

Master Thesis



Czech  
Technical  
University  
in Prague

**F3**

Faculty of Electrical Engineering  
Department of Measurements

## Analysis of the textile fibers unevenness in frequency domain

Ondřej Renza

Supervisor: Ing. Jakub Parák

Field of study: Sensors and Instrumentation

Subfield: Cybernetics and Robotics

May 2016



## Acknowledgements

Děkuji.

## Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 15. May 2016

## Abstract

To be done.

**Keywords:** digital signal processing,  
textile defects, unevenness

**Supervisor:** Ing. Jakub Parák

## Abstrakt

To be done.

**Klíčová slova:** zpracování digitálního  
signálu, textilní vady, nerovnoměrnost

**Překlad názvu:** Analýza periodických  
vad textilních vláken ve frekvenční  
oblasti

# Contents

<b>1 Introduction</b>	<b>1</b>	<b>4 Conclusions</b>	<b>17</b>
<b>2 Theoretical Introduction</b>	<b>3</b>	<b>A Bibliography</b>	<b>19</b>
2.1 Textile Engineering .....	3	<b>B Project Specification</b>	<b>21</b>
2.1.1 Textile Fibers .....	3		
2.1.2 Combing Process .....	4		
2.1.3 Spinning Process .....	4		
2.2 Overview of Yarn Quality Sensors	5		
2.2.1 State of Art .....	6		
2.3 Digital Signal Processing .....	6		
2.3.1 Discrete Fourier Transform ...	6		
2.3.2 Fast Fourier Transform .....	7		
2.3.3 Power spectral density .....	9		
2.4 Digital Processing Techniques ..	10		
2.4.1 Spectrograms .....	10		
2.5 Embedded systems and microcontrollers .....	10		
2.5.1 Embedded systems .....	10		
2.5.2 DSP Hardware Options .....	10		
2.5.3 Real-time constraints .....	11		
2.5.4 ARM Cortex-M3 .....	11		
<b>3 Practical Implementation</b>	<b>13</b>		
3.1 Yarn Quality Analysis .....	13		
3.1.1 Description of Algorithm for Unevenness Analysis .....	13		
3.1.2 Flowchart Diagram .....	13		
3.2 Sliver Quality Analysis .....	13		
3.2.1 Hardware .....	14		
3.3 Description of Algorithm for Unevenness Analysis .....	16		
3.3.1 Flowchart Diagram .....	16		
3.4 Description of Designed System.	16		
3.4.1 Block Diagram .....	16		
3.5 Implementation of Software ....	16		
3.5.1 Filtration .....	16		
3.5.2 Detection of Defects .....	16		
3.5.3 Automatic Evaluation .....	16		
3.5.4 Example of Designed Application .....	16		
3.6 Implementation of Hardware ...	16		
3.6.1 Electronic Circuits Design ...	16		
3.6.2 Description of possible sensors	16		
3.6.3 Designed Prototype .....	16		

## Figures

## Tables

3.1 Block Diagram of the System . .	15
-------------------------------------	----



# Chapter 1

## Introduction

Textile manufacturing has always been important industry field. A major part of this industry is formed by a process called spinning, where twisting strands of fibers together form yarn. The modern spinners - textile machines, that execute the process of spinning - have been significantly improved and now they reach a high level of automation. This allows not only faster and cheaper production but also more focus on quality of the produced textile yarn.

The quality of the yarn could devalue the final product by creating defects, such as rapid changes in color or thickness etc., in the textile material. Even with modern technologies, it is still impossible to produce yarn without any defects. We can't prevent yarn defects by carefully selecting and preprocessing fiber because some defects can be created by spinning process itself. Out of many types of defects, this thesis is focused on the analysis of yarn unevenness (also called yarn irregularity). This is describing yarn with a diameter that is not even along its length, but it is changing its value periodically. We can measure this defect in the form of mass variation per unit length.

Designed system is not aiming to improve the quality of spun yarn, but to monitor quality (specifically unevenness) of produced yarn during the spinning process. Due to this monitoring, it is possible to stop spinning process if a defective yarn is detected. This allows the operator to resolve the issues that caused it, e.g. by replacing spinner fiber source with new one and reconnecting different yarn endings.

