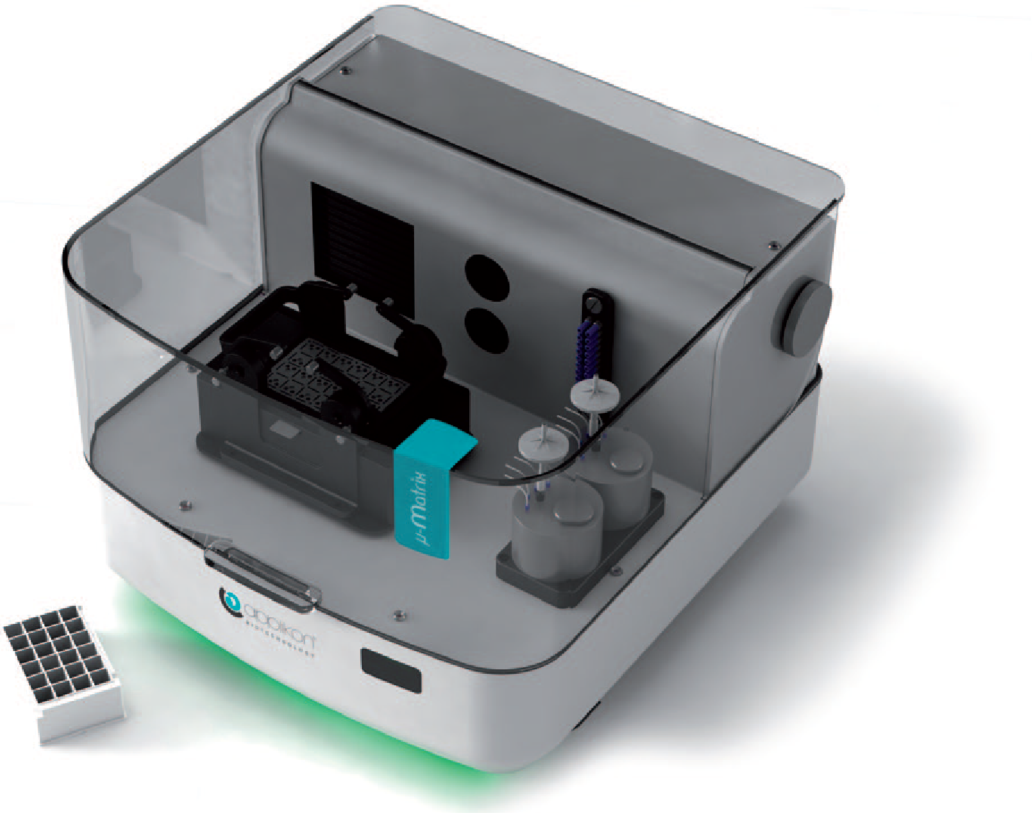
Cell Concentration Sensor for Micro-Bioreactors

Software & Data Processing

Jeroen Keijsers and Ashvant Mahabir



TU Delft Delft University of Technology

Challenge the future

Cell Concentration Sensor for Micro-Bioreactors

Software & Data Processing

by

Jeroen Keijsers and Ashvant Mahabir

4087313
4043278

in partial fulfillment of the requirements for the degree of

Bachelor of Science
in Electrical Engineering

at the Delft University of Technology,
to be defended on Tuesday July 1, 2013 at 13:30.

Supervisors:	dr. ir. A. Bossche	
	ing. J. Bastemeijer	
Thesis committee:	prof. dr. P.M. Sarro	TU Delft
	dr. ir. S. Wong	TU Delft
	dr. ir. A. Bossche	TU Delft
	ir. T. Walvoort	Applikon Biotechnology

TU Delft Delft University of Technology

Abstract

This thesis was written for the bachelor's final project, of the study Electrical Engineering. It describes the design of a cell concentration sensor. A proof of concept was designed which is able to measure the cell concentration of yeast in a suspension by using sensors which measure the impedance and optical properties of the suspension.

The project is divided into three parts where each duo will have their own focus. This thesis focuses on the software and data processing subsystem. The other pairs will focus on the optical and impedance sensors

The signals from both sensors are converted to digital values. To process the raw data, filters are designed to condition the signal. Using the filtered data, routines are implemented to calculate the cell concentration. This value is transferred to a computer, where a user interface is implemented to graphically display the progress of the cell concentration.

Using only the optical sensor, the measurements were accurate to about 40% in the range 0-10 g/l, and 10% in the range 10-120 g/l, making the total range 0-120 g/l. The impedance sensor was extend the range of the measurements to 0-150 g/l with an accuracy of 15%. However, the calculations needed to convert impedance values into cell concentrations were too sensitive for the microcontroller

The routines for the optical sensor were successfully implemented on an Arduino microcontroller showing promising results. This was achieved with the help of a graphical user interface which was designed for validation when actual yeast measurements were performed. Although the routines for the impedance sensor yielded correct results for capacitance, the implementation of the code on the Arduino gave erroneous values when cell concentration was measured.

Preface

This thesis was written in the context of the Electrical Engineering Bachelor final project, EE3842 'Bachelor afstudeerproject'. The work was carried out in a room on the department of Electronic Instrumentation, on the 15th floor of the Faculty Electrical Engineering, Mathematics and Computer Science at the Delft University of Technology. The support and access to measurement facilities was excellent. We want to thank Jeroen Bastemeijer and André Bossche, our supervisors, for their support and feedback during the project. Any needs from our side were directly taken care of, for which we are very grateful.

Also, we would like to thank Timo Walvoort, the contact person from Applikon Biotechnology, for the swift responses, extra information and feedback on our work. In addition, we were given a tour of the new Applikon building at the beginning of the project, which was very informative for the context of the project and the micro-Matrix.

Last but not least, we want to thank the other group members: Jornel van den Hoorn, Michel Jansen, Joost van der Kemp and Erik Lemmens. Together, a pleasant work atmosphere was created in which this proof of concept was ultimately designed and realised.

Jeroen Keijsers and Ashvant Mahabir
Delft, June 2014

Contents

1 Introduction

Applikon Biotechnology B.V. is a Dutch manufacturer of bioreactors. The micro-Matrix is currently their newest and most innovative bioreactor. The micro-Matrix can simultaneously grow 24 cultures in different wells, intended for the determination of the optimal growing conditions for the bio-process. These conditions, which include temperature, dissolved oxygen and pH, must be carefully monitored. For further enhancement of the system, Applikon would like to add more sensors. A parameter which is of great use, is the cell concentration, which is directly coupled to the bioreactor efficiency. The ability to continuously measure the cell concentration with a sensor is a heavily desired addition to the current system.

This thesis describes the design of such a sensor. It is the purpose of this bachelor's final project to design and build a proof of concept of a cell concentration sensor. Since the group consists of six students, it is divided into pairs. This thesis describes the software and data processing subsystem of the sensor system.

The thesis is structured as follows: First, a detailed problem definition is given of the total system, and of the software subsystem. Then, a schematic system overview will be given and each building block is briefly explained. The following two chapters are devoted to one of either data paths: one data path of the optical sensor and one data path of the impedance sensor. The next chapter will discuss the integration of the system as standalone, interfacing with the computer, or the possibilities of integrating the system in the micro-Matrix. Next, the results are presented, after which a discussion and a conclusion complete the thesis.

2

Problem definition

2.1. Total sensor

This section gives a description of the total problem, the demands to its solution, possible solutions and a description of the system. The next section will zoom in on the software and data processing subsystem, which is the subject of the rest of this thesis.

2.1.1. Description of the problem

In the micro-Matrix, parameters such as temperature, pH and gas flow are already monitored. However, a heavily desired parameter is cell concentration. When this parameter is known, it can be directly monitored which conditions are optimal for the specific process. The information from the sensor can be graphically displayed and the user can see which process is performing best. It is the purpose of this project to create a proof of concept of a cell concentration sensor.

2.1.2. Design brief

The design brief for the total cell concentration sensor, including some remarks per demand:

- **The sensor must be able to measure the concentration of cells.**
  This is the main goal of the sensor system.

- **The results of the sensor should be accurate to 10%**
  This demand is agreed upon, as a realistic accuracy. At 10% accuracy, the data is useful for the user to draw conclusions about the bio-processes in the different wells.

- **The sensor should be able to measure cell concentrations in the range 0-200 g/l.**
  The cell concentrations in the micro-Matrix can differ from no cells to as much as 200 g/l. Thus, the sensor should be able to measure cell concentration in this range.

- **The measurement rate should be at least once per hour.**
  The processes to be monitored can last up to two weeks. Therefore, it is not useful to measure the cell concentration each millisecond: the process is relatively slow. As a demand, the period of one hour is chosen to produce a useful time resolution of the measurements. It is noted that this time is for the whole matrix, so 24 individual bioreactor wells have to be measured in this time, allowing a maximum measurement duration of 2.5 minute per well.

- **The used technology has to fit in the measurement wells of the micro-Matrix, which have a working volume of 5 ml.**
  The goal of this project is to create a proof of concept of a cell concentration sensor system. Beyond the scope of this project, the technique may be used to integrate a similar cell concentration sensor in the micro-Matrix. Because of this possibility, the technologies and components used in the proof of concept should be scalable, so that they can be used in the final product.

- **The sensor must be able to be used for cells varying in size from 3 μm to 30 μm.**
  The micro-Matrix will be used for a large number of different cell types, with diameter ranging from 3 μm to 30 μm. Therefore, the sensor should be able to monitor the concentration of these cells.
- **The sensor should not affect the bio-process.**
  The measurement should not interact or disturb the bio-process, as the process is then not representative anymore for later repetitions of the process. If the culture is contaminated, it is no longer useful and can be discarded, which would be disastrous.
- **No samples should be taken for a measurement.**
  This is similar to the previous demand. Also, since the wells of the micro-Matrix only have a working volume of 5 ml, any sample taken will reduce the volume of the bio-process considerably. This reduction adds up when a sample is taken every hour for experiments that last up to two weeks.
- **The system needs to be able to remain sterile.**
  The bio-processes that will be monitored need to be in a sterile environment, to be completely insensitive to disturbances from the outside. This is done to conduct a reliable experiment, where all the factors can be controlled. Any contamination can render the whole experiment useless.
- **The system must give reproducible results.**
  Any accurate system is reliable and thus produces reproducible results. When a measurement is conducted on the same concentration twice, the results must of course be coherent.
- **The system must be safe to use.**
  The operator and environment of the micro-Matrix should not be in danger when using the system.
- **The system must not affect measurements.**
  Besides the cell concentration measurement system, other sensors are situated in the micro-Matrix. The results from these measurements must remain reliable and therefore, the cell concentration sensor may not disturb them.

Besides the demands, it would be desirable that the system is capable of distinguishing between living and dead cells. As only the living cells are of interest to the user, it is a wish to ignore the dead cells. However, since this is not a demand, it is not directly listed in the design brief.

Also, the available time for this project is limited. A strict deadline limits the possibilities of conducting extensive, thorough measurements, or researching novel measurement methods.

In addition, the budget available from the department is limited. This means no expensive measuring devices can be bought, which limits the solution possibilities.

2.1.3. Solution possibilities

A literature study has been conducted to find possible techniques for measuring the cell concentration. This section will shortly describe its findings.

Optical

Optical techniques are the market standard. Within this denominator, several different techniques can be distinguished: spectography [1], scattering [2] and optical density [3–5].

The last is most widely in use in commercial cell concentration sensors today. The spectrography technique is not viable for this project, as the data processing is extensive. In [1], data processing is done in a computer. However, for this project, all data processing is done in a microcontroller so the sensor system can be integrated in the micro-Matrix. Furthermore, spectography is not suitable for our purposes since large optic devices such as lenses and LED-arrays are needed. Measuring the scattering and optical density is viable, and will be taken further into consideration.

Impedance

Measuring the cell concentration of a suspension by measuring its impedance is another option. The relative permittivity ϵ_r of a suspension changes with its cell concentration [6]. When two electrodes are attached, a capacitance is formed with varying permittivity. This capacitance can then be measured [7–13].

This method is promising because in theory, dead cells can be ignored, as their cell membranes have fallen apart so they do not behave as capacitances anymore. Furthermore, a large range can be obtained.

Microwave

The cell concentration can also be measured by microwave sensing. Using microwave signals, the permittivity of the suspension is measured, from which the cell concentration can be calculated [14–17].

The technique shows great results, but it not viable as it operates at very high frequencies and requires much computational power [15].

Acoustic

Acoustic measuring of cell concentration is another option [18]. Its principle of operation is that a suspension with a higher concentration of cells has a higher resonant frequency. The fact that the cassette of the micro-Matrix is spinning around at up to 380 rpm [19] and in a noisy environment makes this technique not viable for this application. Also, it is shown that cells may be damaged by acoustic waves, and gas bubbles may disturb the measurement drastically [20]. As it is a demand that the sensor does not interfere with the process, the acoustic measurement is not adapted.

2.1.4. POINT MATRIX

In order to make a well considered decision, a point matrix is constructed for the possible solutions, as seen in table 2.1.

Table 2.1: Point matrix for possible solutions.

	Optical	Impedance	Microwave	Acoustic
Feasibility	0	0	- -	-
Range	0	+	+	-
Accuracy	0	+	+ +	0
Scalability	0	0	-	-
Distinguish life/death	0	+	+	0
Sterilisable	0	0	+	0
Total score	0	3	3	-3

The optical technique is set as a reference, since it is the current market standard. Impedance scores better than optical in accuracy and is theoretically able to distinguish between living and dead cells [7, 8]. Also, the impedance technique is accurate in higher cell concentrations than optical [7, 11], which is more accurate at low concentrations [1, 3]. Microwave is not feasible since the equipment required is not scalable and it uses very high frequencies[15], which make the data hard to process and parasitic effects will be strong. Its total score is high, but since it is not feasible, it is not chosen as a solution for this project. Acoustic performs worse on many aspects, due to its mechanical nature [20]. Mechanical signals are harder to read out, susceptible to the noise in the spinning environment and hard to scale down to 24 individual micro-bioreactors.

Taking all this in consideration, a choice is made for both an optical sensor and an impedance sensor. The optical sensor will be more accurate in low cell concentrations, and the impedance sensor will be able to measure higher concentrations. Within the optical sensor, both reflection and absorbance is measured for a more accurate measurement. The signals from both sensors are fed to the microcontroller which processes them to result in a cell concentration. This value will be communicated to a computer which will display the results. The system structure will be fully described in chapter 3.

2.2. SUBSYSTEM: SOFTWARE AND DATA PROCESSING

The system will measure the cell concentration with an optical and an impedance method. These two methods will be worked out as two separate sensor modules in the total system. The impedance sensing method will provide a reference voltage signal and an output voltage. The task of the software is to determine both signals' amplitude and the phase difference between them in order to calculate the impedance, which can be directly related to the cell concentration. The optical method determines the reflected and absorbed light with the help of light emitting diodes (LEDs) and photo diodes. The light reflection and absorption can also be related directly to the cell concentration and this information will be processed in the microcontroller.

This thesis covers the software and data processing part of the measurement system. This section will elaborate on formulating the problem and its demands.

