Master's Thesis

Prototype for Measurement Tools for Evaluating the Crusher Feed using Digital Signal Processing

Akram Joumaa

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, 2016.

Prototype for Measurement Tools for Evaluating the Crusher Feed using Digital Signal Processing

Akram Joumaa
elt11ajo@student.lu.se

Department of Electrical and Information Technology
Lund University

Advisor: Mikael Swartling, Lund University
Per Svedensten, Sandvik SRP AB

September 27, 2016

Printed in Sweden
E-huset, Lund, 2016

Abstract

Nowadays, the use of digital signal processing is essential in various measurements of different machines. There has been many projects done at Sandvik AB on how to collect and measure data signals on the cone rock crusher. None of those measurements could present the correct data. The reason could possibly be due to the lack of theory behind the use of digital signal processing which this thesis will focus mostly on.

In this thesis, a complete hardware prototype is constructed to measure the data signals of the cone crusher. The uneven distribution of the materials in the crushing chamber, results in uneven load which affects the machine performance negatively. Therefore, the signals obtained are analyzed and the results will be presented in real time mode using Matlab to investigate the cone crusher performance.

"Life is like riding a bicycle. To keep your balance, you must keep moving"
- Albert Einstein

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Mikael Swartling and my examiner Dr. Nedelko Grbic for all the useful remarks and encouragement through the whole writing and investigating process.

I would also like to thank my advisor at Sandvik AB Dr. Per Svedensten for his continuous support, guidance and encouragement. Also, special thanks to all the staff at Research and Development department at Sandvik. Most of all, I want to thank my loved ones, my family and my beautiful fiancee for providing me endless love and support by standing with me days and nights. Thank you.

Akram Joumaa

Table of Contents

List of Figures

List of Tables

Abbreviations

AD	Analog-to-Digital Converter
AGC	Adaptive Gain Control
CC	Cone Crusher
DA	Digital-to-Analog Converter
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
LMS	Least Mean Squares
LP	Loop Filter
LPLL	Linear Phase Locked Loop
MEMS	Micro Electrical-Mechanical System
OP	Operational Amplifier
PD	Phase Detector
PID	Proportional-Integral-Derivative Controller
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation
QD	Quadrature Detector
QPLL	Quadrature Phase Locked Loop
RF	Radio Frequency
TIA	Transimpedance Amplifier
VCO	Voltage Controlled Oscillator

Chapter 1

Introduction

1.1 Background

The cone crusher is one of the most common type of crusher in crushing plants where the goal is to obtain medium to severe crushed rock. One of the most common challenges of running a cone crusher is to get an even distribution of the materials in the crushing chamber. It is common to result in imbalance feeding arrangment which means that some materials ends up in different sizes due to the different parts of the feed. This leads to uneven load or distribution on the crusher and different degree of the wear parts.

The uneven wear reduces utilization and impacts the crusher performance negatively. To counteract this, a tool needs to be developed that can be used in the field to measure and analyze the feeding conditions of the crusher. The measurement consists of the hydraulic pressure measurement value of the cone crusher, which is directly proportional to the crushing force, and the measurement of the position for the cone crusher. The position is measured by using a 2-axis accelerometer. By combining these two measurements, we can clearly identify the location of the crushing forces and thus be able to determine when the crusher is fed well. This method is called feed tracking.

1.2 Aims and Challenges

The aim for this thesis is to develop a prototype to a handheld analysis tool that can be connected to a crusher and used to obtain pressure measurements. The prototype should be in any appropriate manner presenting "live" data so during the operation, it should observe how the measurement is changing. Moreover, a method for presenting data has to be developed.

Figure 1.1: The cross section of a cone crusher (Sandvik).

1.3 Cone Crusher

The cone crusher is fed through the top of the crusher and flows over the mantle. The vertical main shaft rotates at 300 rpm the mantle below the concave, or bowl liner, squeezing the product and crushing it between the concave and the mantle. Cone crushers are usually run on belt drives driven by an electric motor and are used in the aggregate and mineral processing industry (cf. Fig. 1.1). The advantages of the cone crusher are many, a few of them are: hydraulic clearing and adjustments, high productivity, long life, cost effectiveness, standard replacement parts etc.

Sandvik SRP AB in Svedala has a test facility, where the cone rock crushers are tested. One such facility is located in Dalby, where our prototype was constructed for the cone crusher according to Fig 1.2. A picture is taken inside the cone crusher where the big rocks converts into small rocks (cf. Fig. 1.3).

Figure 1.2: Sandvik's test facility at Dalby Quarry.

Figure 1.3: Inside the cone crusher where the rocks are breaking down.

1.4 Sandvik AB

Sandvik AB is a global industrial group with advanced products and world-leading positions in selected areas-tools for metal cutting, equipment and tools for the mining and construction industries, stainless materials, special alloys, metallic and ceramic resistance materials and process systems. In 2015 the Sandvik Group had about 47,000 employees and representation in 130 countries, with annual sales of more than 90,000 MSEK.

Sandvik Construction is a business area within the Sandvik Group providing solutions for virtually any construction industry application encompassing such diverse businesses as surface rock quarrying, tunneling, excavation, demolition, road building, recycling and civil engineering. The range of products includes rock tools, drilling rigs, breakers, bulk-materials handling and crushing and screening machinery.

1.5 Thesis Outline

Chapter 2 gives a brief introduction of the theory that is needed to understand the thesis.

Chapter 3 describes the definition of the phase locked loop for one input signal.

Chapter 4 gives information about the main algorithm of the problem using the quadrature phase locked loop.

Chapter 5 describes the work performed when choosing the accelerometer and how the real time algorithms were implemented.

Chapter 6 presents the achieved results for this thesis work.

Chapter 7 presents discussions and summaries of the result from this thesis work and proposes improvements and future work.

Chapter 2

Theory

2.1 Accelerometer

An accelerometer is an electromechanical device that is used to calculate and measure forces and the accelerations acting upon it. There are two types of accelerometers, the most common one is the piezoelectric accelerometer. The problem with those accelerometers is that they are bulky and don't have the ability to run on all big operations such as having highly sensing feature to obtain multi-axis sensing[5]. Therefore, another device was developed like the MEMS accelerometer.

The basic principle of operation behind a micro-electrical-mechanical accelerometer or (MEMS) is the displacement of a small proof mass etched into silicon surface of integrated circuit and suspended by small beams [4]. In the other words, the Newton's second law of motion is applied according to F equals m times a, where a is the acceleration and displaces the mass m which produces a force F. Using Hook's law, the spring shows a restoring force F_s which is proportional to the displacement x as described by

$$F_S = k_S x, \tag{2.1}$$

where k_S is the spring constant. Neglecting the air friction and from the Newton's second law of motion the acceleration can be presented as a function of the displacement

$$a = \frac{k_S}{m} x. \tag{2.2}$$

The MEMS accelerometers are using changes in capacitance while sensing [13]. Capacitive sensing relies on the variation of capacitance and the geometry of the capacitor while it is changing. The parallel-plate capacitance can be expressed

$$c_0 = \epsilon_0 \epsilon_r \frac{A}{d} = \epsilon \frac{1}{d}, \tag{2.3}$$

where $\epsilon = \epsilon_0 \epsilon_r A$, d is the distance between the plates, ϵ_r the permittivity of the material and A the area of the electrodes. Any change of those parameters will cause change of capacitance. For instance, if a humidity sensor is needed the change of ϵ_r is based on a change in A and d. In the MEMS accelerometer a movable proof mass with plates which represent capacitors, and is affiliated through a mechanical suspension system to a reference frame. The capacitance difference is using to measure the aberration of the proof mass. The capacitances between the movable plate and two stationary outer plates C_1 and C_2 are functions of the corresponding displacements x_1 and x_2 [14]:

$$\begin{aligned} C_1 &= \epsilon\frac{1}{x_1} = \epsilon\frac{1}{d+x_0} = C_0 - \Delta C \\ C_2 &= \epsilon\frac{1}{x_2} = \epsilon\frac{1}{d-x_0} = C_0 + \Delta C \end{aligned} \tag{2.4}$$

If x is not equal to zero, the capacitance difference can be found as in Equation2.5. If the acceleration is zero, the capacitances C_1 and C_2 are then equal because x_1 and x_2 are equal:

$$C_2 - C_1 = 2\Delta C = 2\epsilon\frac{x}{d^2 - x^2} \tag{2.5}$$

If ΔC is measured, the displacement x can be expressed as a nonlinear algebraic equation:

$$\Delta C x^2 + \epsilon x - \Delta C d^2 = 0. \tag{2.6}$$

For very small displacements, the term Δx^2 is negligible and can be omitted. The final equation is thus,

$$x \approx \frac{d^2}{\epsilon}\Delta C = d\frac{\Delta C}{C_0}. \tag{2.7}$$

Figure 2.1 shows the accelerometer structure where the voltage output V_x is calculated as the voltage of the proof mass using a combination of the equations 2.4 and 2.7. The voltage output is then

$$V_x = V_0\frac{C_0 - C_1}{C_2 + C_1} = \frac{x}{d}V_0, \tag{2.8}$$

where V_x is a square wave with an amplitude proportional to acceleration coming from a simple voltage divider through a buffer and demodulator as showing in Fig. 2.2. In addition, an accelerometer with 46 pairs of sensors are driving by 1 MHz square waves with voltage amplitude V_0 coming out of oscillator[10]. The phases between upper and lower fixed plates differs for 180°. Using the equations 3.31 and 2.2, the acceleration is found to be proportional to voltage output as

$$a = \frac{k_s d}{m V_0} V_x. \tag{2.9}$$

The parameters k_S, d, m, V_x and V_0 are chosen depending on what kind of MEMS accelerometer is used. For example a ±1.5 g accelerometer can be used in gravity measurements, a ±2 g can be used to measure the motion of a car, a ±16 g can be used for shock and vibration, such as in a cone crusher. The frequency of the oscillator has to be larger than the bandwidth frequency, because the electronic circuit has to read changes in capacitance quicker than the acceleration changes and the demodulator needs some number of cycles before it calculates the output [10].

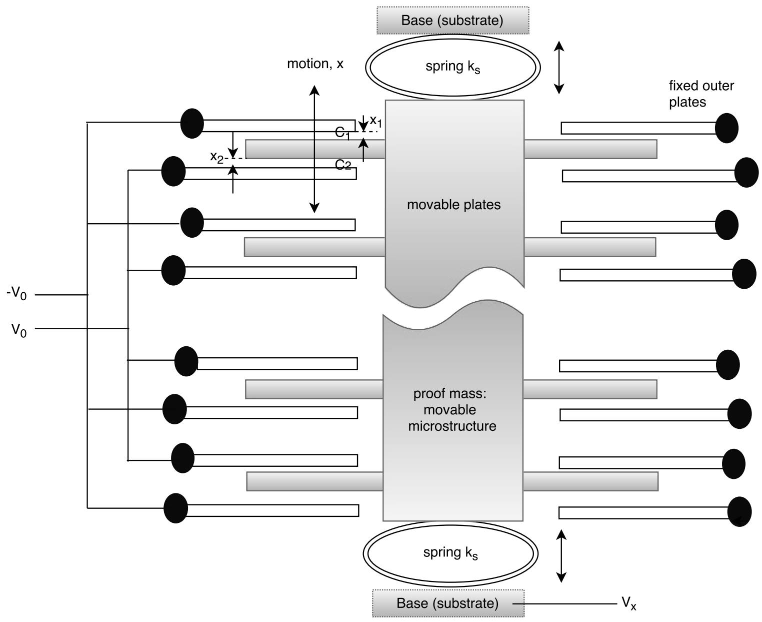

Figure 2.1: The structure of the MEMS accelerometer which is used in this thesis.

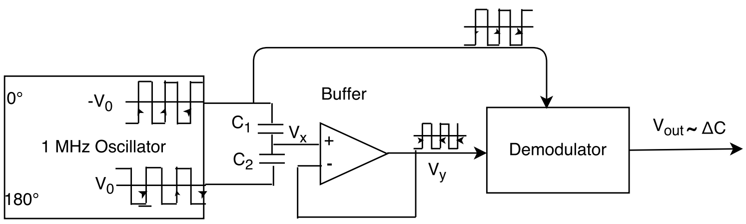

Figure 2.2: An electric circuit measuring the acceleration through capacitor changes.

2.2 Filtering

Filtering can be used to remove unwanted frequencies in signals. Filtering is a class of signal processing, the defining feature of filters being the complete suppression of some aspect of the signal. Most often, it is removing some frequencies and not others in order to suppress interfering signals which reduces the background noise.

