Automatic trimming of ultrasonic pulse in fiber-optical power spectrometer

Ola Forsslund
Abstract

**Automatic trimming of ultrasonic pulse in fiber-optical power spectrometer**

*Ola Forsslund*

The aim of this master's thesis is to develop a method that fully automates a trimming step in the production of a fiber-optical power spectrometer, based on a unique Acusto-Optical Scanning Filter.

The filter is created by letting an ultrasonic mechanical pulse pass through a chirped Fiber Bragg Grating. The pulse introduces a disturbance in the grating, creating a thin optical transmission window in the otherwise reflective bandwidth. The high demands on the window requires a precise, unit dependent pulse form with unknown properties. Thus each unit needs to be trimmed to reach required performance.

The manual trimming is largely a trial and error process, that contains two performance tests. We redefine one, eliminating the need to reroute the optical path and reducing the number of fiber weldings. The tests are then quantified, allowing a figure of merit to be based on weighted performance values.

A brute force method, testing a large set of pulses, is implemented. The set is defined by the parameter space spanned by previously produced units. Due to the large space, the method is too time consuming. Instead it is used to measure the performance spaces of three units. An attempt to largely reduce the parameter space using PCA failed.

An alternating variables method that finds local performance optima in the parameter space is developed. By using a set of several starting points, the method tends to find several qualified pulses. The method is implemented and successfully verified by trimming new units.

Finally we propose where to focus improvements of the method in a production ramp up.

Handledare: Rickard Terfelt
Åmnesgranskare: Tadeusz Stepinski
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F09 047
Sponsor: Proximion Fiber Systems AB

Tryckt av: Ångströmlaboratoriet, Uppsala Universitet
1 INTRODUCTION

Proximion Fiber Systems AB produces Fiber Bragg Gratings (FBG), mainly for use in optical networks. Using a unique patented Acousto-Optical Scanning Filter (AOSF) based on an FBG, Proximion produces the world’s fastest optical spectrum analyzer, WISTOM¹.

To increase the amount of data transmitted in an optical fiber network, more communication channels are added to the same fiber. This is done by letting each channel have its own unique color, called a wavelength, for the transmitted light. This technique is called Dense Wavelength Division Multiplexing (DWDM). In a single fiber, more than 100 channels can be simultaneously transmitted at data rates of up to 40Gbits/s per channel.

To monitor and analyze the optical performance in DWDM networks, the intensity and wavelength of light in each channel have to be measured, hence the need for an optical spectrum analyzer. A small part of the light in the fiber is rerouted into the AOSF in WISTOM to be analyzed by the embedded computer.

![WISTOM](image)

**Figure 1.1:** WISTOM enables non-intrusive, real-time monitoring of power, wavelength and Optical Signal to Noise Ratio for up to 1024 DWDM channels. The pictured model features a built in switch to selectively monitor eight different fibers.

The AOSF consists of a chirped² FBG that is characterized by being reflective for certain wavelengths at well-defined positions along the grating. In its undisturbed state, all wavelengths of interest are reflected by the FBG. As a mechanical longitudinal pulse (hence 'Acoustical') wave passes along the grating, the grating is locally disturbed. By generating the disturbance in a controlled way, a narrow transmission band is created to allow transmission (instead of reflection) of the wavelength corresponding to the current position of the pulse. The transmitted light intensity is measured by a photodiode. By correlating the intensity of the light detected by the photodiode with the position of the acoustic pulse, the wavelength and intensity of incoming light is known, creating an optical spectrum.

---

¹ Proximion product sheet, doc no 102032-B
² A chirped FBG has continuously increased (or decreased) fringes distances.
Figure 1.2 A mixture of light enters the AOSF from the left. All colors are reflected at different positions in the FBG, except for the color matching the current position of the mechanical (acoustic or ultrasonic) pulse, which instead is transmitted.

The pulse passes through the FBG in 40μs, scanning 42nm (1528-1570 nm). A new spectrum is sampled every 80μs. Different wavelength regions could be scanned by using a different FBG.

Due to the high precision pulse needed to open a narrow transmission band, the acoustic pulse form and amplitude must be trimmed for each individual unit during production. The optical response to this single trimmed pulse, has to comply to strict performance criteria for all wavelengths.

The trimming is performed manually in a time consuming process\(^1\) by highly qualified personnel. As production volume increases the production has to be scaled up and thus the manual methods have to be automated. The aim of this master thesis is to develop a method that fully automates the pulse trimming process.

1.1 Disposition

In chapter 2 you will find a short introduction to the basic physical principles on which AOSF is based, while the actual system implementation WISTOM is described in chapter 3. The manual production step that is to be automated is described in chapter 4.

By parameterizing the pulse form, it is shown in chapter 5 that the problem can be viewed as an optimization problem, using the free parameters as variables. By studying the pulses used in trimmed units it is shown that there exists a limited number of reasonable pulses to consider, thus making it possible to add constraints to the variables.

An optimization problem will need to have an objective function. Thus the tests described in chapter 4 are formalized into quantified quality values in chapter 6. In the manual process, the two types of tests are done in separate stages, why one objective function for each test is developed.

A brief overview of the different ad hoc methods that were considered to solve the problem is presented in chapter 7. A discussion leads to the choice to start developing a Brute Force program, testing a huge amount of pulses to find the optimum in the set. While this method is considered as slow, it is very usable to learn more about the system.

Using the Brute Force program developed in chapter 8, performance data from a couple of WISTOM units are collected in chapter 9. The resulting data is analyzed in chapter 10, the analysis indicates that an Alternating Variables method might be feasible. In chapter 11 this method is

\(^1\) Described in chapter 4
described and then simulated on the measured data. Finally it is implemented into the Brute Force program.

A scheme to fully automate the trimming process is proposed in chapter 12 and the concept is proven to work by semi-automatically trimming three units. Examples of where to focus future work to ramp up production is discussed in chapter 13.

1.2 Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Acronym/Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOSF</td>
<td>Acousto-Optical Scanning Filter</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BF</td>
<td>Brute Force (method)</td>
</tr>
<tr>
<td>C-band</td>
<td>Optical wavelength band, 1530 nm to 1565 nm</td>
</tr>
<tr>
<td>D/A</td>
<td>Digital to Analog</td>
</tr>
<tr>
<td>FBG</td>
<td>Fiber Bragg Grating</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array, programmable logic chip</td>
</tr>
<tr>
<td>GP-IB bus</td>
<td>General Purpose Interface Bus, a standard for instrument communication</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
</tr>
<tr>
<td>HOG</td>
<td>Heart Of Gold (The scanning filter casing)</td>
</tr>
<tr>
<td>IR</td>
<td>InfraRed</td>
</tr>
<tr>
<td>LabVIEW</td>
<td>Graphical programming language for measurement and automation</td>
</tr>
<tr>
<td>NSO</td>
<td>Non-Smooth Optimization</td>
</tr>
<tr>
<td>OSA</td>
<td>Optical Spectrum Analyzer</td>
</tr>
<tr>
<td>PABC</td>
<td>Amplification factor of digital ultrasonic pulse form</td>
</tr>
<tr>
<td>RPS</td>
<td>Resonance Pulse Sensor</td>
</tr>
<tr>
<td>SD</td>
<td>Steepest Descent (method)</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TLA</td>
<td>Three Letter Abbreviation</td>
</tr>
<tr>
<td>TLS</td>
<td>Tunable Laser Source</td>
</tr>
</tbody>
</table>
2 PHYSICAL PRINCIPLES

This chapter aims at making a short description of the basic physical principles on which WISTOM is based.

Since wavelength is sometimes incautious substituted for frequency in this report, it might be a good idea to remind the reader that \( \lambda = \frac{v}{f} \), where \( \lambda \) denotes the wavelength, \( v \) the speed of the wave and \( f \) frequency. In this report the wavelength and speed of light is always, unless otherwise stated, referred to as the speed and wavelength in vacuum. Thus

\[
\lambda = \frac{c}{f}
\]

where \( c \) is the speed of light in vacuum (\( c=299792458 \) m/s).

The actual speed of light in the fiber can be calculated by

\[
\nu_{\text{fiber}} = \frac{c}{n}.
\]

where \( n \) is the refractive index of the fiber.

2.1 Fiber Bragg Grating

A grating is a regularly spaced collection of parallel elements, here called fringes, located in the fiber core. According to Bragg's law\(^4\), light hitting a grating with a fringe distance \( d \) at an angle \( \theta \) between the grating and incident light will constructively interfere if the wavelength \( \lambda \) (in the material) obey

\[
2 d \sin \theta = N \lambda , \text{ where } N \text{ is an integer } >0.
\]

(3)

If the incident angle is perpendicular, \( \theta=90^\circ \) and Bragg's law becomes

\[
2 d = N \lambda .
\]

(4)

![Schematic view of a Fiber Bragg Grating](image)

Figure 2.1: Schematic view of a Fiber Bragg Grating. Please note that the illustration is not to scale. The distance between fringes is about one tenth of the width of the fiber core. Illustration from White paper doc no 100499-B.

By introducing a slight change in the refractive index of the core in the optical fiber, a small part of the incoming light will be reflected. By creating fringes of refractive change at periodic distance \( d \), reflections from each fringe will, according to Bragg, constructively interfere with every other only if light has the wavelength according to equation (4). The result is a strong reflection of that particular wavelength,

\(^{4}\) “Optics” by Hecht, 3:ed ed. ISBN 0-201-83887-7 chapter 10
while reflection of other wavelengths will have a random interference resulting in a reflection close to zero.

Note that this wavelength is measured in the material it is propagating in (i.e. the FBG). It is common practice to refer to the wavelength in vacuum, let \( n \) be the refractive index in the fiber, then

\[
\lambda = \lambda \frac{\text{vacuum}}{n}.
\]

To calculate the grating period \( d \) in an FBG, let the refractive index of the optical fiber grating be \( n \), and ignore the effect on optical path by the slight change of index in the fringe. Combining equation (4) with (5) and solving for \( d \) yields

\[
d = \frac{N \cdot \lambda \frac{\text{vacuum}}{n}}{2}.
\]

### 2.1.1 Numeric example

To understand the magnitude of distances in an FBG, let us make a simplified example. In telecommunications, it is common to use infrared (IR) light with wavelengths around \( \lambda = 1500 \) nm. This is due to the particularly low transmission loss for these wavelengths in optical fibers. For a first order FBG (i.e. \( N = 1 \)) with refractive index \( n = 1.5 \), the grating period would be

\[
d = \frac{1500 \text{ nm} \cdot 1}{2 \cdot 1.5} = 500 \text{ nm}.
\]

### 2.1.2 Chirped Fiber Bragg Grating

The FBG used in WISTOM is a so called chirped grating. This means that the grating period is changed along the fiber and thus different wavelengths will be reflected at different positions in the fiber. Here the grating period is monotonically varied, i.e. the distance between fringes constantly gets longer towards one end of the fiber.

### 2.1.3 Production of Fiber Bragg Grating

By exposing a photosensitive fiber to UV light through the side surface it is possible to permanently change the refractive index. There are two main methods to create the FBG pattern; utilizing interference or a phase mask. The interference based method allows flexible variation of the grating parameters such as their period and length, but the method requires very high precision. The phase mask method does not require the same precision, but the possible pattern parameters for the FBG is fixed at the creation of the phase mask. The method deployed by Proximion is based on the interference method.

---


6 "Fibre gratings and their applications" by S.a. Vasil’ev et al., Quantum Electronics 35 2005
The basic principle of the interference method is to split the UV beam in two beams, which are recombined at an angle to each other, thus creating an interference pattern. The pattern is focused at the photosensitive fiber and hence the fringes pattern in the fiber is created.\(^7\)

To reach a high flexibility where advanced FBG properties such as chirp, phase shifts and apodization\(^8\) are controllable through software without change of hardware, partially overlapping subgratings with slightly altered parameters are exposed into the fiber.\(^7\)

Moving the fiber at a constant speed, subgratings can either be performed by exposing the fiber with a short UV pulse (short enough to not cause motion blur of the pattern), or by using a continuous UV source but moving the interference pattern in a sawtooth motion (constant speed flowing the fiber and then quickly restore position). The latter requires the movements to be synchronized with an extremely high precision, but increases the possible speed of the fiber movement and overcomes many other problems.\(^7\)

To measure the translation of the fiber in relation to the interference pattern a He-Ne interferometer is used, resulting in a spatial resolution of 0.6 nm over a translation length of around half a meter.\(^7\) By using this stitching process Proximion is able to produce up to 10 m long continuous gratings.

### 2.2 Piezoelectric effect

In certain anisotropic (i.e. direction dependent) crystal structure materials, electric dipoles are generated in response to applied mechanical stress. The resulting electrical potential across the material is measurable and so forms a sensor of mechanical stress. This effect is called the piezoelectric effect.

The generated voltage is rather high, manually pressing a 20 mm long piezoelectric cylinder would easily generate a potential difference of 125 V.\(^9\)

---


\(^8\) Here, apodization refers to the refractive index change in the fringes. By letting the change approach zero toward the ends of the grating, side-loobs of the reflected spectrum can be reduced.

\(^9\) Piezoelectric ceramics, page 7, ISBN 0 901232 75 0
The effect is reversible; when applying an electric potential across the material the dimension of the material changes, creating an actuator.

The acoustic pulse is controlled by piezoelectric elements fastened at each end of the grating, one acting as an actuator creating the pulse, the other as a sensor to register the exact travel time of the pulse.

### 2.3 Environmental effects

#### 2.3.1 Strain

Strain applied to an FBG will change the refractive index and the distance between fringes. While the strain expands the grating, the refractive index is decreased. The decrease acts like a contraction of the optical path, reducing the effect of the increased distance between the fringes. The net effect is 76% of the applied strain. Since the distance between fringes is changed, the reflected wavelength will be changed.

#### 2.3.2 Temperature

A change in temperature will change the expand or contract the fiber. For an FBG, this means that the fringes distance and hence reflected bandwidth will change. In a chirped FBG that means that the position of reflection for a particular wavelength will move. The speed of a mechanical pulse in the FBG is also dependent on temperature. Thus the temperature will have to be taken into account for the system to work.

### 2.4 The Acousto-Optical filter

Inducing strain by a longitudinal (acoustical) mechanical pulse along a chirped FBG will locally, at the pulse position, change the distance between fringes. The change is small and so should be viewed as a change in phase of the grating period. If the phase change varies with the right slope, a narrow transmission band is opened, creating an acousto-optical filter. Since the transmission window depends on the slope, the transmission band gap is much thinner than the length of the pulse.\(^\text{11}\)

As the transmission window depends on the pulse position, the band gap opens for different wavelengths as the pulse moves along the chirped grating. Because the transmission window scans over the reflected bandwidth, the filter is called an Acousto-Optical Scanning Filer (AOSF).

---

\(^{10}\) "Photoinduced Bragg gratings in optical fibers" by Morey et al., Optics & Photonics News, Volume 5, Issue 2, February 1994, pp.8-14

\(^{11}\) "Kalibrering av WISTOM", Proximion internal doc M-PROJ-NUMB
3 PRODUCT DESCRIPTION

Figure 3.1: The inside of a WISTOM unit. The HOG (golden) contains the AOSF, mounted on top of the WEB (green) that contains control logic and embed computer. This engineering sample has optical connectors (blue), connecting the AOSF output to the optical sensor. In production units these connectors are removed and the fiber is welded together after the unit has been trimmed.

3.1 Scanning filter

Using a D/A converter and a high voltage transformer, a digital pulse is converted into motion of a piezoelectric element. The element is fastened to one end of the FBG; introducing a mostly longitudinal pulse of compression/decompression in the fiber material.

The pulse travels at about 5500 m/s\(^{12}\) (~20 000 km/h), passing across the 220 mm fiber grating in less than 40μs. The exact speed of the pulse depends on the temperature. To determine the position of the pulse at a given time, an exact speed measurement is needed. A second piezoelectric element is used as a sensor (for further details, see chapter 3.4), fastened in the other end of the grating. When the pulse reaches the end it puts strain on the piezoelectric element, generating voltage. By sampling the generated voltage, the exact travel time of the pulse can be measured.

The WISTOM Bragg grating is 220 mm long, inscribed into a 300 mm long fiber. The grating is linearly chirped with a grating period reflecting light from 1570 nm to 1528 nm. The piezoelectric actuator is fitted in the 1570 nm end of the grating; the piezoelectric sensor is fitted at the 1528 nm end. When the pulse reaches the sensor, it reflects back towards the actuator. Since the arrival time at the sensor is known, the time of the reflection reaching the actuator can be very precisely estimated. This makes it possible to generate a new pulse in resonance with the reflected pulse, building higher pulse amplitude than otherwise possible. The higher amplitude is needed to open the transmission window. Another benefit of generating the pulse in resonance is that there will only be one pulse at a time in the fiber.

\(^{12}\) See chapter 5.2

\(^{13}\) "Kalibrering av WISTOM" by PhD Sten Helmfri, Internal Proximion doc M-PROJ-NUMB

PROXIMION Proximion Fiber Systems AB mail: info@proximion.com web: www.proximion.com phone: +46 (0) 8 750 48 88 fax: +46 (0) 8 750 48 80 address: Skalholtsgatan 10 B, SE-164 40 Kista, Sweden vat no: SE556641515301
thus there is no need to wait for reflection pulses to die out before sampling the next sweep. The pulse is generated about every 80 μs.

The light enters the filter through the 1570 nm end of the grating. If the light entering the filter is white (i.e. contains all wavelengths), the light exiting at the 1528 nm end will have a thin bandwidth continuously changing from 1570 nm to 1528 nm and back to 1570 nm with a period of about 80 μs.

3.2 Sampling

While the pulse passes through the grating, the optical sensor is sampled at a clock frequency of 50MHz. The sampling is active as the pulse travels from the actuator towards the RPS; the sampling is shut off during the return of the reflected pulse. The results are 1792 sample points in one sweep, repeated every 80 μs.14

3.2.1 Subsamples

To increase the number of sample points, the clock can be skewed by 1/8 of a clock cycle. Each such sweep is called a sub sweep. Thus, in eight sweeps 14336 subsamples of the spectrum are made. The number of sub sweeps is selectable from any 1, 2, 4 or 8 sweeps.14

3.2.2 Averaging

To increase the signal to noise ratio (SNR), the spectrum is made up of an average of several sweeps. The user can choose the number of sweeps (up to 32 768) used in every spectrum. The number is selectable in multiples of 2, i.e. $2^n$ where $n = 0, 1, \ldots, 15$.14

Each sweep is sampled by an FPGA that creates the averaged spectrum. The spectrum is further processed by software running on an integrated PowerQUICC-II micro controller.15

3.3 Calibration tables

The transmitted wavelength is a function of both position of the pulse, hence time, and the temperature. A very accurate temperature sensor is located inside the HOG. Due to variations in the grating, and due to a slight degradation of the ultrasonic pulse as it moves along the grating, the power measurement will also depend on temperature and time.