2.2.1. DESCRIPTION OF THE PROBLEM

The microcontroller is the core of the data processing. In order to choose a suitable one, several aspects need to be considered. Acquiring the analog data and converting it to digital signals without losing precious information is essential. To convert continuous data, an analog-to-digital (A/D) converter that meets the specifications from the sensors needs to be found. Specifications such as conversion rate and resolution will be crucial for this choice. Another point which needs to be considered, is the operating clock frequency and communication ports of the microcontroller, which are to be used to communicate with the A/D converter and the computer. If the processor operates on a clock frequency which is too low for the A/D converter, the processor cannot keep up with the output data of the sensors and data could get corrupted or could be missed. It is also an option to use a microcontroller with a built-in A/D converter. The only research that would remain is to see if the specifications given by the sensors are met.

Secondly, for the data acquisition part of the system, an analog anti-aliasing filter needs to be implemented. This filter's purpose is to remove high frequencies which are too high for the A/D converter to be sampled. It is used to limit the bandwidth, so high frequency noise is suppressed.

Thirdly, an algorithm needs to be implemented to process the data to a form which makes sense to a user: a cell concentration. The algorithm should contain a digital filter to suppress noise present in the data after conversion. Also, this algorithm needs to be timed precisely to take accurate measurements. The digital filter will be designed according to the signal's frequency. Several decisions will have to be made with regard to the construction of this digital filter such as filter type, order and cut-off frequency. Besides a filter, the algorithm should also contain routines to convert the DC and AC signals (given by both sensors) into actual cell concentrations.

The final part of the problem definition is the presentation of the processed data: the user interface. The cell concentration data needs to be presented in a way where the different reactor conditions together with the cell concentration should be as clear as possible for analysis by potential users.

2.2.2. DESIGN BRIEF

The software and data processing has a specific design brief in order to meet the main design brief as given in subsection 2.1.2. This specific design brief will form the base of the software subsystem in the total cell concentration sensor. In this subsection a more specific explanation per requirement will be given.

- **The sensor must be able to measure the concentration of cells**
  For the software subsystem, this means that data must be acquired and processed to calculate a cell concentration.

- **The sensor read out should be done at least once per hour**
  This demand is similar to the total system's. The total time from acquiring the signals to outputting the cell concentration should take less than 2.5 minute, as 24 measurement cells need to be measured in one hour.

- **The system needs to be able to be implemented in the micro-Matrix.**
  The subsystems of the micro-Matrix communicate with each other through the Controller Area Network (CAN) bus. This information was provided by personal correspondence with Timo Walvoort, prject manager at Applikon Biotechnology. This system should also fit in, and be able to communicate through this bus.

- **The processing of data into cell concentration information should only be performed in the microcontroller.**
  This is a special requirement for the software and processing subsystem. All computations need to be performed on the microcontroller, so the output is purely a measure of the cell concentration. As the previous demand states it should be implemented in the micro-Matrix, it should not occupy the bus constantly, but only communicate the cell concentration.

- **The used technology has to fit in the measurement cells of the micro-Matrix having a working volume of 5 ml.**
  As the system needs to be ultimately implemented in the micro-Matrix, its size should be limited. For the proof of concept, this means that large boards can be used, but the possibility must exist to scale the system to a size suitable for integration in the micro-Matrix. It is noted that this demand is more of a limitation for the sensor-teams, as the probes and electrodes need to be able to fit inside a beaker holding the suspension. For the software, this means that the used components need to fit in the micro-Matrix, so there is limited space.

- **The input signals of the sensors should be in the range 0-3.3V**
  To process data into a useful form the Arduino Due was chosen as microcontroller. The reason why this specific microcontroller is chosen will be explained throughout this thesis. As the Arduino microcontroller operates at 3.3V, its A/D converter has a range of 0-3.3V. Consequently, it was agreed upon at the beginning of the project that the signals from both sensors should be in this range.

- **The phase for the impedance sensor should have a resolution of $0.1\,^{\circ}$**
  The smallest noticeable change in phase should be $0.1\,^{\circ}$. Because the changing cell concentration results in a small change of phase difference, this difference should be measured accurately. So in order to detect different amounts of cell concentration, the phase should have the described phase resolution.

3

SYSTEM OVERVIEW

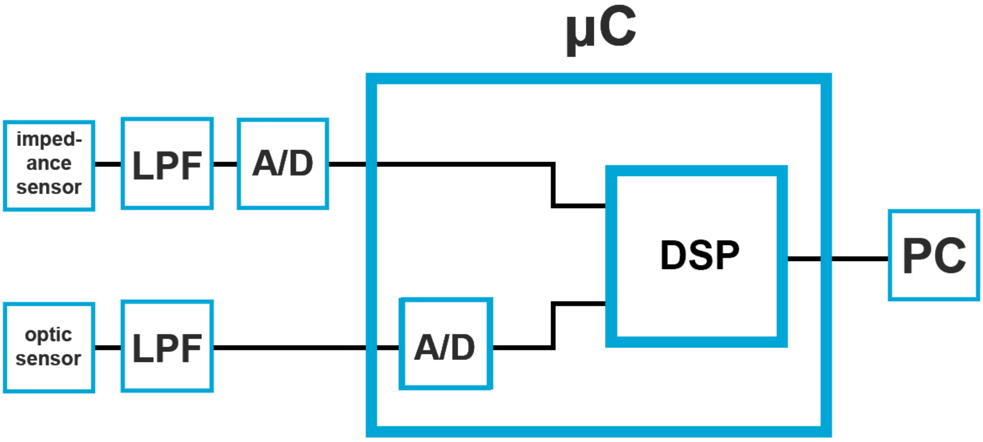


Figure 3.1: Schematic overview of the system

This chapter gives a schematic overview of the system. The purpose of this chapter is to give an idea of the system's architecture and to give an explanation for this specific architecture. Also, the rest of the thesis is structured using this block diagram.

The system contains the two sensors for measuring the cell concentration, two anti-aliasing filters for A/D conversion preparation and a microcontroller for signal processing. When a cell concentration is determined, it is transferred through USB to the computer, which displays the data in a graphical user interface. A block diagram of the architecture can be seen in figure 3.1. It is noted that this is a very abstract diagram. For instance, both sensors supply two signals each, but only one connection is displayed in the figure. It serves solely as a general overview.

As microcontroller, an Arduino Due is used because the Arduino systems are easy to use for prototyping purposes. The Due features a faster clock than most other Arduinos, which is beneficial for signal processing. Also, the microcontroller used in the Due features CAN bus [21], which is the way subsystems communicate in the micro-Matrix. The needed requirements for the system will be compared with the specs of the Arduino Due to confirm this choice.

3.1. OPTICAL DATA PATH

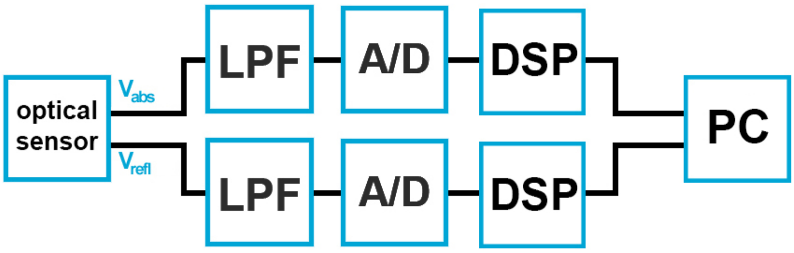


Figure 3.2: Block diagram for the optical data path

This section will give a short overview of the signal paths of the optical sensor, depicted in figure 3.2. In this figure it can be seen which stages the absorbance (V_{abs}) and reflection (V_{refl}) signals pass through. Extended motivations and explanations for these choices can be found in chapter 4.

After anti-aliasing, the choice is made to use the A/D converter of the Arduino, because the incoming signals are DC and the conversion rate and accuracy of the A/D converter of the Arduino is sufficient for this input. This choice is more elaborated upon in section 4.3.

For accurate data processing, a digital filter will be needed to eliminate noise on the DC signals. In this case, averaging is used as a method of filtering, as it is highly effective for obtaining the DC signals.

After averaging, the signals are translated to a cell concentration. Finally, this value is communicated to the computer.

3.2. IMPEDANCE DATA PATH

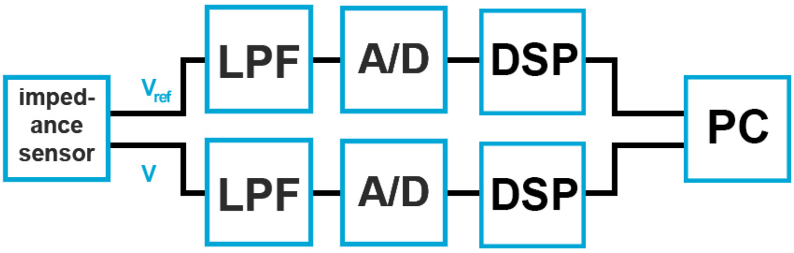


Figure 3.3: Block diagram for the impedance data path

This section will provide an overview of the data path of the signal coming from the impedance sensor, as seen in figure 3.3. In this figure, it can be seen which stages the reference (V_{ref}) and the output (V) signals pass through. An elaborate explanation on the different blocks can be found in chapter 5.

After anti-aliasing, an external A/D converter is used which can sample two signals simultaneously. This is required to accomplish an accurate measurement of the phase difference.

The A/D converter communicates with the microcontroller using the SPI protocol. After this acquisition of data, the signal is filtered using an Infinite Impulse Response (IIR) filter to suppress noise on the signal.

After filtering, the signals are processed to retrieve the amplitude and phase of the signals. Using these values, the impedance can be calculated which will be a measure for cell concentration.

4

Optical data path

This chapter will focus on the signal path given by the optical sensor (see figure 4.1). Each block in this data path will be further explained throughout this chapter.

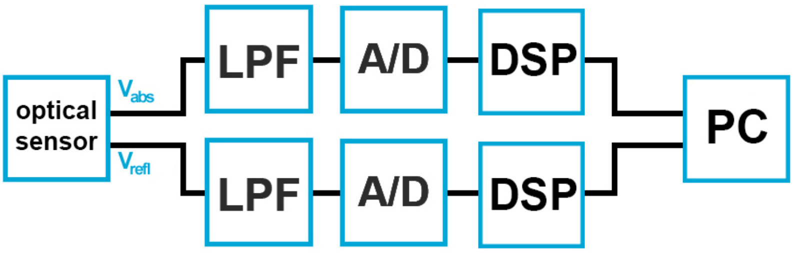


Figure 4.1: Block diagram for the optical data path.

4.1. Sensor signal

The optical sensor provides reflection (V_{refl}) and absorption (V_{abs}) data of cells by putting a probe in a suspension. An abstract overview of the measurement setup is displayed in figure 4.2. The probe contains a photo diode which collects light of a certain wavelength. This photo diode is defined as photo diode one. The light is emitted by an LED which is placed beneath the beaker holding the suspension. Also, a second photo diode is placed beneath the beaker which we will define as photo diode two. The amount of light that is reflected or absorbed can be measured by these two photo diodes. Photo diode one is placed in the probe to measure the optical density of the suspension and photo diode two is placed underneath the beaker to measure the reflection of the light. These photo diodes produce small currents which will need to be amplified before passing on to the next stage in the data path. When this data is converted into digital data, it will be processed into a cell concentration by using an algorithm based on calibration curves (see 4.5 and 4.6) given by the optical sensor team.

The two data signals (reflection and absorbance) are DC with an input range of 0 - 3.3V. This input range is needed because the micrcocontroller uses its A/D converter which operates at these voltages. This A/D converter will be further discussed in section 4.3.

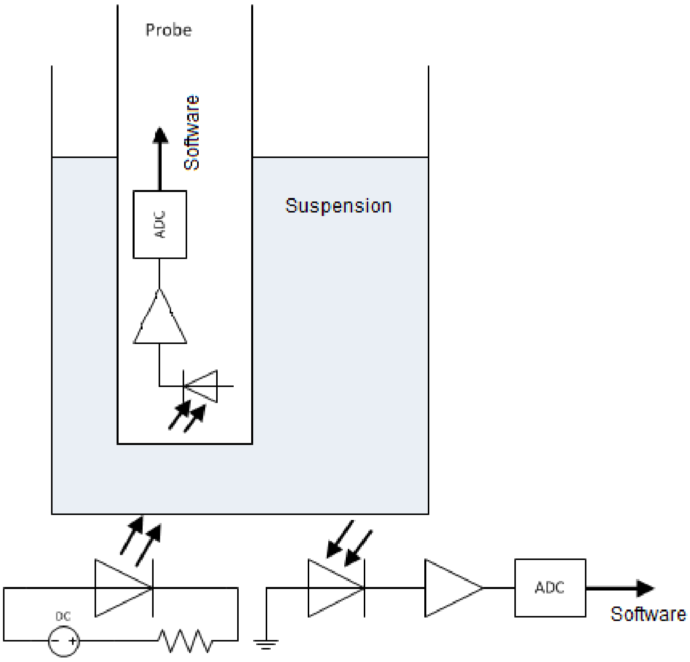


Figure 4.2: An abstract overview of the measurement setup for the optical measurement method.

4.2. ANTI-ALIASING FILTER

Before any A/D conversion can take place, an anti-aliasing low pass filter is needed to suppress noise at high frequencies. This filter is designed by the optical sensor team and implemented in their circuit [22]. An ideal anti-aliasing filter $H_{aa}(f)$ will pass the signal till $f = \frac{f_s}{2.2}$, where f_s = 75kHz is the sampling frequency, and suppress signals after this frequency [23, p. 143]. In theory, the Nyquist criterium states that a sampled signal can be fully recovered when the sampling rate is (more than) twice as high as the bandwidth of the signal. However, in practice it is wise to create some margin for the system. This is the reason why 2.2 is used instead of 2, as the theory says. The ideal anti-aliasing filter is thus given by:

$$H_{aa}(f) = \begin{cases} 1 & \text{for } f < \frac{f_s}{2.2} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The optical sensor will provide DC data signals, so any spectrum component at a higher frequency can be seen as noise which should be suppressed. For this relatively simple application, a first order low pass filter was designed. The rolloff of 20 dB per decade is sufficient as the noise present on the signal is at very high frequencies, and has a low magnitude of itself. This anti-aliasing filter will be implemented by the optical sensor duo. The components that were used to build this filter are:

$$R = 1k\Omega$$

$$C = 1\mu F$$

which yields a cut-off frequency of:

$$f_c = \frac{1}{2\pi RC} = 159.2Hz \tag{4.2}$$

This cut-off frequency is low enough to prevent aliasing when $f < \frac{f_s}{2.2}$ holds (Nyquist criteria).

4.3. A/D CONVERTER

The Arduino Due micro controller has 12 analog input pins and one built-in A/D converter which operates with 12 bits of resolution. A multiplexer is used to handle multiple analog to digital conversions when more than one analog signal is supplied to the Arduino (see figure 4.3). This means that there is a finite switching time to select the proper channel which needs to be sampled. In theory, it is possible to convert data with a conversion rate up to 1 Msps (10^6 samples per second).