2.2.1 Butterworth Filters

The Butterworth filter is an optimal filter with maximally flat response in the passband. The magnitude of the frequency of this family of filters can be written as

$$|H(f)| = \frac{H_0}{\sqrt{1 + (\frac{f}{f_c})^{2n}}}, \tag{2.10}$$

where n is the order of the filter and f_c is the cutoff frequency, H_0 is the gain at zero frequency. The following Fig. 2.3 is illustrating the combination of the cutoff frequency and the order of filter.

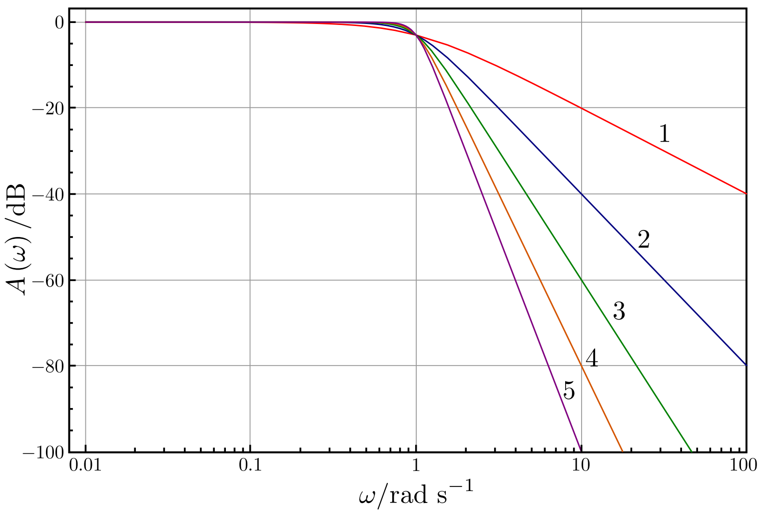

Figure 2.3: The cutoff frequency and the order of filter [17].

2.3 Adaptive Gain Control

An Adaptive gain controller is operating within a specific gain interval, which allows it to control the signal depending on its strength. The AGC:s biggest task is to increase the gain when the signal is weak and decrease it when the signal is too strong.

The AGC can be described mathematically by first calculating the power of the signal with the following formula

$$G(n) = \frac{1}{N} \sum_{k=0}^{N-1} x^2(n-k) \tag{2.11}$$

where $G(n)$ is the gain vector. This vector is divided into a predetermined amount of fragments, each sum of the value in the respective range. A recursive averaging can be used to simplify the sums and allows averaging without using much memory. The formula can be describes as

$$G(n) = \alpha G(n-1) + (1-\alpha)x^2(n) \tag{2.12}$$

where α is a parameter between 0.8 and 1. A digital signal processor has a limit of memory and therefore we are using the formula above. Instead of the parameter, a time constant τ can be designed for more intuitive understanding o the formulas behavior. The impulse response $y(n)$ of the Eq. 2.11 can be expressed as

$$y(n) = \alpha^n \tag{2.13}$$

which can be rewritten as

$$y(n) = e^{n \ln \alpha} \tag{2.14}$$

and differentiating the expression above with respect to x,

$$y'(n) = \ln(\alpha) e^{n \ln \alpha}$$

which gives $y'(0) = \ln \alpha$, the slope of the tangential line at $n = 0$. Since $y(0) = 1$, the complete tangential equation is written as

$$g = n \ln \alpha + 1$$

The time constant is defined as the point where the tangent slope crosses the x-axis, in the other words when $g = 0$, which gives

$$\tau = -\frac{1}{\ln \alpha}$$

Inserting the time constant τ in the equation 2.14 giving us the following relationship

$$\alpha = e^{-\frac{1}{n}} \tag{2.15}$$

and $n = F_s \cdot t$, where F_s is the sampling frequency and t is the time given in seconds.

2.4 Real Time in Matlab

Most real time signal processing platforms such as a DSP use stream processing, a very memory-efficient technique for handling large amounts of data [11]. Using stream processing we can divide incoming data into frames or blocks and fully processes each block before the next block arrives.

To handle large data sets, the main algorithm must manage memory and state information, store previous buffering data and update each buffer and state block by block or even frame-by frame in Matlab.

2.5 Digital Signal Processor

Most of the signals encountered in science and engineering are analog in nature. The signals are functions of a continuous variable, such as time or space, and usually take on values in a continuous range. Those signals may be processed by analog systems such as frequency multipliers or filters for the purpose of changing their characteristics or extracting some desired information [6].

Fig. 2.4 is illustrating the digital signal processing system of the analog signal. To perform the processing in a DSP, there is need for an interface between analog signal and digital processor, such as an analog-to-digital (A/D) converter and a digital-to-analog (D/A) converter. The digital signal processor can be a large programmable digital computer or a small microprocessor programmed to perform the desired operations of the input signal. It can also be a hardwired digital processor set to perform set of operations on the input signal.

There are many reasons why digital signal processing is preferred. A digital programmable system allows flexibility in re-configuring the digital signal processing operations by changing the program. Another reason is because digital signals are easily stored on magnetic disk such as a disk tape without loss of signal fidelity beyond that introduced in the A/D conversion. A digital system provides much better control of accuracy requirements such as, word length, floating-point or fixed-point and similar factors.

Digital implementation of the signal processing system are often cheaper than its analog counterpart. This due to the fact that the digital hardware is cheaper, or maybe it is a result of the flexibility for modifications provided by the digital implementation.

2.6 OP Amplifiers

Operational amplifiers is often devices which take a relatively weak signal as an input and produce a much stronger signal as an output. Operational amplifiers are very useful in equipment such as in this case a DC converter, stereo equipment and medical cardiographs, which amplify the heart beat.

An OP amplifier is an integrated circuit with a high-gain voltage amplifier with

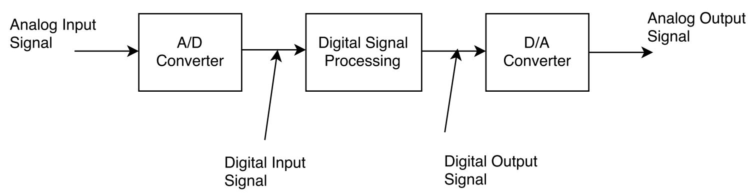

Figure 2.4: Block diagram of a digital signal processing system.

differential input. The amplifier has also a differential input: an inverting and a non-inverting input. A simple sketch of a OP amplifier is shown in Fig. 2.5.

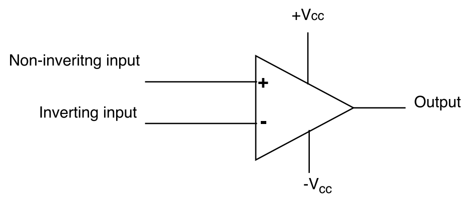

Figure 2.5: A simple design of OP amplifier.

A trans-impedance amplifier is a current-to-voltage converter using an OP amplifier. The concept is viewing in Fig. 2.6. By using voltage node analysis we can determine the closed loop voltage gain of this type of amplifier as

$$G(Av) = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}, \tag{2.16}$$

$$V_{out} = -R_f \cdot i_{in} = \frac{R_f}{R_{in}} V_{in}. \tag{2.17}$$

The equation for the voltage V_{out} shows that the circuit is linear for a fixed amplifier gain. This property can be very useful for converting a smaller sensor signal to a much larger voltage. The negative sign in the equation means an inversion of the output signal with respect to the phase input as it is π out of phase [1]. The trans-impedance amplifier is used to amplify the pressure signal from the cone crusher, which is given in current with the dimension mA to voltage.

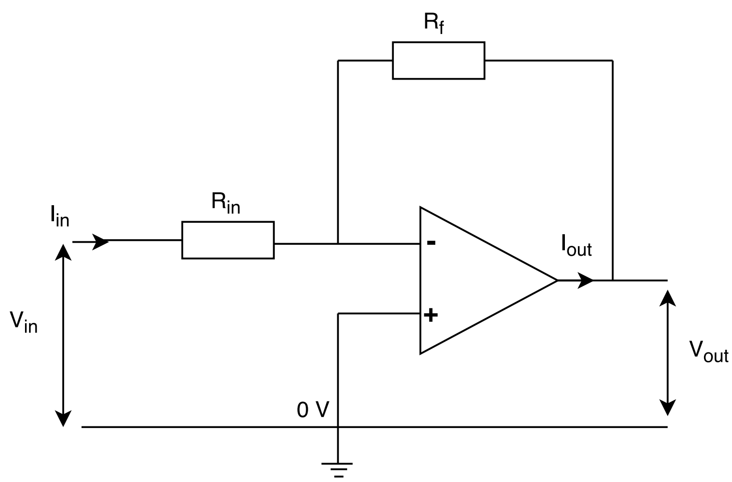

Figure 2.6: A trans-impedance OP amplifier.

Chapter 3

Phase Locked Loop

The phase-locked loop is commonly used in applications that measure frequency, phase of a sinusoid in the presence of signals or random noise. PLL is often used in classical modulation techniques such as amplitude modulation (AM), frequency modulation (FM) and phase modulation (PM) [8]. The PLL was first described in early 1930s, where its application was in the synchronization of the horizontal and vertical scans in television, later on with the development of integrated circuits [2]. The task of PLL is to track the input signal with another simulated signal. It synchronizes an output signal with reference in frequency and in phase. Fig. 3.1 demonstrates the concept where the real signal, a complex z_1, with specific phase and frequency, is followed by the tracking or simulating signal as complex z_2. Both are periodic functions of time, sinusoids or square waves. Instead of viewing the two signals as sinusoids and as functions of time, we can write them as phasors in the complex plane. With complex phasors, the two signals are two vectors rotating around the plane.

In general, the phase is a function of time as

$$\theta_n(t) = \int_{-\infty}^{t} w_n(t)dt. \tag{3.1}$$

The phase is basically the property of interest for the PLL designer. As an example, assume the first vector is rotating at constant speed, angular frequency $w_1(t) = w_c$. To make the new, generated signal follow the input signal, the new one has to adjust its phase to minimize the distance to the reference. That's is the phase error, denoted as θ_e. To adjust its phase the angular frequency needs to increase or decrease, depending on the first vector's movement. When both vectors are moving about the same rate we say that they are locked to each other. In this state the phase error between the two vectors is zero or constant, depending on the system design. If the reference signal deviates from its current angular frequency and a phase error increases, a control system such as PID controller is used to make the make the phase error smaller. The control system will force to lock the phase of the output to the input, and therefore it is called a phase locked loop.

In this thesis, an algorithm is designed to track the real system on Matlab. This will not differ much from the PLL used in receivers.

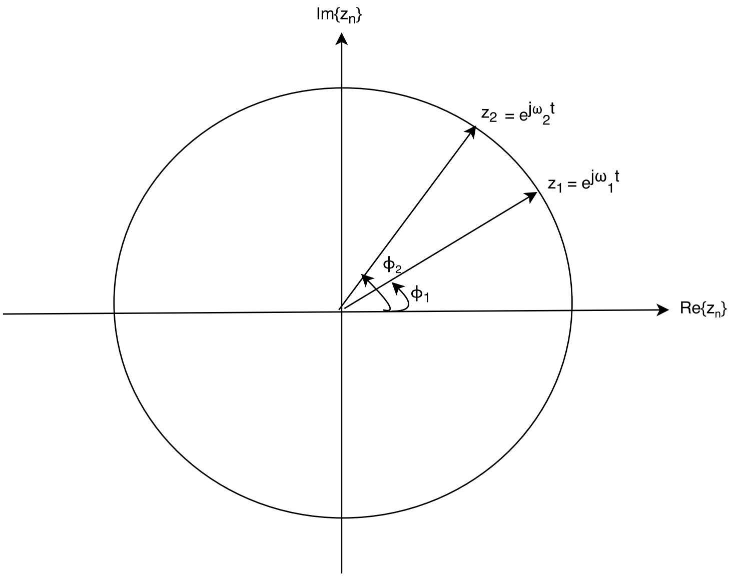

Figure 3.1: Complex representation of two signals. The speed of the rotation is the frequency of the sinusoids.

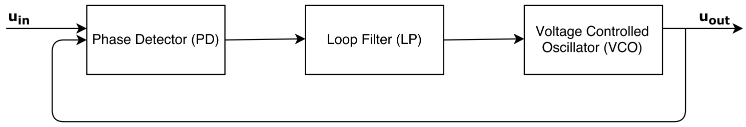

Figure 3.2: Block diagram of a linear PLL.

3.1 The PLL Fundamentals

The principle of a linear PLL is shown in Fig. 3.2, where a PLL consists of three main parts: a phase detector, a loop filter and a voltage controlled oscillator. The phase of the generated signal and the phase of the reference input has to be compared. To achieve this result, a phase detector is made which is used as a multiplier and a loop filter which is used later as a subsequent filtering operation.

