AALBORG UNIVERSITET

Stability of welding

Institut for Elektroniske Systemer
Elektronik og Elektroteknik
Fredrik Bajers Vej 7 B
9220 Aalborg Ø
Tlf.: 9940 8600
http://es.aau.dk

Titel: Stability of welding

Theme: Bachelor of engineering

Project period:
E7, final project for bachelor of engineering

Participant:

Jesper Nysted Andersen

Supervisor:
Per Rubak

Page number: 87

Appendix number and art: 3 bilag, samples, description of Matlab program and Matlab code

Finished 6^{th} of January

Synopsis:

The thesis is a final project for a bachelor of engineering. The objective in the thesis is to determine the stability of the GMA-welding process and the analysis went on with the period method. The short-circuit frequency have been analysed with both a PSD and a period method. It was found to difficult to get a clear result from the PSD. The analysis of the period method ended with the most stable welding being one that didn't go through the welded material.

Read manual

Through the thesis there have been used different methods to show Matlab function. The Matlab functions made is written in bold, **load_file**. Functions incorporated in Matlab is written in italic, *mean*.

References to figures, equations, table, section or chapter is done with one of the handles and then a ref, this will look like this: see figure 2.1. Or something similar to this. A reference to a outside source is done with citep and will look like this: citeref (name, year)

The source code for the Matlab program can be found on the DVD in the folder "Matlab". The functions to plot the different figures in the report is also in the folder "Matlab". The sampled data is laying in the folder "data".

Contents

Chapter 1

Introduction

Welding can be traced back to more than 3000 years ago. Where there have been found tools used for welding iron together. The tools have been found in Egypt and the eastern Mediterranean. Welded gold boxes have been found that can be dated 2000 years back. In the Middle Ages blacksmiths welded iron together by hammering and heating metal. The welding we know of today originated from the eighteenth century where the first arc was accomplished between two carbon electrodes (Cary, 1989). There is a lot of different welding methods and some of the most used today is(Cary, 1989):

- SMA-W - Shielded Metal Arc Welding
- GMA-W - Gas Metal Arc Welding
- GTA-W - Gas Tungsten Arc Welding
- PA-W - Plasma Arc Welding
- Laser welding

The basic idea in these method is to make enough heat to melt the material. These processes is using a filling material in the welding except laser welding. Laser welding is a new method which uses the intensity of a laser to melt the objects. The different method will not be described further only GMA-W which is the method used in this thesis. The GMA-W process is described in Chapter 2

Today the GMA-W process most often is controlled by a micro-controller. The GMA-W process uses many different transfer modes The most used are (Cary, 1989):

- Short-circuit
- Globular
- Spray
- Pulsed-spray

These transfer modes have been used for many years and they are used for different jobs. Today more advanced method is being developed because of the possibilities with the micro-controller. These new method's can control the process much more advanced and they are most often better because they have the possibility to detect changes while welding and change parameters to fit the new parameters. In this thesis the short-circuit transfer mode will be used. The short-circuit mode will be described in chapter 2.

1.1 Corporation

The work in this thesis have been made in cooperation with Migatronic A/S. Migatronic is a company from 1970 that makes welding machines for almost every purpose. The company have the philosophy:

> Tænd, tryk og svejs (Switch on, press and weld)

This philosophy suits the broad selection of welding machines and the function of these.

The machine used in this project is a Galaxy, which is a high end machine for the industry work. The welding machine have pre-installed programs that are made so you decide which material you have to weld in and the thickness of it and the type of gas used then the machine set the voltage, current and gas flow. The machine also have the possibility to control other factors, one of these a hot-start. The hot-start helps the welding to start up faster so the materials is heated up.

Migatronic also have a robot which have been used in the project for the sampling of welding data. The welding robot is described in section 2.1.2

They have also help with question about the project and in particular the welding. The project is based upon a project from Migatronic about a welding quality index.

1.2 Other's work

The stability of the welding process can be found in the short-circuit frequency.Papers about quality and process behaviour is looking at the short-circuit frequency. A description of the articles used for information about the stability of the welding is given in this section.

In the article (Chu et al., 2004) they are describing a way to determine the quality of the welding by making a power spectral analysis. there sampled data is showing the short-circuit frequency as a determining factor for weld process stability. They use a time frequency analysis method to as a automatic defect detector.

In the article (Hermans and Quden, 1999) the process behaviour of short-circuit welding is studied. The article describes the stability of a welding as a result of correspondence between the short-circuit frequency and the calculated frequency of the welding pool. The welding pool will be described in section 2. In the article they hold the voltage and current stable and only changes the wire feed rate to see how the welding frequency act to this and to see how the deviation is changed. They concluded that the process of short-circuit welding only can be performed in a relative small window, dependent of the welding conditions. Maximum process stability happens when the STD of the short-circuit frequency is at a minimum. The best results is occurring when the frequency of the weld pool and the short-circuit frequency is equal.

In the article (S. Adolfsson and Claesson, 1999) they try to find a method to detect flaws in the welding. This analysis is concentrating in a different area than we wish to in this project but gives god analysis of different method to use for monitoring welding. The analysis is detecting flaws in a welding with they analyse the short-circuit frequency from the period of short-circuit and the arc period. This method is shown in figure 1.1

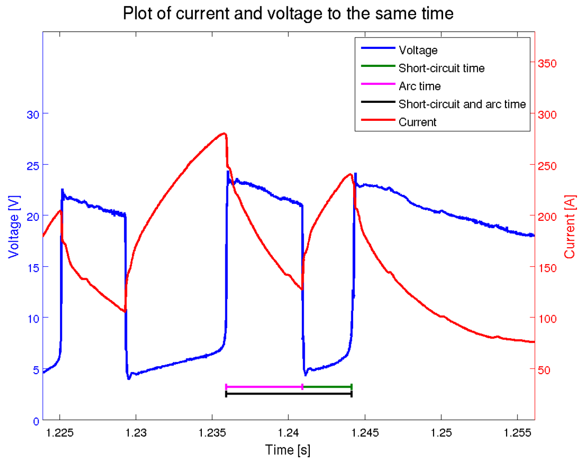


Figure 1.1: *Shows the period for short-circuit and arc*

Where the frequency is determined from equation 1.1

$$f = \frac{1}{t_k + t_t} \tag{1.1}$$

They also look at the short-circuit- and arc time and the current and voltage signals. For flaw detection they finds the use of current and voltage signals characteristic easier to use than the frequency of the short-circuit.

1.3 Thesis outline

In the thesis the procedure for GMA welding and especially the short-circuit transfer mode will be analysed. This will ba an introduction of how the welding works and how the process behave.

The sampling and sampled data will be analysed to see which information they give.

To find the short-circuit frequency two methods will be analysed. The two methods is the PSD, Power Spectral Density, and the period method. In the analysis there will be made an decision on which method to use.

In the end a estimation of the welding stability will be given.

Thesis statement

Analysis of the short-circuit frequency to determine the stability of the welding process

Chapter 2

Welding method

Welding is used in many different areas, normally it's used to weld two pieces of metal together. The analysis made in this project is made upon welding on one piece of metal in a straight line, shown in figure 2.3. There has been used a robot to the test, and therefore some of the thing described in this thesis may not be true for a welding made by a welder.

2.1 The welding method GMA-W

GMA-welding is also known as MIG/MAG welding, which stand for Metal Inert Gas and Metal Active Gas. The difference between these to method is the gas used, else the method is the same. In figure 2.1 a simple sketch of the process for GMA-W is shown.

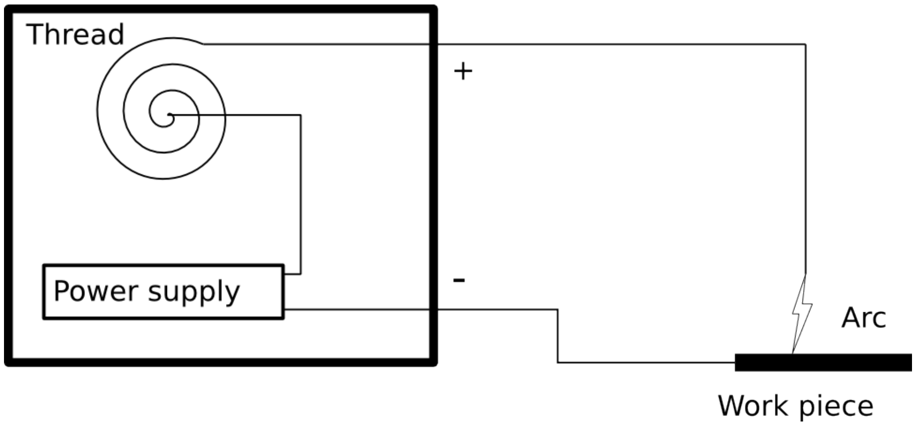


Figure 2.1: *Simple diagram of the welding process for GMA-Welding*

A basic description for the process: A power source is connected to the work piece and a wire. This wire is a filling material. The wire is then heated by the power supply and begins to melt and creates a droplet. This droplet can be transferred in different method. The different transfer method is listed in section 1. They transfers the melted material in different ways. The method used in this thesis will be short-circuit which will be explained in section 2.1.1. When the process is running a weld pool is created at the work piece. The weld pool is melted metal from the filling material and work piece which is floating until it has been cooled down.

The MIG/MAG welding method is most often referred to as a automatic or semi automatic process, because of the use of a motor to deliver wire for the welding. The semi automatic process is with the use of a welder and the automatic process there is used a robot.

As mentioned GMA-W have two different method for the welding MIG or MAG welding. The difference is the gas which is used to protect the welding. In MIG welding the gas only protect the welding for disturbances from the air. In MAG welding the gas can be a support to the welding and control how the welding shall behave. At Migatronic a helium mix was used which is a active gas. The influence from the gas isn't further examined.

2.1.1 Short-circuit welding

The transfer mode used in this thesis is short-circuit welding. This method transfers material after each short-circuit, which makes it ideal for analysis. The short-circuit transfer is shown in figure 2.2.

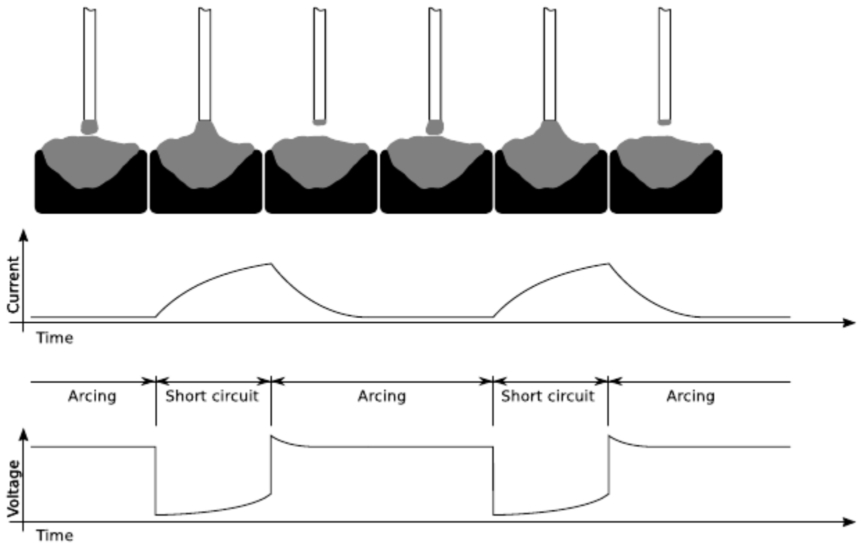


Figure 2.2: *Transfer of material, with illustration of voltage and current (electric, 2006)*

The transfer of material is happening at the same time as the short-circuit. In arc time the filling material is beginning to melt and create a droplet. When the droplet comes in contact with the work piece the droplet will transfer to the work piece. If there is a stable frequency in the short-circuit the drop size of the filling material will be consistent.

2.1.2 Welding robot

In this project there has been used a welding robot. The welding robot has been used under the test welding. The use of a robot removes some parameters that can influence the welding if there have been used a welder. The robot makes sure that the welding pistol is kept at the same distance to the object that has to be welded. It also makes sure that the speed of which the pistol is moving over the object is fixated. In figure 2.3 a welding made by the robot is shown.

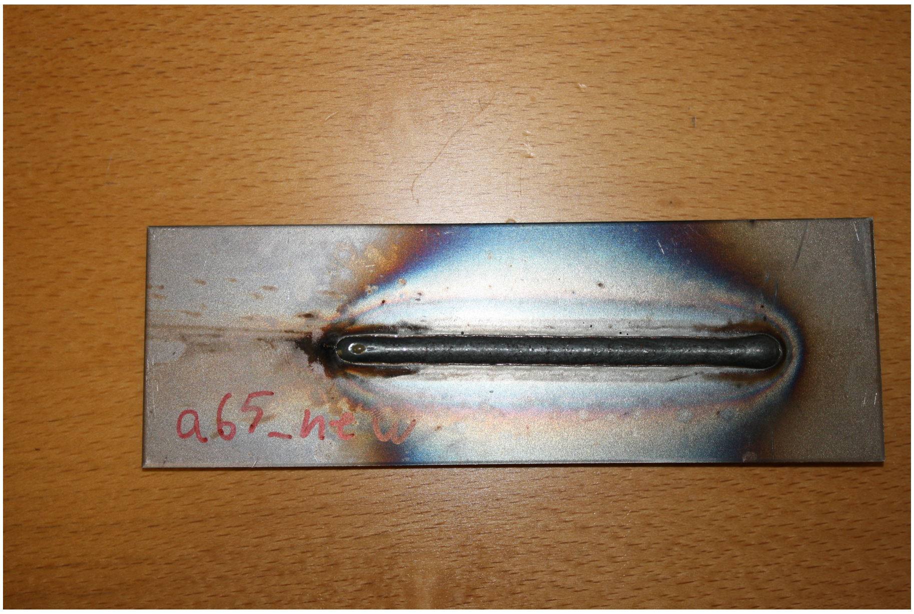


Figure 2.3: *A welding made with the welding robot*

All welding performed is done in a straight line on a single piece of metal.

Chapter 3

Sampling

The data used in the analysis have been sampled with Midas 4.0, which is a program that can sample signals and video together. The video part have not been used in this thesis. In this thesis the program samples the voltage and current to the same time. The sampling frequency used is discussed in section 3.1. A analysis of the sampled date is found in section 3.2.

3.1 Sample frequency

The sampling frequency needed to find the short-circuit frequency is depending on how good a result is needed. The voltage signal being sampled is a square signal so the higher the sampling frequency the better the shift between high and low can be recorded. In figure 3.1, a step from high to low is plotted.

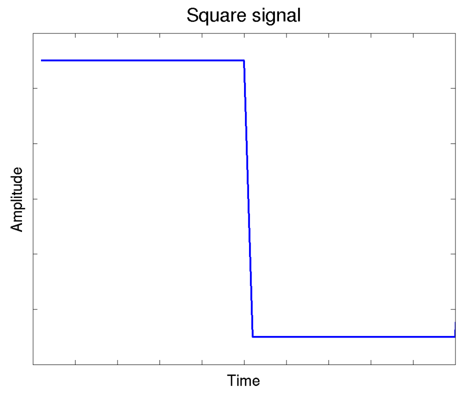


Figure 3.1: *The figure shows the step from high to low*

The settings Migatronic is using for the program is a sampling frequency at 50 kHz. This means that the maximum error between high and low will be a time of:

$$T = \frac{1}{50000} = 0.00002[s] \tag{3.1}$$

This error can occur at the beginning and in the end. To see how much influence this has on the voltage signal a sampling is conducted with 50 kHz sampling frequency. In figure 3.2 a histogram is shown of the short-circuit frequency.