To deliver an exact measurement of wavelength and power, each sample value and corresponding wavelength is calculated with constants from calibrated lookup tables (LUT). The tables has a resolution of 1024 points.16

These calibration tables are built after the pulse trimming is performed. The unit is mounted in a climate chamber and measurements of references with well known characteristics are performed at different temperatures.

The power calibration is performed using a white light source with precisely known spectral shape, while the wavelength calibration is performed with a precisely known interference pattern from the white light source and a laser source.17

3.4 Resonance Pulse Sensor (RPS)

To keep track of the resonance frequency for the acoustical pulse, as well as to calculate the pulse position in the fiber, a piezoelectric sensor is mounted at the end of the grating. It is called

---

14 “WISTOM User guide”, Proximion doc no 0002019 rev B
15 Rickard Terfelt, WISTOM hardware designer, Proximion, personal communication
16 “Functional Description WISTOM SW”, rev D, chapter 3.3.5, Internal Proximion doc no. 100262
17 “Kalibrering av WISTOM”, chapter 5, Internal Proximion doc no. M-PROJ-NUMB
Resonance Pulse Sensor, or RPS for short. Its signal is sampled using the same clock as the optical sensor, thus allowing the same number of subsamples. The sampling is done in a time frame of 64 samples, i.e. \( \frac{64}{50 \text{MHz}} = 1.28 \mu\text{s} \).

The signal is averaged by a selectable number of periods before it is further processed by the automatic control system, see chapter 3.6.2.

The time resolution of the signal is very good, with 8 subsamples the resolution is

\[
\frac{1}{8 \cdot 50 \text{MHz}} = 2.5 \text{ns}.
\]

3.5 System interfaces

The WISTOM system is accessible via serial port and standard (Ethernet) network connection. WISTOM is controlled via a Human Machine Interface (HMI) consisting of command line interpreter accessible over telnet and the serial port, or via an Application Programming Interface (API) over TCP/IP.

3.6 Automatic Control System

Since the speed of the ultrasonic pulse in the fiber is dependent on temperature, the resonance period changes. To be able to keep generating the pulse in resonance with the returning pulse, an automated control system is needed.

There are two control system methods; Temp Compensator relies on the temperature sensor within the HOG to calculate the resonance period, while the Z-regulator method relies on the RPS signal.

3.6.1 Temp Compensator

The Temp compensator uses the temperature sensor inside the HOG to calculate an estimate of the resonance period. This method is normally only used on startup to easily find an estimate used as a starting point for the Z-regulator.

3.6.2 Z-Regulator

The Z-regulator uses the RPS signal to calculate the resonance period. Due to the tiny time frame (sampling window) of 1.28 \( \mu\text{s} \), the sampling must start just before the pulse reaches the RPS. Thus a fairly good estimate of the resonance period must be known before this method can be used.

By locking the sample window to the estimated resonance period, the pulse should ideally be stationary within the sample window. The zero-crossing of the main lobe is used as the pulse position reference; its expected position is stored in a variable called \( \text{regZ} \). Keeping track of the difference between the actual zero crossing and \( \text{regZ} \), the difference between estimated and real resonance is known. Thus the estimated resonance can be corrected.

This method is extremely accurate. Since the resonance depends on temperature, it is the pulse period that is normally used as temperature measurements when calculating the spectrum.

3.7 Optical Performance Monitor (OPM)

One of the roles that WISTOM plays in an optical network is to monitor the optical performance. The Optical Performance Monitor (OPM) is a software based application run on the embedded system.
The OPM module receives the spectrum samples and performs an analysis to detect the optical transmission channels that are represented by peaks in the spectrum. For the detected channels, certain characteristics such as center wavelength, optical power and OSNR are calculated.

The system is able to raise alarms if certain characteristics are changed or not fulfilled. To avoid generating false alarms the system is highly configurable, threshold levels are given an amount of hysteresis, distance between peaks and other methods to suppress changes that are likely to be caused by noise in the optical signal.20

20 "WISTOM Optical Layer Monitor Users Guide", doc no. 0002019 B
4 CURRENT (MANUAL) TRIMMING PROCESS

The aim of this chapter is to describe how the final manual trimming is performed today.

4.1 Method

In order to understand the problem the current process was investigated by interviewing the operators and engineers who developed the processes. Process documentation was also reviewed, but since these were in the process of being updated, emphasis was on the interviews. The process described in this chapter was reviewed by responsible engineers.

4.2 Overview

The mechanical pulse is modified in three stages of the manufacturing process. The first is performed before the fiber grating is mounted in its casing and is used for function verification purposes. The second is the final performance trimming. Measurements in both stages are very similar, why the same software is used.

A flowchart of the process can be viewed in Figure 4.1.

4.3 Process description

4.3.1 Equipment setup

- Tunable Laser Source (TLS)
- Optical Spectrum Analyzer (OSA)
- Computer running PulseWizard
- Ethernet network
- GP-IB bus interconnecting the TLS and OSA

4.3.2 Preparation stage (Connect test item)

Connect test item and enable trimming values readout.

- Connect the WISTOM fiber input to a Tunable Laser Source (TLS)
- Set TLS to 1545 nm and -9dBm (0.063 mW).
- Connect the WISTOM Ethernet port to the local computer network.
- Telnet WISTOM and execute the following commands:
  ```
  set opm# cnfg 4 1
  ```
- Reroute the fiber grating output to a 50/50 coupler and back to the internal sensor.
- Connect the second output of the coupler to an OSA.

---

21 “Production Instruction for Test Rig Pulse Shaping”, Internal Proximion doc no. 100358 rev pA1

22 Phd Cecilia Lundvall, Proximion Fiber Systems AB, personal communication
Automatic trimming of ultrasonic pulse in fiber-optical power spectrometer

4.3.3 Naming and test setup stage

In this stage the operator fills out basic meta data and initiate calibration process. The program requests the following information:

- Production stage (initial trimming of test 'rig' or final trimming of 'HOG')
- Serial number
- IP number
- Operator name

The operator clicks on 'connect', the green light indicates successful connection.

The operator clicks Start Pulse Wizard. The program will automatically move to the resonance search page, see the next step.

4.3.4 Automatic resonance search stage

This stage aims to find the resonance frequency of the ultrasonic pulse. It is fully automatic and outside the scope of this report.

4.3.5 Acoustic pulse forming stage

This is the stage that is to be automated. The aim is to find a feasible acoustic pulse-form for the individual unit. A pulse is defined by vertices (corner points) that can set by the operator in the graphical user interface, as well as a floating point amplification factor called PABC.

The lower graph in Figure 4.3 represents the pulse form. The yellow cross markers are the vertices (in the program called cursors) used to define the shape. They can be moved individually. The user can also add and remove vertices.

The upper graphs of the main page (Figure 4.3) show the optical spectrum, vertical axis shows photo detector amplitude and horizontal x axis shows (uncalibrated) frequency. The graph to the right has a logarithmic vertical axis while the one to the left has a linear vertical axis. The user can freely zoom in to the graphs.

Note: While the grating gets -9dBm power, the sensor gets what would correspond to -12dBm fiber input due to the 50/50 coupler.
4.3.5.1 Trimming procedure for laser performance

Most of the trimming is performed by evaluating the optical spectrum response to a single laser source. The peak produced in the spectrum should have certain characteristics. Some of these characteristics are well defined performance values, see Table 4.1.

The pulse definition, together with the amplification value called PABC, has to be trimmed so that the performance values meet their performance criteria for laser sources in the wavelength range (1528-1570 nm). This is considered fulfilled by testing for three wavelengths, long (1560 nm), middle (1545 nm) and short (1530 nm).

<table>
<thead>
<tr>
<th>GUI Label</th>
<th>Description</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWHM</td>
<td>Width of peak halfway from the peak amplitude</td>
<td>&lt; 4.5 GHz</td>
</tr>
<tr>
<td>Att @ -12.5</td>
<td>Attenuation -12.5 GHz from the peak center</td>
<td>&lt; -20 dB</td>
</tr>
<tr>
<td>Att @ -25.0</td>
<td>Attenuation -25.0 GHz from the peak center</td>
<td>&lt; -30 dB</td>
</tr>
<tr>
<td>Att @ 12.5</td>
<td>Attenuation +12.5 GHz from the peak center</td>
<td>&lt; -20 dB</td>
</tr>
<tr>
<td>Att @ 25.0</td>
<td>Attenuation +25.0 GHz from the peak center</td>
<td>&lt; -30 dB</td>
</tr>
<tr>
<td>OSNR</td>
<td>Peak amplitude divided by highest side peak amplitude</td>
<td>&gt; 40 dB</td>
</tr>
</tbody>
</table>

Table 4.1: Key performance values and pass criteria

These values are not calculated by default. They are enabled by means of the telnet console command set opm# cnfg 4 1.

A pulse called the default pulse is often used as a starting point. The TLS is set to one of the given wavelengths. The cursors defining the pulse form are moved until all performance criteria (see Table 4.1) are met or within a close range. The TLS wavelength is changed and the pulse is further modified to

PROXIMION
Proximion Fiber Systems AB mail: info@proximion.com web: www.proximion.com phone: +46 (0) 8 750 48 88 fax: +46 (0) 8 750 48 80
address: Skalholtsgatan 10 B, SE-164 40 Kista, Sweden vat no: SE556641515301
meet the criteria for the new wavelength. The procedure repeats for all three given wavelengths. A flow chart of the procedure is shown in Figure 4.4.

![Flow chart](image)

**Figure 4.4 Manual trimming flow chart.**

In addition to these criteria, some subjective criteria on how a pulse should look exist: It should have good amplitude, but an exact limit is not known. Further the pulse should be symmetrical, meaning no ‘bulge’ on one side, and that the attenuation values of one side should not differ too much from those of the other. There should not be any pronounced peaks at the side of the main peak (so called ghost peaks). This is essentially the same thing as the OSNR measurement that is calculated by hand.

### 4.3.6 Optical Spectrum Analyzer measurements (TLS–OSA sweep)

To be able to calibrate the optical amplitude measurements, the amplitude variation with wavelength must be fairly smooth. This step aims to modify PABC to make the variation smooth enough.

#### 4.3.6.1 Trimming procedure TLS–OSA sweep

The optical spectrum analyzer (OSA) is programmed to control the TLS to perform a frequency sweep (1520 nm – 1580 nm). The OSA is set up to measure the transmitted amplitude as a function of the TLS wavelength. The resulting graph on the OSA analyzed, such a graph can be viewed in Figure 4.5. The amplitude curve must not do local jumps by more than 1 dB. The definition of a local jump is vague and hence the decision whether the pulse is good enough to pass the calibration relies on operator experience.
4.3.7 Optical performance measurements stage

Measure the performance reached by the trimming.

Set the TLS to 1545 nm, 1530 nm and 1560 nm and record the performance values needed to complete Table 4.2. Measure the sensor saturation level by increasing the TLS power until the spectrum peak amplitude no longer increases. Record the TLS power.

4.3.8 Write to file stage

Save measurements and meta data to file.

4.3.8.1 Procedure

Use the Save to flash/File tab to save the pulse to file and WISTOM flash memory.

Use a text editor to fill out the structure as noted in table below. Values will be found on the resonance search tab and notes from the optical performance measurements.

<table>
<thead>
<tr>
<th>IP-address</th>
<th>192.168.8.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grating serial#</td>
<td>20070420-0958</td>
</tr>
<tr>
<td>HOG serial#</td>
<td>123</td>
</tr>
<tr>
<td>TLS Wavelength/Power</td>
<td>1545nm, -12dBm</td>
</tr>
<tr>
<td>PABC value</td>
<td>0.46</td>
</tr>
<tr>
<td>Resonance frequency</td>
<td>797.39 ns</td>
</tr>
<tr>
<td>HOG temperature sensor value</td>
<td>36.49</td>
</tr>
<tr>
<td>Acoustic pulse resonance centre (sample)</td>
<td>28.45</td>
</tr>
<tr>
<td>Acoustic pulse zero value</td>
<td>130.47</td>
</tr>
<tr>
<td>Acoustic pulse sensor peak to peak</td>
<td>172</td>
</tr>
<tr>
<td>FWHM @ 1545 nm, 1530 nm, 1560 nm</td>
<td>4.47, 4.2, 4.2</td>
</tr>
<tr>
<td>Optical sensor saturation level</td>
<td>-8.7dBm</td>
</tr>
<tr>
<td>Optical signal to noise (signal to unwanted peaks) ratio</td>
<td>20000</td>
</tr>
<tr>
<td>Attenuation at ±12.5 and 25GHz measured @1545 nm, 1530 nm, 1560 nm</td>
<td>-19.7/-30.2/-20.2/...</td>
</tr>
<tr>
<td>Operator name</td>
<td>TLA</td>
</tr>
</tbody>
</table>

Table 4.2: Trimming log file template. The log file is stored in Proximions production database. Typically the acoustic resonance frequency (Hz) is substituted for resonance period (ns).
5  PROBLEM INVESTIGATION

The basic investigation has two prime goals; getting to know the pulse characteristics, and finding relations between the pulse and the optical response, usable in a pulse trimming methodology. A secondary goal was to learn more about the system and the API by modifying the pulse wizard program.

5.1  System identification

To understand how the pulse should be modified to improve the AOSF characteristics, the system must be known. Seeing the complete system as a black box makes the system very complex. Breaking it down into identifiable subsystems could simplify the problem.

5.1.1  WISTOM system model

In this model, the AOSF is modeled as two different devices; the pure acoustical part, and the opto-acoustical part. The later has two inputs, optical and acoustical, with an optical output. The former represents the transformation of the mechanical pulse as the pulse is generated in resonance and transformed when it travels and gets reflected. The output is the mechanical action on the RPS.

5.1.2  Signals

By identifying which subsystems can be modeled, and what information could be gained, it is possible to find where to start the problem investigation. To gain any information, we must have known inputs and outputs, the outputs are listed below. D/A and A/D converter were assumed to not distort the signal and are thus considered transparent.

5.1.2.1  High voltage converter output (PowerLoop)

It is possible to reprogram the RPS A/D converter to sample the high voltage signal sent to the actuator. This is referred to as the PowerLoop. It could then be possible to identify how the high voltage converter affects the analogue signal from the D/A by reading the PowerLoop. This would give a good knowledge of the signal sent to the actuator.

5.1.2.2  Actuator output

This signal is the actual mechanical pulse in the fiber. Since the RPS signal (see below) has unsuitable bandwidth, an external measurement device would be required. Due to the individual variations of the units and the high precision needed, a general model cannot be used, thus the measurement has do be done on each unit, an unfeasible solution.\textsuperscript{23}

5.1.2.3  OASF output

The optical sensor was assumed to not distort the signal, thus the OASF output is considered as well known.

\textsuperscript{23} Michael Bergman, technical expert Proximion, personal communication
5.1.2.4 Acoustical filter output

The output is measured by the RPS.

5.1.2.5 RPS output

The RPS measures the movement of the fiber, while these movements correspond to the contraction and expansion of the mechanical pulse, thus the stress in the fiber. The stress is believed to be the derivative of the movement.

It should be noted that even if the RPS signal has a highly known position in time, the mounting of the fiber makes the precise location of the grating, and thus also the pulse, unknown. This makes it impossible to know exactly where inside the pulse the transmission window is opened. Thus it is hard to find the property in the RPS signal that makes the window open.

Some experiments have been done in an attempt to find a property in the RPS signal that correlates to a performance property of the optical output, see chapter 5.4.3 Investigation of correlations. These attempts were unsuccessful.

The RPS signal is believed to be limited by the bandwidth of the piezoelectric sensor. Still the RPS signal resembles the interferometer measured movement in Figure 5.1.

5.1.2.6 Spectrum

If the input to the OASF is a laser source with very thin bandwidth, it can be considered as a Dirac pulse, and thus the optical spectrum can be considered as an image of the transmission window. This makes it possible to study the window as a function of the mechanical pulse.

5.2 Pulse characteristic

The piezoelectric actuator is expanded as voltage is applied. This means that the pulse will always start with a compression of the fiber. As the piezoelectric element relaxes, the fiber compression not only returns to normal, but continues with an expansion, oscillating until equilibrium is reached, see Figure 5.1.

5.2.1 Mechanical pulse measurement

During the development of the WISTOM product, measurements on pulses were performed by Proximion. The longitudinal strain was calculated as the derivative of the longitudinal movement. An example of such measurement is displayed in Figure 5.1.
According to the measurement, the transversal strain is less than 1/3 of the longitudinal strain. Even though this might not be a negligible effect, the pulse is usually referred to as an acoustical (i.e. pure longitudinal) wave. It should also be noted that the actuator only is controllable in the longitudinal direction and that it is mainly the longitudinal shift of grating fringes that affects transmission.

5.3 Basic pulse investigation

5.3.1 The standard pulse form

The typical digital pulse form (see Figure 5.2) consists of two lobes. The second lobe was originally there to reduce ringing from the main lobe\(^\text{24}\). Ringing from the main pulse opens small unwanted transmission windows, resulting in 'ghost peaks' in the spectrum. Through production experience the secondary lobe has evolved to have a very flack decline.

5.3.2 Effect of secondary lobe

The secondary pulse is supposed to reduce ringing, resulting in less ghost peaks. To investigate the effect, the secondary lobe was altered.

However, it seems to have a much greater impact on the optical response. Also it turns out it does not only affect ghost peaks, but also improves the main transmission window. If the secondary lobe contains too much power, ghost peaks will be amplified instead of reduced.

In the two optical graphs below, the x-axis is measured subsamples in wavelength (nm) and the y-axis is a logarithmic measure of power. The pulse form pictured in Figure 5.2 would travel from right to left.

\(^{24}\)Mikael Bergman, technical expert Proximion, personal communication
thus the x-axis of the optical plots below is plotted likewise, the pulse traveling from 1570 end towards the 1528 end, right to left.

The pulses giving this response have identical main lobes, but the pulse resulting in Figure 5.3 has no secondary lobe. Note the dual peak form; this is very characteristic for the WISTOM response of a single pulse lobe. Adding a secondary lobe resulted in Figure 5.4, note how the lower peak is reduced in amplitude, but also that the peak is moved. However, the most contra intuitive and surprising effect is that it is the leftmost peak that is reduced the most, but the secondary lobe in the mechanical pulse is added to the right of the main lobe.

