

# Controlling the stabilization of a laser- system

A hybrid model of a laser-system and  
its components



**Caution: Laser in Use**

## Master thesis

Michiel Alsters  
michiela@sci.kun.nl  
Nijmegen, The Netherlands, 2001

Thesis number: 487  
University of Nijmegen  
Informatics for Technical Applications

Supervisor: Prof. dr. F. Vaandrager  
Co-supervisor: Prof. dr. ir. J. Vytupil



## Preface

My all-time dream is to work in space. One step in fulfilling this dream is to learn more on how physics and computer science can work together to reach remarkable achievements. With this master thesis I have, in my point of view, created a better understanding for me on the combination of the two worlds.

During the project I have experienced the different ways of thinking of physicists and computer scientists and what is important for their research. I have enjoyed living on the boundary of the two worlds and I think there is still a lot of work to do to create a solid bridge between them.

I had a very tight schedule for finishing the project and write this thesis. In the end I almost made it, but it was just too much. When I look back on the last six months I must conclude this was my own fault. I am a very busy individual and especially my job at the university to create the course schedules for the new school year, had a negative influence on the results of the project. Nevertheless, I have no regrets about this job and I think that altogether the gained experiences had a positive result on me.

I would like to thank Frits Vaandrager as my supervisor and coach. He gave me the inspiration to look beyond the boundaries of my own knowledge (as a good supervisor should). He was critical on my work, but together we made this thesis for what it is. I also would like to thank Jan Vytopil for his patience and his critical eye concerning the realisation of the model. In spite of all the collisions we had, I am very glad you could be my supervisors.

I would like to thank Frans Harren (on behalf of the Molecular Laser Physics department) for giving me the opportunity to work on his department. He guided me into the world of laser physics and with every step we take my interest in the subject grew. During the project my closest companion was Maarten van Herpen. My special thanks goes to him for his time to answer all my questions and making his work a part of mine.

Where would I be without the help of Sabine Houwer and Matthijs who spend many hours in reading my thesis and make the (very) necessary corrections in my use of the English language? Without their help this thesis would probably be unreadable.

A special thanks goes to my two employers Rob Burow and Michiel Geurts of MASC Software. For more than three years now we have shared more than our lives at the office and they supported me in every way they could in finishing my study. Now the time is come to leave the company, which was a very hard decision to make, but the friendship stays forever.

Last, but not least, I would like to thank my parents and my sister for all the years of caring. Without their mental and financial support I would not have come this far.

Michiel Alsters  
September 20, 2001



# Contents

<b>Part 1: Introduction</b> .....	<b>9</b>
Chapter 1 An overview .....	11
Chapter 2 Research questions .....	13
<b>Part 2: The static model</b> .....	<b>15</b>
Chapter 3 The laser .....	17
3.1 Lasers, the basics .....	17
3.2 The actuators .....	19
3.3 Cavity modes .....	20
3.4 The use of the etalon .....	22
3.5 The automaton model .....	24
Chapter 4 The physical layout .....	27
<b>Part 3: The dynamic model</b> .....	<b>29</b>
Chapter 5 The Lock-in .....	31
5.1 Mathematical background .....	31
5.2 The error signal .....	32
5.3 The automaton model .....	33
Chapter 6 The mode-lock .....	35
6.1 The use of the mode-lock .....	35
6.2 The feedback loop .....	35
6.3 The automaton model .....	37
6.4 A correctness assertion .....	39
Chapter 7 De scans .....	41
7.1 Grating scan .....	41
7.2 Mode-hop scan .....	41
7.3 Mode-hop-free scan .....	42
7.4 Pump scan .....	42
<b>Part 4: The implementation</b> .....	<b>45</b>
Chapter 8 LabVIEW .....	47
8.1 Graphical programming .....	47
8.2 The use of vi's and sub-vi's .....	48
Chapter 9 The program .....	51
9.1 The user-interface .....	51
9.2 The mode-lock .....	54
9.3 The scans .....	56
Chapter 10 Conclusions .....	59
<b>Part 5: Appendix</b> .....	<b>61</b>
Appendix A1 Document of demands .....	63
Appendix A2 System documentation .....	67
Appendix A3 Bibliography .....	83



## Table of figures

Figure 3.1:	The interior of the FC laser.	18
Figure 3.2:	The interior of the OPO laser.	19
Figure 3.3:	A schematic reproduction of a laser cavity.	20
Figure 3.4:	The three different transmission peaks and the loss line of the laser.	22
Figure 3.5:	The transmission line of the laser.	23
Figure 3.6:	An automaton of a laser.	26
Figure 4.1:	An overview of the complete system.	27
Figure 4.2:	The use of the cavity end-mirror piezo for the spectrum analysis.	28
Figure 5.1:	Etalon transmission peak with three cavity modes.	32
Figure 5.2:	An automaton of the lock-in.	33
Figure 6.1:	The relationship between the error signal and the adjustment on the piezo voltage	35
Figure 6.2:	An automaton of the controller.	37
Figure 6.3:	The complete mode-lock automaton.	38
Figure 7.1:	Performing a mode-hop scan.	42
Figure 7.2:	The frequency according to the changes in the voltage of the pump laser.	42
Figure 7.3:	A combination of the pump scan and the mode-hop scan.	43
Figure 8.1:	An example of a front panel.	48
Figure 8.2:	An example of a block diagram.	49
Figure 9.1:	Start-up screen of the program.	51
Figure 9.2:	Calibrations and set-up screen of the program.	52
Figure 9.3:	General scan interface. This page is used for all kinds of scanning.	53
Figure 9.4:	Test screen with controls to the piezo's and stepper-motor.	53
Figure 9.5:	The initialisation of the mode-lock.	54
Figure 9.6:	The core code of the mode-lock.	55
Figure 9.7:	The calculation of the voltage difference and the least square fit.	55
Figure 9.8:	The finalization of the mode-lock process.	56
Figure 9.9:	The code for acquiring data from the sensors.	56
Figure 9.10:	Acquiring the data from the input channels and apply the calibrations on them.	57
Figure 9.11:	The visualization code.	58



## *Part 1: Introduction*



# Chapter 1      An overview

Before starting to explain the composition of this master thesis, a few words are used to clarify the reader on the reason why I chose this particular subject. I will tell something about my history regarding the subject of the thesis. It is followed by a summary of my thesis.

In the autumn of the year 2000 I participated in the course "Optics and Lasers" at the Molecular Laser Physics department. Due to this course my interest in laser control grew and I became curious about the computer component in a laser-system. The lecturing professor at that time was dr. Frans Harren, who introduced me to the basics of laser physics. Principles as Einstein-relations, population inversion, optic feedback, threshold conditions, line shape and laser modes, passive and active resonators, single mode working, frequency stabilization, mode-locking and Q-switching are very common in this area of science. Most of these principles lie outside the scope of this thesis. But laser modes and the corresponding mode locking are two of the subjects of this thesis.

Computers are used to control a laser-system. In general the physicist himself develops the computer programs that control such a laser-system. Because these scientists typically have little know-how on computer science, these programs are often badly documented and lack a good (fundamental) structure due to the absence of a well-considered model. One of the objectives of the computer science department is to create methods to produce good and useful computer system models that can be used to develop new software. Computer system models come in different kinds and shapes where they all have their own advantages for different systems.

It was quite a coincidence that at the time I was searching for a subject for my thesis, a PhD by the name of drs. Maarten van Herpen was developing a new laser. This created the great opportunity to also develop new software to control this laser. At first the physics department had the intention to develop the software itself. When I came across this matter I had some good fundamental thoughts about a model for this particular piece of software. I convinced them that a good model is essential for a good program. They agreed to let me develop the necessary software to control the new laser and the subject of my master thesis was born.

The method that seems to be the most suitable for modelling this particular laser-system is the hybrid automaton model. Hybrid automaton models are used when there are continuous and discrete parts in the system that interacts with each other. A clear and simple example is the control-system of a car. The velocity of the car is an analogue value that changes continuously. The goal of the controller is to accelerate the car to a specified velocity. The control of the car can only make discrete steps in determining which action (the next acceleration) is to be taken. For more information on how this modelling process is conducted I refer to [6]. In case of this laser-system, we also deal with a hybrid system. At the one end the continuous analogue laser; and at the other end the discrete computer. To acquire enough background information on how to develop a hybrid automaton model I asked prof. dr. Frits Vaandrager to become my supervisor. After he accepted my proposal we asked prof. dr. ir. Jan Vytöpil to become my co-supervisor. His knowledge and experience on developing large hybrid feedback systems could be of great value. The next step was to construct a plan on how to approach this problem. The thesis that lies before you is the outcome of the construction of this plan.

The paper is divided into five parts and nine chapters. The first part captures the introduction to the paper. In this introduction a brief history is given of the thesis and the relationship of the author with subject. Furthermore an overview is stated to give the reader an idea what to expect. A separate chapter (chapter 2) is used to discuss the research questions and goals. The next two chapters are gathered in the following part that covers the static model and the laser-system as a whole. Chapter 3 deals with a lot of physics to explain the working of the laser itself. In several sections the basics of the laser (section 3.1), the use of actuators (section 3.2), cavity modes (section 3.3) and the etalon (section 3.4) are explained. This chapter is concluded with a model that combines the previous sections (section 3.5). In chapter 4 a quick overview is given on the laser-systems that are used regarding the software that is to be developed. Different parts that are used to control the laser and conduct the experiments are explained.

Three chapters, the dynamic model, form the third part of the thesis. Chapter 5 deals with one of the most important devices: (as later is shown) 'the lock-in'. This device uses a Fourier analysis (a mathematical method to compare different signals) to produce an error signal. The way it performs this analysis and how the error signal is constructed is the topic of this chapter. The last section of this chapter (section 5.3) is used to discuss the corresponding hybrid automaton model. In chapter 7 four different kinds of scans are discussed. Performing a scan is the main activity of these laser-systems gathering information on a particular specimen. For some scans it is necessary to scan all the wavelengths, to realise this a mode-lock is established. This mode-lock is the core of my research. The use of the mode-lock implies a feedback loop that uses previous results to determine the next action. The components of the feedback loop and the final model are explained in chapter 6. In the last section of this chapter (section 6.4) a brief discussion is given about a correctness assertion concerning the complete model.

The fourth part deals with the implementation of the designed model in LabVIEW. LabVIEW is a graphical programming language that is specially developed for physical purposes. In two chapters I will explain the use of LabVIEW and the main components of the program. The last part of the paper is reserved for appendices: document of demands, full system documentation of the program and the bibliography.

## Chapter 2 Research questions

Conducting my research and throughout the thesis, different goals are pursued. The main goal concerning the program is that there will be a working version of the software that satisfies the criteria of good software (as educated throughout the study). Some of these criteria are:

- The program can be used to support the research activities of the user (a document of demands is added to this paper as [appendix A1](#)).
- The software has a modular structure so that new expansions are easily integrated.
- The program should be well documented so that future maintenance can track the different choices that were made during the development of the current version (a system-documentation is added to this paper as [appendix A2](#)).
- The program should be clear and easy to comprehend.

To reach these goals a part of the laser-system will be modelled. This model shall be made of the mode-lock process that controls the stabilization of the laser beam. The mode-lock plays an important role in some of the research that is conducted using the laser. A good model of such a process is especially useful to get a complete and thorough comprehension of the process examined. The model should give insight in:

- The external variables that influence the process as a whole.
- The internal variables that determine (together with the external variables) the behaviour of the process.
- The dependencies of the external variables with the internal variables.
- The transitions that can be taken within the system that can influence the process and the corresponding conditions to these transitions.

Such a model gives us an advantage in the construction of the program. The structure of the model gives a body for the modular framework of the program. The model as such can be used in the documentation of the program. The other important reasons for models to be developed is to create the opportunity to make sound statements on the process. These statements can be proven using the model. The model can also be used to perform a simulation of the process to discover errors in the design of the process control.

The question that is considered throughout the thesis is the bridge between two worlds. At the one hand the world of the computer science, where models and proof tools are used to make correct software. The other world is the world of the physicists, where scientists are interested in the boundaries of the different parameters wherein there system works. The model constructed by the computer scientist gives more information on these boundaries concerning the performance of the computer system. The question that arose was:

*Which advantages or disadvantages does the use of a hybrid model have when developing real-time software used in controlling a laser-system?*

Advantages that are under consideration:

- The capability of the hybrid automaton model to enclose the behaviour of the laser-system.

- The use of the model in providing boundaries for the different parameters.
- The use of the model in constructing the computer program and how easy the different parts of the model are integrated in the program.

Within the scope of this thesis lies the evaluation of the used programming language. The use of LabVIEW within the Molecular Laser department as in the Computer Science is relatively new. This gives both departments the change to gain experience in the field of graphical programming. I will try to compare this language with the more common languages as C and Pascal.

## *Part 2: The static model*



# Chapter 3      The laser

In this chapter we discuss how a laser works. In the experimental set-up two different laser-systems are used. These two systems share the most common parts and despite their differences they work the same. The first section covers the most significant parts of the laser-systems and the basic physics involved. At this point both the laser-systems are explained. In the following sections I discuss only one laser-system, but it is arbitrary which of the stated systems is taken. At some places in the system actuators are assembled to a piece of the system. In the case of our laser-systems there are three such actuators; all of them will be explained.

Section 3.3 covers a necessary part on laser optics. Here I will explain why a laser cannot function on the entire frequency spectrum. The keyword for this phenomenon is called a mode. Modes play an important role in the earlier mentioned mode-lock (see [chapter 6](#) for more details).

The etalon is a key part of the laser-system (that we consider) concerning the stabilization of the laser beam. The stabilization of the laser beam is established due to performing a mode-lock. The use of the etalon and its optics are discussed in a separate section.

The different modes of the laser can be seen as states in a state-machine. In the last section of this chapter a hybrid automaton is discussed derived from the previous sections. This model will be used to formulate a correctness assertion on the whole system.

## 3.1 Lasers, the basics

Two different types of laser-systems are used in the experimental set-up. A brief description on how lasers work should be sufficient here to understand these two systems. For a precise description about the working of lasers in general, see "Lasers and Electro-optics, Fundamentals and Engineering" of Christopher C. Davis [1]. I will use formulae originating from this book throughout the following three chapters.

A laser is actually nothing more than two mirrors with a special medium between them, together called the cavity. The mirrors are high reflective, but at least at one mirror a small percentage of light is passed through. The medium is responsible for generating the laser beam. By putting energy in the medium (with ordinary light, a current or another laser beam) the medium will emit photons in a specific area of the light spectrum. When these photons (treated as waves) fit in the cavity then the light with the corresponding wavelength can be amplified (see for more information [section 3.3](#)). Lasers come in all different forms and sizes. The main difference is found in the kind of amplifying medium that is used. There are solid-state lasers (they use a crystal as the medium), gas lasers (a mixture of gases is used), dye lasers (dye dissolved in a fluid which flow through the cavity) and semiconductor lasers (where one can not speak of a medium as such). The medium chosen determines in which area of the spectrum amplification is possible. For example: A CO<sub>2</sub> gas laser works between 9 μm and 11 μm ( $2.7 \cdot 10^4$ - $3.3 \cdot 10^4$  GHz) [2], a YAG solid-state laser (Y<sub>3</sub>Al<sub>5</sub>O<sub>12</sub> with Nd<sup>3+</sup> ions) works between 0.9 μm and 1.4 μm ( $2.1 \cdot 10^5$ - $3.3 \cdot 10^5$  GHz) [3] and a New Focus Model 6262 semi-conductor laser operates between 1510 nm and 1590 nm ( $1.89 \cdot 10^5$ - $1.99 \cdot 10^5$  GHz) [4].

As mentioned before, two laser-systems are operational within the set-up. One of these lasers is a Colour Centre laser (FC-laser or just FCL; where FC stands for "Farben Centrum" which means 'colour centre' in German). The other laser is being developed by drs. M. van

Herpen and is called an Optical Parametric Oscillator laser (or just OPO-laser). Both lasers are so-called 'tuneable'; an explanation is given in [section 3.3](#)). I shall first explain the colour centre laser, followed by a discussion in which it differs from the OPO laser.

### The Farben Centrum laser

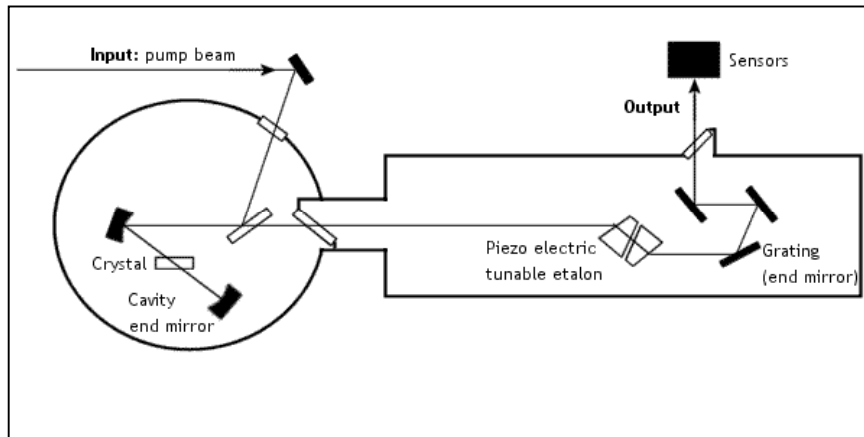


Figure 3.1: The interior of the FC laser.

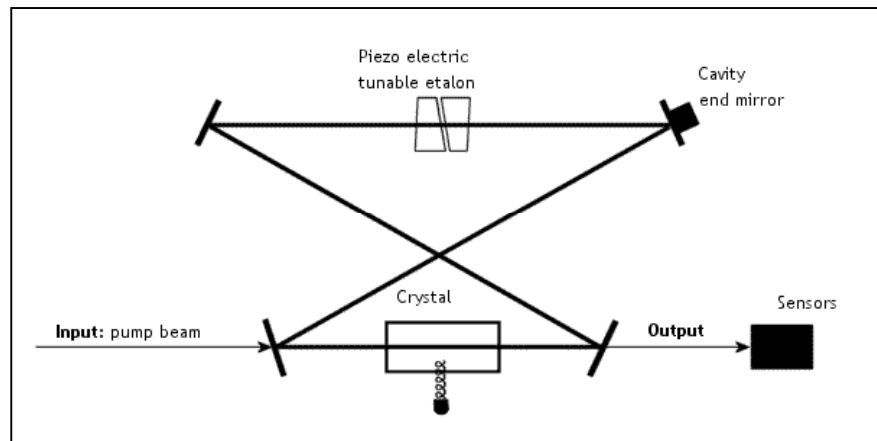
The cavity of this laser (where the interior is displayed in figure 3.1) is made of several mirrors and a grating, on which of one of the end-mirrors a piezo is attached to move the mirror over a small distance. The grating (a mirror with a milled edge) can be turned over an angle. The grating reflects the major part of the laser beam back into the cavity; a small part is directed outside the cavity to use during the experiment. The cavity of the laser is then defined as the area from the left end mirror to the grating.

The pump beam is inserted into the cavity through an opening in the shell of the laser and a mirror that only reflects at a particular angle is used. The energy of the light from the pump beam is used to get the molecular parts of the amplifying medium (in this case a crystal) in a higher energy state. This state is not stable and the molecules will emit energy in the form of photons. These photons will bounce back and forth through the cavity with a chance to trigger another emission when passing through the crystal. Due to this process more photons are filling the cavity. Not all of these photons can survive in the cavity, only those photons which have a particular wavelength (more in [section 3.3](#) on cavity modes). Because also warmth is emitted the crystal needs to be cooled with liquid nitrogen. The nitrogen atoms do not interfere with the laser process.

The mirrors of the cavity have a high reflection, so most of the light is kept within the cavity. Mirrors with a reflection of 98% till 99.99% are used to reflect the light. The end mirror of the laser has a reflectivity of 97%. The etalon, which is placed within the cavity, also has two mirrors situated opposite to each other. The reflection of these mirrors is in general of a much lower value; in the case of the FC laser a reflectivity of 30% is used. More on the etalon and its use can be found in [section 3.4](#).

## The Optical Parametric Oscillator laser

Figure 3.2: The interior of the OPO laser.



The first, and most important, difference between the FC laser and the OPO laser is the fact that the OPO laser has a ring cavity (which is schematically shown in figure 3.2). Within a ring cavity the light is not bounced back and forth, but circulates within the cavity. There is one moveable 'end mirror', an etalon (both driven by a piezo) and a crystal. The OPO laser uses no grating, but the same effect of turning the grating within the FC laser is achieved by moving the crystal perpendicular to the light beam. How the latter is achieved lies outside the scope of this paper, but can be read in [5] which deals with the structure of the crystal.

