Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim Geisenheim
Faculty of Design, Computer Science and Media
General Computer Science (B.Sc.)

Bachelor Thesis
for attainment of the academic title
Bachelor of Science - B.Sc.

Programming of the Wavelength Stabilization for a Titanium:Sapphire Laser using LabVIEW and Implementation into the CERN ISOLDE RILIS Measurement System

Ralf Erik Rossel

Ralf.Rossel@googlemail.com

22^{nd} of December 2011

Reviewer:	Prof. Dr. D. Richter (HSRM)
Co-Reviewer:	Prof. Dr. K. Wendt (Universität Mainz)
On-Site Supervisor:	Dipl. Phys. S. Rothe (CERN)

Erklärung gem. ABPO, Ziff. 6.4.3

Ich versichere, dass ich die Bachelor Thesis selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Wiesbaden, 22. Dezember 2011

....

Ort, Datum Unterschrift Studierende/Studierender

Hiermit erkläre ich mein Einverständnis mit den folgenden aufgeführten Verbreitungsformen dieser Bachelor Thesis:

Verbreitungsform	**ja**	**nein**
Einstellung der Arbeit in die Hochschulbibliothek mit Datenträger		X
Einstellung der Arbeit in die Hochschulbibliothek ohne Datenträger		X
Veröffentlichung des Titels der Arbeit im Internet	X	
Veröffentlichung der Arbeit im Internet		X

Wiesbaden, 22. Dezember 2011

....

Ort, Datum Unterschrift Studierende/Studierender

Vorwort

Die vorliegende Arbeit stellt den Abschluss des Bachelor-Studiengangs *Allgemeine Informatik* an der Hochschule RheinMain dar. Die Flexibilität dieser Ausbildung hat mir bereits ab dem vierten Semester die Möglichkeit eröffnet, anwendungsorientierte Lehrveranstaltungen aus dem Bereich der technischen Informatik zu wählen. Die anschließende Durchführung meines Fachpraktikums am Institut für Physik der Universität Mainz, *Arbeitsgruppe LARISSA* unter Leitung von Prof. Dr. Wendt, ermöglichte mir konsequentes Fortschreiten auf dem Weg der fachübergreifenden Tätigkeit, sowie Setzen eines Schwerpunktes im Bereich der Gerätesteuerung und Automatisierung. Zur Fortsetzung dieser erfolgreichen Kooperation konnte ich meine fachlichen Qualifikationen im Rahmen des Aufbaus eines von der Forschungsgruppe entwickelten Titan:Saphir Lasersystems am CERN anwenden. Seit Beginn meiner dortigen Beschäftigung als wissenschaftlicher Mitarbeiter im Juni 2011 habe ich mich in das entsprechende fachliche Umfeld der Physik eingearbeitet und insbesondere bei der Installation der Infrastruktur zur Messdatenerfassung und Gerätesteuerung mitgewirkt. In dieser Zeit bis zum Beginn der vorliegenden Bachelorarbeit im September 2011 habe ich im Wesentlichen die Tätigkeiten der hier dokumentierten Kapitel 4, *Hardware Setup*, und Kapitel 5, *Preceding Measurements of Parameter Dependencies*, durchgeführt und damit die Voraussetzungen zur Erfüllung meiner Aufgabe der Softwareentwicklung geschaffen. Besonders hat sich mir bestätigt, dass Hardware-Kenntnise und physikalisches Systemverständnis, gelehrt unter anderem in den Curricularfächern Informations- und Systemtheorie, Rechnerorganisation und Digitaltechnik, ein wesentliches Fundament für die Ausbildung im Fach Allgemeine Informatik darstellt. Damit bilden vor allem die Kapitel 6, *Software Setup*, und 7, *System Evaluation*, den Kern dieser zwölfwöchigen Abschlussarbeit am CERN und runden damit meine bisherige akademische Ausbildung durch interessante, fachübergreifende Lern- und Anwendungsbereiche ab.

Abstract

In the context of this work the foundation for the commissioning of a comprehensive environmental and operational data acquisition system was established. This development was performed for the Resonance Ionization Laser Ion Source (RILIS) at the ISOLDE radioactive ion beam facility within the European Organization for Nuclear Research CERN. As an essential step towards long-term automated operation, a remote control and wavelength stabilization system for the RILIS titanium:sapphire lasers was put into operation. This required the installation of an data recording infrastructure to work with a distributed sensor network. After operational data within the CERN technical computing network was collected and analyzed, the required wavelength adjustment was automatically performed by a stepper motor-driven correction system. The configuration of the hardware for acquisition and control and the integration of the dedicated system modules was performed using the graphical and data-flow oriented programming language LabVIEW. By providing an extendable software framework, the whole system favors a modular design which allows for easy expansion and maintenance accommodating to future automation developments. The system was installed and commissioned in the fourth quarter of 2011 and is an on-going development with the aim of improving maintenance, protecting sensitive equipment and ensuring a constant, optimal laser ionization efficiency and ion beam quality for the users of ISOLDE.

Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde der Grundstein für eine umfassende Umgebungs- und Betriebsdatenerfassung der Resonanzionisations Laserionenquelle (RILIS) der ISOLDE Einrichtung am europäischen Kernforschungszentrum CERN gelegt. Neben der Installation und Erstellung einer Infrastruktur zur Messdatenaufnahme verteilter Sensorsysteme wurde ein Modul zum automatisierten Langzeitbetrieb der Fernsteuerung und Wellenlängenstabilisierung für das Titan:Saphir Lasersystem entwickelt und in Betrieb genommen. Durch Erfassung, Bereitstellung im technischen Netzwerk und Auswertung der Operationsdaten wurde die Wellenlängenanpassung durch ein schrittmotorbetriebenes Korrektursystem realisiert. Neben der Zusammenstellung und Ansteuerung der hardwareseitigen Komponenten wurden die einzelnen Softwaremodule in der grafischen, datenflussorientierten Programmiersprache LabVIEW implementiert. Besondere Schwerpunkte bei der Umsetzung des Projekts lagen auf dem modularen Aufbau sowohl auf Hardware- als auch auf der Softwareseite, um die leichte Erweiterbarkeit und den Austausch einzelner Komponenten im Zuge der weiteren Automatisierung zu ermöglichen. Das System wurde im vierten Quartal 2011 in Betrieb genommen und befindet sich in stetiger Weiterentwicklung mit dem Ziel, sowohl den Schutz der vorhandenen Geräte und deren Wartungsfreundlichkeit zu verbessern als auch die Sicherstellung von konstant hoher Laserionisationseffizienz und Qualität des produzierten Ionenstrahls für alle Nutzer der ISOLDE Einrichtung zu gewährleisten.

Contents

1 Introduction

In an increasingly automated environment, computers are used as complex tools that can perform a multitude of uniform and recurrent tasks, such as uninterrupted monitoring of a large set of process variables to provide machine and equipment control and also as operator support where human intervention would be either insufficient or dangerous. In the area of natural sciences and research the extensive long term acquisition and evaluation of measurement data is of equal importance as well as the exact reproducibility of experiments. For this reason, expertise provided by computer science can assist in sorting and evaluating gathered data, as well as implementing a level of autonomous process control to provide manageability and visualization for complex systems. One existing tool applied to completing these tasks is control loop algorithms. By constant monitoring, evaluation and feedback correction, the use of these algorithms assure a reliable experimentation setup. This bachelor thesis emphasizes the utilization of computers as complex scientific laboratory tools to monitor and control long term processes.

2 Motivation

Located on the border of Switzerland and France near Geneva, CERN is a multi-national host for a wide range of scientific experiments in many different research disciplines including particle physics and computer science. One of its renowned centers of nuclear physics is the Isotope Separator On-Line DEtector (ISOLDE) user facility. As service provider, ISOLDE is responsible for the production of isobar-pure particle beams, using these for a wide range of fundamental research in the fields of nuclear and atomic physics, solid-state physics, materials science and life sciences [1]. As an isotope "factory" ISOLDE relies on a wide range of ion sources and beam-probing equipment to provide a versatile setup for different research proposals. A detailed overview can be found in Ref. [2].

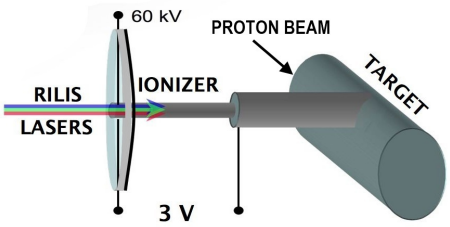


Figure 1: Diagram of the Resonance Ionization Laser Ion Source (RILIS). The heated target unit depicted on the right is being bombarded with protons. The isotopes produced move into the ionizer, where they are selectively ionized by laser beams of up to three different wavelengths. Image source: [3]

One of the ion sources increasingly being used is the Resonance Ionization Laser Ion Source (RILIS). The basic working principle pictured in figure 1 can be outlined as follows:

A high energy[1] proton beam from the Proton Synchrotron Booster (PSB) impinges upon a thick target, producing a great amount of different isotopes via fission, spallation and fragmentation [1]. The target unit is resistively heated to ≈ 2000 °C allowing the products of the nuclear reactions to diffuse into the ionizer tube where they are interacting with the RILIS laser beams.

Of high interest is the property exploration of short-lived radioactive isotopes not commonly found in nature. In order to produce an ion beam of these desired rare isotopes, laser beams of specific wavelengths are directed into the ionizer tube to achieve selective ionization and thus exaggeration of the desired ions. The ions are then electrostatically extracted and mass-separated using the ISOLDE separation magnets. The radioactive ion beam is finally sent to the different ISOLDE experiments.

In spring 2011 RILIS was upgraded with a newly designed solid state Titanium:Sapphire laser system (Ti:Sa) shown in figure 2, complementing the already existing dye laser system. Since the creation of RILIS the demand thereof has constantly increased. For this reason the necessity of developing a monitoring and control system with the possibility of process automation arose. The objectives were and continue to be to ensure the constant ion beam quality, to supervise the system and to reduce the risk of failures due to the increase of process variables and incoming monitored data. A principle design plan displaying the interconnections between distributed sensors and devices is shown in figure 3.

Figure 2: Panoramic view of the RILIS laser table setup (courtesy of S. Rothe). The three newly added Ti:Sa lasers are shown in the lower part of the picture. The upper part shows the focussing and launch setup commonly used by all RILIS lasers.

In the course of planning the system, three main goals have been established:

Operator support during RILIS shift work and maintenance ease the ability to observe all parameters and to create informative long-term measurement tools, which allow dedicating more time to beam development and system improvements.

User support increases the efficient use of online beam time by providing users with increased feedback of condensed relevant data and ensuring optimal experiment preconditions. Also scientific cooperation can be expanded by

[1] 1.4 GeV

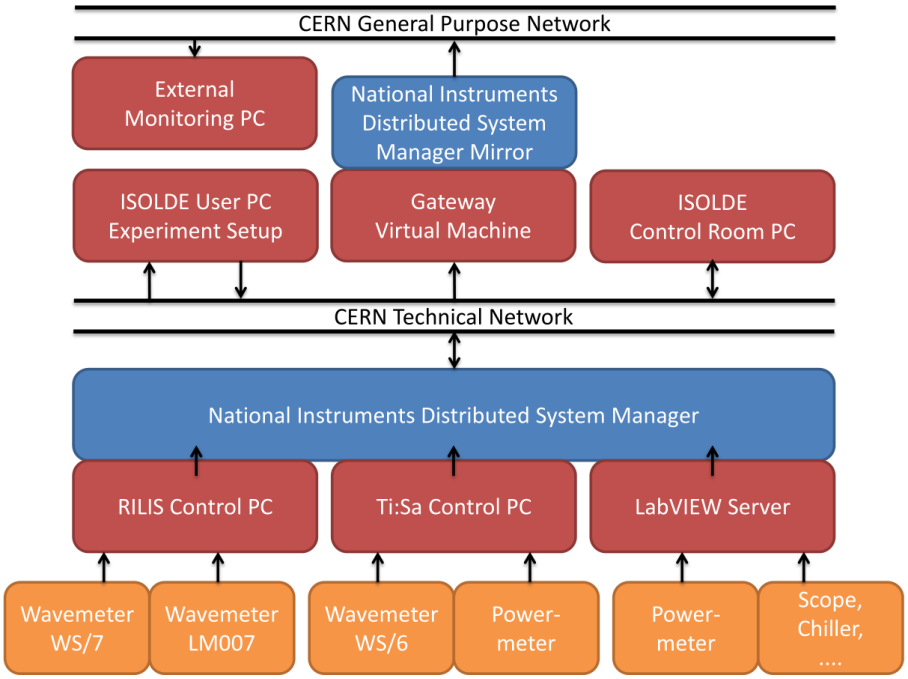


Figure 3: Diagram of RILIS monitoring concept. In the lower part of the diagram, the data acquisition devices of RILIS are represented. The instruments are connected to readout PCs running respective data acquisition programs. Through use of the National Instruments Distributed System Manager the data can be accessed via the CERN technical network.

enabling active remote alteration of the laser ionization by the blocking and unblocking of the lasers and also the tuning of the frequency within a certain range.

Machine protection ensures safe operation and protection of expensive equipment by installing a coordinated shutdown and reporting procedure in case of malfunctions. The basis for this assessment comes from the monitoring of laser specific parameters, such as wavelengths, power and timing, as well as the observation of an extended set of operational parameters ranging from cooling water temperature, water leaks, ethanol-based dye leaks, and the radiation levels.

As the RILIS laser room is considered a non-permanent, restricted access working environment due to the previously mentioned hazards, and also noise and stray light, the overall goal is to allow the possibility to monitor and control relevant parameters of the system from a remote point e.g. the ISOLDE control room. This should be implemented through the use of network attached acquisition devices and allow for the future change from a “on shift” to an “on call” operation mode.

Similar to a feasibility study, the first steps taken towards these developments included

monitoring and automatic controlling and stabilizing the Ti:Sa wavelengths. This is the topic of this thesis and is futher outlined in the following section. As a future goal, a complete intensity, spatial, spectral and temporal (ISST) stabilization is intended. For further details please refer to section 9: *On-going Projects and Future Prospects*.

3 Project Description

The developent of the Ti:Sa wavelength stabilization represents a vital building block of the RILIS monitoring and control project and is outlined in the following sections by recognizing the requirements, identifying the tasks to be fulfilled and finally describing the project design and implementation plan.

3.1 Requirements Analysis

Currently, the RILIS laser room is equipped with two independently operable laser systems consisting of six tuneable lasers and two pump lasers. The first system consists of three dye lasers which have been in operation since 2009/2010. The second system was installed in summer 2011 and consists of three solid-state Ti:Sa lasers which are powered by an Nd:YAG pump laser. All lasers are cooled using water-air chiller aggregates. To compensate for the thermal load generated by coolers, computers and dye circulators, the RILIS cabin is equipped with an air condition unit. However the current system only allows for a very rough temperature control, which causes periodic temperature fluctuations. Furthermore the disadvantageous placement of air intakes and outlets causes a noticeable temperature gradient in the room. Due to thermal expansion and contraction of the optical elements in the Ti:Sa lasers, a measureable influence on the wavelength has been evaluated and documented in section 5.1: *Temperature Measurements*.