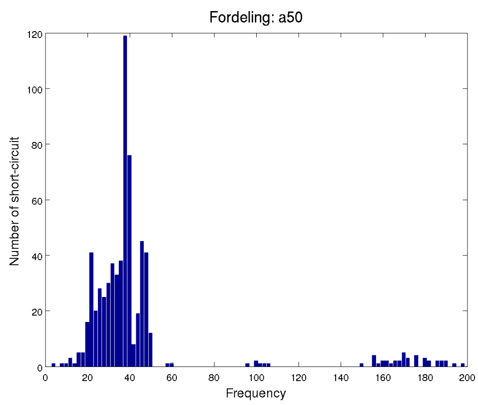


Figure 3.2: *The figure shows the short-circuit freqeuncy for a welding*

A similar result as the one in figure 3.2 is obtained from the other welding. The frequencies is below 250 Hz, therefore a maximum error estimation is made upon this.

$$T = \frac{1}{250} = 0.004s \quad (3.2)$$

With a maximum error of 0.00004 s, the 250 Hz signal will vary with $\mp$ 2.5 Hz.

$$T = 0.00404s \quad (3.3)$$
$$\updownarrow \quad (3.4)$$
$$f = \frac{1}{0.00404} = 247.5Hz \quad (3.5)$$
$$T = 0.00396s \quad (3.6)$$
$$\updownarrow \quad (3.7)$$
$$f = \frac{1}{0.00396} = 252.5Hz \quad (3.8)$$

In figure 3.3 is the error estimation plotted with the shift from high to low.

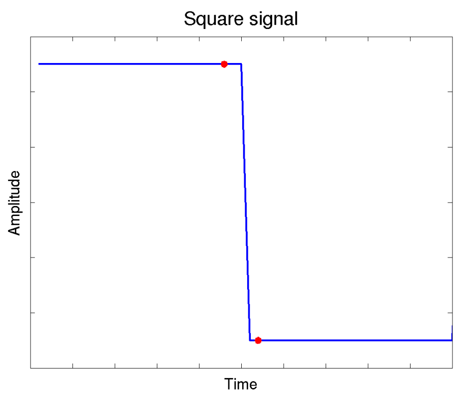


Figure 3.3: *The figure shows a shift from high to low with the maximum error plotted*

3.2 Sampled data

In this section some similarities is found in different welding's. There will be looked at four different welding's in this section, the other welding's are similar. The welding there have been made is described in appendix A1

The beginning of each welding shows that the process takes some time to stabilise, especially if the welding parameter isn't optimal. In figure 3.4 to figure 3.7 four plot of the voltage behaviour in the beginning of four different welding's. Figure 3.4 shows a welding made by Jesper from Migatronic. The three others are examples from other welding's where the machine has been sat by me.

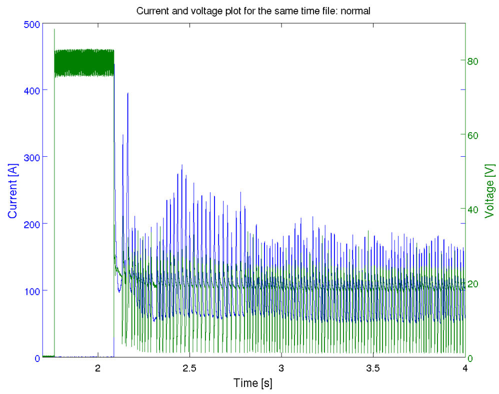


Figure 3.4: The plot is showing the voltage and current plotted to the same time for the welding made by Jesper

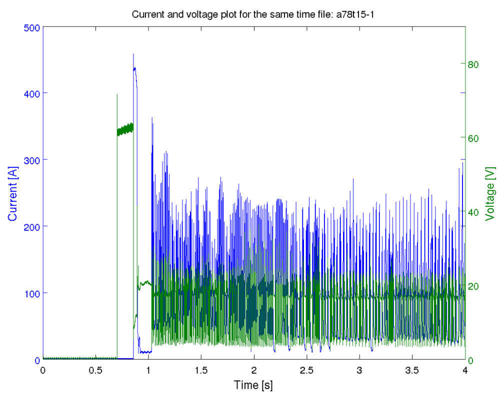


Figure 3.5: The plot is showing the voltage and current plotted to the same time for the welding a78t15-1

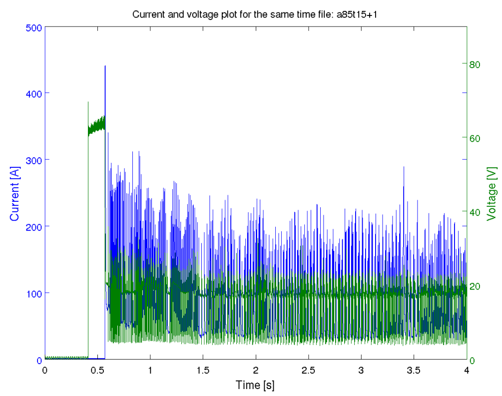


Figure 3.6: The plot is showing the voltage and current plotted to the same time for the welding a85t15+1

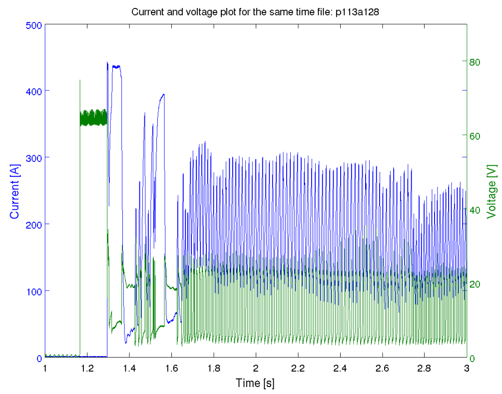
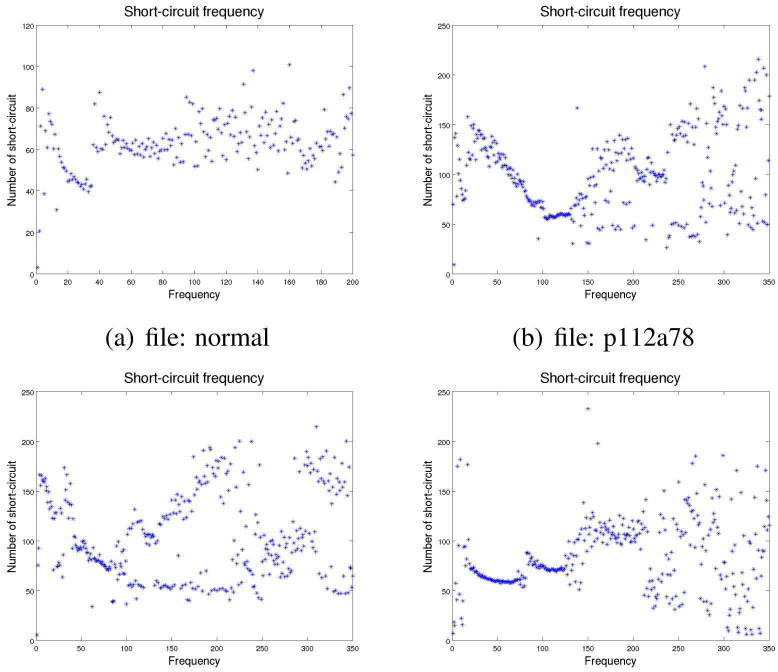


Figure 3.7: The plot is showing the voltage and current plotted to the same time for the welding p113a128

It can be seen that the welding machine is starting with a high voltage. The high voltage continue until the wire comes in contact with the work piece. The current have to be large in the beginning so the material can be heated up and make a much better beginning. This is also the reason for the welding machine to have a hot-start, this is to make the beginning of the welding to look like the rest of the welding. The hot-start influence, mentioned in section 1.1, can be seen in the four plot in figure 3.8



(a) file: normal

(b) file: p112a78

(c) file: p112a85-2

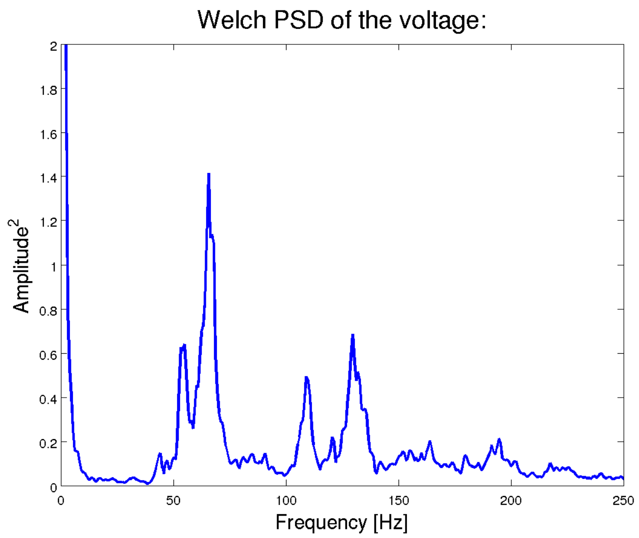
(d) file: p113a128

Figure 3.8: *Four different welding's. The plot is showing the frequency of each short-circuit*

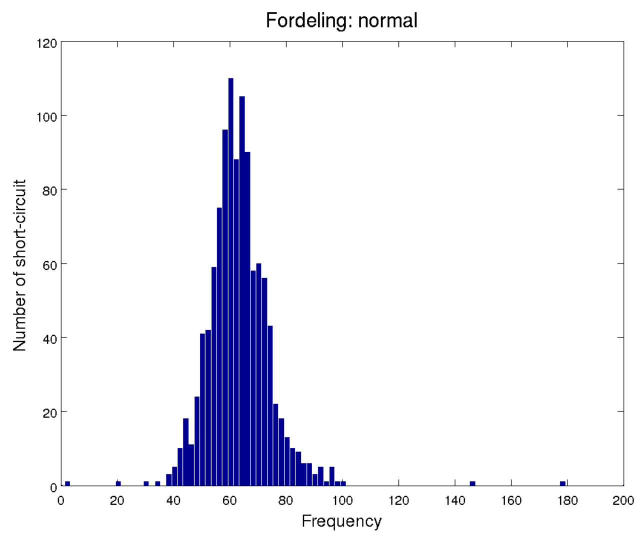
In the four plot the hot-start function work can be seen at the beginning. In figure 3.8(a) the hot-start stops around short-circuit 80, for figure 3.8(b) it's about 140, for figure 3.8(c) it's about 75 and for figure 3.8(d) it's about 135. The unstability in the start is because of the material being cold. When the material is warm up the frequency is stable for a while because of the hot-start. When the hot-start is finished the normal parameters have to take over and the better the machine is sat the more stable the frequency will be.

In figure 3.9 to 3.12 there is plotted four different PSD of the same data. The four different PSD is in different time slot of the welding. In each figure there is plotted a PSD of the data set and the period method. The PSD and period method is described in section 4.

In figure 3.9, it's all the data, in figure 3.10 the first two second is removed, in figure 3.11 it from 2 sec to 8 sec. and in figure 3.12 it's from 8 sec. to 14 sec.

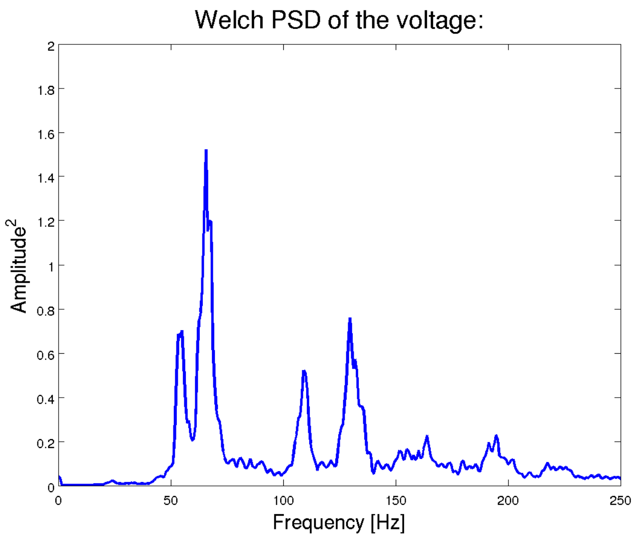


(a) PSD of the hole data set

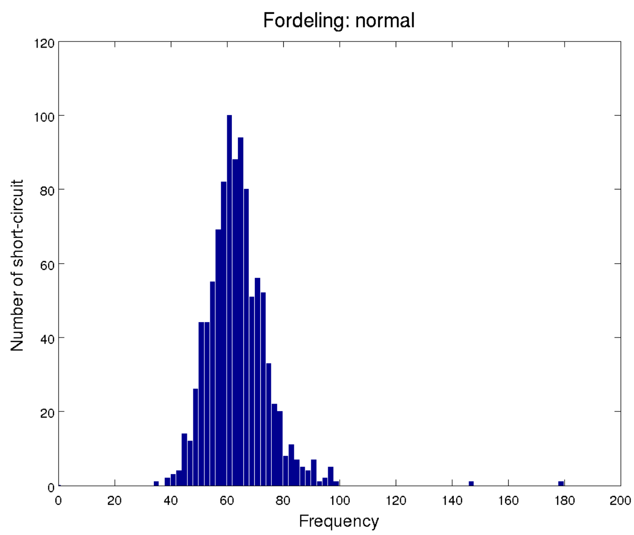


(b) Period method of the hole data set

Figure 3.9: Plot of the frequency for the hole data set, figure a is the PSD and figure b is the period method

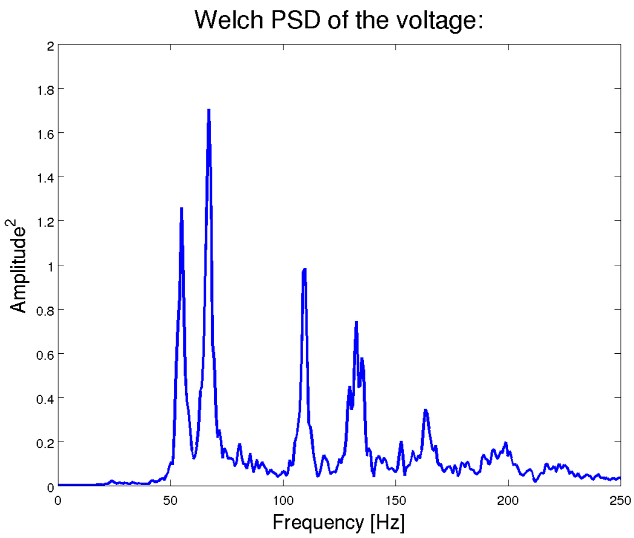


(a) PSD of the hole data set

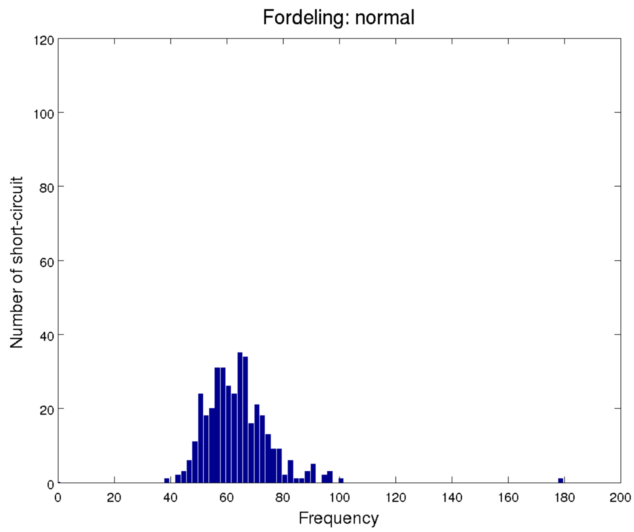


(b) Period method of the hole data set

Figure 3.10: Plot of the frequency for the hole data set except for the first 2 sec., figure a is the PSD and figure b is the period method