The project - described in this thesis - has been made in cooperation with company Rieter CZ s.r.o, who provided the device requirements, critical measurement data, and other important information. The goal of the project was to research and develop an algorithm for analysis and detection of yarn unevenness by using spectrograms and to design an embedded system capable of measuring a diameter of yarn while fulfilling the required time constraints and implement detection algorithm. The system is required to be controlled by ARM M4 microcontroller. Thus, the algorithm has to be implemented in a way that takes in consideration memory and computational limitation of such microcontrollers.

A very important specification was the requirement to analyse quality of two textile fiber types: the yarn and the sliver. Where sliver is the input

textile fiber for the spinning process and the yarn is its final product. The both of which has significant physical differences. Mainly they differ in size because yarn diameter is usually in a range of micrometers and sliver diameter is in a range from millimeters to centimeters. The diameter of the sliver is measured on combing machine, which precedes the spinning process. This requires different measuring system and filtration processing therefore, two systems and algorithms were designed for quality analysis of each fiber type. Their core content is equal but there are some major differences, which are described later in this thesis. The most significant difference is that processing of signal representing sliver diameter requires much more advanced techniques of digital signal processing. This is necessary due to a strong presence of periodical artefacts on sliver diameter caused by machine preprocessing. This type of diameter fluctuations has to be distinguished from the actual sliver unevenness, which is task for complicated digital filtration in a frequency domain.



## Chapter 2

### Theoretical Introduction

Before the process of software and hardware development could begin, detailed theoretical research had to be done. Topics of the research include textile manufacturing, combing process, spinning process and textile fiber defects to understand better device requirements. Another step of research was focused on digital signal processing techniques that could be used on the project, mainly spectrogram estimation and its calculation using Fast Fourier Transform, together with possible filtration algorithms. Another research topic was aimed to cover embedded systems and specifically micro-controller usage and its real-time constraints.

#### 2.1 Textile Engineering

Goal of textile manufacturing is to make fabric from textile fibers, which can be then used for clothes. This process can be separated in several stages:

- Preparatory Processes - prepares the textile fiber for spinning process by blending, carding and combing,
- Spinning - fibers are spun into yarns,
- Knitting or Weaving - yarns becomes fabric,
- Finishing - fabric is transformed into clothes etc.

In regards to the topic of this thesis only two - the spinning and combing processes - are described.

##### 2.1.1 Textile Fibers

Fibers are the basis for all textiles. We distinguished the two main types: natural fibers and synthetic fibers. The natural fibers are:

- Cotton - from the cotton plant,
- Linen - from the flax plant,
- Wool - from sheep and

- Silk - from silkworms.

Examples of widely used synthetic fibers are:

- Viscose - from pine trees and petrochemicals,
- Acrylic, nylon, and polyester - from oil and coal.

The cotton is the most important natural fiber and analysis of the quality in this thesis is aimed specifically at cotton manufacturing. The fibers can have two main forms during the manufacturing process - the sliver and the yarn. The sliver is created by carding the fiber. In this process textile fibers are separated and then joined together into a loose strand of 1 cm to 4 cm in diameter. In the end of textile manufacturing process an yarn is created. It is a textile fiber with significantly smaller diameter, in comparison to sliver, usually in a range of micrometers [9].

### ■ 2.1.2 Combing Process

Combing is a preparation process during textile manufacturing. It is sub-part of a cleaning process which precedes the spinning process. Cotton contains a lot of impurities such as dirt, dust, foreign materials, neps and very short fibers. All of these should be eliminated by cleaning process. The combing process removes mainly short fibers and neps in sliver, which helps to produce stronger and cleaner yarn.

Combing is used in a production of medium-fine or fine yarns, where the quality of yarn is important. This quality improvement is at a cost of loss of raw material and high expenses for buying and operating the combing machines.

Comber (see Image 1) consists of three main parts:

- The Feed
- The Nipper
- The Comb

The process of removing the impurities is formed by attaching the input fibre in form of sliver to the feed roller

An input of this process is textile sliver (described in chapter xx).