In accordance to the objectives outlined in section 2: *Motivation*, the commissioning of an automated wavelength stabilization system to keep up to three wavelenghts on the required atomic resonances was needed to help minimizing the time required for an operator to make manual adjustments during RILIS on-line operation. The long term wavelength stabilization system was installed as a *proof of concept* and to provide the basis for further surveillance and automation. The system needed to be integrated into the current laser setup and to be composed of interchangeable hard- and software components, which could adapt to the changing requirements during experiments and future developments. To achieve this, CERN approved for industry-standard components known for their ease of use. The system had to be maintainable not only by programmers but also by instructed RILIS operators. Additionally the stabilization system had to provide the ability to perform a uniform and constant wavelength scan to verify the resonance wavelength, as well as to quickly switch between different isotopes within a certain range. The acquired data had to be stored in a logfile for long-term analysis in order to monitor, for example, wavelength drifts or gradual power losses due to misalignment or dye decay. Where possible, the system relevant parameters were monitored and controlled using network attached RS-232 acquisition devices enabling the integration of additional sensors into the system.

Summarizing, the following project use cases have been defined as:

- *The **stabilization** of a certain wavelength*
- *The **adjustment** of the wavelength within a certain range*
- *The **scanning** of a certain wavelength range*

The technical requirements have been defined as:

- *Enabling **remote monitoring** of all task relevant wavelengths*
- *The reading out of **distributed sensor data** in the laser room*
- *Allowing **component exchange** of soft- and hardware modules*
- *The **logging** of the acquired data to a file*
- *The execution of limited **automated intervention** to compensate for wavelength deviations*

As an assessment basis for stabilization, three different limiting values create an acceptable range of values. These values have been established based upon the experimentally determined resonant transition wavelength of astatine, which was examined during the development of the system. The acceptable range of deviation of the ion signal strength (99%, 95% and 68% of maximal value) were determined by fitting a Gaussian distribution to the resonance scan measurement and calculating the accordant widths to $\pm$ 0.0189 cm^{-1}, $\pm$ 0.0395 cm^{-1} and $\pm$ 0.1064 cm^{-1}. The value ranges are illustrated in figure 4. Results regarding the performance of the stabilization according to these limitations can be found in section 7: *System Evaluation*.

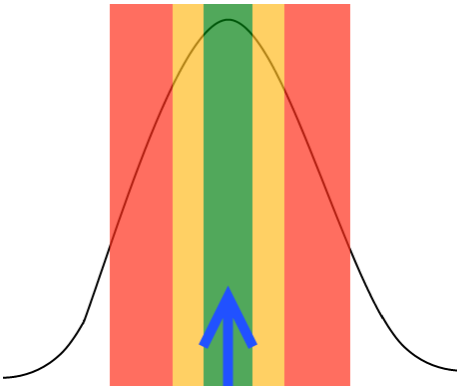


Figure 4: Diagram showing the zones of acceptable wavelength deviation.

If the wavenumber deviates for more than 1 cm^{-1} in two consecutive measurements, the system may not intervene. Operation is resumed if the measured wavenumber is within range. These determinations are based on taking into account the the wavemeter measurement experiences regarding mode jumps described in section 4.3: *Wavemeters* as well as the hysteresis measurements executed in section 5: *Preceding Measurements of Parameter Dependencies*.

3.2 Implementation Concept

Prior to commissioning the system, an assessment of the available hard- and software components, which are described in more detail in the respective sections 4: *Hardware Setup* and 6: *Software Setup*, led to the following specific implementation conditions.

As a prerequisite, the wavelength of the Ti:Sa laser had to be acquired using the wavemeters available in RILIS which are described in section 4.3: *Wavemeters*. These measurements were to be recorded in order to monitor the current wavelength value as well as determine a long-term trend.

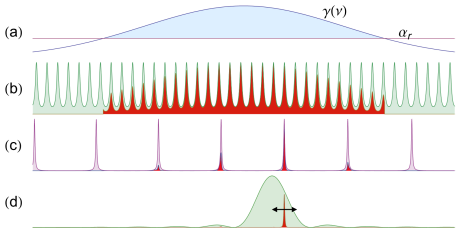


Figure 5: Ti:Sa laser frequency selection (courtesy of S. Rothe). The four plots illustrate the laser gain versus the laser frequency on the horizontal axis.

To change the wavlength, the element determining the smallest mode bandwith of the Ti:Sa laser was to be manipulated. The main elements used to filter the wavelengths are a mirror set coated to reflect a wide range, a bifringent filter and an etalon. A detailed description of this principle can be found in [7] and [6]. To visualize the scheme of wavelength selection applied in the Ti:Sa laser, the laser gain versus the laser frequency on the horizontal axis is illustrated in figure 5.

Plot (a) represents the overall gain of the laser medium $\gamma(\nu)$ and the sum of all losses α_r due to reflection and absorption. The remaining longitudinal mode structure defined by the Ti:Sa cavity (b) and an inserted thin etalon (c) is displayed in both central plots as the magenta curve. Diagram (d) illustrates the wavelength selection performed by inserting a bifringent filter. For the desired application of automatic wavelength stabilization, the etalon was to be tilted as illustrated in figure 6. This was performed by a computerized actuator which used a stepper motor to rotate the thumbscrew of a modified mirror mount. The motor was to be driven by an already existing four channel stepper motor driver manufactured by OWIS GmbH. More specific data concerning this computerized actuator system can be found in section 4.9: *Stepper Motor Driver* and section 4.2: *Motorized Etalon Mount*. Furthermore, a newly created control and data acquisition software should be written in the software

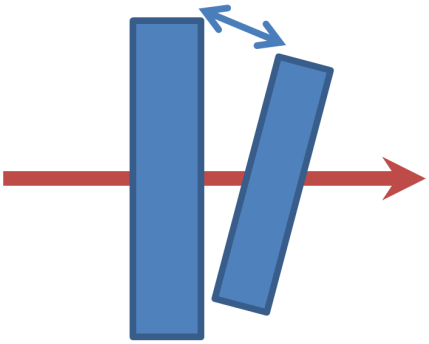


Figure 6: Etalon tilt illustration. The laser beam shown in red is manipulated by the tilting angle of the etalon.

development environment LabVIEW by National Instruments, which is described in more detail in section 6.1: *LabVIEW Concepts*.

By using a LabVIEW controlled stepper driver, the system meets the requirements of modular design, interchangeable standard components and easy to maintenance of a computerized actuator. Also this sytem allows for quick changes and manual intervention on several levels: manual adjustment of the thumbscrew, the changing of positions by direct manual control of the OWIS stepper driver, and manual control in the control loop software. The concept of modularity is furthermore found in software by defining shared variable interfaces for each module to allow for easy change of individual components. The LabVIEW Program implements a closed control loop, in which the measured wavenumber is compared to the setpoint wavenumber. The program then applies the necessary corrections by controlling the driver and actators. An additional project goal is to provide a framework for further developments such as component changes, experimenting with different control loop algorithms and development of data-monitoring tools.

4 Hardware Setup

Representing the interface between computer science and the physical world numerous data acquisition and control hardware was installed and accessed using either a direct serial connection (RS-232), an Universal Serial Bus (USB) connection, or the CERN technical computing network (TN) for remote access.

4.1 Titanium:Sapphire Laser

The main hardware element of the project is the Ti:Sa laser which is shown in figure 7. Its current CERN design was adopted from the research group LARISSA of Johannes Gutenberg-Universität Mainz, Germany. As described in section 3.2: *Implementation Concept*, the wavelength manipulation of the Ti:Sa laser is achieved by tilting the etalon in respect to the resonator cavity. A more detailed description of the overall working principle of the Ti:Sa laser can be found in [6].

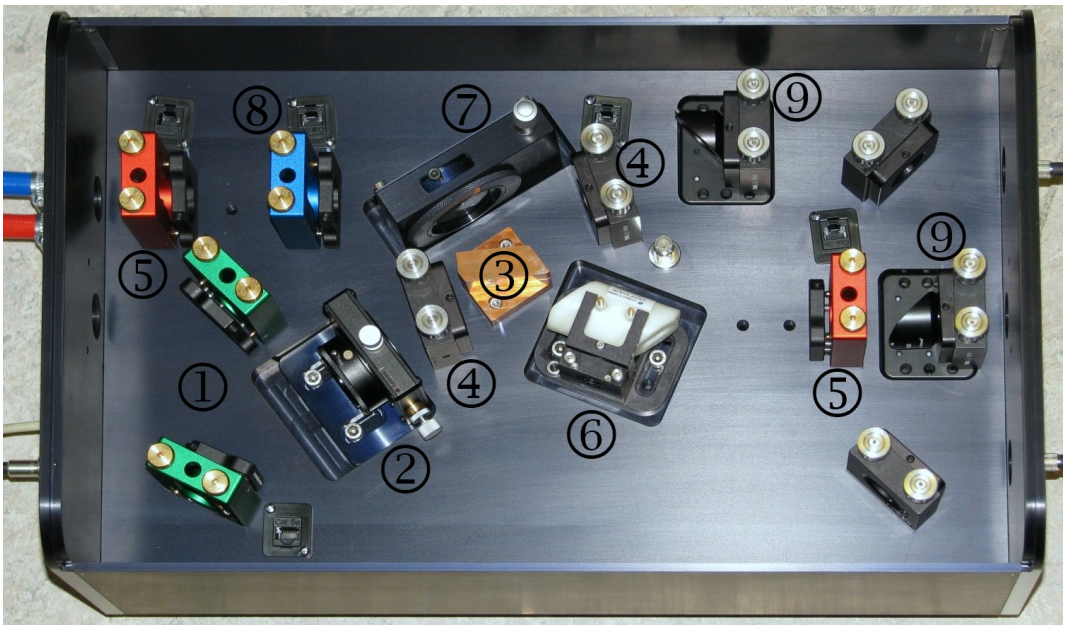
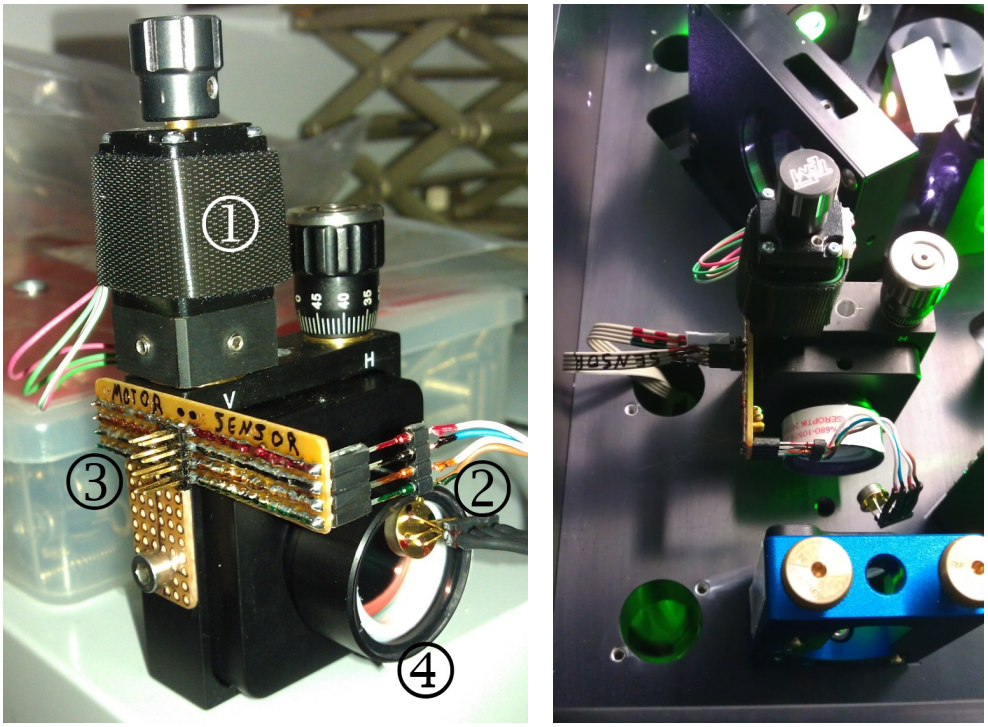


Figure 7: Ti:Sa laser cavity elements (courtesy of S. Rothe). The green mounts ① hold the pump beam reflectors. The beam is focussed by a moveable lens ② into the crsytal ③. The resonator is composed of two concave focussing mirrors ④ and two end mirrors in red mounts ⑤. A Q-switch ⑥ can be used for synchronization. The bifringent filter ⑦ and the Etalon in the blue mount ⑧ perform the laser mode and wavelength selection. Additional elements ⑨ are used for wavemeter coupling and alignment-HeNe input.

4.2 Motorized Etalon Mount



(a) Motorized mirror mount with circuit board and contactless infrared (IR) temperature sensor.

(b) Fixation of the mirror mount in the Ti:Sa z-resonator.

Figure 8: Etalon mount capable of computerized movement control and temperature data acquisition.

The vertical axis micrometer thumbscrew of the mirror mount[2] holding the etalon was equipped with a stepping motor[3] driven linear actuator specified for taking 200 steps per revolution. Additionally a contactless infrared temperature sensor[4] was added to provide complimentary temperature measurements, which are shown in section 5: *Preceding Measurements of Parameter Dependencies*. The modified mount is pictured in figure 8 (a) showing the attachment of the stepping motor ①, the temperature sensor ② and the circuit board ③ leading to the data acquisition microcontroller described in section 4.4: *LabJack Data Acquisition Device* and the motor driver described in section 4.9: *Stepper Motor Driver*. The etalon itself is marked with ④. Figure 8 (b) showing the setup in the Ti:Sa laser, displays that no additional modification of the laser itself is necessary.

[2]Thorlabs VM1 Kinematic Mount with Vertical Drive, Ø1" Optics
[3]Haydon Kerk 21000 series size 8 (21mm, 0.8")
[4]Melexis MLX90614 E SF AAA Single Zone Infrared Thermometer

4.3 Wavemeters

The RILIS laser room is equipped with three independently oparable wavemeters. Two of which[5] were alternatingly used as the data source for this project. This being due to the availability during normal RILIS operation. When in use, the wavemeter needed to be connected to a standard PC via USB interface, as all calculations concerning the wavelength measurement were performed on the host PC. Furthermore a fiber-optics switcher module was used for the rapid successive measurement of up to four different laser wavelengths. This switcher module was controlled also by means of the proprietary software of the wavemeter. Running this software is a prerequisite to using the system. This represents an important factor to be considered during software development and is described in more detail in section 6.3.1: *Wavemeter Readout*.