### 3.2 The actuators

Two different kinds of actuators are used within the laser-system: the piezo and the stepper-motor. The piezo is a small device that can be used to move objects over very small distances. It is possible to change the position of the piezo within micrometers. A continuous current controls the piezo and there is a linear relationship between the current and the position of the piezo. The computer gives a low current between zero and ten volts, but the piezo needs a voltage between zero and hundred volts. So a high-voltage supply is used to boost the signal in sending it to the piezo. The boost is set to multiply the signal one hundred times, hence the signal from the computer is limited between zero and one volt.

The stepper-motor is a different, less precise device. The control of the stepper-motor is also very different. Where the piezo only needs one (analogue) signal, the stepper-motor uses ten digital signals. The first eight signals are used to set the speed and the position of the motor. The ninth signal is used to turn the motor on. Finally the tenth signal is a controller of the stepper-motor that sends a signal back that the motor has reached its position.

### 3.3 Cavity modes

Every photon has its characteristic wavelength. Within a cavity only the photons with the correct wavelength can survive within the cavity. The correct wavelengths are those that fit exactly  $m$  times within the cavity for some natural number  $m$ . The wavelengths that fit this condition are called modes. Hence, these modes are a direct result of the optical path length  $l$  of the cavity and therefore they are also called cavity modes. Modes are also called transmission peaks. In figure 3.3 the cavity is represented as two parallel mirrors ( $L$  meters apart) with a certain medium between them. In general the refractive index of this medium  $n$  will differ from the refractive index of the medium  $n'$  outside the cavity.

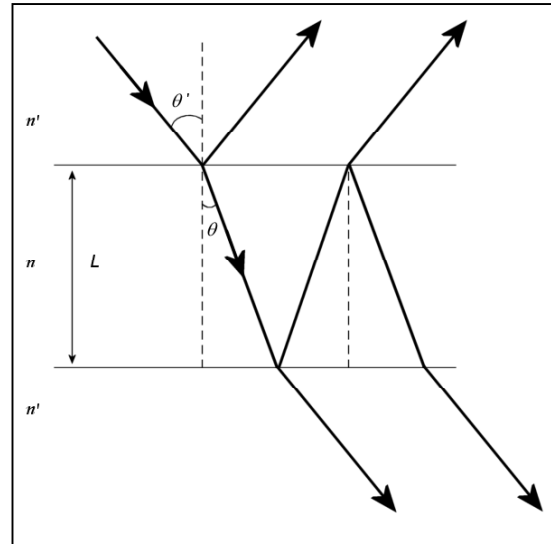


Figure 3.3: A schematic reproduction of a laser cavity.

Hence, the incoming angle of the light  $\theta'$  will differ from the angle of the light  $\theta$  between the two mirrors. The frequencies that have the maximal transmission in this cavity satisfy:

$$v_m = \frac{mc_0}{2nl \cos\theta} \quad (3.1)$$

where  $c_0$  is the speed of light *in vacuo* and  $m$  is an integer [1]. This formula shows that a change in the refractive index or a change in the optical path length results directly in a change in the corresponding frequency of the mode. The laser-system is always adjusted to make the optical path as short as possible, therefore we can assume that the angle  $\theta$  is zero and hence  $\cos\theta$  is one. The optical path length is given by (3.2). Here we take the refractive index ( $n(x)$ ) to be constant and  $L$  is the physical length between the two mirrors of the cavity:

$$\begin{aligned} l &= \int_0^L n(x) dx \\ &= [xn]_0^L \\ &= Ln \end{aligned} \quad (3.2)$$

Between every mode there is a region where no resonance can occur. This region is called the *free spectral range* (FSR). The distance of this free spectral range  $\Delta v$  can be taken from (3.1) using (3.2). This gives us:

$$\Delta v = \frac{c_0}{2n^2 L} \quad (3.3)$$

When the path length is changed, for example the optical path becomes larger due to a miniscule vibration of one of the end mirrors; it has an opposite effect on the FSR. In this case the FSR will become smaller which results in the modes being closer to each other. A minor change in the refractive index or in the path length results in a major frequency shift for the higher cavity modes.

It is also possible to determine the current cavity mode when the frequency ( $\nu_m$ ) is known (what in general will be the case). The mode number  $m$  is calculated using (3.3):

$$m = \frac{2n^2\nu_m L}{c_0} \quad (3.4)$$

Usually not the frequency  $\nu_m$  but the wavelength  $\lambda_m$  is known. The two quantities are related by the equation:

$$\nu_m = \frac{c_0}{n\lambda_m} \quad (3.5)$$

When we combine (3.4) with (3.5) we get the mode number depending on the wavelength:

$$m = \frac{2nL}{\lambda_m} \quad (3.6)$$

### Cavity finesse

A frequency corresponding with a mode is not strict. What this means is that the transmission peak is not an exact point on the wave spectrum, but a function around a specific *centre frequency*. Hence, frequencies that are close to this centre frequency can be amplified. The transmission peak at this centre frequency can be seen as a Lorentzian function with a particular width. To measure the width of Lorentzian we look at the full width at half the maximum transmission peak (FWHM).

It is known that the sharpness of the transmission increases with the reflectivity  $R$  of the mirrors used for the cavity [1]. This is represented in the *finesse*  $F$  of the cavity:

$$F = \frac{\pi\sqrt{R}}{1-R} \quad (3.7)$$

We see that the finesse increases as the reflectivity increases. With this we can give a measurement for the FWHM  $\Delta\nu_{1/2}$ :

$$\Delta\nu_{1/2} = \frac{\Delta\nu}{F}, \quad (3.8)$$

where  $\Delta\nu$  is the free spectral range. The higher the finesse the narrower the transmission peaks get. Also, the higher the free spectral range the broader the transmission peaks become.

### Numerical example

To make the previous section more understandable a little example is given.

We take a look at the FC-laser that is used in the laboratory for research. The cavity of this laser is 54,55 cm. For the most part the laser cavity is plain air, the part where the amplifying takes place is designed not to resonance in that particular section of the cavity. For this we say that the refractive index of the cavity is the refractive index of air ( $n = 1,00029$ ). With this given we can calculate the free spectral range using (3.3):

$$\begin{aligned} \Delta\nu &= \frac{c_0}{2n^2L} \\ &= \frac{2,998 \cdot 10^8}{2 \cdot (1,00029)^2 \cdot 0,5455} \end{aligned}$$

$$= 274.634.456,1\text{K} \approx 275 \cdot 10^6 = 275 \text{ MHz}$$

It is known that the FC-laser operates in the 2,7 - 3,1  $\mu\text{m}$  region of the light spectrum. These wavelengths correspond with respectively the following frequencies:  $1,11 \cdot 10^5$  GHz and  $9,67 \cdot 10^4$  GHz (using (3.5)). The total number of cavity modes  $M$  can be calculated through division using (3.6):

$$M = \frac{2 \cdot 1,00029 \cdot 0,5455}{2,7 \cdot 10^{-6}} = 404.191 \text{ modes}$$

The number of modes that lie within the region of the laser ( $\Delta M$ ) are:

$$\Delta M = 52.153 \text{ modes}$$

To calculate the FWHM we first need to determine the finesse of the cavity. As stated in this section the finesse is totally dependant on the reflectivity of the mirrors. In this case a reflectivity of approximately 97% is used:

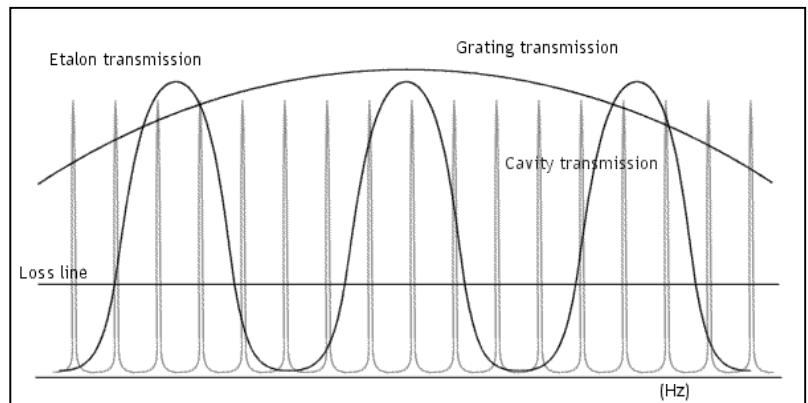
$$F = \frac{\pi \sqrt{R}}{1-R} \approx \frac{\pi \sqrt{0,97}}{1-0,97} \approx 103,14$$

$$\Delta \nu_{1/2} = \frac{\Delta \nu}{F} \approx \frac{274.634.456,1}{103,14} \approx 2,7 \cdot 10^6 = 2,7 \text{ MHz}$$

We see that the width of the transmission peak is about one hundred of the free spectral range of the cavity. This means that the peak is very sharp in contrast with the area where there is no resonance possible.

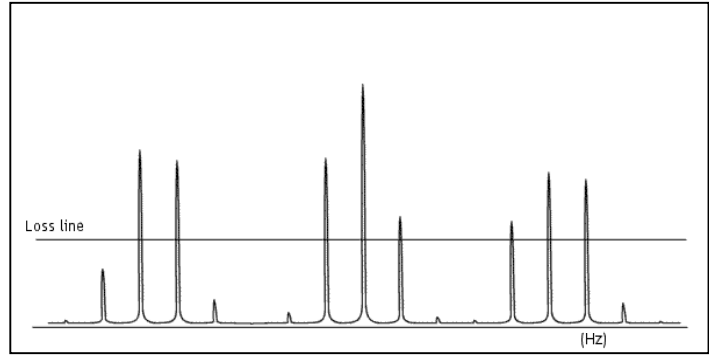
### 3.4 The use of the etalon

Figure 3.4: The three different transmission peaks and the loss line of the laser.



An etalon is a small resonator constructed by only two mirrors (the etalon cavity). The etalon has the same construction as a laser but without the amplifying medium to enhance the power of the light beam. Because the etalon has its own cavity it also has its own transmission peaks and free spectral range. Here the mirrors are far more closely together than the mirrors of the laser cavity. Therefore the FSR of the etalon will be much broader than the FSR of the laser. The finesse of the etalon is very low. In combination with the broad free spectral range a large FWHM is the result. This results in broad transmission peaks of the etalon (etalon modes). We shall see that many cavity modes are captured within one etalon mode.

Figure 3.5: The transmission line of the laser.



The tricky part is to position the top of one of the etalon transmission peaks over the centre frequency of a cavity mode. In this way only the specified frequency will pass through the etalon and the other competing neighbour cavity modes are died out. By looking at figure 3.4 (a what

simplistic reproduction) we see that the centre etalon transmission peak coincides with the transmission peaks of the cavity and the grating. In terms of figure 3.4 we need to multiply the different transmission lines with each other to acquire the transmission line of the laser as a whole (see figure 3.5). The figure clearly shows that only one cavity mode has the best opportunity to beat the other modes.

Within a laser-system there is a particular (constant) amount of energy available for making the laser work. A minimal amount of this energy is necessary to start an amplifying process of a particular mode. This minimal amount is known as the *loss line*. Modes that lie below the loss line have no chance of creating a laser effect.

Nevertheless there are still other modes that lie above the loss line, but due to mode competition the centre mode will beat the others. This principle is easy to explain because building up the cascade requires energy and therefore less energy is available for the other modes. In terms of figure 3.5 the loss line of the other modes becomes higher.

### Power output of the laser

The use of an etalon in a laser-system has an effect on the power output of the laser beam. The power of a laser beam depends mostly on the reflectivity of the mirrors and the kind of amplifying medium that is used. The laser cavity cause losses to the power output as will the etalon cavity. In case of the etalon these losses will depend on the difference between the current frequency  $\nu_c$  and the centre frequency of the etalon  $\nu_e^*$ . In most literature the intensity of the laser beam is given. The relationship between the power and the intensity of the laser beam is:

$$P(\nu_c) = \frac{dI(\nu_c)}{dt}, \quad (3.9)$$

with:

$$I(\nu_c) = I_0 \cdot [\alpha_1 + \alpha_{etalon}(\nu_c)] \cdot \text{sinc}^{-2} \left( \frac{\Delta k(\nu_c) L_c}{2} \right) \quad (3.10)$$

$$\alpha_{etalon}(\nu_c) = \frac{2R_e (1 - \cos\left(\frac{\Delta \nu_c^1}{\Delta \nu_e}\right))}{1 + R_e^2 - 2R_e \cos\left(\frac{\Delta \nu_c^1}{\Delta \nu_e}\right)}$$

$$\Delta \nu_c^1 = \nu_c - \nu_e^*$$

$$\text{sinc}^2(\beta) = \frac{\sin^2(\beta)}{\beta^2}$$

where the factor  $I_0$  is assumed to be independent of the signal frequency. In the second factor,  $\alpha_1$  denotes the losses of the cavity without etalon, and  $\alpha_{etalon}(\nu_c)$  denotes the additional, frequency selective losses induced by the etalon. Finally,  $\Delta k(\nu_c)$  denotes the phase factor within the cavity [8].

## Numerical example

We use the example of the previous section to make the theory more understandable.

Again we take a look at the FC-laser that is used in the laboratory for research. The cavity of the etalon  $L_e$  is 7,0 mm. The interior of the etalon cavity is plain air; for this we say that the refractive index of the cavity is the refractive index of air ( $n = 1,00029$ ). With this given we can calculate the free spectral range of the etalon  $\Delta\nu_e$  using (3.3):

$$\begin{aligned}\Delta\nu_e &= \frac{c_0}{2n^2L_e} \\ &= \frac{2,998 \cdot 10^8}{2 \cdot (1,00029)^2 \cdot 0,0070} \\ &= 21.401.870.829,307\text{K} \approx 21.401 \cdot 10^6 = 21.401 \text{ MHz} = 21,4 \text{ GHz}\end{aligned}$$

The total number of etalon modes  $M_e$  are again given to us using formula (3.6) and knowing that the laser works within the 2,7 - 3,1  $\mu\text{m}$  region of the light spectrum:

$$M_e = \frac{2 \cdot 1,00029 \cdot 0,0070}{2,7 \cdot 10^{-6}} = 5.187 \text{ modes}$$

The number of modes that lie within the region of the laser  $\Delta M_e$  are:

$$\Delta M = 670 \text{ modes}$$

To calculate the FWHM we first need to determine the finesse of the etalon cavity  $F_e$ . In this case a reflectivity  $R_e$  of approximately 30% is used:

$$\begin{aligned}F_e &= \frac{\pi \sqrt{R_e}}{1 - R_e} & \Delta\nu_{\frac{1}{2}e} &= \frac{\Delta\nu_e}{F_e} \\ &\approx \frac{\pi \sqrt{0,30}}{1 - 0,30} & &\approx \frac{21.401 \cdot 10^6}{2,46} \\ &\approx 2,46 & &\approx 8,7 \cdot 10^9 = 8,7 \text{ GHz}\end{aligned}$$

We see that the width of the transmission peak  $\Delta\nu_{\frac{1}{2}e}$  is less then one third of the free spectral range of the etalon cavity. This means that the peak is very broad in contrast of the area where there is no resonance possible. We can give a rough indication on how many cavity modes are under one etalon transmission peak by dividing the FWHM of the etalon through the FSR of the laser cavity. This tells us that there are approximately 31 cavity modes under one etalon transmission peak.

## 3.5 The automaton model

Now we need to define an automaton model that captures the above. The cavity and the etalon modes of the laser-system are perfectly useable for the states in the model. With this choice the model gets a large number of states. Because this amount of states would make the picture difficult to survey I reduced the number of states that is visualized within the picture. In fact I only need two states: one state with the current mode number and another state for a different mode number. When the laser makes a mode-hop it is arbitrary to which new mode it jumps, the only thing that is sure is that the new mode is different from the previous one.

Now the only question is: when does the laser make a mode-hop? We distinct two different causes for the mode of the laser to make a jump: due to a disturbance and due to the user by moving the piezo of the etalon.

We presume that the disturbance of the laser-system is always caused by a change in the refractive index of the air. Changes in the pressure of the room or the presence of moving air can result in changes of the refractive index (with  $\chi$  the maximum disturbance possible). Knowing this we can say:

$$d = \frac{dn}{dt}, \text{ where } d \in [-\chi, \chi] \quad (3.9)$$

Using formula (3.4) we can express the frequency  $\nu_c$  in terms of the length of the cavity  $L_c$ , the cavity mode  $m_c$  and the refractive index  $n$ .

$$\nu_c = \frac{c_0 m_c}{2n^2 L_c} \quad (3.10)$$

The next step is to make a statement about the maximal frequency shift that is allowed before the mode-hop may occur. We estimate that the maximal shift is equal to the half of the full width at half maximum (FWHM) of the etalon. Thus when the frequency corresponding with the current cavity mode is drifting outside the etalon mode a mode-hop may occur. This condition can be expressed as follows:

$$\begin{aligned} \nu_c &\in [\nu_e - \Delta\nu_{\max}, \nu_e + \Delta\nu_{\max}] \\ \Delta\nu_{\max} &\equiv \frac{1}{2} \Delta\nu_{\frac{1}{2}e} = \frac{1}{2} \cdot \frac{\Delta\nu_e}{F_e} \\ &= \frac{c_0}{4n^2 L_e F_e} \end{aligned} \quad (3.11)$$

Also the user can influence the state of the laser by changing the distance between the etalon mirrors  $L_e$ . This change in length is caused by a change in the voltage on a piezo that is attached to one of the etalon mirrors. There is a linear relationship between the distance of the etalon mirrors and the voltage on the piezo  $\rho_e$  (with  $\tau$  a scaling constant and  $L_0$  the minimal distance between the mirrors):

$$L_e = L_0 + \tau \cdot \rho_e \quad (3.12)$$

When applying the definition of  $m_e$  we obtain an expression with a frequency  $\nu_e$ . In this case the frequency is the centre frequency of the etalon mode. We can derive the following equation between the frequency and piezo voltage:

$$\begin{aligned} \nu_e &= \frac{c_0 m_e}{2n^2 L_e} \\ &= \frac{c_0 m_e}{2n^2 (L_0 + \tau \cdot \rho_e)} \end{aligned} \quad (3.13)$$

The above condition and the corresponding equations are gathered in figure 3.6, where the two states are shown with the 'mode-hop'-transition between them.

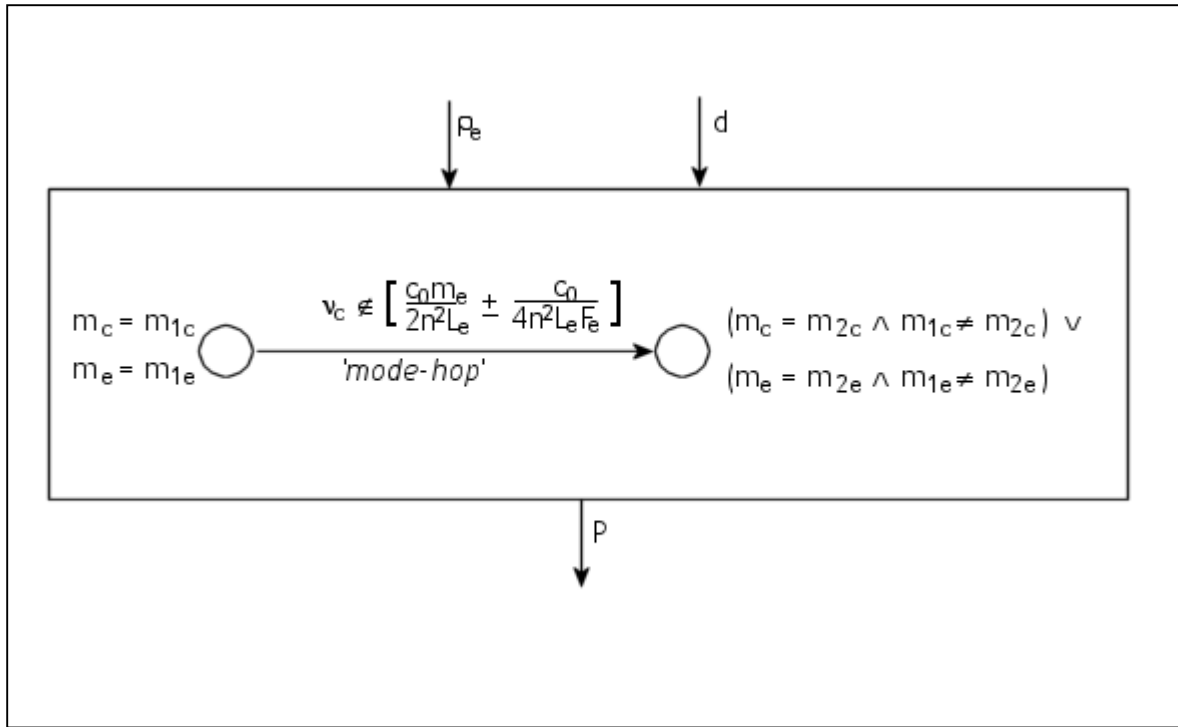


Figure 3.6: An automaton of a laser. An m-state machine (using the mode numbers as state indicators) with the corresponding transition between two states.

