase. If frequency is higher then 20Hz then there is a tendency to a decrease in heart rate.

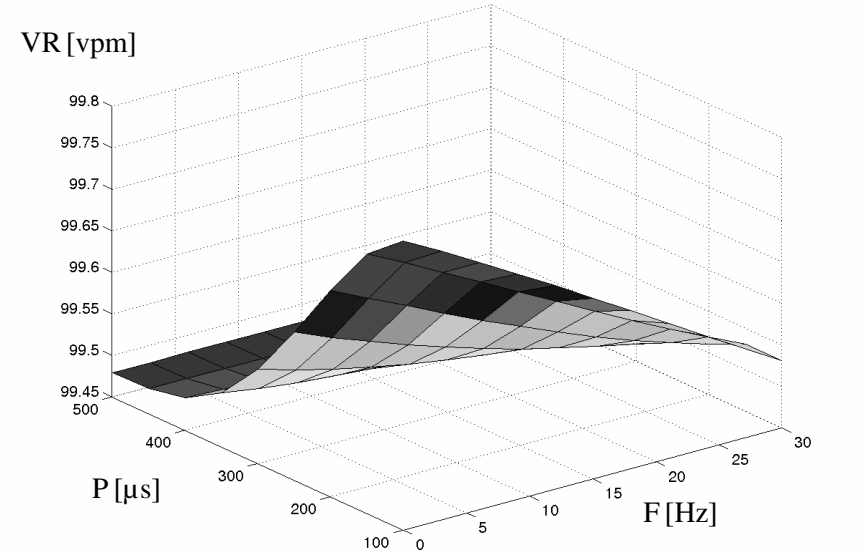


Image 82: ventilation rate as function of P and F

A tendency to an increase in ventilation rate is observed only for small values of F and P.

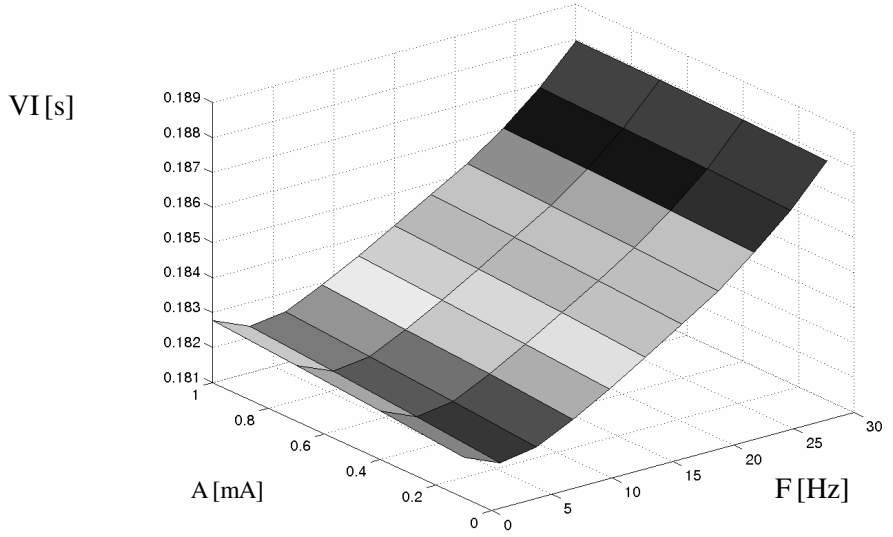


Image 83: Inspiratory time as function of A and F.

The diagram can be approximated by a plane. Inspiratory time tend to increase with high values of A and F. The slope of the tendency to an increase in inspiratory time is bigger as function of F rather than A: F has a greater impact on inspiratory time than A.

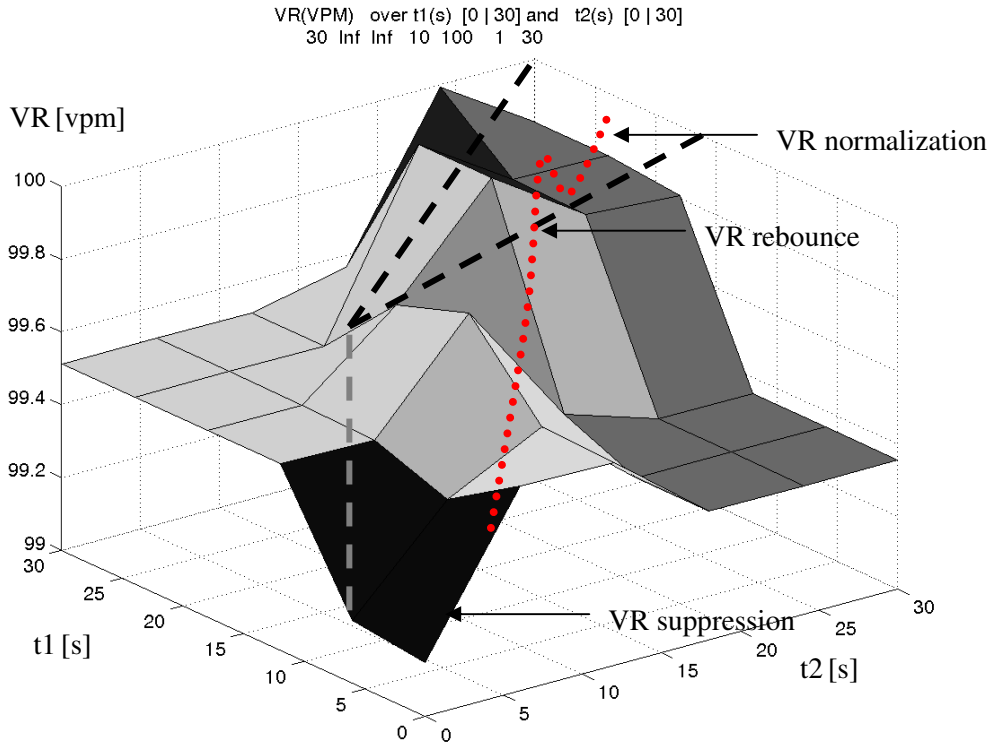


Image 84: The artificial neural network's understanding of the concept of time

In image 84 the network's understanding of the concept of time is presented. The parameters T_{ON} and T_{OFF} define a path through that understanding (red dotted line).

4.10 Optimal VNS Parameters for a requested Cardio-Respiratory Effects

This chapter has been written cooperatively
with Ph.D. Boubker Zaaimi.

Here more different therapy hypotheses are compared for the parameters playing role in them and attainability – that's how probably they can be achieved by the best suggested parameters using the TherapyFinder.

Table 55: investigated cases

	Target Effects	
Case label	**Ventilation rate**	**Heart rate**
A	increase	-
B	decrease	-
C	-	increase
D	-	decrease
E	increase	decrease
F	Increase	constant as possible
G	Increase	Increase
H	decrease	decrease
I	decrease	constant as possible
J	decrease	Increase

Table 56: 3 statistically best VNS parameters settings from the interrogation module inducing requested cardio-respiratory effects

Investigated cardiac and/or respiratory effects	T_{ON}	T_{OFF}	P	A	F	Confidence of occurrence (%)
A : increase in ventilation rate	30	30	100	0.1	3	74,07
	30	30	100	0.4	3	73,70
	30	30	100	0.7	3	73,33
B : decrease in ventilation rate	10	14	200	1	30	71,89
	10	14	200	0.7	30	71,67
	10	14	200	0.4	30	71,41
C increase in heart rate	18	10	500	0.1	15	76,30
	22	10	500	0.1	18	76,00
	18	10	500	0.4	15	75,93
D : decrease in ventilation rate	10	30	100	1	3	59,22
	10	30	100	1	30	58,93
	10	30	100	0.7	3	58,70
E: increase in ventilation rate and decrease in heart rate	30	30	100	0.1	3	62,11
	30	30	100	0.4	3	61,98
	30	30	100	0.7	3	61,87
F : increase in	30	30	100	0.1	6	47,44

ventilation rate and no change in heart rate	30	30	100	0.4	6	47,22
	30	30	100	0.7	6	47,00
G : increase in ventilation rate and increase in heart rate	18	10	500	0.1	15	38,15
	22	10	500	0.1	18	38,00
	18	10	500	0.4	15	37,96
H : decrease in ventilation rate and decrease in heart rate	10	30	100	1	30	46,50
	10	30	100	0.7	30	46,22
	10	30	100	0.4	30	45,93
I: decrease in ventilation rate and no change in heart rate	10	18	200	0.7	27	63,63
	10	18	200	0.4	27	63,20
	10	18	200	0.7	30	63,04
J : decrease in ventilation rate and increase in heart rate	10	30	100	1	30	46,50
	10	30	100	0.7	30	46,22
	10	30	100	0.4	30	45,93

For every requested cardiac or respiratory effect of VNS, a PIF has been calculated from tables with 40 records. The PIF shows how the five parameters influence the measured feature base across a stimulation cycle.

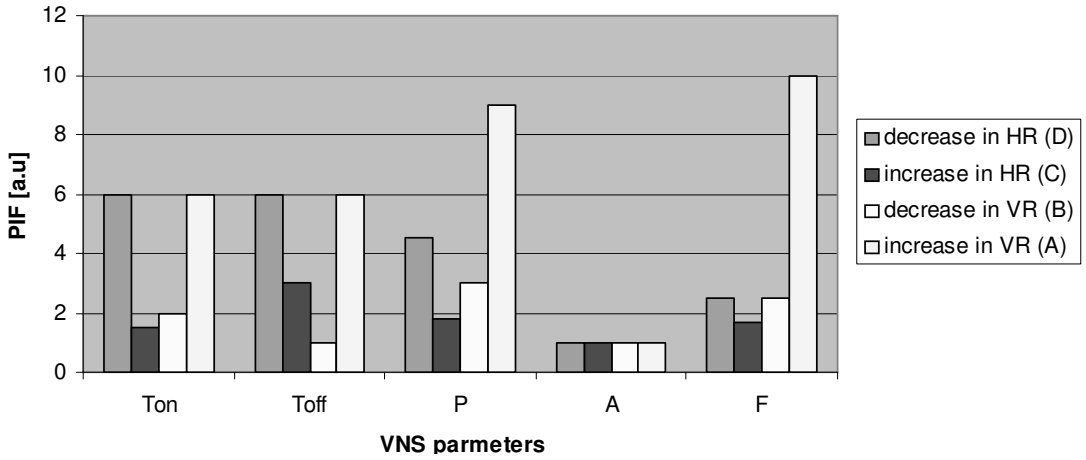


Image 85: bar chart showing statistical importance of VNS parameters (PIF) in inducing one requested cardiac or respiratory effect.