4.4 LabJack Data Acquisition Device

In order to support the connection and readout of a large number of sensors, the specialized data acquisition device[6] LabJack was installed. This device supports both USB and Ethernet connections and is capable of performing digital and analog measurements. Another advantage is the availability of device drivers, which were used in programming the etalon temperature acquisition described in section 6.3.3: *LabJack Temperature Sensor Readout*. Figure 9 shows the connection of the infrared temperature sensor introduced in section 4.2: *Motorized Etalon Mount* to the screw terminals of the LabJack.

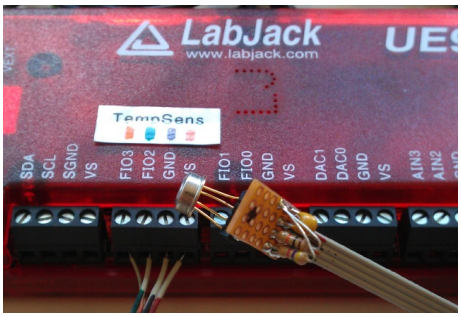


Figure 9: Temperature sensor connected to the screw terminals of the LabJack data acquisition device.

4.5 Communication Equipment

As one of the goals of RILIS automation was the remote accessibility of possibly all acquisition and control devices; the use of serial to Ethernet media converters and device servers[7] was emphasized during the setup phase prior to the software development. The application of this communication equipment has the advantage of allowing both, usage of standardized cabling and infrastructure of the CERN technical network, as well as the establishment

[5] HighFinesse Ångstrom: WS/6 High Precision, WS/7 Super Precision
[6] LabJack UE9
[7] perle IOLAN TS2 Serial to Ethernet Device Servers and IOLAN STS-D Terminal Server

of software framework components, as described in section 6: *Software Setup*. The usage of standard ethernet cabling was extended for serial RS-232 connections by using RJ45 to SUB-D9 adapters. This allows the establishment of connections without gender changers or especially configured cables

4.6 Arduino Prototype

As a proof of concept, the Arduino[8] microcontroller based manual control box pictured in figure 10 was developed. This allows for actuation of the etalon stepper to be controlled by two buttons and a potentiometer at variable speeds. The Arduino Mega microcontroller is capable of addressing up to 54 digital inputs and outputs, 13 of which are capable of 8 bit pulse width modulation (PWM), and can also read out 16 analog inputs with a resolution of 10 bit. A miniature stepper driver[9], also capable of PWM, was connected to the digital output ports of the Arduino to send the stepping signal. As the motor needs a comparably higher current of ≈ 150 mA for each of its two coils, the use of an additional 12 V power supply was needed. Using this control box demonstrated that the wavelength of the Ti:Sa laser can be conveniently and evenly controlled using the modified etalon mount. Furthermore, this setup helped determine hardware properties, such as cabling configuration and motor resonance at variable stepping speeds. The code programmed for this prototype is outlined in section 6.3.6: *Arduino-Based Etalon Motor Driver* The prototype has proven to be useful for future developments explained in section 9.1: *Individual Ti:Sa Microcontroller*.

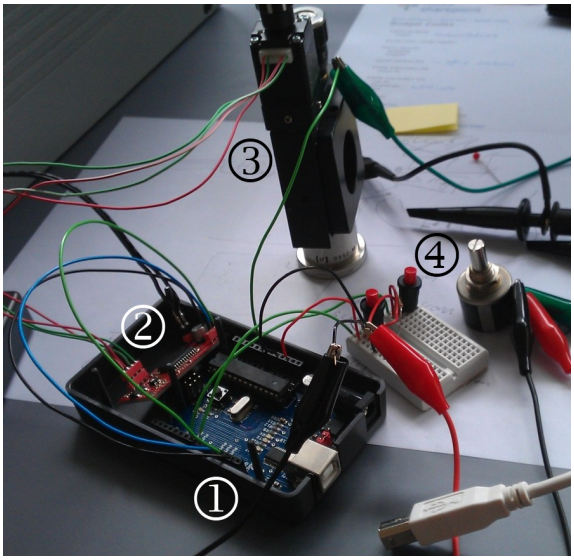


Figure 10: Development setup of Arduino prototype. The microcontroller circuit board ① is connected to a driver circuit board ② which is used to move the etalon stepper motor ③. Manual control of vertical movement at variable speeds is achieved by the use of two buttons and a potentiometer ④.

[8] Arduino Mega 1280 and Mega 2560
[9] EasyDriver V4.4 Stepper Motor Driver by Brian Schmalzhaus

4.7 Air Temperature Sensors

To assess the influence of the RILIS room temperature on the produced laser wavelength of the Ti:Sa wavelength, two PT100-type[10] temperature sensors were installed to measure the air condition unit (A/C) intake and outlet temperatures. Due to the position of the A/C outlet being located directly above the Ti:Sa laser installation, a severe influence on the wavelength stability was presumed. The sensors were connected to two Arduino analog input ports according to the wiring diagram shown in figure 11.

This provided the ability to conduct consistent long-term measurements, as well as the logging of the data for evaluation of wavelength trends. The software created for this process is documented in section 6.3.5: *Arduino-Based RILIS Temperature Readout* and represents an exemplary application for the projected RILIS monitoring software. The Arduino 10 bit resolution input pin has an internal 20 kΩ pull-up resistor with a 5V reference voltage. The accuracy of this setup is strongly dependent on the resistor tolerances. Therefore this approach was used to enable a qualitative statement and to serve as a basic measurement arrangement for creating temperature data sets under comparable circumstances. Thus the achieved results could be compared to estimate and quantify differences after applied changes, such as enabling and disabling the stabilization. The accordant results are documented in section 7: *System Evaluation*. A manual two point calibration of the intake and outlet sensors was performed via comparative measurement using digital thermometer:

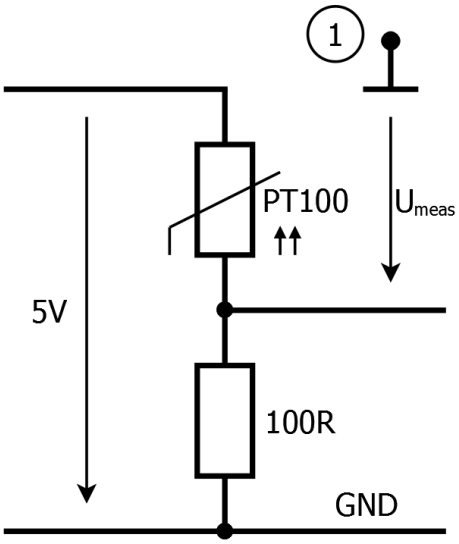


Figure 11: PT100 sensor wiring diagram. Connector ① marks an analog input of the Arduino.

$$
\begin{aligned}
&Sensor1: \\
&2.67793\text{V} \mathrel{\widehat{=}} 22.6^\circ\text{C} \\
&2.7088\text{V} \mathrel{\widehat{=}} 28.7^\circ\text{C} \\
&Sensor2: \\
&2.5823\text{V} \mathrel{\widehat{=}} 8.1^\circ\text{C} \\
&2.668573\text{V} \mathrel{\widehat{=}} 26.75^\circ\text{C}
\end{aligned}
$$

[10] 100 Ω resistivity corresponds to 0°C

4.8 Powermeters

The laser power had to be monitored periodically to ensure maximum possible ionization efficiency. For this purpose, an eight port multi-powermeter[11] and two single head powermeters[12] were added to the sensor network. Both device types use a serial RS-232 connection to communicate with the host PC. The respective readout programs are documented in section 6.3.4: *Powermeter Readout*.

4.9 Stepper Motor Driver

The OWIS stepper motor driver[13] supports the control of up to four actuator units. The communication with the computer is performed using the serial RS-232 interface.

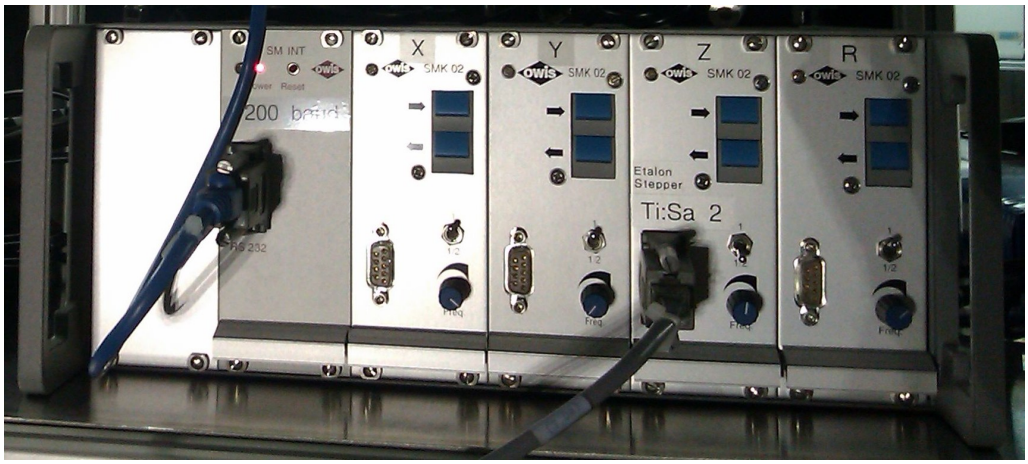


Figure 12: Stepper motor control and driver unit. On the left side the RS-232 communication interface module is visible. On the right side the four driver and control modules are pictured with one cable connected to the Z-axis module.

The front panel pictured in figure 12 shows the RS-232 communication module on the left side. On the right part of the image all four actuator control modules are visible. These modules allow for the connection of an actuator stage, consisting of a stepper motor and two end stops, using a female D-Subminiature connector with 9 pins (DE-9). The buttons on the upper part are used for manual triggering of stepwise movement. As according to the manual, the connected motor operates in continuous movement if the button is pushed for more than 0.5 s. A lever located below the directional control buttons allows for the switching between the available full and half step modes. The use of the half-step mode

[11] Gigahertz-Optik Optometer P-9801
[12] Gentec TPM 300
[13] One *SM INT* RS-232 interface module and four *SMK 02* driver modules

allows for finer coorrections of the wavelengths as the motor uses 400 steps per revolution instead of 200. The motor movement speed can be adjusted by turning the potentiometer knob on the lower part of the driver module. Most importantly, the directional buttons and stepping mode lever of the driver override the commands sent by any control software. This means that changing the lever position on the front panel will affect the amount of correctional steps performed by the stabilization system outlined in section 3.2: *Implementation Concept*.

During the software development of the wavelength stabilization, a LabVIEW-based driver program had been developed with its front panel resembling the front panel controls of the physical stepper driver control unit. This VI is documented in section 6.3.7: *Stepper Driver Control Program*.

To estimate the overall feedback loop timing the communication delay of the serial connection to the driver has been determined. At a baud[14] rate of 1200 the time taken to transmit one symbol is 833μs. Since the communication mode is set to 8N1 (eight data bits, no parity, one stop bit) it takes 10 symbols including the start symbol to transmit one byte. The communication string taken from the manual consists of a maximum of 72 bytes to control three axes (i.e. three etalon stepper motors) simultaneously. This results in a communication delay of

$$72 \cdot 10 \cdot 833\mu s = 599760\mu s \approx 600ms$$

To sum up: for one correction interval it takes the driver 2 s to move the motor and report back the successful execution of the command. As the maximum setting of 9200 baud resulted in severe communication errors a conservative speed of 1200 baud was chosen for safe operation. For future projects this aspect may be a possible sought improvement as described in section 9: *On-going Projects and Future Prospects*.

4.10 Server Computer

During the setup phase of the Ti:Sa laser system the need for a dedicated server system arose. This system had to provide extended readout capacity and minimize interference with normal RILIS operation. To meet these requirements and to support the connection of USB, RS-232 and Ethernet devices a CERN standard issue desktop PC was set up above the RILIS laser room table. This server hosts the readout data of directly connected devices such as the LabJack described in section 4.4: *LabJack Data Acquisition Device*, as well as remotely accessed instruments,which use the terminal servers described in section 4.5: *Communication Equipment*. All acquired data is published into the National Instruments Distributed System

[14] In binary transmissions, as used by the RS-232 interface, the bit rate corresponds to the baud rate.

Manager (DSM), which is described in detail in section 6.2: *National Instruments Distributed System Manager*, and can be accessed from the CERN technical network.

5 Preceding Measurements of Parameter Dependencies

To obtain initial information on the stability of the wavelength and its dependency on external parameters for the present work, preceding long term measurements of the wavelength drift as a function of the ambient room temperature were carried out. By another measurement the hysteresis and response time of the electromechanical actuator for the etalon for wavelength stabilization were determined.

5.1 Temperature Measurements

An ISOLDE experiment run is usually scheduled for several consecutive days. If the RILIS is also required, it is operated in eight hour shifts. Although the RILIS laser room is in a relatively protected and separated area it is subject to temperature changes, due to a number of reasons: the temperature in the ISOLDE hall, opening of the doors, number of people in the room and cooling devices in operation as mentioned in section 3.1: *Requirements Analysis*. To determine the correlation between temperature changes and the deviation of laser wavelengths the etalon temperature of one Ti:Sa was measured for 13 h, to allow for comparability to one RILIS shift. For this a contactless IR sensor was placed angular over the etalon to allow a complete measurement. This was done as close at possible without obstructing the laser beam and without moving the etalon to determine possible drifts. The measurement resulted in verification of a significant correlation and thus the need for a long term stabilization.

The plots in figure 13 separately display the changes in wavenumber and in temperature. The measurement was started at 11 pm and one can clearly see that the fluctuation increases after eight hours corresponding to the start of a work day. Error values resulting from temperature changes occurred during simple events, such as opening of the door to the laser room and the turning on and off of equipment.

Plot 14 displays the correlation between the wavelength and the etalon temperature. The slope of the linear fit ammounts the accordant dependency to -1,0423 $cm^{-1} \cdot K^{-1}$ As a comparison the wavelength of a frequency-stabilized HeNe laser at 473.6125 THz (632,991 nm (vac) $\widehat{=}$ 15798,0138 cm^{-1}) was recorded and is displayed in figure 15 using a similar vertical scale as in figures 13 and 14. Additionally the wavemeter temperature was recorded to investigate for systematic error caused by the wavemeter, as the HeNe can be considered the reference for stability. After three hours a rise in temperature was noticeable and correlated with positive deviation in wavenumber measurement which can be attributed to the wavemeter readout error.