![Figure 5.3 Optical response from a single pulse lobe.](image)

![Figure 5.4 Optical response where a secondary pulse has been added.](image)

### 5.4 RPS signal correlation with Optical Performance

If it is possible to understand how the RPS signal correlates with the optical output, and how the pulse definition correlates to the RPS signal, it should be possible to find a systematic method that results in a pulse with good performance.

Since the RPS measure movements of the fiber, the derivative should correspond to the strain in the fiber. Therefore the Pulse Wizard, developed in LabVIEW, was modified not only to plot the RPS signal, but also its derivative.

#### 5.4.1 Extended view of the RPS signal

The RPS signal window is limited to $64 \times 8$ subsamples. The limited range hides the oscillations of the relaxing pulse. To see what happens to the pulse further away from the main pulse, the view was extended by moving the sampling window.

##### 5.4.1.1 Method

The RPS window can be moved. Since the Z-regulator depends on the RPS window position, the Z-regulator must be turned off before the window is moved.

- Set WISTOM pulse regulator in temperature mode.
- Move the RPS window.
This method was implemented into a modified version of the PulseWizard.

5.4.1.2 Results

![Resonance Pulse Signal (RPS) in white. The green curve is the negative derivate, while the red curve is the positive derivate, both magnified for visibility.]

Figure 5.5: Resonance Pulse Signal (RPS) in white. The green curve is the negative derivate, while the red curve is the positive derivate, both magnified for visibility.

![RPS signal shifted 20 main samples.]

Figure 5.6: RPS signal shifted 20 main samples.

5.4.2 Superimpose RPS signal on optical spectrum

Using a thin laser source considered as a Dirac pulse, the spectrum will be an image of the transmitting window. It would be interesting to plot the RPS signal in the same graph as the optical spectrum. The time scale is equal since the same clock is used, but the RPS signal representing the pulse must be aligned with the transmission window. To do that, the exact position of the grating in relation to the RPS sensor must be known. Unfortunately this measurement, even more so considering the extreme precision needed, would be very difficult to obtain. Thus the alignment must be left as a parameter.

The ability to superimpose the RPS signal on the optical spectrum with scaling and alignment parameters was implemented to the modified PulseWizard.

5.4.3 Investigation of correlations

Armed with the ability to superimpose and extend the view of the RPS signal, as well as its derivative, the modified PulseWizard was used to investigate if RPS signal features could be correlated with the optical response.

5.4.3.1 Method

- Shift the superimposed RPS signal along the x-axis so that the property to be investigated will match the corresponding optical response (e.g. peak or attenuation falloff).
- Change the pulse so that the RPS property is changed or moved
- Investigate the relative change of RPS position and optical peak

5.4.3.2 Results

Despite an extensive amount of direct experimentation, no evident conclusion could be made.

5.5 Parameterization of pulse form

The manual pulse trimming process uses a standard pulse form consisting of two lobes. These are usually described by their vertices. In order to formalize the problem, different parameterizations were tried out. In all parameterizations the main and secondary lobe can be described by the same set of parameters.

5.5.1 Dimension analysis

Each lobe can be described by 4 vertices (corner points). Let these be called $z_0$ to $z_3$, where $z=(x,y)$. 
Table 5.1: The vertices of the standard pulse form and their constraints.

From the Table 5.1 it is evident that each lobe has 5 degrees of freedom.

5.5.2 Parameterizations

In the Initial parameterization, both pulses are considered to be triangularly shaped, but with a maximum amplitude for the sample values, the main pulse would get a flat top by increasing the amplitude beyond the maximum amplitude.

In an effort to use the same type of parameters as those describing the optical peak, Physical parameters were used in the second parameterization. The optical peak is described by peak amplitude, FWHM, and attenuation. Here, the parameters were defined in the same way. Attenuation is replaced by climb- and sink (decline)-time.

The Physical parameters did not simplify the problem, but turned out to be cumbersome to work with. Efforts to find a parameter space that simplifies the problem using pattern recognition techniques are covered in chapter 10.1 Reduction of parameter space using PCA. Aimed at being as simple to use as possible, the Simplified parameterization was developed, and further refined in the Final parameterization.

5.5.2.1 Initial parameterization

The maximum sample value is 255, why a pulse with higher amplitude will get a flat top.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>Horizontal position of peak</td>
</tr>
<tr>
<td>$y$</td>
<td>Vertical position of peak</td>
</tr>
<tr>
<td>$r$</td>
<td>Raise time</td>
</tr>
<tr>
<td>$s$</td>
<td>Sink time</td>
</tr>
</tbody>
</table>

Translation to coordinates:

\[
\begin{align*}
  z_0 &= (x-r,0) \\
  z_1 &= \left( \min\left(255,\frac{y}{y'}\right), r+(x-r), \min\left(255,y\right) \right) \\
  z_2 &= \left( y'-\min\left(255,\frac{y}{y'}\right), s+x, \min\left(255,y\right) \right) \\
  z_3 &= (x+s,0)
\end{align*}
\]

Note that this parameterization hides the fifth parameter in the maximum amplitude constraint (255).
5.5.2.2  Physical parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Amplitude</td>
</tr>
<tr>
<td>p</td>
<td>Center of FWHM</td>
</tr>
<tr>
<td>F</td>
<td>Full Width Half Maximum (FWHM)</td>
</tr>
<tr>
<td>c</td>
<td>Climb time</td>
</tr>
<tr>
<td>s</td>
<td>Sink time</td>
</tr>
</tbody>
</table>

Figure 5.8: Physical parameters

\[ z_0 = (p - F/2 - |c/2|, 0) \]
\[ z_1 = (p - F/2 - |c/2|, A) \]
\[ z_2 = (p - F/2 - |s/2|, A) \]
\[ z_3 = (p - F/2 - |s/2|, 0) \]

5.5.2.3  Simplified parameterization

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Amplitude</td>
</tr>
<tr>
<td>p</td>
<td>Start position</td>
</tr>
<tr>
<td>b</td>
<td>Base width</td>
</tr>
<tr>
<td>c</td>
<td>Climb time</td>
</tr>
<tr>
<td>s</td>
<td>Sink time</td>
</tr>
</tbody>
</table>

Figure 5.9: Simplified parameterization

\[ z_0 = (p, 0) \]
\[ z_1 = (p + c, A) \]
\[ z_2 = (p + b - s, A) \]
\[ z_3 = (p + b, 0) \]

5.5.2.4  Final parameterization

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Amplitude</td>
</tr>
<tr>
<td>p</td>
<td>Start position</td>
</tr>
<tr>
<td>w</td>
<td>Top width</td>
</tr>
<tr>
<td>c</td>
<td>Climb time</td>
</tr>
<tr>
<td>s</td>
<td>Sink time</td>
</tr>
</tbody>
</table>

Figure 5.10: Final parameterization

\[ z_0 = (p, 0) \]
\[ z_1 = (p + c, A) \]
\[ z_2 = (p + c + w, A) \]
\[ z_3 = (p + c + w + s, 0) \]
5.6 Degrees of freedom in standard pulse definition

Adding the extra constraints of the current standard pulse form to the dimension analysis from section 5.5.1 will reveal how many degrees of freedom the standard pulse needs. See Table 5.2 and Table 5.3.

<table>
<thead>
<tr>
<th>Main Lobe Vertex</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>z₂</td>
<td>Fixed at 0</td>
<td>Fixed at 0</td>
</tr>
<tr>
<td>z₁</td>
<td>Free</td>
<td>Fixed at 255</td>
</tr>
<tr>
<td>z₀</td>
<td>Free</td>
<td>Fixed at 255</td>
</tr>
<tr>
<td>z₃</td>
<td>Free</td>
<td>Fixed at 0</td>
</tr>
</tbody>
</table>

Table 5.2: Standard pulse form main lobe vertices and their constraints

<table>
<thead>
<tr>
<th>Secondary lobe Vertex</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>z₂</td>
<td>Free</td>
<td>Fixed at 0</td>
</tr>
<tr>
<td>z₁</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>z₀</td>
<td>Fixed to z₁x</td>
<td>Fixed to z₁y</td>
</tr>
<tr>
<td>z₃</td>
<td>Free</td>
<td>Fixed at 0</td>
</tr>
</tbody>
</table>

Table 5.3: Standard pulse form secondary lobe vertices and their constraints

The main lobe has three free parameters; the secondary lobe has four free parameters. The pulse has an amplification factor called PABC. This adds up to a total of 8 degrees of freedom.

In terms of the final parameterization this would correspond to: Main climb time, main top width and main decline time. Secondary start position in relation to the main pulse end, secondary climb time, secondary amplitude and secondary decline time. Add PABC for the complete set of 8 parameters.

5.7 Parameter naming

During the development of programs and methods in this project, the parameter names have changed somewhat. For reference, Table 5.4 contains the different names and abbreviations used. The shaded rows contain the notation used in this report.

<table>
<thead>
<tr>
<th>PABC</th>
<th>Primary lobe</th>
<th>Secondary lobe</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Climb</td>
<td>Top</td>
</tr>
<tr>
<td>P</td>
<td>C1</td>
<td>T1</td>
</tr>
</tbody>
</table>

Table 5.4: Parameter notation

5.8 Analysis of pulses in produced WISTOMS

In the current manual process, the trimmed pulse is saved to a text file containing the sample values of the pulse. The PABC value is saved in the text file described in 4.3.8 Write to file stage. These files are stored in a directory structure on the server.

5.8.1 Extracting pulse data from directory structure

The pulse data was compiled into a single directory to be further processed. Some units have been pulsed several times and thus there was a need to select what data to use for that unit.

5.8.1.1 Implementation

The complete directory structure was copied to a local disk. For units that have been trimmed several times, all but the latest trimming were deleted.

Using a BASH script, the PABC values were extracted to a tab-separated text file with unit name in the first column and PABC value in the second column.

The pulse file was copied and renamed from pulsef.txt to unitName.pulse. This way all pulse files could be stored in the same directory.
5.8.2 Converting to pulse parameters

5.8.2.1 Method

Find the vertices of the pulse form. Only pulses complying with the standard pulse form were considered useful. Convert the vertices into the parameterized form developed in chapter 5.5.2.

5.8.2.2 Implementation

A Matlab function was developed to convert a pulse form file to coordinates. The filename or a file name pattern is given as input. The outputs are matrices of x and y coordinates as well as the actual pulses.

A coordinate in the standard pulse form is always located where the sign of the derivative changes or becomes zero. Thus the function searches for the location of the change and the sample value at that position, these as used as coordinates. Optionally, the pulses not resulting in exactly 7 coordinates can be cleaned out of the result list. The file name of faulty pulses is displayed. Another Matlab function was developed to convert coordinates to pulse parameters.

5.8.2.3 Results

Pulses from 42 units used the standard pulse form. The resulting parameters, together with the PABC value and unit name, were entered in a spread sheet, see Appendix B.

Almost all pulses are trimmed by the same operator, but some pulses differ less from each other than others. These seem to come from the same production batch. In future work one might want to investigate what makes up for this commonality (e.g. grating, actuator, mounting procedure...).

![Figure 5.11: The span of pulse-parameters in the set of 42 produced units that uses the standard pulse form.](image)

5.8.3 Defining a limited parameter space

The digital pulse is per definition discrete and the PABC value is increased in steps, thus the pulse definition belongs to a finite discrete domain. However, all possible combinations are not likely to result in a good pulse. To get an understanding of the possible pulse combination space, the used pulses were studied.

5.8.3.1 Method

The limited parameter space was defined as the total number of combinations possible within the span of all parameters. The span of parameter $i$ was defined by

$$s_i = \frac{M_i - m_i}{\Delta_i} + 1.$$  \hspace{1cm} (9)

where $m_i$ is the smallest value, $M_i$ the greatest value and $\Delta_i$ the step length used for the parameter. For all but PABC, the step length is 1. PABC is stepped by 0.01.
Thus the limited parameter space is defined as

\[ S = \prod_{i=1}^{9} \left( \frac{M_i - m_i}{\Delta_i} + 1 \right). \]  

(10)

5.8.3.2 Results

<table>
<thead>
<tr>
<th></th>
<th>PABC</th>
<th>C1</th>
<th>T1</th>
<th>D1</th>
<th>S2</th>
<th>C2</th>
<th>D2</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>0.55</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>17</td>
<td>15</td>
<td>70</td>
<td>116</td>
</tr>
<tr>
<td>Min</td>
<td>0.38</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>35</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>Diff</td>
<td>0.17</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>14</td>
<td>11</td>
<td>35</td>
<td>42</td>
</tr>
<tr>
<td>Span</td>
<td>18</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>15</td>
<td>12</td>
<td>35</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 5.5: The span of standard pulse forms in produced WISTOM units.

The total number of combinations is 2.46G.

Figure 5.12 indicates that D2 is directly dependent on the other length parameters, as to fix the length of the pulse! According to the responsible engineer, this is indeed a common practice and the differences can probably be attributed to the person trimming the unit. It is quite likely that this parameter can be removed, thus the dimension of the parameter space is reduced to 7. Then, the total number of combinations would come down to 70M.

Some used pulses differ more from the others. By simply considering these as outliers; the parameter is further reduced, extracting 10 such pulses made the space come down to 3.1M. By using a step length of 5 for the amplitude, the space is downsized to about 720k (see table below).

<table>
<thead>
<tr>
<th></th>
<th>PABC</th>
<th>C1</th>
<th>T1</th>
<th>D1</th>
<th>S2</th>
<th>C2</th>
<th>D2</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>0.53</td>
<td>11</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>70</td>
<td>116</td>
</tr>
<tr>
<td>Min</td>
<td>0.38</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>39</td>
<td>74</td>
</tr>
<tr>
<td>Diff</td>
<td>0.15</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>31</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Span</td>
<td>16</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>43 or 10</td>
</tr>
</tbody>
</table>

Table 5.6: The span of standard pulse forms in produced WISTOM units, when outliers have been removed.
Automatic trimming of ultrasonic pulse in fiber-optical power spectrometer

5.8.3.3 The default pulse

The starting pulse in the manual trimming process, referred to as the Default pulse, see Table 5.7. Length (in samples) of pulses (parts) from produced units using the standard pulse form, where outliers are removed.

Table 5.7: Parameters of the default pulse used as a starting point in the manual trimming procedure.

<table>
<thead>
<tr>
<th>PABC</th>
<th>C1</th>
<th>T1</th>
<th>D1</th>
<th>S2</th>
<th>C2</th>
<th>D2</th>
<th>Length</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.48</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>70</td>
<td>=100</td>
<td>100</td>
</tr>
</tbody>
</table>

It should be noted that this pulse has a C1=7, and therefore is not included in the reasonable pulse space described above.

5.9 Discussion

The trimming process does not seem to follow any apparent structural method, but seems rather random. Indeed, I cannot formulate a method even thought I can trim a unit myself. The seemingly only known relation is the effect of the PABC parameter. It is therefore not likely that a structural method will be found to solve the problem.

The problem can be viewed as an optimization problem. As such, we would like to have an objective function that is to be optimized by changing a set of variables that are subject to some constraints. The set of variables will in this case be the pulse form parameters (including PABC). The objective function is developed in chapter 6. In section 5.8.3 it is shown that it is possible to add some constraints to the variables.

The parameterization resulted in 8 parameters, but from experience with trimming, results might improve with a 9th parameter. The analysis of old pulse forms shows that it might be possible to lock one dimension, thus resulting in 7 (or 8) free variables.

By studying the pulses in trimmed units it is shown that there exists a limited number of reasonable pulses to consider, thus making it possible to add constraints to the variables. The number of reasonable pulses could be as low as 720*10^4, but it might be as many as 70*10^6 or even 2.4*10^9.

It is important to remember that a pulse that is good for one position in the grating might not be good in other parts. Therefore a compromise must be made. However this should not be too much of a problem since the method of trimming for one position at a time is used in the manual process. Thus this problem was not further investigated, a compromise scheme is shown in chapter 12.
It does seem like the OSA sweep can be moved out of the problem, to be solved by minor alterations of the PABC value as this is done in the manual method. Therefore, this measurement could be left out of the report. However, it is hard to prove that a method works without a more rigidly defined test for this property. Thus a definition and measurement method is developed in chapter 6.1.

For an automated method to work, it is important that the stability of the pulse testing methods is vastly improved. This will require modifications to the firmware, as well as to the way the pulse is programmed using the WISTOM LabVIEW API. Proximion let do some modification to the firmware. A method for stable programming is developed in 6.4.

The LabVIEW API for WISTOM is quite complete, it is therefore natural to use LabVIEW for further development.
6 QUANTITATIVE DESCRIPTION

In the manual test, the unit is first trimmed using a single laser source, and then finally trimmed to pass the TLS-OSA sweep test. In this chapter both tests, as well as pulse stabilization and measurement time cost, are quantified.

6.1 Power spectrum smoothness (TLS–OSA sweep)

6.1.1 Background

The transmitted amplitude is not only a function of the pulse form, but also dependent on the position of the corresponding wavelength in the grating. This is further a function of the individual grating. Thus, a perfect white light source, that has perfectly constant power along the wavelength axis, will have an AOSF spectrum with variable power. An example of the AOSF spectrum is in Figure 4.5. This curve is straightened by power calibration tables.

To be able to calibrate the power spectrum, the AOSF spectrum curve must not fluctuate too much, i.e. the curve must be smooth enough. The smoothness performance is evaluated by studying the OSA graph as described in 4.3.6. Even thought the performance limit is defined as “Check that the calibration curve measured with the OSA has a noise level of maximum +/-1dB”\(^{25}\), how the noise level is to be measured is not defined. According to the responsible engineer, the definition should be interpreted as “The amplitude curve must not do local jumps by more than 1dB”\(^{26}\), still, local jump is not strictly defined, but relies on the operator experience of how the curve should look to pass the calibration. Thus, to automate this step, the measurement has to be studied in more detail.

The TLS–OSA sweep measurement method requires that the AOSF output is connected to an OSA. To be able to connect the OSA, an optical connector is welded to the output fiber, as well as to the sensor input. After the trimming process is completed, these connectors are removed and the output is welded to the input. This is done before calibration to eliminate the losses of the no longer needed connectors.

6.1.2 Previous work

Proximion has found that the measurement should be possible to perform without an external OSA. There are many advantages to be gained by doing so, e.g. elimination of extra welding step(s) and a simplified trimming setup.

A small project to develop an OSA sweep measurement in WISTOM had already been started, but never finished. The code from this project was investigated.

The program uses a TLS sweep method, programming the TLS to sweep the wavelength region and then repeatedly during the sweep sampling the maximum peak amplitude found by WISTOM. The result is plotted in a graph.