1) an increase in heart rate
2) a decrease in heart rate
3) an increase in ventilation rate
4) a decrease in ventilation rate

Figure 84 shows four scenarios for which the PIF has been obtained. In the first scenario parameters were optimized for a maximum decrease in heart rate. Those are stimulations where alterations in either phase 1 or phase 2 dominate with lower values. For this case the PIF value from the diagram tells that the precise setting of T_{ON} and T_{OFF} is most important for success. In the second scenario the parameters were optimized for a maximum increase in heart rate. In

that case the diagram suggests that the proper choice of T_{OFF} has the biggest impact on increasing heart rate. In the third scenario the parameters were optimized for a decrease in ventilation rate. The parameters P and F dominate this effect. The fourth scenario parameters were optimized for an increase in ventilation rate. In such scenario the precise choice of P and F is very important. It can be observed, that current strength is not strongly associated with a specific response.

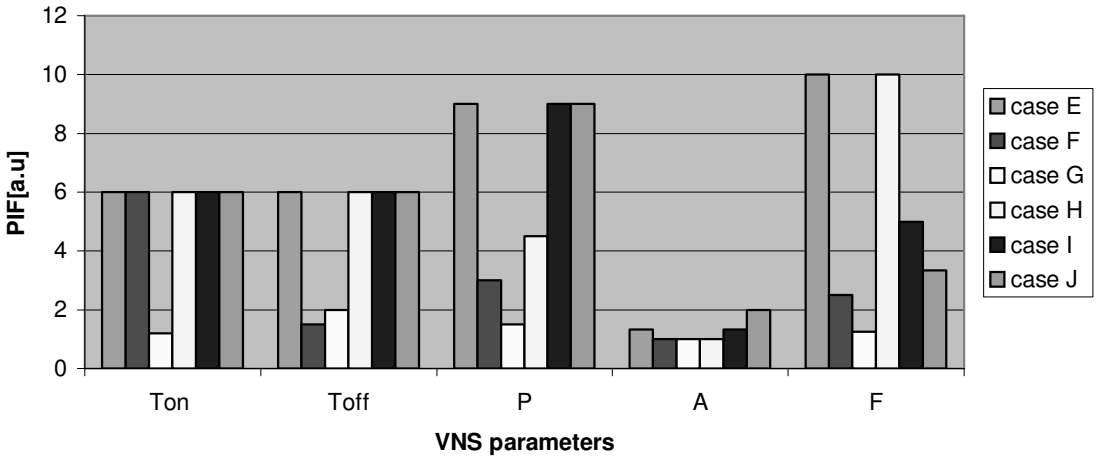


Image 86: Bar chart showing PIF for two requested changes (in heart and respiratory rate). PIF's indicate statistical importance of VNS parameters (PIF) in inducing a requested cardiac or respiratory effect.

Figure 85 shows more complex situations were two concomitant physiological responses are requested (Table 85). In the first scenario the parameters were optimized for an increase in ventilation rate and decrease in heart rate. According to the figure, such behavior is heavily depending on proper values of P and F but T_{ON} and T_{OFF} also have significant influence on the outcome. In the second scenario the parameters were optimized for an increase in ventilation rate and no change in heart rate. in this case Ton is dominating success. The second scenario is easier to obtain than the fist one because only one parameter dominates. In the third scenario we optimized the parameters for an increase in ventilation rate and an increase in heart rate. The VNS parameters seem to have the same importance. The average low value suggests that this is the most common case. In the fourth scenario the parameters were optimized for a maximum decrease in ventilation rate and heart rate. F dominates and Ton and Toff play also an important role to induce that case. The fifth scenario investigates VNS parameters optimized for a decrease in ventilation rate and no change in heart rate. P dominates, ton and Toff play also a role for the requested effect. In the sixth scenario the VNS parameters were optimized for low ventilation rate and high heart rate. This scenario is dominated by the proper choice of P, T_{ON} and T_{OFF}.

4.11 Comparison of Attainability of the 10 Scenarios

This chapter has been written cooperatively
with Ph.D. Boubker Zaaimi.

From the highest confidence score of each scenario figure 87 can be created. This figure is a collection of score values from the optimal parameter vectors set in proportion to the maximally obtainable scores. We can observe that some scenarios have more probability to occur. The scenario with one request is more probable to obtain than a scenario with concomitant requests. Remarkable detail is the position of scenario D on the lower figure because it seems to be more difficult to obtain than cases E and I.

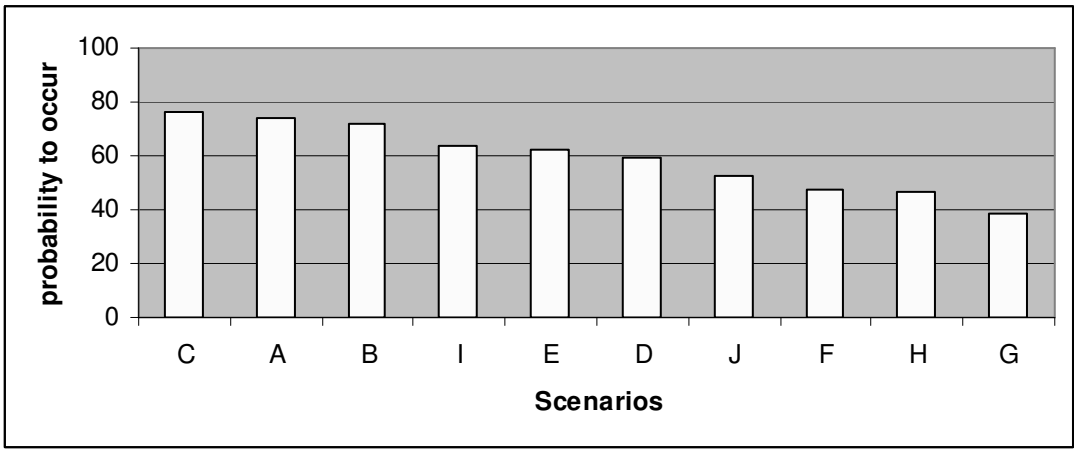


Image 87: The attainability chart shows the probability of success for each of the cases when used the suggested optimal parameters.

4.12 PIF diagrams and attainability scenarios for K-NN made predictions

This chapter is uncommented.

Phase 1	On	Off	P	A	F	Attainability %
CASE A	1,5	1	1,83	1,75	1,2	79,77%
CASE B	1,2	1	1,22	1,17	1,2	83,92%
CASE C	1,5	1,2	1,57	1,4	3	75,91%
CASE D	2	2	2,2	2,33	2	70,74%
CASE E	1,2	1	2,2	3,5	2	72,50%
CASE F	1,2	1,2	2,2	1,75	1,5	78,19%
CASE G	1,5	1,5	3,67	7	3	53,70%
CASE H	3	3	2,75	2,33	3	60,88%
CASE I	1,2	1	3,67	1,75	2	73,29%
CASE J	1,2	1	1,83	7	1,5	65,19%

5 Discussion

5.1 Reflection

The parameter vector decides how a VN-Stimulator produces its electrical pulses. A small rat population was used to record its physical response to some 81 typically used parameter vectors. From this data there were numerous features extracted: ventilation rate, ventilation amplitude, ventilation slope, inhalation time, exhalation time, minute ventilation and heart rate. For each of the features a separate artificial network was designed by means of a genetic algorithm. Although there is no proof that genetically optimized artificial networks fit the structure of the data optimally. Therefore other methods have been pursued in this thesis in order to give verification to the method. After training the networks some statistical representation of the relationships between the inputs and the outputs was obtained. These relationships are not human readable. Therefore this knowledge had to be obtained with the help of a program that extracts working point dependent transfer functions. Every transfer function depends on 7 input dimensions and cannot be displayed as a graph except for mean averaged ones. Thus many thousands of such transfer functions resulting from this transformation were compacted into many less functions by averaging over the five non-visible dimensions. In the end of this process there are numerous planes, one for each pair-wise combination of the 7 input dimensions that show how a specific feature depends on the dimensions of the plane. After specifying the looked for features like i.e. 'constant heart rate' or 'increased ventilation slope' etc. the planes are segmented accordingly. Finally, the full result space is restored from the segmented planes resulting in a specific score for parameter vectors that were not investigated experimentally. This way it can be explained why the methods predicts optimum vectors that lay outside the list in appendix, subchapter 7.1.

Other methods were used, like K-NN and interpolation. They are not quite easily comparable to the artificial neural networks method because they are applied to a much smaller data base called the “phase compacted data”. The predictions made by these methods have different meaning as they predict features’ extremes during stimulation (Phase 1) and after stimulation (Phase2). Phase 3 is a transitional phase that should be used to check for timer shifts in the data. If data of Phase 3 has higher values than those in Phase 1, then data in Phase 1 is replaced by data from Phase 3. The artificial neural networks are predicting overall base shifts in the presented form of interrogation with TherapyFinder.

5.2 Methods and Data

The fidelity of any of the presented methods is difficult to estimate. There are always Pro’s and Con’s to them:

5.2.1 Full UniRat data

The data consists of roughly 5000 stimulations and 1.2 million samples. Timers are included into the data to make the training of the data less ambivalent and help the potential method candidate to use it for creation of a stimulation cycle concept.

5.2.2 Phased UniRat data:

Data consists of 5000 singular feature values belonging to a phase of a stimulation cycle.

5.2.3 Compacted UniRat Phase data:

Phased data is mean averaged across the same parameter vectors in order to obtain 81 records that are suitable to use with more classical ways.