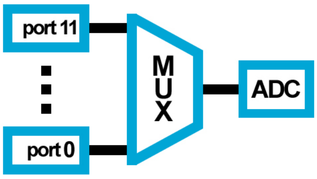


Figure 4.3: A/D converter multiplexer in Arduino

Serial communication with other devices (e.g. user interface computers), takes place over two separate USB lines, and integrated serial communication ports such as I^2C and SPI are included. The Arduino Due also uses the standard CAN (Controller Area Network) bus protocol which allows to communicate with other connected devices to this bus without a host computer.

Its processor, an Atmel processor with an ARM core, operates with a 84 MHz crystal oscillator. A maximum voltage of $V_{max} = 3.3V$ is the voltage limit the I/O pins of the Arduino Due can handle. As a result, the communicating sensors need to provide their analog information between a range of $0 - V_{max}$. There is also Flash memory and SRAM available which are respectively required for the code (our algorithm that needs to be implemented) and data storage (e.g. phase, amplitude, reflection and absorption variables). The flash memory contains 512 kBytes of memory for the code/algorithms and the SRAM contains 96 kBytes of memory for data storage. A short summary of the Arduino microcontroller's specs is given in table 4.1 and for the Arduino A/D converter in table 4.2 (for more detailed information, consult [21]).

Table 4.1: General specs of the microprocessor of the Arduino Due

Feature	Value
Clock speed	84 MHz
Flash memory	512 KBytes
SRAM	96 KBytes
Communication ports	I^2C, SPI, CAN and USB

Table 4.2: Specs of the A/D converter of the Arduino Due

Feature	Value
Resolution	12 bits
Maximum Full Scale Input Range	0 - 3.3 V
Data conversion rate	1 Msps
Allows simultaneous sampling	No
SNR = 6.02N + 176	74 dB
Max gain error	+12 LSB
Min gain error	-64 LSB
Max offset error	+48 LSB
Min offset error	-22 LSB

The offset error is the deviation of the transition from discretization level zero (M = 0) to discretization level one (M = 1). This error applies to the output coding. The gain error is the deviation of the transition to the last discretization level (from $M = 2^{12} - 1$ to $M = 2^{12}$).

As can be seen, the specifications of the errors of the A/D converter are widely varying. For this reason, the A/D converter was calibrated to ensure accurate operation. The A/D converter is first tested with known signals which are produced by a voltage supply. When these signals are applied to the inputs of the A/D converter, it can be determined whether an internal error was made by subtracting the measured voltage value from the known voltage value. The results and the methodology of the calibration measurement of the Arduino A/D converter can be found in section 7.1.1.

4.4. DIGITAL FILTER

When converting analog signal into digital signals, code-transition noise could possibly be added to the signal. When the value of an analog signal changes, then a new discretization level (with an unique bit code) will be assigned. This transistion in discretization levels can be seen as noise source.The DC signals should be kept stable after A/D conversion. Therefore, some kind of filtering is necessary. A relatively simple but effective way to perform this kind of filtering is to average the acquired data. The system will take $N = 1000$ data points and calculate the mean μ in order to stabilize the DC signals obtained from the optical sensor. The routine will sum up to N points to subsequently multiply the sum with $\frac{1}{N}$:

$$\mu = \frac{1}{N} \cdot \sum_{i=0}^{N} V_i \tag{4.3}$$

$N = 1000$ data points are chosen, because this will be sufficient to keep the data stable. This averaging operation can be thought of as a filter with the following impulse response:

$$h[n] = \begin{cases} \frac{1}{N} & \text{for } 0 \le n < N \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

Applying the discrete-time Fourier transform on this impulse response, the following is obtained:

$$H(\omega) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n} = \frac{1}{N} \sum_{n=0}^{N-1} e^{-j\omega n} \tag{4.5}$$

where $\omega = 2\pi \frac{f}{f_s}$. Since

$$\sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \tag{4.6}$$

this is further simplified to

$$H(\omega) = \frac{1}{N} \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} = \frac{1}{N} \frac{e^{-j\omega N/2}}{e^{-j\omega/2}} \frac{e^{j\omega N/2} - e^{-j\omega N/2}}{e^{j\omega/2} - e^{-j\omega/2}} \tag{4.7}$$

Then, using Euler's formula [24, p. 104],

$$\sin(\omega) = \frac{e^{j\omega} - e^{-j\omega}}{j2} \tag{4.8}$$

the transfer function can be written as

$$\begin{aligned} H(\omega) &= \frac{1}{N} \frac{e^{-j\omega N/2}}{e^{-j\omega/2}} \frac{j2\sin(\frac{\omega N}{2})}{j2\sin(\frac{\omega}{2})} \\ &= \frac{1}{N} \frac{e^{-j\omega N/2}}{e^{-j\omega/2}} \frac{\sin(\frac{\omega N}{2})}{\sin(\frac{\omega}{2})} \end{aligned} \tag{4.9}$$

Since it is really the magnitude of the transfer function which is of interest, the modulus of the above is taken to find:

$$|H(\omega)| = \frac{1}{N} \left| \frac{\sin(\frac{\omega N}{2})}{\sin(\frac{\omega}{2})} \right| \tag{4.10}$$

since $|e^{j\omega}| = 1$.

The magnitude response of the averaging filter is evaluated and displayed in figure 4.4.

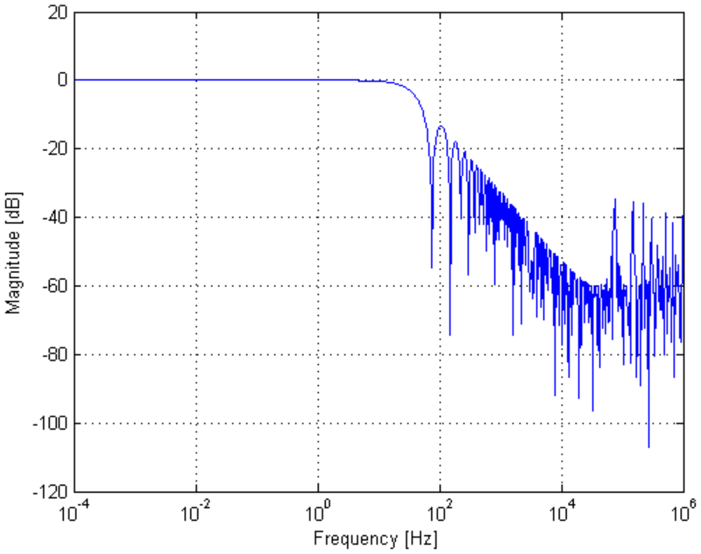


Figure 4.4: Magnitude response of the averaging filter

It can be seen that low frequencies are unaffected, and for frequencies beyond 10 Hz, the filter starts attenuating. High frequency noise beyond 10 kHz is attenuated with about 60 dB.

Because this filter is so simple in terms of implementation, there are no separate measurements performed with this filter. The eventual optical routines will be tested according to the plan that will be described in subsection 7.3.1.

4.5. ABSORBANCE ROUTINE

To convert the measured voltage to a cell concentration, some calibration measurements need to be carried out to form a model. The calibration curves will show the correlation between the measured DC signals and cell concentrations. The next step is to derive a mathematical formula which can be used in the software. In [22], two fellow students of the project group derived such a model. In figure 4.5, the measurements of two measurement series are displayed.

A first order polynomial is fitted, and this formula is given by

$$\rho_{abs} = -46.8360 \cdot V + 154.6451 \tag{4.11}$$

and has $R^2 = 0.9730$. This model will be used in the software to convert a measured voltage to a cell concentration.

4.6. REFLECTION ROUTINE

Similar to section 4.5, a relation between the measured reflection signal and the cell concentration needs to be found.

In [22], a model was derived which does exactly this. The data of two measurements is taken into account and a linear curve is fitted, as displayed in figure 4.6.

The relation between voltage and cell concentration is given by

$$\rho_{refl} = 19.2756 \cdot V - 5.8350 \tag{4.12}$$

This model has $R^2 = 0.9625$, and will be used in the software to calculate the cell concentration.

To test both this relation and the relation of the absorption signal, a test plan is made in subsection 7.3.1.

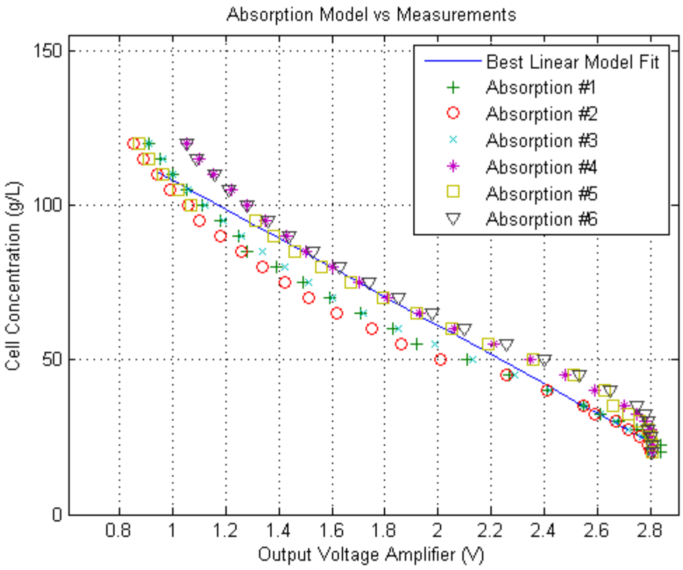


Figure 4.5: Absorbance calibration curve

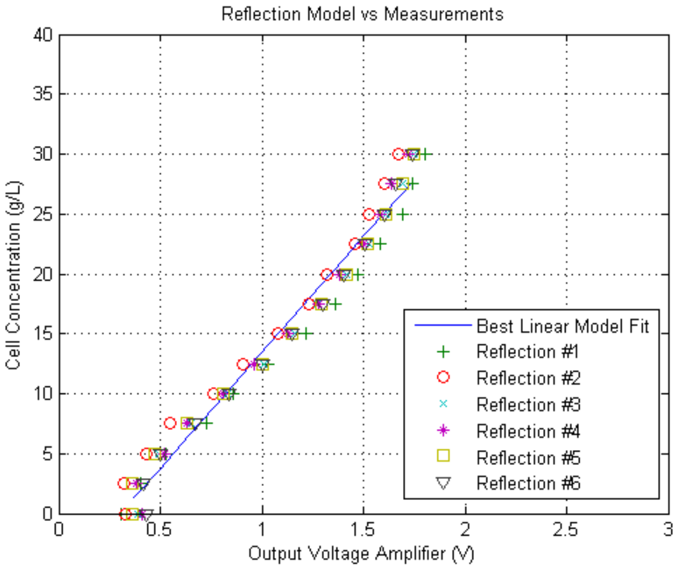


Figure 4.6: Reflection calibration curve

5

Impedance data path

This chapter will explain the signal path of the impedance sensor, as seen in figure 5.1. Each step is elaborated upon in a section.

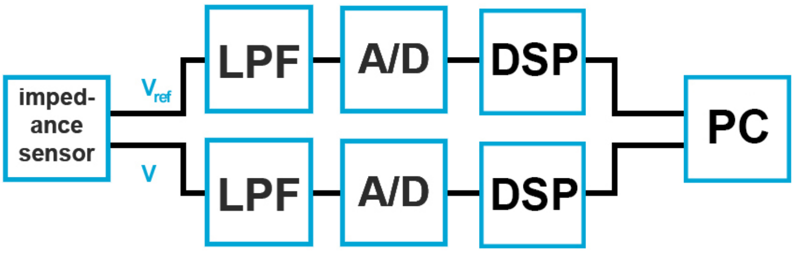


Figure 5.1: Block diagram for the impedance data path

5.1. Sensor signal

The signals coming from the sensor are the reference and output voltages accross an impedance, which is formed by two electrodes positioned on the side of a beaker containing the suspension with the cells. The input signal is produced by a function generator, at 1 kHz. Then the output voltage over the impedance is measured, and scaled to 0 – 3.3 V. The objective is to measure the impedance, by measuring the amplitude of the output and reference voltages by using $|Z| = \frac{|V|}{|I|}$, where $|I|$ is constant at 0.5 mA, and measuring the phase of the output voltage relative to the reference signal. The algorithm for measuring the phase will be discussed in section 5.5. With this information, the imaginary and the real part of the impedance can be calculated, as illustrated by figure 5.3, which can be related to the cell concentration with the use of calibration curves given by two fellow students of the impedance team [25].

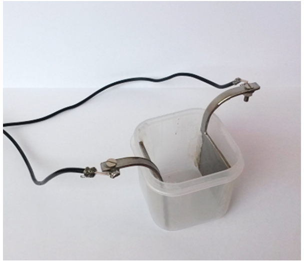


Figure 5.2: Setup for the impedance measurements

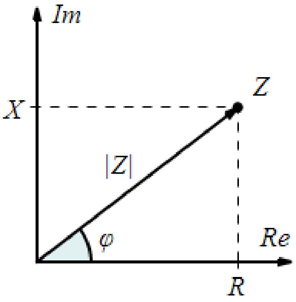


Figure 5.3: Complex impedance Z with phase ϕ

5.2. ANTI-ALIASING FILTER

Similar to the optical data path, an anti-aliasing low pass filter is needed to prevent high frequency signals from folding over to lower frequencies, which will disturb the measurement. This filter is included in the circuit designed by the impedance sensor team. In this data path, a simple lowpass RC filter is implemented with

$$R = 1k\Omega$$

$$C = 22nF$$

This yields a cutoff frequency of

$$f_c = \frac{1}{2\pi RC} = 7.2kHz \tag{5.1}$$

Since the outputs of the sensor are 1 kHz signals and the sampling frequency is 75 kHz, the anti-aliasing filter also lowers the signal bandwidth. Since it is a first order filter, the attenuation of high frequency noise is limited. A digital filter will be implemented to limit noise further, as explained in section 5.4.

5.3. A/D CONVERTER

To measure the phase accurately, both the reference and output voltage need to be measured simultaneously. If this is not the case, a phase difference will occur by the time difference between the two measurements. This phase difference is calculated by

$$\theta = 360° \cdot f \cdot \Delta t \tag{5.2}$$

This is illustrated in figure 5.4. Here, the same 1 kHz signal is applied to two analog input pins of the Arduino. The acquired data is displayed in the figure. As seen, the multiplexer before the A/D converter introduces a phase difference which does not really exist, as both signals are directly connected to a function generator.

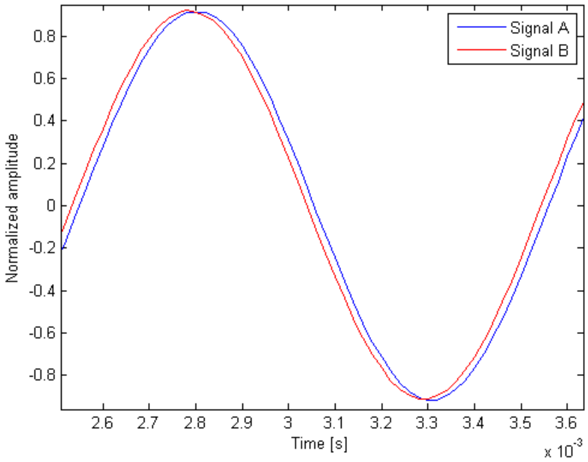


Figure 5.4: Phase difference of subsequent sampling using the Arduino's A/D converter.