### ■ 2.1.3 Spinning Process

The term “spinning” in this context refers to the process that executes conversion of a large quantity of individual unordered fibers of relatively short length into a linear, ordered product of very great lengths by using spinning machines. There are three main methods of executing process of spinning:

- Ring Spinning,
- Rotor Spinning and

### ■ Air-jet Spinning.

All of these systems yields yarn with different structures and properties. Each system has its advantages and limitations in terms of technical feasibility and economic viability.

## ■ 2.2 Overview of Yarn Quality Sensors

Importance of yarn quality on final product became clear in 1950s when first electronic yarn quality sensors were invented. Since then many principles were used in detection of yarn defects - optical, mechanical or even chemical [3].

Sensors of yarn quality are in textile industry often called *yarn clearer*. This term was created for first such devices because goal of yarn clearers wasn't only to discover possible yarn defects but also to immediately remove them. Today, yarn clearers analyse defects that are much more complex (given their properties such as periodicity etc.) and difficult to be immediately removed. Such defects are invisible to naked eye on single yarn and they appear only when turned to fabric.

Following overview is concerned with yarn quality sensors that have similar purpose and functions as the developed device. Therefore, only online yarn clearers that check quality of spun yarn on every spinning point in real-time are listed. All of the following devices are also aimed to be used on the yarn spun by a rotor spinners (see 2.1.3).

### ■ Uster Quantum 3

Uster Quantum 3 is modern, state of art yarn clearer, which provide detection of many types of defects and includes several interesting features:

- full yarn body display,
- foreign matter sensor with multicolored light sources,
- polypropylene detection,
- detection of moiré,
- unevenness calculation CV%,
- IPI classification,
- calculation of hairiness,
- spectrograms calculation.

Uster Quantum 3 has been developed by company Uster Technologies, which is developing yarn clearers for 30 years. Their devices often feature application of new technologies and they offer high-quality products.

This yarn clearer is also one of the two commercially sold products capable of calculating spectrograms and evaluate yarn unevenness CV%.

### 2.2.1 State of Art

## 2.3 Digital Signal Processing

The largest part of work on this project is oriented on digital signal processing (DSP). Usage of modern advanced algorithms from this field allowed designing projected device in the first place. This section covers the most important algorithms that are used in this project.

Digital signal processing is an area of science and engineering that has developed rapidly over the past 40 years as a result of significant advances in digital computer technology. Today, many of the signal processing tasks that were conventionally performed by analog means are now realized by less expensive digital hardware [2].

To perform the processing digitally, there is need for the conversion between an analog signal and digital signal. This is done by an interface called analog-to-digital (A/D) converter, which yields a digital signal as it's output that is appropriate as an input to the digital processor [2, 1].

### 2.3.1 Discrete Fourier Transform

To perform frequency analysis on a discrete-time signal  $x[n]$ , we convert the time-domain sequence to an equivalent frequency-domain representation. This conversion is obtained by Discrete Fourier Transform that can be algebraically formulated as (according to [2, 1]).

Given  $N$  consecutive samples  $x[n], 0 \leq n \leq N - 1$  of a periodic or aperiodic sequence, the  $N$ -point Discrete Fourier Transform(DFT)  $X[k], 0 \leq k \leq N - 1$  is defined by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}. \quad (2.1)$$

Given  $N$  DFT coefficients  $X[k], 0 \leq k \leq N - 1$ , we can recover the  $N$  sample values of sequence  $x[n], 0 \leq n \leq N - 1$  using Inverse Discrete Fourier Transform (IDFT) given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn}. \quad (2.2)$$

If  $x[n]$  has infinite duration, the frequency samples  $X[2\pi k/N], k = 0, 1, \dots, N - 1$  correspond to a periodic sequence  $x_p[n]$  of period  $N$ , which is an aliased version of  $x[n]$ . When the sequence  $x[n]$  has finite duration of length  $L \leq N$ , then  $x_p[n]$  is simply a periodic repetition of  $x[n]$ .