The loop filter coefficients have to be chosen very carefully to make the phases between the generated signal with the feedback signal almost zero. In the other words, to make the PLL act as desired to lock the phase. The filtered phase signal is then fed to a VCO, which is an oscillator working around a quiescent frequency. The deviation from that frequency is determined by the control signal from the loop filter fed to it.

3.2 Phase and Frequency Relationship

According to Best [12], phase signals are a source of much trouble when understanding the phase locked loops in general. Let the input signal be

$$u = A\sin(\omega t + \phi(t)), \tag{3.2}$$

where A is the amplitude, ω is the frequency and $\phi(t)$ is the phase. For a PLL designer a signal such as this only carries its information in the phase. To demonstrate this, assume that the frequency is constant with $\omega = \omega_0$ for all $t < 0$ and then the frequency changes by $\Delta\omega$ at $t = 0$. The frequency step is shown in Fig. 3.3. The reference frequency after $t = 0$ can then be expressed as

$$u_1(t) = A_1 \sin((\omega_0 + \Delta\omega)t) = A_1 \sin(\omega_0 t + \phi_1) \tag{3.3}$$

The phase ϕ_1 can be written as

$$\phi_1(t) = \Delta\omega(t) \tag{3.4}$$

which means that the phase signal ϕ_1 is a ramp with slope $\Delta\omega$. This means that angular frequency of a signal is the derivative of its phase:

$$\omega_1 = \frac{d\phi_1}{dt}. \tag{3.5}$$

If the signal frequency changes linearly, the phase of the signal will increase quadratically, according to the frequency ramp in Fig. 3.4.

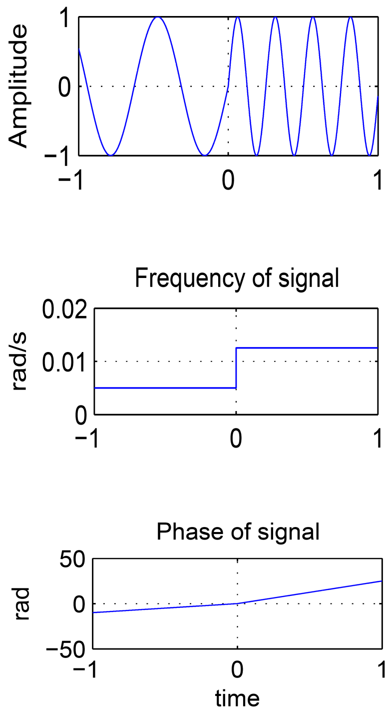

Figure 3.3: The frequency step.

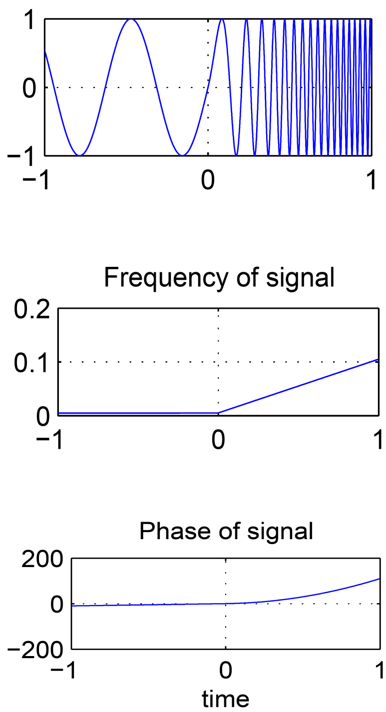

Figure 3.4: The frequency ramp.

3.3 The Transfer Function

Hence we consider the system is a linear model as the Fig. 3.5, we can do a system analysis using the transfer function $H(s)$ with respect to phase relationships. We can also assume that the PLL is locked and remains so in the near future. The transfer function for the input and output phases are given as

$$H(s) = \frac{\Phi_{out}(s)}{\Phi_{in}(s)} \tag{3.6}$$

The function $H(s)$ is a phase transfer function, and the model is configured for small changes in phase of the reference, if the phase error becomes too large in a short time, the linear PLL will unlock and a non-linear process takes place.

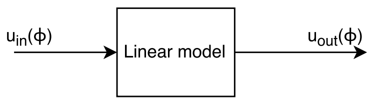

Figure 3.5: Block diagram of a linear model.

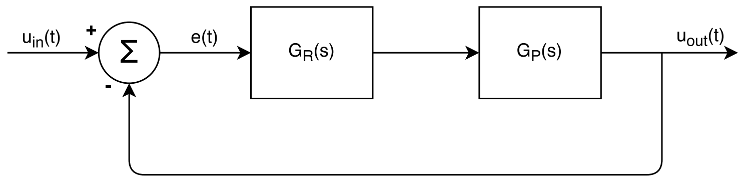

Figure 3.6: Block diagram of a control system.

To find out the transfer function for the whole linear PLL is to express the three building blocks as a control system. On basic control theory, a control system has a regulator part in series with a process where the process output is fed back and subtracted from the system input. A simple control system taken from the control theory is seen in Fig. 3.6.

Let the regulator have the transfer function $G_R(s)$ and the process $G_P(s)$, then the transfer function for the control system is derived in [16] and described by

$$H(s) = \frac{G_R(s)G_P(s)}{1 + G_R(s)G_P(s)}, \tag{3.7}$$

where the process $G_P(s)$ is actually the VCO and the regulator $G_R(s)$ is the loop filter. The phase detector can be realized with the subtraction in the control signal, which is the stationary error and is given as

$$E(s) = U_{in}(s) - U_{out}(s). \tag{3.8}$$

3.4 Phase Detector

Let the input signal be an sine wave as earlier discussed, with this expression

$$u_{in}(t) = A_{in}\sin(\omega_{in}t + \phi_{in}) \tag{3.9}$$

The signal generated from the VCO, the feedback signal, is another sinusoid described as

$$u_{out}(t) = A_{out}\sin(\omega_{out}t + \phi_{out}) \tag{3.10}$$

The phase difference between those two signals is obtained by multiplying the two signals and then filtering the result. That is actually the operation of the phase detector. Assume the PLL is locked or closely locked in frequency which means we have $\omega = \omega_{in} = \omega_{out}$. With the linear model definition the operation becomes

$$u_{pd}(t) = A_{in}\sin(\omega t + \phi_{in})A_{out}\sin(\omega t + \phi_{out}) \tag{3.11}$$

Using the trigonometric identity

$$\sin(\alpha)\sin(\beta) = \frac{1}{2}(\cos(\alpha - \beta) - \cos(\alpha + \beta)),$$

the result of the phase detector then becomes

$$u_{pd}(t) = \frac{A_{in}A_{out}}{2}(\cos(\phi_{in} - \phi_{out}) - \cos(2\omega t + \phi_{in} + \phi_{out})). \tag{3.12}$$

Using a subsequent filter with a higher order the second term $cos(2\omega t+\phi_{in}+\phi_{out})$ will be canceled out. Hence, this can be neglected in a linear model and the final output is then simplified to

$$u_{pd}(t) = K_{pd}\cos(\phi_e), \tag{3.13}$$

where $\phi_e = \phi_{in} - \phi_{out}$ and $K_{pd} = A_{in}A_{out}/2$. This is a non-linear relationship because zero phase error should correspond to zero output of the detector. In the same way, small errors in phase should cause small results from the detector. One solution to this problem is to make $u_{in}(t)$ and $u_{out}(t)$ $\pi/2$ radians out of phase, replacing $u_{out}(t)$ with a cosine function. This method is also called a quadrature detector instead of phase detector according to Louis [9]. The trigonometric identity is

$$\sin(\alpha)\cos(\beta) = \frac{1}{2}(\sin(\alpha - \beta) + \sin(\alpha + \beta)),$$

then gives the quadrature detector

$$u_{qd}(t) = \frac{A_{in}A_{out}}{2}(\sin(\phi_{in} - \phi_{out}) + \sin(2\omega + \phi_{in} + \phi_{out})). \tag{3.14}$$

Neglecting the higher order term and inserting $\phi_e = \phi_{in} - \phi_{out}$, the formula will be simplified to

$$u_{qd}(t) = K_{qd}\sin(\phi_e) \tag{3.15}$$

For small errors in phase, the result assumes to

$$u_{qd}(t) \approx K_{qd}\phi_e. \tag{3.16}$$

Concluding the phase detector equations, the linearized model is only a zero order block with the gain $K_{qd} = A_{in}A_{out}/2$. In the loop filter part, the higher order component will be canceled.

3.5 Voltage Controlled Oscillator

The task of VCO is to generate a sinusoidal or square signal, which its frequency depends on the input signal. It can operate a quiescent frequency ω_c, preferably close to the signal that the system will lock on. The VCO output is actually u_{out} which is sinusoidal and only depends on the input u_{lf} which is the filter loop output. This can be described by

$$u_{out}(t) = \cos((\omega_c + K_{vco}u_{lf}(t)t)) \tag{3.17}$$

where K_{vco} is the VCO gain with the dimension rad/Vs, the angular frequency of the VCO is therefore

$$\omega_{out}(t) = \omega_c + K_{vco}u_{lf}(t) \tag{3.18}$$

For the PLL, the phases are more important and therefor we need to transform Eq. 3.18 to the phase ϕ_{out} which is given by the integration of the frequency variation $K_{vco}u_{lf}(t)$ as

$$\phi_{out}(t) = \int_0^t K_{vco}u_{lf}dt = K_0 \int_0^t u_{lf}dt$$

The Laplace transform for the integral term is $1/s$. Hence, the Laplace transform for the output phase ϕ_{out} is

$$\Phi_{out}(s) = \frac{K_{vco}}{s}U_{lf}(s) \tag{3.19}$$

The transfer function for the VCO part is then

$$H_{vco}(s) = \frac{\Phi_{out}(s)}{U_{lf}(s)} = \frac{K_{vco}}{s} \tag{3.20}$$

We can directly find that the VCO is an integrator for the phase.

3.6 Loop Filter

From the PLL block diagram, the loop filter is the regulator in the control system for phase signals. In control theory the regulator has two important tasks; to compensate for disturbances that affects the system and to take necessary actions when desired value is changed. In our case, the PLL has the phase input signal as the desired value. While choosing the loop filter, several properties needs to be studied such as stability, accuracy and speed and also which type of filter to select.

3.6.1 The Closed Loop System Conditions

A very important property of a control is the speed. When the desired value is changed in some manner, it takes some time for the output of the system to find the steady-state final value. To measure the speed we need to find out which rise time the system has. The rise time is defined as a signal for a specified low value, 10% of its final value to 90% of its final value [19].

Speed and stability are contradictory demands: higher speed implies higher stability and so on. In our case, as quick as possible when stability demands are met. To make the speed goes faster we need to choose a suitable damping factor which will be discussed later in this thesis.

The stability is the most important and therefore the most complex part in this algorithm. The linear PLL is described by a non-linear differential equation. The complete system stability and the filter design is not theoretically derived or predicted by this thesis in all situations due to its high complexity. Instead, testing and simulating the system to ensure that the system is stable even when it is non-linear mode of operation. The linear part of stability can be analyzed using the properties of linear system using the model derived further below in this section.

About the static accuracy, since we deal and tolerate with a small phase difference or drift between the input frequency and the half frequency output. The static accuracy is defined as zero remaining error when the input is changed. It must be asymptotically zero independently how the input is changed.

3.6.2 Choosing the Correct Filter

To choose the right loop filter implies choosing of the type and order of the filter so the closed loop system type and order can fit the above conditions. What type of filters refer to the number of poles in the transfer function, the order is the same as the highest degree of the characteristic equation. Three basic filter types common in PLL is mentioned by Best [12]. They are all low pass filters with different cutoff frequencies. The first one is called passive lag filter or phase-lag comepensator with the transfer function $F(s)$ given by

$$F_1(s) = \frac{1 + s\tau_1}{1 + s(\tau_1 + \tau_2)} \tag{3.21}$$

The second one is also a similar transfer function but with multiplying with a gain term a. This is called Active lag filter and is given by

$$F_2(s) = a \cdot \frac{1 + s\tau_2}{1 + s\tau_1}. \tag{3.22}$$

The last filter is commonly referred as a PI-filter, where PI stands for a proportional and integral action according to the control theory. The transfer function for this type of filter is

$$F_3(s) = \frac{1 + s\tau_2}{s\tau_1} \tag{3.23}$$

The main problem is how to choose the correct filter to obtain the zero phase error which is the criterion of static accuracy. The phase error has the transfer function as $\Phi_e = \Phi_{in} - \Phi_{out}$ and seen in Fig. 3.7, which shows the complete linear model of the PLL.