Simultaneous sampling is impossible using the A/D converter of the Arduino, as the multiple analog inputs are multiplexed to the A/D converter. This is not a problem when the time difference is constant, as this constant error can be easily subtracted from the calculated phase difference. However, due to irregularities in the Arduino's command's timing, this delay is not constant. A test is performed where the same signal is applied to two analog input pins, and both signals are subsequently sampled. The results of this experiment is displayed in a histogram in figure 5.5, and in a table in appendix A.2.

It can be seen that the phase difference is widely spread from the range $3° - 14°$. As the phase measurement must be accurate to $0.1°$, this spread is unacceptable.

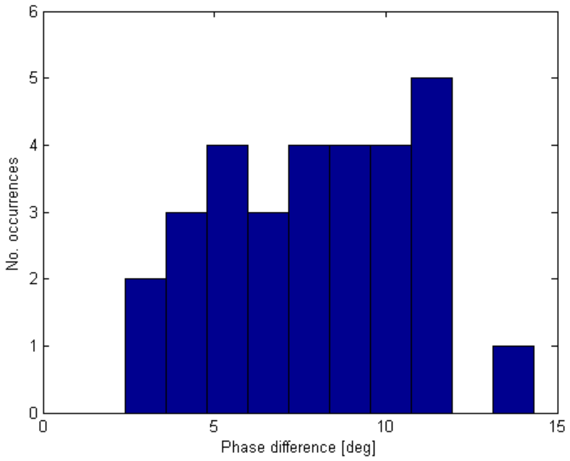


Figure 5.5: Histogram of the phase difference of subsequent sampling using the Arduino's A/D converter.

Therefore, an external A/D converter is used, the AD7266 [26]. This A/D converter can simultaneously sample two signals and communicate their values with the microcontroller through Serial Peripheral Interface (SPI), which will be further explained in section 5.3.1. A short summary of the most important specs is given in table 5.1.

Table 5.1: Specs external A/D converter, the AD7266

Feature	**Value**
Resolution	12 bits
Supply voltage (V_{dd})	$2.7-5.25$ V
Maximum Full Scale Input Range	$0-V_{dd}$ V
Data conversion rate	1.5 Msps
Simultaneous sampling channels	2
SNR = 6.02N + 1.76	74 dB
Max gain error	+2.5 LSB
Max negative gain error	-2.5 LSB
Max offset error	+7 LSB
Max negative offset error	-7 LSB

The AD7266 is used on its evaluation board, the EVAL-AD7265/AD7266 [27]. This board features various peripheral devices to ensure proper functioning of the AD7266 in various configurations. It is used as this allows for fast prototyping, as the whole peripheral circuit does not need to be designed. An external voltage reference IC is used to create a stable 3.3V reference: the ISL21010-33 [28]. This IC is fed by the Arduino's 5V supply and creates a stable reference voltage at 3.3V. A decoupling capacitor $C_l = 1\ \mu F$ is used to ensure stable operation. In figure 5.6, this circuit is displayed. C_b is implemented on the Arduino board.

Lastly, the results and methodology to test the linearity of this external A/D converter can be found in section 7.2.1.

5.3.1. Serial Peripheral Interface (SPI)

SPI is a serial synchronous low level communication protocol. The SPI protocol is loosely specified, as features such as clock frequency can be chosen freely. Also, the rules for transmission are to be chosen by the designer. Clock polarity (CPOL) and clock phase (CPHA) can be set to determine the so called SPI mode. This mode sets when data is sampled from the data line. Peripheral devices are commonly limited to one or two of these modes, which are used to choose which mode is taken for the whole system. The AD7266 uses SPI mode 3, where CPOL = 1 (clock is idle high) and CPHA = 1 (data sampled on the trailing, rising edge of the clock).

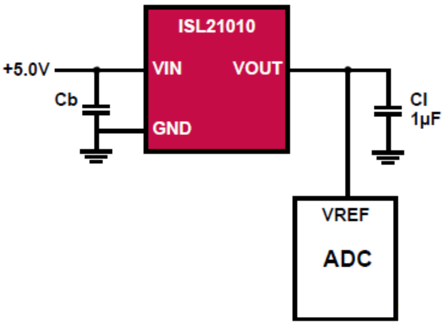


Figure 5.6: Circuit of the ISL21010-33 [28]

SPI uses four communication lines: a clock line (CLK), a master input and slave output line (MISO), a master output and slave input line (MOSI) and a slave select line (SS, active low), sometimes referred to as chip select (CS). Each slave uses an individual slave select line.

SPI can only have one master node. Because the MISO line is shared, slave devices have tri-state outputs so their MISO signal becomes high impedance when the slave is inactive. This allows for multiple slaves using a single MISO and MOSI line, as the inactive slaves do not influence the signal.

A start bit is unnecessary as the slave is activated when the slave select line is pulled low. During each subsequent clock cycle, the master sends a bit via the MOSI line which is read by the active slave. Also, a bit is sent by the slave to the master via the MISO line. In communicating with the AD7266, the MOSI line is not in use. The ADC will start conversion and transmission when the SS line is pulled low.

When the slave select line is pulled low, the AD7266 starts conversion and after two clock cycles, the data is transmitted. A schematic overview of communication is given in figure 5.7.

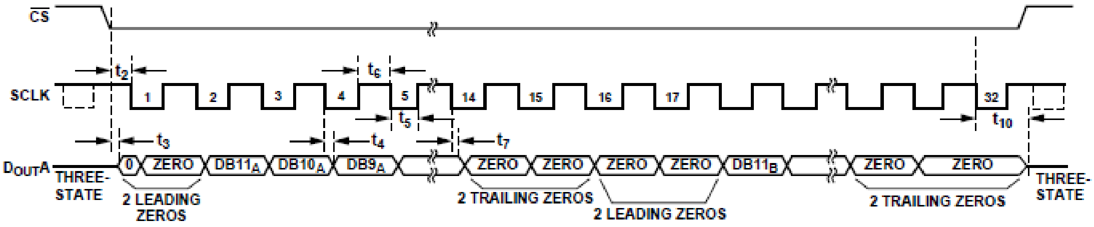


Figure 5.7: Schematic overview of transmission of data from both analog channels through one data output [26].

The serial clock line becomes active when the slave select line is pulled low. After two leading zeros, the 12 bit data is transmitted MSB first. Then, two trailing zeros complete the two-byte message. If the slave select line stays low and the clock continues switching, the information from the second analog input is also transmitted, in the same format.

Because SPI uses a shift register of one byte, the four bytes are individually stored in bytes and have to be manipulated to form the correct 12-bit numbers by the following operation:

```
dataA = byte0 << 6 | byte1 >> 2;
dataB = byte2 << 6 | byte3 >> 2;
```

In 32 clock cycles, four bytes are transmitted containing two 12 bit values. Using a serial clock frequency of 4 MHz, this transmission takes:

$$t_{conversion} = T_{SCLK} \cdot \#\text{bits} = \frac{1}{4 \cdot 10^6} \cdot 32 = 8\mu s \tag{5.3}$$

This communication is done using an interrupt at 75 kHz, in order to sample fast enough to obtain a representative result. With 75 samples per period, enough information points are available to reproduce the original signal. Also, 75 kHz is fast enough to take many samples and process these within a short period of time. Consequently, the interrupt routine is called every $1/75 \cdot 10^3 = 13.3\ \mu s$. With 8 μs for data transmission, about 5 μs is left for converting the bytes to the correct data format and other operations inside the interrupt.

Every 13.3 μs two samples are taken, until 1050 samples are collected. This array of information is further processed. Section 5.5 will elaborate on why 1050 samples are taken, and not an other number.

5.4. DIGITAL FILTER

A digital filter is designed to eliminate noise on the signals which could be added after A/D conversion. Also a digital filter need to be designed to filter out the DC component of specific signals (e.g. offset of the mixed signal for the phase routine). In this section, several options will be discussed and a choice is made for the design of this filter.

5.4.1. IIR VERSUS FIR

When designing a digital filter, a choice must be made between an infinite impulse response filter (IIR) and a finite impulse response filter (FIR). This choice is usually based on whether the system is required to have linear phase response [23]. Linear phase response implies that the delay of the filter is the same at all frequencies. Therefore, a filter having linear phase response does not cause phase distortion.

In general, only an FIR filter can be stable, causal and have linear phase response at the same time. If linear phase response is not required, either an FIR or IIR filter can be implemented. In this case however, an IIR filter is preferable, because its implementation involves fewer parameters, requires less memory and has lower computational complexity [29, p. 612]. In this application, there is no need for a linear phase response. The signal from the impedance sensor is at a fixed frequency, so different delays at different frequencies are not a problem. Therefore, the choice is made to implement an **IIR filter**.

5.4.2. FILTER TYPE

When designing a digital filter, an analog filter is designed first, and then a digital filter is implemented that matches the analog filter's characteristics. Section 5.4.3 will elaborate on this process. A handful of basic standard analog filter types can be distinguished. In this section, some will be briefly mentioned and the implemented filter type is chosen based on each filter's characteristics.

The Butterworth filter is a filter designed to have a maximally flat response in the passband [30, p. 280]. Its transfer function is described by:

$$|H(\omega)| = \frac{1}{\sqrt{1+(\omega/\omega_c)^{2N}}} \tag{5.4}$$

in which ω_c is the -3dB cutoff frequency and N represents the order of the filter.

The Chebyshev filter is designed to have the steepest roll-off, at the cost of more ripple in the passband (type I) or in the stopband (type II). The transfer function of a type I Chebyshev filter is given by:

$$|H(\omega)| = \frac{1}{\sqrt{1+\epsilon^2 C_N^2(\omega/\omega_c)}} \tag{5.5}$$

The transfer function of a type II Chebyshev filter is given by:

$$|H(\omega)| = \frac{1}{\sqrt{1+(\epsilon^2 C_N^2(\omega_c/\omega))^{-1}}} \tag{5.6}$$

Here, ϵ is a design constant determining the amount of ripple and C_n is the Chebyshev polynomial of order N. The Chebyshev polynomial will not be discussed further, but it is noted that evaluating this polynomial adds complexity to the system. The Bessel filter attempts to maintain linear phase in the passband. It has the following transfer function:

$$H(\omega) = \frac{K_n}{B_n(\omega/\omega_c)} \tag{5.7}$$

where K_n is a constant to make $H(0) = 1$, and $B_n(x)$ is the Bessel recursion relation. This recursion relation will not be discussed further.

Besides these, an elliptical filter can also be implemented. This filter has adjustable ripple in the pass- and stopband. It also has the fastest transition in gain between the pass- and stopband. Its transfer function is given by:

$$H(\omega) = \frac{1}{1+\epsilon^2 R_n^2(\xi,\omega/\omega_c)} \tag{5.8}$$

where R_n is the elliptic rational function.

A great advantage of the Butterworth filter is that its implementation and design are relatively simple. Other filters have polynomials and recursive functions in their transfer function, which make these filters more difficult to implement [23]. Since our choice does not directly follow from the requirements (for instance, when a very steep roll-off is required it is obvious to choose Chebyshev), the choice is made to design a **Butterworth filter**, due to its relative simplicity.

5.4.3. DESIGN PROCESS

Knowing that a Butterworth IIR digital filter is a good choice for our goal, several design steps to accomplish the filter design have to be taken. This subsection will give the design steps for the digital filters that will be implemented in the micro controller. The filters will be built with the help of the design steps given by [23], [31] and [32]. For determination of the coefficients of the digital filters, the Filter Analysis & Design Tool (FDATool) is used, which is provided by the MATLAB software environment.

- **Step 1:** First, the cutoff frequency f_c of an analog Butterworth low pass filter will have to be determined. Later, the analog filter will be transformed into the digital Butterworth IIR low pass filter. At the cutoff frequency, the signal power is halved and begins to attenuate even further. The information signals (reference and output voltages) have a frequency of f_w = 1 kHz, so the filter needs to have a cut-off frequency $f_c > f_w$ = 1 kHz, because the signal information has to remain unaffected by the filter.
- **Step 2:** The next step is to determine the filter order to be used. The filter order gives an indication of the "sharpness" of the filter after passing the cut-off frequency. A "sharper" filter means that more decibels will decrease per decade. The filter order also raises the filter complexity (the amount of filter sections), so an optimum between enough suppression and filter complexity has to be found.
- **Step 3:** The last step is to transform the designed analog filter into a digital filter. The digital filter will be built in a Direct Form II realisation and the coefficients of the digital filter need to be determined. The Direct Form II realisation will later on be converted into implementable code which will be programmed on the microcontroller.

DESIGN STEP 1

Because the cut-off frequency needs to be greater than the signal frequency ($f_c > f_w$), a cutoff frequency of f_c = $2 \cdot f_w$ = 2 kHz is chosen. This frequency is well above the 1 kHz of the information signals, and near enough so that high frequencies will be attenuated.

DESIGN STEP 2

A fourth order filter is chosen as the optimum between complexity and sufficient suppression. The higher the filter order, the more complex the filter's implementation. This means that more filter sections will be used for the Direct Form II realisation of the IIR filter. The more sections used in the filter, the more time it will take to filter the signals.

The magnitude response is displayed in figure 5.8. At 1 kHz, the filter does not suppress yet. This ensures that the signal remains intact. At higher frequencies, the filter starts to suppress the signals with up to 140 dB at 30 kHz.

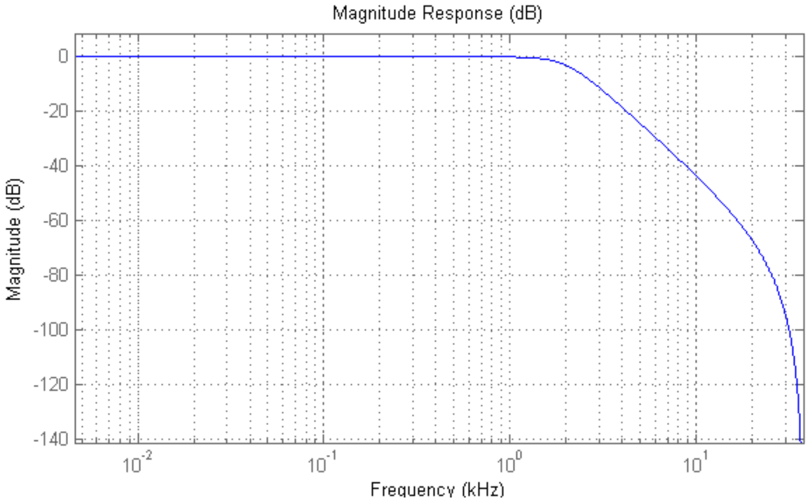


Figure 5.8: Frequceny response of the 2 KHz filter

In section 7.2 the results of the implemented filter can be seen. The used filter order was simulated and tested to verify our choice to select an optimum.

DESIGN STEP 3

It is now the goal to find the filter coefficients a_k and b_k such that digital filter's response matches the analog filter's ([23, p. 170]). A standard digital filter response with the unknown filter coefficients is given in equation 5.9.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 \cdot z^{-1} + \ldots + b_M \cdot z^{-M}}{1 + a_1 \cdot z^{-1} + \ldots + a_N \cdot z^{-N}} \tag{5.9}$$

For the determination of these coefficients MATLAB's FDATool was used which has a built-in algorithm to convert an analog low pass Butterworth filter (when the cut-off frequency and filter order are given) into a digital low pass IIR Butterworth filter in a Direct Form II realisation. A schematic circuit of the filter is given in figure 5.9. The values of the coefficients can be found in appendix A.4.