Input variables:  $\rho_e (\mathfrak{R}), d ([-\chi, \chi])$

Output variables:  $P (\mathfrak{R})$

Internal variables:  $v_c (\mathfrak{R}), v_e (\mathfrak{R}), L_c (\mathfrak{R}), L_e (\mathfrak{R}), L_0 (\mathfrak{R}), F_c (\mathfrak{R}), F_e (\mathfrak{R}), R_c (\mathfrak{R}), R_e (\mathfrak{R}), n (\mathfrak{R}), m_c (\mathfrak{S}), m_e (\mathfrak{S})$

States: All combinations of valuations of the internal variables  $m_c$  and  $m_e$

Transitions: see figure 3.6

Equations:

$$L_e = L_0 + \tau \cdot \rho_e$$

$$d = \frac{dn}{dt}$$

$$v_c = \frac{c_0 m_c}{2n^2 L_c}$$

$$v_e = \frac{c_0 m_e}{2n^2 L_e}$$

$$F_c = \frac{\pi \sqrt{R_c}}{1 - R_c}$$

$$F_e = \frac{\pi \sqrt{R_e}}{1 - R_e}$$

$$P(v_c) = \frac{dI(v_c)}{dt}$$

$$I(v_c) = I_0 \cdot [\alpha_1 + \alpha_{etalon}(v_c)] \cdot \text{sinc}^{-2} \left( \frac{\Delta k(v_c) L_c}{2} \right)$$

$$\alpha_{etalon}(v_c) = \frac{2R_e (1 - \cos(\frac{\Delta v_c^1}{\Delta v_e})}{1 + R_e^2 - 2R_e \cos(\frac{\Delta v_c^1}{\Delta v_e})}$$

$$\Delta v_c^1 = v_c - v_e^*$$

$$\Delta v_e = \frac{c_0}{2n^2 L_e}$$

$$v_c = \frac{c_0}{n \lambda_c}$$

$$v_e^* = \frac{c_0}{n \lambda_e}$$

$$\text{sinc}^2(\beta) = \frac{\sin^2(\beta)}{\beta^2}$$

# Chapter 4 The physical layout

In this chapter I will give an overview of the physical layout of the laser-system. This will be a brief description of the different devices that are used in the set-up of an experiment. The devices that I will discuss are (in order): the wavetek, the photodiode, the lock-in, the spectrum analyser and the computer (see figure 4.1). In the previous chapter the laser is described and in the next chapter the lock-in will be discussed in more detail.

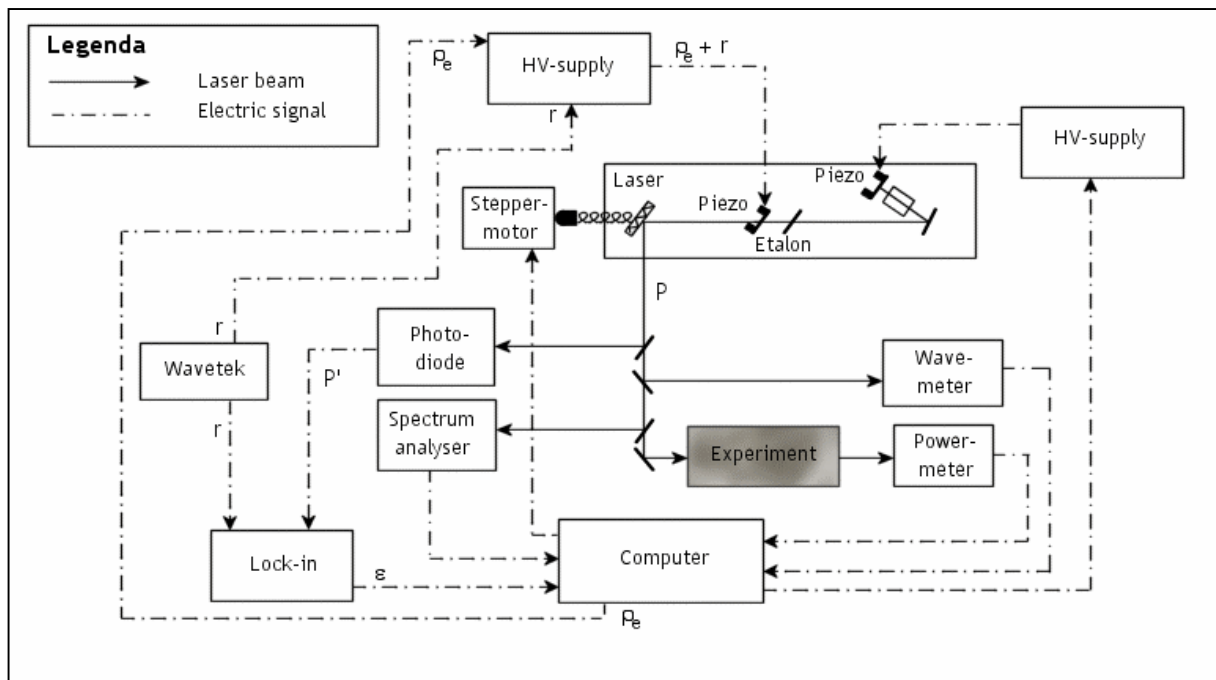


Figure 4.1: An overview of the complete system.

## The wavetek

The wavetek is a device that produces a variable current. The way the current is changed over time is adjustable. The wavetek can produce a sinus-current, a saw-current and a step-current, were in any case the amplitude and the period is adjustable. In this case, the wavetek is used to put a sinus-current on the etalon piezo (a jitter). This jitter is used to create an error signal (see for more information [section 5.2](#)).

The reference frequency (the result of the sinus-current) is sent to the same HV-supply as where the computer program sends its signal. The HV-supply combines these signals into one signal that positions the piezo. The reference signal is also used by the lock-in.

## The photodiode

The photodiode is a very simple device that only measures the changes in the power output of the laser beam. Hence, it takes the derivative of the power function produced by the laser.

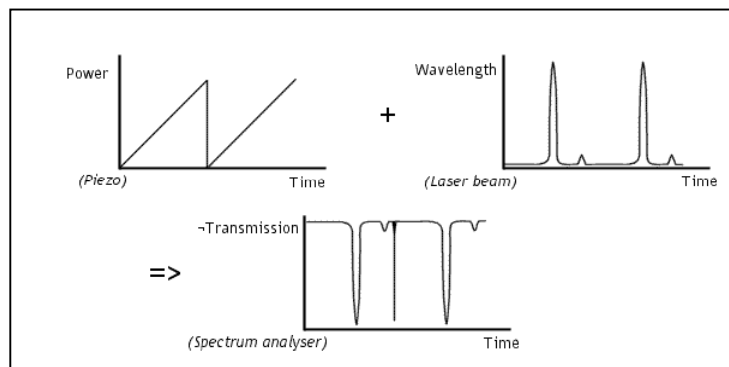
## The lock-in

The lock-in compares two signals and returns a real number that indicates the equality of the two signals. In this case the signal from the wavetek is compared with the derivative of the power output. The output signal of the lock-in is used as error signal; a signal that plays an important role in the feedback loop that controls the mode-lock (see [chapter 5](#)).

## The spectrum analyser

When conducting an experiment, the spectrum analyser plays an important role in retrieving significant information about the specimen examined. With the analysis of the spectrum of the different substances the error signal is also used, but not to keep the mode locked. The goal of a spectrum analysis is to determine the composition of the different frequencies of the laser beam.

Figure 4.2: The use of the cavity end-mirror piezo for the spectrum analysis.



This goal is accomplished by moving the end mirror of the laser cavity (using a piezo) over a specified distance (see figure 4.2). The mirror is moved with a constant velocity and displaces periodically over the specified distance.

The distance is determined by setting a particular maximum voltage on the piezo. A saw-function on the voltage of the piezo is used to move the end mirror. This results in a shift over the frequency domain. When the transmission peak of the etalon is in phase with a cavity mode it will be visualised by a peak in the diagram. The figure above shows two peaks, the large peak shows the main frequency, but the smaller peak indicates a second mode is competing and is getting a change to amplify. In a perfect situation only one frequency is active at a time.

The spectrum analysis is contra intuitive visualised. It gives a high signal when there is no transmission and a low signal when there is full transmission. This phenomenon is caused by a photodiode that is used for the analysis. In general a broader area is scanned to ensure that the whole frequency range between two cavity modes is scanned. The small peak in the middle of the spectrum analyser is a result from the quick return of the piezo to its start position.

## The computer

Last, but not least, there is the computer. The role of the computer is very broad. It is used to store the great amount of sensor data that is produced throughout an experiment. It collects the data and stores it in files so it can be used for post-analysis. The computer has also control over the different actuators connected to the laser (the two piezos and the stepper-motor). Moreover the computer is in charge when performing a scan. There are four different kinds of scans that the computer can perform: the grating scan, the mode-hop scan, the mode-hop-free scan and the pump scan (see for more information about scans [chapter 7](#)).

When using the mode-hop-free scan it is important that the laser stays in a particular mode. To keep the laser in this mode a feedback loop called 'mode-lock' is used. At this point the computer is also used to act as a controller that closes the feedback loop (more about the mode-lock in [chapter 6](#)).

## *Part 3: The dynamic model*



# Chapter 5 The Lock-in

The lock-in is a sort of analogue-digital converter that converts the output signal of the laser into an error signal. In this chapter the mathematical background of the lock-in is explained and completed with some practical examples. Furthermore the error signal is discussed and an automaton model is given of this device.

## 5.1 Mathematical background

To understand the working of the lock-in a little analytic mathematic is necessary. The lock-in uses a kind of Fourier analysis to compare the incoming signals. A Fourier analysis is used to determine whether a particular signal (of some frequency) is present in a signal. When this is the case, it also tells us the contribution of the particular signal in the whole signal. To make this analysis the following integral is solved:

$$\int_0^T r(t) \cdot (S(t) + \phi) dt, \quad (5.1)$$

where  $r(t)$  is the reference signal and  $S(t)$  is the signal in question. The addition  $\phi$  is used to correct any unnecessary phase shifts. This  $\phi$  is manually set by the user and therefore has no origin in the rest of the process. To demonstrate this Fourier analysis a few examples are given. Within these examples  $\phi$  is chosen to get the right phase and in the end we take  $T = 2\pi$ .

*Example 1:*

$$\begin{aligned} \int_0^T \cos(\omega t) \cdot \cos(\omega t) dt &= \int_0^T \cos^2(\omega t) dt \\ &= \left[ \frac{t}{2} + \frac{\sin(2\omega t)}{4\omega} \right]_0^T \\ &= \frac{T}{2} + \frac{\sin(2\omega T)}{4\omega} \\ &= \pi \end{aligned}$$

*Example 2:*

$$\begin{aligned} \int_0^T \cos(\omega t) \cdot \sin(\omega t) dt &= \left[ \frac{\sin^2(\omega t)}{2\omega} \right]_0^T \\ &= \frac{\sin^2(\omega T)}{2\omega} \\ &= 0 \end{aligned}$$

*Example 3:*

$$\begin{aligned} \int_0^T \cos(\omega t) \cdot \cos(2\omega t) dt &= \left[ \frac{\sin((\omega-2\omega)t)}{2(\omega-2\omega)} + \frac{\sin((\omega+2\omega)t)}{2(\omega+2\omega)} \right]_0^T \\ &= -\frac{\sin(-\omega T)}{2\omega} + \frac{\sin(3\omega T)}{6\omega} \\ &= 0 \end{aligned}$$

Example 4:

$$\begin{aligned} \int_0^T \cos(\omega t) \cdot \cos(\omega t + \pi) dt &= - \int_0^T \cos^2(\omega t) dt \\ &= - \left[ \frac{t}{2} + \frac{\sin(2\omega t)}{4\omega} \right]_0^T \\ &= - \frac{T}{2} - \frac{\sin(2\omega T)}{4\omega} \\ &= -\pi \end{aligned}$$

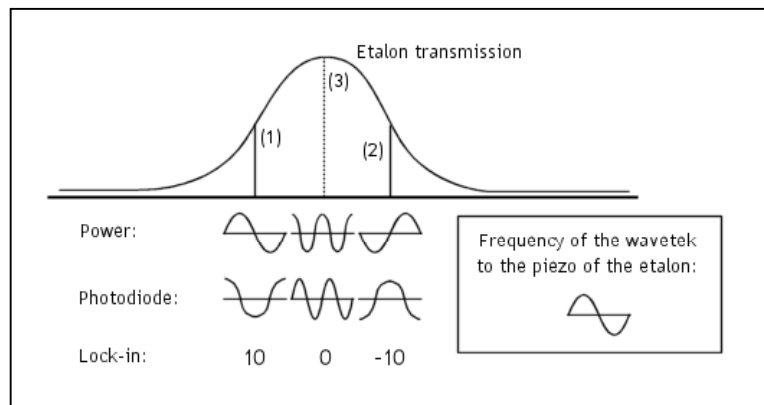
Example 5:

$$\begin{aligned} \int_0^T \cos(\omega t) \cdot \left( \frac{1}{3} \cos(\omega t) + \frac{2}{3} \cos(2\omega t) \right) dt &= \frac{1}{3} \int_0^T \cos^2(\omega t) dt + \frac{2}{3} \int_0^T \cos(\omega t) \cos(2\omega t) dt \\ &= \frac{1}{3} \cdot \pi + \frac{2}{3} \cdot 0 \\ &= \frac{1}{3} \cdot \pi \end{aligned}$$

The preceding examples make it possible to indicate some particular phenomena. Example 1 shows two similar signals give a positive value (in this case  $\pi$ ). Example 2 indicates that when two sinus-signals with the same period and amplitude have a phase difference, the analysis shows us that they cannot be considered identical. With example 3 we show that when a signal has a double period compared to the reference signal, the analysis also concludes that the two signals are not identical. Example 4 shows that a phase difference of  $\pi$  gives us just the opposite value of example 1. The last example (example 5) is the most interesting one. When a signal is a composition of different signals the Fourier analysis can indicate which part of the composition is the reference signal. In example 5 we see that one third of the signal is identical with the reference signal.

## 5.2 The error signal

Figure 5.1: Etalon transmission peak with three cavity modes.



As a result of the Fourier analysis (as explained in the previous section) the error signal is obtained.

This error signal is formed through four steps. At first a jitter is put

on the etalon. This jitter comes from a device that is called a wavetek (see [chapter 4](#)). The signal that causes the jitter is called the *reference signal*. A result of the jitter is that one of the mirrors from the etalon starts to move constantly over a small distance in a small amount of time. A common used frequency is 200 Hz, thus the mirror is moved back and forth 200 times per second. As a direct result little changes will occur in the power output of the laser beam. In figure 5.1 an etalon transmission peak is shown with three possible positions of the current cavity mode. The etalon is in the right position when the etalon transmission peak is exactly in line with a cavity transmission peak (situation 3). With a sinus as reference signal there will first be a drop in the power output, then a rising occurs and this will repeat itself in the same period as the reference signal. When the transmission peak of the etalon is out of phase with the peak of the cavity another signal signature will be emerged (situation 1 and 2).

In this case not the power itself but the changes in the power are measured by a photodiode. The photodiode gives us the differentiation of the power output. The measured signal is then send to the lock-in that compares the signal with the reference signal coming from the wavetek. The comparison is done with a sort of Fourier analysis where a possible phase difference with the reference signal is taken into account (see [section 5.1](#)). The lock-in sends a continuous signal to the computer called the error signal (indicated with the letter  $\varepsilon$ ). This signal is zero when the two input signals are completely different and thus the transmission peaks of the etalon and the cavity coincide with each other. When there is some comparison the lock-in will send a value from zero through ten (or minus ten). The computer uses this value to calculate the action that needs to be taken to get both the transmission peaks back in phase.

### 5.3 The automaton model

To complete the automaton model two local variables are introduced (*hist\_r* en *hist\_P'*) to store the history of the reference signal and the signal coming from the photodiode. These variables store the values of the corresponding signals of the last time interval with size  $T$ . The following hybrid automaton is shown in figure 5.2. In this figure a state machine with only one state is shown without any transitions. This indicates that there is a continuous analogue output.

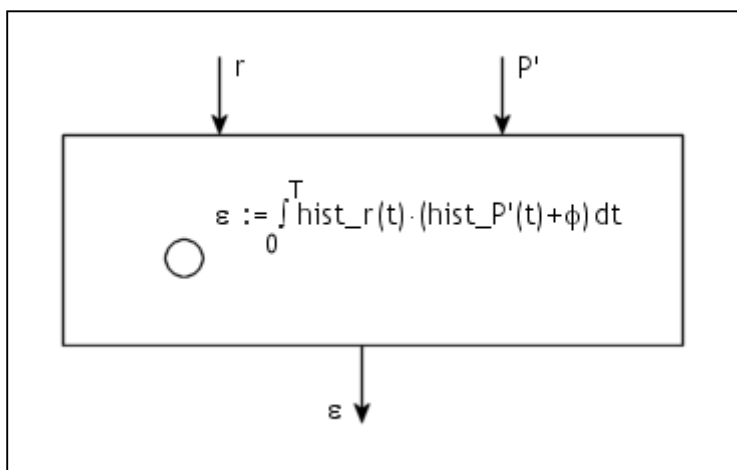


Figure 5.2: An automaton of the lock-in. A one-state machine producing an analogue output.

Input variables:  $r (\mathfrak{R}), P' (\mathfrak{R})$

Output variables:  $\varepsilon (\mathfrak{R})$

Internal variables:  $\text{hist}_r ([0,T] \rightarrow \mathfrak{R}), \text{hist}_{P'} ([0,T] \rightarrow \mathfrak{R}), \phi (\mathfrak{R}), T (\mathfrak{R})$

States: There is one continuous state

Start state: The first  $T$  seconds the automaton assigns zero to  $\varepsilon$

Transitions: There are no discrete transitions

Equations:

$$\forall_{t \in [0, T]} [\text{hist}_r = r]$$

$$\forall_{t \in [0, T]} [\text{hist}_{P'} = P']$$

$$\varepsilon = \int_0^T \text{hist}_r(t) \cdot (\text{hist}_{P'}(t) + \phi) dt$$



# Chapter 6 The mode-lock

The mode-lock is used in the mode-hop-free scan (explained in [section 7.3](#)). In this chapter I will explain why it is so important to avoid mode-hops during these scans and how a mode-lock can help achieve this. Furthermore, I will explain the complete feedback loop that controls the mode-lock. Together this results in a model of the mode-lock as a process. This chapter concludes with a discussion on a correctness assertion on this model.

## 6.1 The use of the mode-lock

As mentioned in [section 5.3](#), mode-locks are used to cover complete spectrum of frequencies while performing a scan. When a mode-hop occurs, there is no way to control which frequency is selected after the hop. Engaging a mode-lock ensures that the frequency used during the scan is known at all times.

The mode-lock aligns the transmission peak of an etalon mode with the transmission peak of a cavity mode. To keep both transmission peaks in line we need to adjust the position of the etalon mode as soon as the cavity modes starts to shift. Due to a disturbance in the laser-system, or by moving one of the end mirrors (the mode-hop-free scan), the free spectral range of the cavity will change. This results in a mode-shift and the free spectral range of the etalon has to be adjusted to suit the new conditions. To detect mode-shifts an error signal is used. The error signal is generated using a jitter on the etalon, a photodiode and a lock-in. The computer is used to interpret the error signal and make changes to the free spectral range of the etalon. The first three parts of this process have been discussed in previous chapters; the next few sections concern themselves with the role of the computer in this process.

## 6.2 The feedback loop

The computer closes the feedback loop. For this purpose the program uses the error signal and a value  $\alpha$  that represents the history of the previous actions to determine the next action. Depending on the value of the signal send by the lock-in, the program calculates the new voltage that is put on the piezo of the etalon. When the error signal lies within a particular range around zero or the new signal value differs too little from the old, the situation stays unchanged. The program itself computes the increase or decrease in voltage needed to correct the mode-shift.