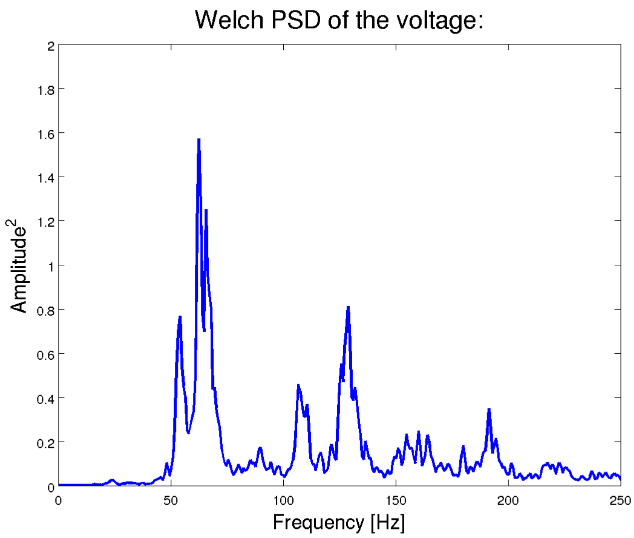


(a) PSD of the hole data set

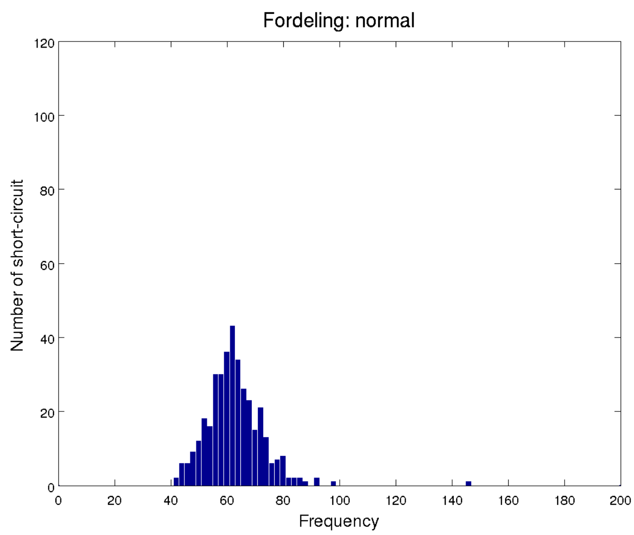


(b) Period method of the hole data set

Figure 3.11: Plot of the frequency for 2 sec to 8 sec., figure a is the PSD and figure b is the period method



(a) PSD of the hole data set



(b) Period method of the hole data set

Figure 3.12: Plot of the frequency for 8 sec. to 14 sec., figure a is the PSD and figure b is the period method

The Figures shows that the main peaks is located the same places. Two of the peaks have a larger amplitude in figure 3.11. This possible happens because the process haven't stabilised yet.

Chapter 4

Frequency analysis

Two methods is used to find the frequency for the short-circuit. One of the methods calculates the frequency from the time when there is short-circuit and the time when there is arc. This method is described in section 4.1.

The other method is a PSD, Power Spectrum Density, which is a type of DFT, Discrete Fourier Transform. The PSD is further explained in section 4.2.

4.1 Period method

This period method calculates the frequency from each of the short-circuit and arc periods. In figure 4.1 the voltage and the current is plotted to the same time. The short-circuit, arc and short-circuit+arc period is marked in the plot.

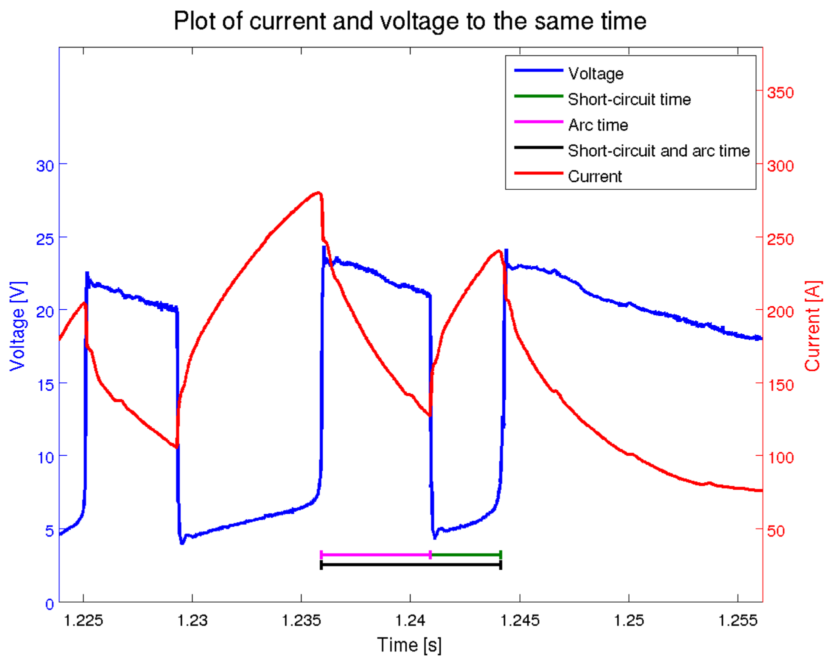


Figure 4.1: *Plot of the voltage and current, with the periods marked.*

The frequency for a period can be calculated from equation 4.1.

$$f_{Period} = \frac{1}{P_{short} + P_{arc}} = \frac{1}{P_{short+arc}} \tag{4.1}$$

The Matlab function **gettime** finds the time for short-circuit start/stop and the time for arc start/stop. See appendix B3 for a description of the program and appendix C3 for the code. In figure 4.2 the voltage signal is plotted and the times found by **gettime** is plotted in.

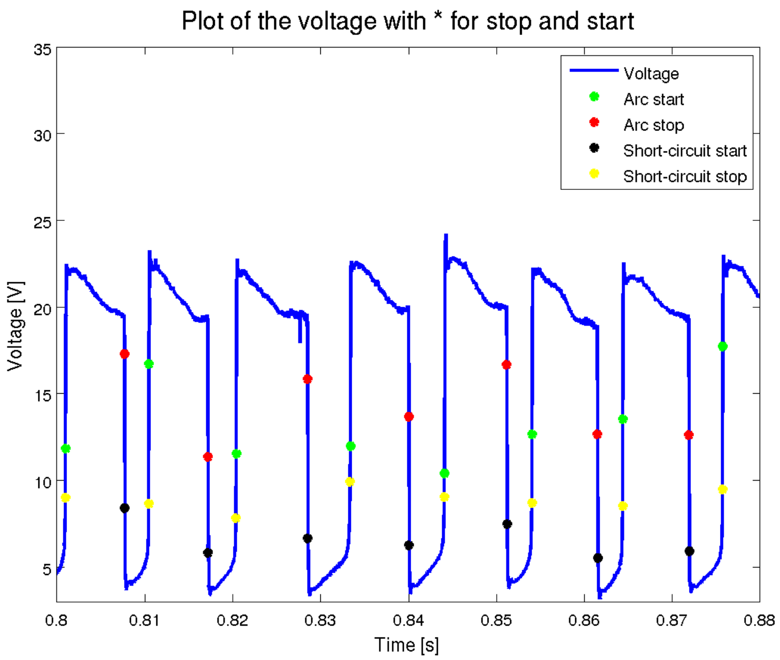


Figure 4.2: *Plot of the voltage, with the times found by* ***gettime***

The dots doesn't fit exactly with the amplitude of the voltage signal. This is because of a simplification in the function **gettime**. With the calculation of the periods the duty-cycle can be found using equation 4.2

$$duty = \frac{P_{short}}{P_{short} + P_{arc}} \tag{4.2}$$

The function **freq_duty** calculates the frequency and duty-cycle for each period found by **gettime**. The frequency is calculates from equation 4.1 and the duty-cycle is calculated by equation 4.2. The function **freq_duty** see appendix B4 for the description and appendix C4 for the code.

When the frequencies is calculated they are split into 2 Hz areas, and a histogram is created with the function *bar*. The function that split the frequencies into these areas is **arrange_freq_duty**. See appendix B5 for a description of the program and appendix 5 for the code.

4.2 Welch PSD

There are different method to calculate the PSD the method used here is the Welch-PSD. The Welch-PSD uses a window function on the data, and overlapping. The window and overlap have influence on the variance of the PSD. A standard Welch-PSD uses 50% overlap. The Matlab function **frequency_psd_pwelch** see appendix B6 for a description and appendix C6 for the code.

The overlap is used to decide how much of the signal is reused. In figure 4.3 a sketch of the process of the Welch-PSD is plotted.

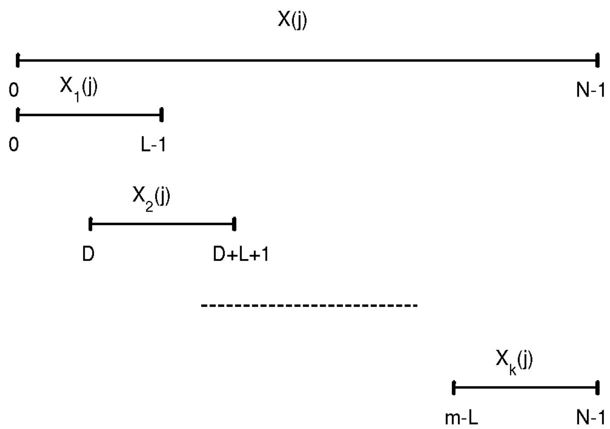


Figure 4.3: *Description of the Welch-PSD Welch (1967)*

Welch is based on averaging. Larger overlap means more spectra to average over which thereby gives a better estimate -> lower variance.

4.2.1 Window

The window have influence on the precision of the PSD. In the figures shown next, the Welch-PSD is plotted with different window size. The overlap is all sat at 50 % and the overlap will be discussed in section 4.2.2. The window used by *pwelch* is the hamming window.

The frequency resolution of the PSD is defined in equation 4.3

$$\triangle f \approx \frac{1}{T_{window}} \tag{4.3}$$

Where T_{Window} is defined in equation 4.4

$$T_{window} = L \cdot \frac{1}{T} \tag{4.4}$$

where L is the length of the window and T is the sampling frequency. With a sampling frequency of 50000 Hz and a window length of 50000 the frequency resolution is 1 Hz. In figure 4.4 the window size is as long as 1 second with a sample frequency of 50kHz

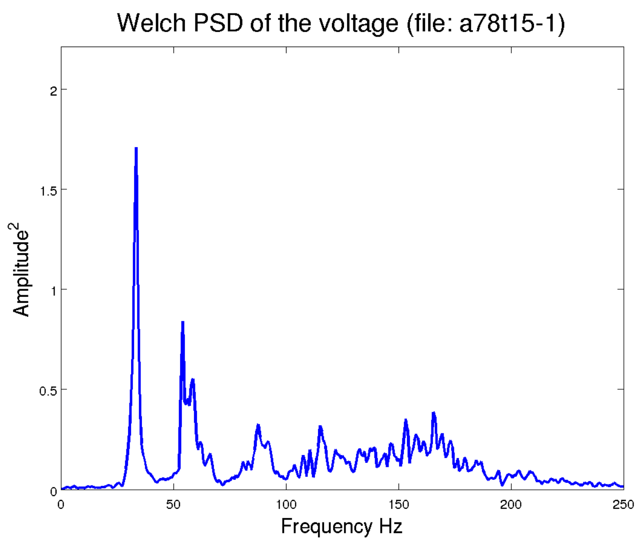


Figure 4.4: *Welch-PSD, window = 50000, noverlap = 25000, Fs = 50000,* $\triangle f \approx$ *1 Hz*

If the window size is reduce by half it's seen that the precision is worse. In figure 4.5 the Welch-PSD is plotted with window size of 25000.

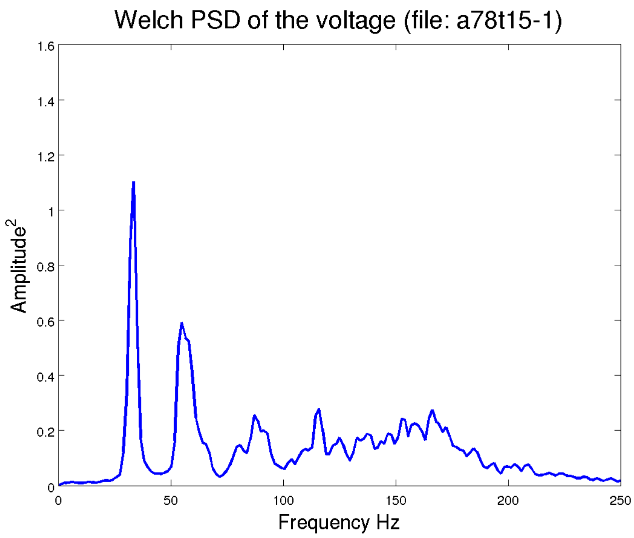


Figure 4.5: *Welch-PSD, window = 25000, noverlap = 12500, Fs = 50000,* $\triangle f \approx 2$ *Hz*

The details isn't as dominant, but the important characteristics of the PSD is preserved. If the window size get to low the details in the PSD falls together as in figure 4.6, caused by the low frequency resolution.