There are several problems with this approach. Since the calibration table contains 1024 measure points, the spectrum has to be measured in at least twice as many points\(^{27}\), preferably more. But only one point is measured at a time, thus the TLS sweep-time must be carefully matched to the amount of needed subsampling and averaging in WISTOM. To get a read in every point of the needed spectrum, some kind of synchronization method with the TLS position or read wavelength must be performed. None of these issues were addressed in the implemented conceptual code.

\(^{25}\) "Production Instruction for Test Rig Pulse Shaping", Proximion internal doc no 100358

\(^{26}\) Phd Cecilia Lundevall, Proximion Fiber Systems AB, personal communication

\(^{27}\) According the Nyquist–Shannon sampling theorem
A threshold value must be defined to remove noise becoming measures when the TLS is sweeping outside the grating frequency. The TLS sweep time and WISTOM settings must be carefully tuned so that the laser does not move too much during a measurement, or the power measurement or peak position will be wrong.

While none of these issues are hard to solve, rather big improvements in the implementation code must be performed.

The implementation does not include any analysis of the resulting spectrum.

6.1.3 Choice of method

The method described in 6.1.2 Previous work requires a great deal of implementation, but this type of implementation is outside the scope of this project and the method is therefore not feasible.

Another method would be to set a long average time, exceeding the sweep time of the TLS to get the swept power spectrum. But this would result in a very low amplitude since the laser light would only reach the sensor during a very short time for every wavelength. Increasing sampling time will also accumulate the noise, resulting in a very noisy measurement. Due to the noise, this method is not feasible either.

Instead, a new method was developed, described below.

6.1.4 Defining 'local jump'

Due to the properties of the photodiode in WISTOM, and variations of the transmission in the AOSF for a given pulse, the measured power is highly dependent on wavelength and temperature. The power transmission curve is therefore adjusted by the calibration tables (see 3.3 Calibration tables). The tables have a resolution of 1024 points for frequency. The transmission power may therefore not oscillate too much between calibration points. With 1792*8 subsamples and 1024 calibration points, the oscillation must be limited within $1792^{*}8/1024=14$ subsamples.

Proximion let define the maximum amplitude difference within the 14 subsamples to be $1\text{dB}$.

6.1.5 New method

The power calibration uses a white light source to build the calibration tables. The white light source is not perfectly white, but the power spectrum is smooth enough. Here, smooth enough means that between calibration points, the white light spectrum may be considered as being linear. Using the same kind of light source, the needed power spectrum is given directly by WISTOM. The smoothness of the power spectrum $P(n)$ can therefore be measured by the maximum peak to peak difference $d$ within a moving window of length $w$ over the spectrum.

$$d(n) = \max (P(n+i)) - \min (P(n+i)), i = [n, n+w]$$

The performance is therefore defined as the maximum peak to peak difference found. The lower, the better.

$$W = \max (d(n)), n = [s(\lambda), s(\lambda)−w]$$

Where $s(\lambda)$ is the sample number of the shortest and the longest wavelength within the calibrated WISTOM spectrum.

6.1.6 Implementation

As the wavelength calibration is not yet performed, $s(\lambda)$ is not yet known. The end points of the spectrum are therefore measured by setting the TLS to the end points and register the peak sample.
The end points are given by the user, as well as the window length. The default is 1570 nm – 1528 nm with a window length of 14 subsamples.

An OSW is switched to the white light source, the averaging value specific for this test is set. The power spectrum (in dB) is read from WISTOM and the maximum peak to peak value within the moving window is calculated. The resulting values are plotted, and the worst window frame peak to peak value and position are presented to the user.

6.1.7 Code

The **DoWhiteSourceTest.vi** programmes each given pulse into WISTOM (see chapter 6.4) and waits for stabilization and data generation (see chapter 6.7). The optical power spectrum is read and sent to the **WhitePerformance.vi** (Figure 6.2) and the results are stored in arrays.
6.1.8 Results

The new method is by far more accurate than the manual method, no unit that passed this test has needed recalibration due to bad white light performance. This method is also used for automation of the performance test. A program controlled optical switch is used to select either a TLS input or a white light source.

Further, production WISTOM is simplified by reduction of the number of fiber weldings needed.

6.2 Laser performance measurements

As described in chapter 4.3.7 “Optical performance measurements stage”, there are eight different measurements of the optical response from a laser input. There is also a performance measurement using the TLS/OSA sweep described in 4.3.7.

Further, there are several subjective measurements:

- Enough peak amplitude, limit needed.
- The pulse should be 'even'. This could probably be measured by the ratio of left/right attenuation.
- Ghost peaks are measured by the OSNR measurements

These performance measurements are summarized in Table 6.1. The table also indicates if the measurement is directly readable from the WISTOM OPM system and if performance criteria exists. Missing definitions need to be defined.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Measure readable</th>
<th>Criteria exists</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWHM</td>
<td>Width (GHz) of peak halfway from the peak amplitude</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Attenuation at -12.5</td>
<td>Amplitude 2.5GHz from the peak center</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Attenuation at -25.0</td>
<td>Amplitude at -25.0GHz from the peak center</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Attenuation at +12.5</td>
<td>Amplitude at +12.5GHz from the peak center</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Attenuation at +25.0</td>
<td>Amplitude at +25.0GHz from the peak center</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>OSNR</td>
<td>Peak amplitude divided by highest side peak amplitude</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Sensor saturation</td>
<td>Maximum distinguishable input power.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Amplitude</td>
<td>Peak amplitude</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OSA sweep</td>
<td>Measure amplitude as a function of wavelength.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Symmetry</td>
<td>Measure the symmetry of the optical peak</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 6.1: Performance measurements and their need for definitions.

---

29 Mattias Holmkvist, Proximion, personal communication

---
6.2.1 Defining measurements

6.2.1.1 Amplitude

The transmission depends not only on the pulse, but also on the individual grating and the reflecting point in that grating. This in turn depends on temperature. To compensate for this the amplitude measure is calculated using constants in calibration tables. Since this calibration is dependent on the pulse, it is performed after the pulse trimming. Therefore the amplitude reported by WISTOM at this stage is a relative value, not comparable between units or working points. The unit should therefore be in dB (e.g. as opposed to dBm or mW).

However, in an automatic process the amplitude is an important measurement that must not be too low. To get an approximation of the amplitude, Proximion let calculate the average of calibration tables from a set of produced units. The deviation from average in the set was ±3dB.

As a result, the average value could be loaded into WISTOM together with flat calibration tables, so that WISTOM will report an approximation of the power. The accuracy of this power measurement, according to the deviation in the set, should be ±3dB.

6.2.1.2 OSNR

WISTOM has the ability to detect optical peaks in the spectrum. In the laser performance test, there is only supposed to be one peak. Let the amplitude of this peak be called $P$. If there are more peaks, these are called ghost peaks. By definition, ghost peaks will always have less amplitude than the main peak.

The definition of what the OPM system is to considered as a peaks is defined by several parameters. If these parameters are set so that unwanted ghost peaks are detected, it is possible to use the second largest found peak amplitude as a measurement of the noise level. Let this ghost peak amplitude be $G$. If no second peak is detected, an estimation of noise floor level is used. The amplitude is measured in a logarithmic scale (dB), why $OSNR$ was defined as

$$OSNR = P - G.$$  \hspace{1cm} \text{(13)}

In the trimming process, the main purpose of the $OSNR$ performance value is to avoid ghost peaks, therefore a rather arbitrary noise floor estimate of -40dB was substituted for $G$.

6.2.1.3 Sensor saturation

This measurement is not used for trimming and is therefore excluded.

6.2.1.4 Symmetry

An asymmetric peak is considered as one that has a larger attenuation on one side than on the other. When inspecting the peak, it is plotted with a logarithmic amplitude scale. Therefore a measurement of symmetry was defined as

$$S = A_- / A_+ .$$  \hspace{1cm} \text{(14)}

Where $S$ denotes symmetry, $A_-$ is the left side (longer wavelength) attenuation and $A_+$ is the right side attenuation.

Since attenuation is given at ±25GHz and ±12.5GHz, two corresponding symmetry values are calculated.
6.2.2 Defining limits

6.2.2.1 OSNR
Consulting the WISTOM specifications shows that an OSNR should be greater than 40dB.

6.2.2.2 Symmetry
Ad hoc values of 30 at 25GHz and 25 at 12.5GHz

6.2.2.3 Amplitude
The accuracy of the amplitude measurement is only within ±3dB before the calibration. An ad-hoc value of -12dBm was used when input light was at -9dBm.

6.3 Quality measurements, figure of merit

6.3.1 Quality measurement and good pulses
To be able to rank pulses, a single figure of merit is needed. The quality of a performance figure was defined as a weighted distance to the limit. Thus, a lower quality number is considered as better. Using fixed weights and limits, the figure of merit is a function of the measurement of a pulse with parameters θ, or \( Q(\theta) \). There are several ways of defining \( Q \), this chapter describes the definitions used during the project.

Let the different measurements be indexed by \( i \) and define

\[
p_i \quad \text{Performance} \\
l_i \quad \text{Limit} \\
w_i \quad \text{Weight} \\
q_i \quad \text{Quality}
\]

Then, for performance values where a lower value is considered better,

\[
q_i = \frac{p_i - l_i}{l_i} 
\]  
(15)

For performance values where a greater number is better, negate \( p_i, l_i \), and \( w_i \).

Let define a 'good pulse' as a pulse that meets all performance criteria. Those are characterized by \( q_i \leq 0 \forall i \).

6.3.2 Figure of merit
The single figure of merit was then defined as the sum of all quality measurements

\[
Q_S = \sum q_i w_i 
\]  
(16)

The drawback of the \( Q_S \) definition is that it is not possible to determine whether a pulse is 'good' by the merit function. To compensate for this, the quality was defined as

\[
Q_m = \sum \max (q_i, 0) 
\]  
(17)

It should be noted that using this measure it is not possible to compare 'good pulses'. The idea is that once all criteria are met, other properties might be of importance, and thus other weights or quality measures could be used.

To allow for ranking among pulses that meet all performance criteria, introduce new weights \( \hat{w} \), if those are set to \( 1/\hat{w} \) they will have the same effect as for values that do not meet the criteria.
Automatic trimming of ultrasonic pulse in fiber-optical power spectrometer

\[ Q_{MS} = \begin{cases} \sum_i q_i \hat{w}_i, & q_i \leq 0 \ orall \ i \\ \sum_i \max(q_i w_i, 0), & \text{else} \end{cases} \]

Using this definition, a good pulse will be characterized by a negative merit value.

Later, it was proven more efficient (see chapter 10.2.2) to use a weakest link analogy, using the worst quality, which is the highest value, as figure of merit

\[ \hat{q}_i = \begin{cases} q_i \hat{w}_i, & q_i \leq 0 \\ q_i w_i, & \text{else} \end{cases} \]

\[ Q_M = \max(\hat{q}_i) \]

6.4 System stability while modifying the pulse

The system can become instable, failing to keep the resonance frequency, when the pulse is modified. This is due to the z-regulator getting fooled when the pulse is heavily modified or even moved out of the RPS window. In manual trimming this is not a big problem, since the system can be restarted in 30 seconds and changes usually are small enough for the z-regulator to handle them safely. However, for an automated system loosing stability would be disastrous and the problem was therefore investigated.

The z-regulator relies on the position of the zero-crossing after the main lobe in the RPS signal. If this position is changed by a large amount, the regulator cannot lock on to it. Using the temp regulator would avoid this problem, but the performance of the system degrades. Trying to optimize a system in a suboptimal state is not a very good idea, why a method for changing the pulse without disturbing the z-regulator must be developed.

Since the system is not built for continuously reprogramming of the pulseform, the WISTOM firmware could at times lock and force a restart. These issues were identified and Proximion let develop new firmware to handle the reprogramming stress.

6.4.1 Implementation

6.4.1.1 Reg-z compensation

The ideal zero-crossing position, stored in regZ, is set during the resonance search. This value is therefore only valid for that particular pulse used while searching for resonance. If the main lobe is changed, the zero crossing will be moved and thus the regZ value must be modified.

The pulse is generated from the digital form using the same clock frequency as the RPS signal sampler, but the RPS is subsampled. RegZ is a decimal value measure in main samples. Thus a relative regZ value is calculated by subtracting the main lobe length of the active pulse. The new regZ value is then calculated by adding the length of the new main lobe.

The relative regZ value is calculated when the program starts. The number is then sent as a parameter to the pulse programming Vi.

6.4.1.2 Disable regulators

As to avoid timing problems, the z-regulator is disabled just before the pulse is programmed and the new reg-z value is set. In the first iteration, the temperature regulator was activated when the z-regulator was switched off. However, the stabilization time was not improved, even thought the system became stable even for large changes to the pulse.
Further investigation showed that the transition between regulators is not optimized\(^3\). The preferred method is therefore to disable the temperature regulator and lock the resonance to the current frequency.

### 6.4.1.3 Code

![Diagram of Set Pulse Form Vi](image)

### 6.4.1.4 Result

The method allows for stable reprogramming of the pulse, even for large modifications. Time to stabilize is measured in chapter 6.5.

### 6.5 Development time (stabilization time)

Because the pulse is generated in resonance, a number of pulses must be generated before the pulse is stable and peak resonance is reached. When developing WISTOM the number of pulses needed was measured to be of the order of tenths of pulses\(^3\). This would be equal to a stabilization time of less than 10 ms (80 μs/pulse × 100 pulses = 8000 μs = 8ms <10ms). However, experience from manual pulse trimming indicates that the stabilization time is much longer, noticeable by the operator, maybe up to a couple of seconds. The new pulse programming method developed in chapter 6.4 greatly decreased the time for stabilization, but it is still noticeable. This chapter aims to measure the stabilization time for the new method.

The figure of tenths of pulses was measured using a fixed resonance frequency (i.e. without regulator) from an idle state. In the manual trimming process the frequency regulator is active, and a pulse is active in the fiber. The latter should reasonably at most only account for a doubling of the stabilization time, not likely noticeable. But the active regulator could indeed have a great impact. The regulator system must be active while trimming or the resonance would quickly get lost.

It is important to know how long time one should wait from the programming of a new pulse until reading performance measurements. Therefore a program to measure the development time when making a change to the pulse was developed. The exact mechanism behind the stabilization time is outside the scope of this project, but might be interesting to study for improvements of pulse trimming time consumption.

### 6.5.1 Theory

The RPS signal might not have the bandwidth to exactly describe the mechanical pulse, but should be usable to determine the stability of the pulse by analyzing the signal change over time.

The basic idea is to read the RPS signals directly following the change of pulse and plot the RPS change as a function of time.

---

\(^3\) Mattias Holmkvist, Proximion, personal communication

\(^{31}\) Mikael Bergman, Manager mechanics R&D at Proximion Fiber Systems
Like the optical spectrum, the RPS signal is sub sampled and averaged by a selectable amount. The numbers selected set the balance between speed and accuracy of the measurement. The lower limit on the amount of averaging is 32.

Time needed to sample an RPS signal is estimated as

\[ t = 80 \mu s \cdot k \cdot S \quad (20) \]

where \( t \) is the time, \( k \) is number of averaging sweeps and \( S \) denotes number of subsamples.

The amount of averaging and subsampling affects the control system, and could therefore have an impact on the measured values. The normal values during trimming, as well as in the final product, are 8 subsamples averaged by 512 full sweeps. The time resolution would therefore be about 80\( \mu s \cdot 8 \cdot 512 \approx 328 \)ms. Since two pulses are needed, the first measurement could, at best, be retrieved after 656ms. This time would have to be improved in order to resolve the development time.

According to the engineer responsible for the control system, the subsampling is likely to have greater impact on stabilization time than a change in amount of averaging\(^{32}\). Thus averaging should be decreased before subsampling when higher precision in time is needed.

6.5.2 Method

- Select a pulse in the domain of ‘reasonable pulses’ and measure the level of variation in the RPS signal
- Select another pulse far from the first pulse, but within the domain. Measure the time until the variation is at the same level as the stable pulse.

To find the minimum time resolution the above steps should be repeated for different timing settings. Also different types of changes to the pulse should be performed.

6.5.3 Implementation

The difference \( d_n \) between the previous \( x_{n-1} \) RPS signal and the current signal \( x_n \), was defined as the mean of the absolute difference of each sample (indexed by \( i \) of \( N \) samples).

\[ d = \frac{1}{N} \sum_{i=1}^{N} |x_{1,i} - x_{2,i}| \quad (21) \]

Another way of defining the difference would be to use the average standard deviation in a set of two pulses.

\[ \sigma_i = \sqrt{\left( \frac{1}{M} \sum_{n=1}^{M} (x_{n,i} - \bar{x})^2 \right)} \]

\[ s = \frac{1}{N} \sum_{i=1}^{N} \sigma_i \quad (22) \]

It is easy to see that this is the same type of measurement:

\(^{32}\)Rickard Terfelt, personal communication
The program was therefore changed to measure the average standard deviation as in equation (22). The benefit is that the program now can allow the user to select the number of pulses to use for difference calculation, simply by extending the set of pulses from two to a selectable amount using a FIFO buffer as a moving window. This allows reduction of the number of averaging done on the RPS signal in WISTOM, thus increasing the time resolution in the measurement, while keeping the effect of noise low.

Initially the start pulse is set, when the user presses the set in WISTOM button, the new pulse is set and the reading of the RPS signals begins. If the reading loop is set too tight, or if the WISTOM processor is overloaded, it happens that the same pulse is read twice. If this is the case, the duplicated pulse is discarded. The time axis is defined by the time the latest signal is read. Before reading the next signal, the program waits the time estimated by equation (20) to complete a signal.

It is also possible to view the first individual signals. The number of signals viewed is set by the window size parameter.

Figure 6.4: Screenshot of the Find Development Time program. The pulse has been changed from Default to Max. The mean standard deviation is plotted in the lower graph, here indicating a stabilization time of 600ms.
6.5.4 Result

When setting the subsampling to 8 and averaging to 32, the time resolution should be $80 \mu s \times 8 \times 32 = 20480 \mu s = 20 \text{ms}$. But surprisingly, using this setting the time resolution is 10 times as long. Changing the subsampling to 4 should cut the resolution in half, but the signals are read at intervals as low as 14 ms, close to the theoretical interval of 10 ms. The reason for this behavior remains an open question.