5.2.4 Genetically Designed Artificial Neural Networks

1. The relationship of the feature strength and the values between each dimension is truly statistical and can be very different for specific choice of the working point among the other five dimensions. That means that even though the vector was identified to be optimal there must be a step where the pattern is actually generated and checked for validity. This could be done with the compacted phase data and K-NN.

2. The planes are generated by forming mean averages over the other dimensions. The method could be improved if the grading would be weighted for instance with the PIF factors.

3. Currently, dimensions of time t1 and t2 handled just like any other dimension which is not quite correct, because any stimulation cycle contains many relations to t1 and t2. A new graphical interrogation module could be designed to specify timely behavior as an envelope that would be taken into account during segmentation. This would save the additional check as suggested under point 1.

4. The classification of probable decreases and increases is built on the assumption that they corresponded to low or high values in the plane. This assumption hasn't been proven, yet. Therefore it is strongly recommends to add methods as described under point 1 and 3.

5. The data is based upon 9 relatively uniform rats. Eventually it might improve the base adding up more rats and rats of other races.

6. Ventilation rate and amplitude seem to be depending on each other. This suggests that direct amplitude measurement is not sufficient. An improvement would be if VA was expressed as ventilation amplitude to ventilation rate ratio or if VA was normalized at the DAQ level or post processing level.

7. The exact specification of what features are to be expected in order to obtain the best therapy is not known. Therefore the results table must be enjoyed with caution. Depending on what input is done in the interrogation module different tables will be generated.

5.2.5 Decision Trees

Decision Trees are easily generated with tools like InfmationMiner or KNIME. It is difficult to extract more generic knowledge, no matter whether forward (Parameter-Stimulus) or back-

ward (Stimulus -Parameter) directed. The data sorting method has proven more informative - it belongs to the class of visualization methods.

5.2.6 Effect Prediction with LVQ

With LVQ networks it is possible to use systematic search with the help of table 58, 59 and 60. It is easy to compare the retrieved strings with the previously specified hypothesis. This method is fast and generates long lists of various lengths that are considered useful in order to obtain the specified goal but does not discern better from worse ones. Because of this fact it is not used for optimum parameter vector search but should be used as a fail back solution in cases where other attempts have failed and discover other parameter vector that roughly fit the scheme. Testing every vector of the list is still more systematic than searching with no guide.

5.2.7 Effect Prediction with K-NN

The K-NN method is very similar to LVQ method but it interpolates the effect strength between 3 (number is arbitrary) nearest points. It generates a full list representing the whole investigated parameter space. This full list is used in a second step with K-NN again but this time it returns a limited number of records with a specific order to try. This method is extremely quick and can be used from any command line thus not requiring any GUI components. If for some reasons a prediction should be made on a VN-Stimulator this method seems to be predestined on computing systems with very scarce resources.

5.2.8 Effect Prediction by Interpolation and Inference

The interpolation is much faster than generating transfer functions from artificial networks but requires the same principle of inference for result space generation. This result space is being graded by compliancy with the hypothesis and returned. Unfortunately, the TherapyFinder has not been adapted to the decreased dimensionality of the result space and therefore no optimal results table can be found in this work. Nevertheless the byproduct of this process which are the VNS effect map diagrams in chapter 4.2 , give relatively detailed overview over the feature alteration in each of the phases and could be used for guessing a ‘walking’ direction when a parameter vector - that has been working previously - stops to work. They are much more detailed than the tendency diagrams or maximum effect diagrams generated in conjunction with the artificial neural network.

5.2.9 Effect Prediction with Radial Basis Function Networks

RBF networks can be used to learn from compacted phase data and does not require the same type of inference mechanisms as interpolation. The performance is well suited by a standard office PC. They can be used to produce a result space directly by visiting all result space points. For vectors close to original supporting vectors the RBF offer similar prediction quality as K-NN. Further vectors become unrealistic because it is difficult to set a proper spread factor. Therefore K-NN should be used favorably before RBF networks.

5.3 Conclusion

The artificial neural networks have failed to convince as an effective tool to model large data streams with a significant amount of ambiguity. The use of Matlab as framework was not the obstacle. Large amounts of data have led to RAM expansion on the laboratory machines. The data had to be trained in portions where a forgetting through over-learning effect was observed which is common to online and hybrid (online/offline) training. Full batch training was technically not possible – only in genetic design process. In summary it can be said that the use of artificial neural networks did not provide the benefit of not needing to overly simplify or pre-process the data. And even on compacted phase data they did an average job, which was not mentioned, yet. Of course they did much better than with full data, but the results seemed to be bumpier in their response than the other methods. The poor performance of the networks which can be explained with too few unique parameter vectors was topped by the necessary amount of work to implement a management and control software in Matlab. Every step required manipulation of large data portions with consequence for data logistic and increased level of mistakes that had to be corrected later.

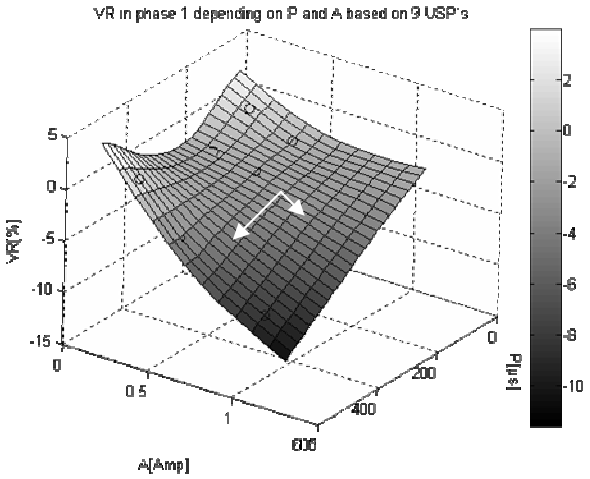
In compare more classical approaches were using a pre-processed data that was not even trying to represent a full cycle. Of course this pre-processing required human intervention but could be done with a minimal level of analysis and a-priori knowledge of VNS. This data has shown more robustness in the sense of less inconsistencies and the interpretation was easier. Classical methods of Data Mining are fairly fast, sometimes one second fast and are easy to use where the whole ANN generation process has taken sometimes weeks. At the same time there is more reason to believe a method that hardly modifies the data and sticks close to it like K-NN than a neural network where not enough data (in this case unique vectors) exist to verify its correctness.

Nevertheless the project has not been a failure. The results of TherapyFinder – no matter what they looked like – are matter to clinical testing and seem to be consistent with practically obtained guide-lines used in praxis [53-55]. That's maybe the overall changes of ventilation performance has greater impact on therapy success than implications of the stimulation on effects during a single stimulation cycle. If effects on a cycle are more important, then K-NN tables can be used which are not quite consistent with the practical experience.

Also knowledge was discovered how to progress with therapy, when stimulation does not show the prospected benefits anymore. By combining PIF values with the diagrams a direction

PIF CASE I	ON:1,2	OFF:1	P:3,67	A:1,75	F:2	73,29%

for the change of the parameters can be estimated. In Case I this means if P was 250µs and A was 0.5mA then both should be increased to roughly 500µs and 0.75mA in order to reflect the PIF ratio between the two parameters.



In future this knowledge could be applied to an autonomous stimulator system that would rationalize human interference. Such system should be equipped with a means to sense heart, ventilation and wake state changes. If placed into a patient it would start with optimal parameters as described in the results chapter for the case of two features to be controlled. The longer the parameters do not lead to the expected results the more the stimulator would softly translate to optimize only for one physical feature which typically has more “aggressive” parameter settings. If failure of satisfying conditions is sustained even longer more and more random elements should be added on top of the optimal vector. Eventually, the LVQ based list should be scanned. If no success can be achieved, then random search should be initiated. The random should be applied proportionally to the PIF values. Once the stimulator has arrived at the random side of the action spectrum medical personnel should be notified.

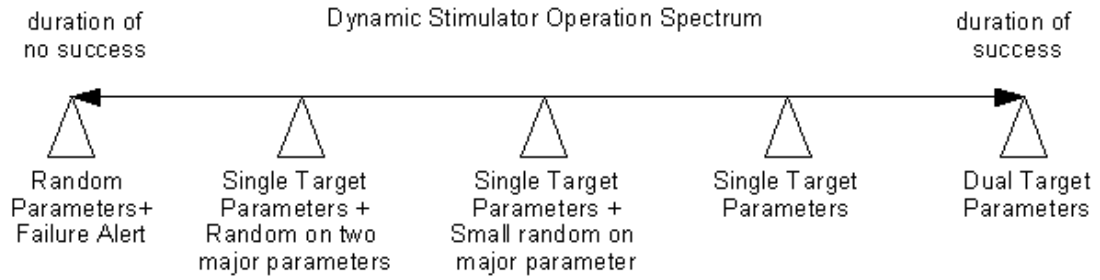


Image 88: Action-Spectrum for autonomous VN-Stimulator

6 Directories

6.1 Index

6.2 Images

6.3 Tables

6.4 References and Literature

[1] Wilder BJ, Uthman BM, Hammond EJ. *Vagal stimulation for control of complex partial seizures in medically refratory epileptic patients*. PACE 1991;14:108-15.

[2] Uthman BM, Wilder BJ, Hammond EJ, Reid SA. *Efficacy and safety of vagus nerve stimulation in patients with complex partial seizures*. Epilepsia 1990;31:S44-S50.

[3] Holder LK, Wernicke JF, Tarver WB. *Treatment of refractory partial seizures: preliminary results of a controlled study*. PACE 1992;15:1557-71.

[4] Landy HJ, Ramsay RE, Slater J, Casiano RR, Morgan R. *Vagus nerve stimulation for complex partial seizures: surgical technique, safety, and efficacy*. J Neurosurg 1993;79(4):636-7.

[5] The Vagus Nerve Stimulation Study Group; *A randomized controlled trial of chronic vagus nerve stimulation for treatment of medically intractable seizures*. Neurology 1995;45:224-30.

[6] Handforth A, DeGiorgio CM, Schachter SC et al. *Vagus nerve stimulation therapy for partial-onset seizures: 1. A controlled study of effect on seizures*. Neurology 1998;51:48-55.