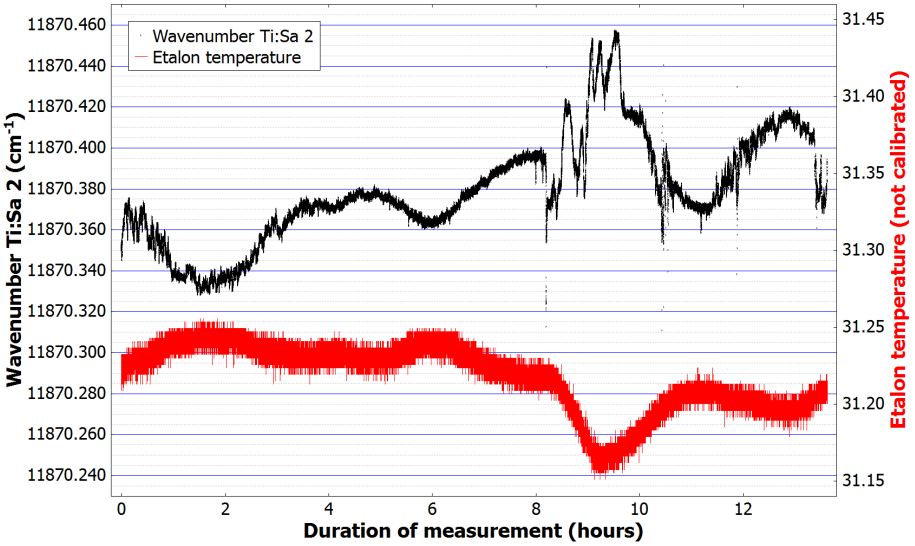


Figure 13: Plot of wavenumber and etalon temperature trend. The collected wavenumber values are depicted in black and correspond to the left vertical axis. The collected etalon temperature values are depicted in red and correspond to the right axis. The running time of the measurement is displayed on the horizontal axis.

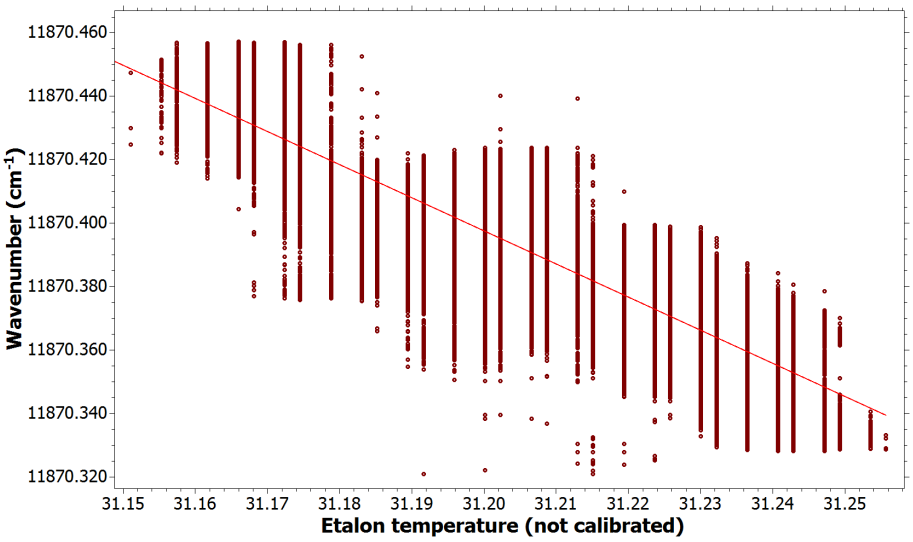


Figure 14: Plot of wavenumber values versus etalon temperature. The acquired wavenumber values correspond to the vertical axis while the temperature corresponds to the horizontal axis. A linear fit applied to the data set is depicted as the red line.

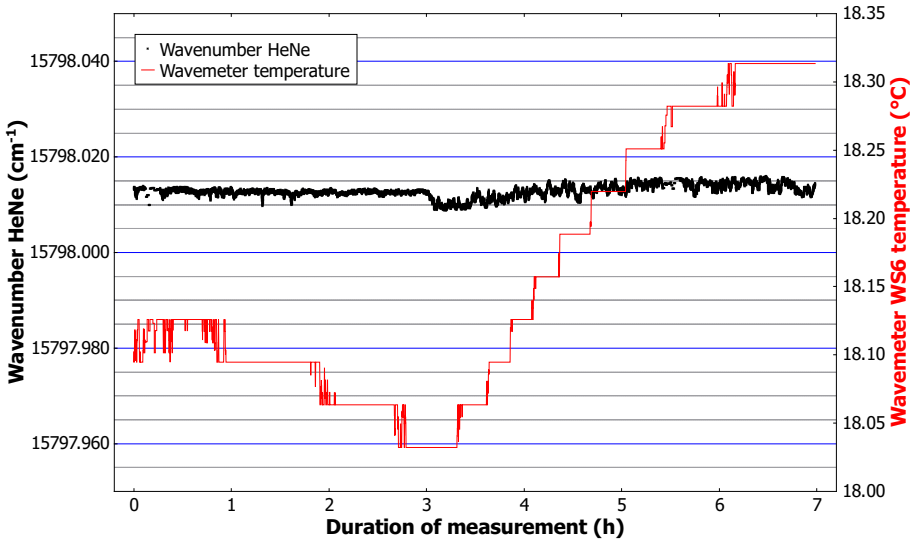


Figure 15: Plot of HeNe wavenumber values (left scale) and internal wavemeter temperature sensor values (right scale) recorded for seven hours (bottom scale).

5.2 Hysteresis Measurements

As the mirror mount consists of several mechanical components, which are subjected to manufacturing tolerances and moving interdependently a movement hysteresis is observed. Turning the micrometer screw clockwise and counterclockwise for the same angle will not result in the same positioning of the etalon and thus a different wavelength. To determine these mechanical properties in comparison to the stepper motor position the thumbscrew of the etalon mount was rotated alternatingy. An exemplary measurement data set evaluating 100 steps over 5 repetitions is plotted in figure 16. By starting from an mechanical end position of the etalon mount, acquired values were stretched out on the left side of the plot. The hysteresis at this etalon position was determined to 20 steps.

5.3 Etalon Movement Range Scans

As the Ti:Sa laser was designed to be tunable over a wide range of wavelengths, the possible areas of control loop setpoints had to be determined prior to enabling the stabilization. An exemplary tuning profile is shown in Figure 17. The horizontal lines on the left and right correspond to the mechanical movement restrictions of the etalon mount although the adjustment micrometer screw can still be turned. The vertical gaps in the wavenumber values can be described as “mode jumps” occurring when the etalons comb-shaped gain profile passes under the correnspondig gain profile of the bifringent filter as described in section 3.2: *Implementation Concept*. The outlying values can be traced back to wavemeter

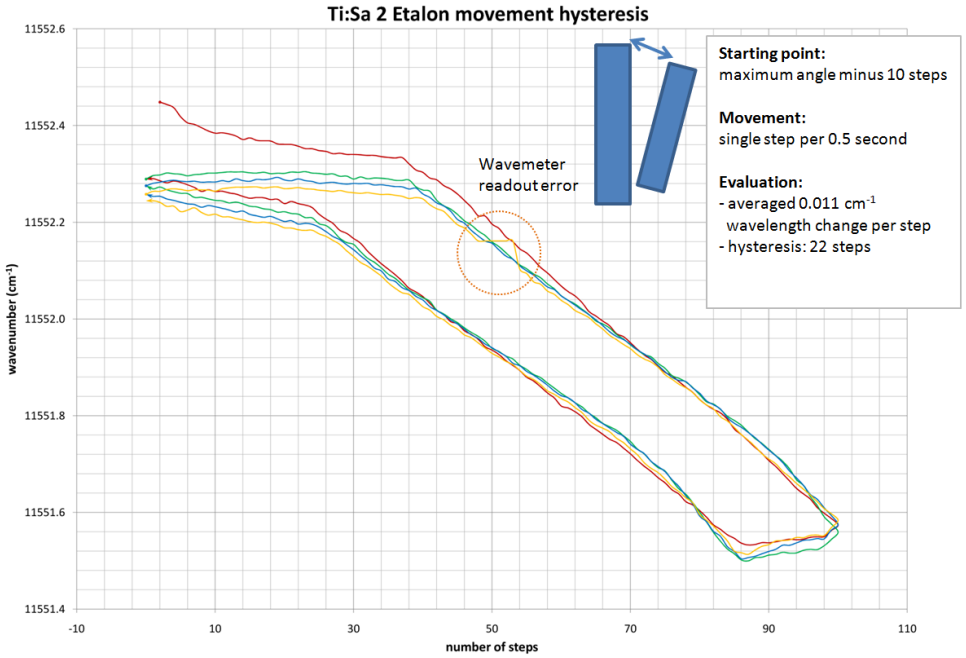


Figure 16: Plot of Ti:Sa wavenumber values (left scale) versus etalon position (bottom scale).

readout and calculation errors which were likely due to the processing speed of the server computer. For the stabilization routine this was taken into consideration, so as not to compensate for a wrongly acquired actual wavelength.

Additionally, an areal scan encompassing the whole tuning range of a specific etalon was conducted to illustrate the change in wavenumber and power in respect to its horizontal and vertical pitch. In figure 18 the resulting data plots reveal a bowl-shaped structure where the relative minimum value of the wavenumber and the relative maximum value of the power corresponds to the "flash" position perpendicular to the laser beam. The plateaus on the plots' edges represent the mechanical movement limitations of the mirror mount despite the motor turning. The "hole" in the power plot is caused by an impurity on the etalon surface which prevented the Ti:Sa to lase in this position. Furthermore, the sudden drop in the back of the power plot reveals a drop in the power of the pump laser.

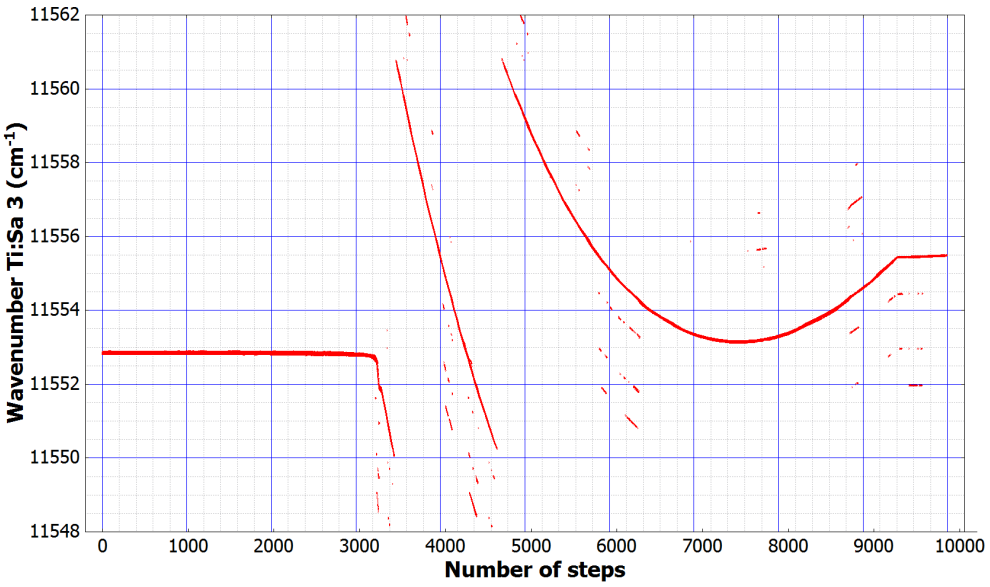
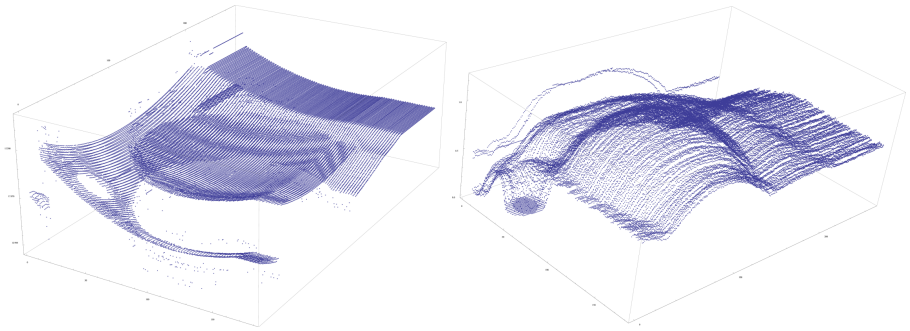


Figure 17: Plot of the tuning range of a specific etalon displaying the wavenumber in respect to the number of steps taken.



(a) 3D plot of Ti:Sa laser wavenumber values (b) 3D plot of Ti:Sa laser power values

Figure 18: Plots of areal scans of etalon tuning range. Both data sets were acquired simultaneously.

6 Software Setup

The software development platform Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) by National Instruments Corporation (NI) is already used in many of the control and monitoring applications at CERN [10]. Therefore, its use in this project was necessary to meet the requirement of easy maintainability and integration into the existing system structure. Additionally, LabVIEW provides an intuitive graphical programming interface which facilitates the modification or extension of LabVIEW programs by users who lack a thorough training in programming.

This chapter provides a general overview of LabVIEW, based on selected examples and describes the programs developed for the RILIS monitoring and control system as well as their specific use for wavelength stabilization.

6.1 LabVIEW Concepts

In addition to its integrated development environment (IDE), LabVIEW incorporates similar programming structures to those found in high level programming languages. A wide range of hardware integration libraries specifically designed for instrument interaction and data acquisition are also included. Through the use of examples, the following paragraphs describe the most prominent features, as promoted on the National Instruments product web page [11]. The IDE is available for Windows, Mac, Linux, and real-time operating systems such as VxWorks and provides a wide range of separately installable functional modules each specified for different tasks ranging from data acquisition and visualization to audio and video stream display and analysis.

6.1.1 Front Panel and Block Diagram

A LabVIEW program, also called a virtual instrument (VI), consists of a front panel and a block diagram. An example displaying both windows can be found in figure 19. The front panel handles manual data input and display via a modular and easily customizable graphical user interface. Most of the controls and indicators are represented by a real life metaphor such as dials, knobs, levers, LEDs, gauges and graphical XY plots. These elements can be dragged to the front panel by accessing a stacked right-click menu pictured in figure 20 containing all available user interface elements. In this feature, a comparison can be made to addon packages such as the Qt framework or the “Visual” IDEs available for text-based programming languages. Each front panel input element is represented by an icon or “glyph” on the block diagram. The block diagram contains the program code of a VI which is written in the distinguished programming language G.