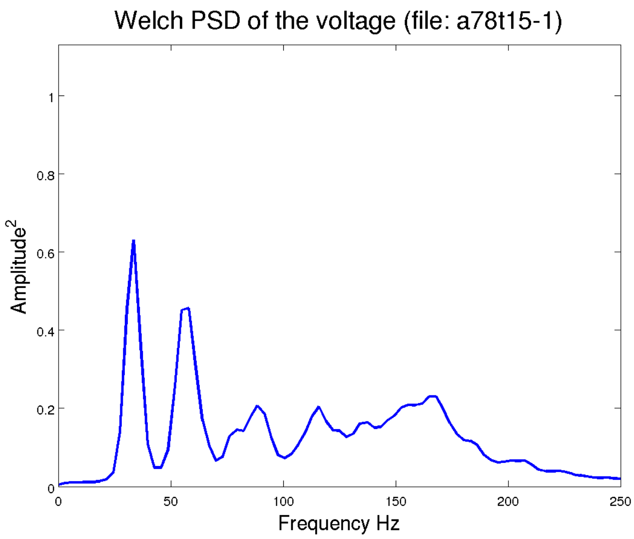
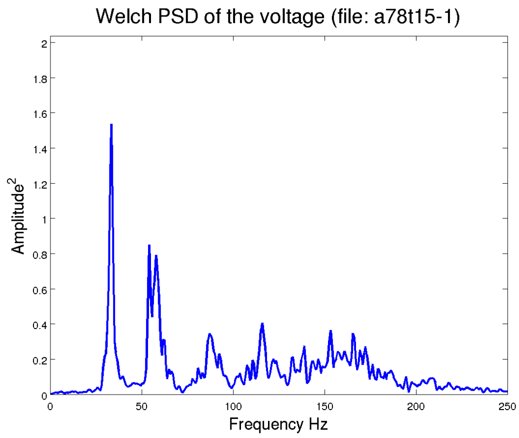


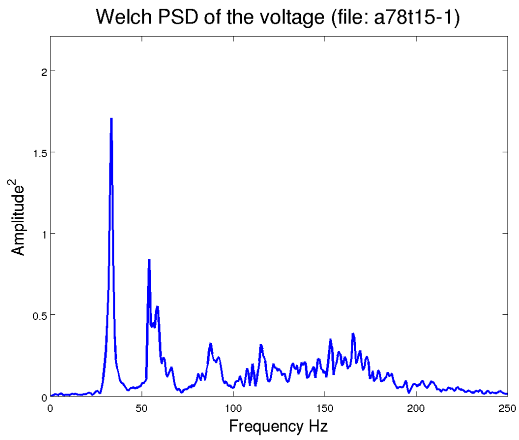
Figure 4.6: *Welch-PSD, window = 10000, noverlap = 5000, Fs = 50000,* $\triangle f \approx 5$ *Hz*

4.2.2 Overlap

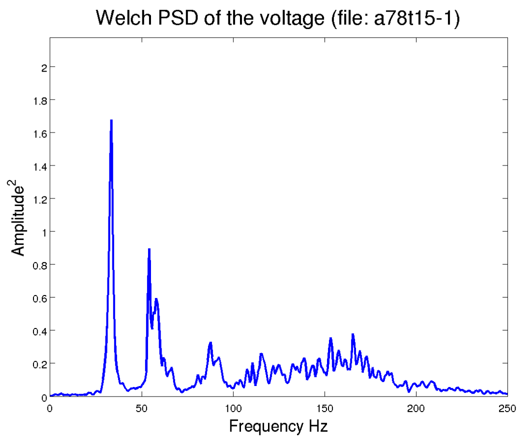
In the next three figures the Welch-PSD is plotted with different overlap, a = 25%, b = 50% and c = 75%. In figure 4.7 the PSD is plotted with a window length of 50000. In figure 4.8 the PSD is plotted with a window length of 25000. In figure 4.9 the PSD is plotted with a window length of 10000.



(a) *Overlap: 12500, 25 %*

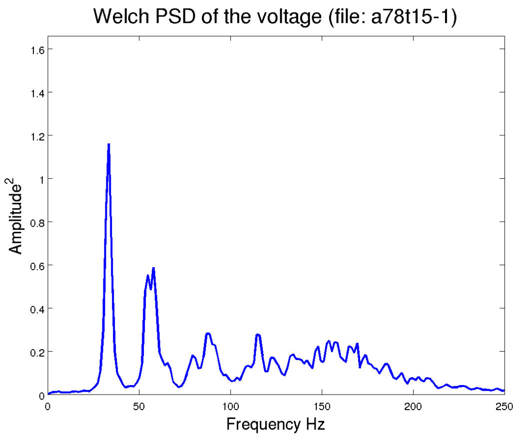


(b) *Overlap:25000, 50 %*

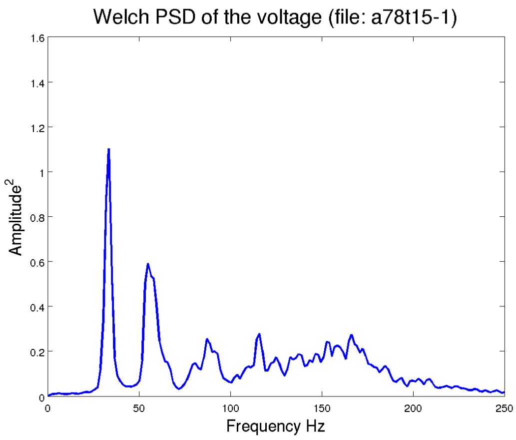


(c) *Overlap: 37500, 75 %*

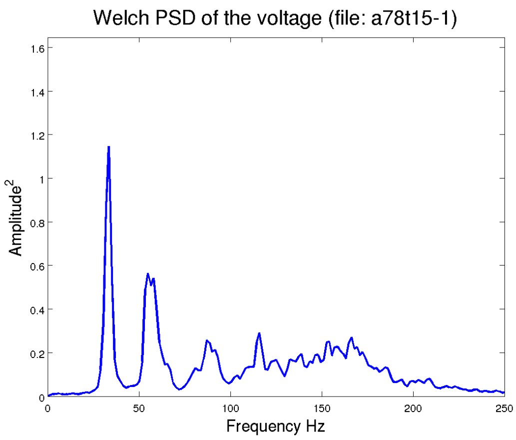
Figure 4.7: *The three figures shows the Welch-PSD with different overlap: window = 50000, Fs = 50000*



(a) *Overlap: 6250, 25 %*

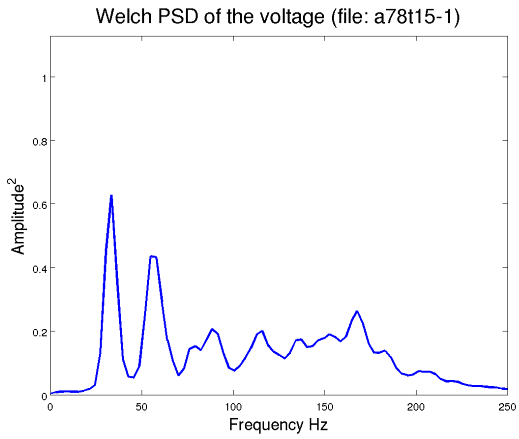


(b) *Overlap:12500, 50 %*

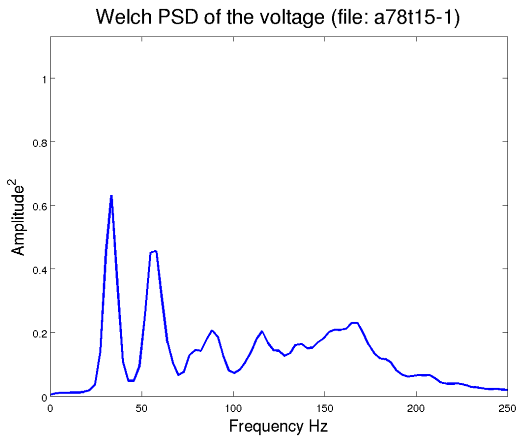


(c) *Overlap: 18750, 75 %*

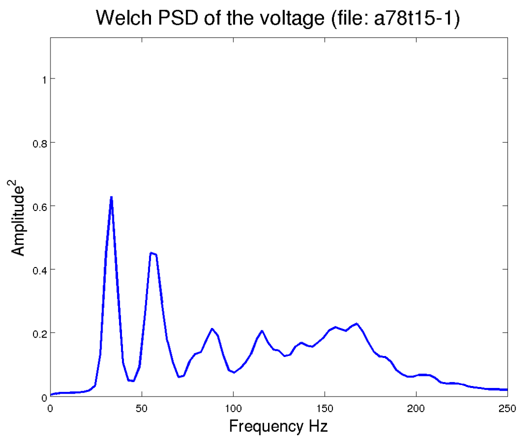
Figure 4.8: *The three figures shows the Welch-PSD with different overlap: window = 25000, Fs = 50000*



(a) *Overlap: 2500, 25 %*



(b) *Overlap:5000, 50 %*



(c) *Overlap: 7500, 75 %*

Figure 4.9: *The three figures shows the Welch-PSD with different overlap: window = 10000, Fs = 50000*

From the three figures 4.7 to 4.9 it can be seen that the influence from the overlap have a small influence in the measured data.

4.2.3 Duty-cycle

In the calculation of the PSD the duty-cycle of the signal is of great importance. The duty-cycle determines how much influence the harmonic frequencies have and the amplitude of these when the time signal is Fourier transformed. The first harmonic frequency is also called the fundamental frequency. The next four figures shows the influence from the duty-cycle on the Welch-PSD. The signal used is 100 Hz square signal with sampling frequency of 5 kHz. In figure 4.10 the square signal is plotted.

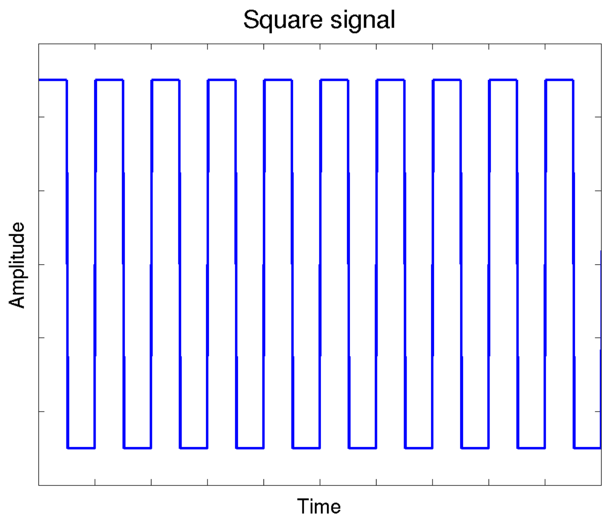


Figure 4.10: *Square signal used, the one shown is 50 % duty-cycle*

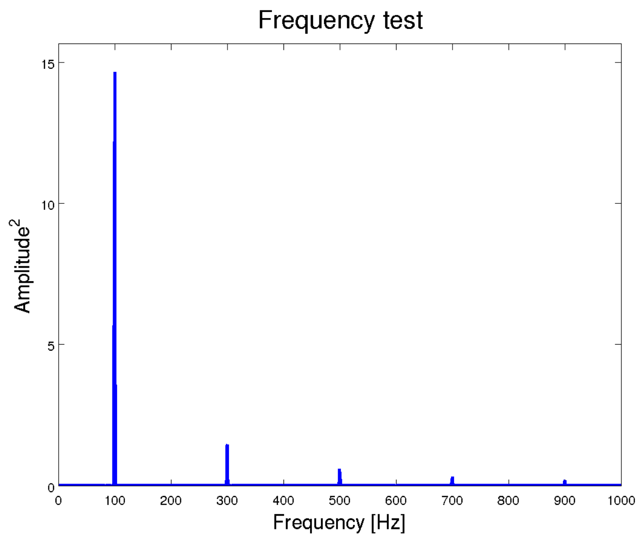


Figure 4.11: *100 Hz signal with duty-cycle of 50 %, Welch-PSD, window = 5000, noverlap = 3500, Fs = 5000*

With a duty-cycle of 50 % the only harmonic frequencies that gives a result is the uneven ones. The 1. harmonic frequency is easy to distinguish from the other harmonic frequencies.

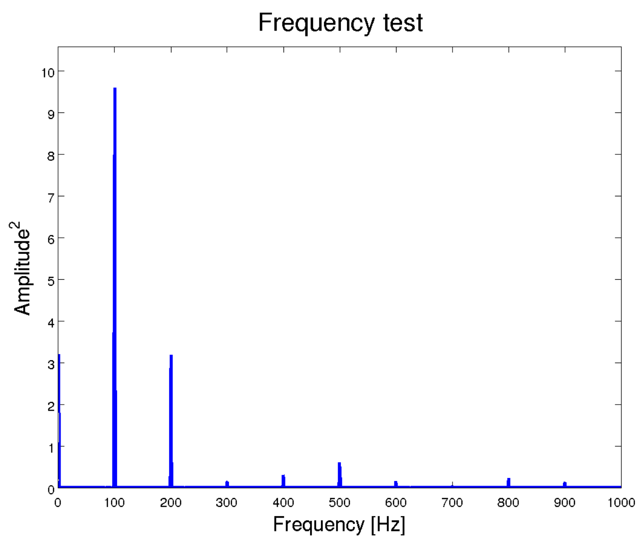


Figure 4.12: *100 Hz signal with duty-cycle of 70 %, Welch-PSD, window = 5000, noverlap = 3500, Fs = 5000*

With 70 % duty-cycle the second harmonic frequency is about 30 % the amplitude than the first harmonic frequency.

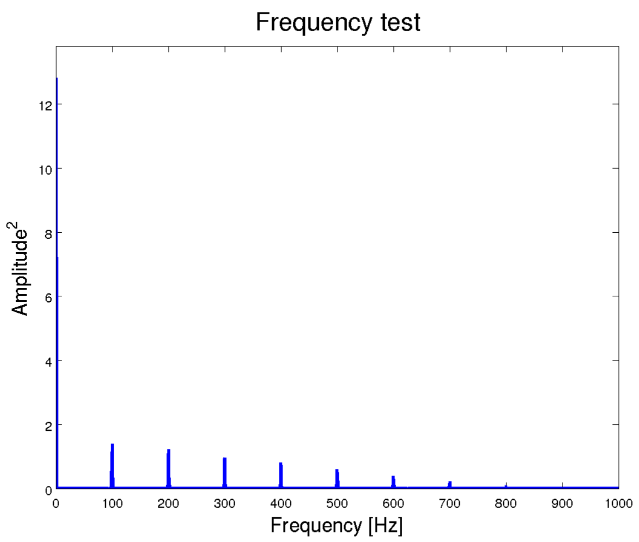


Figure 4.13: *100 Hz signal with duty-cycle of 90 %, Welch-PSD, window = 5000, noverlap = 3500, Fs = 5000*

With 90 % duty-cycle the harmonic frequencies decay slowly.

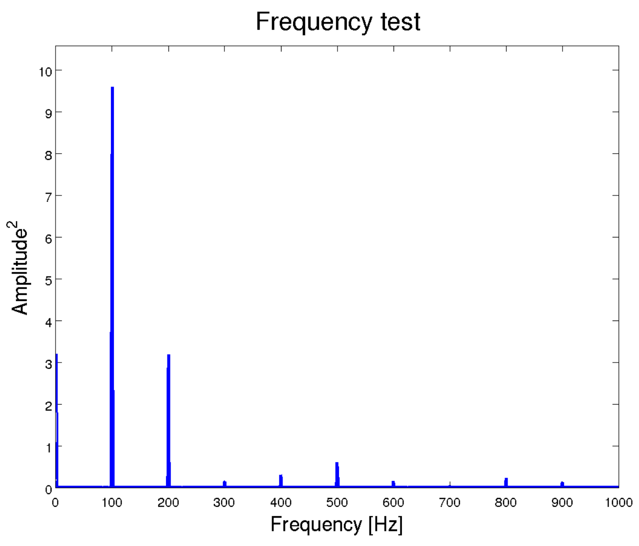


Figure 4.14: *100 Hz signal with duty-cycle of 30 %, Welch-PSD, window = 5000, noverlap = 3500, Fs = 5000*

Figure 4.14 gives the same result as figure 4.12. This is because the function is symmetric around the 50 % duty-cycle.