[7] Ben-Menachem E, Manon-Espaillat R, Ristonovic R et al. *Vagus nerve stimulation for treatment of partial seizures: 1. A controlled study of effect on siezures*. Epilepsia 1994;35:616-26.

[8] George R. et al. *Vagus nerve stimulation for treatment of partial seizures: 3. Long-term follow-up on first 67 patients exiting a controlled study*. First International Vagus Nerv Stimulation Study Group. Epilepsia. 1994 May-Jun;35(3):637-43. [PMID: 8026410]

[9] DeGiorgio CM. et al. *Prospective long-term study of vagus nerve stimulation for the treatment of refractory seizures*. Epilepsia 2000 Sep;41(9):1195-200. [PMID: 10999559]

[10] Boubker Zaaimi. et al. *Vagus Nerve Stimulation Induces Concomitant Respiratory – Alterations and a Decrease in SaO2 in Children*. Epilepsia. 46(11):1–8. 2005 Blackwell Publishing. Inc. © 2005 International League Against Epilepsy

[11] Lodwich A. et. al., *Optimierung der Parameter für die Vagusnervstimulation mit KNN zur Behandlung refraktärer Epilepsie bei Kindern (OptiVaNeS)* pp. 210, Proceedings – 16. Workshop Computation Intelligence – Dortmund 29.Nov.–1.Dez.2006, R. Mikut, M. Reischl, Institut für Angewandte Informatik/Automatisierungstechnik, Universitätsverlag Karlsruhe, ISBN 3-86644-057-X

[12] Zaamini B.**, PhD-Thesis 05.07.2007**, Université de Picardie Jules Verne, Amiens

[13] Walter A. Brown, MD, *Brain Stimulation Methods for Treating Depression.* Applied Neurology, Februar 03, 2006

[14] Perner P., Trautzsch S. *Wissensakquisistion in der medizinischen Diagnose mittels Induktion von Entscheidungsbäumen*, (Zeitschrift) Künstliche Intelligenz, 3 (1997), p. 32-33

[15] Perner P. *Data Mining on Multimedia Data*, Springer Verlag (Berlin, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo) 1998, ISSN 0302-9743, ISBN 3-540-00317-7

[16] Perner P., Trautzsch S., *Multinterval Discretization for Decision Tree Learning*, In: Advances in Pattern Recognition, A. Amin, D. Dori, P. Pudil and H. Freeman (Eds.), LNCS 1451, Springer (Heildelberg), p. 475-482

[17] Rawling JO., Pantula SG., Dickey DA., *Applied Regression Analysis – A Research Tool*, 2nd Edition, Springer Verlag (Heidelberg, 1998)

[18] Atkinson A., Riani M. *Robust Diagnostic Regression Analysis*, Springer Verlag (Heidelberg, 2000)

[19] Herrero J., Valencia A., Dopazo J.: *A hierarchical unsupervised growing neural network for clustering gene expression patterns*.

[20] Zhang C. Zhang S. *Association Rule Mining*, LNAI 2307, Springer Verlag (Heidelberg 2002)

[21] Adamo J-M. *Data Mining for Association Rules and Sequential Patterns*, Springer Verlag (Heidelberg 2001)

[22] Hipp J. Güntzer U., Nakhaeizadeh G., *Data Mining of Association Rules and the Process of Knowledge Discovery in Databases*, In: P. Perner (Ed.) Advances in Data Mining, Applications in E-Commerce, Medicine and Knowledge Management, Springer Verlag 2002, LNAI 2394, p. 39-57

[23] Ebert, Favre, JM, Peikert R. (Eds.), *Data Visualization 2001*, Springer Verlag (Heidelberg 2001)

[24] Papageorgieou M., *Optimierung - Statische, dynamische, stochastische Verfahren für die Anwendung*, 2. Auflage, R. Oldenbourg Verlag (München, Wien 2006), ISBN 3-486-23775-6

[25] Fletcher R., *Practical Methods of Optimization*, 2nd Edition, John Wiley & Sons (Chichester 1987)

[26] Gill P. E., Murray W., Wright M. H., *Practical Optimization*, Academic Press (New York 1981)

[27] Powell M.J.D., Lenstra J.K., Ronnooy Kan A.H.G., Schrijver A., *A view of nonlinear optimization. In History of Mathematical Programming*, North-Holland (Amsterdam), p. 119-125

[28] Henry Gray (1821–1865). *Anatomy of the Human Body*. 1918.

[29] Paintal AS. *Vagal sensory receptors and their reflex effects*. Physiol Rev. 1973 Jan; 53(1):159-227. Review.

[30] Marlot D, Duron B. *Postnatal development of vagal control of breathing in the kitten*. J Physiol (Paris). 1979;75(8):891-900

[31] Cambridge Electronic Design - Spike2 http://www.ced.co.uk

[32] Christian Maas, *XVI32*, http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm

[33] Zell A., *Simulation neuronaler Netze*, 4. Aufl., Oldenbourg Wissenschaftsverlag GmbH (München 2003)

[34] Aurenhammer, F. and Klein, R. *Voronoi Diagrams*. Ch. 5 in *Handbook of Computational Geometry* (Ed. J.-R. Sack and J. Urrutia). Amsterdam, Netherlands: North-Holland, pp. 201-290, 2000.

[35] Kohonen T., *The Self-Organizinh and Associative Memory*, 3rd, Springer Series in Information Sciences, Springer Verlag (1989)

[36] Kohonen T., Kangas J., Laaksonen J., Torkkola K., LVQ-PAK – *The Learning Vector Quantizatzion Program Package*, Version 2.1, LVQ Programming Team of the Helsinki Univ. of Technology, Lab of Comp. and Inform. Science, Rakentajanaukio 2C, SF-02150 Espoo.

[37] Kerber R., ChiMerge: *Discretization of Numeric Attributes*, Learning: Inductive, AAAI 92, pp. 123-128, (1992)

[38] Russel S., Norvig P., *Artificial Intelligence – a modern approach*, Pearson Education Inc. (New Jersey 2003), ISBN 0-13-080302-2

[39] Lusti M*., Data Warehousing and Data Mining – Eine Einführung in entscheidungsstützende Systeme*, Springer Verlag (Berlin, Heidelberg, New York 2002), ISBN 3-540-42677-9

[40] Resampling Stats, *Help to XLMiner*, http://www.resample.com/xlminer/help/rtree/rtree_intro.htm

[41] Breiman, et al, *Classification and* Regression *Trees*, Chapman and Hall (1984), p.316

[42] Seidelmann G., *Using Heuristics to* Speed *Up Induction on Continuous Valued Attributes*, In: P.B. Brazdil (Ed.), Machine Learning: ECML-93, Springer Verlag (Berlin, Heidelberg 1993), p.390-395

[43] Fayyad U.M., Irani K.B.*, Multi-Interval Discretization of Continuous Valued Attributes for Classification Learning*, Machine Learning, 13th IJCAI, vol. 2, Chambery, France, Morgan Kaufmann, p.1022-1027, (1993)

[44] Perner P., *An Architecture for a CBR Image Segmentation System*, Journal on Engineering Application in Artificial Intelligence, Engineering Applications of Artificial Intelligence, vol 12(6), 1999, p.749-759

[45] Kohonen T., *Self Organizing Maps*, Springer Verlag 1995

[46] Wu C., Landgrebe D., Swain P., *The decision tree approach to classification*, School Elec. Eng., Purdue University, W. Lafayette, IN Rep. RE-EE 75-17, 1975

[47] Santner, T. J.; Williams, B. J.; Notz, W.I. *In Design and Analysis of Computer Experiments*; Springer-Verlag, 2003.

[48] E. Fix and J.L. Hodges. *Discriminatory analysis, non-parametric discrimination*. Technical report, USAF School of Aviation Medicine, Randolf Field, Tex. Project 21-49-004, Rept. 4, Contract AF41(128)-31, February 1951.

[49] Cuisenaire O., Macq B., *Fast k-NN Classification with an Optimal k-Distance Transformation Algorithm*, 1 Signal Processing Laboratory, EPFL, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland. Communications and Remote Sensing Laboratory, Universit_e catholique de Louvain, Place du Levant 2, B-1348 Louvain-la-Neuve, Belgium.

[50] Barber, C. B., D.P. Dobkin, and H.T. Huhdanpaa, *The Quickhull Algorithm for Convex Hulls*, ACM Transactions on Mathematical Software, Vol. 22, No. 4, Dec. 1996, p. 469-483. Available in HTML format at http://www.acm.org/pubs/citations/journals/toms/1996-22-4/p469-barber.

[51] National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center), University of Minnesota. 1993.

[52] Lodwich A., *Modellierung der Kraftentfaltung bei der Interaktion der Rumpfschale mit dem Positionierwerkzeug*, Fachhochschule Südwestfalen, Prof. Dr. Hühne

[53] Mu Q, Bohning DE, Nahas Z, Walker J, Anderson B, Johnson KA, Denslow S, Lomarev M, Moghadam P, Chae JH, George MS
Acute vagus nerve stimulation using different pulse widths produces varying brain effects.
Biol Psychiatry. 2004 Apr 15;55(8):816-25.
PMID: 15050863 [PubMed - indexed for MEDLINE]

[54] Woodbury DM, Woodbury JW.
Effects of vagal stimulation on experimentally induced seizures in rats.
Epilepsia. 1990;31 Suppl 2:S7-19.
PMID: 2226368 [PubMed - indexed for MEDLINE]

[55] Lomarev M, Denslow S, Nahas Z, Chae JH, George MS, Bohning DE.
Vagus nerve stimulation (VNS) synchronized BOLD fMRI suggests that VNS in depressed adults has frequency/dose dependent effects.
J Psychiatr Res. 2002 Jul-Aug;36(4):219-27.
PMID: 12191626 [PubMed - indexed for MEDLINE]

[56] Mathworks, *Help to surface fitting function gridfit()*,
http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8998&objectType=file#

6.5 Related Literature

[L1] Hartmut Siemes, Blaise F. D. Bourgeois: *Anfälle und Epilepsien bei Kindern und Jugendlichen.* Thieme, Stuttgart-New York 2001, ISBN 3-13-127031-4

[L2] Weiss, S. M., & Indurkhya, N. (1997). *Predictive data mining: A practical guide*. New York: Morgan-Kaufman.