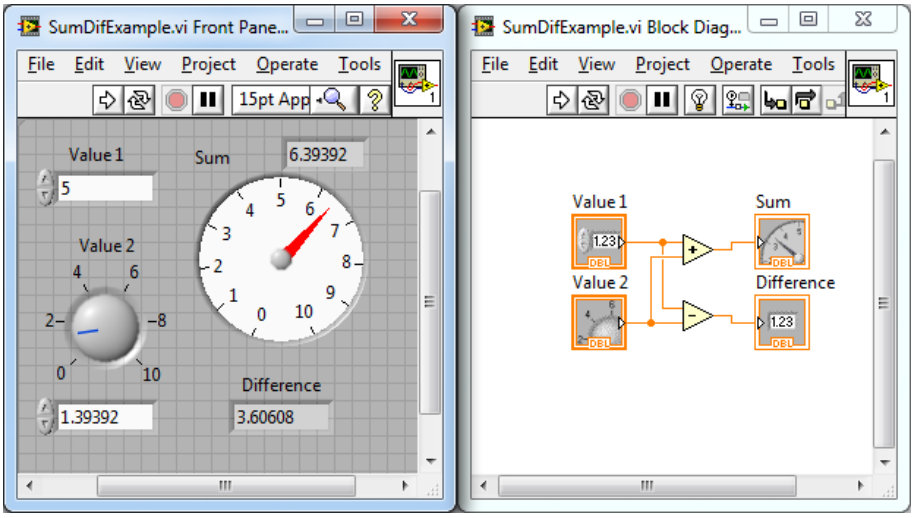


Figure 19: Front panel and block diagram windows of an example LabVIEW program. Values can be entered on the front panel (left) using interactive graphical elements such as dials and input boxes. In the block diagram (right) the data flow and parallel processing is illustrated by color coded virtual wires corresponding to the data type transported.

Interconnections between these elements and functional programming modules are made using virtual color-coded wires corresponding to the respective data type. This coding method also allows the visualization of the data flow principle during the program development process which will be explained in the next section: *Graphical Data Flow*. However, as complexity increases the code also becomes visually "entangled". To counter this problem, the code can be modularized and also design patterns applied as described in more detail in section 6.1.3: *Design Patterns*. To modularize code, extensive functions and methods can be stored in separate sub-VIs and libraries. These correspond to procedures, functions, shared objects (SO) and dynamic link libraries (DLL) commonly used in other application development frameworks and programming languages such as C/C++ and Java. The interface of a sub-VI is defined by a connector pane which is comparable to parameters and return values in functions and methods in text-based

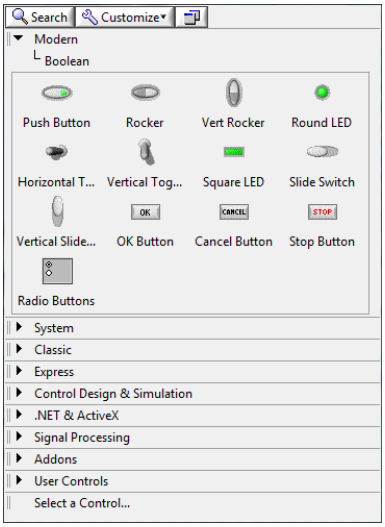


Figure 20: LabVIEW control panel right-click context menu. A wide range of input metaphors can be accessed via this menu and dragged to the front panel of a VI.

programming languages. A sub-VI can be tested individually and called by placing its icon in the block diagram of other VIs thus allowing simple code reuse and clean modularization.

An application example of this concept is displayed in figure 21 which is taken from the stepper motor driver VI described in more detail in section 6.3.7: *Stepper Driver Control Program*. The connector pane shown on the upper right part of the window displays a small grid of connector terminals. The grid can be configured to support more connections as needed. Each front panel input or output element can be assigned to correspond to a connector terminal which then changes color according to the associated data type. The resulting icon representation of the sub-VI is pictured above the front panel in the top center of figure 21. A comparable pseudocode function header could be written as

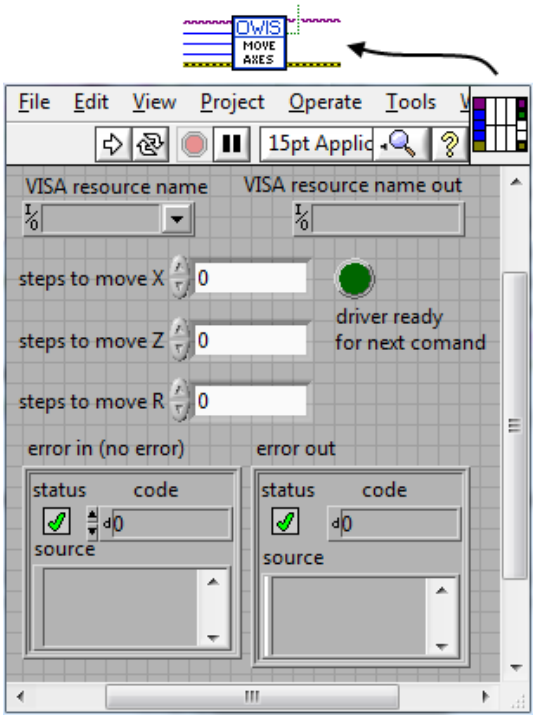


Figure 21: Screenshot of a sub-VI front panel. The connector pane is visible in the upper right corner of the frontpanel. The inclusion of the sub-VI icon is shown on the uppermost part of the image.

```
bool move_axes(
    int steps_to_move_x,
    int steps_to_move_y,
    int steps_to_move_z,
    string *visa_resource,
    err_struct *error_cluster)
```

In this example the pseudo-function takes three integer values representing the "steps to go" for each axis and two references, one to a visa resource (serial port) and the second to an error cluster comparable to a struct in the C programming language. The return value is a boolean statement representing whether the driver is ready for the next command.

A more detailed overview as well as an extensive number of examples can be found in

the integrated LabVIEW help function.

6.1.2 Graphical Data Flow

LabVIEW follows a data flow oriented design and development paradigm and is programmed in the graphical, general purpose programming language G [12]. The virtual wire connections between functional modules represent a definite information flow from a source such as an input control to a data sink such as indicators. This concept also comprises the inherent parallel processing of data through automatic multithreading which is also illustrated in the block diagram of figure 19. However, this also can lead to unpredictable program execution order, especially when using parallel loops. This behavior may be desired for specific calculations to improve program performance. If this is not the case, it can be countered by applying synchronization mechanisms also known in other programming languages such as semaphores, shared memory and message-based process and thread notifications.

Comparable to the Java virtual machine run-time environment a VI relies on the LabVIEW run-time engine which handles all calls to subVIs, references and parallel code execution. Therefore, this environment has to be installed on the target system to support the execution of compiled LabVIEW code. [15]

6.1.3 Design Patterns

When developing a large scale program or software project the use of a well defined style guide greatly enhances the overall maintainability of a program. As opposed to one dimensional code flow in text-based programming languages a LabVIEW block diagram allows liberal and spaced out placement of the functional elements on its virtually boundless two dimensional area. Adhering to meticulously defined placement rules serves as a useful tool towards readability of the programmed code at all project scales.

An additional software engineering tool is the application of design patterns. A design pattern serves as a helpful outline how to efficiently perform recurring programming tasks throughout the whole development process. As in text-based programming languages the combination of well established templates helps to create and maintain complex program structures.

As a basis for style oriented programming and development in LabVIEW this wavelength stabilization project heavily relies on the instructional work provided by Peter A. Blume and his descriptions of best practice approaches described in [4]. Examples of the most common design patterns used in LabVIEW development are briefly described in the following paragraphs. They were taken from this book and subsequently applied during the development of the RILIS monitoring and control VIs described in section 6.3: *RILIS Monitoring and Control Programs*

Continuous Loop

The continuous loop consists of a single *do-while* loop executing the instructions contained inside until the break condition is met. Figure 22 displays a small non-functional template for instrument interaction using the Virtual Instrument Software Architecture (VISA) serial interface[9]. The serial interface is configured and initialized prior to executing the while loop by the functional element shown on the left part of the image. *Write* and *read* instruction modules within the loop perform the communication with the instrument. These modules provide the functionality of performing system calls and access the serial interface. Once the break condition is met, i.e the stop button is pressed, the loop stops and executes the *close interface* module on the right.

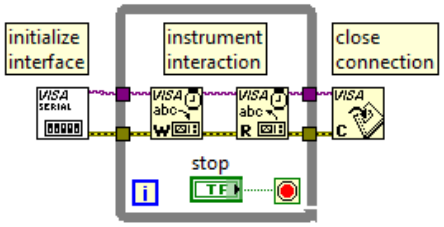


Figure 22: Block diagram of a continuous loop template. After initialization of the serial interface, the write and read operations inside the loop will be executed until the stop condition is met, followed by closing the interface.

The continuous loop is one of the basic design pattern also commonly found in other programming languages. It is suitable for recurring tasks such as receiving a measurement value from a connected device without the need of manual intervention.

State Machine

State machines can be represented in the design process by a flow graph as shown in figure 23. To adapt this concept for LabVIEW the continuous loop pattern described in the previous paragraph is extend with a conditional structure, an enumeration and a shift register. An exemplary implementation is pictured in figure 24. The task of each state is represented by the code inside the different layers of the conditional structure. For illustration purposes the layers of the case structure have been placed adjacent to each other. An enumeration containing the state identifiers is propagated through a shift register from each loop iteration to the next. The transition condition within the states is determined by the combination of functional boolean elements. This construct can be compared to the switch case concept in text-based

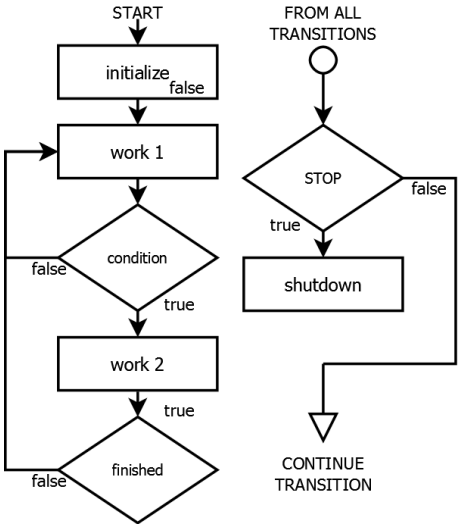


Figure 23: State machine flow graph example.

programming languages. This design pattern allows for both cleaning up the graphical code in the block diagram and also simplifies the testing process by step by step comparison of single states to the design specifications. However the printed documentation of the code can cause difficulties because of the stacking of the states.

As in other programming languages, there are many more design patterns available to suit a wide range of project specifications and complexities. In LabVIEW 6.1 for example the use of an event handling structure was introduced to allow for more efficient event-based user interaction processing and thus diverting more CPU time to parallel tasks. It is recommended mainly for VIs that require intensive user interaction [4]. For the development of the wavelength stabilization the patterns mentioned above have been widely used and are illustrated in detial in section 6.3: *RILIS Monitoring and Control Programs*

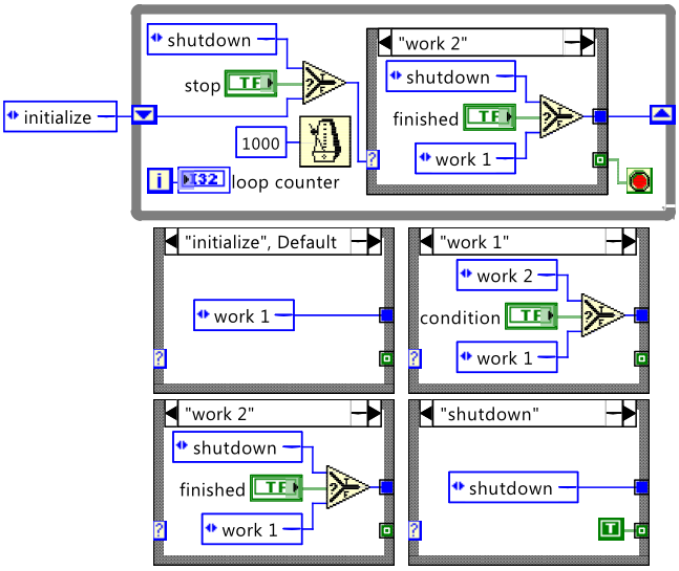


Figure 24: State machine block diagram example. In LabVIEW the states are represented by a stacked conditional structure. For illuatration purposes, the states have been placed adjacent to each other in this image.

6.2 National Instruments Distributed System Manager

To expand the concept of data flow, National Instruments has included the Distributed System Manager (DSM) enabling the access of shared variables throughout a network to allow for easy exchange of values between several physically separated machines. The DSM is a computer process management software running permanently in the background once activated. It serves as a host for user-created sub-processes, each reserving memory for shared variable management.

The RILIS monitoring VIs rely heavily on this concept as shown in section 2: *Motivation*, figure 3 on page 3. The numerous devices of the RILIS setup such as wavemeters, lasers, water chillers, and sensors mentioned in section 4: *Hardware Setup* are distributed in various locations around the laser room installation and in the ISOLDE experimental hall.

As described in the following section *RILIS Monitoring and Control Programs*, a small dedicated readout VI was programmed for each of the devices. These readout VIs are run on the computer the device is connected to. For network attached devices, the server computer described in section 4.10: *Server Computer* was designated to perform the readout. The values acquired are published either to a locally running instance of the DSM on the respective computer or also to the RILIS server, also running the DSM. Figure 25 displays a screenshot of the tree of subprocesses running in the DSM on the RILIS server. Each process hosts a number of shared variables created for a specific device. All variables pictured can be accessed from any point in the technical network (TN) using either the DSM front panel application for monitoring purposes or other VIs. This extraction is performed by specifying the network address in a shared variable node on the block diagram. The data can then be processed for display or further calculations. For example, an acquired wavenumber value can be combined with the Faraday readout data to generate a correlation plot serving for tuning purposes.

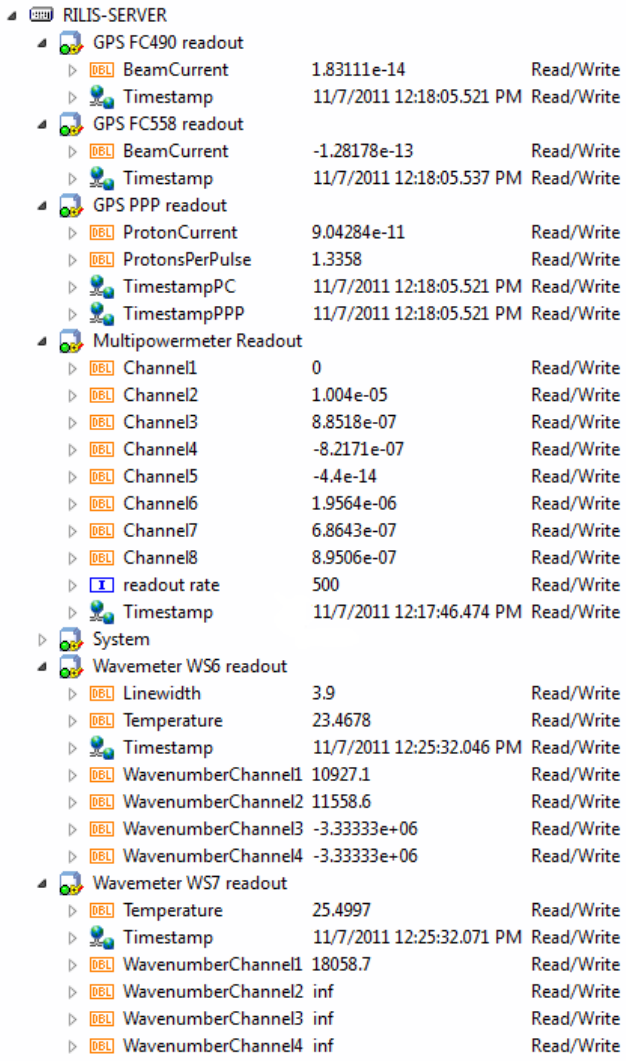


Figure 25: Screenhot of DSM process tree and shared variables. For each device exists a subprocess hosting shared variables.