A test on a square signal with frequency of 100 Hz and 200 Hz, shown in figure 4.15, was also conducted.

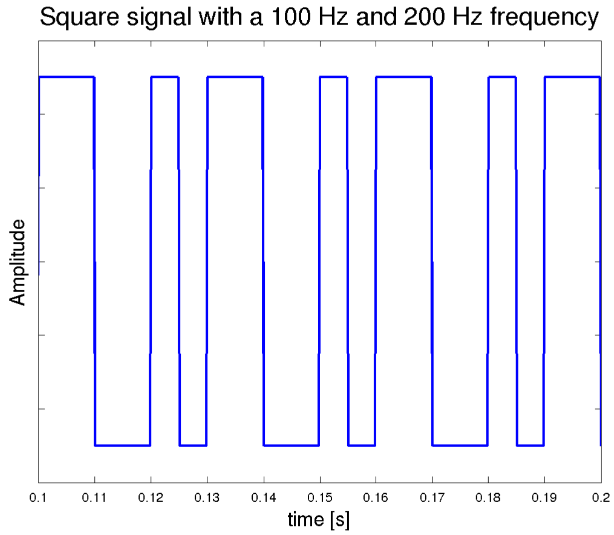


Figure 4.15: *100 Hz + 200 Hz square signal with 50 % duty-cycle used to test how the PSD act on a sample with two frequencies*

In the PSD of this signal a problem with the PSD is found. In figure 4.16 the PSD of this signal is plotted.

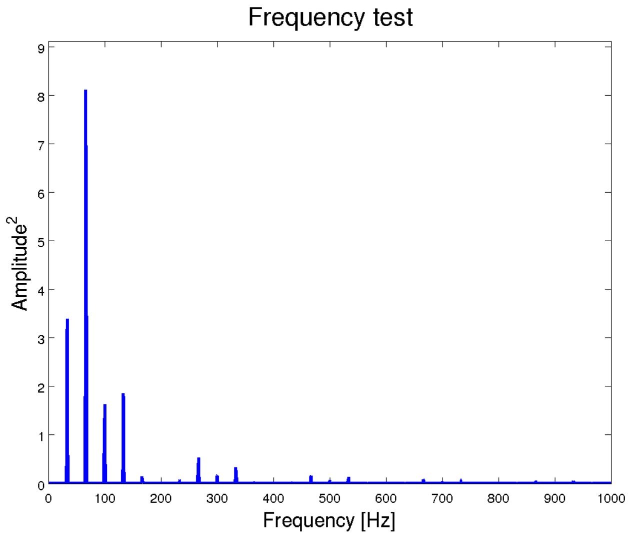


Figure 4.16: *PSD of the square signal with a frequency of 100 and 200 Hz.*

The 200 Hz signal disappear and two lower signal shows. The two signal have a frequency at 33 and 66 Hz. The 66 Hz signal can be from the 100 Hz and 200 Hz signal together which will have a time of:

$$t_{100+200} = 1/100 + 1/200 = 0.015 \quad (4.5)$$

$$\updownarrow \quad (4.6)$$

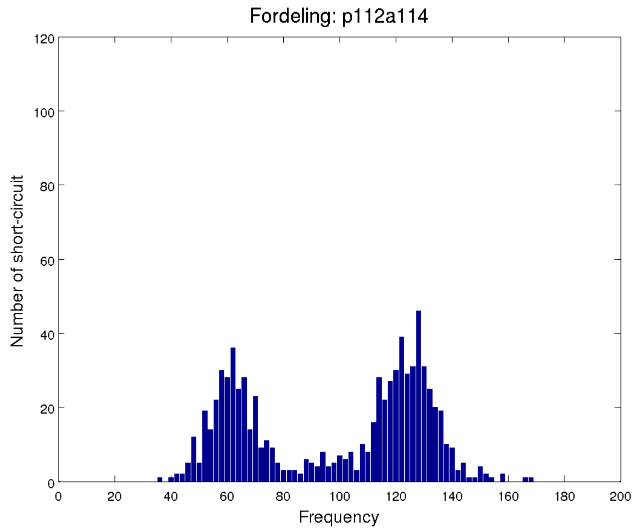
$$f = \frac{1}{0.015} \approx 66Hz \quad (4.7)$$

And two periods of this will give the 33 Hz signal.

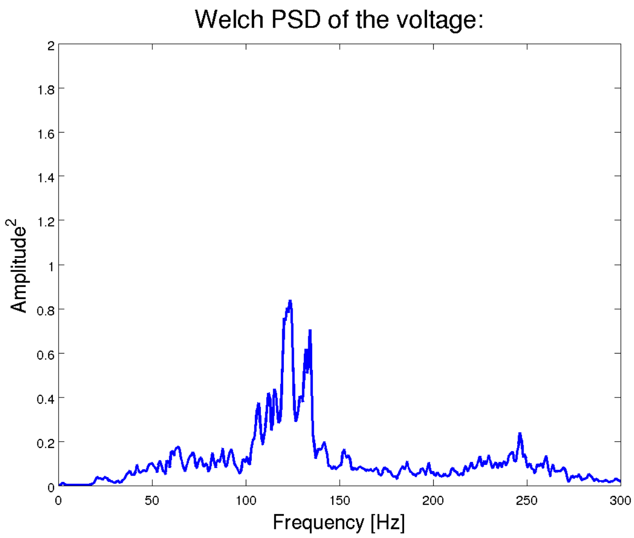
From the four figures it is seen that if the duty-cycle become to large, or small, it can be difficult to distinguish the first harmonic frequency from the other harmonic frequencies.

4.3 Comparison

In this section a comparison between the two method will be analysed. The two methods is compared in figure 4.17. As seen there is a substantial difference between the two methods.



(a) Period



(b) PSD, pwelch, window = 50000, noverlap = 25000

Figure 4.17: *Shows the frequency output from the Welch-PSD and the period method*

Figure 4.17(a) is the period method which gives two areas with frequencies one area with centre around 60 Hz and another area with centre around 120 Hz. Where as the PSD method shown in figure 4.17(b) gives a centre around 120 Hz and doesn't have significant frequencies around the 60 Hz area. In figure 4.18 the short-circuit frequency is plotted as a function of the duty-cycle.

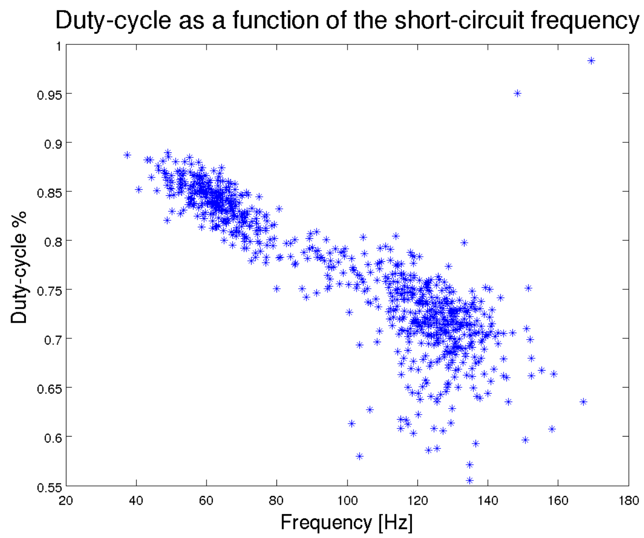


Figure 4.18: *Plot of the short-circuit frequency as a function of the duty-cycle*

From this it can be seen that the frequencies around the 60 Hz area have a duty-cycle around 80 - 90 % and the frequencies around 120 Hz is laying around 70 % duty-cycle. The influence of the duty-cycle does that the 60 Hz frequencies area much smaller in amplitude.

4.4 Conclusion

In this chapter we have looked at the two different ways to obtain the short-circuit frequency, the period method and the Welch-PSD. The period method calculates the frequency from each welding period, the period where the process is short-circuited and the arc period.

In the analysis of the two method it have been found that the Welch-PSD and the period method gives different result, which can be seen in section 4.3. It was found that the Welch-PSD makes some of the frequencies disappear. A possibility for this reason is the duty-cycle of the periods, discussed in section 4.2.3.

Different window size have been used to illustrate the quality changes of the PSD with different window size. The quality changes can be seen in section 4.2.1 The influence from the overlap have also been analyses and in this case the overlap don't have much influence, which is seen in section 4.2.2.

In section 4.2.3 the duty-cycles influence in the PSD was analysed. It was found that the duty-cycle have a great deal to say on how the signal is represented with the harmonic frequencies. If two different frequencies is changing between each other the PSD can hide the information from one of them.

In the further analysis the PSD will not be used instead the period method will be analyses in the next chapter to determine the stability of the welding.

Chapter 5

Stability estimation

5.1 STD Theory

The STD is used to give a representation of the deviation from the mean value of a given signal. The stability can be seen as the deviation from the mean. If the deviation from the mean value is small the stability is better than if the deviation is larger. The STD is defined as the square-root of the variance. Where the variance is calculated from equation 5.1.

$$\sigma_X^2 = E[D(X)] = E[(X - m_X)^2] \quad (5.1)$$

D(X) is the square of the deviation of the variable X about it's mean. m_X is the expected value of X also know as the mean. The variance is the expected value of D.

For the sample x the expected value can be approximated to the sample mean which can be calculated from equation 5.2.

$$\bar{x} = \frac{1}{n-1} \cdot \sum_{i=1}^{n} x_i \quad (5.2)$$

Where x_i is the i'th sample, n is number of samples and $\bar{x}$ is the sample mean. Equation 5.1 can be rewritten to equation 5.3

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (5.3)$$

Where $\bar{x}$ is the sample mean. The STD is then defined as equation 5.4.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (5.4)$$

5.2 Sample mean and Median

The sample mean is defined as the average of the data set. The sample mean is defined in equation 5.2. This can be a problem if the data set contains extreme value. This is where the sample median, $\tilde{x}$, comes in handy it defined in equation 5.5 and 5.6: 1. for odd data samples:

$$\tilde{x}_{odd} = (n+1)/2 \tag{5.5}$$

2. for even data samples:

$$\tilde{x}_{even} = \frac{1}{2}(n/2 + n/2 + 1) \tag{5.6}$$

The sample median is more likely to lay in the middle of a data set, where as the mean search toward large values.

5.3 Stability of the period method

Calculation of the STD of the calculated frequencies gives an idea of how good the welding is, but there can be a number of problems with this method. If we look at the data collected from the 14. Dec. the details of the data can be seen in appendix A1. From these the table 5.1 is collected, the table shows the STD of the frequencies for each file.

File	STD of frequency	Median
a45	-	
a50	34.6	38.9
a55	49.6	44.2
a60	52.5	49.1
a65	48.9	53.2
a70	43.4	48.6
a75	35.7	58.3
a80	37.1	57.2

Table 5.1: STD of the frequency for the data found 14. Dec.

The median is calculated because it's used in the calculation of the three area described later. From the table it can be seen that the STD is lowest for the welding a50 and a75. a50 is shown in figure 5.1 and a70 in figure 5.2

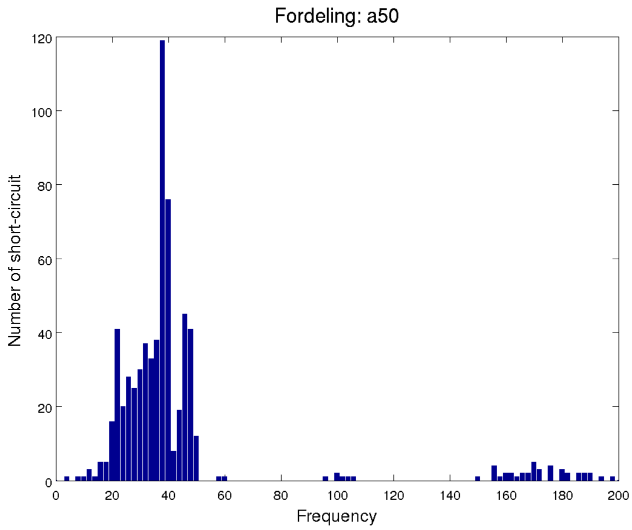


Figure 5.1: *Shows a histogram plot of welding a50 welding*

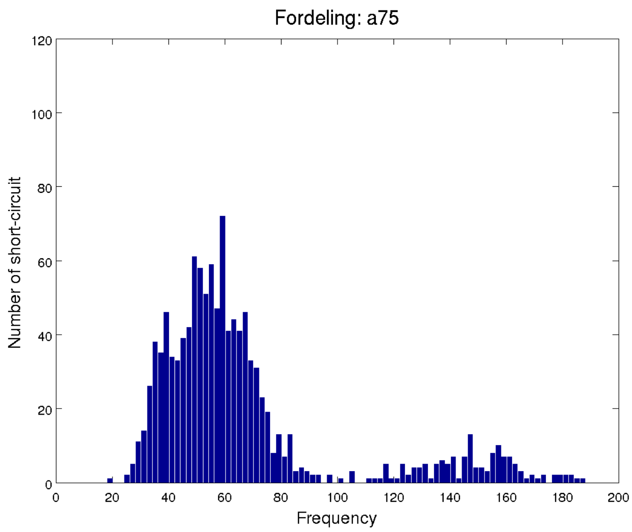


Figure 5.2: *Shows a histogram plot of welding a70 welding*

From this it can be seen that it looks like the a70 welding have a bit more frequencies spread out than the a50 welding. To find out which one is better than the other there will be looked at the median and the STD. If the short-circuit frequencies is split into three different areas, the one is the area laying between the median $\pm$ the $\frac{STD}{2}$, the two other areas is the ones on both sides of this. If we plot the histogram of the two again with the "main" peak area defined as median $\pm\frac{1}{2}$STD we get figure 5.3 and 5.4. The red line is the median and the black is the STD, from table 5.1. The median is chosen because the possibility for extreme values.

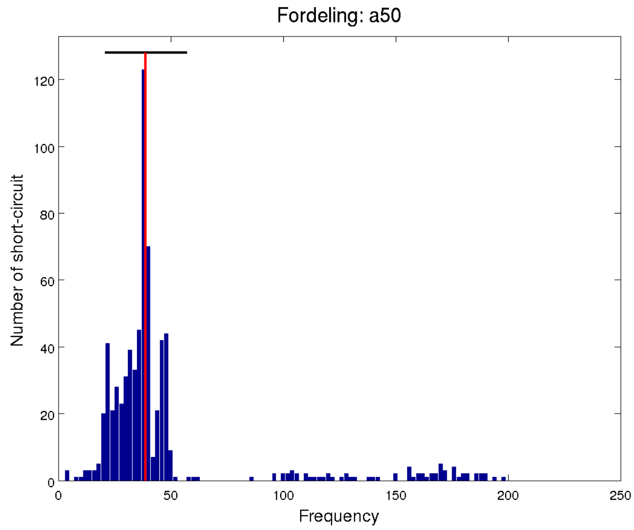


Figure 5.3: *Shows a histogram plot of welding a50 welding, with STD and median drawn*

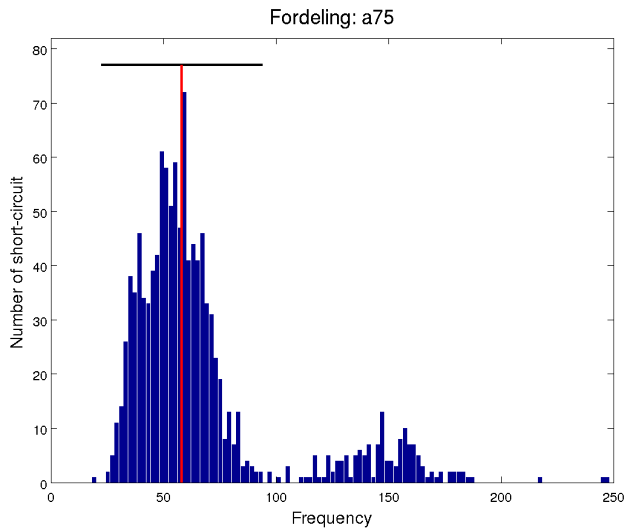


Figure 5.4: *Shows a histogram plot of welding a70 welding, with STD and median drawn*

From the figures it can be seen that we approximately have found the area where the main frequencies is located. Then we have the area under and the area above the middle area. The frequencies that are separated into three areas can now be analysed with the STD again. This method gives a more precise result of the STD and the middle area that has been pick out is with a good approximation the area with most frequencies. In table 5.2 the STD for the three different area is found for the welding made the 14. Dec.