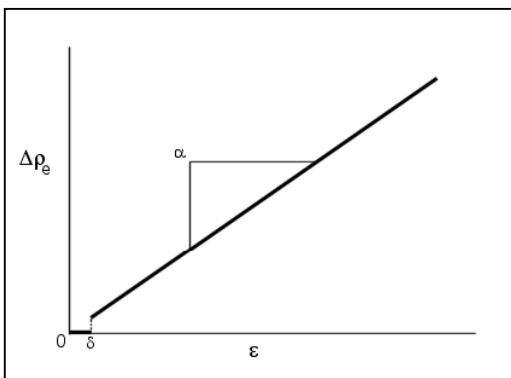


Figure 6.1: The relationship between the error signal and the adjustment on the piezo voltage

To determine the new piezo voltage a function  $f(\varepsilon, \varepsilon_0, \alpha)$  is evaluated. This function returns the voltage that needs to be added to or subtracted from the old piezo voltage. To make a first approximation we consider the relationship between

the error signal and the corresponding change in the voltage of the piezo as a linear function (see figure 6.1). We normalize this function by presuming that it always go through the origin of the axis. To determine the final value of the function  $f$  the effect of the previous action is taken into account. To calculate the effect the previous error signal is used. If the effect is not too small the calculated value is divided by the effect. In the case of a very small effect the result value of the function will blow up. A high voltage on a piezo may result in damaging the device.

$$f(\varepsilon, \varepsilon_0, \alpha) := y, \text{ where} \tag{6.1}$$

$$y \equiv \frac{\varepsilon \cdot \alpha}{\text{eff}(\varepsilon, \varepsilon_0)}$$

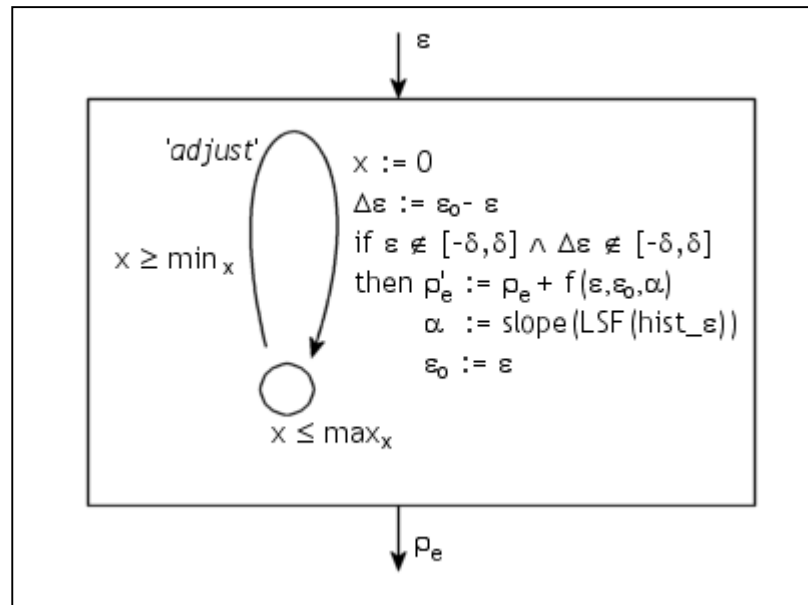
$$\text{eff}(\varepsilon, \varepsilon_0) \equiv \left| 1 - \frac{\varepsilon}{\varepsilon_0} \right|$$

To calculate the new slope of the function a *best-fit algorithm* is used on the previous results. The values of the last several points are stored to determine a *best fit*. The best-fit algorithm is similar to a least square fit algorithm (see for more information on this topic [7]). The function results in new values for the slope  $\alpha$ .

### 6.3 The automaton model

The results from the previous section can be summarised in the following hybrid automaton. It is a one-state machine with only one transition. The machine makes use of a clock  $x$  that control the next transition. When a transition takes place, the clock is reset. The resulting automaton is shown in figure 6.2.

Figure 6.2: An automaton of the controller. A one-state machine with the corresponding transition using one clock.



Input variables:  $\varepsilon$  ( $\mathfrak{R}$ )

Output variables:  $\rho_e$  ( $\mathfrak{R}$ )

Internal variables:  $x$  ( $\mathfrak{R}$ ),  $\min_x$  ( $\mathfrak{R}$ ),  $\max_x$  ( $\mathfrak{R}$ ),  $\Delta\varepsilon$  ( $\mathfrak{R}$ ),  $\varepsilon_0$  ( $\mathfrak{R}$ ),  $\delta$  ( $\mathfrak{R}$ ),  $\alpha$  ( $\mathfrak{R}$ ),  $\text{hist}_\varepsilon$  ( $\mathbb{N} \rightarrow \mathfrak{R}$ )

States: There is one state

Transitions: See figure 6.2

Equations:

$$\forall_{i \in \{0, N\}} [\text{hist}_\varepsilon(i) = \varepsilon]$$

$$\rho'_e = \rho_e + f(\varepsilon, \varepsilon_0, \alpha)$$

$$f(\varepsilon, \varepsilon_0, \alpha) = \frac{\varepsilon \cdot \alpha}{\text{eff}(\varepsilon, \varepsilon_0)}$$

$$\text{eff}(\varepsilon, \varepsilon_0) = \left| 1 - \frac{\varepsilon}{\varepsilon_0} \right|$$

$$\alpha = \text{slope}(\text{LSF}(\text{hist}_\varepsilon))$$

Now it is possible to unite the different pieces into one hybrid automaton. The last part, the photodiode, is not separately defined because of its simplicity. The photodiode is a one-state automaton that has a continuous analogue output (the derivative of the input). As is shown in figure 6.3 the hybrid automaton of the mode-lock consists of four independent, smaller hybrid machines. This reflects the physical situation in which the four machines are indeed independent devices. The arrows between the different machines represent the wires that connect them. There are two signals that come from outside the system. There is a disturbance  $d$  that influences the working of the laser itself; no control of any kind is possible on the value of this disturbance. The other signal is the reference signal originating from the wavetek. This signal finds its way to the lock-in to be used in the Fourier analysis. Finally, within each box of the model the local variables are shown.

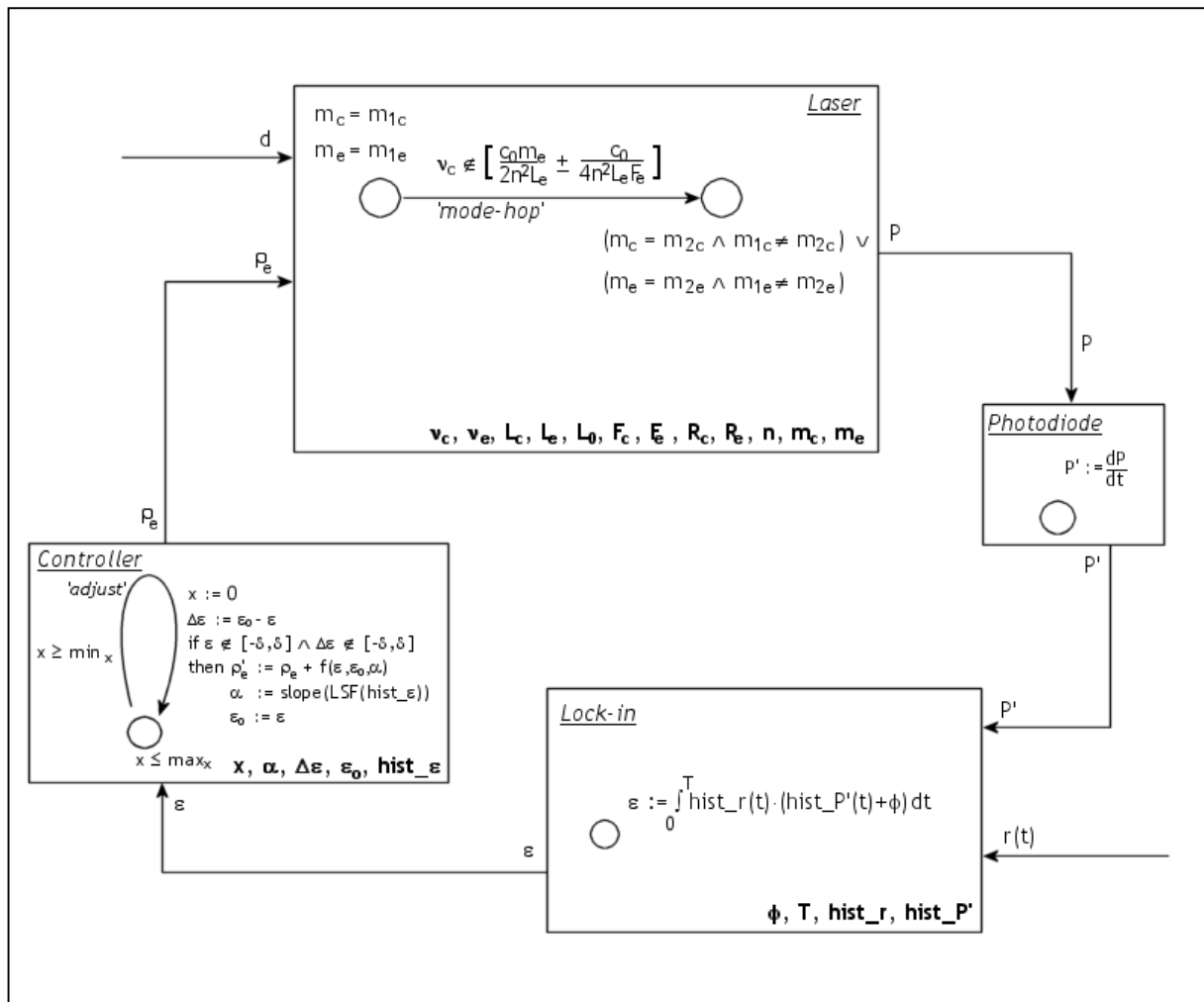


Figure 6.3: The complete mode-lock automaton. The bolded texts are the local variables.

## 6.4 A correctness assertion

The most important issue during a mode-lock is, of course, the prevention of a mode-hop. The automaton as described in the previous section needs to enforce this behaviour. The unwanted transition is the transition in the laser model as described in [section 3.5](#). When too much time passes where the centre frequency of the cavity mode can drift away from the centre frequency of the selected etalon mode, the laser-system may make a mode-hop. To detect these drifts a jitter is placed on one of the etalon mirrors. This results in small movements of the transmission peak of the etalon. When this movement is too big and the transmission peak of the etalon becomes too far from the cavity mode, a mode-hop also may occur.

We can formulate this condition in a mathematical expression. The idea behind the expression is that at every moment in time  $t$  the mode-hop-free-condition will hold. This means that the frequency of the laser must lie within the range as expressed in formula 6.2. The correctness assertion states for all reachable states:

$$\nu_c \in \left[ \frac{c_0 m_e}{2n^2 L_e} \pm \frac{c_0}{4n^2 L_e F_e} \right] \quad (6.2)$$

The question is: for which values of the parameters does the correction assertion hold? It is shown in formula 6.2 that the length of the etalon ( $L_e$ ) plays an important role in the condition. When the jitter, which is placed on one of the mirrors of the etalon, is too large the length of the etalon can be too small at some point in time and a mode-hop may occur. The disturbance  $d$  that is introduced in [section 3.5](#) is countered by changing the length of the etalon. With an increase of the refractive index the distance between the etalon mirrors is decreased (and vice versa). When the disturbance is too large the system may not be able to react fast enough and a mode-hop may occur.

Because there is very little experimental data gathered it is very difficult to give indications on the range of the parameters. Further research is needed to create a clear view on the boundaries of the parameters.



# Chapter 7 De scans

Several types of scans are possible when conducting an experiment. Different ways of scanning a light spectrum of a specimen give the researcher different information. The user has the opportunity to scan a broad area, to get a quick but inaccurate reading on the spectrum; a *grating scan* is used to perform the latter. When a more precise reading on the spectrum is wanted a *mode-hop scan* is used. Other ways of accurate scanning are the *mode-hop-free scan* and the *pump scan*. This chapter includes the description of the previously named scans and explains which parts of the laser-system are used to perform these scans.

Scans are used to gather information on the light spectrum of the specimen. Every substance has its own characteristic absorption spectrum. When a specific frequency of light is within this absorption spectrum the light will not pass the specimen. A scan is a systematically controlled sweep along a particular part of the light spectrum. When measuring which frequencies are absorbed, the absorption spectrum of an unknown substance can be established. Comparing this spectrum with known absorption spectra give the scientist information on the composition of the unknown substance.

## 7.1 Grating scan

The grating scan is not a real scan in terms of gathering experimental data on the specimen. The grating scan is used to roughly set the laser in a specific area of the frequency spectrum. The laser can operate on a very large area of the total light spectrum, but the other (more precise) scans can only scan within a particular small part. With the grating scan the particular area is set.

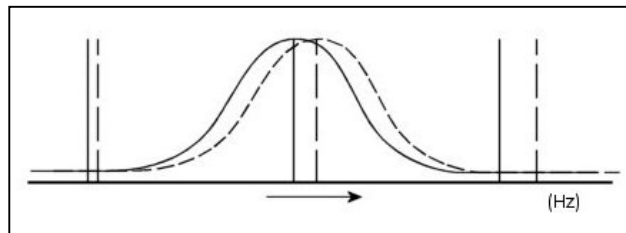
There is a difference in performing a grating scan on the two specific laser-systems. In case of the FC-laser the grating can be turned using the stepper-motor. As we have seen in figure 3.3 the grating has its own transmission peak. By turning the grating this transmission peak is moved. In case of the OPO-laser the same effect can be reached by moving the crystal perpendicular to the laser beam. The moving of the crystal is established by using the stepper-motor. The crystal is in such way constructed that moving it has the same result as turning the grating in the FC-laser (see for further reading [2]). Despite the fact that the OPO-laser uses the crystal to perform this functionality we keep using the term *grating scan* in both laser-systems. Because both the crystal as the grating is controlled due to a stepper-motor, we do not have to distinguish the two laser-systems.

## 7.2 Mode-hop scan

The mode-hop scan is used to acquire data points on the frequency spectrum of the specimen. With the mode-hop scan the cavity length is locked. To scan the spectrum the distance between the etalon mirrors is changed. By doing this the transmission peak of the etalon moves and shall coincide with the following cavity mode (see also [section 3.4](#)). In normal circumstances the selected mode will stay active, despite the fact that it is not situated in the centre of the etalon transmission peak anymore. To select the next cavity mode a chopper is used to turn the laser "*on and off*". The chopper is a rotating disk with holes in it. Whenever a hole passes the laser beam the laser is turned *on*, otherwise it is

turned *off*. To get the maximal transmission for the desired cavity mode the transmission peak of the grating is also moved (see figure 7.1).

Figure 7.1: Performing a mode-hop scan. The grating transmission peak is visible with three (simplistic) etalon modes.



The mode-hop scan is very quick and it has a relatively large range (in contrast of the mode-hop-free scan). The drawback is that not every frequency of the spectrum is scanned. Because the laser hops from cavity mode to cavity mode the frequencies that lie between them in the free spectral range are skipped. It depends on the size of the free spectral range and the goal of the user, whether this is desired.

### 7.3 Mode-hop-free scan

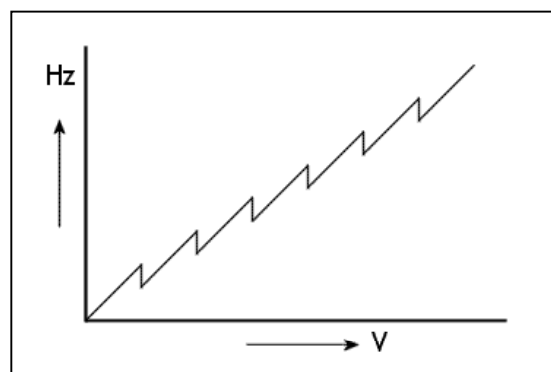
The mode-hop-free scan is also used to gather information about the frequency spectrum of the specimen. As the name implies no mode-hops are allowed when performing this scan. To ensure that the laser does not make such a mode-hop the so-called *mode-lock* is used (see [chapter 6](#) for more details). Moving one of the end mirrors of the laser cavity controls scanning over the frequency spectrum. The length of the etalon cavity can only be set by means of the feedback loop that controls the mode-lock. As mentioned in the pervious section the mode-hop-free scan is slower than the mode-hop scan and also covers a smaller range. The reason for using this kind of scanning is that the whole spectrum is covered.

### 7.4 Pump scan

The pump scan is a scanning technique that uses the pump laser to scan the spectrum. With the two laser-systems in use, only the OPO-laser is fit to use this technique (due to characteristics of the pump laser). When using a pump scan nothing within the laser-system is changed. The frequency of the pump laser is changed and as a result the frequency of the laser-system will also change. It is possible to change the frequency of the pump laser by changing the voltages that controls this laser.

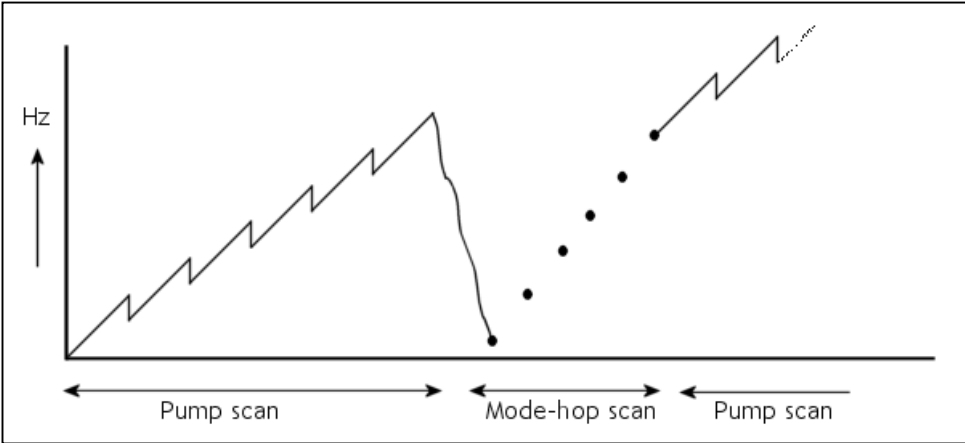
The frequency shift is not steady due to characteristics of the pump laser. After a small shift the frequency falls back to a lower value. This phenomenon is illustrated in figure 7.2 where the voltage on the pump laser is set against the frequency. It is important to the users to realise that there are 'jumps' in the frequency. In after-calculations the user needs to compensate the experimental data for these jumps.

Figure 7.2: The frequency according to the changes in the voltage of the pump laser.



The pump scan can be compared to a mode-hop-free scan, in speed as in range. The prominent advantage of the pump scan is its simplicity. The range of the pump scan is about 40 GHz, but this is not enough to make a complete analysis. Therefore a combination of the pump scan and a mode-hop scan is used to increase the range of the scan (as illustrated in figure 7.3). Now a larger area of the spectrum can be scanned without making a lot of changes on the laser-system.

Figure 7.3: A combination of the pump scan and the mode-hop scan.





## *Part 4: The implementation*



# Chapter 8      LabVIEW

In this chapter I will explain and discuss the use of the graphical development language known as LabVIEW. This language exists only of objects that can be wired to each other. The objects are called *virtual instruments* and good use can make the program clearly and understandable. In chapter 9 the program itself will be explained.

## 8.1 Graphical programming

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a development workbench that is based on the graphical programming language G. LabVIEW is capable of communicating with different hardware to gather experimental data and to control many different sorts of processes. Furthermore, LabVIEW has many built-in libraries that make it possible to use different software standards such as TCP/IP and ActiveX. LabVIEW has a 32-bit compiler to create fast applications; additionally it is possible to create stand-alone version of the software.

Because LabVIEW is completely graphical orientated the programming consists of placing graphical symbols (*virtual instruments* or just *vi*) for the user-interface and the instructions that lie behind them. Furthermore, LabVIEW offers the basics of many other workbenches. It is possible to define breakpoint instructions to debug the program and there is an extensive help. Everything is completed with a huge amount of examples that come with the workbench itself. Finally it is possible to simulate the program without wiring it to the physical world.

Using a graphical programming language has its advantages and disadvantages (just like any other programming language). The chance of endless debugging as a result of a minor type error is drastically reduced. It is still possible to make simple mistakes by wiring the wrong objects to each other or by using the wrong object, but these mistakes are much easier to spot and less easier to make. The use of a simulator is a very powerful instrument of the workbench. Especially when used by instrument makers it is indispensable if this functionality should be missing. Instruments that are used in physical experiments are in general very sensitive and a wrong or too powerful signal could easily destroy such a device; a computer crash is one of the least problems a physicist has.