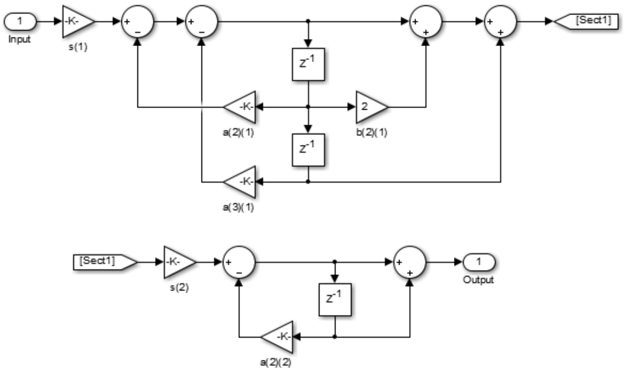


Figure 5.9: Realisation of the 2 kHz filter

This realisation was converted into a implementable code for the microcontroller. To test the digital filter, a measurement method is derived. This measurement method will test the functionality of the filter. A more specific description of this measurement method will be elaborated in 7.2.2.

The same principle was used and the same design steps were taken for the design of the DC offset filter.

Design step 1

Because offset components can be considered as a DC signals without any frequency components, the cut-off frequency should be chosen as close to $f = 0Hz$. This will filter out the mean of a oscillating signal which can be seen as the signal offset. In our case we have chosen to use a cut-off frequency $f_c = 50Hz$. The reason for this specific number is dependent of the filter order. As said before, the filter order will determine the delay for the filtered signal.

Design step 2

Just as in design step 2 for the first filter, we have chosen to use filter order 4. This gave enough time for the filter to stabilize the output and to give a constant value for the DC offset. In figure 5.10 the time response of the designed 50 Hz can be seen. It can be seen that this filter takes about 0.04s to stabilize.

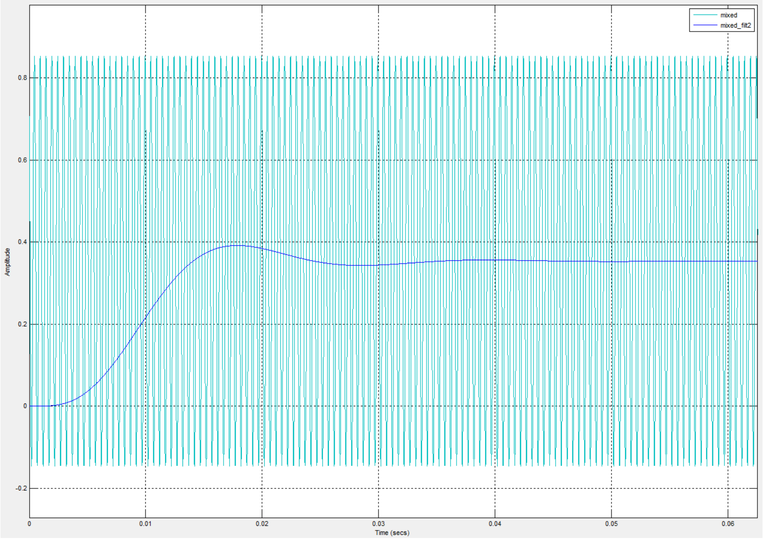


Figure 5.10: Time response of the 50 Hz Filter

Design step 3

This is design step is exactly the same as described for the previous filter. The coefficients of this 50 Hz filter are given in appendix A.5

5.5. Phase routine

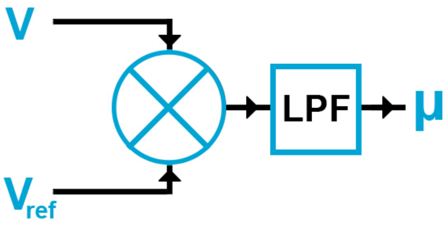


Figure 5.11: Schematic overview of the phase routine

In order to calculate the imaginary part of the impedance, the phase between the output voltage and reference voltage is calculated. This phase is obtained by multiplying the measured signals, as seen in figure 5.11. These periodic signals have the same frequency ω_0 and a phase difference θ. Multiplying these signals with amplitude $|V|, |V_{ref}|$ gives:

$$
\begin{aligned}
V(t)\cdot V_{ref}(t) &= |V|\cos(\omega_0 t)\cdot |V_{ref}|\cos(\omega_0 t+\theta) \\
&= \frac{|V||V_{ref}|}{2}\left[\cos(\omega_0 t-\omega_0 t-\theta)+\cos(\omega_0 t+\omega_0 t+\theta)\right] \\
&= \frac{|V||V_{ref}|}{2}\left[\cos(-\theta)+\cos(2\omega_0 t+\theta)\right]
\end{aligned} \tag{5.10}
$$

The product of these two signals results in a signal of double the frequency, with an offset of $\mu = \frac{|V||V_{ref}|}{2}\cos(\theta)$, since $\cos(-x) = \cos(x)$. The consequence of this identity is that only the absolute phase difference can be obtained using this method. However, since it is known that the measured impedance is capacitive, this is no limitation. To obtain the phase difference θ, the DC offset μ of the resulting signal must be taken, so the $\cos(2\omega_0 t+\theta)$ term is cancelled. From this offset, the phase difference can be obtained by

$$\theta = \arccos\left(\frac{2\mu}{|V||V_{ref}|}\right) \tag{5.11}$$

The offset μ is calculated by filtering the signal using a 4th order IIR low-pass filter with a cutoff frequency of 50 Hz (see figure 5.11). The filter is implemented in the same way as the filter in section 5.4. Its coefficients are given in appendix A.5. This filter makes sure the signal component on 2 kHz is filtered out, and the remainder is the DC offset μ.

5.6. AMPLITUDE ROUTINE

In order to measure the absolute impedance $|Z|$, the amplitude of the output voltage must be measured, and then divided by the amplitude of the current:

$$|Z| = \frac{|V|}{|I|} \tag{5.12}$$

where $|I|$ is constant at 0.5 mA.
To determine the amplitude of the signal, the root mean square is calculated, and multiplied by $\sqrt{2}$:

$$|V| = \sqrt{2}\cdot V_{RMS} = \sqrt{2}\cdot\sqrt{\frac{1}{N}\sum V_n^2}$$

where $N = 1050$. Both phase and amplitude routines will be tested according to a measurement method described in 7.2.3.

5.7. CELL CONCENTRATION ROUTINE

This section will describe the algorithm which is developed to convert the measured amplitude and phase to a cell concentration. For the full circuit and a derivation of the algorithm, see the original thesis [25].

Since the reference signal is only inverted once and the other signal twice, the phase has to be corrected for this 180° offset, and converted to radians:

$$\alpha = \frac{\theta - 180}{180}\pi \tag{5.13}$$

This is used to calculate the feedback impedance, which consists of the measurement cell in series with an extra capacitor C_s and the impedance of the wires Z_p. This impedance is displayed in figure 5.12.

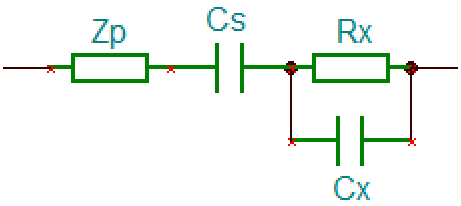


Figure 5.12: $Z_{feedback}$ in the impedance read out circuit

$$Z_{feedback} = \frac{|V|}{G \cdot |I|} \cdot e^{j\alpha} \tag{5.14}$$

where G is the gain of the amplifier after this circuit. Then the series impedance has to be subtracted:

$$Z_x = Z_{feedback} - Z_s \tag{5.15}$$

where Z_s is the equivalent combined impedance of Z_p and C_s. This impedance is measured to be $Z_s = 1.49616 - j1.55857 \cdot 10^2$ [25]. The impedance is then rewritten to extrahere the capacitance:

$$C_x = \frac{1}{2\pi f} \cdot \mathrm{Im}\left(\frac{1}{Z_x}\right) \tag{5.16}$$

This value C_x is averaged over 30 calculations using a moving average to achieve more constant and reliable signals:

$$\bar{C}_x = \frac{1}{30} \sum_{i=0}^{29} C_x^{n-i} \tag{5.17}$$

Through calibration the initial capacitance of the beaker with just water, $C_x(0)$, is determined so the capacitance change by the changing cell concentration is given by

$$\Delta C = \bar{C}_x - C_x(0) \tag{5.18}$$

Finally, through multiple measurements, the capacitance change is related to the cell concentration. The measurements, along with the fitted third order polynomial are displayed in figure 5.13.

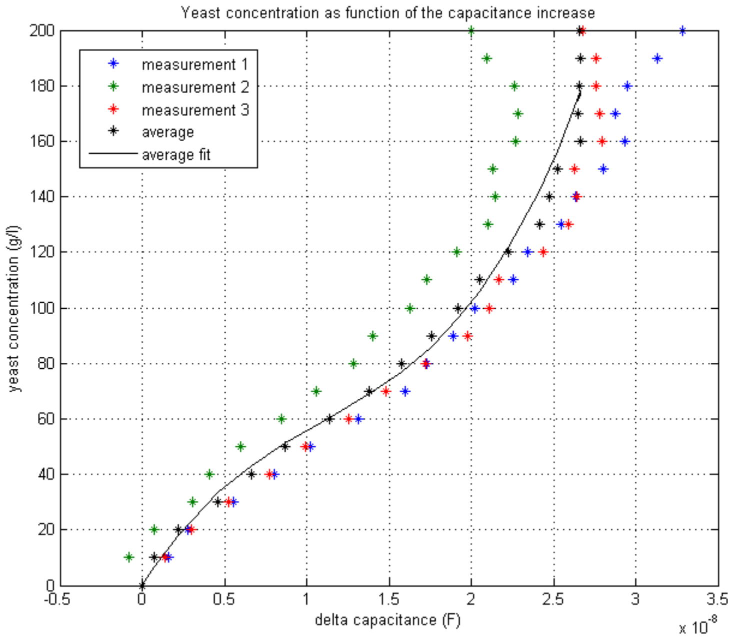


Figure 5.13: Impedance callibration curve

The fitted polynomial is given by

$$\rho = 1.71655 \cdot 10^{-25} \cdot \Delta C^3 - 5.64485 \cdot 10^{-17} \cdot \Delta C^2 + 9.51241 \cdot 10^{-9} \cdot \Delta C \tag{5.19}$$

This relation has $R^2 = 0.981$, and will be used in the software to relate the capacitance change to the cell concentration. On the next page (see figure 5.14), a schematic is given of the total system with its filters and phase/amplitude/cell concentration routines.

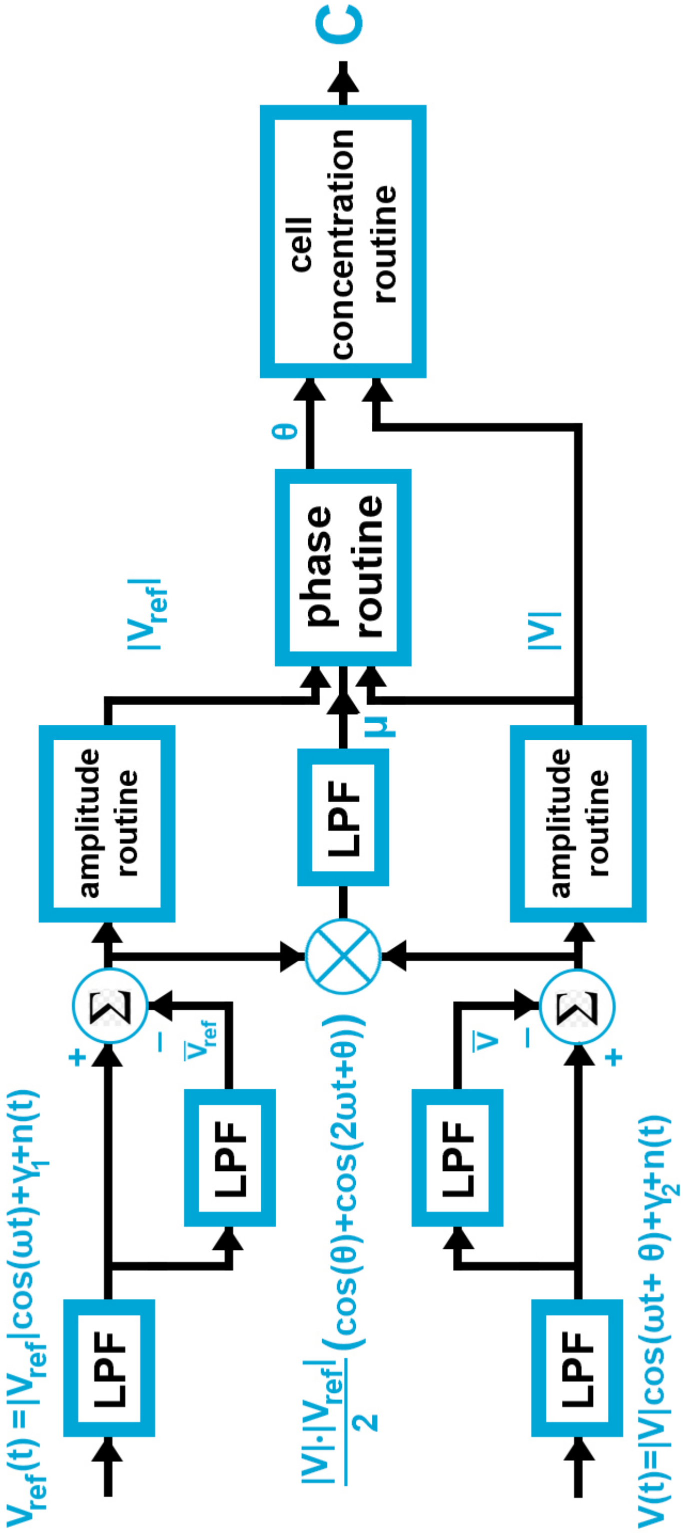

C
cell concentration routine
θ
phase routine
$|V_{ref}|$
$|V|$
μ
amplitude routine
LPF
amplitude routine
+
−
$\overline{V}_{ref}$
LPF
−
$\overline{V}$
+
LPF
LPF
LPF
$V_{ref}(t) = |V_{ref}|cos(\omega t)+\gamma_1+n(t)$
$\frac{|V|\cdot|V_{ref}|}{2}(cos(\theta)+cos(2\omega t+\theta))$
$V(t)=|V|cos(\omega t+ \theta)+\gamma_2+n(t)$

Figure 5.14: Schematic overview of the total processing system of the Impedance data path

6
Interface & Integration

This chapter will give the information for the integration of the total system including both optical and impedance sensors combined with micro controller. A test plan will be derived to test the implemented routines. The test will be described by a measurement method for real yeast concentration measurements. The test plan and results will be explained in chapter 7.

6.1. System integration

To give an overview of the program written in the Arduino, a flowchart is displayed in figure 6.1. After initialization of the communication ports and the digital filter, 2x1000 samples are taken of the output and reference voltage signals. This routine is time-critical, and an interrupt is enabled guarantee this precision.