File	STD, middle area	STD, low area	STD, high area
a45	-	-	-
a50	7.6	4.2	33.5
a55	8.2	4.7	37.1
a60	10.4	7.7	34.1
a65	11.3	5.5	30.8
a70	7.8	6.1	46.1
a75	9.3	3.5	38.1
a80	9.1	3.2	43.7

Table 5.2: STD of the frequency for the data found 14. Dec.

From table 5.1 there are three a50, a75 and a80 that are laying close together. From table 5.2 it is more a50, a55 and a70. To get a better criteria to tell which one is more stable than the other the percentage of the middle area have been calculated. The percentage is shown in table 5.3

File	Percentage
a45	
a50	88 %
a55	77%
a60	69 %
a65	67 %
a70	76 %
a75	67%
a80	72 %

Table 5.3: Percentage of the short-circuit frequencies located in the middle area

From this it is clear that the a50 is the best welding of the eight welding. But by looking at the bottom of the a50 welding shown in figure 5.5, it can be seen that the welding doesn't goes through the work piece.

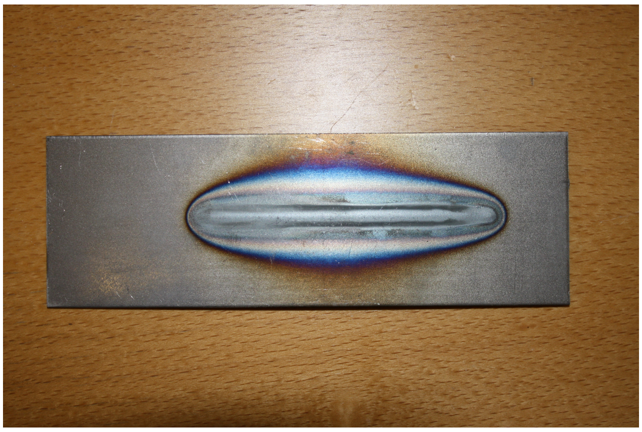


Figure 5.5: Welding bottom

This isn't optimal for a welding. The most optimal would probably be something between the a65_new, shown in figure 5.6, and a50.

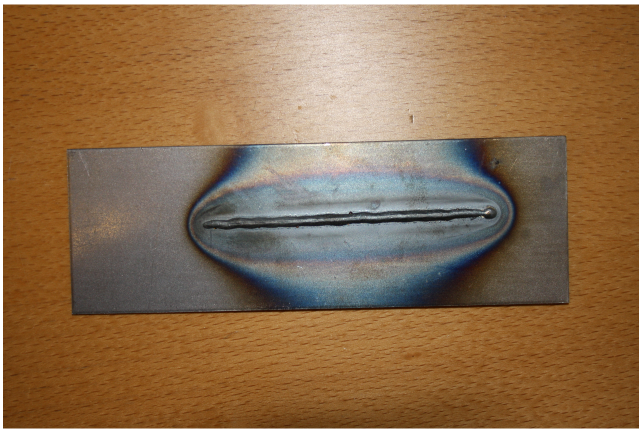


Figure 5.6: Welding bottom

Chapter 6

Conclusion and reflection

In this report the stability of welding is used to determine if one welding is better than another. To determine the stability of the welding there have been looked at the frequency of the short-circuit. Two method have been used the Welch-PSD and the period method. The Welch-PSD which is a Fourier transformation, calculates the discrete time signal from the time-domain to the frequency domain. The period method uses the times from the arc and short-circuit periods, and calculates the frequency with a simple equation, see equation 4.1.

The Welch-PSD and the period method was analysed in chapter 4. It was found that the Welch-PSD had problems with showing all the frequencies in the signal. This was seen in the difference between figure 4.17(a) and 4.17(b). In figure 4.17(a) the period method shows two areas with frequencies whereas the PSD in figure 4.17(b) only shows one area.

Test was conducted to see the influence of window size and overlap. The window size have influence on the frequency resolution, but the changes in this didn't gave any new information. The overlap changes didn't have any influence on the PSD at all, only few unimportant differences.

In chapter 4 some simple tesdt on a 100 Hz square signal was conducted. The test showed that the duty-cycle had a high influence on the PSD. A test with a square signal changing between 100 Hz and 200 Hz showed that the 200 Hz signal disappeared in the PSD and the two frequencies instead gave information in 33, 66 and 100 Hz.

In chapter 5 the stability of the short-circuit frequency is analysed. For the short-circuit analysis the period method is chosen. Because the short-circuit frequency changes under the welding and in many cases isn't laying in the same area, the frequency spectra had to be split into three areas.

The three areas are called, frequencies middle, frequencies less and frequencies larger. The middle area is the area with most short-circuit frequencies. The two other areas can change to be the one with most frequencies. The test from chapter 5 gave a wrong answer for the welding being the best. The one the analyse picked didn't burn through the metal at all.

In the thesis two method was used too find the short-circuit frequency, one failed because of incorrect answer in the sense that it hid some of the frequencies. The other method was used to determine the stability which was completed but the result didn't find the best welding.

Some analysis of the amplitude of the welding voltage and current could be performed to check if the have a influence on the stability of the process. This is what they do in the article (S. Adolfsson and Claesson, 1999). This project doesn't look exactly like the one from section 1.2. This is because they look at the stability of the process to determine if there is disturbance from exterior problem. Where this project have tried to tell two welding apart.

Appendiks A Samples

A1 Test på Migatronic

File: a45

file	a45
Program	P112
Current	45 a
Voltage	14.9
Wirespeed	1.5 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	1.6
end	24

File: a50

file	a50
Program	P112
Current	50 a
Voltage	15.2
Wirespeed	1.7 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	2.1
end	21.7

Figure 1: Welding top

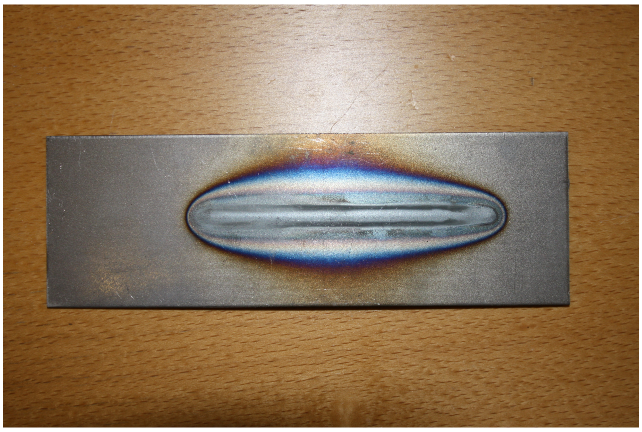


Figure 2: Welding bottom

File: a55

file	a55
Program	P112
Current	55 a
Voltage	15.6
Wirespeed	2.0 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	2.1
end	22.8

File: a60

file	a60
Program	P112
Current	60 a
Voltage	15.8
Wirespeed	2.2 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	1.6
end	20.4

File: a65

file	a65
Program	P112
Current	65 a
Voltage	16.1
Wirespeed	2.3 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	0
end	15.3

File: a65_new

file	a65_new
Program	P112
Current	65 a
Voltage	16.1
Wirespeed	2.3 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	1.6
end	22.3

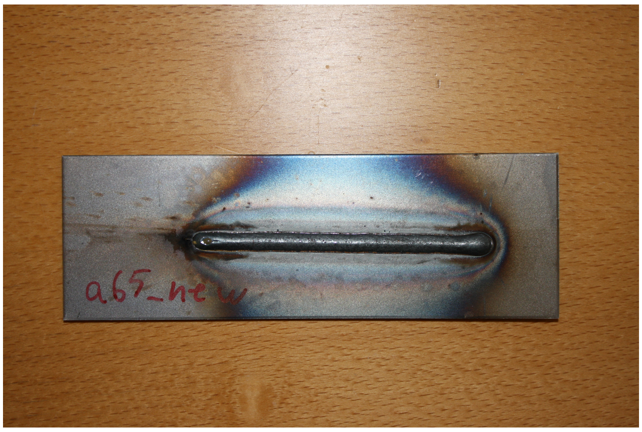


Figure 3: Welding top

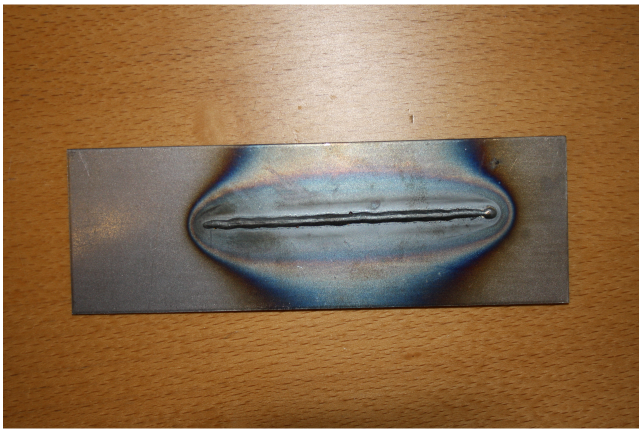


Figure 4: Welding bottom

The difference between a65 and a65_new is that 65_new have the beginning of the welding in the sampled data.

File: a70

file	a70
Program	P112
Current	70 a
Voltage	16.3
Wirespeed	2.5 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	1.6
end	20

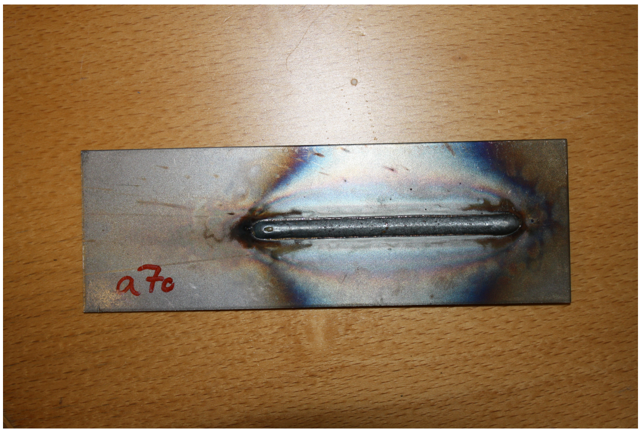


Figure 5: Welding top

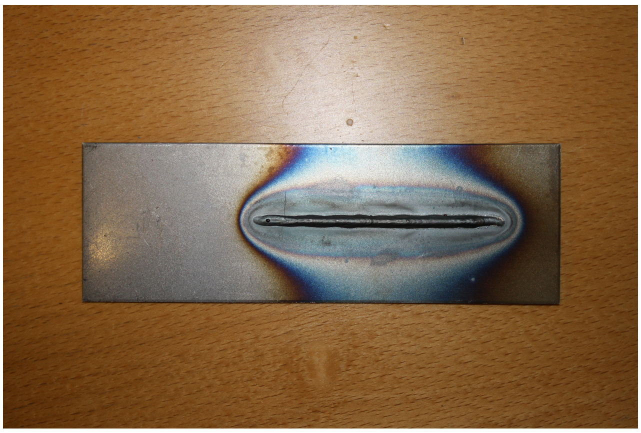


Figure 6: Welding bottom

File: a75

file	a75
Program	P112
Current	75 a
Voltage	16.5
Wirespeed	2.7 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	1.94
end	23.2

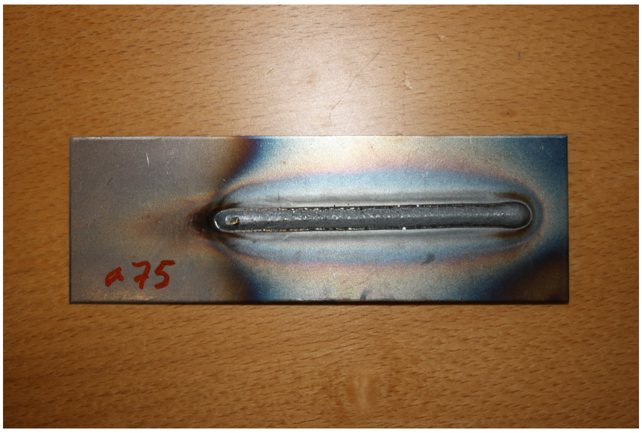


Figure 7: Welding top

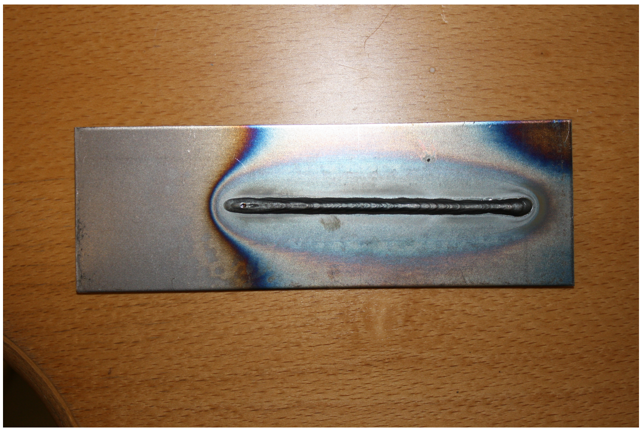


Figure 8: Welding bottom

File: a80

file	a80
Program	P112
Current	80 a
Voltage	16.8
Wirespeed	2.8 m/min
Plate	1.5 mm
Robotspeed	30 cm/m
begin	0.84
end	21.25

Appendiks B Description of Matlab code

B1 Description of the Matlab functions

The overall Matlab program takes a file and runs it through the six functions described in this appendix. In figure 9 the flow chart for main is shown.