This concept has proven to be very useful during the setup process of the Ti:Sa laser and normal RILIS operation throughout the on-line running period.

6.3 RILIS Monitoring and Control Programs

For commissioning and usage of all available and newly added devices in the RILIS laser room a set of LabVIEW driver and interaction/readout VIs has been developed. They will form the monitoring, control, and measurement system and are subject to constant improvement and expansion. As mentioned in section 2: *Motivation* the wavelength stabilization software package resembles a part of the whole RILIS monitoring system which already extends towards automatized control.

6.3.1 Wavemeter Readout

As mentioned in section 4.3: *Wavemeters*, two of the wavemeters used in RILIS operation were built the same manufacturer and thus make use of a similar proprietary software. This software handles the communication with the device and also performs all necessary calculations needed to determine the wavelength from the measured interferometers. Furthermore, the software is used to control all parameters and functions of the wavemeter. The wavelength, wavenumber and additional data such as interferometer fringe patterns are displayed in a graphical user interface. The acquired values can also be recorded to a file and be displayed as a long term graph. However, to link the measured data to additional readouts and enable network accessibility a LabVIEW interface was required. To interface to the wavemeter, the manufacturer provides a dynamic link library (dll) containing all relevant access functions listed in a detailed programmer's manual. This dll is loaded into memory once the wavemeter software has been launched. The calculation results can then be accessed by utilizing a *call library function node* placed on the block diagram of the readout VI. However for correct

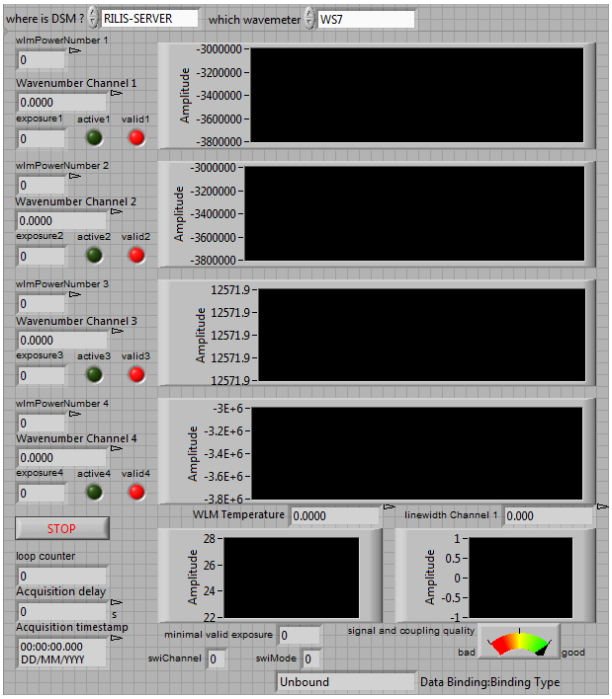


Figure 26: Wavemeter readout VI front panel. Each channel value is displayed numerically on the left and graphically on the right. Additional informational values regarding the current status are displayed by indicators on the lower part.

wavenumber value acquisition the original program must be running. This has to be taken into account when using the stabilization. The front panel of the wavemeter readout VI is pictured in figure 26. The VI supports the simultaneous display of all four channels as well as providing additional information concerning the current status such as temperature and channels in use. All four wavenumber wavenumber values are constantly published to the DSM as described in section 6.2: *National Instruments Distributed System Manager* The VI can also be easily adapted to be used with either of the Highfinesse wavemeters as LabVIEW automatically updates the library call dependencies when the VI is opened.

6.3.2 Faraday Cup and Protons per Pulse Readout

For an ISOLDE experiment which used the RILIS to perform in-source resonance ionization spectroscopy of Astatine isotopes, it was necessary to display the ion current in respect to the applied laser wavelength. The ion current was measured using an ISOLDE Faraday cup. For normalization of the signal, this value had to be registered alongside the proton beam intensity. Both of these values are available within the established ISOLDE infrastructure and Front-End Software Architecture (FESA) framework. FESA is an equipment software running on Front-End computers connected to the acquisition hardware and represents an unified interface to a control middleware [13]. It can be accessed by installing the Rapid Application Development Environment (RADE) architecture described in [14]. This adds FESA support as a set of software libraries to the block diagram context menu of LabVIEW. The use of these functional nodes allows direct and subscribed access to the live acquisition values. Dedicated readout VIs were programmed to publish the values to the DSM. Figure 27 displays a screenshot of the readout front panel for the Faraday cup. Previously to the usage of this VI it was possible to only monitor the beam current history on a dedicated PC running Linux and the Java-based ISOLDE control software.

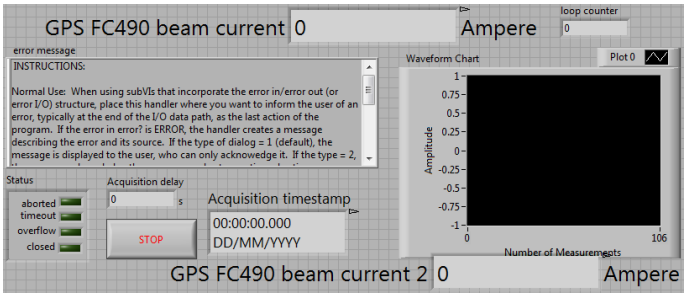


Figure 27: Faraday cup readout front panel. The current readout value history is displayed in the graph on the right. On the left side of the window status and debug messages are displayed.

6.3.3 LabJack Temperature Sensor Readout

The data acquisition functions of the LabJack device described in section 4.4: *LabJack Data Acquisition Device* can be accessed in a similar way as described in the previous section for the FESA framework. The manufacturer JabJack corporation provides a set of device drivers which are integrated into the programming environment. The functional nodes thus added represent sub-VIs which access a dll installed in the operating system to enable hardware communication. The LabJack was in use for communication with the infrared temperature sensor described in section 4.2: *Motorized Etalon Mount* which uses the System Management Bus (SMB) and I^2C protocol according to the data sheet to output measurement values. To achieve this readout an example provided on the LabJack homepage [16] was modified to enable long-term recording of the acquired values.

6.3.4 Powermeter Readout

To monitor the changes in laser power the RILIS is equipped with a set of different powermeters introduced in section 4.8: *Powermeters*. As a step towards extended automated monitoring a readout VI for the Gentec TPM-300 powermeter type has been programmed which can be used in parallel for all devices of the same type. This way code can be reused by applying only minimal changes to its DSM variable network publication address. Also, for development of this VI the state machine design pattern was applied to be able to use it as a template for other devices supporting a serial connection.

6.3.5 Arduino-Based RILIS Temperature Readout

The National Instruments Developer Zone Community has released a LabVIEW interface for the Arduino microcontroller [17]. The software provided on this website is also installed as a LabVIEW add-on similar to the device driver libraries described in the previous two sections. Using the functional nodes provided the Arduino pinout can be configured and inputs can be read via an USB or wireless connection. Figure 28 displays the example VIs currently provided with the Arduino interface library. To test this library and execute long term measurements of the RILIS laser room temperature an Arduino controller was used to read out two PT100 temperature sensors as described in section 4.7: *Air Temperature Sensors*. In the readout VI programmed for this

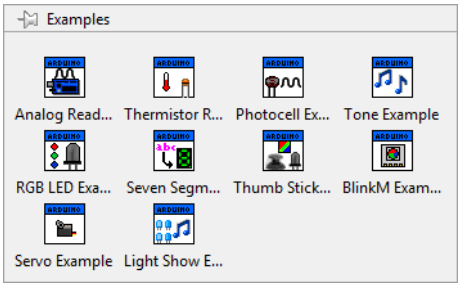


Figure 28: Example VIs available in the LabVIEW interface for Arduino software package.

purpose the Arduino is initialized, the pins are configured and after that the sesor data is read and logged in a continuous loop. The use of the Arduino driver library also opens up the possibility to easily control servo motors, LEDs, and acquire sensor data which will be useful for future projects towards RILIS automatization.

6.3.6 Arduino-Based Etalon Motor Driver

The Arduino prototyping platform is programmed using a textual programming language similar to C code as shown in the example below.

```
#define DO_DIR   2
#define DO_STEP  3
#define DI_UP    4
#define DI_DOWN  7

void setup() {
  pinMode(DO_DIR,  OUTPUT);
  pinMode(DO_STEP, OUTPUT);
  pinMode(DI_UP,   INPUT);
  pinMode(DI_DOWN, INPUT);
}
void loop() {
  if (digitalRead(DI_UP) && !digitalRead(DI_DOWN)) {
    digitalWrite(DO_DIR, HIGH); go();
  }
  if (!digitalRead(DI_UP) && digitalRead(DI_DOWN)) {
    digitalWrite(DO_DIR, LOW);  go();
  }
}
void go(){
  for(int i=0; i < 1000; i++){
    digitalWrite(DO_STEP, HIGH); delayMicroseconds(100);
    digitalWrite(DO_STEP, LOW);  delayMicroseconds(100);
  }
}
```

In the `setup()` block, the pin out of the microcontroller is defined. For improved readability, the use of type definitions is advised. The `loop()` block will run continuously when the controller is powered. Two digital inputs designated for UP and DOWN movement

are read in each iteration. If specifically one input is registered as HIGH, the motor will be moved accordingly for 1000 steps and otherwise do nothing.

The prototype described in section 4.6: *Arduino Prototype* was programmed using the development environment provided in the download section of the Arduino prototyping platform [18]. The corresponding programs are in on-going development and the future prospects for applications for the Arduino are outlined in section 9.1: *Individual Ti:Sa Microcontroller*

6.3.7 Stepper Driver Control Program

For manual software-based control of the Ti:Sa etalon stepper motor from a remote location, a LabVIEW driver VI was programmed. In this VI the state machine design pattern was applied implementing the flow graph pictured in figure 29. This approach generated a clearly arranged program code, despite its increased complexity. The block diagram of this driver is shown in Figure 30. In this view, the stacked case structure encapsulating the states is visible to the right of the center. boolean input nodes shown in green correspond to the directional movement control buttons on the front panel shown in figure 31. If manual control is disabled, the movement values are read from shared variable nodes displayed as blue boxes in the lower part of the case structure. The front panel is designed to mirror the control interface on the front panel of the hardware stepper driver unit described in section 4.9: *Stepper Motor Driver*. To complement the directional control buttons, a display of the current motor position, a switch to enable or disable manual control, and an option to program the number of steps per button press were included. These extra features are particularly useful when performing a manual scan. The control program is capable of sending movement commands to three different axes of the OWIS stepper driver. This enables all three Ti:Sa lasers to be stabilized simultaneously.

6.3.8 Ti:Sa Value Display

To display, log and control the wavenumber values of the Ti:Sa a dedicated viewing VI was developed. It consists of a main window with a large plotting area for showing the measured wavenumber history versus time as the main element. Its size improves readability from a distance to also use the program for monitoring purposes during alignment and manual scanning. Controls for zooming and displaying additional plots simultaneously in the same graph were also added. This allows for displaying averaging, setpoint and stepping position values as well as performing measurements using the cursor tool already integrated in the graph window.

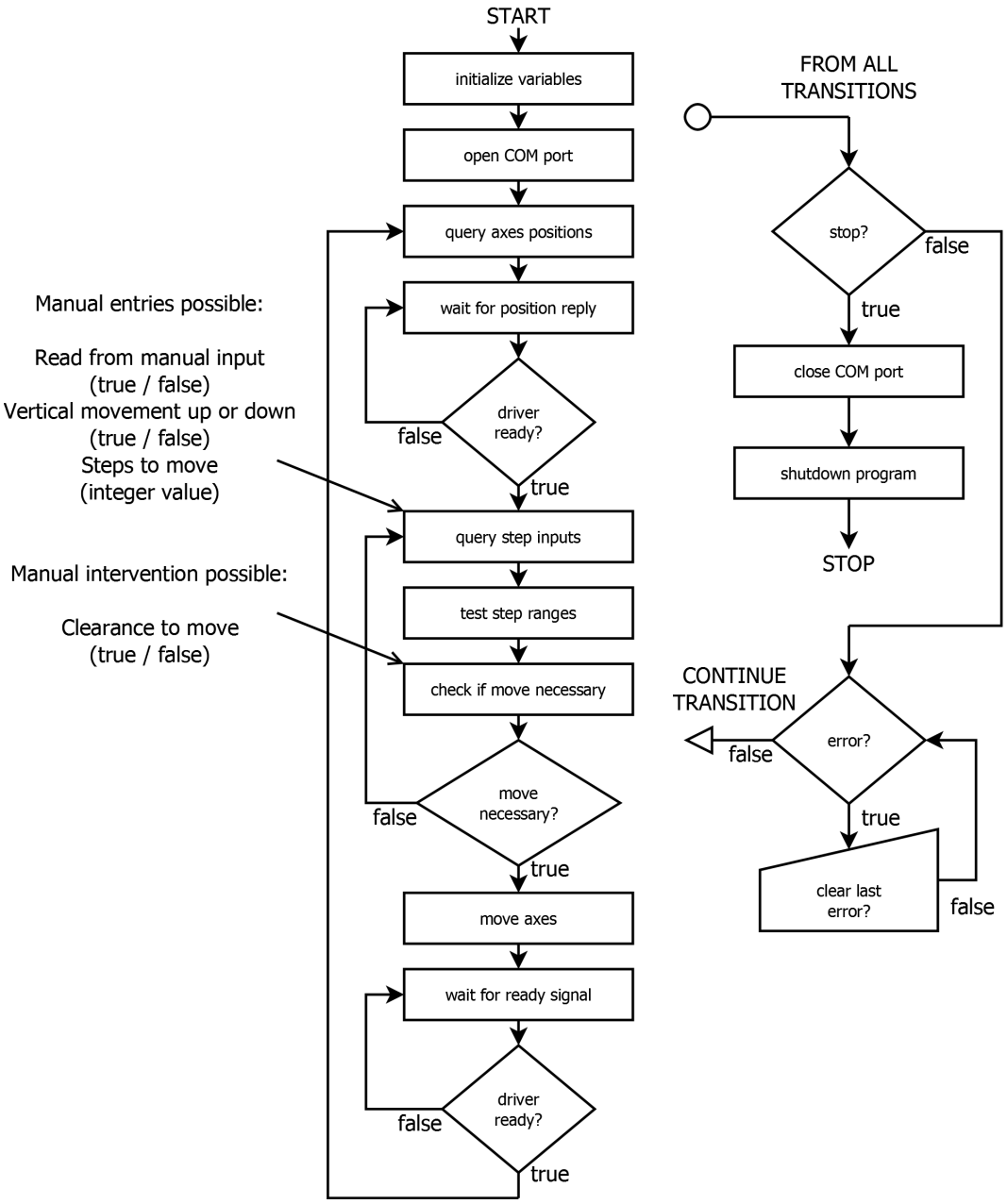


Figure 29: Stepper driver state machine flow graph. Each state is displayed as a rectangular box while the boolean branches are shown diamond shaped. The overall while loop is represented by the leftmost arrow. At each transition the right path of the diagram is executed to allow for error detection and program exit.