Then, the signals are filtered to remove the noise. After this, the amplitude routine is executed to obtain $|V|$ and $|V_{ref}|$. Using these and the signals, the phase is determined. With these values, the impedance can be calculated and this value is related to a cell concentration. This value is communicated through USB to the computer, which displays the value.

Next, 1000 samples are taken for the absorbance sensor and 1000 for the reflection sensor. These signals are filtered by taking the mean to obtain the DC value. With these values, a cell concentration is calculated and sent to the computer, similar to the value obtained from the impedance sensor.

At the end a timer is implemented for the program to wait until the next measurement should be taken. In the proof of concept, this is not of the essence, but when integrated in the micro-Matrix, all 24 wells will be measured subsequently. Then, a timer waits until an hour (or shorter, depending on the configuration) has passed, after which all wells are measured again. After measuring all cell concentrations, the timer waits. This ensures that the measurements are as close to each other as possible, so the process can be accurately monitored. An other option would be to measure one well each 2.5 minute. However, all measurements would have a substantial time difference between them, so the data can not be optimally compared.

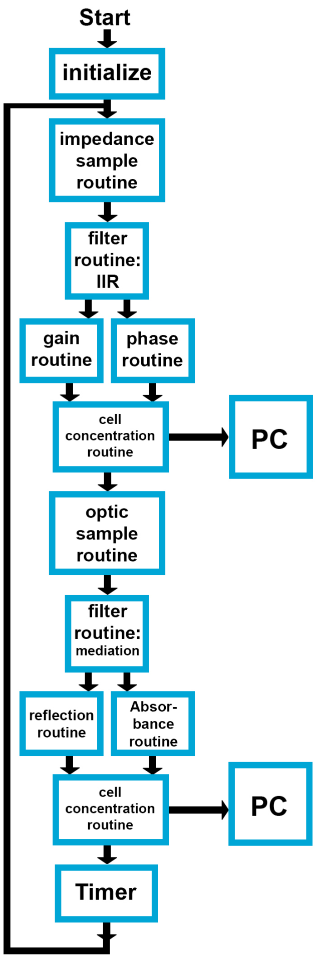


Figure 6.1: Flowchart of program

6.2. USER INTERFACE

From personal correspondence with Hans Tanke, head of the department of molecular cell biology, it became clear how the information should be presented to the user. As seen in figure 6.2, a large portion of the screen is dedicated to the graphical display of the data.

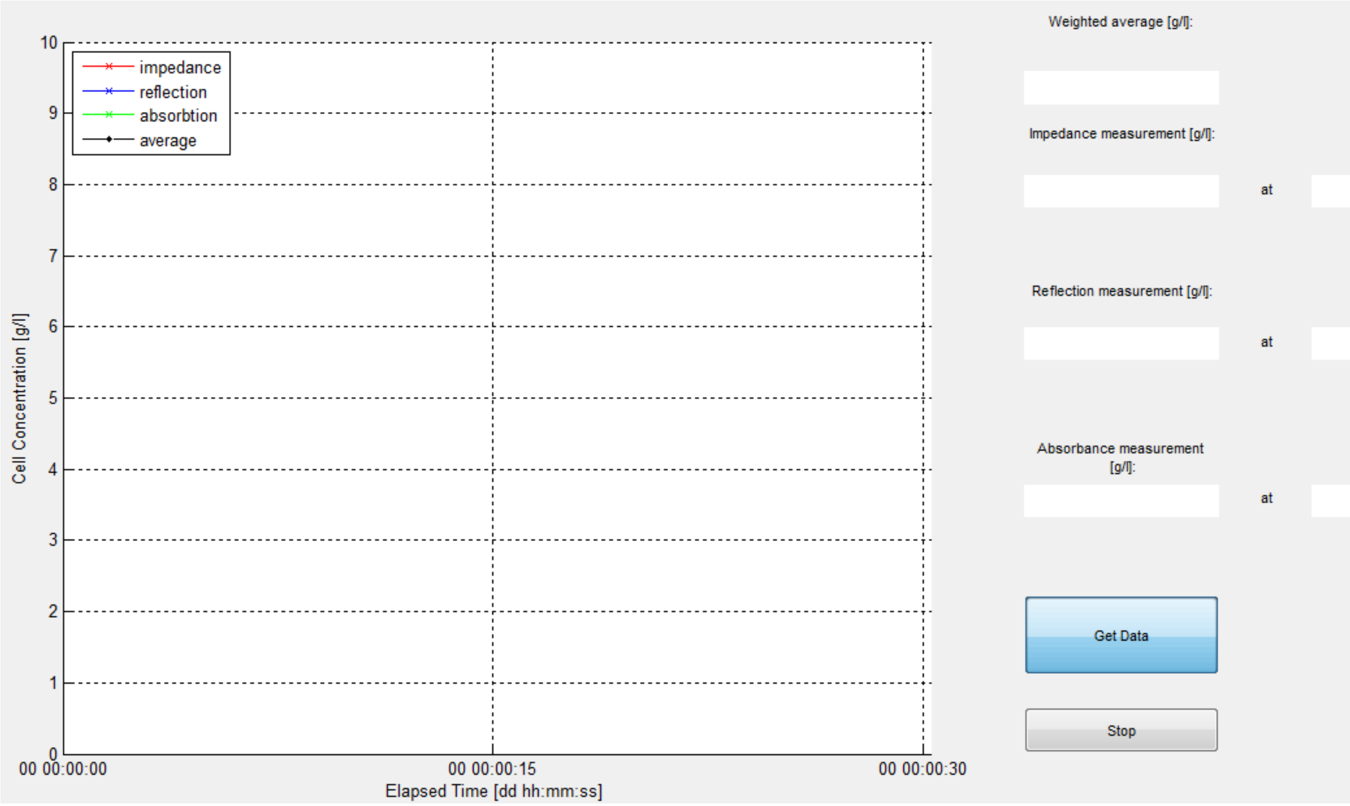


Figure 6.2: Screenshot of the graphical user interface

A large button labeled 'Get Data' initiates communication with the Arduino, and stores the data so the lines can be displayed. Before each value, a number is sent to identify to which sensor the measurement corresponds. For instance, a '-3' notes that the origin of the measurement is the reflection sensor. Negative numbers are used, so these numbers cannot be mistaken for actual data. A stream of data will be somewhat similar to:

```
-3
70.27
-4
72.03
-5
71.81
-3
71.15
-4
72.17
-5
72.45
```

This approach doubles the amount of numbers communicated, but since this part is not time critical, it is no disadvantage. It is more important that the transmitted value is assigned to the correct data group. Besides in the graph, data is also displayed at the right of the graph, below the corresponding sensor name. Also, the time of the measurement is presented in 'dd hh:mm:ss' format. This allows for an exact reading of the data, even for experiments lasting up to two weeks. This part will change when 24 wells are subsequently measured, and most probably will change to a table-like format.

Since experiments can take up to two weeks, the x-axis of the graph is displayed in date and time since the first button press of 'Get Data'. The x- and y-axis expand automatically when the values are about to exceed the limits.

6.3. WEIGHTED AVERAGE

To obtain a single cell concentration from the three different cell concentrations calculated by the three sensors, a weighted average is implemented. The weights are based on the accuracy at different ranges, and are graphically displayed in figure 6.3.

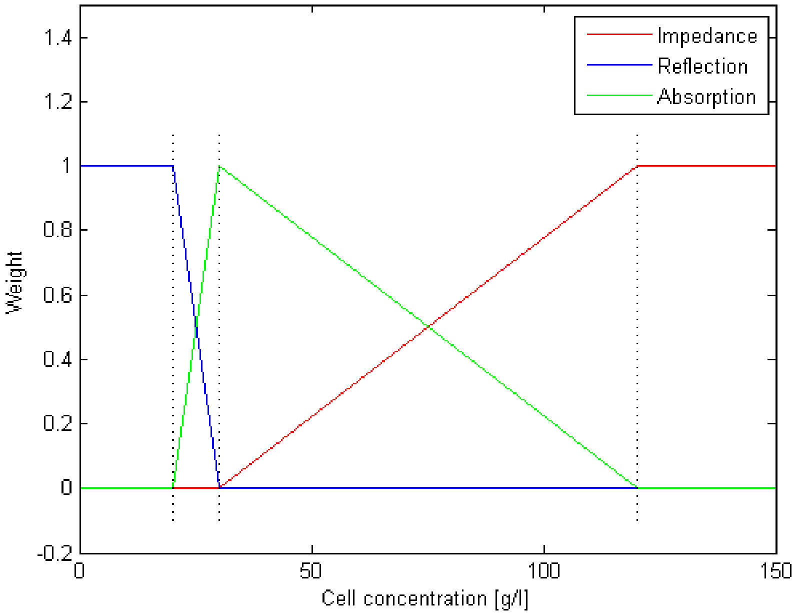


Figure 6.3: Weighing of the different sensors for determination of eventual cell concentration

Since the reflection sensor is most accurate in the lowest range, this method determines the eventual cell concentration. The absorption method is accurate in the range 20-120 g/l, so this method has a weight in this range. To obtain a smooth transition, in the transition period from 20-30 g/l, the cell concentration results of both measurements are linearly weighted, where the weight of the reflection measurement decreases and the weight of the absorption method increases. From 30 g/l, the impedance measurement produces useful results and gets an increasing weight, until at 120 g/l it is the only useful indicator of cell concentration. For higher concentrations than 120 g/l, the impedance measurement is the only measurement leading to the final cell concentration.

7

Results

This chapter is dedicated to the findings which have been obtained by the measurements. Each data path will have a section, and an other section is dedicated to the integration of the software subsystem with each separate system. This is practically the final system, as the final system simply switches between taking values from the two sensors.

7.1. Results optical data path

In [22], the optical sensor subsystem reported their results. The achieved relative error of the absorbance measurement method is about 10% in the range from 20-120 g/l. With the reflection method, a relative error can be achieved of about 40% in the absolute minimum range near 0 g/l. From 10-30 g/l, an accuracy of 10% can be achieved.

7.1.1. A/D converter results

This subsection will give a description of the measurements performed on the A/D converter of the Arduino. The goal of measuring the A/D converter is to find an accurate calculation method to convert the integer values coming from the A/D converter to the corresponding voltage at the input. Ideally, this conversion is given by:

$$V_{in} = \frac{3.3}{2^{12}} \cdot M = 8.057 \cdot 10^{-4} \cdot M \tag{7.1}$$

where M is the integer value, the reference voltage is 3.3V and the A/D converter gives values in 12 bits resolution. A linear fit will be made of the results, of which a coefficient of determination (R^2) can be determined, to check whether the fit is accurate enough to be used in the actual system to read out the A/D converter.

A known DC voltage is applied to one of the analog input pins of the Arduino. A multimeter was used to verify this voltage, defined as V_{in}. The voltage was applied between a range from 0 to $V_{dd} = 3.3V$, the maximum operating voltage of the Arduino (see table 4.2). The voltage was increased with step sizes of 0.1V. A table containing all 34 results is given in appendix A.1. The results are graphically displayed in figure 7.1. Using a linear regression, a first order polynomial was fitted with $R^2 = 0.9999$:

$$\hat{V}_{in} = 8.008 \cdot 10^{-4} \cdot M + 8.712 \cdot 10^{-3} \tag{7.2}$$

This relation will be used to convert measured integer values to the corresponding applied voltage. This function can also be used to determine the error of the A/D converter compared to the ideal case.

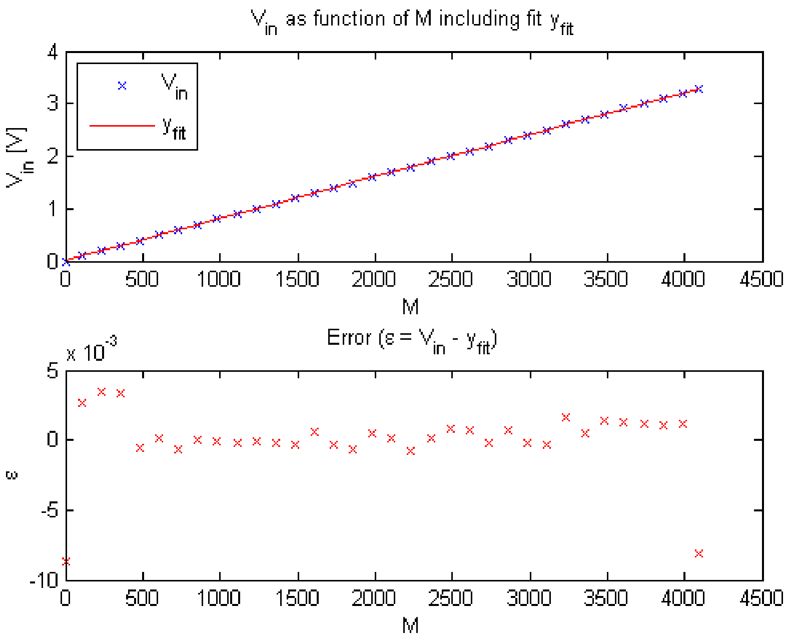


Figure 7.1: DC calibration results of the Arduino A/D converter

The offset E_o is given by:

$$E_o = 8.712 \cdot 10^{-3} V = 8.712 \cdot 10^{-3} \cdot \frac{4096}{3.3} = 11 LSB \tag{7.3}$$

The gain error E_g is given by:

$$E_g = 8.008 \cdot 10^{-4} - 8.057 \cdot 10^{-4} = -4.900 \cdot 10^{-6}\ V/level \tag{7.4}$$

Over the full range, this results in:

$$E_g = -4.900 \cdot 10^{-6} \cdot \frac{4096}{3.3} \cdot 4096 = -25\ LSB \tag{7.5}$$

Compared to the specifications of the Arduino's A/D converter, given in table 4.2, it is clear that these errors are within the specifications. The most important result of this experiment is that equation 7.2 will be used to convert integer values to the corresponding input voltage.

7.2. RESULTS IMPEDANCE DATA PATH

In [25], the impedance sensor subsystem reported their results. The achieved relative error of the measurement is less than 15% in the range 30-150 g/l.

7.2.1. A/D CONVERTER RESULTS

In the same way as for the Arduino's A/D converter, the AD7266 is tested on linearity and tested whether the results match the specifications from the datasheet. A DC voltage V_{in} was applied to the input of the A/D converter, and the value given was read out. A full table of the results can be found in appendix A.3. The result of the experiment is graphically displayed in figure 7.2.
A linear curve is fitted through the measurement, giving the following relation:

$$\hat{V}_{in} = 8.064 \cdot 10^{-4} \cdot M + 1.600 \cdot 10^{-4} \tag{7.6}$$

where M is the integer value received from the A/D converter. The linear fit that was found has R^2 = 0.99999.