[L3] Westphal, C., Blaxton, T. (1998). *Data mining solutions*. New York: Wiley.

[L4] Witten, I. H., & Frank, E. (2000). *Data mining*. New York: Morgan-Kaufmann.

[L5] Mark S. George, M.D. *New Methods of Brain Stimulation (ECT, MST, TMS, VNS and DBS) Are Improving OCD Research and Therapy*
http://www.musc.edu/fnrd/dbscns.htm, CNS Spectrum, 2000

[L6] Perner P., Feature Discretization, IBaI Report, 2000

6.6 Abbreviations

A	Stimulator Parameter: Current Strength
ANN	CI Method: Artificial Neural Network
BFGS	Optimization Method: Quasi Newton Gradient Descent developed by Broyden, Fletcher, Goldfarb and Shanno
BPM	Beats Per Minute
CI	Part Of Computer Science: Computational Intelligence
CUT	Data: Meta data defining where to delete erroneous data from UniRat
DAQ	Process: Data Acquisition
DT	Decision Tree
ECG	Diagnostic Method: Electrocardiography
F	Stimulator Parameter: Frequency
GA	CI Method: Genetic Algorithm
HD	Heart rate Deviation
HR	Investigated Feature: Heart Rate
Hz	Physical Unit: Hertz
LVQ	CI Method: ANN: Learning Vector Quantization
K-NN	CI Method: K Nearest Neighbors Algorithm
OFF	Stimulator Parameter: Stimulation Pause
ON	Stimulator Parameter: Stimulation Duration
P	Stimulator Parameter: Pulse Width
PAF	Data: Matrix with the three columns P, A and F
PIF	Parameter Importance Factor
PG	Plethysmography
RAM	Random Access Memory
RBF	CI Method: ANN: Radia Basis Function
RT	Regression Tree

s	Physical Unit: second
s	Math. Operator: Laplace Operator
SA	Sensitivity Analysis
SOM	CI Method: Self Organizing Maps
SOTA	CI Method: Self Organizing Tree Algorithm
t1	Data Channel: Timer reset at stimulation beginning
t2	Data Channel: Timer reset at stimulation ending
UniRat	Data: Universal Rat
VA	Investigated Feature: Ventilation Amplitude
VI	Investigated Feature: Inhalation Time
VNS	Therapy Method: Vagus Nerve Stimulation
VR	Investigated Feature: Ventilation Rate
VS	Investigated Feature: Ventilation Slope
VT	Investigated Feature: Minute Ventilation
VX	Investigated Feature: Exhalation Time
UA	Uncertainty Analysis
WP	Working Point
XVI32	Program Name – no acronym

7 Appendix

7.1 Recording Protocol (DAQ)

This protocol has been applied to any of the 9 (11 altogether) rats. Every rat was stimulated with roughly 7 stimulation periods for each of the stimulation parameter vectors listed.

Table 57: Recording Protocol

Nr.	A (mA)	P (µs)	f (Hz)	On (s)	Off (s)
1	0.25	500	15	7	30
2	1	250	15	7	30
3	0.5	130	15	7	30
4	1	130	15	7	30
5	1	500	30	7	30
6	1	250	30	7	30
7	0.5	500	30	7	30
8	0.5	130	5	7	30
9	0.25	130	15	7	30
10	0.5	250	5	7	30
11	1	500	5	7	30
12	1	130	30	7	30
13	1	500	15	7	30
14	0.5	500	5	7	30
15	0.25	130	30	7	30
16	0.5	250	30	7	30
17	1	130	5	7	30
18	0.25	500	30	7	30
19	0.25	250	15	7	30
20	0.25	250	5	7	30
21	0.25	130	5	7	30
22	0.5	130	30	7	30
23	0.5	500	15	7	30
24	1	250	5	7	30
25	0.5	250	15	7	30
26	0.25	250	30	7	30
27	0.25	500	5	7	30
28	0.25	500	15	14	20
29	1	250	15	14	20
30	0.5	130	15	14	20
31	1	130	15	14	20
32	1	500	30	14	20
33	1	250	30	14	20
34	0.5	500	30	14	20
35	0.5	130	5	14	20
36	0.25	130	15	14	20
37	0.5	250	5	14	20
38	1	500	5	14	20
39	1	130	30	14	20

Nr.	A (mA)	P (µs)	f (Hz)	On (s)	Off (s)
40	1	500	15	14	20
41	0.5	500	5	14	20
42	0.25	130	30	14	20
43	0.5	250	30	14	20
44	1	130	5	14	20
45	0.25	500	30	14	20
46	0.25	250	15	14	20
47	0.25	250	5	14	20
48	0.25	130	5	14	20
49	0.5	130	30	14	20
50	0.5	500	15	14	20
51	1	250	5	14	20
52	0.5	250	15	14	20
53	0.25	250	30	14	20
54	0.25	500	5	14	20
55	0.25	500	15	21	12
56	1	250	15	21	12
57	0.5	130	15	21	12
58	1	130	15	21	12
59	1	500	30	21	12
60	1	250	30	21	12
61	0.5	500	30	21	12
62	0.5	130	5	21	12
63	0.25	130	15	21	12
64	0.5	250	5	21	12
65	1	500	5	21	12
66	1	130	30	21	12
67	1	500	15	21	12
68	0.5	500	5	21	12
69	0.25	130	30	21	12
70	0.5	250	30	21	12
71	1	130	5	21	12
72	0.25	500	30	21	12
73	0.25	250	15	21	12
74	0.25	250	5	21	12
75	0.25	130	5	21	12
76	0.5	130	30	21	12
77	0.5	500	15	21	12
78	1	250	5	21	12
79	0.5	250	15	21	12
80	0.25	250	30	21	12
81	0.25	500	5	21	12

7.2 Record (Pre-) Classification

The classifications table were done with PreClassifyCompacted_SCRIPT.m and basically represent a. The tables subsum the VNS response of the tested rat population.

7.2.1 Limits

	VR	VA	VS	VI	VX	VT	HR
Phase 1	6.3%	4%	1.8%	0%	3.6%	2.8%	0.2%
Phase 2	4.2%	1%	1.4%	-	0%	1.5%	0.45%
Phase 3	3.25%	6.75%	4.4%	4.8%	3.3%	12.8%	0.7%

Comments: 0% means that ‚constant' states are not obtainable

- means that only negative states occurred in the data.

7.2.2 Phase 1

Table 58: Pre-classified compact data from Phase 1

Vector ID	VR	VA	VS	VI	VX	VT	HR
1	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	INCREASE	INCREASE
2	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
3	NO CHANGE	NO CHANGE	DECREASE	DECREASE	NO CHANGE	NO CHANGE	INCREASE
4	NO CHANGE	DECREASE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
5	DECREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE	DECREASE
6	NO CHANGE	INCREASE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	DECREASE
7	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	NO CHANGE	DECREASE
8	NO CHANGE	DECREASE	DECREASE	DECREASE	INCREASE	DECREASE	INCREASE
9	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
10	NO CHANGE	INCREASE	NO CHANGE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
11	NO CHANGE	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
12	NO CHANGE	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
13	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	DECREASE
14	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	INCREASE
15	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	DECREASE	NO CHANGE
16	NO CHANGE	DECREASE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE
17	NO CHANGE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
18	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	DECREASE
19	NO CHANGE	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
20	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
21	INCREASE	NO CHANGE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
22	NO CHANGE	DECREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	DECREASE
23	NO CHANGE	NO CHANGE	NO CHANGE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
24	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	INCREASE

25	NO CHANGE	NO CHANGE	DECREASE	DECREASE	NO CHANGE	NO CHANGE	INCREASE
26	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
27	NO CHANGE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	INCREASE	NO CHANGE
28	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
29	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
30	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
31	DECREASE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	NO CHANGE
32	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
33	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE
34	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE	INCREASE
35	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE	NO CHANGE
36	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
37	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
38	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
39	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
40	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE	NO CHANGE
41	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE
42	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
43	DECREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
44	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
45	DECREASE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	NO CHANGE
46	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
47	NO CHANGE	NO CHANGE	DECREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
48	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
49	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
50	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
51	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
52	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
53	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
54	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	DECREASE	NO CHANGE
55	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
56	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE	DECREASE
57	DECREASE	DECREASE	DECREASE	INCREASE	INCREASE	NO CHANGE	DECREASE
58	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	INCREASE	DECREASE
59	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE	DECREASE
60	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE	NO CHANGE
61	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE	INCREASE
62	NO CHANGE	INCREASE	INCREASE	INCREASE	INCREASE	NO CHANGE	DECREASE
63	INCREASE	NO CHANGE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
64	NO CHANGE	DECREASE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
65	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
66	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
67	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
68	NO CHANGE	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	INCREASE
69	DECREASE	INCREASE	INCREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
70	NO CHANGE	DECREASE	INCREASE	INCREASE	INCREASE	NO CHANGE	INCREASE
71	DECREASE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE

72	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	INCREASE	NO CHANGE
73	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
74	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	INCREASE	INCREASE
75	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
76	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	DECREASE	DECREASE
77	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	INCREASE	DECREASE
78	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
79	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
80	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
81	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE

7.2.3 Phase 2

Table 59: Pre-classified compact data from Phase 2

Vector ID	VR	VA	VS	VI	VX	VT	HR
1	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	INCREASE	INCREASE
2	INCREASE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	INCREASE
3	NO CHANGE	INCREASE	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE
4	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
5	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
6	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
7	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
8	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
9	NO CHANGE	INCREASE	INCREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE
10	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
11	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
12	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
13	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
14	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
15	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
16	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
17	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	INCREASE
18	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	INCREASE	NO CHANGE
19	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
20	NO CHANGE	NO CHANGE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
21	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	NO CHANGE
22	INCREASE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
23	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	INCREASE	NO CHANGE
24	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
25	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
26	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
27	NO CHANGE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	DECREASE
28	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
29	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
30	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE

31	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
32	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
33	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
34	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
35	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	DECREASE
36	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
37	NO CHANGE	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
38	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	INCREASE
39	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
40	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
41	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
42	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
43	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
44	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
45	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
46	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
47	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	NO CHANGE
48	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	INCREASE
49	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
50	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
51	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	INCREASE
52	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
53	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
54	NO CHANGE	NO CHANGE	INCREASE	DECREASE	INCREASE	DECREASE	NO CHANGE
55	NO CHANGE	NO CHANGE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
56	DECREASE	INCREASE	DECREASE	DECREASE	INCREASE	DECREASE	INCREASE
57	NO CHANGE	INCREASE	INCREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE
58	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
59	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	DECREASE	INCREASE
60	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
61	NO CHANGE	DECREASE	DECREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE
62	NO CHANGE	INCREASE	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE
63	NO CHANGE	INCREASE	DECREASE	DECREASE	DECREASE	DECREASE	NO CHANGE
64	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	DECREASE	NO CHANGE
65	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE
66	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
67	NO CHANGE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	INCREASE
68	NO CHANGE	INCREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	DECREASE
69	NO CHANGE	INCREASE	INCREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE
70	DECREASE	INCREASE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE
71	INCREASE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE	DECREASE
72	NO CHANGE	INCREASE	DECREASE	DECREASE	INCREASE	NO CHANGE	INCREASE
73	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	INCREASE
74	NO CHANGE	NO CHANGE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE
75	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
76	NO CHANGE	INCREASE	NO CHANGE	DECREASE	NO CHANGE	NO CHANGE	DECREASE
77	NO CHANGE	INCREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE	INCREASE

78	NO CHANGE	INCREASE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE
79	NO CHANGE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE	DECREASE
80	INCREASE	INCREASE	INCREASE	DECREASE	INCREASE	INCREASE	NO CHANGE
81	NO CHANGE	NO CHANGE	DECREASE	DECREASE	NO CHANGE	DECREASE	DECREASE

7.2.4 Phase 3

Table 60: Pre-classified compact data from Phase 3

Vector ID	VR	VA	VS	VI	VX	VT	HR
1	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	INCREASE
2	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE	INCREASE
3	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
4	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
5	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
6	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
7	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
8	NO CHANGE	INCREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
9	NO CHANGE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
10	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
11	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
12	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
13	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
14	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
15	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
16	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
17	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
18	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	DECREASE	NO CHANGE
19	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
20	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
21	NO CHANGE	DECREASE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	NO CHANGE
22	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
23	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
24	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
25	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	DECREASE
26	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
27	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	INCREASE	NO CHANGE
28	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
29	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
30	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
31	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
32	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
33	INCREASE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
34	INCREASE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
35	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
36	NO CHANGE	DECREASE	DECREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE

37	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
38	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
39	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	DECREASE
40	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
41	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
42	NO CHANGE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	NO CHANGE
43	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
44	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
45	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
46	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	INCREASE	NO CHANGE
47	NO CHANGE	DECREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
48	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	INCREASE
49	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
50	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	INCREASE
51	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
52	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
53	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
54	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
55	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
56	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	DECREASE
57	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE
58	INCREASE	DECREASE	NO CHANGE	INCREASE	DECREASE	INCREASE	INCREASE
59	INCREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE	DECREASE
60	INCREASE	DECREASE	NO CHANGE	INCREASE	DECREASE	INCREASE	NO CHANGE
61	NO CHANGE	DECREASE	DECREASE	DECREASE	DECREASE	DECREASE	NO CHANGE
62	NO CHANGE	INCREASE	INCREASE	INCREASE	INCREASE	NO CHANGE	DECREASE
63	NO CHANGE	INCREASE	NO CHANGE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
64	NO CHANGE	DECREASE	DECREASE	INCREASE	INCREASE	DECREASE	INCREASE
65	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	INCREASE
66	INCREASE	DECREASE	DECREASE	DECREASE	DECREASE	NO CHANGE	INCREASE
67	NO CHANGE	DECREASE	NO CHANGE	INCREASE	DECREASE	NO CHANGE	DECREASE
68	NO CHANGE	DECREASE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	INCREASE
69	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	INCREASE	INCREASE
70	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	INCREASE
71	NO CHANGE	INCREASE	INCREASE	INCREASE	DECREASE	NO CHANGE	NO CHANGE
72	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	DECREASE
73	NO CHANGE	DECREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	NO CHANGE
74	NO CHANGE	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
75	NO CHANGE	NO CHANGE	INCREASE	INCREASE	NO CHANGE	NO CHANGE	DECREASE
76	NO CHANGE	DECREASE	INCREASE	INCREASE	DECREASE	NO CHANGE	DECREASE
77	NO CHANGE	DECREASE	DECREASE	INCREASE	DECREASE	NO CHANGE	DECREASE
78	NO CHANGE	NO CHANGE	INCREASE	INCREASE	INCREASE	NO CHANGE	NO CHANGE
79	NO CHANGE	NO CHANGE	DECREASE	INCREASE	NO CHANGE	DECREASE	INCREASE
80	NO CHANGE	DECREASE	INCREASE	DECREASE	DECREASE	NO CHANGE	NO CHANGE
81	NO CHANGE	DECREASE	INCREASE	INCREASE	DECREASE	NO CHANGE	INCREASE

7.3 Scripts and functions for OptiVaNeS data

7.3.1 General description

The data, scripts and functions used for the OptiVaNeS project are stored in one directory called „Processing“. This top level folder contains the scripts required for the graphical control of the building process. In this folder there exist subfolders with the base name „Rat“ that contain the configured data and partially unique set of scripts and functions to process this data. „Processing“ also contains the final result of the building process: the file „universal.double“. There also exists a folder named „Transferfunctions“, „Tools“, „Phases“ and „UniRat“. The folder „Transferfunctions“ contains a set of diagrams from the trained network analysis. The „Tools“-folder contains a set of functions that are not directly used for the process of building a rat. The „UniRat“ folder contains diagrams that reproduce the protocol stimulations in order to validate the correct function of UniRat networks.

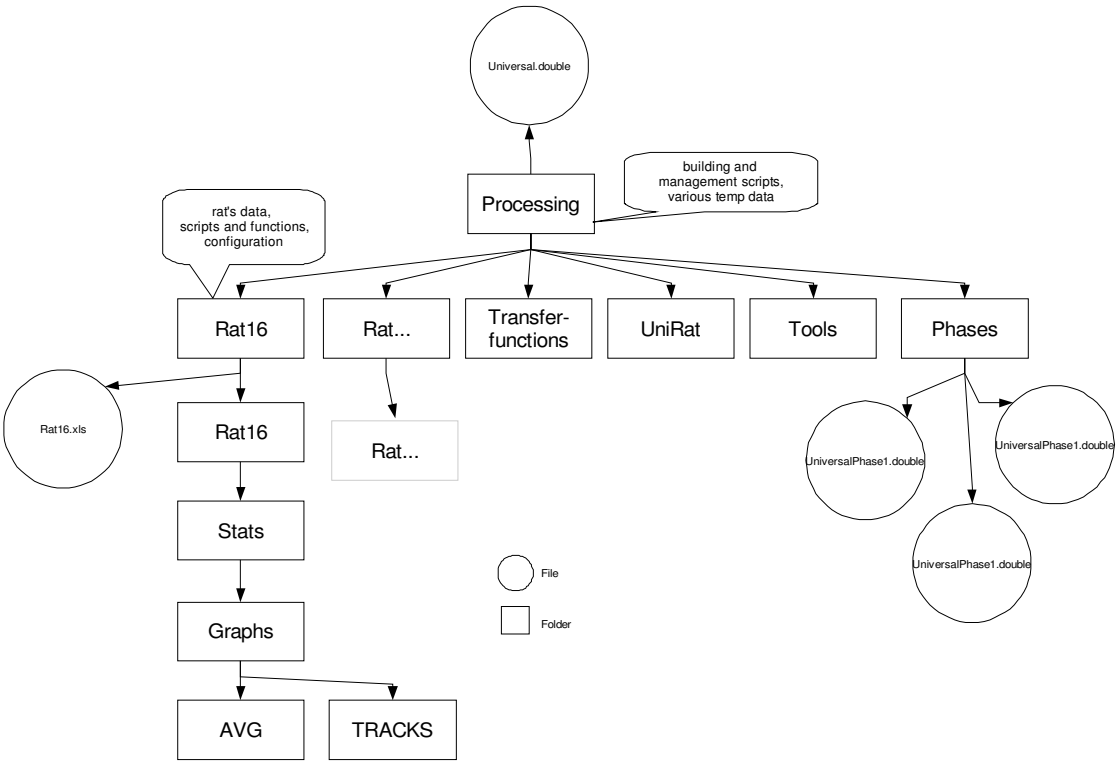


Image 89: Overview over the file and folder structure of the matlab scripts and data

Any „Rat“ folder has a subfolder with the same name that again has a subfolder „Stats“. In „Stats“ exist Excel tables with the data used for visualisation in the subfolder „Graphs“. This subfolder has two folder „AVG“ and „TRACKS“. „AVG“ contains the average VNS cycle and TRACKS has the individual curves shown as a swarm. Those diagrams are generated in order to ensure correct feature extraction and in order to better understand the results generated in consequence. Any „Rat“ folder has a set of scripts that process the data contained in the folder. Those scripts were adapted to the data and are therefore not easily exchangeable with other „Rat“ folders. The central function for invoking the building process has the name „Process()“. It checks a config file i.e. „Rat16.xls“ which is also specially adapted to the data. The result here is named i.e. „Rat16.double“. All files extended with .double indicate a binary file format

that contains 15x8 byte rows (15 binary double values). There are no headers in those files because reading is done until EOF.

The „Phases“ folder will be generated automatically if it does not exist. Call the SortToPhases function, when the regular build is finished.

7.3.2 Usage

The process of building is started with the BuildAll.m file. From the matlab command line it is required to change to the directory with this file. Then invoke it.