The DFT defined in (2.1) can also be rewritten as

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \dots, N - 1, \quad (2.3)$$

where

$$W_N^{kn} = e^{-j\frac{2\pi}{N}kn} = \cos\left(\frac{2\pi kn}{N}\right) - j \sin\left(\frac{2\pi kn}{N}\right), \quad 0 \leq k, n \leq N-1 \quad (2.4)$$

The parameters  $W_N^{kn}$  are called the twiddle factors [6].

Understanding properties of DFT is critical for application of the transformation to practical problems. List of the main DFT properties contains:

- Linearity,
- Periodicity,
- Complex Conjugate,
- Circular Convolution,
- DFT and the z-transform.

For detailed description of DFT properties see [2, 6].

The operation of selecting a finite number of samples called windowing is equivalent to multiplying the actual sequence  $x[n]$  defined in a range  $-\infty < n < \infty$ , by a finite-length sequence  $w[n]$  called window. Using simplest rectangular windowing (truncation) on a signal, can cause an effect called *leakage*, which transfers power from frequency bands that contain a large amount of signal power into bands that contain only a little. This may create "false" peaks, peaks at wrong frequencies or changes the amplitude of existing peaks.

Another effect of time-windowing is *smearing*. Which causes a spread of spectrum accordingly to the width of the mainlobe of the window spectrum. This result in loss of resolution [1] .

Therefore, a "good" window should have low-level sidelobes and a narrow mainlobe to minimize both of these effects. There are four most known windows used for time-windowing:

- Rectangular,
- Triangular (or Bartlett),
- Hann,
- Hamming.

Their differences (as shown in image XXX) relays in a different width of mainlobe and peak sidelobe level.

### ■ 2.3.2 Fast Fourier Transform

Difficulty in using the DFT for practical applications is its high computational requirements. Direct computation of the N-point DFT requires computational cost of  $N^2$ . However class of efficient DFT algorithms called *Fast*

*Fourier Transform (FFT)* has computational cost proportional to  $N \log_2 N$  [6, 1].

Decimation-in-time FFT algorithms are based in splitting the  $N$ -point DFT summation into two summations, that one sum over the even-indexed points of  $x[n]$  and another sum over the odd-indexed points of  $x[n]$ . Therefore, we obtain

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \dots, N-1 \\ &= \sum_{m=0}^{N/2-1} x[2m] W_N^{k(2m)} + W_N^k \sum_{m=0}^{N/2-1} x[2m+1] W_N^{k(2m)} \end{aligned} \quad (2.5)$$

Dividing sequence  $x[n]$  we get two shorter sequences:

$$a[n] = x[2n], \quad n = 0, 1, \dots, N/2 - 1 \quad (2.6)$$

$$b[n] = x[2n+1], \quad n = 0, 1, \dots, N/2 - 1 \quad (2.7)$$

Shorter sequences are obtained by *decimating*<sup>1</sup> the sequence  $x[n]$ , thus, this FFT algorithm is called decimation-in-time. Substituting definitions 2.6 and 2.7 into 2.5 yields

$$A[k] = \sum_{m=0}^{N/2-1} a[m] W_{N/2}^{km}, \quad k = 0, 1, \dots, N/2 - 1 \quad (2.8)$$

$$B[k] = \sum_{m=0}^{N/2-1} b[m] W_{N/2}^{km}, \quad k = 0, 1, \dots, N/2 - 1 \quad (2.9)$$

where  $A[k]$  and  $B[k]$  are  $N/2$ -point DFTs [1, 2].

Thus, we can calculate  $N$ -point DFT  $X[k]$  from the  $N/2$ -point DFTs  $A[k]$  and  $B[k]$  (2.8, 2.9) using the following merging formulas

$$X[k] = A[k] + W_N^k B[k], \quad k = 0, 1, \dots, N/2 - 1 \quad (2.10)$$

$$X[k + \frac{N}{2}] = A[k] - W_N^k B[k], \quad k = 0, 1, \dots, N/2 - 1 \quad (2.11)$$

These formulas (2.10, 2.11) can be applied to any FFT of even length [1].