The tests were performed with 4 subsamples, 32 averaging and a window size of 2 or 4. As start pulse, the default pulse was used. The maximum and minimum pulses from the reasonable pulse space were used as end pulses. The total length was set to 100. See Table 6.2.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>T1</th>
<th>D1</th>
<th>S2</th>
<th>C2</th>
<th>D2</th>
<th>Length</th>
<th>A2</th>
<th>PABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start (Default)</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>70</td>
<td>100</td>
<td>100</td>
<td>0.48</td>
</tr>
<tr>
<td>End (Max)</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>17</td>
<td>15</td>
<td>32</td>
<td>116</td>
<td>100</td>
<td>0.55</td>
</tr>
<tr>
<td>End (Min)</td>
<td>8</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>78</td>
<td>100</td>
<td>100</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 6.2: Pulses used for stabilization time measurements.

![Figure 6.5: Mean standard deviation of RPS signal when Default pulse is changed to Max pulse.](image)

![Figure 6.6: Mean standard deviation of RPS signal when Default pulse is changed to Min pulse.](image)

In Figure 6.5, the pulse was changed from the Default pulse to the Max pulse, stabilization time seems to be about 600 ms. Note that the time axis starts at 200 ms. Measure of Default to Min has a quicker stabilization of 400 ms, as seen in figure Figure 6.6.

To be on the safe side, the stabilization time was set to 750 ms. Future work should try to determine the differences in unit variation and dependencies of how the pulse is changed. This could evolve in algorithms using different stabilization times for different pulse changes, or to add the stabilization measurement method to determine when the pulse is stable.

An interesting observation is that some pulses are not as stable as other pulses. Measure of stability could therefore be a quality measurement in future product improvements.

6.6 Data accuracy (averaging)

The precision of a performance measurement is determined by the number of scans used by WISTOM to calculate the spectrum. But the more scans used, the more time is needed for each measurement.
Therefore it is important to not use more scans than necessary, while using too few could lead to bad results which in turn could lead to an incorrectly rated a pulse in comparison to another.

The specifications given for the accuracy and performance of WISTOM in the white paper do not concern the precise filter shape measurements needed for the trimming. Therefore, a program measuring the accuracy of the performance parameters was developed.

6.6.1 Method

Using a stable light source, the performance values should ideally be equal between measurements. Therefore, the standard deviation of a set of measurements measured close in time is a measure of the unbiased error.

6.6.2 Implementation

First the number of averaging scans is set. The time needed by WISTOM to collect the data is calculated by multiplying the number of scans by the number of sub scans (eight) and the rough estimate of 0.8 ms per subscan.

The performance data is repeatedly read from WISTOM, waiting the calculated time before each read so that a new measurement will be read. The number of times to repeat is set by the user. The standard deviation and mean value of each quality value is calculated. The standard deviation in percent of the mean value is stored in an array, browseable by the user.

6.6.3 Result

One should be careful selecting the pulse for the test, an optimal pulse might not be representative for the trimming process. In this case, a pulse resulting in a ghost peak was selected, since it is known that this will have a large effect on the OSNR value.

With a low amount of averaging, small ghost peaks risk getting hidden in the noise. By direct experimentation it was found that 2048 average scans were needed to detect the ghost peaks.

With this rather high minimum requirement, it was decided to use this number without further analysis.

6.7 Measurement time cost

The cost to test a pulse is the time needed to program it in WISTOM until the data is collected. This time is limited by the time needed to wait between programming the pulse and reading the performance data.

The optical response is sampled by the FPGA that creates the averaged spectrum. When reading the spectrum you will always get the latest fully completed averaged spectrum. But there is no way of knowing exactly when the sampling of this spectrum was started. Nor are the spectrums numbered. To make sure the averaging did not start before the pulse was stable, one period is discarded.

Hence, to be sure that we analyze a spectrum of $N_{sub}$ subsamples, built on $N_{avg}$ averages and from a newly programmed and stable pulse we have to wait:

$$t_{wait} = t_{stable} + 2 \cdot t_{avg}$$

$$t_{avg} = N_{sub} \cdot N_{avg} \cdot 80 \mu s$$

For 2048 averages and 8 subscans, the time cost, excluding the actual programming and instrument communication comes to:

33Rickard Terfelt
\[ t_{\text{avg}} = 8 \cdot 2048 \cdot 80 \ \mu s \]
\[ t_{\text{wait}} = 750 \text{ ms} + 2 \cdot t_{\text{avg}} \approx 3370 \text{ ms} \]

The time for programming and communication is in the order of milliseconds and is therefore ignored here.

Indeed, the measurement is quite a costly process. A possible way to reduce the cost would be to allow control of the averaging start, or to number the spectrums so that a new spectrum can be identified.

Modifying the WISTOM firmware to allow restarting of the average sampling would result in a time cost of \( t_{\text{wait}} = 750 \text{ ms} + 1 \cdot 8 \cdot 2048 \cdot 80 \ \mu s \approx 2061 \text{ ms} \), or close to 40% reduction.
7  PROPOSAL OF AUTOMATION METHODS

This chapter contains a brief overview of the different ad hoc methods that were considered in the first stage.

7.1  List of methods

7.1.1  Structural method

7.1.1.1  Description

Use pattern recognition or other statistical method to find general correlations between pulse and optical performance parameters. The known correlations could be used to reduce the dimension of the problem to a lower order optimization problem.

7.1.1.2  Prerequisites

Statistical methods rely on performance data from different pulses. Thus a great pulse domain would have to be measured with respect to the performance parameters. A method for measuring the pulse domain must therefore be developed before this method can be used.

7.1.1.3  Advantages and Disadvantages

A structural method would probably be the fastest method, if such a method exists. The biggest problem is then that we do not know if it does, and even if it does, it might be very difficult to find.

7.1.2  Brute Force

7.1.2.1  Description

To be able to rate a pulse as better than another, a single figure of merit has to be calculated. Once such a figure is defined, the best pulse among a set of pulses can be chosen.

By testing every reasonable pulse definition and measure the optical performance, the best pulse for the tested optical wavelength can be found. The key aspects to make this method work are:

- Define pulse definition in as few parameters possible
- Define a reasonable (i.e. limited) pulse domain
- Define measurable performance values
- Define a single figure of merit based on performance values

7.1.2.2  Prerequisites

- A method able to change the pulse while keeping the system stable.
- A fast method for changing the pulse and reading the resulting performance parameters.
- A limited pulse domain.

7.1.2.3  Advantages and Disadvantages

While this method is guaranteed to find a suitable pulse, if it exists within the 'reasonable' pulse domain, it could be very time consuming. The time needed depends on how large the 'reasonable' domain is, and the amount of time needed to test the pulse.

It should also be noted that if the performance is measured from a single laser source, the performance is only valid in that working point of the grating. As in the manual procedure, the measurements could be repeated for different wavelengths.
7.1.3 Brute Force Enhanced

7.1.3.1 Description

Step each parameter by larger steps to find regions of interest and use the brute force method on these regions. The success of this method depends on the smoothness of the optical performance space as function of the pulse definition.

7.1.3.2 Prerequisites

The optimum pulse must exist in a sub domain with some identifiable characteristic (e.g. good merit value) so that areas of interest can be selected. To determine this, the pulse domain would have to be measured and analyzed.

7.1.3.3 Advantages and Disadvantages

The idea is to be faster than the BF method, but this method might still be slow. Also, there is no guarantee to find the best pulse, as is guaranteed with the BF method.

7.1.4 Optimization method

7.1.4.1 Description

Use some general optimization method that is applicable on this high dimensional, discrete problem with expensive measurements.

7.1.4.2 Prerequisites

The parameter space must be measured with respect to performance functions, and preferably one merit function.

If more than an ad-hoc method should be used, some literature studies will be needed.

7.1.5 Performance space estimation by Neural Network

7.1.5.1 Description

Estimating the performance space using neural network based on the result of a set of test pulses from the unit to be trimmed. The general idea is to identify regions of interest by using two estimates of the space, finding the residue and make new measurements in those regions to create the next better model.

7.1.5.2 Prerequisites


7.1.6 Self learning system

7.1.6.1 Description

Use a self learning system on the units being produced.

7.1.6.2 Prerequisites

A larger production volume. The current production volume does not allow for more than a couple of units a month, making the basis for self learning systems very weak.

7.2 Choice of initial method

With the very high sweeping frequency of WISTOM and a pulse change theoretically only needing about 10 sweeps to reach stable amplitude\(^{34}\), testing speed might only be limited by data accuracy and API communication performance. If it is possible to test 100 pulses a second, the limited set of pulses

\(^{34}\) Mikael Bergman, technical expert Proximion, personal communication
estimated in 5.8.3 of 720k pulses is tested in two hours. Testing them in all three working points, the automatic trimming would need 6 hours, which is considered as a feasible time. This is a very optimistic estimate, that later was proven largely inaccurate.

Selecting and extending the BF algorithm to also save all measured data to disk, the method would be useful for measuring the performance space as a function of pulse parameter space. Thus, even if this method turns out to be too time consuming for pulse trimming, it will provide a useful measurement tool for data collection. Among other things, this data would enable feasibility studies for different methods, such as optimization algorithms or enhancements to the BF method.

Further, most of the problems needed to be solved in an implementation, such as automatic performance evaluation, merit function, measurement methodology, as well as software implementations are needed for the other automatic methods as well.

Thus, even if the BF method shows to be too time consuming for production purposes, the implementation would still be very useful. It was therefore decided to implement the BF method.
8 IMPLEMENTATION OF THE BRUTE FORCE METHOD

For usage instruction, see appendix A.

8.1 Method outline

- Set the TLS to the wavelength that should be tested
- Define the range of every parameter and create all possible combinations of those parameters.
- Repeat for every combination:
  - Program a combination in WISTOM, wait for stabilization time plus the twice the time needed to create a new spectrum. (Chapter 6.7 Measurement time cost).
  - Read the peaks detected by the WISTOM OPM system, and the attenuation values of the largest peak.
  - Calculate the performance values. (Chapter 6.2 Laser performance measurements)
  - Calculate quality figures and the figure of merit, based on the limits and weights given by the user. (Chapter 6.3 Quality measurements, figure of merit)
  - Save performance and quality values, as well as the tested pulse form
  - Compare the figure of merit to the best figure of merit found, replace if better.
- Program the best pulse found.

8.2 LabVIEW implementation

For sake of clarity, the code is sometimes simplified. Mainly GUI, error handling and exception/assertion code have been removed.

In the main loop (Figure 8.1), each pulse in the Pulses to test array is tested. The sub Vi's denoted Test pulse and Quality are described below. The results are saved to file. The best pulse found and its performance values are stored (and returned to the calling Vi).
8.2.1.1 Test pulse Vi

The Test pulse.vi (Figure 8.2), first sets the given pulse into WISTOM using the setPulseForm method developed in chapter 6.4. It then waits the time calculated according to equation (24) before reading the peak information from the OPM system. The OPM values are then converted to performance values, see Figure 8.3. Note that a noise floor is set to -40 dB here, this is to avoid infinity values which can inflict some types of figure of merit definitions (e.g. $Q_{MS}$).

8.2.1.2 Performance to Quality Vi

The PerformanceToQuality Vi (Figure 8.4) calculates $Q_M$ and $Q_s$ defined by equation (19) and (16) respectively. But the function only uses one set of weights.
9 EMPIRICAL DATA COLLECTION

9.1 Method

Defining pulse parameter regions of interest, the Brute Force program was run on three different units. Depending on pulse parameter space, the data collection could take from days to weeks.

The pitfall however, was that it was hard to tune the WISTOM OPM system parameters so that it would detect ghost peaks. The OPM system is normally tuned to ignore this type of false data, whereas to distinguish poor pulse forms, ghost peaks need to be detected.

Due to the long execution time and the iterative program development of the Brute Force pulser, some data collection were done with different program versions. The performance measurements will therefore sometimes differ between measurement sets due to the use of earlier stages of the program.

The OSNR measurement was in this development iteration of Brute Force defined as the maximum ghost peak. In addition, the amplitude measurement was early given in W/Hz, later in dBm.

For an explanation of how the BruteForce program is operated, the resulting files and their format, and other details, please refer to Appendix A.

Some Matlab scripts and functions to load the resulting data, calculate different types of figure of merit, and plot the data, were created.

9.2 Equipment setup

The TLS was connected to an attenuation module to allow for adjustment of the light power. The attenuation output was then connected to a set of cascaded light splitters, splitting the light in two equal beams and input those into two equal light splitters, resulting in four light outputs of equal power. One of the outputs was measured with an OPM and the attenuation module was adjusted so that the OPM would read -10dBm. To verify the connection, each output was measured; the difference being within +/-0.5dB was considered acceptable. The setup can be viewed in Figure 9.1.
The largest pulse domain tested consists of 504000 measured pulses. The measurements were performed using 8 subsamples and 256 averaging counts. Waiting time was about 1.1s, resulting in about 3000 pulses per hour, execution time was up to one week. The results were saved to file for further analysis using Matlab, see chapter

<table>
<thead>
<tr>
<th>Collection Name</th>
<th>Unit id</th>
<th>Pulse count</th>
<th>Pulse Dimension</th>
<th>Amplitude unit</th>
<th>Reliable OSNR</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>238_2</td>
<td>238</td>
<td>504000</td>
<td>8</td>
<td>W/Hz</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Island</td>
<td>19</td>
<td>233295</td>
<td>8</td>
<td>W/Hz</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Tre_3</td>
<td>3</td>
<td>378000</td>
<td>8</td>
<td>dB</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Nitton_1</td>
<td>19</td>
<td>299520</td>
<td>7</td>
<td>W/Hz</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Nitton_3</td>
<td>19</td>
<td>215240</td>
<td>8</td>
<td>dBm</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Tjugoett_1</td>
<td>21</td>
<td>345611</td>
<td>7</td>
<td>W/Hz</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.1: Summary of collected data
9.3.1 Parameter space

| Name   | 238_2  | | Name   | Tre_2  | |
|--------|--------| |--------|--------|
|        | P      | C1 | T1 | D1 | S2 | C2 | T2 | A2 |
| Start  | 0.42   | 6  | 4  | 6  | 5  | 4  | 5  | 86 |
| End    | 0.52   | 11 | 8  | 9  | 9  | 8  | 20 | 110|
| Stepping | 0.02 | 1  | 1  | 1  | 1  | 1  | 5  | 4  |

| Name   | Nitton_1 | | Name   | Nitton_3 | |
|--------|----------| |--------|----------|
|        | P        | C1 | T1 | D1 | S2 | C2 | T2 | A2 |
| Start  | 0.40     | 8  | 4  | 6  | 4  | 4  | 0  | 74 |
| End    | 0.50     | 11 | 8  | 9  | 9  | 11 | 0  | 110|
| Stepping | 0.02   | 1  | 1  | 1  | 1  | 1  | 0  | 3  |

| Name   | Nitton_3 | | Name   | Nitton_3 | |
|--------|----------| |--------|----------|
|        | P        | C1 | T1 | D1 | S2 | C2 | T2 | A2 |
| Start  | 0.44     | 5  | 5  | 6  | 8  | 4  | 5  | 86 |
| End    | 0.50     | 11 | 8  | 9  | 11 | 8  | 15 | 114|
| Stepping | 0.02   | 1  | 1  | 1  | 1  | 1  | 5  | 4  |

From the experience with manual trimming, we know that the secondary pulse might benefit from a flat top, i.e., using a non-zero $T_2$. However, this is not part of the standard pulse we found in the majority of produced units. I believe, although it is not proven, that the addition of this dimension will reduce the span of the secondary pulse amplitude in the set of reasonable pulses. As it turns out, in many cases the globally best pulse had a nonzero $T_2$, a good indication that it should be included. Thus the parameter space was increased to 8 dimensions. Three of the above large sets do include the $T_2$ parameter.

Nitton_3 has good measurements of ghosting peaks and therefore a useful OSNR measure. 215040 pulse forms were tested, merely 549 (0.2533%) passed all performance criteria. However, the low amount of passed pulses is largely an effect of the used parameter-space, while the OSNR measurement has less effect.\(^{35}\)

\(^{35}\) By studying which performance measurements that failed the pulse.
10 DATA INVESTIGATION

An in depth theoretical study of optimization techniques for problems involving ‘expensive’ measurements and many but short dimensions, would be beneficial for selecting what properties of the merit functions to study. However, due to the time constraints of this project, only brief investigations on the data were brought out. The aim is to find out which of the proposed methods to empirically test next.

10.1 Reduction of parameter space using PCA

10.1.1 Theory

The high dimensional order of the problem makes the data analysis hard. The performance space was reduced to one dimension by the merit function; it would be interesting to see if it is possible to reduce the number of dimensions in the parameter space.

“Principal Components Analysis is often the first stage in a data analysis and is often used to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original dataset.”

A reduction in dimensionality would mean that the data is easier to visualize and analyze, but it could also be used for parameter stepping and reduction of parameter space in the trimming algorithm.

Since principal components are dependent on the scale of the original variables, the data should be standardized. Transforming the original parameter set to have zero mean and unit variance will give the variables equal importance.

Given a standardized subset of the tested parameters, e.g. the set of ‘good’ pulses, a PCA algorithm will give the principal components spanning that subset. The principal components therefore form a new parameter basis. The base vector corresponding to the lowest eigenvalue has the lowest variance, meaning the set has the least spread along this vector. Thus the least amount of information will be lost if this vector is removed. If such vectors can be found and removed, the dimension of the problem can be reduced.

PCA can also be used to limit the parameter space in which we should be looking for good pulses. All combinations in the original space might not be very likely to work, such as a combination of very small parameter values, or very large parameter values. A PCA method does not require us to understand the physical principles to create such a parameter space. The described method below tries to do this.

Another way of seeing the use of this method is the attempt to introduce physical parameters in the parameterization (chapter 5.5.2.2). That method failed because it is hard to understand the correlation between the pulse parameters and the optical properties of the transmitted light. Using PCA, such correlation can be found without any knowledge of the system.

10.1.2 Method

- Define a subset of the pulse definition space, for example the set of pulses that meet all performance criteria. Or the subset resulting in an interesting optical property we would like to find correlation to.
- Standardize each parameter in the pulse definition subset space to have zero mean and unit variance.
- Run PCA to find a new basis for the subset.
- Plot figure of merit (y-axis) as a function of each parameter (x-axis).
- Reduce the parameter space by cutting each principal component outside the outermost good pulse.

To determine how efficient this method is, each parameter is limited by the, for that parameter, outermost good pulse. Comparing the number of pulses within these new limits to the original set will determine how effective the new parameter space is.