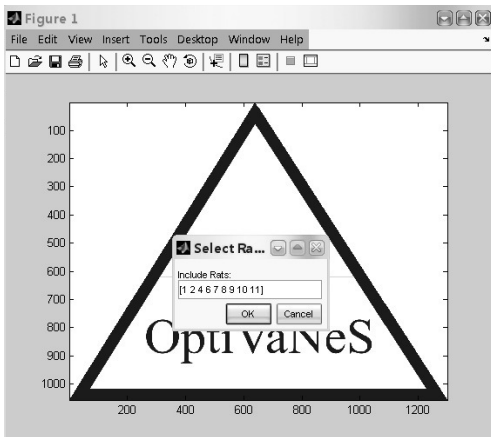


Image 90: Initial dialog asking which rats to include into the process

The first step in the process is a dialog box that contains a matlab matrix with the rats to be included in the further build process. There are rats 16 to 26 which are numbered 1 to 11. Please note that by default rats 3 and 5 are not included because they are considered faulty.

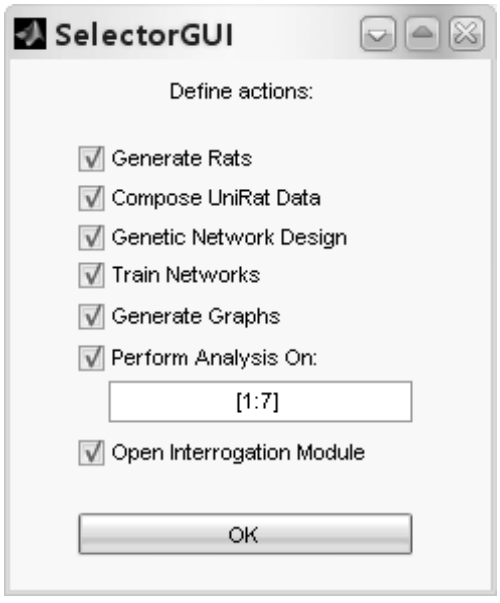


Image 91: Selector dialog where you can select the stages to run

The process stage selector dialog opens up in the next step. Each stage saves the results on disk. If some stages have been run before it is not required to rerun them every time again

when only later stages have been modified. Use the possibilities offered with the selector extensively, because every stage is taking long time, sometimes days. Press OK in order to start executing the process. Once started it can only be aborted by CTRL+C.

During the building process there are three ways how the process can be supervised. The first method is the automatically shown OptiVaNeS terminal:

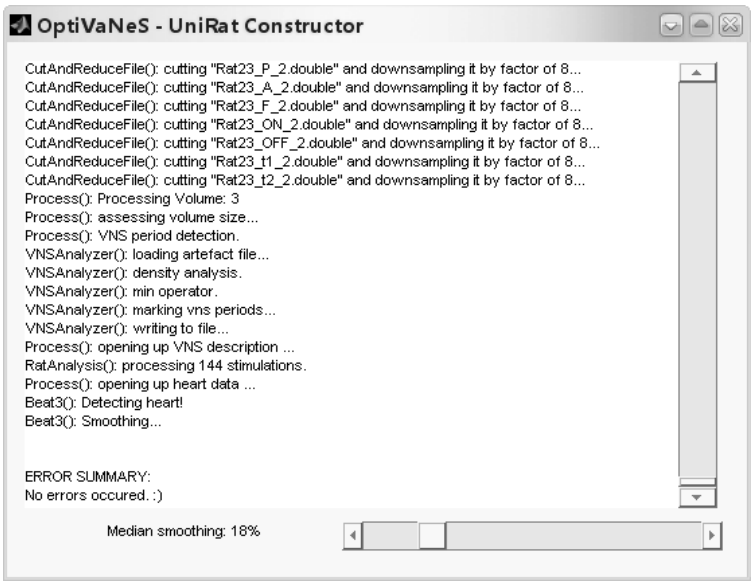


Image 92: OptiVaNeS-terminal

The OptiVaNeS-terminal is read-only and gives a bird-perspective view of the process. It is not as detailed as the output made at the matlab terminal and therefore offers a better overview of the situation. An advantage of the OptiVaNeS-terminal is that it shows any kind of error in the error summary at the bottom. This way you can quickly figure out, whether any errors occured and which. This is not possible with the matlab command line because errors are dumped and the process continues. The second method is the matlab command line where detailed reports are made. The third method is to investigate the Log.txt file in the Processing folder. At the end of the process the TherapyFinder module is opened:

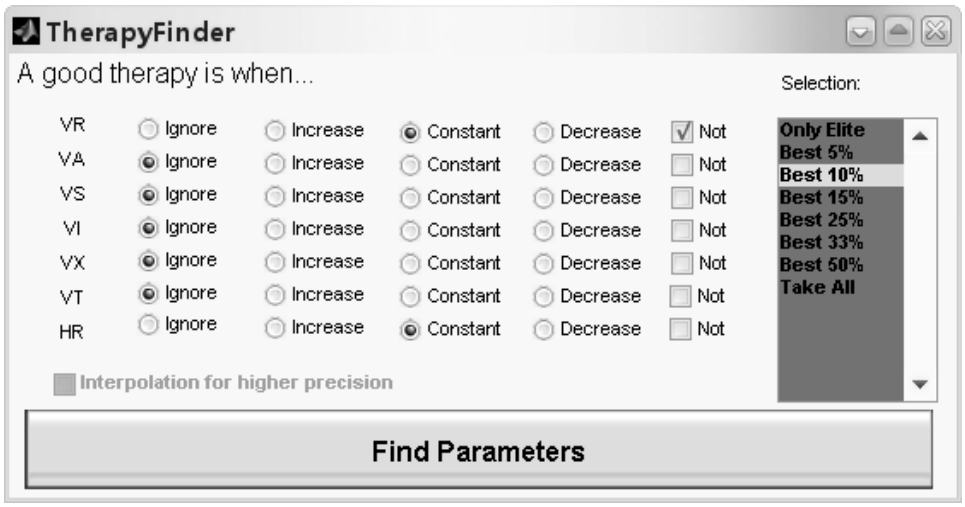


Image 93: TherapyFinder module

7.3.3 What happens?

When BuildAll is started then every checked stage is invoked. Some stages can be skipped but then you must be aware of that latter stager operate with older data. Some of the stages will be auto-checked for consistence and will be repeated even though the user did noch check them.

Generate. If every stage was checked then the first step in the process is the generation of a binary file for every rat. The generation of these files is done by the Process.m files which reside in the „Rat“ subfolders. If all the rats are chosen then there will be a „rat??.double“ file in every of the „Rat“ folders. These files are binary written double values in the multiples of 15.

Compose. After the generation stage the files are merged together into a file called „universal.double“. This is the base data for every data mining method and is referred to as UniRat data. After this stage you can use the SortToPhases() function in order to get the time stripped phase data which should be compacted with the CompactPhase() function.

Genetic Design. The next step is the genetical network design which takes a small but the most demanding fraction of the UniRat data. This process generates the best network for every feature to be modelled. Those templates are saved in the „Default.mat“ file.

Training. Those templates are trained with the full data in the next stage. The trained results are saved in the „temp.mat“ file.

UniRat Graphs. After the networks have been trained you can generate graphs into the „UniRat“ folder for every protocol vector. These graphs should be verified because if they are bogus this means that the training of the networks have failed.

Network Analysis. After the netoworks have been trained it is necessary to produce 2d maps of the network function. Those maps are stored in the files „DataFor1.mat“, „DataFor2.mat“... At the same time 3d diagrams (max and tendency) are generated into the „Transferfunctions“ directory. You can skip some of the feature analysis in the SelectorGUI, where default indexing matrix is [1:7]. You can enter any row matrix with values between 1 and 8. Analysis can be quite time consuming. Therefore a time estimation function was included into this part of the scripts.

Interrogation. In the last step the TherapyFinder is started that uses data from the files „DataFor1.mat“, „DataFor2.mat“... Specify your medical theory in the categories increase, decrease or constant. You can use the not button in order to invert the statement.

7.3.4 Rat Construction Scripts

7.3.4.1 Top Level Report

This report shows interdependencies between the scripts. Built-in functions and files in toolbox/matlab are not shown.

Table 61: Function dependencies in Processing directory

M-files	**Children**(called functions)	**Parents**(calling functions, current dir. only)
ANNConfig		CreateAnnValidateANN
Any		
ApplyEnvelope		FilterChannels
AverageSection	current dir : Count current dir : Find	
BestColumnOrder	current dir : GroupSortByColumns other : D:\Programme\Matlab7\work\Unique.m other : D:\Programme\Matlab7\work\Perm.m	
BuildAll	current dir : GetSetup current dir : ClearDisplay current dir : ClearErrors current dir : NoProgress current dir : SelectorGUI current dir : NewLine current dir : Generate current dir : Compose current dir : FilterChannels current dir : CreateNetwork current dir : TrainNetArray current dir : GenerateGraphsFromNetArray current dir : Error current dir : EnvelopeAnalysis current dir : TherapyFinder subfunction : IsSelected	
BuildAll>IsSelected		BuildAll
CheckNetQuality	current dir : CheckRange current dir : NewLine current dir : FindMostSignificantData current dir : Error subfunction : LookForStored subfunction : AddToPool toolbox : nnet\nnet\newcf.m toolbox :	CreateNetwork

	nnet\nnet\@network\train.m toolbox : nnet\nnet\mse.m	
CheckNetQuality>LookForStored		CheckNetQuality
CheckNetQuality>AddToPool		CheckNetQuality
CheckRange		CheckNetQualityCreateNetwork
ClearDisplay	current dir : OverallStatus	BuildAll
ClearErrors	current dir : OverallStatus	BuildAll
CompactPhaseData		
Compose	current dir : NewLine current dir : GetSetup current dir : SetProgress current dir : VerifyExistence current dir : Convolute Finalize ParamVecs	BuildAll
Convolute		Compose
Count	current dir : OpenUniversal	AverageSectionTotalSection
CreateAnn	current dir : ANNConfig	
CreateNetwork	current dir : NewLine current dir : Evolution current dir : CheckNetQuality current dir : CheckRange current dir : Error toolbox : gads\gads\gaoptimset.m MAY_LAYERS	BuildAll
CrossANN		
DrawCompacted		
EnvelopeAnalysis	current dir : SetProgress current dir : TransferOfANN subfunction : Combinations subfunction : StrippedDimensions subfunction : TimeEstimation subfunction : Expand	BuildAll
EnvelopeAnalysis>Combinations		EnvelopeAnalysis
EnvelopeAnalysis>StrippedDimensions		EnvelopeAnalysis
EnvelopeAnalysis>Seconds		TimeEstimation
EnvelopeAnalysis>TimeEstimation	subfunction : Seconds	EnvelopeAnalysis
EnvelopeAnalysis>Expand		EnvelopeAnalysis
Error	current dir : OverallStatus current dir : Log	BuildAllCheckNetQualityCreateNetworkGenerateSetProgress
Evolution	current dir : NewLine	CreateNetwork
ExactOne		SelectBestParameters