One major disadvantage of graphical programming is its so-called ease of use. This ease of use makes it comparatively simple for inexperienced programmers to quickly write a computer program, without possessing the in-depth knowledge of the exact semantics of the programming language. This unfortunately, makes it rather tempting to start writing a new program without planning in advance. Obviously, the result of such an approach is a system that is very hard to understand and, for all practical purposes, difficult to maintain.

A major advantage is that dataflow diagrams can be easily copied into the 'program-code'. So is it possible to use the diagrams, which may result from a modelling phase, directly into the program.

## 8.2 The use of vi's and sub-vi's

As mentioned in the previous section a program consist of one or more *virtual instruments* (vi's) that can be wired together. Such a vi consists of a front panel (figure 8.1) and a back panel which is also known as a block diagram (figure 8.2). The front panel is used to construct the user interface combined with all the local variables the program needs. Objects on this panel can be invisible so the user will only see what he is meant to see. Common objects that can be found on a front panel include buttons, switches, editing boxes, numerical input boxes, diagrams and indicators.

As an example the '*Parse Arithmetic Expression*'-vi is used. The description of this vi reads:

This somewhat more advanced example vi shows how to use the *Scan String for Tokens* and *Scan From String* primitives to write an infix arithmetic expression parser.

To use this vi we need to push the run-button. We are then prompted to insert an arithmetic expression. The result of this expression will automatically appear in the lower indicator as the 'Result'.

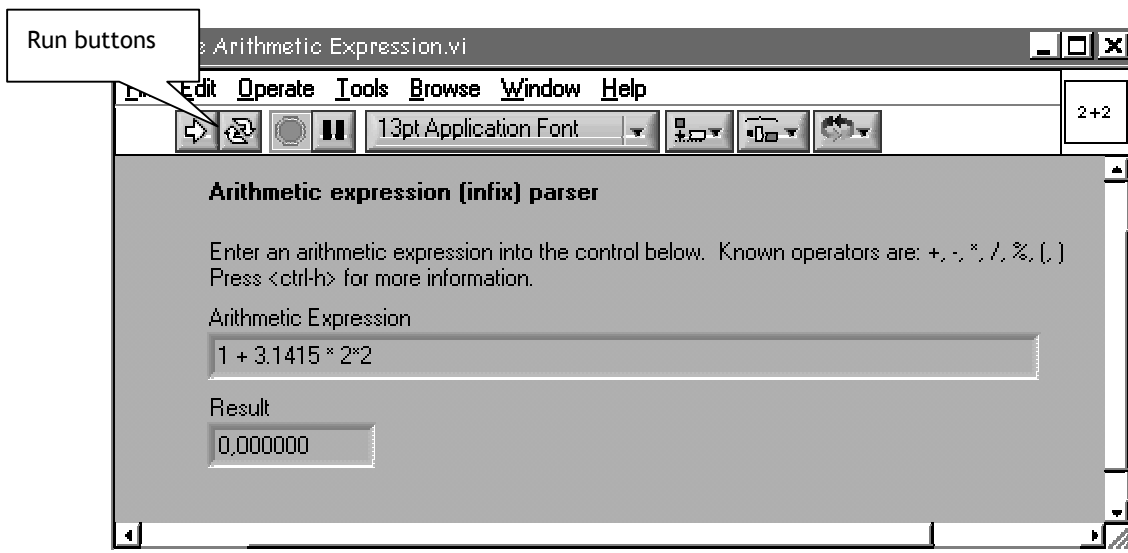


Figure 8.1: An example of a front panel. Two boxes are shown, the above is an input box and the lower is an indicator.

The block diagram is used to describe the actions that a given front panel should perform. The block diagram can be viewed as a dataflow chart; every object in the diagram waits for all its input values before it executes itself. Additionally it is possible to define optional input parameters. During the execution of the program the exact execution path is unknown. The only certainties for a programmer are that every parameter of a vi is valuated before executing a vi and that all the vi's are executed before a new iteration is taken.

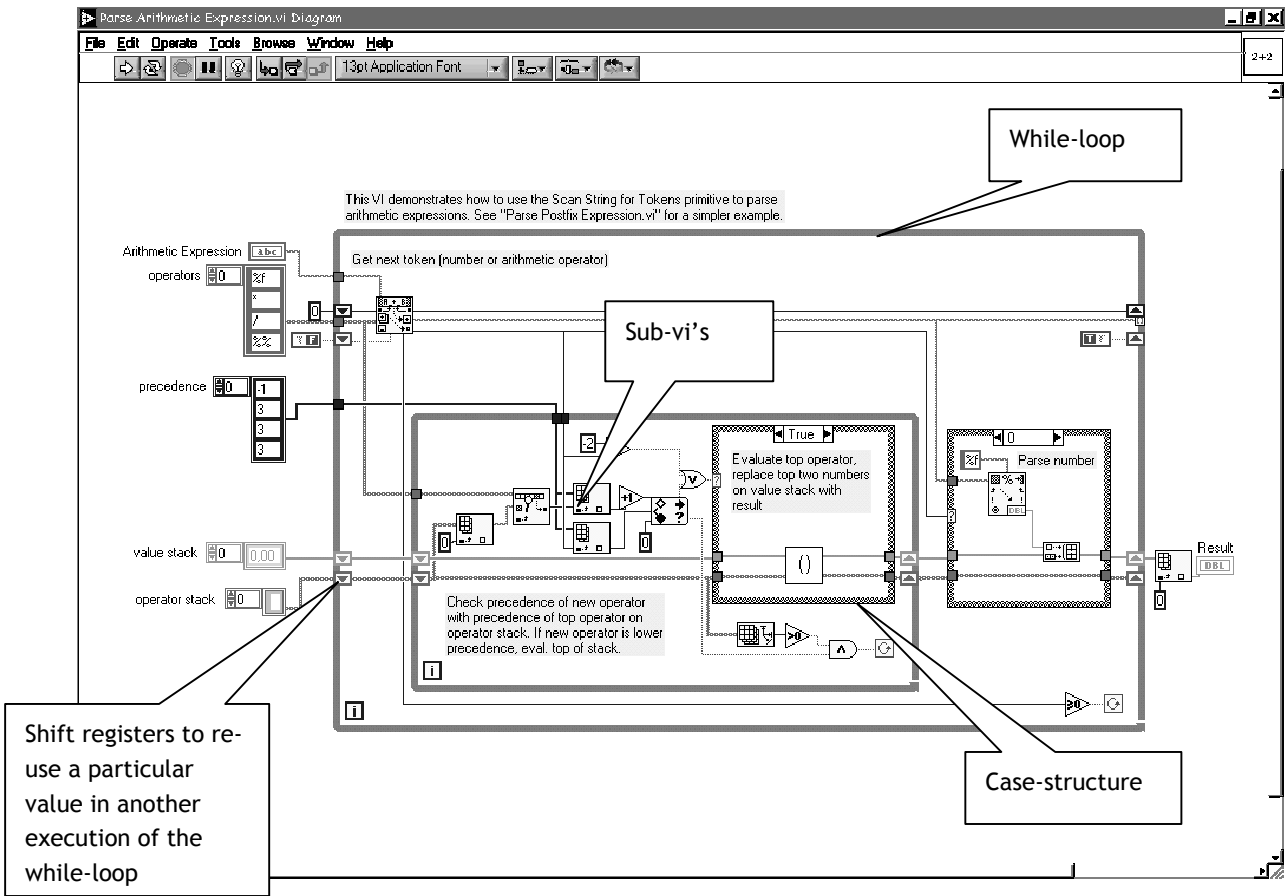


Figure 8.2: An example of a block diagram. The corresponding 'code' of the front panel (figure 8.1) is shown.

As in any other programming language it is possible to create your own modules, in this case called sub-vi's. A sub-vi is not the same as an object in a regular OO-language, where objects have its own properties and methods. Rather, a sub-vi is similar to a function with the necessary local variables and input parameters. Furthermore, a sub-vi allows the possibility to define more than one output parameter (unlike more classical functions).



# Chapter 9 The program

This chapter will explain in further detail some parts of the program. First a brief description of the user-interface is given to give the reader an idea on the looks of the program. The next section holds the code for the mode-lock. Different parts of the code are explained and where possible a link is made with the hybrid model. The last section covers the scans that are implemented in the program.

## 9.1 The user-interface

The user-interface consists of four panels that each has their own functionality. The first panel (figure 9.1) is the start-up panel where the user controls the next action. From here three different actions can be taken. The user has the ability to make a scan, to set the calibrations for performing a scan, or the laser-system can be tested. The use of different panels gives the user a clear overview of the functionalities of the program. The user is not overwhelmed by a large amount of buttons, input-boxes and waveform charts. It is possible to stop the program at all times by pushing the 'Stop program'-button (situated in the low-right corner of the program). In the low-left corner the 'Enable mode-lock'-button is situated (more on its functionality in [section 9.2](#)).

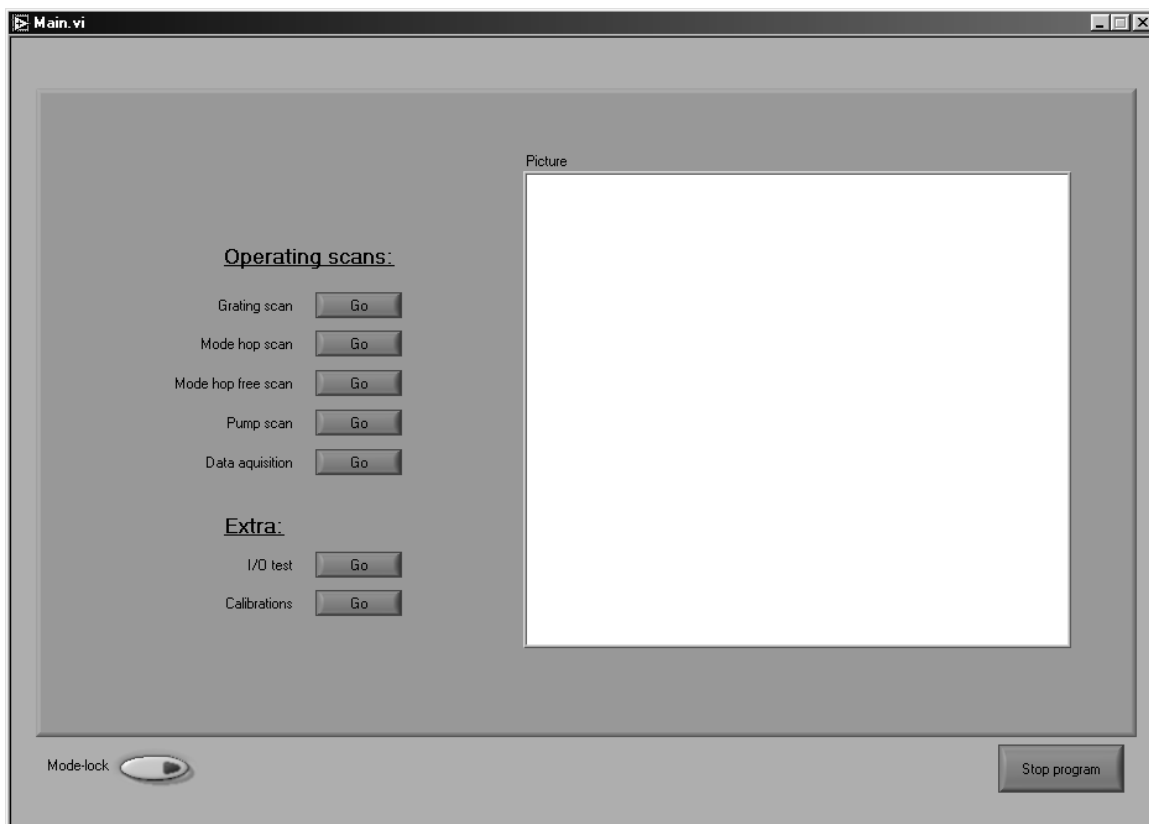


Figure 9.1: Start-up screen of the program. The buttons on the left give access to other parts of the program.

Before performing a scan, the user has to pre-calibrate the different sensors. This gives the user the opportunity to scale a particular signal relative to another, or give it an offset so that, for instance, two signals do not overlap. The calibration page is shown in figure 9.2; there are seven signals that can be scanned, six predefined and one free. The page is also used to set up other variables used throughout the application.

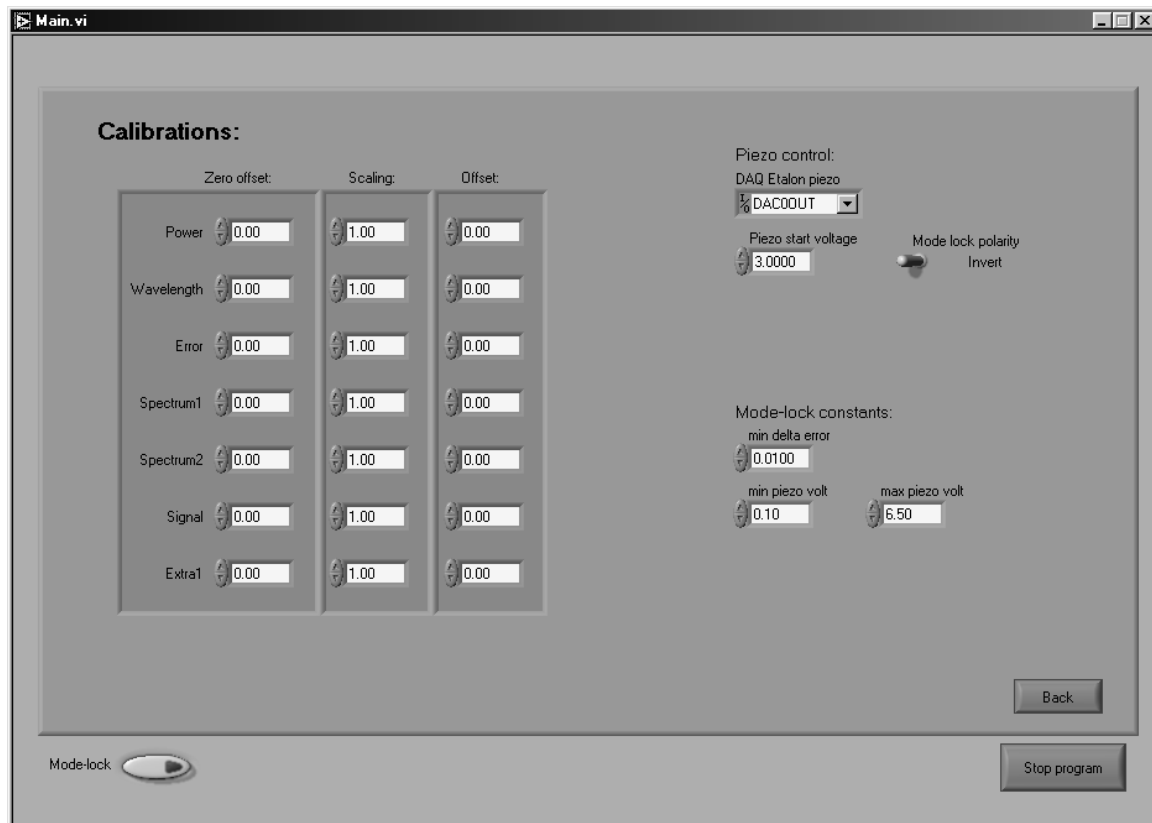


Figure 9.2: Calibrations and set-up screen of the program.

There are, as explained in [chapter 7](#), four different scan options. The program offers to perform these scans (grating scan, mode-hop scan, mode-hop-free scan and pump scan) and the possibility to acquiring data from the laser without performing a particular scan. Within the screen (figure 9.3) it is possible to select a base for the waveform chart (time is the default here). It is also possible to visualize only a subset of the signals. When the user is finished with a scan the program offers the opportunity to store the acquired data into a text file. At the time of finishing this thesis only data acquisition is possible. Due to the absence of a piezo on one of the end mirrors of the laser cavity and a setback on the control of the stepper-motor, it was impossible to implement the functionality of the scans. In [section 9.3](#) more details are discussed on the implementation of the scans.

The test page can be used to test the different actuators and check if the sensors work correctly. The test screen is shown in figure 9.4.

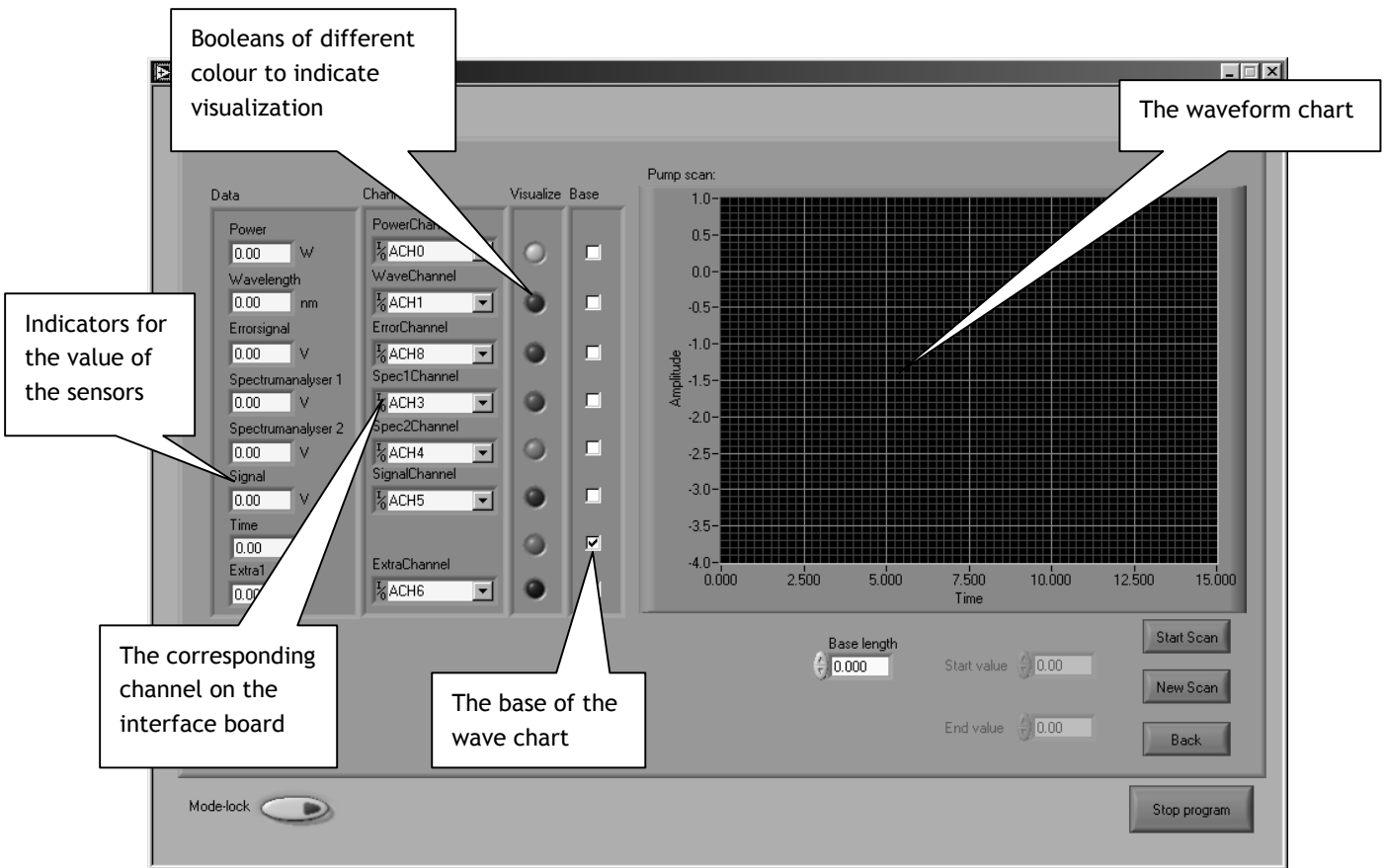
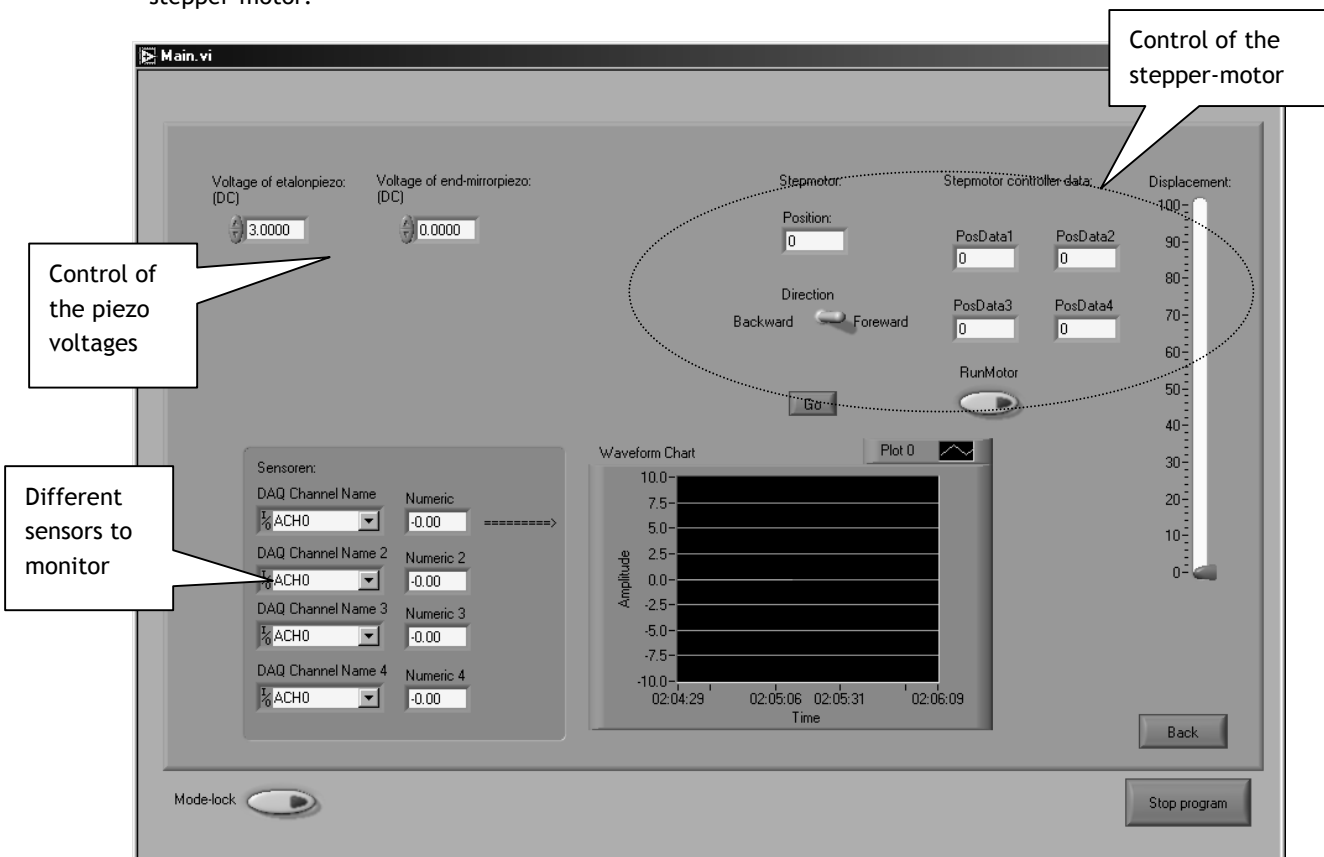