10.1.3 Implementation

A Matlab script was created that
- Standardize each parameter by $\Phi_i - \text{mean}(\Phi_i) / \text{stddev}(\Phi_i)$, where $\Phi_i$ is the set of tested values of parameter $i$.
- Runs PCA on the subset with the selected property (here good pulses)
- Projects the complete parameter space on the subset PCA base vectors.
- Plots the investigation property (here figure of merit) as a function of each base vector.

10.1.4 Results

First let plot the figure of merit for each parameter in the original but standardized basis (parameter basis).

![Graphs](image.png)

Figure 10.1: Numbered from left to right, top to bottom the plots represent the change in parameter $C_1$, $T_1$, $D_1$, $S_2$, $C_2$, $A_2$ and $PABC$. The x-axis contains the (standardized) parameter value. The y-axis contains the figure of merit $Q_S$. Pulses that meet all criteria are marked with red color. Dataset is nitton_1.
As can be seen from Figure 10.1, good pulses are well spread out in the discrete parameter space, there is no form of clustering. Decreasing the size of any of the x-axes would result in lost good pulses, and therefore no reduction of the parameter space can be done.

Running PCA on the subset of good pulses gives a new parameter basis. By creating the same plots as Figure 10.1, but with the new principal components on the x-axes gives the Figure 10.2.

Figure 10.2: Each plot represents one of the principal components with the principal component value on the x-axis and the figure of merit $Q_s$ on the y-axis. Pulses that meet all criteria are marked with red color. Dataset is *nittion_1*.

Here, some clustering can be seen. By cutting each x-axis at the outmost good pulse, the parameter space can be reduced, without losing good pulses. This reduces the parameter space from 299520 combinations down to 171680, or 57% of the original set. While this is a reduction, it is still not enough to make the BF method feasible. One should also keep in mind that this space is only valid for this particular unit; the reduction may not be as efficient when combing data from several units.

One could argue that the number of good pulses in the set is exaggerated due to the faulty OSNR measurement. The data set *nittion_3* from the same unit has a reliable measurement, but is performed on a different set of parameters. This set does not overlap at all with the above set, but still contains a great number of good pulses, an indication of that the parameter space used in run *Nitton_1* is too small. Further studies would therefore need an even larger dataset, and of course benefit from a correct OSNR measurement.
Figure 10.3 Numbered from left to right, top to bottom the plots represent the change in parameter $C_1, T_1, D_1, S_2, C_2, T_2, A_2$ and $PABC$. The x-axis contains the (standardized) parameter value. The y-axis contains the figure of merit $Q_s$. Pulses that meet all criteria are marked with red color. Dataset is nitton_3.

The test contains 215040 pulses, of which only 549 is considered as good. In this set it is obvious that some parameters are not limiting in one direction, and therefore that part could easily be left out using the same method as for PCA but in the original parameter space. This reduces the parameter set to 40320 pulses, or 19% of the original set.
Figure 10.4: Each plot represents one of the principal components with the principal component value on the x-axis and the figure of merit $Q_s$ on the y-axis. Pulses that meet all criteria are marked with red color. Dataset is nitton_3.

Using PCA the clustering is even more obvious in this set, and the space is limited to only 14734 pulses, 6.9% of the original set.

10.2 Smoothness of the figure of merit functions

Due to time constraints for the project, instead of studying each performance parameter, the two merit functions $Q_s$ (equation (16)) and $Q_M$ (equation (19)) were studied. Since gradient based method would tend to converge to local minima, the number and spacing of local minima in the space were studied. It would be interesting to study if the merit function has some convex properties.

Plotting a function of more than three variables is hard, the merit is a function of 7 or 8 parameters making visualization of the space difficult.

10.2.1 Minima of the merit function

10.2.1.1 Theory

The count and spacing of the merit function minima will give a hint of how many, and how far apart, the starting points of a steepest descent-like method need to be.

However, we need to define what a local minimum is in our domain, since it is not continuous. A local minimum of a function is defined as a point $x$ where the function of all points in the neighborhood is greater. Thus we need to define what the neighborhood is.
If the neighborhood is defined by all points where the parameter has changed at most 1 step, the neighborhood of a 2 dimensional space would have 9 points (including the center point), three dimensions would have $3^3=27$ points. Following this it is easy to see that for eight dimensions the neighborhood would be quite large ($3^8=6561$). Therefore the neighborhood was defined as "all points where one parameter has changed (at most) one step". In eight dimensions this neighborhood consists of 16 points, excluding the center point. The closest possible Euclidian distance between two minima with this definition would be $\sqrt{2}$.

10.2.1.2 Method

- For $Q_M$ and $Q_S$
  - Find the local minima points that are not on the border.
  - Find the minimum distance between minima
  - Find the mean distance between minima

The distance is defined as the 2-norm of the number of discrete steps between two pulses. For the PABC parameter that is not discrete, 0.01 was considered as one step.

10.2.1.3 Implementation

As the data from BF comes in large tables of pulse parameters and results, the search for the result of one particular pulse combination was simplified by a multi dimensional index matrix, one dimension per input parameter.

With the index matrix in place, a simple Matlab script finds the local minima by iterating through the inner space, finding minima by comparing the merit function of its neighborhood.

10.2.1.4 Results

<table>
<thead>
<tr>
<th>Data set</th>
<th>Count</th>
<th>Inner Count</th>
<th>$Q_S$ Minima count</th>
<th>Shortest distance</th>
<th>Mean distance</th>
<th>Change closest</th>
<th>$Q_M$ Minima count</th>
<th>Shortest distance</th>
<th>Mean distance</th>
<th>Change closest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tre_2</td>
<td>378000</td>
<td>4320</td>
<td>11 (0.25%)</td>
<td>1.41</td>
<td>3.32</td>
<td>$C_2,A_2$</td>
<td>19 (0.44%)</td>
<td>1.41</td>
<td>3.12</td>
<td>$C_1,C_2$</td>
</tr>
<tr>
<td>Nitton_3</td>
<td>215040</td>
<td>1440</td>
<td>3 (0.21%)</td>
<td>1.41</td>
<td>1.73</td>
<td>$C_1,A_2$</td>
<td>5 (0.35%)</td>
<td>1.41</td>
<td>2.67</td>
<td>$C_2,A_2$</td>
</tr>
<tr>
<td>238_2</td>
<td>504000</td>
<td>8640</td>
<td>73 (0.86%)</td>
<td>1.41</td>
<td>3.47</td>
<td>$S_2,C_2$</td>
<td>39 (0.45%)</td>
<td>1.41</td>
<td>3.20</td>
<td>$S_2,T_2$</td>
</tr>
</tbody>
</table>

Table 10.1: The number of minima found (excluding border points) using the $Q_M$ and $Q_S$ merit function. Weights were set to 1.

Note the huge amount of pulses that are on the border (i.e. not an inner point), the actual analyzed space is quite small. Using a larger neighborhood space could even exhaust the entire inner space. The close proximity of the closest distances represents a problem. (Since the optimal solution is a global minimum, we would like to have as few local minima as possible if optimization techniques are to be used.)

It might be possible to make the shortest distance larger by selectively extending the neighborhood definition to allow for a change in two parameters simultaneously, e.g. allow a change in $C_2$ together with an arbitrary parameter.

10.2.2 Merit function development from a fixed point

By fixing all but one parameter and plot the figure of merit as a function of the free parameter, it is possible to see some characteristics of the merit function.

If the function is smooth, the development of the merit function is smooth as one parameter is changed while all other parameters are fixed. Since such a plot is easy to produce, various plots from different local minima, measured sets, and merit functions were analyzed.
10.2.2.1 Method

- Selecting the global minimum with respect to some figure of merit measurement.
- Lock all but one parameter.
- Plot the figure of merit as a function of the free parameter.
- Repeat for all parameters.

10.2.2.2 Results

These figures show how the figure of merits develops as we change a single parameter, while keeping the other parameters fixed. The fixed parameters are set to the global minima, as defined by one of the figure of merit functions.

Figure 10.5 indicates that \( Q_s \) (equation (16)) might be a rather nice function, except for the secondary climb time parameter that introduces several local minima. For most parameters the function is convex, this is promising, since sub gradient optimization can be used in such cases\(^\text{37}\).

Figure 10.5: Merit function \( Q_s \) development from \( Q_s \) global minimum, set \( N_{itchion_1} \).

Figure 10.6: Merit function \( Q_M \) development from \( Q_s \) global minimum, set \( N_{itchion_1} \).

Using the same data, centered at the same position but with the \( Q_M \) merit function (equation (19)) provided even better results, as can be seen in Figure 10.6. The climb time of the secondary pulse does not have the same bumpy effect on the merit function as in the previous case.

The same behavior could be found on the other datasets see Figure 10.7 and Figure 10.8. Here the parameters are fixed at the global minimum for \( Q_M \). Just as in the previous case, the \( Q_M \) function behaves better than \( Q_s \).

\(^{37}\) "Practical methods of optimization" by R. Fletcher, page 364, ISBN 0471915475
10.3 Discussion

The brief data investigation in this chapter shows that much could be learned by collecting data with the Brute Force Pulser tool. Further studies would benefit from larger datasets, as well as equal datasets from several units.

PCA forms an interesting way of limiting the set of pulse parameters, but the parameter set will still be too large for the BF method to be efficient. However, it could most likely be used to form parameter sets for other methods. PCA might be a very efficient way to improve a stepped search algorithm, both in terms of reduced parameter space, as well as using it to create smart stepping directions and allowing variable step-sizes.

However, the investigation of how the two merit functions $Q_s$ and $Q_M$ behaviors when one parameter is stepped is much more interesting. By studying the plots, we learn two important things.

1) When stepped in one direction, the merit function is convex, or piece wise convex

2) The $Q_M$ merit function behaves much better with regards of being convex

Of course, this does not mean that the merit function of the multidimensional space spanned by the parameters is convex. But the first discovery tell us that a method where we search for better values by stepping might work. The second discovery tell us that $Q_M$ merit function is better for such method. Before spending more time on data investigation, a simple implementation of such search algorithm on the measured datasets should be implemented.
11 ALTERNATING VARIABLES METHOD

In chapter 10 it was found that the merit function of each parameter are convex, or piece-wise convex. Of course, this does not mean that the merit function of the multidimensional space spanned by the parameters is convex. But it is an indication that an Non-Smooth Optimization (NSO) algorithm could work.

According to Fletcher, “The simplest method for basic NSO is an analogue of the steepest descent method…”\(^{38}\) The method is called the subgradient method. Here, the method is further simplified into the ad-hoc method generally called Alternating variables method in the literature. But in the code and program documentation it is referred to as the Steepest Descent method, or SD for short.

Investigations of the collected datasets showed that the merit function has some nice characteristics (see 10.2.2). But this test is far from a proof that a particular method would converge. To evaluate the feasibility of the proposed method, it was first implemented in Matlab and tested on the measured datasets.

11.1 Theory

Select a starting point as core point, then repeatedly change the core point to the neighbor with lowest (best) merit value. The minimum is found when the core point is the lowest point in the neighborhood. By testing all neighbors convergence is assured (for that choice of neighborhood).

11.2 Method evaluation on measured data

The method was implemented as a recursive function. If a neighbor is outside the dataset, the figure of merit is substituted with a very high value.

The method has a tendency to move out to the borders of the dataset, indicating that the measured sets are too small. But the method does seem to work very well. It was run on several datasets, see Table 11.1. For example the set Nitton_3 contains 215040 pulses, but only 549 meet the performance criteria. Still, 682 of the 1000 randomly selected starting points led to a good pulse.

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>Good count</th>
<th>Random start points</th>
<th>Found good</th>
<th>Max steps</th>
<th>Min steps</th>
<th>Mean steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitton 3</td>
<td>215040</td>
<td>549</td>
<td>1000</td>
<td>682</td>
<td>22</td>
<td>1</td>
<td>8.938</td>
</tr>
<tr>
<td>Tre 2</td>
<td>378000</td>
<td>59983</td>
<td>1000</td>
<td>993</td>
<td>15</td>
<td>1</td>
<td>6.115</td>
</tr>
<tr>
<td>Tre 2*</td>
<td>378000</td>
<td>59983</td>
<td>1000</td>
<td>221</td>
<td>20</td>
<td>1</td>
<td>7.329</td>
</tr>
<tr>
<td>Zip_2</td>
<td>504000</td>
<td>29322</td>
<td>1000</td>
<td>981</td>
<td>16</td>
<td>1</td>
<td>6.387</td>
</tr>
</tbody>
</table>

Table 11.1: Alternating variables method simulated on measured data, all using merit function \(Q_m\) except for the test marked (*) that uses \(Q_s\).

It should be of no surprise that the \(Q_s\) merit function is really bad for this method, since a local minimum with a merit value well below zero is not guaranteed to meet all performance criteria.

11.3 Feasibility of the method

With 16 tested pulses per iteration and an average of 7 iterations per starting point, a test with 128 starting points gives about 14500 pulses to test. With 3.4s per tested pulse, it should take less than 14h for one working point (i.e. TLS wavelength). A method to find a pulse that works in all three working points used in the manual trimming procedure has to be found. Letting the other two points consume the same amount of time gives an execution time of 42 hours, which is still feasible.

\(^{38}\) “Practical methods of optimization” by R. Fletcher, page 384, ISBN 0471915475

PROXIMION
Proximion Fiber Systems AB mail: info@proximion.com web: www.proximion.com phone: +46 (0) 8 750 48 88 fax: +46 (0) 8 750 48 80
address: Skalholtsgatan 10 B, SE-164 40 Kista, Sweden vat no:SE556641515301
Due to the promising results of the evaluation on measured data, it was decided to implement the method into the Brute Force program.

11.4 Implementation outline

11.4.1 Code

In place of the main loop for the BF method, there is a similar loop. Instead of just testing the pulses given, this loop runs the alternating variables method starting at each given location, called StartPulse. Here, the method is called steepest descent method. See Figure 11.1. The output is a minimum point for every given StartPulse.

![Figure 11.1: Outer loop for the Steepest Descent vi](image1)

Using the given step sizes, The CreateDimStep Vi (Figure 11.3) creates an array of delta parameter definitions, one for every dimension and direction. These define the neighborhood to test.

The first core point is the given StartPulse. It is tested and its figure of merit is calculated. The shift registers of the while loop always represents the core point. The starting core point is therefore their initial value. The for loop uses the neighborhood array to calculate the pulse parameters of each neighbor to test. The output of the for loop is the pulse with lowest figure of merit.

If the output of the for loop happens to be the core point, that core point is the local minimum and the while loop exits, else the pulse becomes the new core point in a new iteration of the while loop.

![Figure 11.2: Steepest descent Vi](image2)
Figure 11.3: CreateDimensionStep.vi. The total length of the pulse is fixed, why the secondary sink time is adjusted accordingly. The neighborhood created consists of 16 pulses.
12 THE FULLY AUTOMATED METHOD

It is important to remember that the results from the described alternating variables method only are valid for the laser performance test of the used wavelength. The performance criteria are to be met at all wavelengths. In the manual procedure, this is considered fulfilled when the pulse has passed the performance criteria at the three given wavelengths. There is no guarantee that a minimum at one wavelength is a minimum in the two others. Further, it should also pass the white light test developed in chapter 6.1.

Here, a straightforward scheme that is easily automated, is presented in Figure 12.1.
Figure 12.1: Flow chart of an automatizable method to trim and select the best pulse considering all performance criteria.

The search starts by setting the TLS to the middle wavelength (1545nm). A pre-defined Default set of starting pulses are fed into the Alternating variables method. Each start pulse will result in a local minimum defined by the merit function. (Duplicated minima can occur.) Pulses that match all performance criteria form the set Good middle 1, their performance data and pulse definition are stored.

The next step is to set the TLS to the low wavelength (1530nm). Now the Good middle 1 is fed as start-pulses into the Alternating variables method. The idea being that these starting points...
could be a better guess for this unit, than the default set. Not only is this set smaller, it will hopefully need less iterations to find local minima. The resulting good pulses form the set $\text{good low 1}$.

To make the final compromise, all pulses considered need to be evaluated for all wavelengths. Thus, the BF method is used to test the $\text{Good middle 1}$ set at low wavelength as well. The good pulses from this set then form the $\text{Good low 2}$ set.

The $\text{Good low 1}$ set needs to be tested at middle wavelength, thus the laser is reset to 1545 nm and the BF method is used to test them and form the $\text{Good middle 2}$ set.

The final step for laser performance is to test both $\text{Good middle 2}$ and $\text{Good low 2}$ at the long wavelength (1570 nm). The pulses that pass this final test have thus passed the test for all wavelengths and form the $\text{Final set}$.

An optical switch can be used to change from the TLS source to a white light source. The $\text{Final set}$ is then loaded into the white light performance method.

The white light quality figure is then weighted into the final figure of merit and thus the best pulse can be selected (analogue to the method described in 6.3).

### 12.1 Implementation

The BruteForcePulser program was not built with operator usability in mind. Rather, it was designed to test out the methodology. Thus, in future work it would be easier to use the components to form a new automatic program, with usability in mind, rather than to modify the BF program. The BF program, however, is well suited for verification of the proposed method.

For a detailed explanation of how to perform this operation on a WISTOM unit using the BruteForcePulser program, see appendix A.
12.2 Method verification, results

Three WISTOM units were successfully trimmed using the method described in chapter 12, of which one was a new production unit for a customer.

The 128 starting pulses from Table 12.1 were used, weights were set to 1, except for the white light test that got a weight of ¼ when passing the performance limit and 4 when not passing the performance limit (by the implementation in Figure 6.2). The averaging count was set to 2048.

The first alternating variables stage at middle wavelength was finished in about 6h. Since some minima did not meet the performance criteria, the next alternating variables stage used a little less time. In all, it is estimated that the automatic trimming with those settings will use less than 12h, which is considered feasible for production.

<table>
<thead>
<tr>
<th>Name</th>
<th>default_start.param</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
</tr>
<tr>
<td>Start</td>
<td>0.44</td>
</tr>
<tr>
<td>End</td>
<td>0.48</td>
</tr>
<tr>
<td>Stepping</td>
<td>0.04</td>
</tr>
<tr>
<td>Steps</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 12.1: A set of $2^7=128$ starting pulses for the alternating variables method
13 FUTURE WORK (OUTLOOK)

There are many ways to improve on both the chance of finding the optimal pulse (in contrast to just finding a pulse that meet the minimum performance criteria), and execution time. For example, modifying the WISTOM firmware to allow restarting of the average sampling could save up to 40% of the execution time\(^{39}\).

However, changing the hardware setup and adding parallelism is probably the fastest way to ramp up production.

By changing the hardware setup, it is possible to remove the Alternating variables step at low wavelength, as well as all Brute Force method steps. This will not only cut execution time in half, but probably also improve the chances of finding the optimal pulse.