By looking at Fig.3.7 and seeing the error signal $\Phi_e(s)$ as a system output, we can define a phase error transfer function as

$$E(s) = \frac{\Phi_e(s)}{\Phi_{in}(s)} = \frac{1}{1 + K_{qd}F(s)\frac{K_{vco}}{s}} \tag{3.24}$$

Finding out the error transfer function, we can also find the error transfer function $\phi_e(t)$ for any given input signal $\phi_{in}(t)$ since

$$\Phi_e(s) = \Phi_{in}(s) \cdot E(s), \tag{3.25}$$

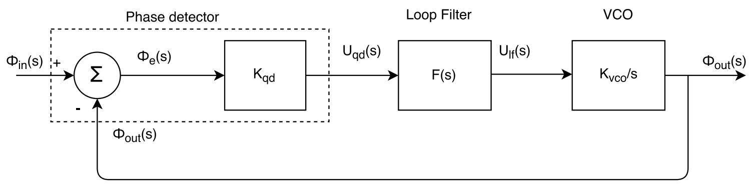

Figure 3.7: Linear model of the PLL.

by the inverse transform of $\phi_e(s)$. But since we want to find the final error as a function of time which goes to infinity, we can use the final value theorem of the Laplace transforms. Assume that g is causal and that $G = L\{g\}$ is rational. If all poles to $sG(s)$ has negative real part, then

$$\lim_{t\to+\infty} g(t) = \lim_{s\to+0} sG(s)$$

This will give us the opportunity to find the error as time goes to infinity without having to transform $\Phi_e(s)$ back to the time domain.

Tracking the frequency with a PLL means that the output frequency $\omega_{out}(t)$ has to be adjusted correctly by the VCO if the input frequency changes in the reference $\omega_{in}(t)$. The system senses like a frequency step. Recall that the frequency step according to Fig. 3.3, is a phase ramp for the linear system. The Laplace transform of the phase ramp input is

$$Y_i(s) = \frac{C_i}{s^2},$$

which is given by Laplace transform table. The variable C_i is the frequency difference in radians per second of the phase detector. Let's use Eq. 3.25 then we will get the following expression

$$\Phi_e = \Phi_{in}(s)E(s) = \frac{C_i}{s^2}\frac{1}{1 + K_{qd}F(s)\frac{K_{vco}}{s}}. \tag{3.26}$$

Using the final value theorem we will get

$$\lim_{t\to+\infty} \phi_e(t) = \lim_{s\to+0} s\Phi_e(s) = \lim_{s\to+0} s\frac{C_i}{s^2 + K_{qd}K_{vco}F(s)} \tag{3.27}$$

Hence, the final expression is

$$\lim_{t\to+\infty} \phi_e(t) = \lim_{s\to+0} \frac{C_i}{s + K_{qd}K_{vco}F(s)} \tag{3.28}$$

Any filter transfer function $F(s)$ can be inserted in Eq. 3.28 to obtain the final error when the system is subject to a phase ramp. A more generalized filter transfer function can be described by

$$F(s) = \frac{N(s)}{D(s)s^n}, \tag{3.29}$$

where $D(s)$ and $N(s)$ are the denominator and nominator polynomials. In the Laplace domain, the factor s^2 are actually double poles at $s = 0$. Inserting Eq. 3.29 into Equation 3.28 we have

$$\lim_{t \to +\infty} \phi_e(t) = \lim_{s \to +0} \frac{C_i}{s + K_{qd} K_{vco} \frac{N(s)}{D(s)s^n}} \tag{3.30}$$

$$= \lim_{s \to +0} \frac{C_i D(s) s^n}{D(s)s^{n+1} + K_{qd} K_{vco} N(s)} \tag{3.31}$$

From Eq. 3.31, if $n \geq 1$ the limit will approach zero as time goes to infinity. Therefore, we can state that if we want to track the reference with zero phase error, then a filter pole must have a pole in $s = 0$. From the filter types described by Best [12], inserting the passive lag filter as shown in Eq. 3.21 into Eq. 3.28

$$\lim_{t \to +\infty} \phi_e(t) = \lim_{s \to +0} \frac{C_i}{s + K_{qd} K_{vco} \frac{1+s\tau_1}{1+s(\tau_1+\tau_1)}} \tag{3.32}$$

Observing that $\frac{1+s\tau_1}{1+s(\tau_1+\tau_1)} \to 1$ as $s \to 0$, Eq. 3.32 reduces to

$$\lim_{t \to +\infty} \phi_e(t) = \frac{C_i}{K_{qd} K_{vco}}, \tag{3.33}$$

which means that the final error is not reduced to zero. The error approaches zero only if the loop gains K_{qd} and K_{vco} are large enough. The second filter $F_2(s)$ has a similar property, which gives us the same result as the first filter, meaning that the final error $\phi_e(t)$ is not converging to zero but to $\frac{C_i a}{K_{qd} K_{vco}}$. The third filter, PI-filter has a pole in $s = 0$ as seen in Eq. 3.23 and should reduce the final error to zero. The final error using this type of filter is calculated as

$$\lim_{t \to +\infty} \phi_e(t) = \lim_{s \to +0} \frac{C_i}{s + K_{qd} K_{vco} \frac{1+s\tau_2}{s\tau_1}} \tag{3.34}$$

Observing that $\frac{1+s\tau_2}{s\tau_1} \to \infty$ as $s \to 0$, which mean that the final error goes towards zero. The pole in $s = 0$ provides the filter with infinity gain at DC and therefore the system reduces any remaining phase error to zero.

To conclude about determining the suitable filter, the PI-filter, $F_3(s)$, is the one that will be used in this type of PLL. What is remaining is determining the constants K_{qd}, K_{vco}, τ_1 and τ_2.

3.7 Determining the Constants

Reviewing the transfer function of the blocks, covering in Fig 3.7 we can rewrite the phase transfer function as

$$H(s) = \frac{\Phi_{out}(s)}{\Phi_{in}(s)} = \frac{K_{qd} F_3(s) \frac{K_{vco}}{s}}{1 + K_{qd} F_3(s) \frac{K_{vco}}{s}}. \tag{3.35}$$

By inserting the third filter $F_3(s)$ and simplifying the equation we will get the following expression

$$H(s) = \frac{K_{qd}\frac{1+s\tau_2}{s\tau_1}\frac{K_{vco}}{s}}{1 + K_{qd}\frac{1+s\tau_2}{s\tau_1}\frac{K_{vco}}{s}} = \frac{K_{qd}K_{vco}\frac{1+s\tau_2}{\tau_1}}{s^2 + \frac{K_{qd}K_{vco}\tau_2}{\tau_1}s + \frac{K_{qd}K_{vco}}{\tau_1}} \tag{3.36}$$

Taken from control theory the general equation for the transfer function as seen in Eq. 3.7. Analyzing the closed loop it is convenient to rewrite the transfer function on a normalized form by making the denominator be

$$D = s^2 + 2\zeta\omega_n s + {\omega_n}^2$$

where ζ is the damping factor and ω_n is the natural frequency. Substituting the denominator in Eq. 3.36, we identify

$$\omega_n = \sqrt{\frac{K_{qd}K_{vco}}{\tau_1}}, \qquad \zeta = \frac{\omega_n\tau_2}{2} \tag{3.37}$$

The final phase transfer function can then be written as

$$H(s) = \frac{2\omega_m\zeta + {\omega_n}^2}{s^2 + 2\omega_n\zeta s + {\omega_n}^2} \tag{3.38}$$

Hence, the phase error transfer function as Eq. 3.24 can be rewritten in the form of natural frequency and the damping factor as

$$E(s) = 1 - H(s) = \frac{s^2}{s^2 + 2\omega_n\zeta s + {\omega_n}^2} \tag{3.39}$$

Plotting the transient response of the PLL using Eq. 3.38 we will tune the parameters as the natural frequency ω_n and the damping factor ζ. Once they are tuned and determined, the filter parameters τ_1 and τ_2 can be obtained and the PLL design is complete. Increasing the damping factor means that the system is over-damped and then the response may be sluggish. Decreasing the damping factor means instead that the system is under-damped, then we have an oscillating system. To get a good trade-off between stability and speed the damping factor sets to $\zeta = \frac{1}{\sqrt{2}}$. In Fig. 3.8a we can see how the damping factor ζ influences the step response of the error function ϕ_e. When designing a correct natural frequency it is very important that the generated signal is pure and not noisy. Looking at the amplitude frequency of the closed loop as seen in Fig.3.8b, we could easily view a PLL as a filter. A double frequency in the generated input signal is nothing but noise in the phase detector part. A noise from phase detector will also cause noise in VCO. That's why we want to tune down the bandwidth of the system to make the damping high at double frequency. This will lead us to a non-liner behavior, which is discussed in the next section.

A low loop bandwidth will reject high frequency noise fed into the system. For this particular PLL implementation, the input signal is likely stable and it does not contain any basband information that needs to be preserved.

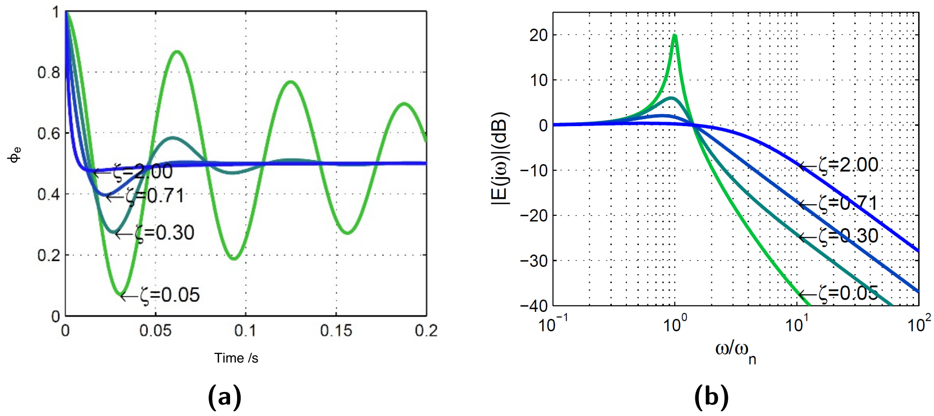

Figure 3.8: (a) Step response of PLL error function with various damping factors ζ. (b) Step response of PLL error function for different damping factors ζ as function of the normalized frequency ω/ω_n.

Determining a suitable natural frequency ω_n implies determining some important non-linear properties of the PLL. In this thesis, we will not go through in detail into non-linearity but a brief overview of non-linear PLL is explained in the next section.

3.8 Non-linearity Propertions

If the reference frequency and the VCO frequency are different, we say that the PLL is not in locked mode and the linear part is not valid anymore. Therefore, we will not derive non-linear equations due to its complexity but we will take some good information about the behavior of non-linear properties.

According to Best [12], three important questions regarding the PLL-lock is explained. The first one is under what conditions the PLL get locked, the second one is about how much timed does it take to achieve the lock-in process, the last one is under what conditions the PLL lose the lock. These questions are not answered by mathematical derivation as in the liner mode.

3.8.1 Unlocked Process

Four stability regions are mentioned in [12], where these regions are defined as deviations in frequency from quiescent frequency of the VCO and also how fast the reference frequency changes. Those four regions are described in the Fig. 3.9.

The hold range $\Delta\omega_H$ is the range where PLL can statically preserve phase tracking. If the reference frequency deviates this far from the designed quiescent frequency

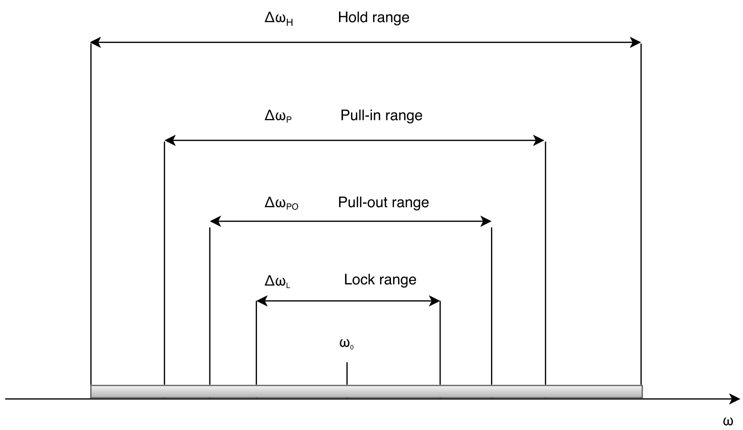

Figure 3.9: The four regions described by Best [12].

the PLL will unlock and the phase error goes into infinity. This is also independent of the speed of PLL, when the frequency is changed.