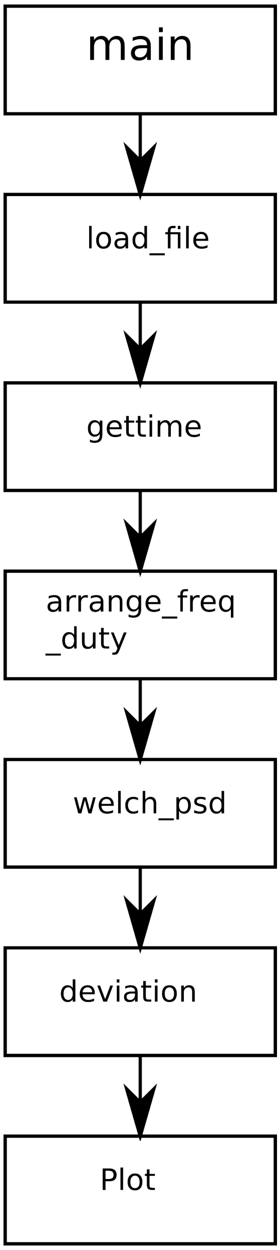


Figure 9: Float chart of main

The plot functions isn't shown in the report, but the are on the DVD in the folder "matlab".

B2 Loadfile

There has been made a function that loads the files made by the samplings-program used under each welding. The program used is described in section 3. First of all the files is checked if it exist, if not it's terminate the program. If the file exist the function load_file loads the file with the Matlab function *importdata*. In *Importdata* we decide how many header lines there is and what type of separator used. The header lines of the file gives information about the sampling. an example of the header file is shown in the table 1.

Scan Rate:	50000	-	-	-
Scans Per Frame	10	-	-	-
Total Scans	1000000	-	-	-
ZP:	999990	TP:	100	-
Number of Channels:	3	-	-	-
Scaled:	Yes	-	-	-
Scale Factor:	-	100	10	1
Offset:	-	0	0	0
Label:	-	Current	Voltage	ShortCircuit
Units:	-	A	V	bool
Index	Time	Ch.0	Ch. 1	Ch. 2

Table 1: Header file, found with *importdata*

In figure 10 is the float chart plotted for load_file.

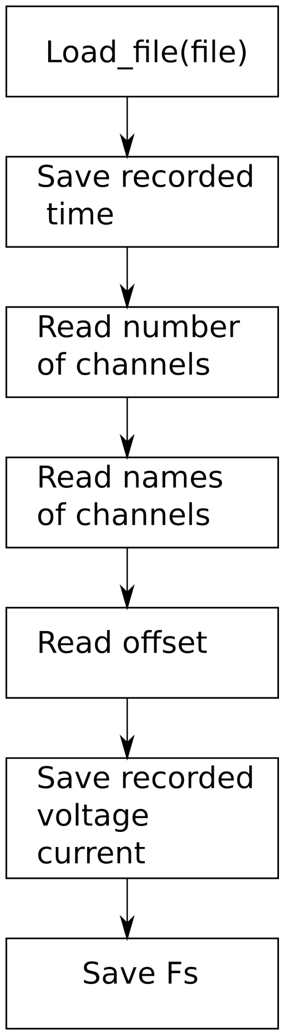


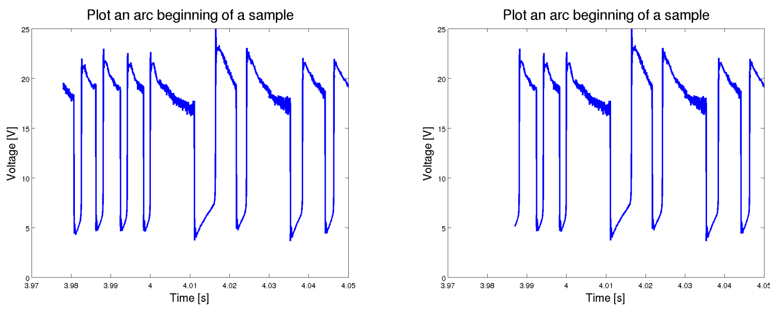
Figure 10: Float chart of load_file

First of all the program check how many channels there have been sampled on. This is done to decide how many variables there needs to be created. Then the program reads the label to get the names for the variables and it gets the offset which have been used under the sampling. The offset have to be deleted from the sampled data, the reason for the offset is that the plot when sampling the data is much easier to distinguish from each other.

From the header files the function load_file reads the Scan Rate, which is the sampling frequency and saves it in a global variable, "Fs".

B3 Gettime

The function **gettime** search the sampled data and finds the times for short-circuit start/stop and arc start/stop. First the function removes the first data, because of the change that the sampling first started in the middle of a welding. The two cases is shown in figure 11, in figure a the data has started recording in the middle of an arc period and in b in a short-circuit period.



(a) *Arc period* (b) *Short-circuit period*

Figure 11: *Shows the case of a sample beginning in either a short-circuit period or a arc period*

The time has to start at the beginning of a arc period to be sure it does this there have been made a loop to find a start index. The flow chart for this procedure is shown in figure 12.

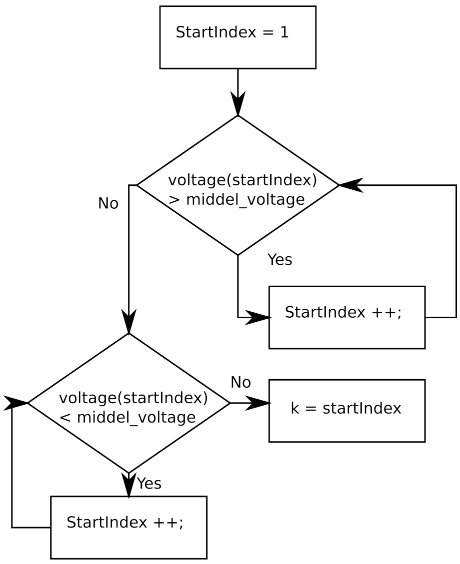


Figure 12: *Flow chart of the loop to find the startIndex*

When the start index is found the function goes through the sampled data saving each value of arc- and short-circuit start/stop. The flow chart in figure 13

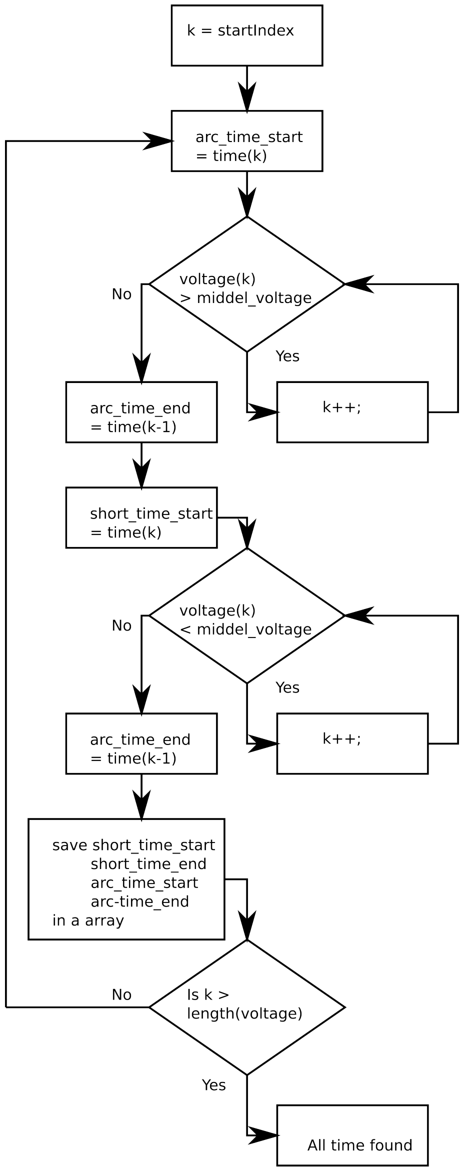


Figure 13: *Flow chart of the loop to find the individual times*

The flow chart in figure 13 only contains the times. The function also get information about the amplitude of the voltage and saves all the steps, k. The amplitude of the voltage is used when the times needs to be plotted so they can be plotted with the voltage signal. The source code is in appendix C3 on page C5.

B4 Frequency and duty cycle

The function **freq_duty** calculates the frequency and dutycycle from the times found with **gettime**. The function runs through all times for arc and short-circuit if the frequency is too high the function removes it. This is done if there can be an error in the recorded times. The frequency and duty-cycle is calculated with equation 4.1 and equation 4.2 described in section 4.1.

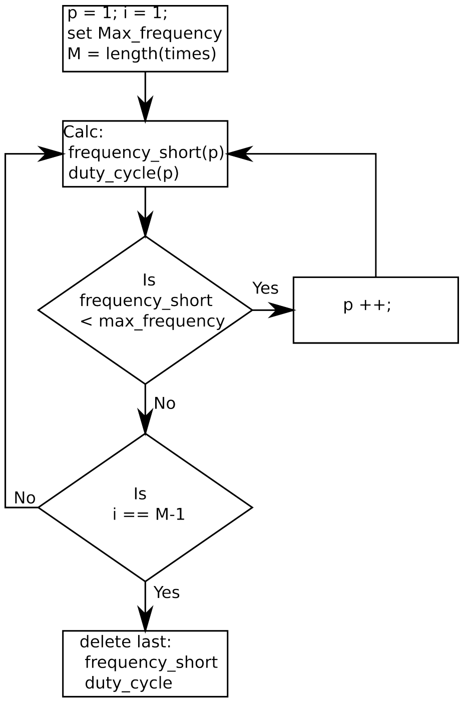


Figure 14: *Flow chart of the loop to find the short-circuit frequency and duty-cycle of these*

B5 Arrange frequency and duty cycle

This function save the frequencies calculated with **gettime** and **freq_duty** into blocks. The block are 2 Hz wide. The program is running through all the frequencies to see how many there is room for in the middle area first. The it calculates to new areas and check which one fits in these, and this is running until a counter reaches 200. When reaching 200 it will have moved 400 Hz to both sides and captured all the frequencies. When the frequencies have been places they can be plotted with the function *bar*, which then will plot a histogram.

B6 Frequency calculation with the Welch-PSD

This function is setting the window size and the overlap for the Welch-PSD. Then it use the function *pwelch*.

B7 Deviation

The function split the area of the frequency into three. The middle area is the one where most frequencies is. Then calculates the STD and mean of the three areas. At the end it saves the result in a .mat file. in figure 15 the flow chart is shown.

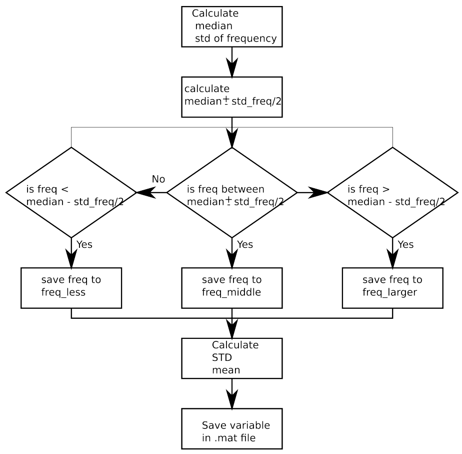


Figure 15: *Flow chart of deviation function*

Appendiks C Matlab-kode

C1 main

Listing 1: PSD udregningen

```
%% Start —— ) ( ——
clear all

format long
%% Global variables used/defined in the functions
%global file_loca2;
global file_loca;
global file_folder;
global file_name;
global getnumber1;
global getnumber2;


%global Fs;

% load files
global Voltage;
global Current;
%global diode;
global time;



% gettime
global time_arc;
global time_short;

%global time_short_end;global voltage_short_end;global step_short_end;
%global time_short_start;global voltage_short_start;global step_short_start;
%global time_arc_end;global voltage_arc_end;global step_arc_end;
%global time_arc_start;global voltage_arc_start;global step_arc_start;

% freq_duty
global frequency_short;
global duty_cycle;

% arrange_freq_duty
%global kort_low;global kort_middel;global kort_high;
%global duty;global duty1;global duty2;


% max_min_current
```

```
%global max_current_all;global max_current_step_all;
%global min_current_all;global min_current_step_all;

% frequncy_psd
%global f;
%global plotY;


%% Choose file
% file
%system = input('Ubuntu? Y/N:','s');
system = 'Y';
if system == 'Y'
    %file_loca = 'Documents/uni/p7/matlab/maalinger/26juli/';
    %file_loca = 'Documents/uni/p7/matlab/maalinger/9sep/';
   %file_loca = 'Documents/uni/p7/matlab/maalinger/14dec/';
    file_loca = 'Documents/uni/p7/matlab/maalinger/';
else
    file_loca = 'C:\Users\nysted\Documents\uni\maalinger\maalinger\';
end
% File to import


file_folder = '9sep/';

find_file(file_folder)

load_numbers(file_folder,file_name);

file_ext = '.au2';

file_full = [file_loca file_folder file_name file_ext];

exist_file = exist(file_full,'file');



%% The new test program - load of files
if exist_file == 2
    load_file(file_full,file_folder);
else
    error('File does not exist')
end


%% Current - Voltage flipper
%If the current and voltage was sampled in the wrong channel use this
% script makes sure it uses the right factors
voltage = Voltage;
current = Current;



voltage = voltage(getnumber1:getnumber2);
time = time(getnumber1:getnumber2);
current = current(getnumber1:getnumber2);
voltage1 = voltage;
current1 = current;


%% Functions

gettime(voltage,time);
```

```

freq_duty(time_arc,time_short)

%arrange_information(frekvens_kort,duty_cycle)

arrange_freq_duty(frequency_short,duty_cycle)
%arrange_freq_duty1(frekvens_kort1,duty_cycle1)
%max_min_current(current,voltage,time)

frequency_psd_welch(voltage)

frequency_psd(voltage,0)


deviation(file_name)



%getnumber1 = int2str(getnumber1);
%getnumber2 = int2str(getnumber2);
%% Plot

%plot_psd_period(voltage)

%plot_voltage_beginning(voltage,file_name)

%plot_testtime(time1,voltage,file_name)

%plot_arrange_freq_duty(file_name)

%plot_voltage_current(time1,voltage1,current1,file_name)

%plot_voltage_current_time(time1,voltage1,current1,file_name)

%plot_frekvens_boxplot(file_name,system)

%plot_frequency(file_name,0)

%plot_frekvensen(file_name,0)

%plot_welch(voltage,file_name)
%plot_welch_window(voltage,file_name)
%plot_welch_overlap(voltage,file_name)
%plot_psd(file_name,system,0)
%clear f;
%clear plotY;
%frequency_psd(voltage,1)
%plot_psd(file_name,system,2)

%plot_drawing(file_name)

format short;
```