Figure 9.4: Test screen with controls to the piezo's and stepper-motor.

Figure 9.3: General scan interface. This page is used for all kinds of scanning.



## 9.2 The mode-lock

The mode-lock is a separate process within the program. The process is divided into three different steps: the initialisation, the main loop and the finalization. During the initialisation (figure 9.5) that takes place during the initialisation of the entire program, the variables receive their initial value. Some of the variables receive values that were stored during a previous execution of the program (see finalization).

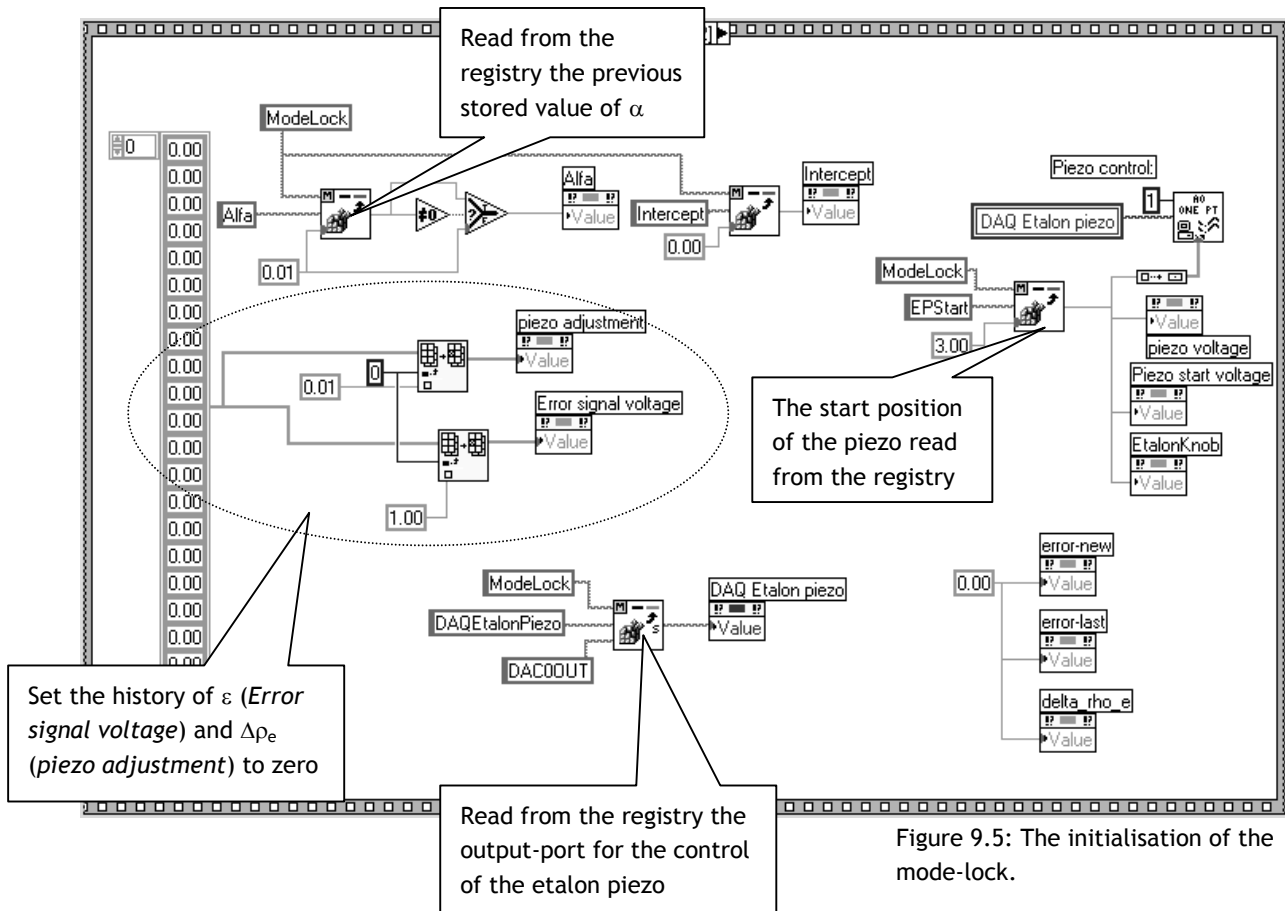


Figure 9.5: The initialisation of the mode-lock.

The core of the mode-lock consists of one big while-loop that is performed every  $T$  milliseconds ( $\min_x \leq T \leq \max_x$ ). The computer reads during the iteration the value of the error signal ( $\varepsilon$ ) from the corresponding input-port. When the new value is not too small and it differs enough from the previous value ( $\varepsilon \notin [-\delta, \delta] \wedge \Delta\varepsilon \notin [-\delta, \delta]$ ) a voltage difference is calculated ( $f(\varepsilon, \varepsilon_0, \alpha)$ , with  $\varepsilon_0$  as the previous error value). The outcome of the function is added to the known voltage of the piezo ( $\rho_e$ ) and passed to the corresponding output-port. This check is shown in figure 9.6. Most of the work is concentrated in the sub-vi that calculates the voltage difference (the function  $f$ ) and performs the least square fit (the function  $BF$ ). In figure 9.7 is shown how both functionalities are combined.

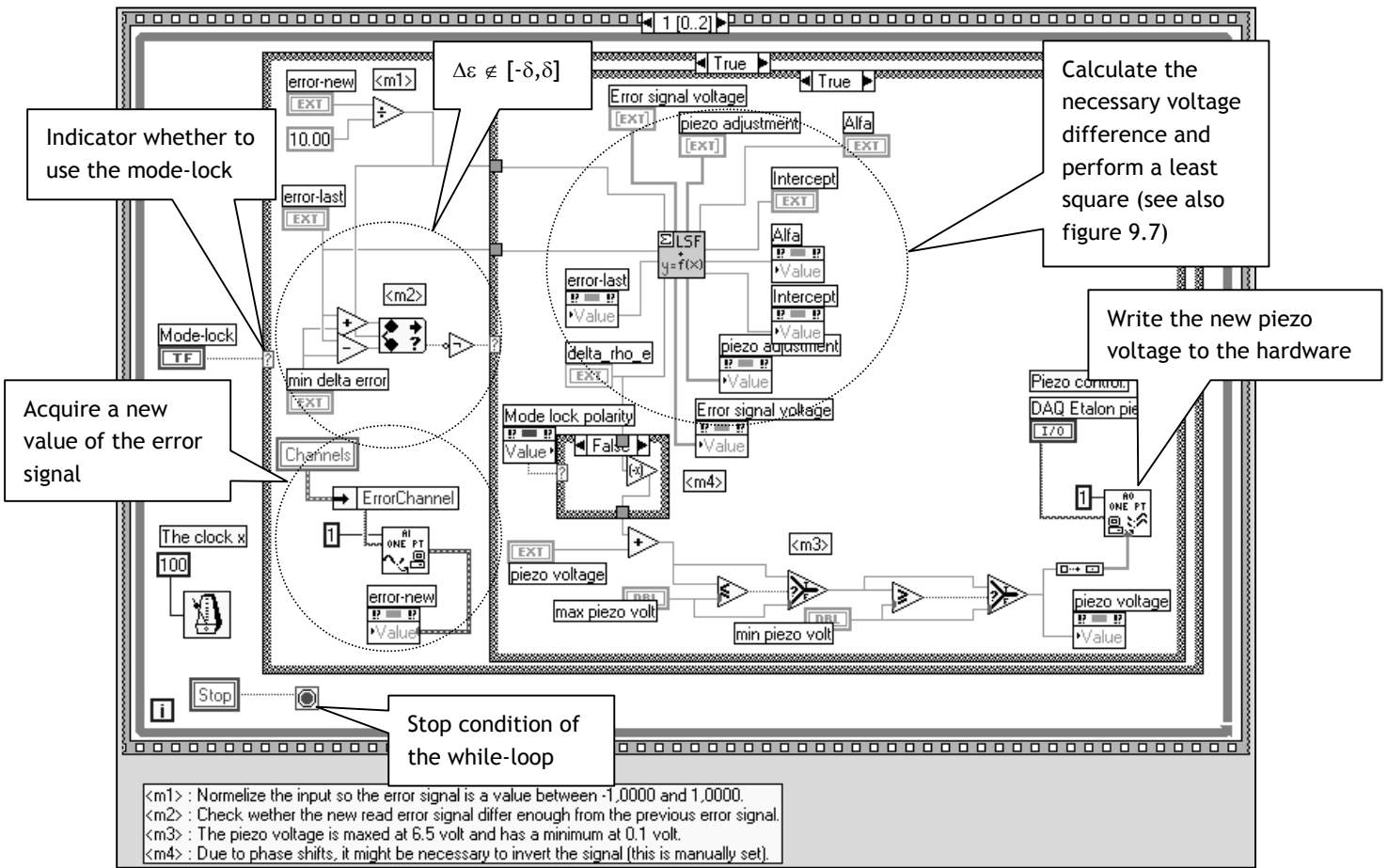
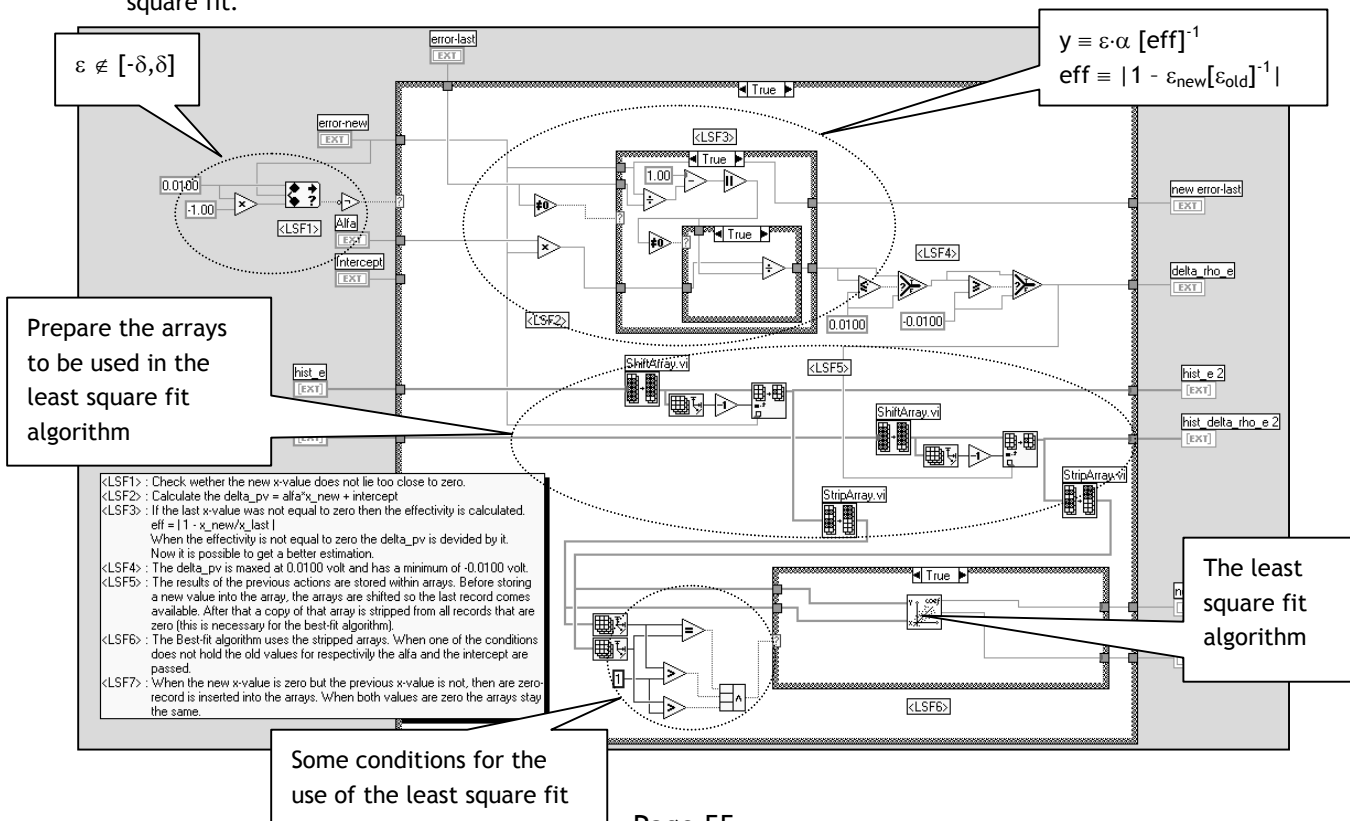


Figure 9.6: The core code of the mode-lock.

Figure 9.7: The calculation of the voltage difference and the least square fit.



When the main loop is terminated the last step is taken; the program waits until all the activities within the loop are finished. Finalizing the mode-lock process corresponds with the storage of some values in the registry (see figure 9.8).

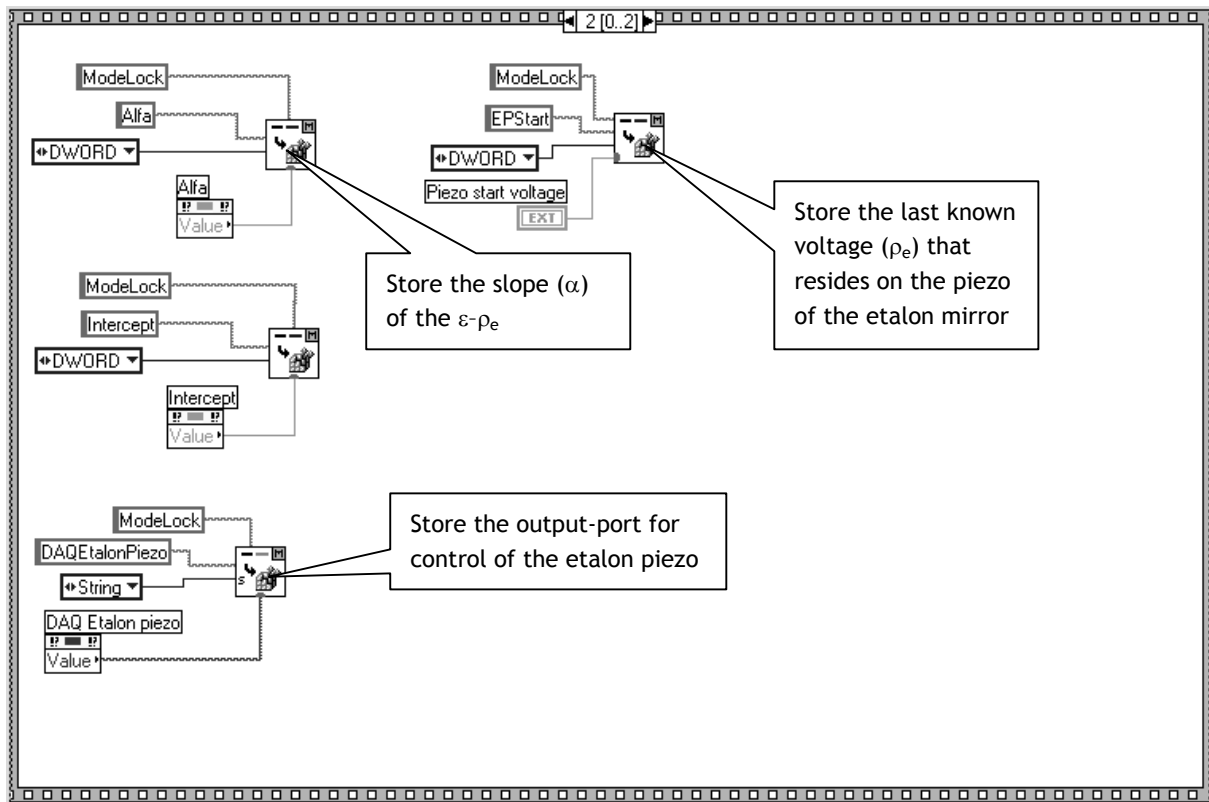


Figure 9.8: The finalization of the mode-lock process.

### 9.3 The scans

For the sake of example only the general data acquisition is explained here, other kinds of scanning is very similar to this case (and not implemented yet as explained in the beginning of this chapter). The process of scanning the lasers is divided into three steps: setting up the laser-system, collecting the sensor data and visualizing the collected data into a waveform chart. Actions that take place during the first step are moving the grating, the end mirror of the laser cavity and/or the mirror of the etalon cavity. The other two steps are shown in figure 9.9.

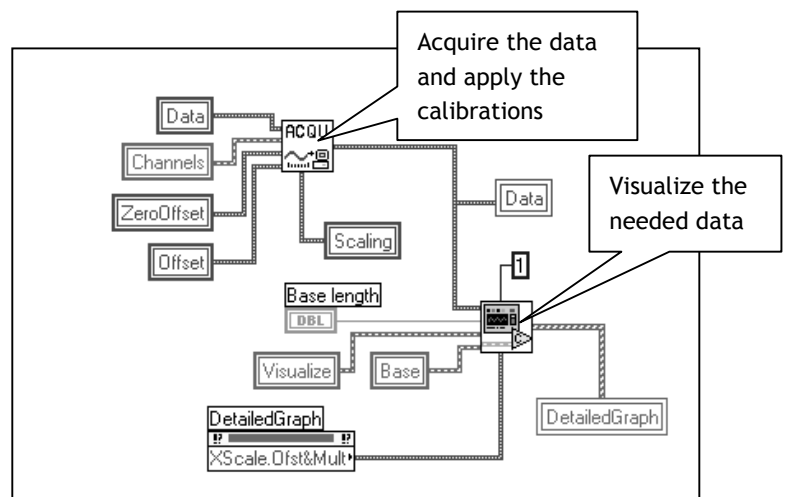


Figure 9.9: The code for acquiring data from the sensors.