The pull-in range $\Delta\omega_P$ is the range whitin which a linear PLL will always belcome locked if unlocked. This type of range can be infinite if the correct filter is used. This pull-in process is very slow. After the pull-in process, the linear phase takes place and it settles $N\ x\ \pi$ out of phase, which is a true phase lock. In the unlocked state, the phase detector modulates the VCO in a nonharmonic way. This implies that the output of VCO is also nonharmonic. That is, we have a time dependent frequency diffrence, but the frequency of the VCO varies in such way that is more often closer than further from the reference. The asymmetry of the VCO causes it to change the average frequency even faster towards the reference. The VCO is pulling itself in.

The pull-out range $\Delta\omega_{PO}$ is the dynamic limit for stability operation of a PLL. If the PLL is lost within this range, the PLL will lock again. This process can be slow if it is about a pull-in process. This defines how large a frequency step the PLL can handle without any unlocking.

The lock range $\Delta\omega_L$ is the frequency range within which a single beat note between reference and output frequency within a PLL locks. This is also independent of the speed of PLL when the phase is changed, as long as it does not exceed this range.

3.8.2 Designing a Criterion from Non-Linear Properties

To make the PLL be locked we need to select the correct filter and depending on this choice the PLL will stay unlocked or slowly toward the locked state. This can be probably shown by design a filter which has a pole in $s = 0$. This will lock a PLL eventually. We can state that it has an infinite pull-in range. If the speed is never an issue, the natural frequency ω_n can be set low to make the system minimize the phase jitter noise. In this thesis we will deal with an input frequency around 6 Hz, and this is the rotation of the cone crusher which is considered to be stable.

Chapter 4

Quadrature Phase Locked Loop

In the previous chapter, we have been discussed about only one real input signal which has to be tracked by the linear PLL. But, how about if we want to track two signals at the same time? The accelerometer signals in two directions x and y will be tracked together by a method called the quadrature PLL. The difference between this and LPLL is that QPLL is tracking two dependent signals at the same time. Due to the symmetry, the relationship between x and y directions is a phase of $90°$. Let the input phase of these signals be presented as a complex vector:

$$p(t) = x(t) + iy(t), \tag{4.1}$$

where $x(t)$ and $y(t)$ are the accelerometer signals in x- and y-direction. This design will simplify how to present the real position of the cone crusher with complex numbers. The simulated signal coming out from the QPLL is defined by $\hat{p}(t)$. In Fig. 4.2, the complex signal with the input phase $p(t)$ and the simulated QPLL signal $\hat{p}(t)$ are presented.

The quadrature PLL is consisted of three parts: complex representation, quadrature detector and a PID controller. The quadrature detector is calculating the phase difference between the input and output signal. The PID controller is minimizing the stationary error coming from the quadrature detector, so that the QPLL is locked. The block diagram for this type of PLL is presented in Fig. 4.1.

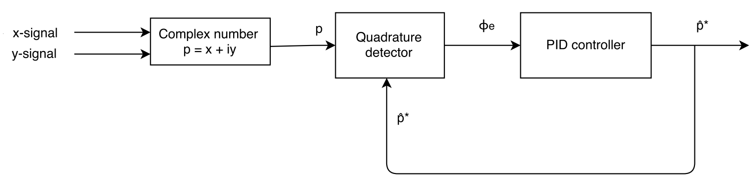

Figure 4.1: The block diagram of the quadrature PLL.

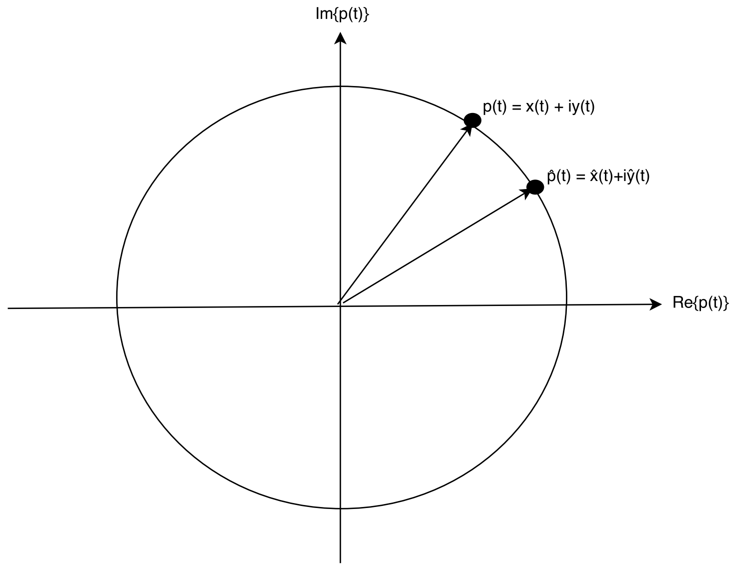

Figure 4.2: The complex representation of the two accelerometer signals, as a complex input signal with the QPLL output signal.

4.1 Quadrature Detector

As we mentioned before, a phase detector is a multiplier of the input real signal with the signal coming from the PLL. Rewriting the complex number to polar form we have

$$p(t) = Re^{j\phi(t)} \tag{4.2}$$

Assume that the absolute value of the phase R is one. The same way, we have the simulated tracking phase signal as $\hat{p}(t) = Re^{j\hat{\phi}(t)}$. In order to use a PLL to lock the phase of two quadrature signals x and y, a phase detector is used. A quadrature signal is also a complex signal composed of two sinosoids at the same frequency but with a 90° phase offset between them. Eq. 4.2 can be developed to

$$p(t) = \cos(\phi(t)) + i\sin(\phi(t)) \tag{4.3}$$

The quadrature phase detector generates its error by multiplying the complex signal with the conjugate of the QPLL phase signal and taking the imaginary part of the output. The stationary error can also be expressed as

$$
\begin{aligned}
&\mathrm{Im}\{p(t) \cdot \hat{p}(t)^*\} \qquad (4.4)\\
&= \mathrm{Im}\{[\cos(\phi(t)) + i\sin(\phi(t))] \cdot [\cos(\hat{\phi}(t)) - i\sin(\hat{\phi}(t))]\}\\
&= \mathrm{Im}\{\cos(\phi(t))\cos(\hat{\phi}(t)) + i\sin(\phi(t))\cos(\hat{\phi}(t))\\
&- i\cos(\phi(t))\sin(\hat{\phi}(t)) + \sin(\phi(t)\sin(\hat{\phi}(t))\}\\
&= \sin(\phi(t))\cos(\hat{\phi}(t)) - \cos(\phi(t))\sin(\hat{\phi}(t))\\
&= \sin(\phi(t) - \hat{\phi}(t))\\
&= \sin(\phi_e(t))
\end{aligned}
$$

And for small phase difference, we have $\sin(\phi_e(t)) \approx \phi_e(t)$. When this stationary error is zero, then the QPLL is locked.

4.2 PID controller

The PID controller is the most common controller form of feedback. Nowadays, in process control, more than 95 % of the control loops are of PID type, most loops are PI controller [7].

The main algorithm of an PID controller is described by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \qquad (4.5)$$

where K_p, K_i and K_d are non-negative coefficients for the proportional, integral and derivative terms (PID), y is the measured process variable, r the reference variable, u is the control signal and e is the control error. Fig. 4.3 is illustrating the PID model.

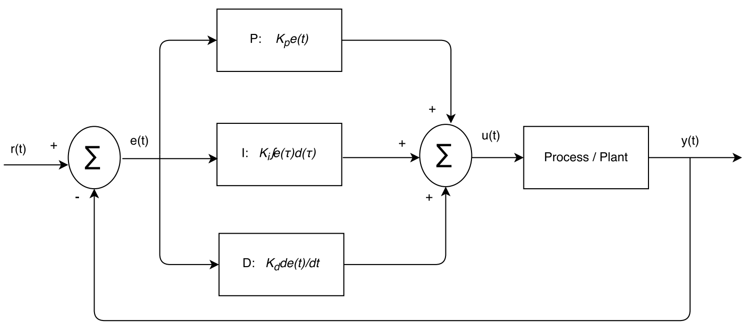

Figure 4.3: Block diagram of a PID controller.

The PID controller in the QPLL is decreasing the phase error $\phi_e(t)$ between the input signal $p(t)$ and the output signal $\hat{p}(t)$. If the output signal has a lower phase compared to the input signal, then the PID controller will increase the frequency of the output signal so that it will minimize the phase error. By choosing the suitable parameters such as K_p, K_i and K_d in the controller, the QPLL will lock the output signal. To get the stationary error be minimum, the integral term of the PID controller is not suitable for this. Therefore, the PID controller only becomes a PD-controller to reduce the phase error.

Chapter 5

Method

5.1 Hardware Implementation

The first part in this thesis is how to get the correct signals from the cone crusher. In Fig. 5.1, a brief overview of the hardware is sketched. The hardware implemmentation is consisting of three parts: The acceleromter signals, the trans-impednace amplifier and the audio recorder. To get the suitable signals, a three-axis $\pm$ 16 g acceleromter (ADXL326) is used. We are only interesting of two axis in two dimensions x and y to find out the position of the cone crusher.

The trans-impedance amplifier is used to convert the pressure signal from current to voltage, in other words it is working as a DC-converter. The last block is about the audio recorder which is consisting of 4 in- and 4 out-channels. According to its specification, the audio recorder can record a sampling frequency up to 96 kHz. The recording mode here is only 8 kHz because the signals from the cone crusher are quite low. Because it is about audio recorder, two 10 m RCA cables are used as connectors between the audio recorder and together with the accelerometer and the trans-impedance amplifier. A figure of the real implementation is presented in Fig 5.3

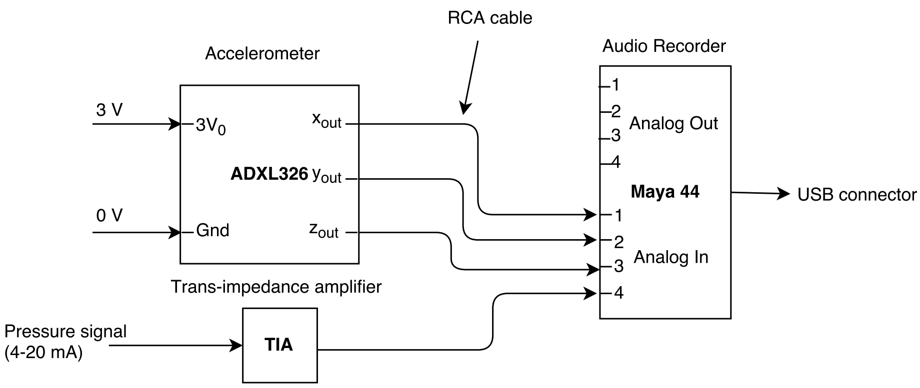

Figure 5.1: Hardware implementation.

5.1.1 Selecting the Accelerometer

It is challenging to choose the suitable accelerometer to the installation on the cone crusher. Some questions that has to be answered are:

- Analog or digital accelerometer?

  The answer to the this question is determined by the hardware itself. Analog accelerometers have an output which is a continuous voltage proportional to the acceleration. Digital accelerometers use instead PWM, so that there is a square wave at a specific frequency. Therefore, the time period of high voltage is proportional to the amount of the acceleration. Because we are dealing with sinusoidal signals from the cone crusher, an analog accelerometer should be the correct choosable one.

- Single or multiple axes accelerometer?

  Because we want to know the position in the cone crusher, a single-axis accelerometer is not enough. We need at least two axis accelerometer to get the movement signals from the cone crusher. Due to movement and vibrations from the cone rock crusher at least $\pm 10\ g$ is recommended.

- High level in sensitivity or mid-range?

  The sensitivity is the ratio of change in acceleration input to change in the output signal. This defines a straight-line relationship between acceleration and output. Sensitivity is specified at a particular supply voltage and is typically expressed in units of mV/g for analog output accelerometer. For analog-output sensors, sensitivity is ratiometric to supply voltage.

- What bandwidth do we need?

  The bandwidth ensures reliable readlings and this is about how many times per second the accelerometer will read the data. For vibration and movement measurement, as in our case, the bandwidth needs to be several hundred Hertz.

A $\pm 16\, g$ three-axis accelerometer (ADXL326) is used because it fully covers all the requirements above (cf. Fig 5.2).