To do this, the TLS is replaced by three fixed wavelength lasers that are multiplexed into the WISTOM optical input. Thus, there will be three peaks simultaneously at each given wavelength (long, middle and short). This change would require smaller modifications to the WISTOM OPM configuration and the performance calculation algorithms. The cost of the hardware setup would probably be less than using the TLS.

Using this setup, parallelism is straightforward; a number of units \(M\), could simultaneously be trimmed by adding \(M\) 2-way optical switches and and two \(M\)-way splitters. Each WISTOM unit would be connected to the output of one of the optical switches. The inputs would be the multiplexed lasers and the white light source, respectively. Both sources split in \(M\)-equal parts by the splitters.

13.1 Epilog

After the end of this project, a fully automatic program as proposed in 12.1 was built, and is now in production use. The production has not yet needed the larger ramp up.
REFERENCES

Printed information
J. Van Randeraat and R. E. Setterington “Piezoelectric ceramics”, ISBN 0 901232 75 0
S. Helmfrid “Kalibrering av WISTOM”, Internal Proximion doc no M-PROJ-NUMB
“WISTOM product sheet”, Proximion doc no 102032-B
“WISTOM Optical Layer Monitor Users Guide”, Proximion doc no 0002019 B
“Functional Description WISTOM SW”, rev D, Internal Proximion doc no 100262
“Production Instruction for Test Rig Pulse Shaping” rev pA1, Internal Proximion doc no 100358

Personal communication
Cecilia Lundevall, Manager production, Proximion Fiber Systems AB
Mikael Bergman, Manager mechanics R&D, Proximion Fiber Systems AB
Rickard Terfelt, WISTOM hardware designer, Proximion Fiber Systems AB
Mattias Holmkvist, WISTOM software designer, Proximion Fiber Systems AB
BruteForcePulser documentation

Purpose & Scope

This is an overview of the BruteForcePulser software. It is not intended to describe the source code or the details of automatic pulsing. Rather, it is aimed for engineers who want to use the BruteForcePulser to measure how different pulse parameters affect the optical response, or how automatic routines are affected by input parameters.

For an in depth study of automatic pulse trimming, see Xjobbsrapporten.

Approvals

If the document is controlled by Matrix, delete this chapter since the information is stored in Matrix.

<table>
<thead>
<tr>
<th>Date</th>
<th>Name / Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reviewed By</td>
<td></td>
</tr>
<tr>
<td>Approved By</td>
<td></td>
</tr>
</tbody>
</table>
1 INTRODUCTION

BruteForcePulser is a program for development of pulse trimming methodologies. It provides different methods for automatic pulse trimming (Brute Force and Steepest Descent), as well as means for different types of performance measurements (laser performance and white light performance).

It should be noted that the aim of this program is to prove the concept of finding a pulse that complies with the different performance requirements without a priori knowledge of the individual unit. As such, the program and the described usage methodologies are rather rigid without shortcuts or optimizations.

1.1 Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Made By</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA</td>
<td>2007-10-25</td>
<td>OFO</td>
<td>Initial version</td>
</tr>
<tr>
<td>PA2</td>
<td>2007-10-31</td>
<td>OFO</td>
<td>Second P version</td>
</tr>
</tbody>
</table>

1.2 Related Documents/References

<table>
<thead>
<tr>
<th>No.</th>
<th>Document No</th>
<th>Document Rev</th>
<th>Document Name</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23456</td>
<td></td>
<td>X-jobbsrapporten</td>
<td>OFO</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.3 Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym/Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>Brute Force (method)</td>
</tr>
<tr>
<td>SD</td>
<td>Steepest Descent (method)</td>
</tr>
<tr>
<td>TLS</td>
<td>Tunable Laser Source</td>
</tr>
<tr>
<td>Att</td>
<td>Attenuation</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>RPS</td>
<td>Resonance Pulse Sensor</td>
</tr>
</tbody>
</table>
2 TABLE OF CONTENTS

1 INTRODUCTION ............................................................................................................................................. 2
  1.1 REVISION HISTORY ................................................................................................................................. 2
  1.2 RELATED DOCUMENTS/REFERENCES .................................................................................................... 2
  1.3 ACRONYMS AND ABBREVIATIONS ....................................................................................................... 2

2 TABLE OF CONTENTS .................................................................................................................................. 3

3 FUNCTION DESCRIPTION ............................................................................................................................... 4
  3.1 AUTOMATIC SEARCH .............................................................................................................................. 4
    3.1.1 Brute Force ......................................................................................................................................... 4
    3.1.2 Steepest Descent ............................................................................................................................... 4
  3.2 WHITE SOURCE TEST .............................................................................................................................. 4
  3.3 TLS COMMUNICATION .............................................................................................................................. 5
    3.3.1 Purpose .............................................................................................................................................. 5
    3.3.2 Protocol definition ............................................................................................................................. 5

4 PARAMETER DEFINITIONS ............................................................................................................................ 6
  4.1 PULSE DEFINITION .................................................................................................................................. 6
  4.2 PERFORMANCE DEFINITION .................................................................................................................. 6
  4.3 QUALITY DEFINITION .............................................................................................................................. 6
  4.4 FIGURE OF MERIT .................................................................................................................................... 6

5 GRAPHICAL USER INTERFACE ..................................................................................................................... 7
  5.1 GENERAL LAYOUT ................................................................................................................................... 7
  5.2 ACOUSTIC GRAPH ................................................................................................................................... 7
  5.3 CONTROL TAB VIEW ................................................................................................................................ 8
    5.3.1 Connection ......................................................................................................................................... 8
    5.3.2 TLS config ......................................................................................................................................... 8
    5.3.3 Manual .............................................................................................................................................. 9
    5.3.4 Automatic .......................................................................................................................................... 10
    5.3.5 Create trimming space ..................................................................................................................... 11
    5.3.6 Limits & Weights .............................................................................................................................. 11
    5.3.7 White Source Test ........................................................................................................................... 12
  5.4 OPTICAL RESPONSE ............................................................................................................................... 12
  5.5 CONTROLS ................................................................................................................................................ 13
    5.5.1 Current ............................................................................................................................................. 13
    5.5.2 Best ................................................................................................................................................... 14
    5.5.3 Results ............................................................................................................................................... 14
    5.5.4 Steepest Descent Parameter Change ............................................................................................... 15
    5.5.5 White source test ............................................................................................................................. 15
  5.7 MESSAGES ................................................................................................................................................ 15

6 FILE FORMAT DESCRIPTION .......................................................................................................................... 16
  6.1 .INFO ....................................................................................................................................................... 16
  6.2 .PARAM ................................................................................................................................................... 16
  6.3 .PERFORMANCE .................................................................................................................................... 16
  6.4 .QUALITY ............................................................................................................................................... 17
  6.5 .WHITE ................................................................................................................................................... 17

7 USAGE EXAMPLES AND FLOWCHARTS .......................................................................................................... 17
  7.1 TRIMMING LASER LIGHT PERFORMANCE ............................................................................................ 17
    7.1.1 Procedure flowchart ......................................................................................................................... 18
  7.2 TRIMMING WHITE LIGHT PERFORMANCE ............................................................................................ 19
    7.2.1 Procedure flowchart ......................................................................................................................... 19
3 FUNCTION DESCRIPTION

3.1 Automatic search

The automatic function aims to test a grand set of pulses using different methods. Two methods have been developed, Brute Force and Steepest Descent.

In both these cases, WISTOM is connected to a TLS with a very thin bandwidth.

Each test consists of programming the pulse form in WISTOM and reading the resulting performance quantities. The definition of the pulse form is described in chapter 4.1 and the performance quantities in chapter 4.2.

3.1.1 Brute Force

This method simply tests each given pulse and saves the result to file. This method is designed for huge numbers of pulses, why only the single best pulse form and its result is kept in memory. However, an added feature lets you keep good pulses (i.e. performance within specification, see chapter 5.3.6 Limits & Weights) in result memory.

The result of the current testing pulse is shown in the Performance Log->Current tab.

Files created are .info, .performance, and .param (see chapter 6 File Format Description). As each result is ready, it is written to the file. The filename used is the given OutputName. However, as not to overwrite existing files, a number is automatically added to the filename if it already exists.

3.1.2 Steepest Descent

In this method the given pulses are used as start pulses. Each pulse form parameter is increased and decreased, one at a time. If the change of pulse gave better performance (see chapter 4.4 Figure of merit), the pulse with the best performance increase is used for another iteration of increase/decrease. When performance no longer can be improved by a single step, the final performance and pulse form is saved in result memory and written to file.

Files created are .info, .performance, .param and .sparam. As each result is ready, it is written to the file. The filename used is the given output name. However, as not to overwrite existing files a number is automatically added to the filename if it already exists.

3.2 White source test

White source test is used to evaluate the ripple in amplitude. If the ripple is to steep it cannot be calibrated to a straight line with high enough precision. This test is used to calculate the maximum ripple among a set of pulses.

In this trimming method, a white light source is connected to WISTOM. The ripple is calculated as the maximum difference in amplitude within a window (normally 14 samples). The window is moved to cover the complete wavelength span.

Even though the white light source does not have a linear spectrum, the spectrum is smooth enough as not to affect the ripple within the window.

The mean ripple, as well as the maximum and its window start location, is given as result.
When the White Source Test tab is active, the optical spectrum is shown with the wavelength on the x-axis\textsuperscript{1}. The spectrum is limited by the WL left and WL right controls.

The test can be performed on a previously recorded snapshot, or on spectrums generated from a set of pulses defined in a file. When testing such a set of pulses, the result of the previous iteration is shown while testing the next pulse. When all pulses are tested, the result can be saved to file by clicking the Save to file button in the White Source Performance tab. A white file will be created using the filename of the pulse definitions.

### 3.3 TLS communication

#### 3.3.1 Purpose

The program supports a simple custom TLS communication protocol over TCP/IP. The benefits are that one does not have to run the program on a computer with GPIB or other means of TLS communication. Instead, the program can be run on any desktop computer, while a simple server program is run a computer connected to the TLS.

#### 3.3.2 Protocol definition

The connection is non persistent, i.e. the client connects, sends a command and waits for a response and then closes the connection.

There are two commands; SET and GET.

- **3.3.2.1 Get command**
  - ASCII characters GET (3 bytes)

- **3.3.2.2 Get response**
  - 4 byte int of text length
  - Clear text response containing the current wavelength

- **3.3.2.3 Set command**
  - ASCII character SET (3 bytes)
  - Binary double (8 bytes)

- **3.3.2.4 Set response**
  - Binary 32 bit integer (4 bytes) of following text length
  - Clear text response containing set wavelength or error message

---

\textsuperscript{1} This is an inappropriate method since the wavelength is not calibrated at this production stage. See Xjobbsrapport XX.XX for further details.
4 PARAMETER DEFINITIONS

4.1 Pulse definition

Figure 1 Example of pulse definition and corresponding generated pulse.

The pulse definition consists of two equally defined pulse lobes and an amplification value.

Each lobe is defined by 6 values; start position, climb time, amplitude to climb to, top width and time to sink back to zero. (Note that width, position and time are actually the same thing, i.e. sample counts.)

The secondary pulse form is added directly behind the main form. The amplification factor is called \( PABC \).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Better criteria</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude</td>
<td>Peak amplitude</td>
<td>Higher is better</td>
<td>dB</td>
</tr>
<tr>
<td>FWHM</td>
<td>The width of the peak measured at half of the maximum amplitude.</td>
<td>Lower is better</td>
<td>GHz</td>
</tr>
<tr>
<td>Attenuation</td>
<td>Amplitude at a distance from peak divided by the peak amplitude.</td>
<td>Lower is better</td>
<td>dB</td>
</tr>
<tr>
<td>SNR</td>
<td>SNR is defined as the largest peak amplitude minus the second largest peak amplitude, measured in dBm. If no second peak is detected, the second peak is set to -60dBm.</td>
<td>Higher is better</td>
<td>dB</td>
</tr>
<tr>
<td>Symmetry</td>
<td>( Symmetry = 100 \left( \frac{Att_{left}}{Att_{right}} - 1 \right) )</td>
<td>Lower is better</td>
<td>%</td>
</tr>
</tbody>
</table>

4.2 Performance definition

4.3 Quality definition

Here, quality is defined as a weighted distance to the specification limit, hence quality values are usually referred to as distances in the GUI. Thus, the lower the quality value, the better. If the quality value is below zero, the corresponding performance is better than the specification limit.

For values where lower is better

\[ q_i = w_j \frac{p_i - l_i}{l_i} \]

For values where higher is better

\[ q_i = w_j \frac{-p_i + l_i}{l_i} \]

See chapter 5.3.5.4 for setting weights and limits.
Observe that with this definition, when a value is below limit, the weight has the opposite effect as then above limit.

### 4.4 Figure of merit

As the single figure describing the combined quality of all performance parameters the maximum (i.e. the weakest) value is used.

\[
Q = \min \left( w_i \cdot \frac{p_i + l_i}{l_i} \right)
\]

### 5 GRAPHICAL USER INTERFACE

#### 5.1 General layout

Left side contains parameter settings and functions, while the right side displays results.

In the upper left is the **Control Tab**, below is the **Acoustic graph**, displaying the pulse form and the RPS sensor signal. In the upper right is the **Optical Graph**. Below is Performance Logs Tab, displaying performance and figures of merit.

#### 5.2 Acoustic graph

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>The current (in WISTOM) active pulse form</td>
</tr>
<tr>
<td>Red</td>
<td>RPS signal (shifted to match the zero-crossing of the pulse form)</td>
</tr>
<tr>
<td>Green</td>
<td>Manual set form, or the starting form for trimming space</td>
</tr>
<tr>
<td>Blue</td>
<td>End form of the trimming space</td>
</tr>
</tbody>
</table>
5.3 Control tab view

5.3.1 Connection

5.3.1.1 Main purpose

To connect to the unit to be trimmed and set up basic parameters in that unit.

5.3.1.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>The ip address of the unit to connect to</td>
</tr>
<tr>
<td>Connec/Disconnect button</td>
<td>Connects or disconnects the unit. On successful connection the button changes to ‘Disconnect’.</td>
</tr>
<tr>
<td>Averaging slider</td>
<td>This slider is used to change the number of averaging sweeps WISTOM will use for spectrum analysis. On connect it is set to the active value in WISTOM.</td>
</tr>
<tr>
<td>Development time</td>
<td>The number of milliseconds the program will wait from change of pulse until the pulse is considered as stable.</td>
</tr>
<tr>
<td>Setup OPM</td>
<td>Sets the OPM settings needed for automatic trimming.</td>
</tr>
</tbody>
</table>

5.3.1.3 Indicators

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan time</td>
<td>Estimated time for a complete averaged sweep</td>
</tr>
<tr>
<td>WaitTime</td>
<td>Time to wait from change of pulse until reading OPM and spectrum. Value calculated as [ \text{WaitTime} = \text{DevelopmentTime} + 2 \cdot \text{ScanTime} ]</td>
</tr>
<tr>
<td>Compression</td>
<td>Spectrum compression</td>
</tr>
<tr>
<td>SubScans</td>
<td>Number of subscans</td>
</tr>
</tbody>
</table>

5.3.2 TLS config
5.3.2.1 Purpose
Read or change TLS wavelength.

5.3.2.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS Server IP address</td>
<td>The IP-address of the GPIB equipped computer running the TLS server</td>
</tr>
<tr>
<td>Get WaveLength button</td>
<td>Reads the current wavelength from the TLS</td>
</tr>
<tr>
<td>Set WaveLength button</td>
<td>Sets the WL indicated by the WL slider</td>
</tr>
<tr>
<td>WaveLength slider</td>
<td>Indicate what WL the TLS will be set to</td>
</tr>
</tbody>
</table>

5.3.2.3 Indicators:

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS response field</td>
<td>The response from the TLS server when Get/Set wavelength buttons are clicked. This text will also be saved to the <code>.info</code> file when running automatic pulse trimming.</td>
</tr>
</tbody>
</table>

5.3.3 Manual

5.3.3.1 Purpose
Used to manually set a pulse in WISTOM.

5.3.3.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual pulse</td>
<td>The pulesdefinition to set, se ‘Pulse definiton’ chapter.</td>
</tr>
<tr>
<td>Set in WISTOM</td>
<td>Sets the given pulse in WISTOM</td>
</tr>
</tbody>
</table>

5.3.3.3 Indicators

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PABC field</td>
<td>While not a pure indicator, upon connection of the unit this value is set to the value in WISTOM</td>
</tr>
<tr>
<td>Main coordinates</td>
<td>Indicates each vertex of the main pulse loop</td>
</tr>
<tr>
<td>Secondary coordinates</td>
<td>Indicates each vertex of the secondary pulse loop</td>
</tr>
<tr>
<td>Acoustic Graph</td>
<td>The pulse form is shown graphically as the green line in the Acoustic graph (below this tab).</td>
</tr>
</tbody>
</table>
5.3.4 Automatic

5.3.4.1 Purpose
Controlling the automatic pulse trimming procedures

5.3.4.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse definition source</td>
<td>Select what source will be used for pulse definitions.</td>
</tr>
<tr>
<td>Created</td>
<td>Definitions by tab Create pulse definition</td>
</tr>
<tr>
<td>From file</td>
<td>Definition from file</td>
</tr>
<tr>
<td>Results</td>
<td>Results from SD, or good pulses from BF if these are saved in result memory</td>
</tr>
<tr>
<td>Manual</td>
<td>The single pulse defined by the tab Manual</td>
</tr>
<tr>
<td>Index of starting pulse</td>
<td>This number of pulses will be excluded from testing</td>
</tr>
<tr>
<td>Pulse definition file</td>
<td>The .pulse definition file that will be used when From file is selected as pulse source.</td>
</tr>
<tr>
<td>Save results to file</td>
<td>Save pulses from a SD (or BF if in memory) to file</td>
</tr>
<tr>
<td>Keep only good pulses in memory</td>
<td>Will only keep pulse forms that are within performance within specification in result-memory</td>
</tr>
<tr>
<td>Step</td>
<td>Select the parameter-step that will be used in SD method</td>
</tr>
<tr>
<td>Output filename</td>
<td>The base filename (without extension) that will be used to continuously save the results from a BF or SD run.</td>
</tr>
<tr>
<td>Method</td>
<td>Select Steepest Descent (SD) or Brute Force (BF)</td>
</tr>
<tr>
<td>Start</td>
<td>Starts the selected method, the process is finished when the green led turns dark.</td>
</tr>
<tr>
<td>Read spectrum</td>
<td>Shows the spectrum of the current pulse (BF) or the iteration final pulse (SD) in optical response graph. Will slow down operation, why you are recommended to turn this of when not monitoring the graph.</td>
</tr>
<tr>
<td>Pause</td>
<td>Will pause before starting on next pulse</td>
</tr>
<tr>
<td>STOP</td>
<td>Will stop SD after current iteration</td>
</tr>
</tbody>
</table>

5.3.4.3 Indicators

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.3.5 Create trimming space

5.3.5.1 Purpose
Create evenly spaced pulse definitions.

5.3.5.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>The smallest value</td>
</tr>
<tr>
<td>End</td>
<td>The greatest value</td>
</tr>
<tr>
<td>Step length</td>
<td>How much to increase each parameter until End is reached</td>
</tr>
</tbody>
</table>

5.3.5.3 Indicators

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of pulse</td>
<td>The total length of the pulses is held constant to this number by changing the sink time.</td>
</tr>
<tr>
<td>Combinations</td>
<td>Number of combinations created by the current definition.</td>
</tr>
<tr>
<td>Estimated time</td>
<td>The estimated time needed to test the combination with the selected method (SD or BF)</td>
</tr>
</tbody>
</table>

5.3.5.4 Note
When step length or start value is changed, the end values are coerced.

5.3.6 Limits & Weights

5.3.6.1 Purpose
Used to define the limits for what is considered as within specification, and to weight the distance when limits are outside specification.
5.3.6.2 Purpose
Set the limits and weights used to evaluate a figure of merit.