C2 Load file

Listing 2: PSD udregningen

```
function load_file(file_full,folder)
```

```
% load file , importdata imports the data , the au2 file has 11 header lines
% and it 's seperated with tabular (\t)

global Voltage;
global Current;
global diode;
global time;
global Fs;


import = importdata([file_full],'\t',11);
% Because the program samples the data with a minus sign , the data has to
% be flipped and it has to be the abs value
time = abs(flipud(import.data(1:end,2)));

[~, remain] = strtok(import.textdata{5},'Number of Channels');
channels = remain;
remain = str2double(remain);

if remain == 1 % If the data only contains one data set read what it is and
    save the data under that name so far (Voltage , Current or diode , remember
    the large char)
    [~, remain] = strtok(import.textdata{9},char(9));
    varname = genvarname(remain);
    eval([varname ' = import.data(1:end,3);'])
else if remain == 2

    % Get the factors from the samplings
    [~, remain] = strtok(import.textdata(8),char(9)); % removes the first text ,
        and place the important text in remain
    [factor1 remain] = strtok(remain,char(9));
    [factor2] = strtok(remain,char(9));
    factor1 = str2double(factor1{1}); % saves the first factor in factor1
    factor2 = str2double(factor2{1}); % saves the second factor in factor2

    [~, remain] = strtok(import.textdata{9},char(9));
    [name1, name2] = strtok(remain,char(9));
    name1 = genvarname(name1);
    name2 = genvarname(name2);

    eval([name1 ' = import.data(1:end,3) - factor1;'])
    eval([name2 ' = import.data(1:end,4) - factor2;'])
    else if remain == 3

    % Get the factors from the samplings
    [~, remain] = strtok(import.textdata(8),char(9)); % removes the first text ,
        and place the important text in remain
    [offset1 remain] = strtok(remain,char(9));
    [offset2 remain] = strtok(remain,char(9));
    [offset3] = strtok(remain,char(9));
    offset1 = str2double(offset1{1}); % saves the first factor in offset1
    offset2 = str2double(offset2{1}); % saves the second factor in offset2
    offset3 = str2double(offset3{1}); % saves the third factor in offset2


    % Get the names from the labels in the textfile and save them as the
    % names in the textfile
    [~, remain] = strtok(import.textdata{9},char(9));
    [name1, name2] = strtok(remain,char(9));
    [name2, name3] = strtok(name2,char(9));
    name1 = genvarname(name1);
    name2 = genvarname(name2);
    name3 = genvarname(name3);
```

```

    eval([name1 ' = import.data(1:end,3) - offset1;'])
    eval([name2 ' = import.data(1:end,4) - offset2;'])
    eval([name3 ' = import.data(1:end,5) - offset3;'])
        else if remain >= 4
                disp(['Make your own script here,' channels ' channels have
                    been used'])
                return
            end
        end
    end
end
% Remember to check the data after it's loaded, do you get the expectet
% data? You can check text.import to see what the eval have called the
% files normaly in remain == 2 or 3 the first data is called Current and
% the second Voltage. It's deceided when you sample the data.

% Takes the sampling frequency from the importet file
[token] = strtok(import.textdata(1),'Scan Rate:');
Fs = str2double(token{1}); % samplingfrekvensen

switch folder
    case '9sep/'
        current = Voltage;
        Voltage = Current/10;
        Current = current*-10;
    case '15sep/'
        current = Voltage;
        Voltage = Current/10;
        Current = current*-10;
    otherwise
        display('Nothing done')
end
```

C3 Gettime

Listing 3: PSD udregningen

```
function gettime(voltage,time1)
% This function finds the times for short- and arc start/stop, it takes
% two input, the first input is the voltage and the second the time

% Global variables

global time_arc;
global time_short;
global time_short_end;global voltage_short_end;global step_short_end;
global time_short_start;global voltage_short_start;global step_short_start;
global time_arc_end;global voltage_arc_end;global step_arc_end;
global time_arc_start;global voltage_arc_start;global step_arc_start;

% other stuff

voltage1 = voltage;
arc_voltage = 10; % Arc voltage is the voltage to split between arc and short-
    circuit
%arc_voltage_low = arc_voltage - std(voltage1);
%arc_voltage_high = arc_voltage + std(voltage1);
```

```
N = length(voltage1); % calculatets the length of the vector voltage1

% For fil 1
% constants
i = 1;k = 1;

% Removes the first short-circuit
if voltage1(1) > arc_voltage % check if it start top (if so removes the first
    top and bottum)
    while voltage1(k) > arc_voltage
        k = k + 1;
    end
    while voltage1(k) < arc_voltage
        k = k + 1;
    end
else % else it removes the first bottum
    while voltage1(k) < arc_voltage
        k = k + 1;
    end
end

% Finds the time for all short-circuits

while k <= N
    %if voltage(k) > arc_voltage

    time_arc_start1 = time1(k); % starting value for the arc time
    diode_vol_arc_start1 = voltage1(k); % starting value for the arc voltage
    voltage_arc_start1 = voltage(k);
    step_arc_start1 = k;
    while k <= N && voltage1(k) > arc_voltage % goes thrue the values of the
        arc period til it's finished
        k = k + 1;
    end
    time_arc_end1 = time1(k-1); % save the last value of the arc-periods time
    diode_vol_arc_end1 = voltage1(k-1); % save the last value of the arc-
        periods voltage
    voltage_arc_end1 = voltage(k-1);
    step_arc_end1 = k-1;
    if k <= N
        while voltage1(k) > arc_voltage
            k = k+1;
        end
        time_short_start1 = time1(k); % saves the start of the short-circuit
            time
        diode_vol_short_start1 = voltage1(k); % saves the start of the short-
            circuit voltage
        voltage_short_start1 = voltage(k);
        step_short_start1 = k;
        while k <= N && voltage1(k) < arc_voltage % goes thrue the values of
            the short-circuit period till it's finished
            k = k + 1;
        end
        time_short_end1 = time1(k - 1);% saves the end of the short-circuit
            time
        diode_vol_short_end1 = voltage1(k-1);% saves the end of the short-
            circuit voltage
        voltage_short_end1 = voltage(k-1);
        step_short_end1 = k-1;
    end
    %Saves the datapoints for calculating the arc and short periods, used
    %to check if it finds the right time stamps and amplitude of the
```

```
    %voltage
    time_short_end(i) = time_short_end1;voltage_short_end(i) = voltage_short_end1;step_short_end(i) = step_short_end1;
    time_short_start(i) = time_short_start1;voltage_short_start(i) = voltage_short_start1;step_short_start(i) = step_short_start1;
    time_arc_end(i) = time_arc_end1;voltage_arc_end(i) = voltage_arc_end1;step_arc_end(i) = step_arc_end1;
    time_arc_start(i) = time_arc_start1;voltage_arc_start(i) = voltage_arc_start1;step_arc_start(i) = step_arc_start1;

    time_arc(i) = time_arc_end1 - time_arc_start1; % calculates the time of the arc period

    time_short(i) = time_short_end1 - time_short_start1; % calculates the time of the short-circuit period
%    clear time_arc_end; clear time_arc_start; clear time_short_end; clear time_short_start;
    i = i + 1;
%    isglobal(time_short1)
end
```

C4 Frekvens og dutycycle

Listing 4: PSD udregningen

```
function freq_duty(arc,short)
% Finds the frequency for all the shortcircuit periods, it takes the time
% when in arc- and the time when in short mode

% global
global frequency_short;
global duty_cycle;

M = length(arc); % calculate the length of the arc time it's the same as the length og the short-circuit
p = 1; % constant start at one because of matlab
max_frequency = 300; % freqency which we are not intereste to get any above
% calculation of frequency and dutycycle
for i = 1:M-1 % M - 1 removes the last data, because of the chance of stopping in the middel of a period
    frequency_short(p) = 1/(arc(i)+short(i));
    duty_cycle(p) = arc(i)/(arc(i)+short(i));
    if frequency_short(p) < max_frequency % If the frequency is larger than max remove it
        p = p+1;
    else if frequency_short(p) > max_frequency && i == M-1 % If the last frequency is larger than max delete it.
            p = p-1;
            frequency_short = frequency_short(1:p);
            duty_cycle = duty_cycle1(1:p);
        end
    end
end
```

C5 Arrange freqency and duty-cycle

Listing 5: Arrange freqency and duty-cycle

```
function arrange_freq_duty(frekvens_kort1,duty_cycle1)
% Seperates dutycycle and freqency into diffrent areas.

% arrange_freq_duty
global kort_low;
global kort_middel;
global kort_high;

global duty;
global duty1;
global duty2;

global ti_together;
global steps;


%sample mean, deviation and median
sample_mean = mean(frekvens_kort1);
deviation = std(frekvens_kort1);
sample_median = median(frekvens_kort1);
% Step calculations
step_for_frequency = 1:length(frekvens_kort1);
step_deviation = int8(deviation/5);
%k = 0;
t = 1;t1 = 1;t2 = 1; % constants
i1 = length(frekvens_kort1);

step_dev = double(step_deviation);
step_dev = 200;
%for i = 1:step_dev
%    k = k + 1;
%end
    dev_low = 0;
    dev_low1 = 0;
    dev_high = 0;
    dev_high1 = 0;
for k = 0:step_dev
dev_low = sample_median - (1 + 2*k);
%dev_low1 = deviation - (2.5 + 5*(k +1));
dev_high = (sample_median + 1 + 2*k);
%dev_high1 = deviation + (2.5 + 5*(step_deviation -1));
    for i = 1:i1
        if dev_low1 == 0 && frekvens_kort1(i) > dev_low && frekvens_kort1(i) < dev_high % the first interval +- 2.5
        kort_middel(t) = frekvens_kort1(i);
        step_middel(t) = step_for_frequency(i);
        duty(t) = duty_cycle1(i);
        t = t+1;
        else if frekvens_kort1(i) < dev_low1 && frekvens_kort1(i) > dev_low && dev_low > 0% takes the low intervals +- 5 * k
            kort_low(t1) = frekvens_kort1(i);
            step_low(t1) = step_for_frequency(i);
            duty1(t1) = duty_cycle1(i);
            t1 = t1 +1;
            else if dev_low1 ~= 0 && frekvens_kort1(i) > dev_high1 && frekvens_kort1(i) < dev_high % takes the high intervals +- 5 * k
                kort_high(t2) = frekvens_kort1(i);
```

```
                step_high(t2) = step_for_frequency(i);
                duty2(t2) = duty_cycle1(i);
                t2 = t2 +1;
                end
            end
        end

    end
% Saves the number of number in each interval so the intervals can be
% seperatet
ti_middel(k+1) = t; % middel area
ti_low(k+1) = t1; % low area
ti_high(k+1) = t2; % high area
%step_deviation = step_deviation -1;
% saves the old values before next run
dev_low1 = dev_low;
dev_high1 = dev_high;
end

kort_low1 = flipdim(kort_low,2); % flips the data around so it looks better
    when plottet
ti_middel1 = ti_middel(1)-1;
M = length(ti_high);
k = 1;
for i = 1:M-1
    ti_low1(k) = ti_low(i+1)-ti_low(i);
    ti_high1(k) = ti_high(i+1)-ti_high(i);
    k = k+1;
end
ti_low1 = flipdim(ti_low1,2);
ti_together = [ti_low1 ti_middel1 ti_high1];

i = 1;

for k = 0:step_dev
    step1(i) = sample_median - 1 - 2*k;
    step2(i) = sample_median + 1 + 2*k;
    i = i +1;
end

steps = [flipdim(step1,2) step2(1:end-1)];

% calculate mean and standard deviation
mean_kort = mean([kort_low1 kort_middel kort_high]);
std_kort = std([kort_low1 kort_middel kort_high]);
mean_kort1 = std([kort_low1(21:end) kort_middel kort_high(1:end-8)]);
std_kort1 = mean([kort_low1(21:end) kort_middel kort_high(1:end-8)]);

% constants used in the next calculations
M = length(frekvens_kort1); % length calculation
% sample_median or sample_mean?
middel_frekvens = sample_median - deviation; % calculate the frequency to
    determine the area for which information we want.
maks_frekvens1 = sample_median + deviation; % It takes the median and get the
    middle area to be between this and the deviation
t = 1;
%i = 1;
k=1;

% Seperates the frequencies in three, one less than middel_frekvens and one
% between middel_frekvens and maks_frekvens1 and the last above
% maks_frekvens1
for i = 1:M
```

```
    if frekvens_kort1(i) > middel_frekvens && frekvens_kort1(i) <
        maks_frekvens1
        frekvens_kort1_1(t) = frekvens_kort1(i);
        t = t +1;
    else if frekvens_kort1(i) < middel_frekvens
            frekvens_kort1_2(k) = frekvens_kort1(i);
            k = k +1;
        end
    end
end
```

C6 Welch-PSD

Listing 6: Welch-PSD calculation

```
function frequency_psd_welch(voltage)
% PSD udregninger removes the mean value from the voltage signal so the
% signal is calculatet around 0.

%global
global fw;
%global plotY;
global Fs;
global pxx;


window = Fs;
noverlap = Fs/2;
nfft = length(voltage);


voltage_mean = mean(voltage);
voltage = voltage - voltage_mean;

[pxx,fw] = pwelch(voltage,window,noverlap,[],Fs,'onesided');
```

C7 Deviation

Listing 7: Deviation

```
function deviation(name)

global frequency_short;
global time_arc;
global time_short;
global duty_cycle;
global file_loca;
global file_folder;
global file_name;

median_freq = median(frequency_short);

std_frequency = std(frequency_short);
```

```
%std_time_arc = std(time_arc);
%std_time_short = std(time_short);
%std_time_shortarc = std(time_short+time_arc);
mean_freq = mean(frequency_short);
%mean_time_arc = mean(time_arc);
%mean_time_short = mean(time_short);
%mean_time_shortarc = mean(time_short+time_arc);
%var_time_short = var(time_short1);
%var_time_arc = var(time_arc1);

l = length(frequency_short);
t = 1;g = 1;p = 1;q = 1;
frequency_less = 0;
frequency_larger = 0;
for i = 1:l
if frequency_short(i) <= median_freq + std_frequency && frequency_short(i) >=
    median_freq - std_frequency
 frequency_middle(t) = frequency_short(i);
 t = t+1;
%else if frequency_short(i) > median_freq && frequency_short(i) <= median_freq
    + mean_freq/2
 %   frequency_larger_median(g) = frequency_short(i);
  %  g = g+1;
    else if frequency_short(i) > median_freq + std_frequency
            frequency_larger(p) = frequency_short(i);
            p = p+1;
        else if frequency_short(i) < median_freq - std_frequency
            frequency_less(q) = frequency_short(i);
            q = q+1;
            end
      %  end
    end
end
end
std_middle = std(frequency_middle);
mean_middle = mean(frequency_middle);
std_larger = std(frequency_larger);
mean_larger = mean(frequency_larger);
std_less = std(frequency_less);
mean_less = mean(frequency_less);

mid_area_procent = 100*length(frequency_middle)/length(frequency_short);

save([file_loca file_folder file_name '.mat'],'median_freq','frequency_less','
    frequency_larger','frequency_middle','std_middle','std_larger','std_less',
    'mean_middle','mean_larger','mean_less','duty_cycle','frequency_short','
    std_frequency','mid_area_procent');%'std_time_arc','std_time_short','
    std_time_shortarc','mean_freq')
```

Bibliography

Cary, Howard B. 1989. *Modern Welding Technology*. Prentice Hall, Inc.

Chu, Y.X., S.J. Hu, W.K. Hou, P.C. Wang and S.P. Marin. 2004. "Signature Analysis for Quality Monitoring in Short-Circuit GMAW." *Welding Journal* 83(12).

electric, Lincoln. 2006. "GMAW welding guide." *Lincoln eletric* . lincolnelectric.com.

Hermans, J.M. and G. Den Quden. 1999. "Proces Behavior and Stability in Short Circuit Gas Metal Arc Welding." *Welding Journal* 78(04).

S. Adolfsson, A. Bahrami, G. Bolsjö and I. Claesson. 1999. "On-Line Quality Monitoring in Short-Circuit Gas Metal Arc Welding." *Welding Journal* 78(02).

Welch, Peter D. 1967. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms." *IEEE Trans. Audio and Electroacoust.* . https://cs.utd.edu/ cpb021000/EE%204361/Great%20DSPiodogram.pdf.