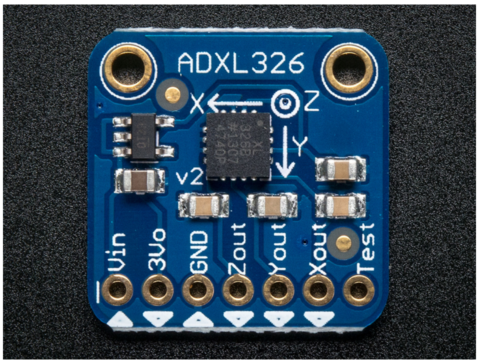

Figure 5.2: ADXL326 chip.

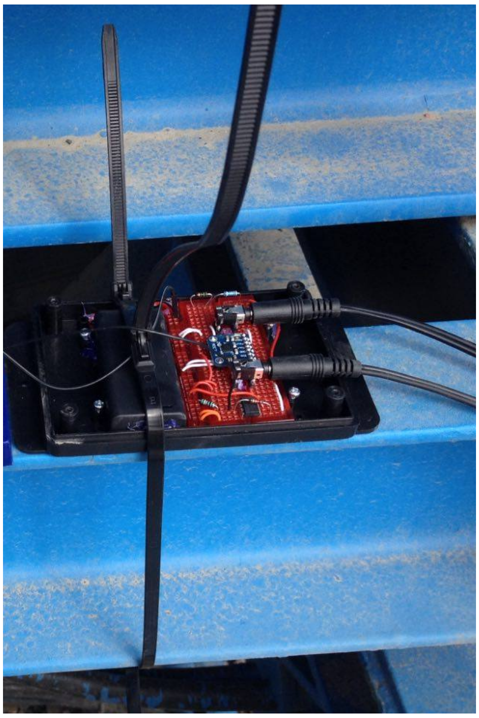

Figure 5.3: The hardware circuit mounting on the cone crusher.

5.1.2 Audio Recorder

An audio recorder, such as Maya44, is used to take care of the signals coming from the accelerometer and the pressure. Maya44 is a high quality 24-bit 96/192 kHz 4-in/4-out audio interface providing a number of powerful and amazing features. There is also a headphone output, which is perfectly for monitoring. The audio recorder is using a method called full scale its AD-converter, which means that our signal results will have no specific dimension, neither in voltage $[V]$ nor acceleration $[m/s^2]$.

Figure 5.4: Maya44 USB audio recorder.

5.1.3 Position of the Hardware Circuit

To get the correct signals from the accelerometer, it is very important to know how to place the accelerometer on the cone crusher. Different placements have been tested and the results are noticed and discussed later in the next chapter. The best way to place it is seen in Fig. 5.5. We need to observe that the figure is taken from above and the accelerometer signals are connected to the audio recorder by RCA cables.

5.1.4 Pressure Signal

To determine the skew distributions in the cone crusher, we need to find out the hydraulic pressure signal because this signal is directly proportional to the force. How to receive the correct signal from the cone crusher is described in detail in Appendix B. According to data sheet of cone crusher, the pressure signal is given in current with a range 4 mA-20 mA from a pressure transmitter. The audio recorder can *only* read the signals in voltage, up to 5 V. Therefore, a trans-impedance OP amplifier is used to convert the current data to voltage. The resistances R_f and R_{in} are set to 250 Ω and 1 $k\Omega$ respectively.

To calculate the voltage input we use Ohm's law as

$$V_{in} = R_{in} \cdot i_{in} \tag{5.1}$$

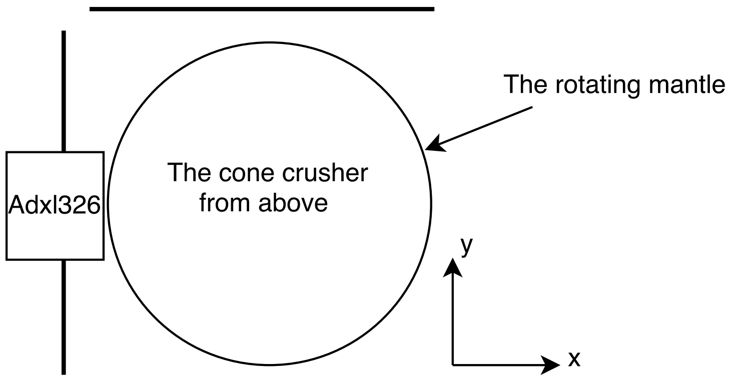

Figure 5.5: Position of the accelerometer on the cone crusher.

Choosing the input resistance R_{in} to be 1 $k\Omega$ and with the current between 4 mA and 20 mA, we get V_{in} to be $4V$ to 20 V. Using Eq. 2.17, with $R_f = 250\ \Omega$, we will get V_{out} as an interval from 1 V to 5 V, which covers the specifications of the audio recorder.

5.2 Software Implementation

When recording the signals we can process the results in Matlab. The first thing is to make a good low pass FIR filter to get rid of the high noise and disturbances for our sinusoidal signals. Then, the amplitude of the signals must be at the same level by using an adaptive gain controller. Taking the fast Fourier transform for the accelerometer signals can be interesting if you want to look at the average speed for the cone crusher. Some initial values needs to be chosen such as the gain constants K_{pd}, K_{vco}, the filter constants τ_1 and τ_2 in the PI filter for the PLL. In Fig. 5.6, the main algorithm for the whole software implementation is described.

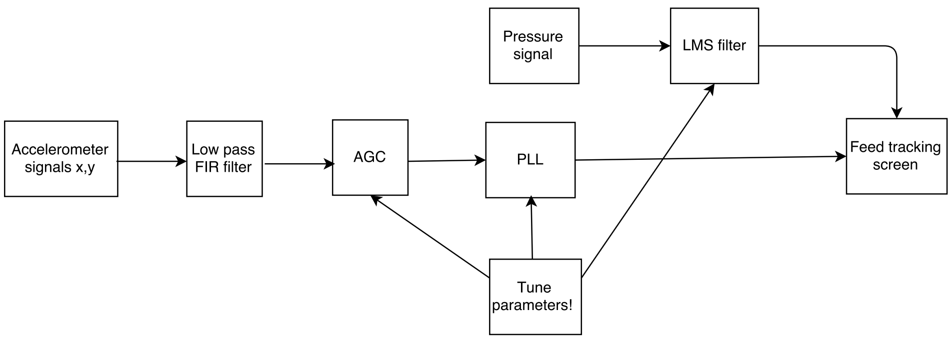

Figure 5.6: The whole algorithm for the software implementation from the recording input signals to the plots.

5.3 Real Time Implementation

The real-time implementation is the final implementation for the whole prototype in this thesis. Given in data sheet for the audio recorder Maya44, the signal can only proceed a real time implementation with a specific sampling frequency f_s at 48 kHz. And with this number we choose to down-sample the process to a sample rate of 6 kHz to get the correct signals in the system. Then we have to implement the frame-by frame method to achieve the real time processing as seen in Fig. 5.7.

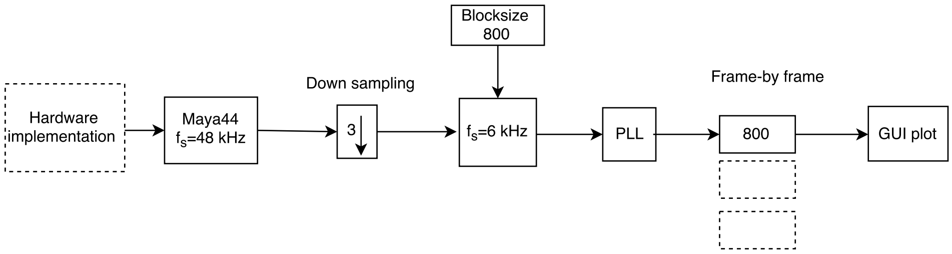

Figure 5.7: The real time processing from the hardware implementation through the audio recorder.

Chapter 6

Results

6.1 Accelerometer Signals

The chapter presents all the signals coming from the accelerometer. The cone crusher has different execution modes. The first mode is when the cone crusher is idle, in the other words the cone crusher runs without any load of rocks. The second mode is when the cone crusher just runs after insertion of the rocks. The third mode is where the crusher runs with a rock load. The final mode is about the crusher is being discharged from all the rocks.

6.1.1 The Cone Crusher is Idle

In this mode, the cone crusher is running in idle, which means without any rock load. The signals are very noisy and needs to be filtered. Butterworth low pass filter method is used to filter out most of the noise and disturbances. We choose the order of filter to be $n = 4$ with a cut-off frequency $f_c = 0.0025 \cdot 8\,kHz = 20\ Hz$.

Plotting the position with the system coordinates (x, y) from the filtered signal, we will get the following result as seen in Fig. 6.1. In the same figure we can directly find out that the movement of the accelerometer is elliptical where the amplitude in y-axis is almost three times longer than the amplitude in x-axis. That is, we have a stable movement along the y direction from the cone crusher.

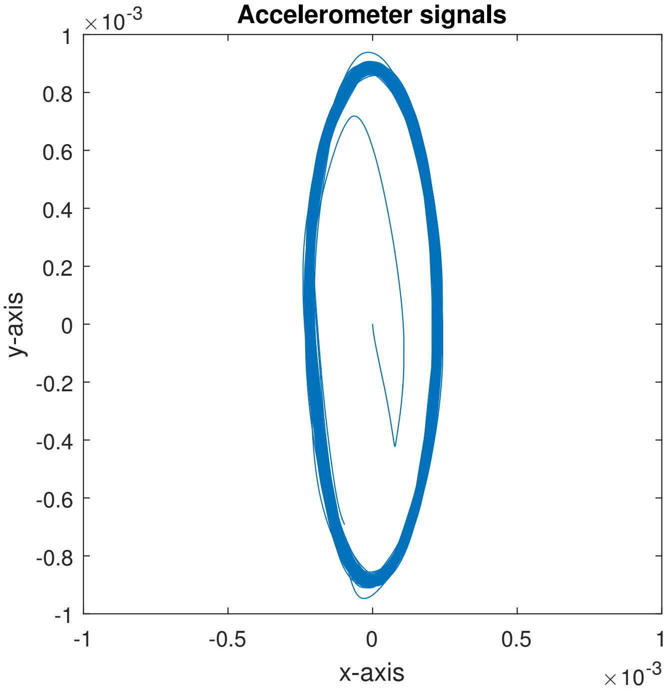

Figure 6.1: The accelerometer coordinate system (x, y) when the cone crusher is idle.

6.1.2 The Cone Crusher is just Loaded

This is the first mode where the cone crusher is loaded with rocks. As usual, we use the same filtering coefficients for all measurements to get the result in Fig. 6.3. We can directly find out that there are disturbances in the cone crusher, especially along the x direction, with big vibrations. The position on the cone crusher is now different compared to when the cone crusher is idle.

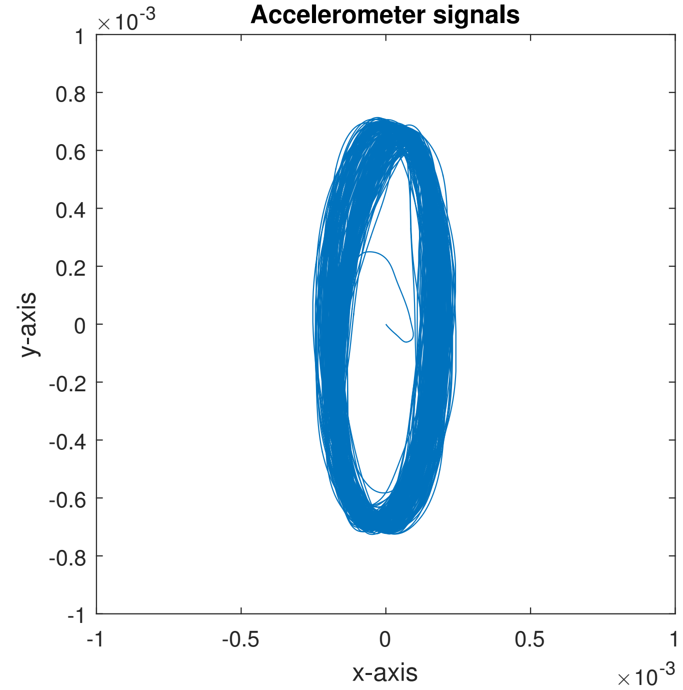

Figure 6.2: The accelerometer coordinate system (x, y) when the cone crusher is just loaded.

6.1.3 The Cone Crusher is fully Loaded

In this mode, the cone crusher is running with rock loaded and running at a constant process. This mode constitutes for the most time of the process since the cone crusher has to break down all the rocks. This is where the cone crusher is running in a stable state. The vibrations are less than the previous mode but the signal is lower during breaking the rocks.