The second step, acquiring the sensor data, uses only one sub-vi. This vi reads a particular channel and returns the corresponding value. The rest of the code deals with the application of the calibrations (see figure 9.10). The last step is to visualize the data into a waveform chart. The user has specified which of the eight input channels has to be visualized and which one acts as the base of the chart. In figure 9.11 shows some of the code, more specifically the last part of the visualization process. On the left-hand side are the eight values that were acquired in the previous step. The top line is the base value. When the user does not want to visualize a particular value it is reset to zero. All the values (including the reset ones) are stored in the data object again. The rest of the code is used to format the data in the right form to be used for the waveform chart.

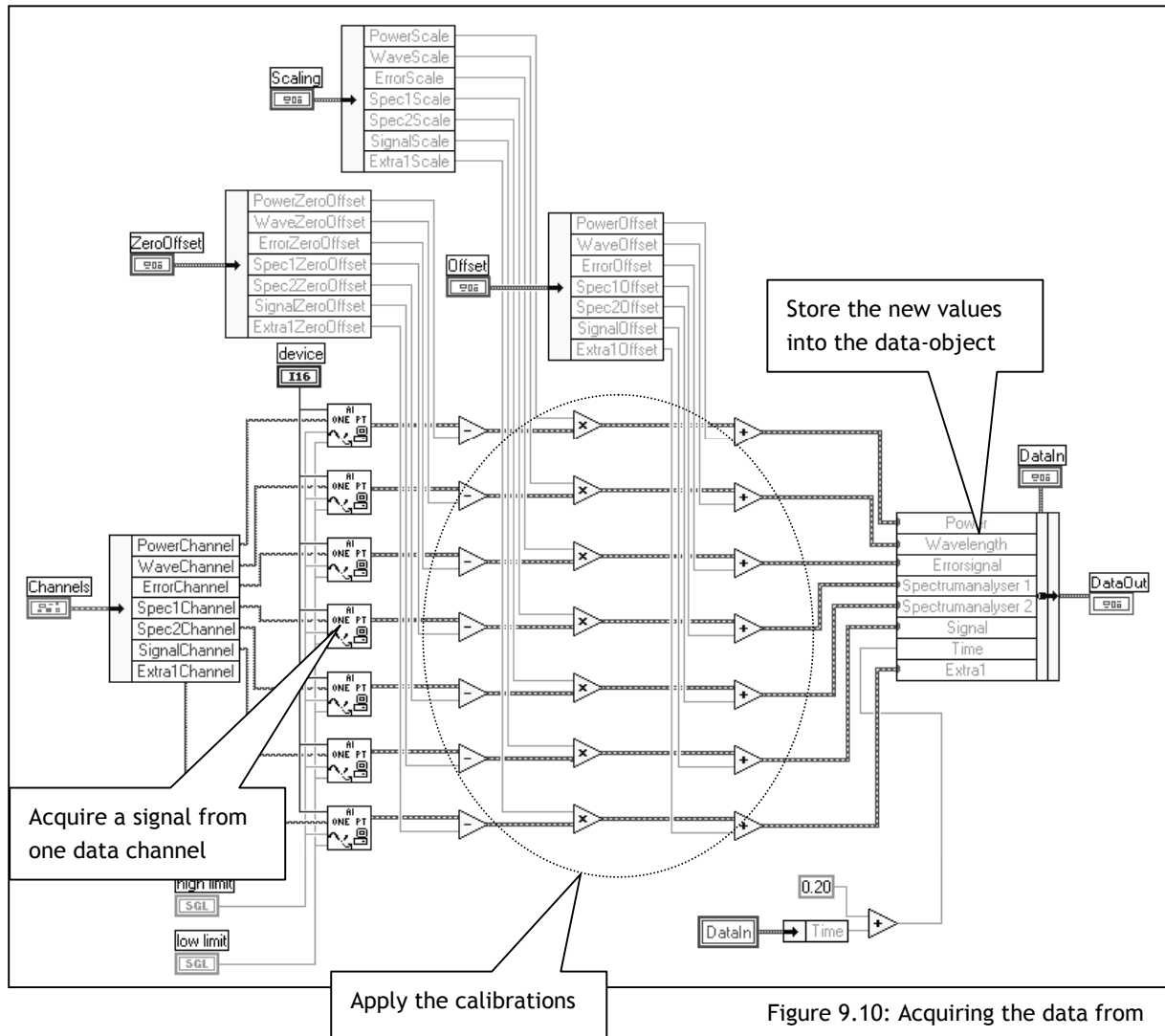


Figure 9.10: Acquiring the data from the input channels and apply the calibrations on them.

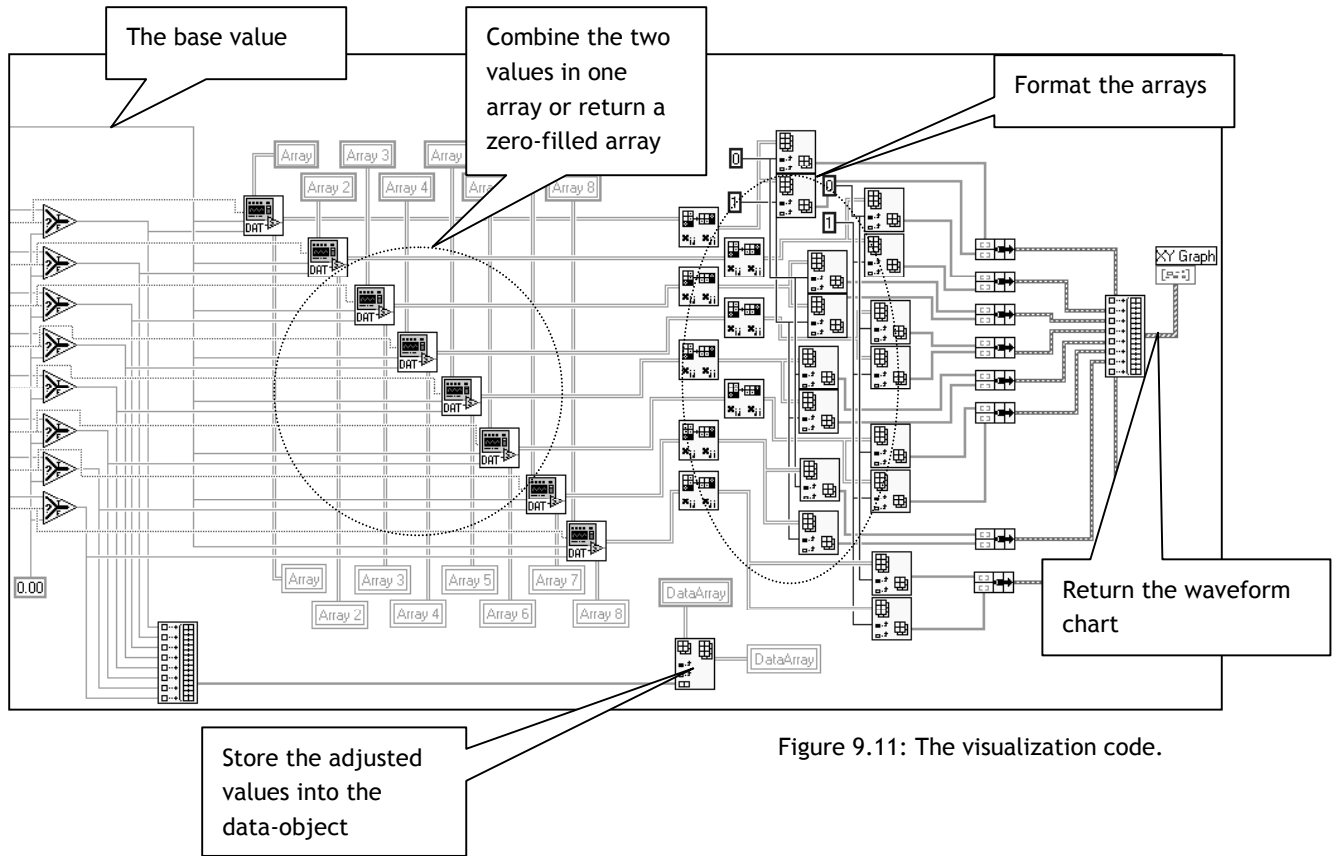


Figure 9.11: The visualization code.

# Chapter 10      Conclusions

This final chapter contains the conclusions that I have drawn during the execution of the project. The making of the hybrid automaton model and the usability of this model in the implementation are the main subjects of this chapter. The use of LabVIEW will also be discussed in combination with the hybrid automaton. This chapter ends with a final thought on the whole project.

## **The automaton model**

A hybrid automaton model of the mode-lock process is made as a result of the project. For this four separate hybrid automatons are developed: the laser, the photodiode, the lock-in and the controller (the computer program). The following conclusions can be drawn concerning the creation of the automaton model:

- Modelling a laser-system is not to be underestimated. The many dependencies within the system make it very hard to discover the essential equations for the model.
- It is possible to construct a solid hybrid automaton model of a process that is present within a laser-system. Modelling the mode-lock process gives us insight in the ability to build such a model. The hybrid automatons are powerful enough to encapsulate the continuous analogue behaviour of the laser-system.
- The complete model shows clearly the dependence on the disturbance that can influence the system.
- A correctness assertion is formulated that can be used in future work to prove that the automaton model is correct.

## **The implementation**

During the execution of the project a rapid development strategy is followed. The project consists of three phases. The first phase is used to develop the user-interface and the control of the actuators. After the modelling phase the rest of the program was implemented. The following conclusions can be drawn concerning the implementation of the computer program:

- The user-interface had more work to it than initially was accounted for. The consequence was that less time was available for the realisation of the rest of the program.
- Computer aided control of actuators is a major risk factor in the execution of a project. This is shown with the control of the stepper-motor that, at the time of writing this thesis, still does not respond to the data it is receiving. Another setback was that the error signal from the FC-laser appeared to be unstable, which resulted in an inaccurate response of the controller. Due to this bad error signal a mode-lock on the FC-laser with the new software was no longer possible. For this reason we needed to concentrate our efforts on the OPO-laser which was not operational at that time.
- Because not all the actuators were installed in the OPO-laser (the piezo on the cavity end-mirror was missing) most of the scans were not realised.
- Acquiring the data was easy in contrast to the visualization of the collected data.

- The conditions on the value of the error signal for making adjustments to the voltage of the etalon piezo were easily integrated. The identification of these conditions during the modelling phase was of great value for the realisation of the mode-lock.

### The programming language

The following conclusions can be drawn about the use of LabVIEW as programming language and how to make programs that are clear and easy to read:

- Using LabVIEW had both its advantages and disadvantages. At first the language appeared easy to master and very intuitive. But to create a good and stable user-interface the language appeared to lack simplicity. The 'code' rapidly became a bush of wires and the total overview grew out of the window size.
- For implementation of the program, more sub-vi's were used. This made the program as a whole more viewable. However, using sub-vi's has one great disadvantage: a sub-vi has only input and output parameters. This means that for every variable that is used as input and receives a new value after execution, the sub-vi has a pair of variables to implement this behaviour.
- It was easy to implement the hybrid automaton within the program. But, as was said before, the major difficulties lay in the action of the transition. Comments about the program can be placed within the code using text boxes. The code can be placed within different coloured boxes for easy referencing. Altogether LabVIEW is good enough to make clear and readable code.

### Final thought

*Which advantages or disadvantages does the use of a hybrid model have when developing real-time software used in controlling a laser-system?*

The above question was the subject of this master thesis and I think a satisfying answer has been given. The major advantage of building such a model is creating a better understanding of the processes at work. Questions on the use and working of the laser-system and the program, which normally arise much later, were now in an early stage confronted. To give a full and complete answer to this question, a more thorough exploration of the model should take place. For this the correctness assertion from [chapter 6](#) should be proved and a simulation of the model can be used to give insight in the boundaries of the many parameters. It is a pity that the time for this project was too short to include proving the assertion and run the simulations.

## *Part 5: Appendix*



# Appendix A1 Document of demands

This appendix holds the document of demands that is created at the beginnings of the project. The document describes the demands of the end user and gives an overview of the functionalities the program should have.

## Terminology

First a few conceptions are discussed to increase the clarity of this document for both the computer scientist as the physicists. These conceptions are important to understand the demands stated in this document.

The program has to operate on two different laser-systems. These laser-systems are known as the OPO set-up (Optical Parametric Oscillator) and the FCL set-up (Farben Center Laser). Both systems contain comparable actuators and sensors. The actuators that are used in both systems are attached to the following parts of the laser-systems (see figure A1.1 and figure A1.2):

1. End mirror: an electric optic piezo is attached to the mirror to determine the length of the laser cavity.
2. Grating/crystal: the other end mirror of the FCL set-up is a grating that can be turned due to a stepper-motor, the OPO set-up uses the stepper-motor to shift the crystal gaining the same effect.
3. Etalon: an electric optic piezo is attached to one of the mirrors of the etalon to determine the length of the etalon.

Figure A1.1: The actuators of the OPO set-up. The numbers in the figure refer to the numbers of the above numeration.

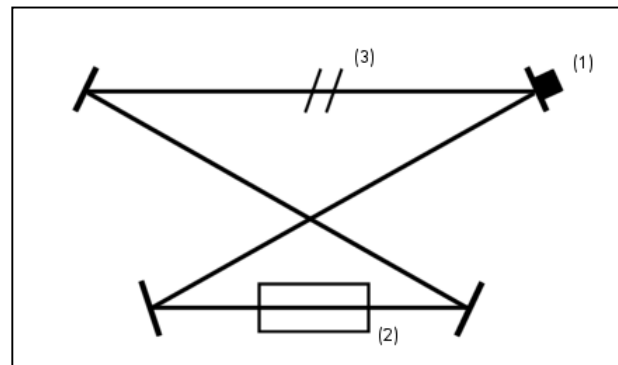
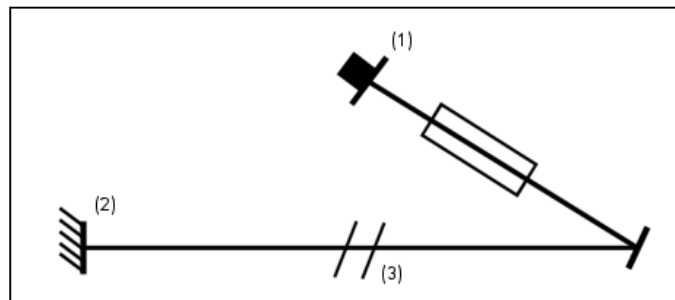


Figure A1.2: The actuators of the FCL set-up.



## Main functionalities

The language used within the program shall be the English language. Many different nationalities work at the molecular physics department; so one common language to communicate with each other is a requirement. Because in the future other people are likely concerned with the maintenance of the program all the names and comments used within the program are to be in English. Finally the system-documentation shall also be in English.

The program must support three kinds of scans that can be used to execute different experiments. A fourth kind of scanning (specific of the OPO set-up) is added as an optional feature.

- Grating scan: Turning the grating over a particular angle performs the scan. This results in a very broad shift of the frequency domain. The same effect can be reached in case of the OPO set-up by moving the crystal. In both cases the stepper-motor is used as actuator. A specific device is used as controller of the stepper-motor that uses an eight-bit stream as input.
- Mode-hop scan: A mode-hop scan can be seen as an etalon scan because the length of the etalon is systematically changed during the execution of the scan. This results in small frequency shifts. Due to characteristics of the laser these shifts will not be continuous but make hops. The grating has to move according to the changes in the frequency to get the maximal output intensity.

The voltage needed to make such a hop will be experimental determined. The value that is acquired can be manually inserted into the program. An optional feature is to automate the program to determine its own parameters.

- Mode-hop-free scan: A mode-hop-free scan can be seen as a cavity scan because the length of the laser cavity is systematically changed during the execution of the scan. This results in small frequency shifts. The grating has to move according to the changes in the frequency to get the maximal output intensity. To prevent the laser making mode-hops a mode-lock is used. This mode-lock (or etalon-lock) is one of the objects of the thesis.

A mode-lock is used with the mode-lock-free scan but is switched off during the grating, mode-hop and pump scan. The mode-lock can at any time be enabled except for the cases mentioned. A reaction of at least 1 kHz has to be made (this means that the program should respond within 1 millisecond).

- Pump scan: This kind of scanning is typical for the OPO set-up due to characteristics of the pump laser. During the execution of the scan only the frequency of the pump laser changes, hence the frequency of the OPO laser changes accordingly. The frequency change is not continuous but contains jumps (see figure A1.3). The program does not have to compensate for these jumps.

The range of the pump scan is approximately 40 GHz. The desired range for a complete scan is however more; the pump scan is often combined with a mode-hop scan to increase the range (see figure A1.4).

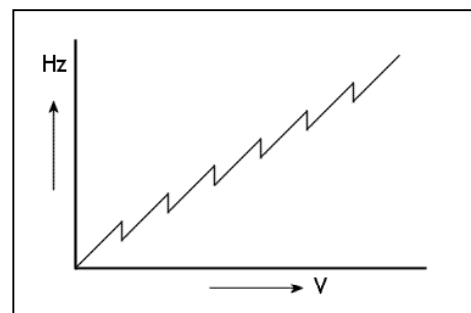


Figure A1.3: The change in frequency of the pump laser due to changes in the input voltage.

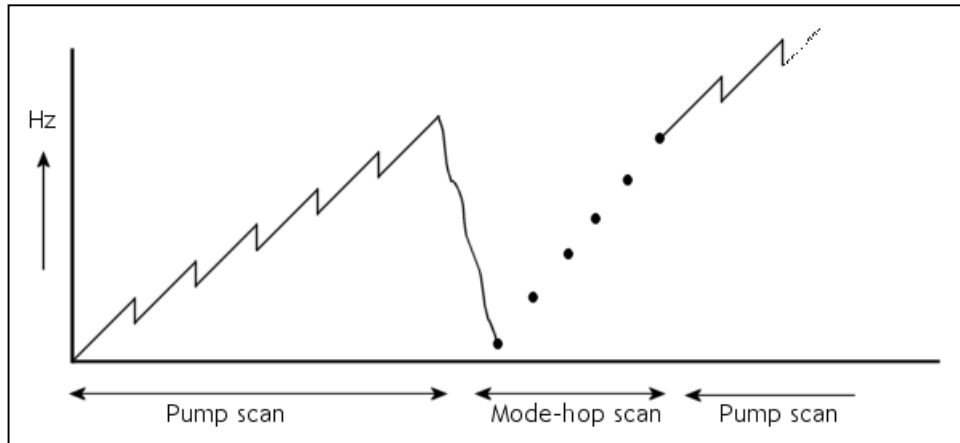


Figure A1.4: A combination of the pump scan and the mode-hop scan.

The acquired data has to be visualized in a clearly arranged window. The following distinct sensor readings have to be visualized:

- The wavelength
- The absorption (a.k.a. signal)
- The output power
- Two spectrum analysers
- The error signal
- A dynamic input signal

For every signal can be determined whether it should be visualized in a waveform chart. All the selected signals are combined in one chart where as default the signal is expanded against time. It is possible to select another signal as the base for the chart. The collected data of an experiment is stored to disk. A computer crash may not lead to a loss of data.

A special use of the mode-hop scan is also known as the analysis scans. There are two ways of performing an analysis scan: a slow scan with multiple data points at a specific frequency or multiple scans with less data points per scan.

There are two different ways to end the analysis scan: a stop button to stop the scan at all times and a timer that goes off after a previously user defined execution time. The time between two scan is also adjustable by the user.

#### **Automatic set-up**

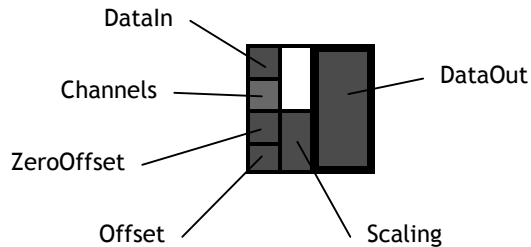
The voltage difference needed on the etalon piezo to shift exactly one cavity mode is, during the start of the project, manually determined. An optional feature is to automate this. The program has to detect a cavity mode and the voltage needed to make one mode hop.

To perform a good spectrum analysis it is necessary to start precisely on a cavity mode. The program needs to be able to set-up the laser-system in such way that only one cavity mode is active. The latter is also an optional feature.