This procedure is shown in Figure XXX ([1]). The displayed structure in the figure is called the butterfly network. Each butterfly consist of just a single complex multiplication by the twiddle factor  $W_N^k$ , one addition and one subtraction.

An example for  $N = 8$  is show in Figure XXX. Each  $N/2$ -point DFT can be computed by two smaller  $N/4$ -point DFTs. By repeating the same process, we will obtain a set of two-point DFTs, which is illustrated in Figure XXX [6].

<sup>1</sup>Decimation of a signal with sampling rate  $f_s$  by a integer factor  $D$  results in the lower sampling rate  $f'_s = f_s/D$ .

FFT algorithm *decimation-in-frequency* is similar to the decimation-in-time, with important differences, that the decomposition and symmetry relationships are reversed. The bit reversal occurs at the output instead of the input and the order of the output samples  $X[k]$  will be rearranged [6].

The FFT algorithms shown in the previous paragraphs can be modified to calculate the inverse FFT (IFFT).

### 2.3.3 Power spectral density

Energy Spectral Density  $S(\omega)$  defined as 2.12

$$S(\omega) = |Y(\omega)|^2 \quad (2.12)$$

Which can be obtained from Parseval's theorem (2.13)

$$\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega \quad (2.13)$$

This equality shows that  $S(\omega)$  represents the distribution of sequence energy as a function of frequency. For this reason,  $S(\omega)$  is called the *energy spectral density*. Most of the signals in practical applications are such that their variation in the future cannot be known. It is only possible to make probabilistic statement about the variation. Such sequences are called *random signals*.

A random signal usually has finite average power, therefore, we can use average power spectral density for its characterization. Which is, for simplicity, better known under name *power spectral density* (PSD). As shown in [4] is defined as

$$\phi(\omega) = \sum_{t=-\infty}^{\infty} r(k) e^{-i\omega k} \quad (2.14)$$

where  $r(k)$  is auto covariance function  $r(k) = E[y_t y_{t+k}]$ . The PSD is very useful in the analysis of random signals since it provides us with information about distribution of the average power over the frequency. There are several different methods for estimating the PSD.

Methods that require direct use of finite signal for purposes of autocorrelation calculation are called *non-parametric* methods. On the other hand, methods that rely on a model for signal generation are called *parametric* methods [6]. Widely used non-parametric methods are

- Bartlett method,
- Blackman-Turkey method,
- Welch method or
- Danielle method.

They differ mainly in the resolution and variance level of the result. For detailed description of methods see [2].

## 2.4 Digital Processing Techniques

### 2.4.1 Spectrograms

## 2.5 Embedded systems and microcontrollers

There are many definitions of embedded systems. The one picked for this thesis subjectively seems well accurate in the description of the given term. It goes as follows.

*"An embedded system is a specialized computer system that is usually integrated as part of a larger system. An embedded system consist of a combination of hardware and software components to form a computational engine that will perform a specific function. Unlike desktop systems which are designed to perform a general function, embedded systems are constrained in their application [5]."*

Embedded systems very often perform in reactive and real-time environments, which has to fulfill the real-time constraints that are described in 2.5.3. Not satisfying these constraints can cause significant system consequences. If the consequences consist of a degradation of performance, but not failure, the system is referred to as a soft real-time system (e.g. highway car counter). On the contrary, if the consequences are system failure, the system is referred to as a hard real-time system (e.g. a braking system in vehicle).

Typical embedded system receives information about the surrounding environment via sensors and responds with actuators. General block diagram of such system is in Figure XXX.

### 2.5.1 Embedded systems

### 2.5.2 DSP Hardware Options

DSP algorithms can be implemented on different types of digital hardware. The following are the widely used options for DSP systems:

- Special-purpose chips such as application-specific integrated circuits (ASICs).
- Field-programmable gate arrays (FPGAs).
- General-purpose micro-processors or micro-controllers ( $\mu\text{P}/\mu\text{C}$ ).
- General-purpose digital signal processors.
- DSP processors with application-specific hardware accelerators [6].