Export		
FilterChannels	current dir : NewLine current dir : OpenUniversal current dir : SetProgress current dir : ApplyEnvelope current dir : SaveUniversal subfunction : MedianFilter InRange	BuildAll
FilterChannels>MedianFilter		FilterChannels
Find	current dir : OpenUniversal	AverageSectionTotalSection
FindMostSignificantData	current dir : OpenUniversal current dir : SetProgress InRange	CheckNetQuality
Generate	current dir : GetSetup current dir : NewLine current dir : Error CreateContro Process	BuildAll
GenerateDiagram	Scales	
GenerateGraphsFromNetArray		BuildAll
GetSetup		BuildAllComposeGenerateLog
GroupSortByColumns	recursion : GroupSortByColumns	BestColumnOrderGroupSortByColumns
Log	current dir : GetSetup	ErrorNewLine
MutationANN		
NewLine	current dir : OverallStatus current dir : Log	BuildAll CheckNetQuality Compose CreateNetwork Evolution FilterChannels GenerateSetProgressTrainNetArray
NoProgress	current dir : OverallStatus	BuildAll
OpenUniversal		Count FilterChannels Find FindMostSignificantData SortToPhases TrainNetArray
OverallStatus		ClearDisplayClearErrorsErrorNewLineNoProgressSetProgress
OverallStatus>OverallStatus_OpeningFcn	subfunction : Draw	
OverallStatus>Draw	subfunction : GetLines subfunction : ScrollTo	OverallStatus_OpeningFcnslider2_Callback
OverallStatus>slider2_Callback	subfunction : Draw	

OverallStatus>GetLines		Draw
OverallStatus>ScrollTo		Draw
SaveUniversal		FilterChannelsSortToPhases
SelectBestParameters	current dir : ExactOne subfunction : ScalarMaximum subfunction : Rows subfunction : Columns subfunction : Depth	pushbutton1_Callback
SelectBestParameters>Rows		SelectBestParameters
SelectBestParameters>Columns		SelectBestParameters
SelectBestParameters>Depth		SelectBestParameters
SelectBestParameters>ScalarMaximum		SelectBestParameters
SelectorGUI		BuildAll
SetProgress	current dir : NewLine current dir : Error current dir : OverallStatus	ComposeEnvelopeAnalysisFilterChannelsFindMostSignificantData
Show	current dir : TransferOfANN	
ShowComparable		
SortToPhases	current dir : OpenUniversal current dir : SaveUniversal	
TherapyFinder		BuildAll
TherapyFinder>pushbutton1_Callback	current dir : SelectBestParameters	
TotalSection	current dir : Count current dir : Find	
TrainNetArray	current dir : NewLine current dir : OpenUniversal toolbox : nnet\nnet\@network\train.m toolbox : nnet\nnet\mse.m CurrentError	BuildAll
TransferOfANN		EnvelopeAnalysisShow
ValidateANN	current dir : ANNConfig InRange	
VerifyExistence		Compose

7.3.4.2 Dependency Report for Rat directories

Built-in functions and files in toolbox/matlab are not shown

Table 62: Function dependencies in Rat?? directories

M-files	**Children**(called functions)	**Parents**(calling functions, current dir. only)
ArtificialVNS		
Beat3	Error NewLine SetProgress	Process
Beat4	current dir : DumpTrack NewLine Error	
BuildAll	toolbox:signal\signal\@sigwin\generate.m toolbox:symbolic\@sym\compose.m GetSetup TrainNetArray UniRat GenerateGraphsFromNetArray	
CreateControl		
CreateStatistics2	current dir : Minimum subfunction : FindRecord Error NewLine SetProgress	Process
CreateStatistics2>FindRecord		CreateStatistics2
CutAndReduceFile	NewLine Error SetProgress	Process
DumpTrack	Error	Beat4GeneratePAFTracksProcessVentilation
Finalize	Error	Process
GenerateHeartRate		
GeneratePAFTracks	current dir : DumpTrack NewLine	Process
GenerateTiming	NewLine	Process
GenerateVentilationData		
GetSize		Process
LowPass	current dir : alFFT NewLine	Process
Merge	NewLine Error	Process
Minimum		CreateStatistics2
NoiseReduction		VNSAnalyzer

Process	current dir : GetSize current dir : VNSAnalyzer current dir : GenerateTiming current dir : DumpTrack current dir : Beat3 current dir : LowPass current dir : Ventilation current dir : GeneratePAFTracks current dir : CutAndReduceFile current dir : Merge current dir : Finalize current dir : CreateStatistics2 urrent dir : VisuResult unknown : Error unknown : NewLine	
RatAnalysis		
VNSAnalyzer	current dir : NoiseReduction NewLine SetProgress	Process
Ventilation	current dir : DumpTrack NewLine Error SetProgress	Process
VisuResult		Process
alFFT		LowPass
test		

7.3.4.3 Function Almanach

Table 63: Function Overview (CD)

File&Function/Script	Description
ANNConfig.m	configuration parameter for artificial networks
alFFT.m	FFT filter
ApplyEnvelope.m	Applies envelope over a spektrum
ArtificialVNS.m	Generates artificial VNS artefact where no natural artefact could be measured
AverageSection.m	Finds mean average stimulation cycle
Beat3.m	Heart beat detection
Beat4.m	Ventilation detector (absolete: use Ventilation.m)
BestColumnOrder.m	Finds best colmn order for the sorting method
BestLimitForVoronoi.m	Finds best limits for LVQ classification
BuildAll.m	Top level build management and control
CalculatePIF.m	Generate PIF values from tables
CheckNetQuality.m	Fitness function (network training)
CheckRange.m	Range checking for scalars
ClearDisplay.m	Clears text in OptiVaNeS terminal
ClearErrors.m	Clears error text in OptiVaNeS terminal
CompactPhaseData.m	Compacts thousands of stimulation features to 81 generic ones
Compose.m	Creates UniRat
Convolute.m	Convolution of matrices
Count.m	Counts no. of specific stimulations in UniRat
CreateAnn.m	Creates prototype network (discrete)
CreateControl.m	Creates a default control parameter structure for a rat
CreateDiagramsFromPhaseData.m	SCRIPT! setup the intern parameter in order to create 7 diagrams for phase x and feature y
CreateNetwork.m	Creates prototype network (continuous-old)
CreateStatistics2.m	Subprogram for statistics generation for individual rats
CutAndReduceFile.m	Cuts rat binaries and reduces sampling rate
DumpTrack.m	Extract a single binary track
EnvelopeAnalysis.m	Creates planes from local transfer functions - very long runtimes!
Error.m	Output error to OptiVaNeS terminal
Evolution.m	Discrete Evolution Algorithm as replacement for analog ga() (discrete)
ExactOne.m	Remove duplet rows
FilterChannels.m	Transform absolute UniRat in relative UniRat
Finalize.m	UniRat to Workspace transporter
Find.m	Finds VNS cycles with specific parameters in UniRat
FindMostSignificantData.m	Finds the hardest data to train at in UniRat
Generate.m	Top level build controler
GenerateDiagram.m	Generates a diagram from GUI selector

GenerateGraphsFromNetArray.m	Automatic generation of cycle graphs for trained network array
GeneratePAFTracks.m	P, A and F Track generation from meta data
GenerateTiming.m	Creates timer channels t1 and t2
GetSetup.m	Global building setup parameter
GetSize.m	Size of a volume
gridfit.m	EXTERNAL SOURCE: non-uniform 2D interpolation
GroupSortByColumns.m	Special group-wise sorting with more than one sort column
K_NN.m	K-NN classifier
List_from_K_NN.m	Result space generation with K-NN classifier
Log.m	Writes information to the log file
LookUpAverage.m	Works like CompactPhaseData but with 2 parameters
LowPass.m	Filter vectors
Merge.m	Puts binary tracks together
Minimum.m	Minimum of two values
MutationANN.m	Mutation function (discrete)
NewLine.m	Output new text line into OptiVaNeS terminal
NoiseReduction.m	Median filter
NoProgress.m	Hide progress bar on OptiVaNeS terminal
OpenUniversal.m	load UniRat to workspace as matrix
OverallStatus.m	Show and set progress bar on OptiVaNeS terminal
PreClassifyCompacted_SCRIPT.m	Pre-Classify compacted phase data
Process.m	Any rat's central building controler
ResultsBrowser.m	GUI component
SaveUniversal.m	Save UniRat matrix to binary file
SelectBestParameters.m	Linguistic inference function
SelectorGUI.m	Build control GUI component
SetProgress.m	Update progress bar in OptiVaNeS terminal
ShowComparable.m	Norm columns and display in a plot
SortToPhases.m	Feature extraction from UniRat to phase data
TherapyFinder.m	Interrogation GUI component
TotalSection.m	Like Find() but plots data.
TrainNetArray.m	Network array training function
TransferOfANN.m	Generates a single transfer function matrix
ValidateANN.m	Check network genes to be valid (discrete)
Ventilation.m	Ventilation detector
VerifyExistence.m	Verifies DAQ protocol in data
VisuResult.m	Visualization GUI for rat binaries
VNSAnalyzer.m	VNS artefact detector

Other not documented/described files have no relevance.

7.4 Additional Hard-Copies