# Appendix A2 System documentation

## Acquire Data From Channels.vi



### Functionality:

Assigns a set of values to *DataOut* that corresponds with the values read from the *Channels* added to *DataIn*. Before these new values are added calibrations are applied ( $\text{new\_value} = (\text{signal}_i - \text{ZeroOffset}_i) * \text{Scaling}_i + \text{Offset}_i$ ). The time is added as eighth array.

### Input parameters:

*DataIn* : Set of eight arrays with data points  
*Channels* : Set of seven channel names  
*ZeroOffset* : The zero offset per channel  
*Offset* : The offset per channel  
*Scaling* : The scaling per channel

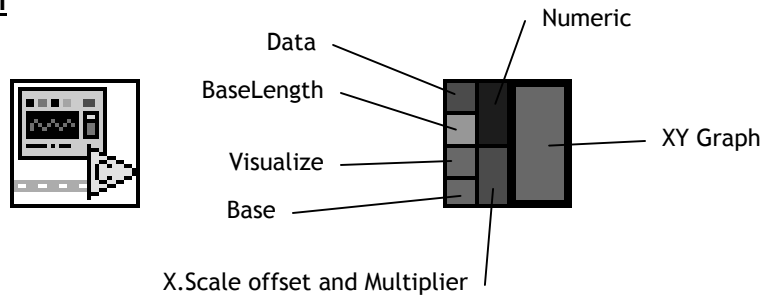
### Output parameters:

*DataOut* : Set of eight arrays with data points

### Used in:

Main.vi

## Calc Waveform.vi



### Functionality:

Given a particular dataset *Data* a waveform chart *XY Graph* is constructed. The construction depends on which of the eight arrays has to be visualized. One of these arrays is used as the base of the other data points. This means that the data points from the chosen array are set on the x-axis and the other arrays on the y-axis. The *BaseLength* and the *X.Scale offset and Multiplier* are experimental parameters to realise a clear overview of the collected data. The *Numeric* parameter is used to determine the functionality of the vi:

- 0 : Initialise the arrays and with that clear the waveform chart.
- 1 : The main functionality as explained above.
- 2 : Write the data points stored in the arrays to a file.

### Input parameters:

**Numeric** : An integer that determines the functionality  
**Data** : A set of eight arrays with the data points  
**BaseLength** : A double with contains the length of the base line  
**Visualize** : A set of eight Booleans  
**Base** : A set of eight Booleans  
**X.Scale offset and Multiplier** : A record with the scale offset and multiplier of the chart

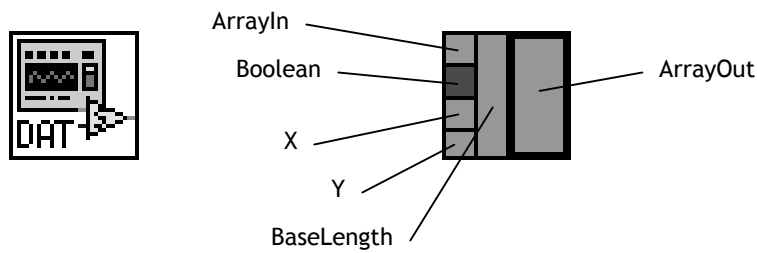
### Output parameters:

**XY Graph** : The chart itself

### Used in:

Main.vi

## Select Data.vi



### Functionality:

The functionality depends on the value of the *Boolean* parameter. When the Boolean parameter is set to "true", the *X* and *Y* value are added to the *ArrayIn* and assigned to *ArrayOut*. The latter is the case when  $X \text{ modulo } \textit{BaseLength}$  is not zero; otherwise *X* and *Y* are the first record in a new array that is assigned to *ArrayOut*. In the case the Boolean parameter is set to "false", the *ArrayOut* is assigned with a zero-record.

### Input parameters:

*ArrayIn* : A two dimensional array  
*Boolean* : A Boolean value which determines the functionality  
*X* : A double for the x-value  
*Y* : A double for the y-value  
*BaseLength* : A double for the base length as explained with Calc Waveform.vi

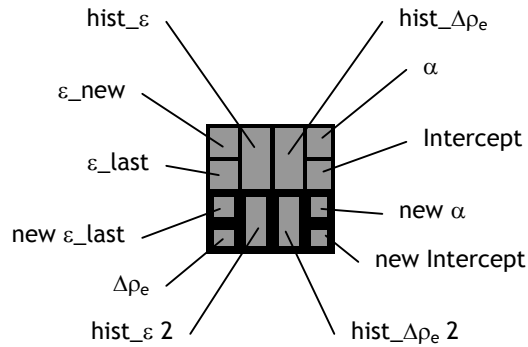
### Output parameters:

*ArrayOut* : A two dimensional array

### Used in:

Calc Waveform.vi

## LeastSquareFit.vi



### Functionality:

Calculates a value for the etalon piezo voltage difference ( $\Delta\rho_e$ ) if it is necessary. The following conditions are evaluated:

- |  |   |
|--|---|
| $\varepsilon_{new} \notin [-\delta, \delta]$                     | → The error signal must be large enough                       |
| $\varepsilon_{old} - \varepsilon_{new} \notin [-\delta, \delta]$ | → The error signal must differ enough from its previous value |

If an adjustment to the etalon piezo voltage is necessary the following calculation is used:

$$\Delta\rho_e = \frac{\varepsilon_{new} \cdot \alpha}{\left| 1 - \frac{\varepsilon_{new}}{\varepsilon_{old}} \right|}$$

The slope of the relationship between the error signal and the etalon piezo voltage difference is symbolised with  $\alpha$ . To get an accurate value for  $\alpha$  a least square fit algorithm is used on the previous results ( $hist_{\varepsilon}$  and  $hist_{\Delta\rho_e}$ ) to acquire new values for  $\alpha$  ( $new \alpha$ ) and the intercept ( $new Intercept$ ) of the relationship. Before execute the least square fit algorithm the value of the error signal  $\varepsilon_{new}$  and the calculated voltage difference  $\Delta\rho_e$  are added to respectively  $hist_{\varepsilon}$  and  $hist_{\Delta\rho_e}$ .

### Input parameters:

- |                       |   |
|-----------------------|---|
| $\varepsilon_{new}$   | : The current value of the error signal   |
| $\varepsilon_{last}$  | : The last value of the error signal  |
| $hist_{\varepsilon}$  | : An array of error signal values   |
| $hist_{\Delta\rho_e}$ | : An array of etalon piezo voltage differences that corresponds with $hist_{\varepsilon}$ |
| $\alpha$              | : The slope of the relationship between $\varepsilon$ and $\Delta\rho_e$                  |
| Intercept             | : The intercept of the relationship between $\varepsilon$ and $\Delta\rho_e$ (not used)   |

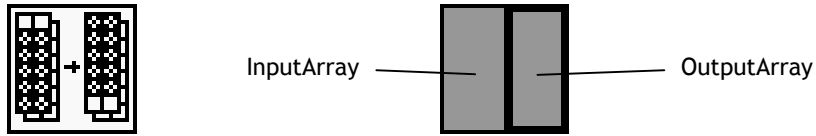
### Output parameters:

- |                          |  |
|--------------------------|--|
| $new \varepsilon_{last}$ | : The current value is stored                  |
| $\Delta\rho_e$           | : The etalon piezo voltage difference          |
| $hist_{\varepsilon} 2$   | : An array of error signal values              |
| $hist_{\Delta\rho_e} 2$  | : An array of etalon piezo voltage differences |
| $new \alpha$             | : The new slope                                |
| $new Intercept$          | : The new intercept (not used)                 |

### Used in:

Main.vi

## ShiftArray.vi



### **Functionality:**

Shift the records within the array:

$$\text{OutputArray}_i := \text{InputArray}_{i+1} \text{ for } i \in [2, \text{Length}(\text{InputArray})-1]$$

The first record is not shifted.

### **Input parameters:**

InputArray : A two dimensional array

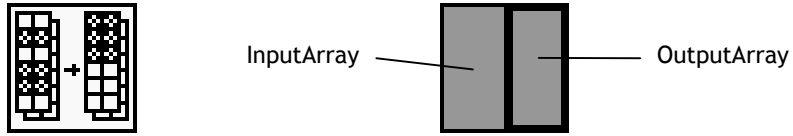
### **Output parameters:**

OutputArray : A two dimensional array

### **Used in:**

LeastSquareFit.vi

## StripArray.vi



### **Functionality:**

Removes all but one of the zero-records within the array.

### **Input parameters:**

InputArray : A two dimensional array

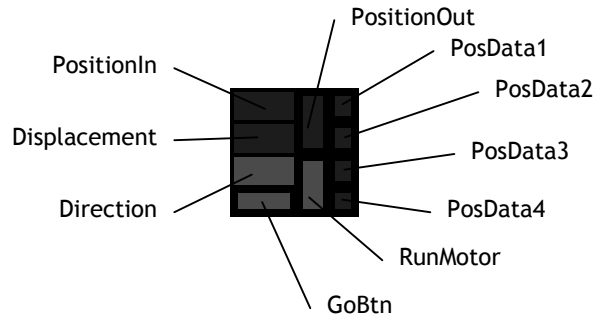
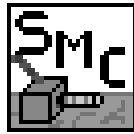
### **Output parameters:**

OutputArray : A two dimensional array

### **Used in:**

LeastSquareFit.vi

## StepperMotorControl.vi



### Functionality:

Controls the outgoing signals to the stepper-motor controller. The *Displacement* is added or subtracted from *PositionIn* depending on the *Direction*. The vi returns the new position of the motor (*PositionOut*). The controller uses BCD to store the digits of the position. To indicate the order of digits that is sent to the controller the vi returns these digits (*PosData1*, *PosData2*, *PosData3*, *PosData4*). Seven steps are used to position the stepper-motor. The *GoBtn* is set to "true" when all the data is sent to the controller and the stepper-motor starts to run (*RunMotor* is set to "true"). The final step is a while-loop that reads the input channel for confirmation that the new position is reached.

The control of the stepper-motor does not work at this point. Due to a very unclear reason the controller does not react to the signals it receives.

### Input parameters:

*PositionIn* : The old position of the stepper-motor  
*Displacement* : A double to indicate the displacement of the stepper-motor  
*Direction* : A Boolean to indicate the direction ("true" = addition)

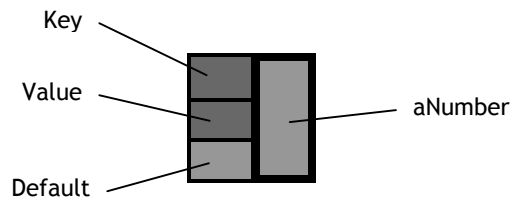
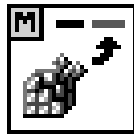
### Output parameters:

*PositionOut* : The new position of the stepper-motor  
*PosData1* : The first digit of the new position  
*PosData2* : The second digit of the new position  
*PosData3* : The third digit of the new position  
*PosData4* : The fourth digit of the new position  
*GoBtn* : A Boolean to indicate that all data is sent  
*RunMotor* : A Boolean to indicate that the stepper-motor is running

### Used in:

Main.vi

## ReadValue.vi



### **Functionality:**

A DWORD is read from the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

When no entry is present within the registry, a new entry is created and the default value is returned.

### **Input parameters:**

Key : The key entry of the registry  
Value : A particular value of the key entry  
Default : The default in case the key or the value does not exist

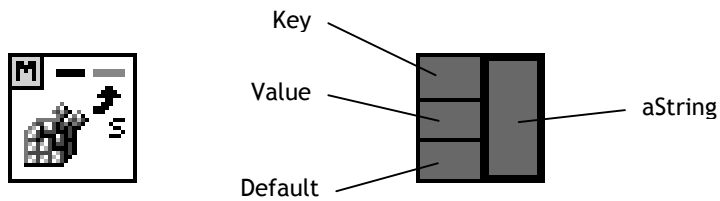
### **Output parameters:**

aNumber : A DWORD is read from the registry

### **Used in:**

Main.vi

## ReadSValue.vi



### **Functionality:**

A String is read from the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

When no entry is present within the registry, a new entry is created and the default value is returned.

### **Input parameters:**

Key : The key entry of the registry  
Value : A particular value of the key entry  
Default : The default in case the key or the value does not exist

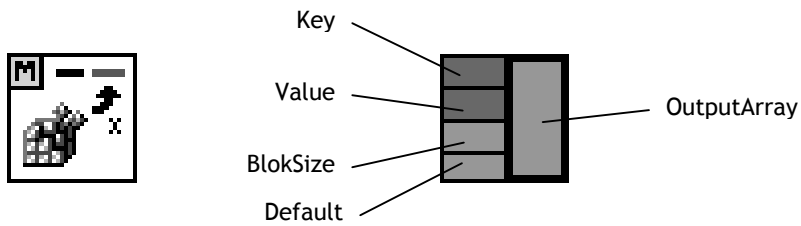
### **Output parameters:**

aString : A String is read from the registry

### **Used in:**

Main.vi

## ReadValueBlok.vi



### Functionality:

An array of DWORD is read from the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

When no entry is present within the registry, a new entry is created and the default value is returned. There are  $x$  values read from the registry (where  $x$  equals *BlokSize*). The name of the value is composed by:

`Value+i, i ∈ [1, BlokSize]`

### Input parameters:

Key : The key entry of the registry  
Value : A particular value of the key entry  
BlokSize : The number of values  
Default : The default in case the key or the value does not exist

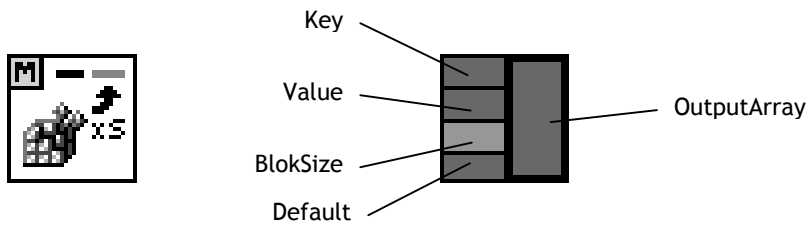
### Output parameters:

OutputArray : An array of DWORD with records read from the registry

### Used in:

Main.vi

## ReadSValueBlok.vi



### Functionality:

An array of String is read from the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

When no entry is present within the registry, a new entry is created and the default value is returned. There are  $x$  values read from the registry (where  $x$  equals *BlokSize*). The name of the value is composed by:

`Value+i, i ∈ [1, BlokSize]`

### Input parameters:

Key : The key entry of the registry  
Value : A particular value of the key entry  
BlokSize : The number of values  
Default : The default in case the key or the value does not exist

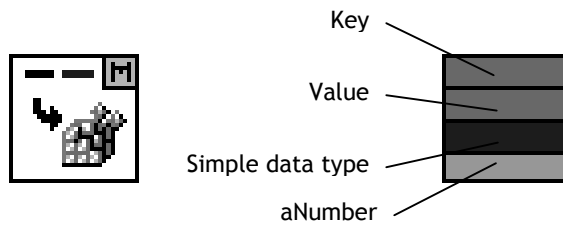
### Output parameters:

OutputArray : An array of String with records read from the registry

### Used in:

Main.vi

## WriteValue.vi



### **Functionality:**

A DWORD is written to the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

The *Simple data type* is a parameter that has always "DWORD" as value. This vi can be combined with WriteSValue.vi in which this parameter will take a more important role.

### **Input parameters:**

Key : The key entry of the registry  
Value : A particular value of the key entry  
Default : The default in case the key or the value does not exist  
aNumber : The DWORD that is written to the registry

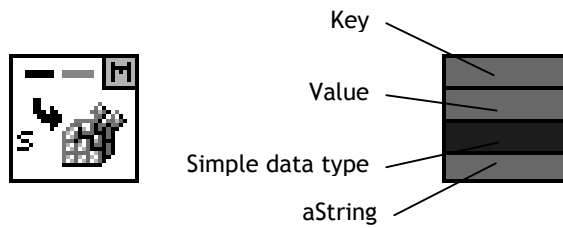
### **Output parameters:**

None

### **Used in:**

Main.vi

## WriteSValue.vi



### **Functionality:**

A String is written to the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

The *Simple data type* is a parameter that has always "String" as value. This vi can be combined with WriteValue.vi in which this parameter will take a more important role.

### **Input parameters:**

Key : The key entry of the registry  
Value : A particular value of the key entry  
Default : The default in case the key or the value does not exist  
aString : The String that is written to the registry

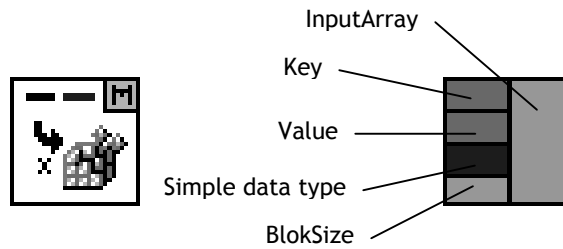
### **Output parameters:**

None

### **Used in:**

Main.vi

## WriteValueBlok.vi



### Functionality:

An array of DWORD is written to the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

There are  $x$  values written to the registry (where  $x$  equals *BlokSize*). The name of the value is composed by:

`Value+i, i ∈ [1, BlokSize]`

The *Simple data type* is a parameter that has always "DWORD" as value. This vi can be combined with WriteSValueBlok.vi in which this parameter will take a more important role.

### Input parameters:

Key	: The key entry of the registry
Value	: A particular value of the key entry
BlokSize	: The number of values
Default	: The default in case the key or the value does not exist
InputArray	: The array of DWORD where the records are written to the registry

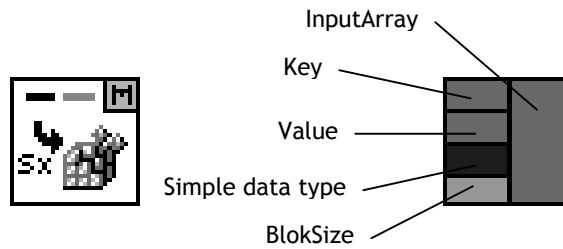
### Output parameters:

None

### Used in:

Main.vi

## WriteSValueBlok.vi



### Functionality:

An array of String is written to the registry. The registry entry is composed by:

`"Maarten\OPO\"+Key`

There are  $x$  values written to the registry (where  $x$  equals *BlokSize*). The name of the value is composed by:

`Value+i, i ∈ [1, BlokSize]`

The *Simple data type* is a parameter that has always "String" as value. This vi can be combined with WriteValueBlok.vi in which this parameter will take a more important role.

### Input parameters:

Key	: The key entry of the registry
Value	: A particular value of the key entry
BlokSize	: The number of values
Default	: The default in case the key or the value does not exist
InputArray	: The array of String where the records are written to the registry

### Output parameters:

None

### Used in:

Main.vi



# Appendix A3      Bibliography

- [1] Christopher C. Davis, "Lasers and Electro-optics, Fundamentals and Engineering", Cambridge University Press 1996 reprinted 2000 [page 73-81].
- [2] Christopher C. Davis, "Lasers and Electro-optics, Fundamentals and Engineering", Cambridge University Press 1996 reprinted 2000 [page 217-222].
- [3] Christopher C. Davis, "Lasers and Electro-optics, Fundamentals and Engineering", Cambridge University Press 1996 reprinted 2000 [page 147-154].
- [4] R. Engeln, G. Berden, R. Peeters, G. Meijer, "Cavity Enhanced Absorption spectroscopy and Cavity Enhanced Magnetic Rotation spectroscopy", *Rev. Sci. Instrum.* **69** (1998) [page 3763].
- [5] M. van Herpen, S. te Lintel Hekkert, S.E. Bisson, F.J.M. Harren, "Wide single mode tuning of a 3.0-3.8 micrometer, 1.5 W, continuous wave Nd:YAG-pumped optical parametric oscillator based on periodically poled lithium niobate", submitted to *Opt. Lett.*
- [6] N. Lynch, R. Segala, F. Vaandrager, "Hybrid I/O Automata", MIT Laboratory for Computer Science; Dipartimento di Matematica, Università di Bologna; Computer Science Institute, University of Nijmegen 2001 [page 21-26]
- [7] Son H. Bui & M. Kamaraj, "Least Square & Minimax Optimization Algorithms for circles, cylinders, straight line, sphere, and plane", web page: [http://www.coe.uncc.edu/~shbui/btry053/lsg\\_minimax/lsg\\_minimax.htm](http://www.coe.uncc.edu/~shbui/btry053/lsg_minimax/lsg_minimax.htm)
- [8] M.E. Klein, "Wavelength tuneable, diode laser pumped optical parametric oscillators based on quasi-phase-matching", Universität Kaiserslautern 2000 [page 21-40].