Each hardware (HW) platform has different advantages and constraints for different applications, thus, there is no *best* HW platform that could be used for every practical project. Instead each option should be carefully



considered from point of flexibility, required design time, power consumption, performance and cost.

Characteristic of mentioned hardware options are summarized in Table XXX ([6]).

### 2.5.3 Real-time constraints

Generally, a real-time system is one that must process information and produce a response within a specified time, else risk severe consequences, including failure. A real-time DSP system demands that the signal processing time  $t_p$ , must be less than sampling period  $T$ , that is

$$t_p + t_o < T, \quad (2.15)$$

where  $t_o$  is overhead time of input-output (I/O) processing. Thus, this limitation gives constraint to the highest frequency signal that can be processed by DSP systems in sample-by-sample processing, given as

$$f_M \leq \frac{f_s}{2} < \frac{1}{2(t_p + t_o)} [6]. \quad (2.16)$$

Using different techniques of processing can reduce the I/O overhead time and increase the performance of the DSP hardware platforms. For example by applying a block-by-block processing, where the I/O operations are handled by DMA controllers, which place data samples in memory buffers [6].

Today, with performance improvement of hardware platforms, it is even possible to calculate FFT (see 2.3.2) of 64-points in matter of tenths of milliseconds with low-cost ARM microprocessor (e.g. specifically in  $0.16ms$  for microcontrollers of STM32F1xx series according to [7]).

### 2.5.4 ARM Cortex-M3

First manufactured microprocessor of Cortex series was ARM Cortex-M3. It was developed in 2006 and its target group of application were 32-bit microcontrollers. Its main advantage is an excellent efficiency which yields high performance and a low energy consumption without need for a very high system clock frequency. It is based on architecture *ARMv7*.

An ARMv7 architecture was developed as modern type of a general architecture that could be used for low-level microprocessors as well as for high-performance application processors. Design of the architecture is split in three main profiles:

- Profile A - aimed on application processors for performance-intensive systems capable of using embedded operational systems (such as Symbian, Android or Linux Embedded).
- Profile R - aimed on high-performance processors for real-time applications.

- Profile M - aimed on wide-range processors for deeply embedded applications [8].

## ■ Instruction Set

There are two instruction sets supported by ARM cores in general: the ARM instructions which are 32-bit and the Thumb instructions which are 16-bit. During the execution, the processor can switch either to the ARM state or to the Thumb state accordingly to the instruction set it is currently using.

The ARMv7 architecture is using new instruction set called a Thumb-2 which consist of the 32-bit Thumb instructions and also 16-bit Thumb instructions. This yields big improvement from the perspective of the ease of use, performance and code size. It allows execution of complex operation in state Thumb which increases the effectivity [8].

## Chapter 3

### Practical Implementation

Prior to the design of the system architecture a list of requirements for the device had to be created. This was done in discussion with company Rieter CZ s.r.o. who provided resources for development. The most important requirement is that the device shall be able to analyse quality of a yarn and sliver. This separates the project in two parts, where core software algorithms are shared but other parts significantly different.

#### 3.1 Yarn Quality Analysis

The yarn quality analysis requires only software algorithm and no additional hardware. This is because measurement of a yarn diameter is done by a row camera sensor that is connected to processing board which was already developed in Rieter. Processing board has CAN interface that sends values of diameters to the microcontroller.

Therefore, in this part of the project, only input we consider are values of diameter send over CAN. This determines the project as software system with main requirements and features defined as follows:

- The system shall receive yarn diameter values and process them in a way that yields spectrogram values as output.
- The system shall communicate via CAN bus.
- The system shall be able to run on ARM microcontrollers STM32F1xx series.