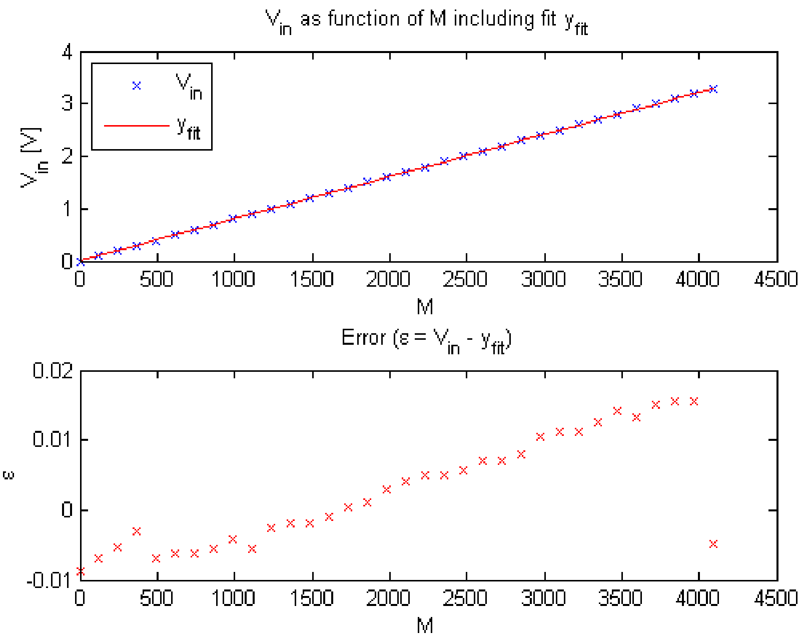


Figure 7.2: DC calibration results

In the ideal case, V_{in} would be given by:

$$V_{in} = \frac{3.3}{4096} \cdot M = 8.057 \cdot 10^{-4} \cdot M \tag{7.7}$$

In this ideal relation, all 4096 point are equally spaced over the 0-3.3V range. We can conclude that the AD7266 has an offset E_o of 0.160 mV, which is equal to:

$$E_o = 1.600 \cdot 10^{-4} \cdot \frac{4096}{3.3} = 0.2LSB \tag{7.8}$$

The gain error E_g is given by:

$$E_g = 8.064 \cdot 10^{-4} - 8.057 \cdot 10^{-4} = 7.00 \cdot 10^{-7}\ V/level \tag{7.9}$$

Over the full range, this results in

$$E_g = 7.00 \cdot 10^{-7} \cdot \frac{4096}{3.3} \cdot 4096 = 3.5\ LSB \tag{7.10}$$

Compared to the specifications of the AD7266, given in table 5.1, it is clear that the offset is within the specifications. The gain error is 1 LSB larger than the specifications given in the datasheet. Although the cause can be found and solved, it is not important for this application as the goal of the experiment is to obtain a relation between the integer value and the applied voltage. This is given by equation 7.6 and will be used in the software.

7.2.2. FILTER RESULTS

To test the digital IIR filter that was implemented in the impedance data path, a measurement method needs to be determined. In the microcontroller, a signal was simulated consisting of a sine wave at 1 kHz, the same frequency as the signal from the impedance sensor, with added noise at 10% of the magnitude of the sine wave. The digital filter was implemented on the Arduino, and the signal is passed through the filter. The resulting signal is communicated to the computer and plotted together with the original signal in figure 7.3.

It can be seen that the noise is completely eliminated and a smooth sine wave remains. A small delay is introduced by the digital filter. This is no problem for the phase measurement, as both signals are delayed by the same amount.

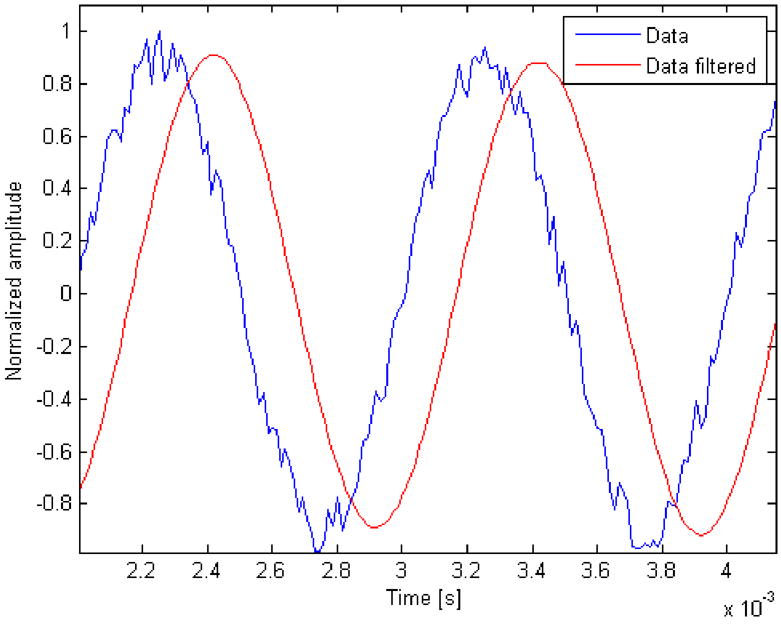


Figure 7.3: Simulation of a noisy signal and the IIR filter.

7.2.3. PHASE & AMPLITUDE ROUTINE RESULTS

To verify the algorithms used for the determination of the amplitude and the phase, first, the algorithms are programmed in MATLAB. Second, the algorithms are programmed in Arduino to see whether the extensive computations can be correctly evaluated on the microcontroller.

For the first verification, the algorithms were implemented in MATLAB, and the Arduino is used only for sampling, filtering and communicating the whole signal to the computer. The impedance sensor is connected, and the corresponding beaker is filled with water. The signals are displayed in figure 7.4, along with the product of the two signals.

By the algorithm discussed in section 5.6, the amplitude is determined, and results in

$$|V| = 0.9996$$

$$|V_{ref}| = 0.9998$$

In accordance with the algorithm discussed in section 5.5, the product of the signals is taken and the mean of this signal is taken. By using equation 5.11, the phase difference θ is calculated:

$$\theta = \arccos\left(\frac{2\mu}{V_0 I_0}\right) = 155.0727° \tag{7.11}$$

After correcting for the 180° phase shift introduced by the inversion, the phase difference is 24.9°. The expected value of a beaker filled just with water is about 23° [25]. The derived value is within 2 degrees of the expected value. The second verification for the phase and amplitude routine was performed under the test plan as described in section 7.3.

7.3. RESULTS INTEGRATION

This section will discuss the results of the software integrated with the two sensors. This section only discusses the integration of the software with one sensor at the time. However, this is representative because eventually, the different sensors will simply be subsequently read out. This is equivalent to shifting between the two different sensors.

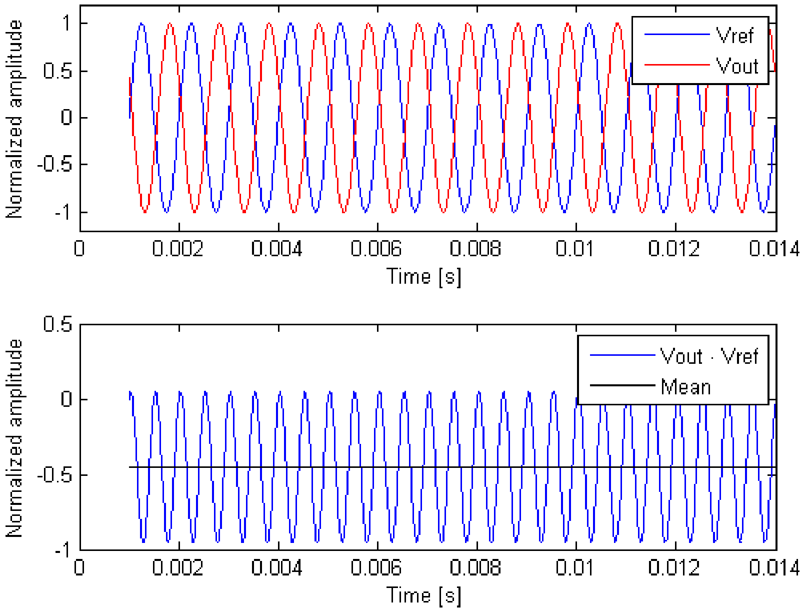


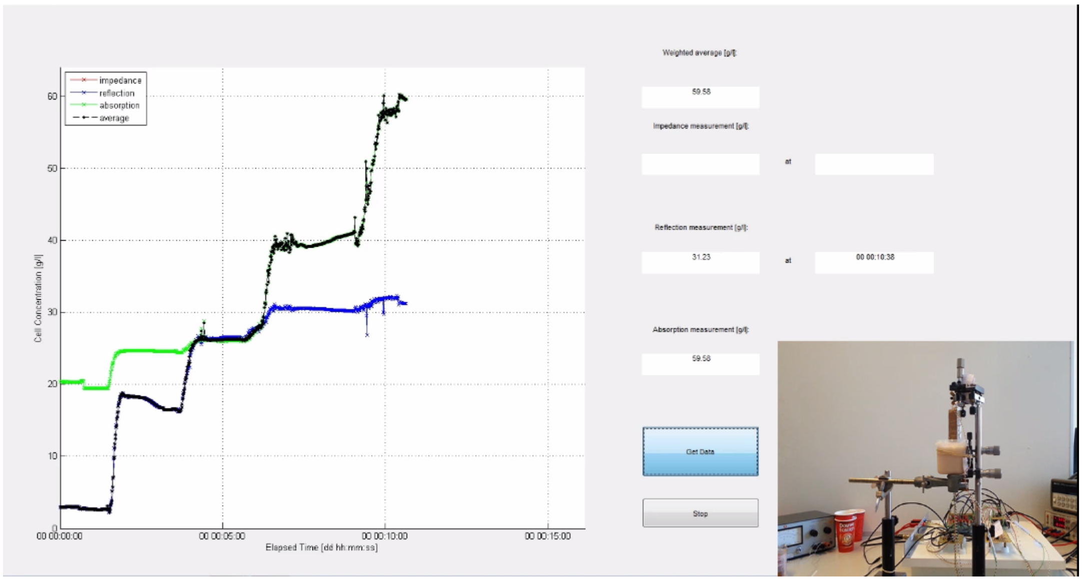
Figure 7.4: Signals V_{out} and V_{ref}, and their product

7.3.1. OPTICAL SENSOR ROUTINES

To test the optical routine, the optical sensor will be connected to the Arduino. The optical sensor will be placed in a beaker filled with water where a certain amount of yeast is slowly added. The mass of each addition is known and the time of each yeast addition will be listed. Because the instants of yeast delivery and the actual amount of yeast mass at each addition are known, the routine can be checked with the serial communication output (USB) to verify the measured cell concentration.

The cell concentration of the yeast cells will be monitored with the help of the implemented user interface, where the time of the measurement will also be displayed. The yeast additions are weighted with a precision scale and will have known masses. Lastly, it must be mentioned that the described test plan in this subsection will be the standard which will be used in the other test plans. Executing this test plan gave the result displayed in figure 7.5. In the bottom right, a picture of the setup is shown.

Adding yeast in steps of 15 g/l, the weighted average gave results with 10% accuracy in the range 10-120 g/l. At the time of the screenshot of figure 7.5, 60 g/l was present in the beaker, and the measurement says 59.58 g/l.



Weighted average [g/l]: 59.58

Impedance measurement [g/l]: at

Reflection measurement [g/l]: 31.23 at 00 00:10:38

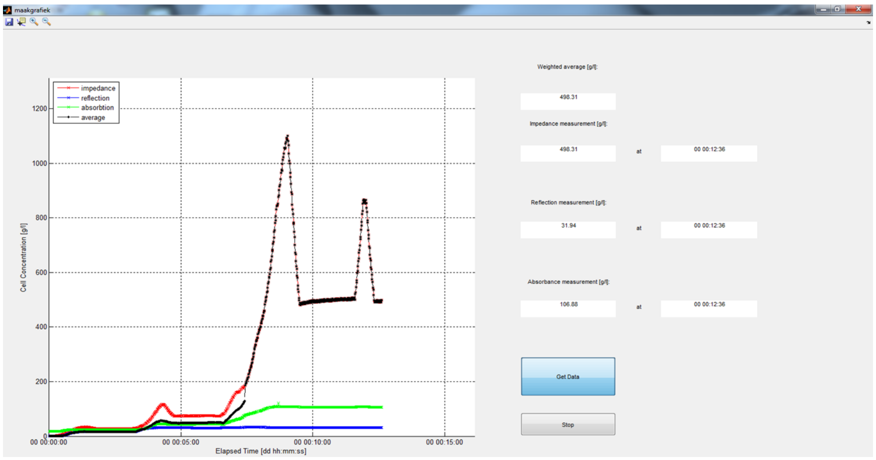
Absorption measurement [g/l]: 59.58

Figure 7.5: Screenshot of the graphical user interface

7.3.2. IMPEDANCE SENSOR ROUTINES

When integrating the microcontroller with the impedance sensor, the voltage is measured, along with a reference voltage. Using these signals and a known value of the current amplitude, the impedance can be calculated. With this, the cell concentration can be estimated. The full routine used to calculate cell concentration is given in section 5.7.

The goal of the test was to imitate the measuring steps made when testing the optical sensor. However, due to the extensive calculations and imperfections of the model, the cell concentration measurement was not quite constant. This is displayed in figure 7.6. Here, a measurement was done using all sensors. The impedance measurement, represented by the red line, gets out of control when yeast was added.



Weighted average [g/l]: 498.31

Impedance measurement [g/l]: 498.31 at 00 00:12:36

Reflection measurement [g/l]: 31.94 at 00 00:12:36

Absorbance measurement [g/l]: 106.88 at 00 00:12:36

Figure 7.6: Screenshot of the graphical user interface

Before implementing the whole routine, calibration curves have been made where the capacitance is calculated and displayed in the user interface. These experiments showed good results, corresponding to the capacitances calculated by [25] using other measuring devices. This shows that the routines used to calculated the capacitance are correct, and the model to convert the capacitance value to cell concentration is the cause of the incorrect measurements.

Using the setup with all systems connected, a full measurement time including communication to the computer was found to be 200 ms.

Altough the impedance sensor did not give correct results when implemented in the Arduino, the capacitance could be accurately calculated. This gave the suggestion that the model used to convert capacitance to cell concentration is the cause of the erroneous measurements.

8
Discussion

This chapter will evaluate and analyse the results. Also, the system is compared to the design brief to see whether the demands are met. If they are not met, an explanation will be given.

8.1. A/D conversion

The method used for the calibration can be seen as a standard calibration method where linearity errors are determined with linear regression. The advantage of this method is that it provides a straightforward way to calculate the compensation for the errors with a least squares sense. This is because a certain linear relationship needs to be followed and with simple subtractions the errors can be found. A limitation of this method is that it assumes that the relationship between the two variables is linear. When this is not the case, the linear regression will produce errors that will not match the data sheet. This because non-linear functions cannot be followed entirely with a linear function.

In this case, a R^2 value of 0.9999 was achieved with the linear regression for both the optical and impedance sensors. This states that correlation of the measured and known values is almost maximized ($R^2 = 1$) and we can assume that the models that were found have linear relationships.

8.2. Implemented routines

The optical routine (consisting an absorbance, a reflection and an averaging routine) was relatively simple to implement and thus robust. The absorbance and reflection routines are based on models which were found by the optical sensor team. The input of these models are DC voltages and the outputs are cell concentrations. Also, the digital filter used in this path was a simple averaging filter. The routines can be seen as robust because the routines works properly in the MATLAB and Arduino environments.