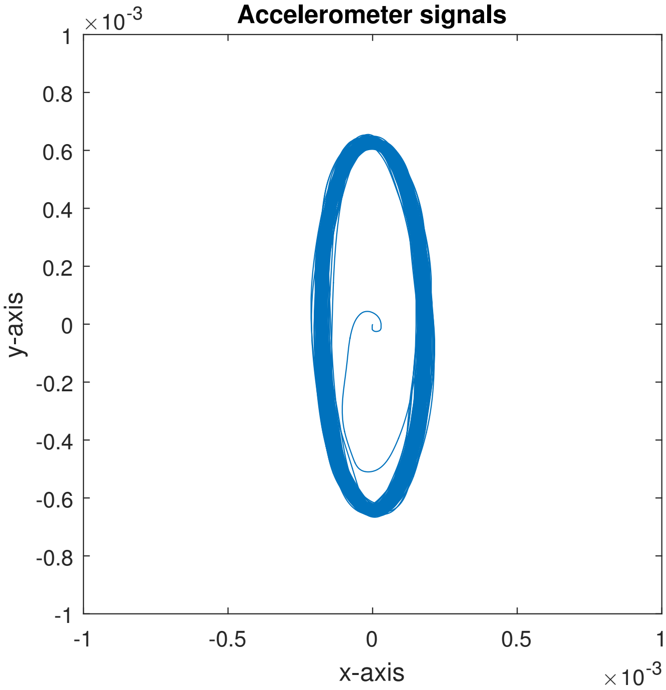

Figure 6.3: The accelerometer coordinate system (x, y) when the cone crusher is fully loaded.

6.1.4 The Cone Crusher is Unloaded again

The cone crusher is empty again from rocks and is idle again with a stable position according to the coordinate system (x, y) as seen in Fig. 6.4. This mode is very similar to the first mode, when the cone crusher is idle.

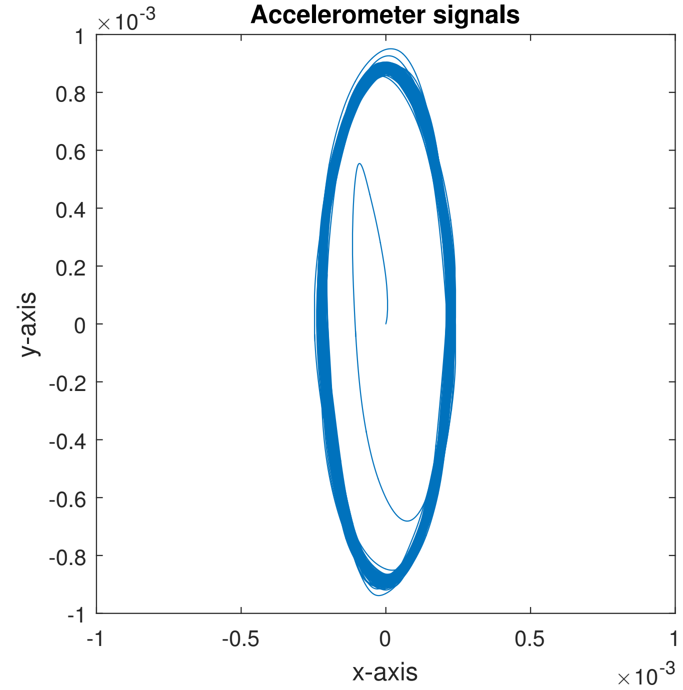

Figure 6.4: Accelerometer coordinate system (x, y) when the cone crusher is unloaded (idle).

6.2 Adaptive Gain Control

In this section, an adaptive gain controller is used to stabilize and normalize the amplitude of the accelerometer signals. This will make the accelerometer movement from ellipse to circle with a radius of almost 1. Let's take for example when the cone crusher is idle. As seen in Fig. 6.5 and 6.6 the adaptive gain controller is trying to normalize and increase the radius of both axes to be 1, without changing the phase or angle of the position.

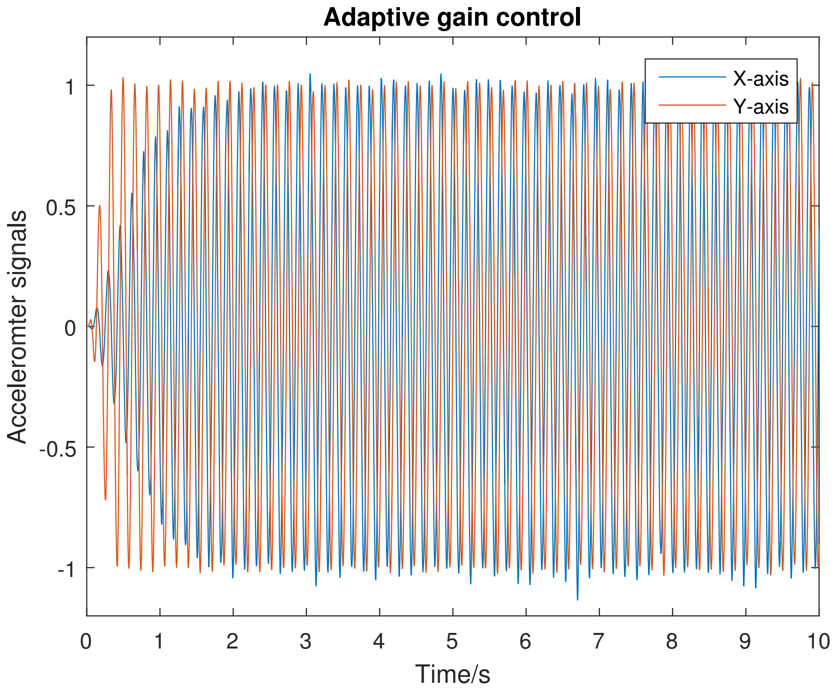

Figure 6.5: Adaptive gain control for the accelerometer signals (x, y) when the cone crusher is idle.

6.3 Phase Locked Loop

As mentioned in the theory, PLL tracks the real signal by following its phase. In other words, the phase gives us the coordinate position of the PLL signal and the frequency is the speed for this signal. By choosing suitable parameters we can find the quickest way to track the real signal.

Figure 6.7 shows how the PLL is tracking the real signal with some oscillating part in the beginning. The next figure 6.8 is illustrating the signal coming from the phase detector which is the same thing as the phase difference between the real and PLL signal. In the control system this signal is also called stationary error. There is some oscillations in our control system in the first 2-3 seconds, which makes the tracking signal difficult and has to struggle to find its lock-in operation.

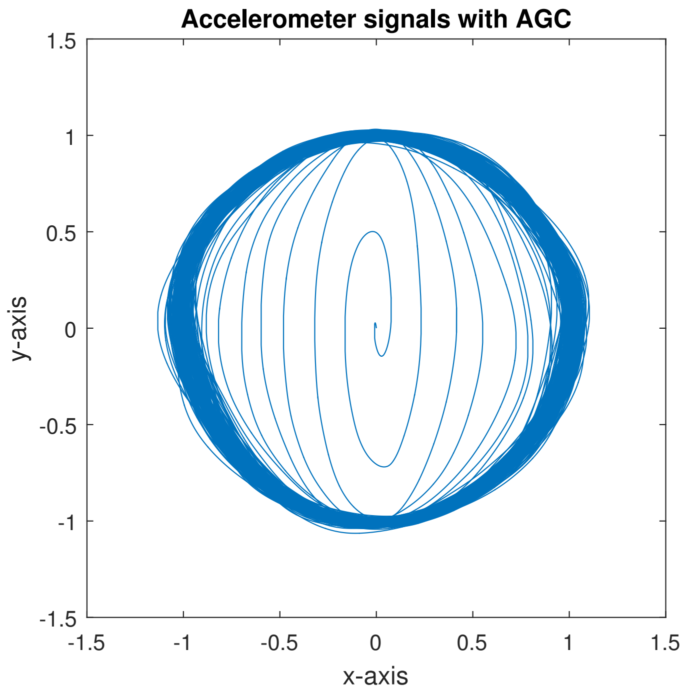

Figure 6.6: Adaptive gain control for accelerometer coordinate system (x, y) when the cone crusher is idle.

6.4 Feed Tracking Screen

This is the last and the most important part in this thesis where the pressure together with the position of the cone crusher is plotted. Using the LMS filter on the pressure signal, which consists of the error signal $e(n)$, which is the difference the desired signal $d(n)$ and the output signal $y(n)$. Parameters are needed to fulfill the LMS filter specification such as the step size μ. Therefore, we can finally plot the feed tracking screen as in the following figures. Fig. 6.9 illustrates the big pressure variations where the cone crusher is just loaded with rocks, and Fig. 6.10 describes the situation where the the cone crusher is fully loaded with rocks.

According to Fig. 6.9, the pressure signal is not in symmetry and the uneven load is very clear at this moment from the phase position from about 90° to 300°. This means the density of the rocks in the left is much bigger when the cone crusher is just loaded. The pressure signal has also spikes, which is fundamental and coming from the vibrations during the process. In Fig. 6.10 the cone crusher has an even distribution of the rocks when it is fully loaded. We can not find out any uneven load from this, which means that the cone crusher is breaking all the rocks from all the positions at the same time. For more analysis about the feed tracking see Appendix A.

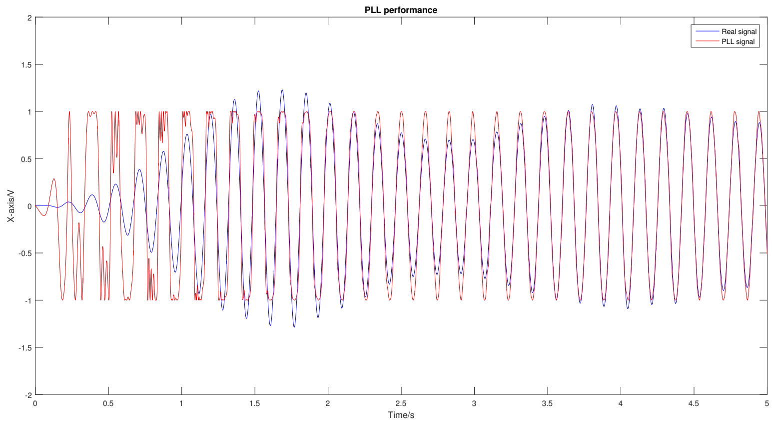

Figure 6.7: PLL performance tracking the real signal.

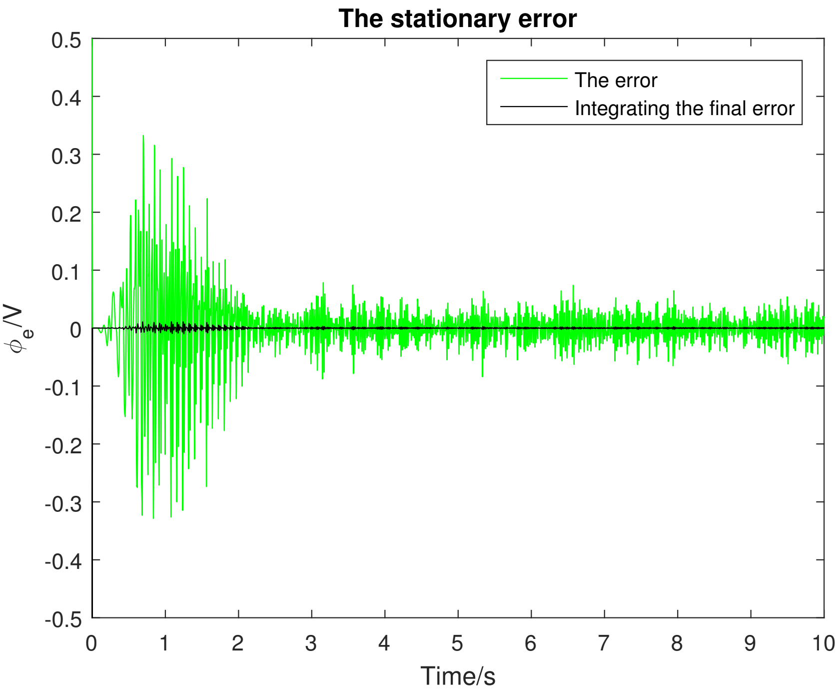

Figure 6.8: The phase detector signal or the stationary error (Green) with its integration plot (Black).

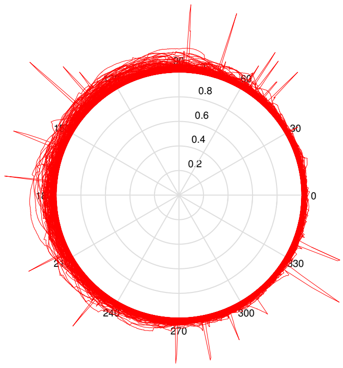

Figure 6.9: The feed tracking when the cone crusher is just loaded.