##### 3.1.1 Description of Algorithm for Unevenness Analysis

##### 3.1.2 Flowchart Diagram

#### 3.2 Sliver Quality Analysis

The sliver quality analysis require development of software and also additional hardware. In this case - opposite to yarn quality system - it is necessary to design new processing board due to the fact that sliver and yarn has significant

size differences (see 2.1.1 and different sensors needs to be used). Also it was decided that device developed in this project will be only prototype version, thus, it has additional functionality that can be removed in final product. The main requirements and features for the device prototype were defined as follows:

- The device shall be able to measure diameter of a sliver.
- The device shall be able to filter, and process measured data in a way that allows to calculate CV% value and Variance-length Curve.
- The device shall be able to measure the diameter with one or more specified sensors simultaneously.
- The device shall be able to send measured data to the service computer in real-time.
- A PC application shall be implemented for receiving and storing data received from the device.
- The device shall be able to send measured data to the service computer in real-time.
- The device shall run on ARM microcontroller STM32F446.

### ■ 3.2.1 Hardware

Purpose of the hardware section relays in providing the correct supply to the control unit and the sensors. Another feature of hardware platform is processing the signals from sensors and converting it to the range which is suitable for ADC<sup>1</sup> unit. Additionally the hardware platform allows to communicate with other distributed systems or PC by providing necessary interfaces.

The hardware block diagram of the system is shown in Figure 3.1. Description of each blocks follows in next paragraphs.

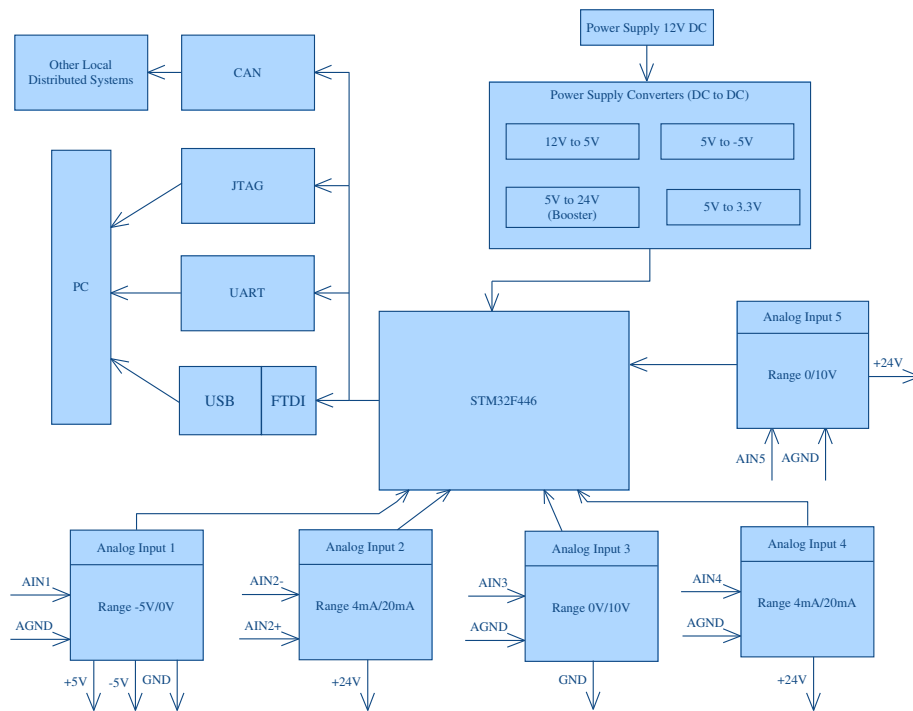
### ■ Control Unit

The main part of the hardware is ARM microcontroller STM32F446 from STMicroelectronics. Its choice was given by a requirement from Rieter CZ s.r.o with a description that it was chosen for its low cost, sufficient performance and company's good long-term experience with manufacturer's support.

Microcontroller STM32F446 is based on ARM Cortex-M4 technology with ARMv7 architecture 2.5.4. With DSP instructions it is capable of handling DSP algorithms such as FFT calculation in very short time (less than 0.5ms for 1024-point FFT). Maximal operation frequency is 180MHz at which gives 225DMIPS (Dhrystone benchmark). The most important features and peripherals of this microcontroller are:

---

<sup>1</sup>Analog to digital converter



**Figure 3.1:** Block Diagram of the System

- ARM 32-bit Cortex-M4, 180 MHz,
- 512 kB of Flash memory,
- 128 kB of SRAM,
- SWD and JTAG debug interface,
- 17 Timers and 3 ADC,
- Communication interfaces such as USART, CAN, SPI, SAI, SDIO, I2C, LIN etc.