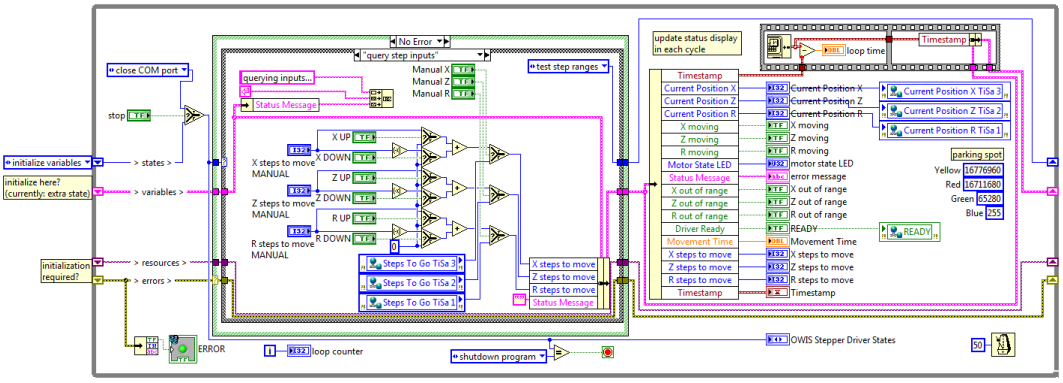


Figure 30: Stepper driver control program block diagram. Despite an increased complexity, the application of the state machine design pattern helped to maintain a tidy code. The corresponding front panel is shown below.

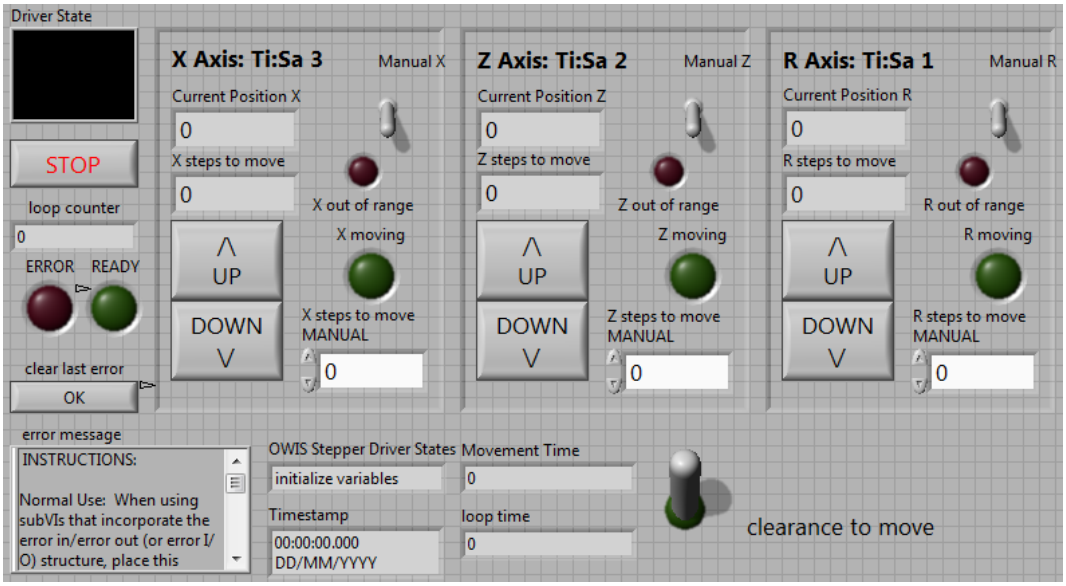


Figure 31: Stepper driver control program front panel. The directional buttons resembling theis physical counterparts are the main inputs of this VI. Additional stepping value inputs have been added as well as position and status indicators. The corresponding block diagram is shown above.

6.4 Wavelength Stabilization LabVIEW Project

After setting up the hardware and programming a basic set of monitoring VIs described in the previous section, selected modules were combined to commission the first component of the automated RILIS control system. The Ti:Sa laser wavelength stabilization software framework is described in the following sections.

6.4.1 Design Concept

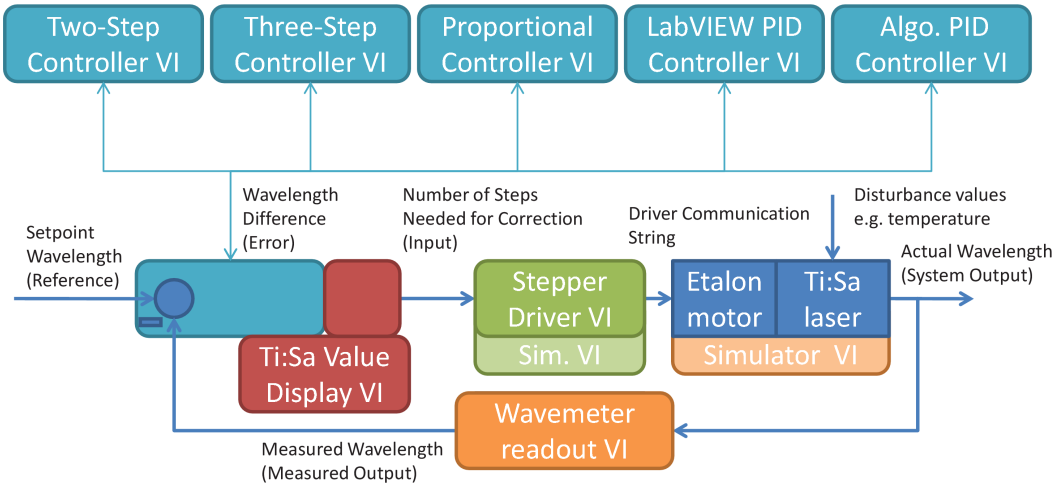


Figure 32: Ti:Sa wavelength stabilization project design diagram. Each module represents a part of a closed feedback control loop. The process variable is measured using the wavemeter and its readout VI. By comparison to the setpoint reference, the difference value is determined. Exchangeable controller VIs calculate a corresponding correction which is used as the input of the stepper driver. By corresponding movement of the etalon a disturbance value correction is performed.

The overall modularity of the RILIS monitoring concept was continued in the software design of the wavelength stabilization. A schematic diagram of the interaction of the modules is displayed in figure 32. The arrangement resembles a feedback control loop with the its constituent elements represented by the corresponding VIs and hardware components. The LabVIEW project consists of different modules each performing a specific task as described in the previous sections. The communication between these modules was realized by the use of shared project variables which are automatically stored and made accessible through the DSM. Each module uses a specific set of input and output variables to define its interface. This can be compared to parameters and return values in text-based programming languages.

For the acquisition of the process value, the wavemeter readout VI saves the current wavenumber value to a shared variable in the DSM. This module is started first and runs continuously to monitor and log the wavenumber values. To display the current wavenumber and to set the setpoint, the Ti:Sa value display VI is used. It can also be used to record and export the value history as well as visualize the current stability. No processing or feedback to the Ti:Sa is performed by this VI. The main components of the stabilization are exchangeable stabilization modules implementing different correction algorithms. These controller VIs read the current wavenumber and setpoint values from the DSM. The algorithm calculates the set step value. This correction value is then stored in the corresponding shared variable in the DSM. The stepper driver control VI reads the correction value and sends the corresponding string to the hardware driver.

Figure 33 shows the VI hierachy created for this project, as well as the shared project variables. These constitute the communication interfaces between the VIs. For development and testing, each module also has a simulation counterpart which does not interact with

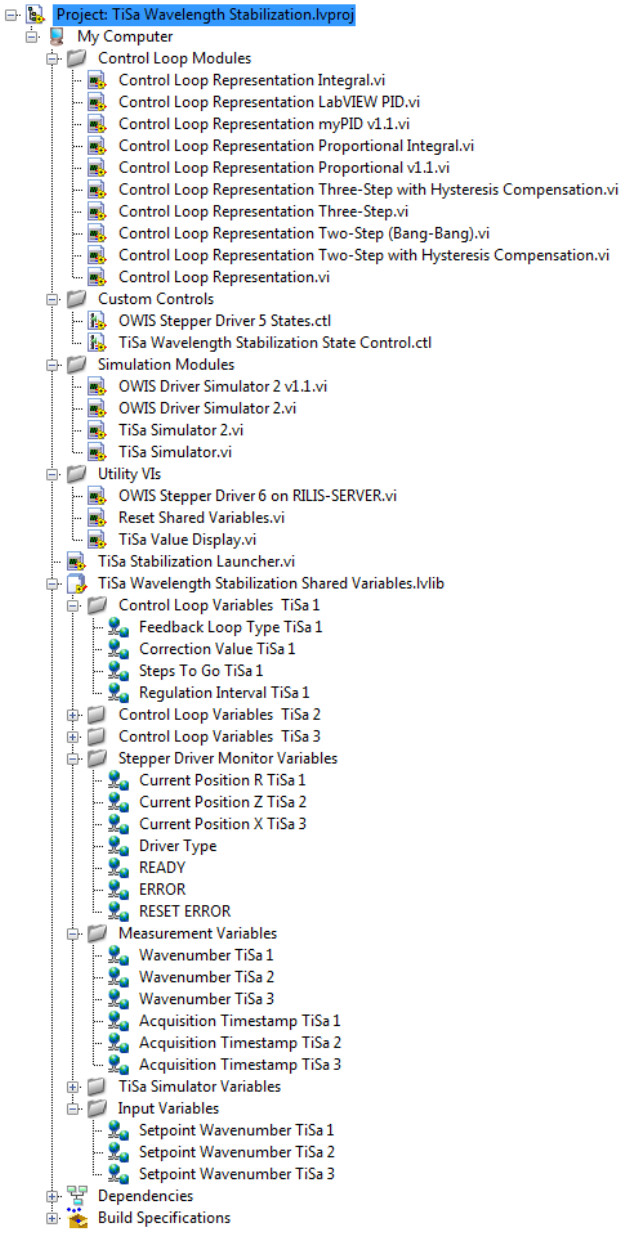


Figure 33: Ti:Sa wavelength stabilization project VI hierarchy. In the upper part, the VIs used for the stabilization are displayed. On the lower part, the shared project variables used as interface an communication elements are shown.

the hardware, but merely creates an artificial signal to check the correct function of the processing modules and data connections in the DSM as well as response times.
Due to the relatively slow communication and processing time described in section 4.9: *Stepper Motor Driver*, the time for one control loop iteration takes ≈ 2 s. As the main task of the software is to correct drifts occurring over the timescale of hours this is deemed sufficient and can be seen as a starting point for improvements in future versions as described in section 9: *On-going Projects and Future Prospects*.

6.4.2 Stabilization Specific Control Loop Modules

The stabilization system can be operated with a set of different control loop modules which implement different processing algorithms. These determine a correction value to send to the stepper motor and also include an option to specify a hysteresis compensation value. The latter value can be changed while the stabilization is running, in case there would be a change of the mechanical setup. In the following paragraphs, five different software control loop modules are described. Three of these controllers were also used in section 7.1: *Test of Stabilizing Loop Modules* to assess the initial functionality of the system and to choose one module for long-term testing.

Two-Point Control Loop Module

Initially, a two-point or on-off control loop module was implemented. The *on* respectively *off* states, commonly associated with switching heating devices, are represented in this case by the movement direction of the stepper motor. The sign of the difference between the setpoint value and the actual value determines the movement direction. The magnitude of the difference does not affect the correction value, instead the amount of steps to go in each iteration can be defined by the user in the front panel of the VI. As a default the stepping value is set to 1 in this control loop module. A schematic wavenumber trend plot, illustrating the functionality of the two-point algorithm implemented in the control loop module is shown in figure 34. Additionally, a hysteresis compensation can be configured so as to be added to the correction value. If the sign of the difference value changes in two consecutive measurements, the hysteresis compensation factor is added, otherwise only the correction value is taken for the steps to go. Utilization of this control loop module resulted in constant motor movement as the setpoint could never be exactly reached. Rather the process value oscillates around the setpoint as

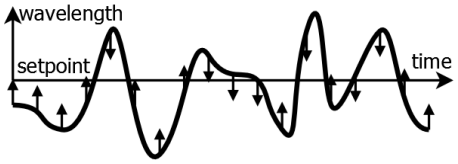


Figure 34: Two-point controller behavior diagram. The directional motor movement at each correction instant is indicated by arrows originating from the wavelength plot.

is shown in the test of this control loop module described in section 7.1: *Test of Stabilizing Loop Modules*.

Three-Point Control Loop Module

The three-point control loop module was an extension the two-point implementation mentioned above with a programmatic dead zone within two barriers above and beyond the setpoint. To continue with the heater example mentioned above: the actions *heat*, *do nothing*, and *cool* correspond to *move up*, *do nothing*, and *move down* in this application. If the difference value lies within the dead zone, no correction value is applied. The corresponding schematic wavenumber trend plot is illustrated in figure 35. Utilization of this control loop module reduced the strain on the motor but also defined an uncorrected drift zone. Furthermore the trend was estimated to show less severe oscillations due to the "free running" of the Ti:Sa laser in the dead zone. Outside the barrier values this VI acted similar to the two step controller described above, also incorporating the hysteresis compensation feature.

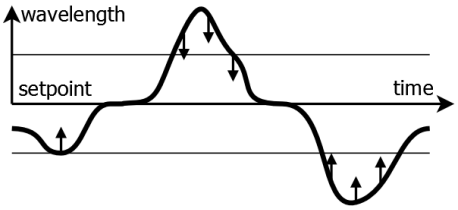


Figure 35: Three point controller behavior diagram. If the wavelength is within a certain range of the setpoint, no movement takes place.