5.3.7 White Source Test

5.3.7.1 Purpose
Test amplitude difference

5.3.7.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL high</td>
<td>Check amplitude difference below this wavelength</td>
</tr>
<tr>
<td>WL low</td>
<td>Check amplitude difference above this wavelength</td>
</tr>
<tr>
<td>Window length</td>
<td>The amplitude difference within this number of samples is measured</td>
</tr>
<tr>
<td>Evaluate snapshot</td>
<td>Test amplitude difference in the snapshot (from optical tab)</td>
</tr>
<tr>
<td>Evaluate pulsedefs from file</td>
<td>Test amplitude difference for each of the pulses in the pulse definition file indicated in the Settings-&gt;Automatic tab</td>
</tr>
<tr>
<td>Save to file</td>
<td>Save the performance results to tab separated file</td>
</tr>
</tbody>
</table>

5.4 Optical response
Displays the optical spectrum.
The white graph is the current optical spectrum, while the red is the snapshot view.

However, in **White Source Test** mode, the red line shows the difference in amplitude (dB) within each window, and the horizontal (x) axis is inverted since it shows wavelength instead of time.

### 5.5 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dBm</td>
<td>Displays dBm when green led is lit, or watts when dark, on the vertical axis</td>
</tr>
<tr>
<td>Go to peak</td>
<td>Zooms in on the peak with highest amplitude</td>
</tr>
<tr>
<td>AutoScale X</td>
<td>Fits the whole horizontal axis within the graph window</td>
</tr>
<tr>
<td>AutoScale Y</td>
<td>Fits the whole vertical axis within the graph window</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Creates a snapshot of the next read spectrum for easy comparison. The snapshot is shown in red.</td>
</tr>
</tbody>
</table>

### 5.6 Performance Log tab

#### 5.6.1 Current

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Pulse</td>
<td>The definition of the current pulse</td>
</tr>
<tr>
<td>Performance</td>
<td>Current performance</td>
</tr>
</tbody>
</table>

5.6.1.1 **Purpose**

Shows current performance and distance to specification. In automatic mode the current pulse definition is also shown.

5.6.1.2 **Indicators**

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Pulse</td>
<td>The definition of the current pulse</td>
</tr>
<tr>
<td>Performance</td>
<td>Current performance</td>
</tr>
</tbody>
</table>
5.6.2 Best

When running the Brute Force method, the so far best pulse definition and performance is shown here.

5.6.3 Results

When an SD run has been done, the resulting pulses and their performances are shown here.

If Keep good pulses in memory is selected when doing BF, the good (within specification) pulses are shown here.

5.6.3.2 Indicators

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulses</td>
<td>Pulse definition of selected pulse</td>
</tr>
<tr>
<td>Performances</td>
<td>The performance values of selected pulse</td>
</tr>
<tr>
<td>Iteration Counts</td>
<td>Number of steps SD used to arrive to selected pulse</td>
</tr>
<tr>
<td>Qualities</td>
<td>The single figure of merit of selected pulse</td>
</tr>
</tbody>
</table>
5.6.3.3 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select result nr</td>
<td>Select the pulse to be shown</td>
</tr>
<tr>
<td>Set in Wistom button</td>
<td>Sets the selected pulse definition in WISTOM</td>
</tr>
</tbody>
</table>

5.6.4 Steepest Descent Parameter Change

A visual view of parameter changes of each iteration for the current start pulse.

5.6.5 White source test

5.6.5.1 Indicators

Performance values

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Mean amplitude difference in spectrum</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum amplitude difference in spectrum</td>
</tr>
<tr>
<td>Max WL</td>
<td>The wavelength of the maximum difference</td>
</tr>
</tbody>
</table>

5.6.5.2 Controls

<table>
<thead>
<tr>
<th>GUI identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current WhitePerformance</td>
<td>Performance of current testing pulse, or the snapshot</td>
</tr>
<tr>
<td>White Performance</td>
<td>The performance of the tested pulses</td>
</tr>
</tbody>
</table>

5.7 Messages

Displays error-messages from automatic runs.
6  FILE FORMAT DESCRIPTION

6.1 .info

Provides information of settings during an automated trimming. File contains one value per line, with variable name preceding it. An equal sign (=) is used as delimiter between variable name and value.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Typical value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProgramNameVersion</td>
<td>BruteForcePulser P1B1</td>
<td>Name and version of Brute Force Pulser</td>
</tr>
<tr>
<td>IPadress</td>
<td>192.168.8.3</td>
<td>IP of WISTOM unit</td>
</tr>
<tr>
<td>UnitSerial</td>
<td>A04-00340</td>
<td>Serial of WISTOM unit</td>
</tr>
<tr>
<td>compression</td>
<td>1</td>
<td>Active spectrum compression</td>
</tr>
<tr>
<td>subScans</td>
<td>8</td>
<td>Number of subsens per sweep</td>
</tr>
<tr>
<td>Averageing</td>
<td>2048</td>
<td>Number of sweeps per spectrum</td>
</tr>
<tr>
<td>Development Time (ms)</td>
<td>750</td>
<td>Time before pulse is considered as stable</td>
</tr>
<tr>
<td>WaitTime (ms)</td>
<td>3370</td>
<td>Total time to wait from programming of new pulse form to spectrum readout</td>
</tr>
<tr>
<td>TLS response</td>
<td>Set to 1545</td>
<td>Response from TLS server</td>
</tr>
<tr>
<td>Step.PABCstep</td>
<td>0.01</td>
<td>PABC step length in steepest descent</td>
</tr>
<tr>
<td>Step.AmplitudStep</td>
<td>5</td>
<td>Amplitude step length in steepest descent</td>
</tr>
<tr>
<td>Step.TimeStep</td>
<td>1</td>
<td>Time step length in steepest descent</td>
</tr>
<tr>
<td>Method</td>
<td>Steepest Descent</td>
<td>Method used</td>
</tr>
<tr>
<td>Pulse definition source</td>
<td>From File</td>
<td>Source of pulse definitions</td>
</tr>
</tbody>
</table>

6.2 .param

Tab separated file of pulse definitions.

<table>
<thead>
<tr>
<th>Field nr</th>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Main lobe: Start position</td>
<td>Sample count</td>
</tr>
<tr>
<td>1</td>
<td>Main lobe: Climb time</td>
<td>Sample count</td>
</tr>
<tr>
<td>2</td>
<td>Main lobe: Top Width</td>
<td>Sample count</td>
</tr>
<tr>
<td>3</td>
<td>Main lobe: Amplitude</td>
<td>Sample value</td>
</tr>
<tr>
<td>4</td>
<td>Main lobe: Sink time</td>
<td>Sample count</td>
</tr>
<tr>
<td>5</td>
<td>Secondary lobe: Start position</td>
<td>Sample count</td>
</tr>
<tr>
<td>6</td>
<td>Secondary lobe: Climb time</td>
<td>Sample count</td>
</tr>
<tr>
<td>7</td>
<td>Secondary lobe: Top Width</td>
<td>Sample count</td>
</tr>
<tr>
<td>8</td>
<td>Secondary lobe: Amplitude</td>
<td>Sample value</td>
</tr>
<tr>
<td>9</td>
<td>Secondary lobe: Sink time</td>
<td>Sample count</td>
</tr>
<tr>
<td>10</td>
<td>PABC</td>
<td></td>
</tr>
</tbody>
</table>

6.3 .performance

Tab separated file of performance figures

<table>
<thead>
<tr>
<th>Field nr</th>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FWHM</td>
<td>GHz</td>
</tr>
<tr>
<td>1</td>
<td>Attenuation @ -12.5 GHz</td>
<td>dB</td>
</tr>
<tr>
<td>2</td>
<td>Attenuation @ -25.0 GHz</td>
<td>dB</td>
</tr>
<tr>
<td>3</td>
<td>Attenuation @ +12.5 GHz</td>
<td>dB</td>
</tr>
<tr>
<td>4</td>
<td>Attenuation @ +25.0 GHz</td>
<td>dB</td>
</tr>
<tr>
<td>5</td>
<td>SNR</td>
<td>dB</td>
</tr>
</tbody>
</table>
6.4 \textbf{.quality}

Same as \textbf{.performance}, but with quality values, i.e. weighted distance from specification limits. Observe that these values depends on the ‘limits & weights’ values, but that these are not saved.

6.5 \textbf{.white}

Tab separated file of performance from the white source test

<table>
<thead>
<tr>
<th>Field nr</th>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Mean</td>
<td>dB</td>
</tr>
<tr>
<td>1</td>
<td>Maximum amplitude difference</td>
<td>dB</td>
</tr>
<tr>
<td>2</td>
<td>Wavelength of the maximum difference</td>
<td>nm</td>
</tr>
</tbody>
</table>

7 USAGE EXAMPLES AND FLOWCHARTS

Please observe that these examples are procedures used to prove the concept of automatic pulse trimming, they are not intended as production procedures.

7.1 Trimming Laser Light Performance

7.1.1.1 \textbf{Connect and setup}

- Make sure the unit has the correct Calman state (pulse trim).
- Connect the TLS to WISTOM optical input.
- Enter IP and click Connect
- Verify the unit serial number.
- Set the averaging to 2048 and the development time to 750 ms
- Go to Limits & Weights and set the values to your specification
- Go to the TLS config tab page and set the wavelength to 1545nm
- Verify that the TLS response shows the correct wavelength

7.1.1.2 \textbf{Initial search at 1545}

- Go to the Automatic tab.
- Select the radio button \textit{From file}
- Select \textit{Keep only good pulses in memory}
- Select the \textit{Steepest Descent method}
- Choose Output Name and create a directory and filename named after the unit serial number
- Click \textit{Start}

This step may take several hours. The estimated time will be more precise as a couple of start pulses are finished.

- Click \textit{Save result to file}, use filename \textit{middle_good.param}

7.1.1.3 \textbf{Search at 1530 nm}

- Go to the TLS config tab and set the wavelength to 1530.
- Go back to the Automatic tab
- Select \textit{From File} and pulse definition file \textit{middle_good.param}
- Select \textit{Steepest Descent method} and click \textit{Start}

This step may take quite some time, depending on the number of start pulses.
7.1.1.4 Select good at 1545
- Go to the TLS config tab and set the wavelength to 1545.
- Go back to the Automatic tab
- Select From File and pulse definition file low_good01.param
- Select Brute Force method and click Start
- Click Save result to file, use filename low_good02.param.

7.1.1.5 Combine definition files
- Concatenate middle_good02.param and low_good02.param into combined.param

7.1.1.6 Select good at 1565 nm
- Set TLS to 1565 nm
- Select From File and pulse definition file combined.param
- Select Brute Force method and click Start
- Save the result as UnitNameFinalGood.param
- Go to the Connect tab and click Disconnect.

7.1.2 Procedure flowchart
7.2 Trimming White Light Performance

7.2.1.1 Connect and setup
- Connect a white light source to WISTOM.
- Enter IP and click Connect.
- Verify the unit serial number.
- Set the averaging to 4096 and the development time to 750 ms

7.2.1.2 Perform white source test
- Go to the White source tab
- Select the input file UnitNameFinalGood.param and click Evaluate pulse definitions from file
- Manually look thought the performance results in the White Source Performance tab, select the best pulse (lowest max value). Record the index and the performance value in your report file.
- Find that line (indexed from zero) in the FinalGood.param file, and record the parameters.
- Also find the performance from TLS tests of that particular pulse and save that in the report too.
- Set the pulse form in WISTOM using the ‘manual’ tab, write to the flash memroy with some telnet-command.

7.2.2 Procedure flowchart
Appendix B

### Standard form pulses in production units.

<table>
<thead>
<tr>
<th>Id #</th>
<th>PABC</th>
<th>Climb</th>
<th>Top</th>
<th>Sink</th>
<th>Start</th>
<th>Climb</th>
<th>Sink</th>
<th>Amp</th>
</tr>
</thead>
<tbody>
<tr>
<td>W308</td>
<td>0,45</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>54</td>
<td>80</td>
</tr>
<tr>
<td>W309</td>
<td>0,38</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>43</td>
<td>90</td>
</tr>
<tr>
<td>W310</td>
<td>0,4</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>47</td>
<td>80</td>
</tr>
<tr>
<td>W312</td>
<td>0,42</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>44</td>
<td>76</td>
</tr>
<tr>
<td>W318</td>
<td>0,48</td>
<td>10</td>
<td>4</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>42</td>
<td>90</td>
</tr>
<tr>
<td>W319</td>
<td>0,49</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>45</td>
<td>100</td>
</tr>
<tr>
<td>W320</td>
<td>0,47</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>40</td>
<td>116</td>
</tr>
<tr>
<td>W322</td>
<td>0,46</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>41</td>
<td>100</td>
</tr>
<tr>
<td>W323</td>
<td>0,42</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>40</td>
<td>99</td>
</tr>
<tr>
<td>W324</td>
<td>0,48</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>W325</td>
<td>0,46</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>9</td>
<td>10</td>
<td>39</td>
<td>100</td>
</tr>
<tr>
<td>W326</td>
<td>0,45</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>W327</td>
<td>0,52</td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>11</td>
<td>50</td>
<td>94</td>
</tr>
<tr>
<td>W332</td>
<td>0,42</td>
<td>11</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>57</td>
<td>106</td>
</tr>
<tr>
<td>W339</td>
<td>0,46</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>66</td>
<td>116</td>
</tr>
<tr>
<td>W341</td>
<td>0,43</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>62</td>
<td>74</td>
</tr>
<tr>
<td>W350</td>
<td>0,48</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>63</td>
<td>107</td>
</tr>
<tr>
<td>W352</td>
<td>0,46</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>11</td>
<td>64</td>
<td>102</td>
</tr>
<tr>
<td>W353</td>
<td>0,45</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>45</td>
<td>99</td>
</tr>
<tr>
<td>W354</td>
<td>0,42</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>60</td>
<td>75</td>
</tr>
<tr>
<td>W355</td>
<td>0,43</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>W356</td>
<td>0,46</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>5</td>
<td>7</td>
<td>64</td>
<td>93</td>
</tr>
<tr>
<td>W357</td>
<td>0,46</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>W358</td>
<td>0,46</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>60</td>
<td>97</td>
</tr>
<tr>
<td>W359</td>
<td>0,45</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>65</td>
<td>84</td>
</tr>
<tr>
<td>W360</td>
<td>0,5</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>66</td>
<td>100</td>
</tr>
<tr>
<td>W361</td>
<td>0,46</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>67</td>
<td>103</td>
</tr>
<tr>
<td>W362</td>
<td>0,45</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>65</td>
<td>100</td>
</tr>
<tr>
<td>W363</td>
<td>0,48</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>62</td>
<td>100</td>
</tr>
<tr>
<td>W364</td>
<td>0,43</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>64</td>
<td>86</td>
</tr>
<tr>
<td>W365</td>
<td>0,44</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>70</td>
<td>79</td>
</tr>
</tbody>
</table>

**Outliers:**

<table>
<thead>
<tr>
<th>Id #</th>
<th>PABC</th>
<th>Climb</th>
<th>Top</th>
<th>Sink</th>
<th>Start</th>
<th>Climb</th>
<th>Sink</th>
<th>Amp</th>
</tr>
</thead>
<tbody>
<tr>
<td>W301</td>
<td>0,42</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>42</td>
<td>93</td>
</tr>
<tr>
<td>W311</td>
<td>0,48</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>11</td>
<td>48</td>
<td>83</td>
</tr>
<tr>
<td>W314</td>
<td>0,55</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>9</td>
<td>42</td>
<td>87</td>
</tr>
<tr>
<td>W315</td>
<td>0,54</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>W317</td>
<td>0,5</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>5</td>
<td>45</td>
<td>80</td>
</tr>
<tr>
<td>W321</td>
<td>0,46</td>
<td>10</td>
<td>5</td>
<td>12</td>
<td>8</td>
<td>11</td>
<td>44</td>
<td>76</td>
</tr>
<tr>
<td>W323</td>
<td>0,39</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>17</td>
<td>6</td>
<td>65</td>
<td>91</td>
</tr>
<tr>
<td>W331</td>
<td>0,46</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>13</td>
<td>63</td>
<td>103</td>
</tr>
<tr>
<td>W335</td>
<td>0,5</td>
<td>14</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>9</td>
<td>59</td>
<td>115</td>
</tr>
<tr>
<td>W336</td>
<td>0,48</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>15</td>
<td>35</td>
<td>87</td>
</tr>
<tr>
<td>W337</td>
<td>0,45</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>15</td>
<td>35</td>
<td>87</td>
</tr>
</tbody>
</table>

**Max** 0,53 11 8 9 11 70 116
**Min** 0,38 8 4 6 3 4 39 74
**Avrg** 0,452581 9,1612903 5,9 7,97 7,097 7,645 54,35 95,03
**Diff** 0,15 3 4 3 6 7 31 42
**Std-dev / avrg** 0,066768 0,0850044 0,18 0,08 0,213 0,301 0,195 0,117
**Comb** 16 4 5 7 8 1 43
**Time per pulse (s)** 0,1 1 2 3 7 4
**Total time (days)** 4 36 71 132 143

*Energy defined as the integral of the pulseform*