### ■ 3.3 Description of Algorithm for Unevenness Analysis

#### ■ 3.3.1 Flowchart Diagram

### ■ 3.4 Description of Designed System

#### ■ 3.4.1 Block Diagram

### ■ 3.5 Implementation of Software

#### ■ 3.5.1 Filtration

#### ■ 3.5.2 Detection of Defects

#### ■ 3.5.3 Automatic Evaluation

#### ■ 3.5.4 Example of Designed Application

### ■ 3.6 Implementation of Hardware

#### ■ 3.6.1 Electronic Circuits Design

#### ■ 3.6.2 Description of possible sensors

#### ■ 3.6.3 Designed Prototype



# Chapter 4

## Conclusions

*Proof.*    8 Bla

- 1. Blo

□





## Appendix A

### Bibliography

- [1] V. K. I. DIMITRIS G. MANOLAKIS, *Applied Digital Signal Processing*, Cambridge University Press, 2011.
- [2] D. G. M. JOHN G. PROAKIS, *Digital Signal Processing*, Pearson Prentice Hall, 2007.
- [3] P. KOUSALÍK, *Inteligentní snímač pro on-line vyhodnocování kvality příze*, dissertation, Technická univerzita v Liberci, 2011.
- [4] R. M. PETRE STOICA, *Spectral Analysis of Signals*, Prentice Hall, 2005.
- [5] M. K. ROBERT OSHANA, *Software Engineering for Embedded Systems*, Elsevier, 2013.
- [6] W. T. SEN M. KUO, BOB H. LEE, *Real-Time Digital Signal Processing*, John Wiley and Sons, Ltd, 2013.
- [7] STMICROELECTRONICS, *STM32F10x DSP library*, 2010.
- [8] J. YIU, *The definitive guide to the ARM cortex-M3*, Newnes, 2010.
- [9] A. ZIABICKI, *Fundamentals of Fiber Formation*, John Wiley and Sons, Ltd., 1976.



## název práce

Analýza periodických vad textilních vláken ve frekvenční oblasti

Analysis of the textile fibers unevenness in frequency domain

katedra obhajoby

katedra měření

obor

&lt;neurčen&gt;

Nesouhlasí obor práce s oborem studijní plánu studenta!

studijní program

Magisterský

vedoucí

Parák Jakub Ing.

katedra teorie obvodů (13131)

[parakjak@fel.cvut.cz](mailto:parakjak@fel.cvut.cz)

oponent

student

Renza Ondřej (studuje) - zadáno

[renzaond@fel.cvut.cz](mailto:renzaond@fel.cvut.cz)

Senzory a přístrojová technika

studijní plán: Kybernetika a robotika - Senzory a přístrojová technika (MPKYR2)

katedra obhajoby podle studijního plánu: katedra měření

literatura

[1] Joseph Yiu: The Definitive Guide to the ARM Cortex-M3, Oxford, 2007.

[2] Kousalík, P.: Inteligentní čidlo pro on-line vyhodnocování kvality přize. Liberec, 2011. dizertační práce, TUL.

[3] Uhlíř, J., Sovka, P.: Číslíkové zpracování signálů. Ediční středisko ČVUT, Praha, 2002. Monografie CVUT FEL.

[4] Katalogové listy jednotlivých komponent a součástek.

pokyny

1. Seznámit se s problematikou vad textilních vláken a spřádných strojů
2. Navrhnout systém pro detekci vad textelních vláken
3. Navrhnout implementaci filtračního detekčního a detekčního algoritmu ve frekvenční oblasti
4. Implementovat softwarovou knihovnu s navrženým algoritmem pro mikrokontrolér
5. Experimentálně otestovat navržený systém s implementovaným algoritmem

zadavatel

spolupráce s Rieter CZ s.r.o.

vytištěno: 28.11.2015 14:04:00

vytiskl: Parák Jakub Ing.