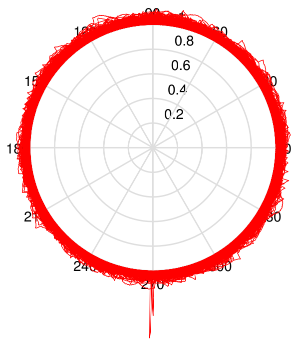

Figure 6.10: The feed tracking when the cone crusher is fully loaded.

The speed of mantle rotation in the cone crusher can be very interested in some aspects. The feed tracking screen can always be rewritten as function of time as seen in Fig 6.11 (where the cone crusher is fully loaded). The speed is almost constant at 6 rps or Hz.

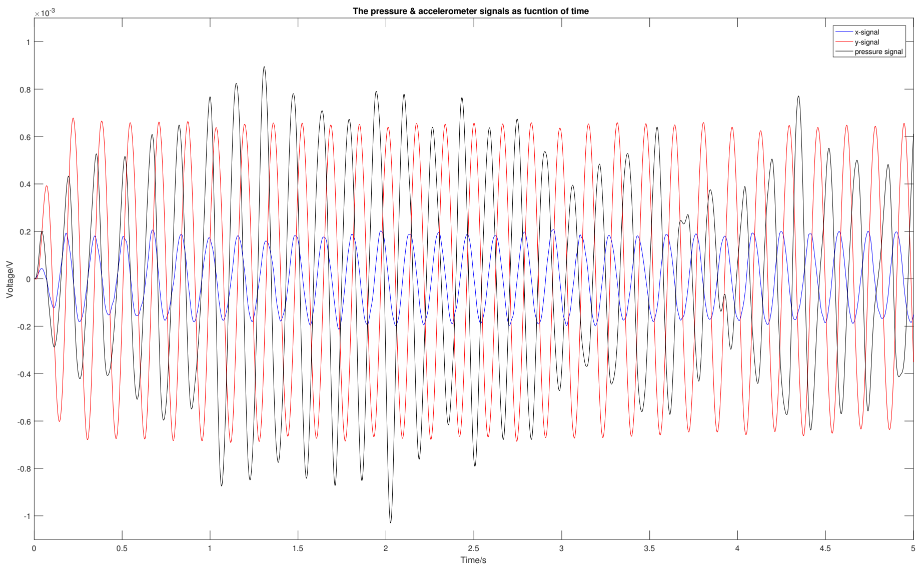

Figure 6.11: The pressure and accelerometer signals as function of time.

Chapter 7

Discussion and Conclusion

7.1 Selecting the Correct Accelerometer

It is very difficult to find out which accelerometer is the best choice for the cone crusher. The cone crusher has many disturbances and vibrations, specially when it is just loaded with rocks. In this thesis two different accelerometers are used to measure the position signals from the cone crusher.

In our first test we used a three-axis $\pm 200\ g$ accelerometer (ADXL377) to measure the required signals. By plotting the signals, we could directly find that signals were very low and lacked sensitivity. Therefore, we chose lower value in acceleration ± 16, which gave us the best choice in terms of sensitivity.

On the other hand, there were many accelerometers with better tolerance in terms of vibrations but they were much more expensive. This three-axis $\pm 16\ g$ accelerometer (ADXL326) is probably one of the cheapest on the market and it is made for constructing simple prototypes.

Installing the accelerometer in cone crusher in a good position will give us the correct signals. In the beginning of our tests, the phase shift between the x-axis and y-axis was $\pi/4$ instead of the expected one $\pi/2$. installing the accelerometer across the cone crusher as seen in Fig. 5.5 will give us the correct symmetric results.

7.2 Audio Recorder versus Arduino Uno

Before the audio recorder, we were using a Arduino Uno to record the accelerometer signals using its Arduino software. The Arduino component has an onboard 6 AD-converter. The converter has 10 bit resolution, returning integers from 0 to 1023. This is not enough to cover and record all the information in the signals. By this reason, we were choosing the audio recorder with its AD/DA converter with 24 bit and a much better resolution, which can cover a signal up to sampling frequency $f_s = 96$ kHz.

7.3 Phase Locked Loop

There are many ways to calculate phase locked loop algorithm. In this thesis, the PLL algorithm together with the control theory are presented. The lock-in state is our steady-state, which we will lock on the real signal as soon as possible. To achieve the lock-in state as stable as possible during the disturbance and vibration periods. Therefore, we need to to choose the parameters of loop filter carefully and tune the parameters of the PLL. It will take about 5-10 seconds to find its steady-state.

On the other hand, the adaptive gain controller method will also take about 5 seconds to get the correct gain for our signals. In total, we must be aware that 15 seconds in our real-time processing is not counting until the system finds its stable state. By looking at accelerometer signals as shown in Fig. 6.3, there are some movements which go far away from our expected movement (ellipse or circle). This is due to the PLL signal has not find its lock-in state, which is also the oscillating part in the beginning of the signal as seen in Fig. 6.7.

7.4 Future Work

7.4.1 Accelerometer

The accelerometer is to give us the correct position in the cone crusher. There are always some disturbances in form as vibrations from the cone crusher. A three-axis accelerometer might give us enough data to solve the problem but might also lose precision compared to if we would have used two accelerometer. Installing another accelerometer will give us more data that we can process and will give us more precise result about the position. This will also lead to the accelerometer signals becomes in circular form instead of getting the elliptical form. The problem with this suggestion is that it will be expensive to invest in another accelerometer and it will also have lower mobility when installing another accelerometer.

7.4.2 Phase Locked Loop

There has been a lot of research in PLL algorithms for the last decades, especially in radio electronics and television screen development. The phase locked loop needs always to be improved and tuned by the user. This thesis presents a simple working linear phase locked loop algorithm but it can always be improved and tuned depending on the different products in different environments.

7.4.3 Implementation on DSP

The next step, given more time would be to implement the phase locked loop algorithm on the handheld DSP. This implementation means that our Matlab code should be implemented and translated to C code.

References

[1] Electronics Tutorials (2015), *Inverting Operational Amplifier* http://www.electronics-tutorials.ws/opamp/opamp_2.html [visited 14-04-2016]

[2] G. Reehal (1998), *A Digital Frequency Syntheesizer Using Phase Locked Loop Technique*, The Ohio State University, http://www2.ece.ohio-state.edu/ie/main/Publications/PLL/reehal_thesis.pdf [visited 05-05-2016]

[3] J. Fessler (27-05-2004), *Design of Digital Filters*, http://web.eecs.umich.edu/~fessler/course/451/l/pdf/c8.pdf [visited 18-06-2016]

[4] Jadmin (18-04-2016), *Accelerometer*, SensorWiki, http://www.sensorwiki.org/doku.php/sensors/accelerometer [visited 25-06-2016]

[5] John (17-08-2011), Instrumentation Today, *MEMS Accelerometer*, http://www.instrumentationtoday.com/mems-accelerometer/2011/08/ [visited 25-06-2016]

[6] John G. Proakis and Dimitris G. Manolakis (1996), *Digital Signal Processing: Principles, Algorithms and Applications*, Prentice Hall, Upper Saddle River, New Jersey, IBSN:0-13-394338-9

[7] Karl Johan Åström (2002), *Control System Design*, http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf [visited 21-04-2016]

[8] Kevin M.Rolfes (1994), *The automated software phase-locked loop and the exploration of an adaptive algorithm for the adjustment of PLL parameters*, University of Wisconsin, http://www.rolfes.org/kevin_rolfes_msee_thesis.pdf [visited 01-05-2016]

[9] Louis Litwin (2003), *Using PLLs to Obtain Carrier Synchronization Part 2*, EE Times, http://www.eetimes.com/document.asp?doc_id=1277127 [visited 25-03-2016]

[10] M. Andrejasic (2008), *MEMS accelerometers*, University of Ljubljana, http://mafija.fmf.uni-lj.si/seminar/files/2007_2008/MEMS_accelerometers-koncna.pdf [visited 25-06-2016]

[11] MathWorks (n.d.), *Stream Processing in MATLAB*, http://se.mathworks.com/discovery/stream-processing.html [visited 15-03-2016]

[12] Roland E. Best (2003), *Phase Locked Loops: Design, Simulation, and Applications*, McGraw-Hill, IBSN:0-07-141201-8

[13] S. Beeby, G. Ensell, M. Kraft, N. White (2004), *MEMS mechanical sensors*, Artech house incl., USA

[14] S. E. Lyshevski (2002), *Mems and Nems: systems, devices and structures*, CRC Press LLC, USA

[15] Simon S. Haykin (2001), *Adaptive filter theory*, Upper Saddle River, NJ: Prentice Hall, IBSN:0130901261 9780130901262

[16] Torkel Glad, Lennart Ljung (2006), *Reglerteknik: grundläggande teori*, ISBN:9789144022758

[17] Wikipedia(a) (2016), *Butterworth Filter*, `https://en.wikipedia.org/wiki/Butterworth_filter` [visited 20-06-2016]

[18] Wikiepedia(b) (2016), *PID controller*, `https://en.wikipedia.org/wiki/PID_controller#/media/File:PID_en.svg` [visited 25-04-2016]

[19] Wikipedia(c) (2016), *The Rise Time*, `https://en.wikipedia.org/wiki/Rise_time` [visited 06-05-2016]

Appendix A

Appendix

A.1 25-50 mm Rocks

This group has not so big variations in the size of rocks. Plotting the feed tracking screen we will get in 10 s running cone crusher as Fig. A.1. The speed is 6 Hz, meaning the cone crusher has done 60 rps. The crusher has a large even distribution, and it is very difficult to see the uneven parts.

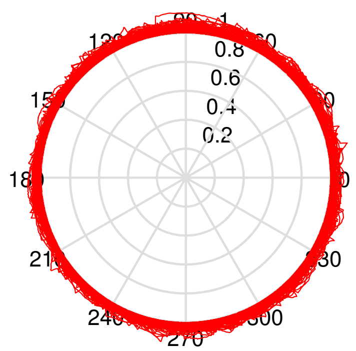

Figure A.1: Feed tracking when CC is fully loaded with 25-50 mm rocks

A.2 20-80 mm Rocks

Here, we have more variations in the rocks which theoretically cause more uneven load on the crusher. But, by looking at Fig. A.2, there is no specific uneven load on the crusher. The force, which is directly proportional to the pressure is very well distributed on the crusher. The same as the previous results, this has also a great rock distribution on the crusher.

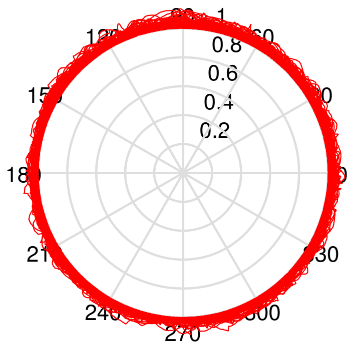

Figure A.2: Feed tracking when CC is fully loaded with 20-80 mm rocks

Appendix

To connect to the pressure transmitter we need to study sketch of B2N measurement systems of the cone crusher. The transmitter has to be fed with 24 Vdc in the first pin (BROWN), the second pin (WHITE) is our pressure signal given in current (4-20 mA). The fourth pin (BLACK) is set to ground. Fig. B.1 gives a brief overview of the measurement systems with the different pins included.

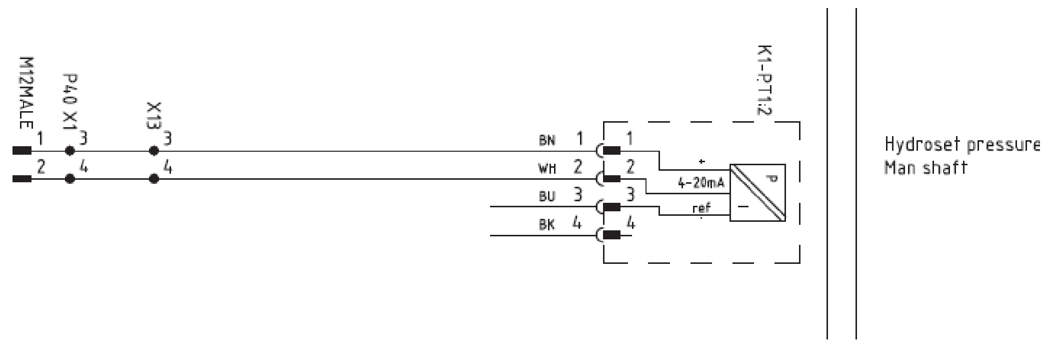

Figure B.1: B2N measurement sketch

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2016-541
http://www.eit.lth.se