On the other hand, the impedance routine (consisting of a amplitude, a phase and a filtering routine) was complex to implement. The functions used were sensitive so a small error in input can lead to a widely varying output. In the end, the capacitance of the beaker could be reliably calculated. However, the conversion from capacitance to cell concentration gave erroneous results, due to imperfections of the model.

8.3. Meeting the design brief

The main goal of the proof of concept is that cell concentration could be determined. Using the optical sensor, this is achieved. The impedance sensor works in theory but the conversion from capacitance to cell concentration does not accurately give the cell concentration.

The 10% accuracy and 0-200 g/l range demand is not met. However, the reflection measuring method of the optical sensor is accurate to about 40% in the range 0-10 g/l and about 10% in the range 10-30 g/l. The absorption method is accurate to 10% in the range 20-120 g/l, and the impedance sensor is accurate to within 15% in the range 30-150 g/l. Using these effective ranges, a consideration can be made which signal is used as the most reliable one. In this way, the measurement system may be made more reliable.

The system meets the demand that the sensor does not affect the bio-process. The electrodes are made of stainless steel, and most of the electronics is placed outside of the beaker. The current techniques make sure no samples are taken. Eventually, the probe should be made disposable, so that it is sterile with every use. The rest of the electronics are not in contact with the suspension, so keeping this sterile is not an issue. The light of the optical measurement method does not influence the cells, and the voltage over the electrodes in the beaker is limited to an amplitude of 0.3 V, to prevent the measurement from influencing the cells [25]. These limited voltage levels make sure the user and environment of the sensor system is not in danger. Also, the input signals to the microcontroller are set to be in the range of 0-3.3V. Both sensors include protection circuitry to prevent the input voltage from exceeding its limits. The absence of hazardous materials make sure the device is safe to use. In section 7.3.1, it can be seen that the results are reproducible and consistent. A total measurement by both sensors takes 200 ms. This means that the maximum measuring time of 2.5 minute is easily met. This is beneficial, as all 24 wells can be subsequently measured without significant time difference, which allows for exact interpretation of the results by the user.

The measurement system is scalable to the containers of the micro-Matrix, having a working volume of 5 ml: the impedance sensor will operate similarly when using smaller electrodes. Also, the LEDs of the optical sensor can be replaced with smaller SMD versions. The components used in the software subsystem can be concentrated to a single PCB to save space. In this way, the limited size within the micro-Matrix is efficiently used. This implementation can be done using the microcontroller present on the Arduino: the AT91SAM3X8E, as this has a CAN port, for communicating with other systems in the micro-Matrix. To prevent the cell concentration measurement system from disturbing other measurements such as the pH measurement, which also uses optical sensors, the measurements should be executed at different times. The pH measurement is executed on a single well at a time, and multiplexed to measure all 24 wells. The cell concentration measurement will also be multiplexed. When these measurements are adapted, the measurements will not be disturbed as both measurements are not executed at the same time on the same well. As the system is now, the phase cannot be accurately determined to 0.1°. However, for the optical sensor, the routines are simple and robust and can be easily integrated in the microcontroller. The current system is only used with yeast from the supermarket, having a diameter of about 5 μm. Further research should perform tests with cells of other cell types to make sure the techniques used also perform as specified for these cells.

Lastly, a more general reflection on the total project will be given. Because of the fact that the research groups worked parallel on the sensor, some choices for the software and signal processing were taken without any knowledge about the data that needed to be processed. In the early stage of the project, data properties such as AC/DC characteristics, input/output ranges and signal frequencies were unknown. The choice to use an Arduino was based on the fact that this is a world wide well known platform with lots of possibilities concerning prototyping. The second reason to use the Arduino was due to the fact that the reserved time for the project needed to be used efficiently. What is meant by this, is that Arduino was chosen in the early stage of the project in order to start working on the processing algorithms without losing time. For future research, the choice for a microcontroller could be considered again. Properties such as data rate and communications protocols need to be taken into account. These properties can be determined because we assume that the data properties, of both sensors, are known in contrast to our situation.

9

Conclusion

The cell concentration sensor presented in this thesis is able to measure the cell concentration of yeast cells in a beaker, by using an optical sensor and an impedance sensor. These two sensors provide data which is processed with the help of digital signal processing.

The data processing includes anti-aliasing filters for preparation of A/D conversion, two digital filters for bandwidth lowering and some routines for the conversion of raw data into cell concentrations.

The conversion from the impedance sensor's signals to an actual cell concentration did not yield reliable results. However, the calculation of the beaker's capacitance did give good results. It can be concluded that the model used to convert capacitance to cell concentration is not accurate and reliable enough. The optical sensor's conversion is simple and robust as it could be simply implemented on the Arduino. The routines used for the optical sensor can be easily used to display the cell concentration in a graphical user interface. This graphical user interface is designed for validation of the routines of both sensors when actual yeast cell concentration measurements were performed which were partly successful.

Using solely the optical sensor, a range can be obtained of 0-120 g/l, where the measurements have a relative error of up to 40% at 0-10 g/l and about 10% at 20-120 g/l.

If the problems currently present with the routine of the impedance sensor could be overcome, a range of 0-150 g/l could be achieved with a relative error of less than 15% in the range 120-150 g/l.

This observation shows that the specification (0-200 g/l with 10% relative error) is not met. However, considering the limited time available, a good effort is made to design a proof of concept. The prospect of implementing these techniques in the micro-Matrix is promising, as all elements are scalable to fit in the micro-Matrix. The sensor does not disturb the bio-process or measurements of other sensors in the micro-Matrix. The time needed to perform a complete measurement is also well within the specification.

Further research has to be done to verify whether the techniques also work in the same way with cells other than yeast.

A

Appendix

A.1. Arduino A/D converter result

The results from testing the A/D converter of the Arduino. Next to the measured result, the ideal result is given by $M_{ideal} = V_{in} \cdot \frac{4096}{3.3}$.

Table A.1: Data gained from the Arduino A/D converter.

V_{in}	**M**	**M (ideal)**	V_{in}	**M**	**M (ideal)**
0	0	0	1.697	2108	2106
0.097	107	120	1.797	2234	2230
0.198	232	245	1.898	2359	2356
0.298	357	370	1.998	2483	2480
0.395	483	490	2.098	2608	2604
0.495	607	614	2.198	2734	2728
0.595	733	739	2.299	2859	2854
0.695	857	863	2.399	2985	2978
0.795	982	987	2.499	3110	3102
0.895	1107	1111	2.601	3235	3228
0.996	1233	1236	2.700	3360	3351
1.096	1358	1360	2.801	3485	3477
1.196	1483	1484	2.901	3610	3601
1.297	1608	1609	3.001	3735	3725
1.397	1734	1734	3.101	3860	3849
1.496	1858	1857	3.202	3986	3974
1.598	1984	1983	3.280	4095	4071

A.2. ARDUINO A/D CONVERTER TIMING RESULTS

Table A.2: Table displaying the phase difference between subsequent sampling of two identical signals using the Arduino's A/D converter.

# experiment	θ [°]	**# experiment**	θ [°]
1	10.8	16	5.7
2	6.8	17	9.7
3	6.5	18	3.9
4	10.7	19	8.8
5	9.3	20	7.8
6	8.3	21	5.4
7	11.8	22	8.4
8	9.9	23	5.7
9	10.0	24	4.8
10	2.5	25	8.2
11	4.1	26	14.3
12	11.6	27	4.7
13	9.2	28	11.5
14	2.4	29	11.5
15	6.5	30	7.2

A.3. AD7266 RESULT

The results from testing the AD7266 A/D converter. Next to the measured result, the ideal result is given by $M_{ideal} = V_{in} \cdot \frac{4096}{3.3}$.

Table A.3: Data gained from the AD7266 A/D converter.

V_{in}	**M**	**M (ideal)**	V_{in}	**M**	**M (ideal)**
0	0	0	1.697	2103	2106
0.097	119	120	1.798	2228	2232
0.198	243	245	1.898	2353	2356
0.298	368	370	1.998	2477	2480
0.395	491	490	2.098	2600	2604
0.495	615	614	2.198	2725	2728
0.595	740	739	2.299	2850	2854
0.695	864	863	2.400	2973	2979
0.795	987	987	2.500	3097	3103
0.895	1111	1111	2.601	3223	3228
0.996	1236	1236	2.700	3345	3351
1.096	1360	1360	2.801	3469	3477
1.196	1485	1484	2.901	3595	3601
1.297	1610	1609	3.002	3735	3726
1.397	1733	1734	3.101	3842	3849
1.497	1857	1858	3.202	3968	3974
1.598	1981	1983	3.280	4091	4071

A.4. IIR Low Pass Filter coefficients

The digital IIR filter consists of two section. The coefficients of section 1 and 2 are given in the table below:

Table A.4: Table displaying the coefficients of the IIR Butterworth digital low pass filter

Section 1	**Section 2**
Numerator:	Numerator:
1	1
2	1
1	0
Denominator:	Denominator:
1	1
-1.820	-0.845
0.846	0
Gain:	Gain:
0.00646	0.0774

A.5. IIR Low Pass Filter for DC offset coefficients

The digital IIR filter consists of two sections. The coefficients of section 1 and 2 are given in the table below:

Table A.5: Table displaying the coefficients of the IIR digital low pass filter

Section 1	**Section 2**
Numerator:	Numerator:
1	1
2	2
1	1
Denominator:	Denominator:
1	1
-1.9935	-1.9846
0.9936	0.9846
Gain:	Gain:
0.00001749	0.00001741

Bibliography

[1] S. Beutel and S. Henkel, *In situ sensor techniques in modern bioprocess monitoring,* Applied Microbiology and Biotechnology , 1493 (2011).

[2] A. Boiarski, *Fiber optic particle concentration sensor,* in *Proc. Fiber Optic and Laser Sensors III 0566* (San Diego, United States, 1985) pp. 122–125.

[3] F. Zhang, P. Scully, and E. Lewis, *An optical fibre yeast concentration sensor based on inter-fibre distributed coupling,* in *Proc. Lasers and Electro-Optics Europe* (Amsterdam, Netherlands, 1994) p. 329.

[4] S. Marose, C. Lindemann, R. Ulber, and T. Scheper, *Optical sensor systems for bioprocess monitoring,* Analytical and Bioanalytical Chemistry , 342 (2003).

[5] F. Zhang, E. Lewis, and P. Scully, *An optical fibre sensor for particle concentration measurement in water systems based on inter-fibre light coupling between polymer optical fibres,* Transactions of the Institute of Measurement and Control , 413 (2000).

[6] K. Asami and T. Yonezawa, *Dielectric analysis of yeast cell growth,* Biochimica et Biophysica Acta , 99 (1995).

[7] A. Soley, M. Lecina, X. Gámez, J. Cairó, P. Riu, X. Rosell, R. Bragós, and F. Gòdia, *On-line monitoring of yeast cell growth by impedance spectroscopy,* Journal of Biotechnology , 398 (2005).

[8] J. Carvell and K. Turner, *New applications and methods utilizing radio-frequency impedance measurements for improving yeast management,* in *Proc. MBAA National Convention* (Austin, United States, 2002) pp. 2183–2187.

[9] K. Asami, *Characterization of biological cells by dielectric spectroscopy,* Journal of Non-Crystalline Solids , 268 (2002).

[10] H. Wakamatsu, *A dielectric spectrometer for liquid using the electromagnetic induction method,* Hewlett-Packard Journal (1997).

[11] E. Krommenhoek, J. Gardeniers, J. Bomer, A. V. den Berg, X. Li, M. Ottens, L. van der Wielen, G. van Dedem, M. V. Leeuwen, W. van Gulik, and J. Heijnen, *Monitoring of yeast cell concentration using a micromachined impedance sensor,* Sensors and Actuators , 384 (2006).

[12] K. Mishima, A. Mimura, Y. Takahara, K. Asami, and T. Hanai, *On-line monitoring of cell concentrations by dielectric measurements,* Journal of Fermentation and Bioengineering , 291 (1991).

[13] K. Asami, E. Gheorghiu, and T. Yonezawa, *Real-time monitoring of yeast cell division by dielectric spectroscopy,* Biophysical Journal , 3345 (1999).

[14] K. Grenier, D. Dubuc, P. Poleni, M. Kumemura, H. Toshiyoshi, T. Fujii, and H. Fujita, *Integrated broadband microwave and microfluidic sensor dedicated to bioengineering,* IEEE Transactions on Microwave Theory and Techniques , 3246 (2009).

[15] B. Kapilevich and B. Litvak, *Microwave sensor for accurate measurements of water solution concentrations,* in *Proc. Asia-Pacific Microwave Conference* (Bangkok, Thailand, 2007) pp. 1–4.

[16] U. Raveendranath, S. Bijukumar, and K. T. Mathew, *Broadband coaxial cavity resonator for complex permittivity measurements of liquids,* Transactions on instrumentation and measurement , 1305 (2000).

[17] K.Saeed, R.Pollard, and I.Hunter, *Substrate integrated waveguide cavity resonators for complex permittivity characterization of materials,* in *IEEE Transactions on Microwave Theory and Techniques*, Vol. 56 (2008) pp. 2340 – 2347.

[18] B. Blake-Coleman, D. Clarke, M. Calder, and S. Moody, *Determination of reactor biomass by acoustic resonance densitometry,* Biotechnology and Bioengineering , 1241 (1986).

[19] Applikon Biotechnology, *The applikon website - introducing micro-matrix,* (2014).

[20] J. Ellwart, H. Brettel, and L. Kober, *Cell membrane damage by ultrasound at different cell concentrations,* Biophysical Journal , 3087 (2003).

[21] Atmel, *AT91SAM ARM-based Flash MCU,* (2012), datasheet AT91SAM3X.

[22] E. Lemmens and M. Jansen, *Cell Concentration Sensor for micro-bioreactors, Optical sensor system,* (2014).

[23] B. Champagne and F. Labeau, *Discrete time signal processing,* Class Notes for the Course ECSE-412 (2004).

[24] J. Brown and R. Churchill, *Complex Variables And Application*, 8th ed. (McGraw-Hill, 2009).

[25] J. van der Kemp and J. van den Hoorn, *Cell concentration sensor for micro-bioreactors, Impedance Sensor,* (2014).

[26] Analog Devices, *AD7266 Differential/Single-Ended Input, Dual 2 MSPS, 12-Bit, 3-Channel SAR ADC,* (2011), datasheet AD7266.

[27] Analog Devices, *Evaluation Board for the AD7265/AD7266,* (2009), datasheet EVAL-AD7266.

[28] Intersil, *Micropower Voltage Reference ISL21010,* (2011), datasheet ISL21010-33.

[29] J. Proakis and D. Manolakis, *Digital Signal Processing*, 4th ed. (Pearson Education, 2009).

[30] L. Couch, *Digital and Analog Communication Systems*, 8th ed. (Pearson Education, 2013).

[31] A. van der Veen EE2521 Digitale Signaal Bewerking Lecture 5, *topic: "analoog filter ontwerp",* Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology (2013 May 21).

[32] A. van der Veen EE2521 Digitale Signaal Bewerking Lecture 6, *topic: "digitaal filter ontwerp",* Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology (2013 May 23).