Proportional (P) Controller

The implementation of the proportional controller can also be viewed as an extension of the two-point controller. However in this algorithm the magnitude of the difference value also determines the magnitude of the correction value through the multiplication of the difference value with the proportional gain factor. The utilization of this control loop module promises two advantages: Firstly, stronger deviations will be corrected more quickly than with the two control loop implementations mentioned above due to the application of the proportional gain factor. Secondly, an inherent "dead zone" is created by rounding of the calculated correction value to an integer step value. For this reason constant movement was also avoided by utilization of this control loop module. Figure 36 illustrates a schematic wavenumber trend created by the use of this control loop module. The following exemplary calculation of the correction value $e(t)$ determines the resulting dead zone for a P-gain value K_P of 50:

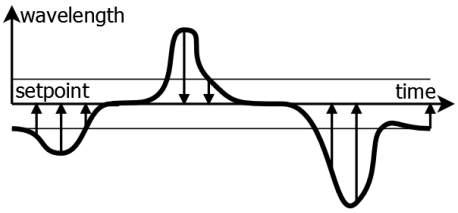


Figure 36: Proportional controller behavior diagram. The zone where no correction is executed is determined by the P-gain factor.

$$
\begin{aligned}
0.5 &> K_P \cdot e(t) \\
e(t) &> \frac{0.5}{K_P} \\
e(t) &> \frac{0.5}{50} \\
e(t) &> 0.01
\end{aligned}
$$

At a P-gain value of 50 the wavenumber difference must be greater than 0.01 for one step to be moved. According to section 3.1: *Requirements Analysis*, this dead zone is well within the range of acceptable values of 1% signal loss (0.0189 cm^{-1} wavenumber difference).

Proportional Integral (PI) Controller

In an attempt to narrow the dead zone for the proportional controller mentioned above, its implementation was extended by an integral algorithm implementation. In addition to the magnitude of the difference value, the integrating part also takes into account the time elapsed between each correction feedback. This implementation applies the concept of calculating a Riemann sum from the acquired values and the time elapsed between each iteration. This is done in the following manner: The difference value is multiplied with the elapsed feedback loop iteration time. Before propagating the resulting value to the next loop iteration, the integral value of the last iteration is added to the equation thus producing the Riemann sum. This sum value is then multiplied with the user-definable integral gain value. The resulting number is finally added to the proportional correction value, creating a combined proportional and integral correction value. For this reason, deviations smaller than the dead zone of the proportional controller add up over time, resulting in eventually one correction step towards the setpoint. As this algorithm is time-dependent, an integral sum value of 0 was defined as startup value was well as a reset-to-zero option has been included. Initial tests showed fluctuations in readout time resulting in sensitive and difficult selection of the integral gain value. Because of this the utilization of the integral part in the stabilization setup was postponed to future tests. With the integral gain value set to zero, this PI control loop module acts as a proportional control loop module. To incorporate the requirement of error tolerance towards mode jumps and readout errors, as defined in section 3.1: *Requirements Analysis*, a "spike threshold" value was included, to enable users to define a specific stabilization operation range of the stabilization. If the absolute difference value exceeds this threshold value, no intervention is performed by the wavelength stabilization system. Figure 37 shows the front panel of the control loop

VI which was used in *proportional* mode for the test described in section 7.2.2: *Stabilized System*. In addition to the functional user interface elements, several decorative elements have been added to create a visual resemblance to a feedback control loop. On the right part of the front panel the controlled Ti:Sa laser is chosen. On the left side a spike threshold value can be entered in cm^{-1}. The central part of the front panel contains the inputs for the proportional and integral gain factors as well as an activation switch to allow for disabling the output by setting the *steps-to-go* variable to zero. The button below the activation switch serves as a reset button for the integral implementation. The corresponding Block diagram is shown in figure 38. The conditional structures in the upper part of the diagram contain the shared variables used for inter-VI communication as described in section 6.4.1: *Design Concept*. The left box corresponds to the input values, the right box corresponds to the output values. On the left side of the block diagram, the user interface input elements are shown. The output and display elements are grouped on the right part. In the central part of the block diagram the algorithm implementation is shown.

Proportional Integral Derivative (PID) Controller

Additionally to the controller VIs described previously, two different proportional integral derivative (PID) controllers were also implemented in a similar manner into separate VIs. One was taken from the built-in LabVIEW control and automation package and one implemented the position algorithm ("Stellungsalgorithmus") described in [5]. Initial tests yielded unsatisfying results due to uneven loop iteration and stepper motor movement times and also required extensive tests to determine the PID gain values. When run, the integral and derivative values of the stabilization, as well as the LabbVIEW PID functional node, had to be frequently reset. For this reason the utilization of these feedback loop controllers has been postponed to future test series.

The applicability of the different algorithms were evaluated in initial short-term tests and the results are documented in section 7.1: *Test of Stabilizing Loop Modules*.

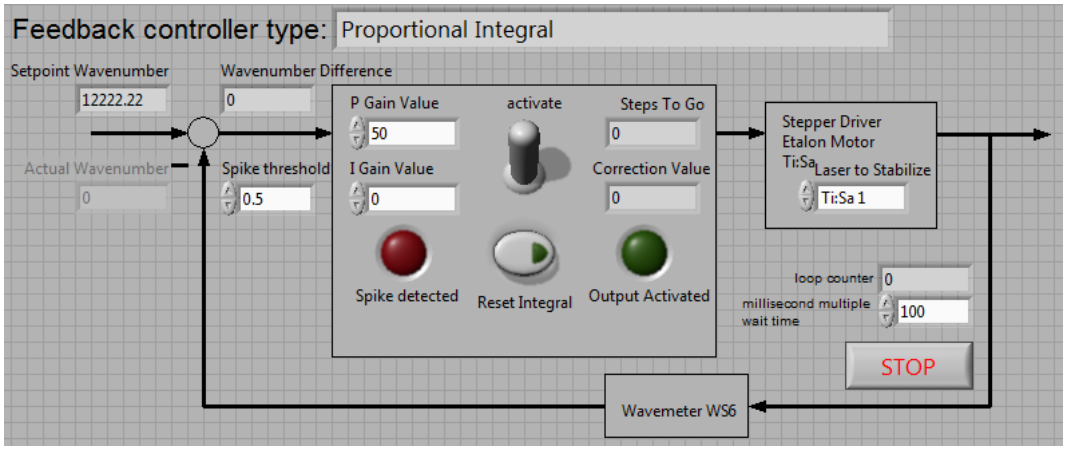


Figure 37: Proportional integral controller front panel. All elements are arranged to resemble a feedback control loop. The controller parameters can be entered using the interface elements in the central part. The corresponding block diagram is shown below.

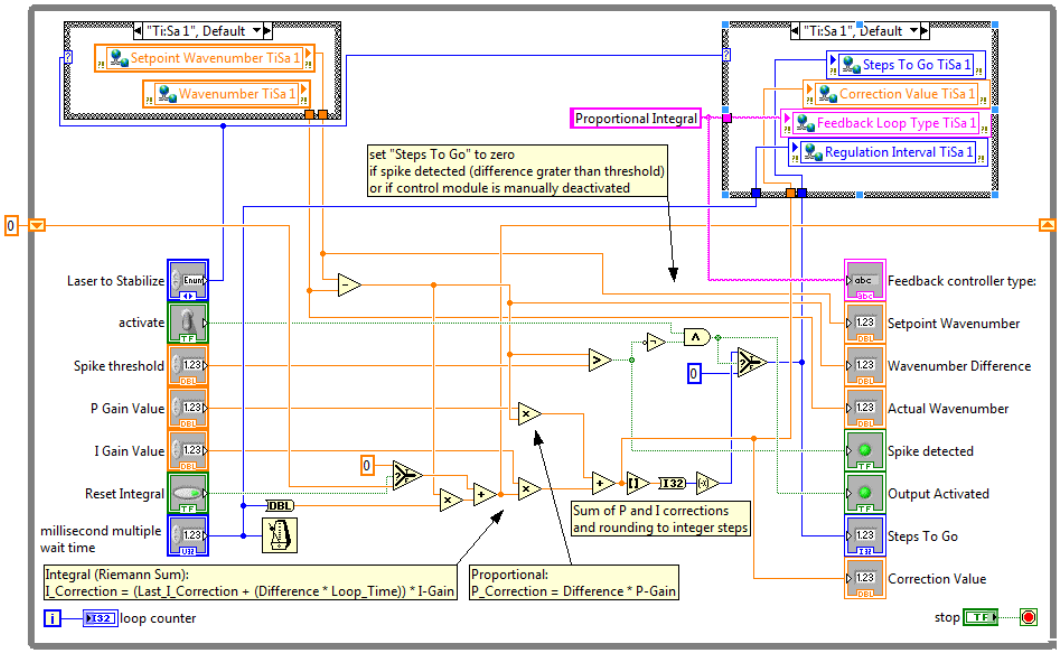


Figure 38: Proportional integral controller block diagram. All functional elements have been grouped according to their respective association: Input elements are shown on the left while output elements are shown on the right. The central part contains the algorithm implementation.

7 System Evaluation

The system evaluation was conducted in two phases: during the first assessment, the correct interaction of the VIs was verified. Additionally, the temporal performance of the stabilization was determined and the initial parameter settings of three different control loop algorithms were investigated. The results are illustrated in the following section 7.1: *Test of Stabilizing Loop Modules*.

Subsequently, a long term wavenumber measurement utilizing the most successful control loop module of the initial test was performed. The accordant results are documented in section 7.2: *Long-Term Wavenumber Measurements* followed by a discussion of the system performance in section 7.3.

A reference measurement of an unstabilized system similar to those described in section 5.1: *Temperature Measurements*, was executed prior to both evaluations.

7.1 Test of Stabilizing Loop Modules

The initial test of the stabilizing loop modules served as a software functionality examination. Furthermore, to assess the performance of the control loops and to determine a candidate for the long term measurement, three different control loop modules were tested over a period of 30 minutes each.

Additionally, the plot shown in figure 39 illustrates the behavior of the unstabilized system as a reference to the subsequent tests. The horizontal axis designates the time elapsed in minutes. The wavenumber values are shown in black and correspond to the left scale. Over the course of the measurement the wavenumber values show a clear positive trend, reciprocal to the temperature values shown in red and corresponding to the right scale. This is consistent with the observations already outlined in section 5.1: *Temperature Measurements*. The fluctuation observable in the wavenumber values can be attributed to the readout tol-

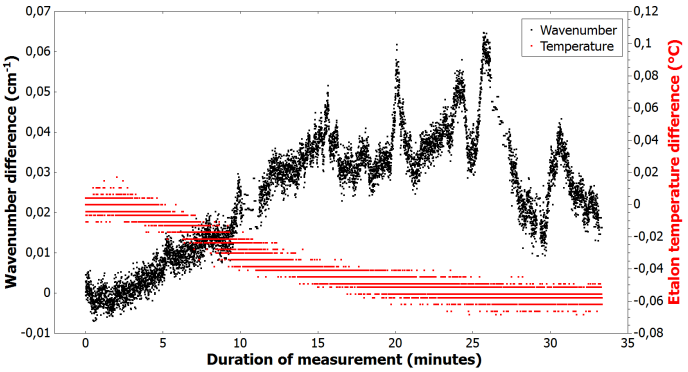


Figure 39: Wavenumber and temperature plot of the unstabilized Ti:Sa laser. The wavenumber values are shown in black and correspond to the left scale. The temperature values are shown in red and correspond to the right scale. The bottom scale designates the time elapsed.

erance of the wavemeter, as the distinct levels of the temperature values originate in the limited resolution of the temperature sensor.

7.1.1 Two Point Controller

The behavior of the two point controller is illustrated in figure 40. The course of the wavenumbers, shown in black, confirms the general ability of this controller type to correct for wavenumber deviations. However, spikes are visible as the recording progresses. These can be attributed to the constant movement of the motor according to the implementation described in section 6.4.2: *Stabilization Specific Control Loop Modules*. The system permanently strives to correct for minimal deviations from the setpoint, thus displaying the accordant oscillation. The sections where no oscillations occur can be attributed to the mechanical hysteresis of the mirror mount as the motor movement does not cause any movement of the etalon mount.

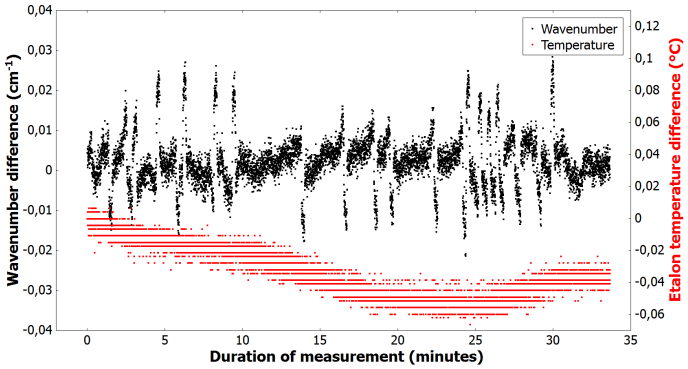


Figure 40: Wavenumber and temperature plot of the two point stabilized Ti:Sa laser.

7.1.2 Three Point Controller

The wavenumber plot in figure 41 displaying the behavior of the three point controller reveals a similar course as the plot of the two point controller; with the exception that the spikes are more stretched out. This can be described by the dead zone implemented in the three step controller. The number of corrections is reduced due to the motor not moving if the

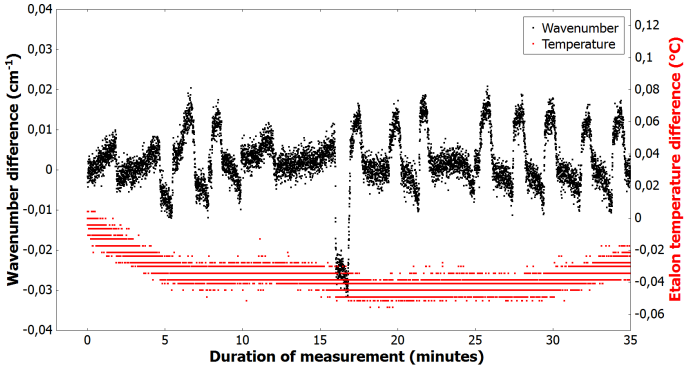


Figure 41: Wavenumber and temperature plot of the three point stabilized Ti:Sa laser.

acquired wavenumber value is
within the threshold range defined in the implementation in section 6.4.2: *Stabilization Specific Control Loop Modules*.

7.1.3 Proportional Integral Derivative Controller

With the goal of combining the requirements of quick wavenumber correction and low motor strain, the PID controller module was also tested and the results are shown in 42. The plot reveals a strong fluctuation in the wavenumber, as well as a runaway value eight minutes after the start of the measurement at ①. These strong fluctuations are attributed to a disadvantageous choice of integral and derivative gain values originating from the feedback speed of the motor driver described in section 4.9: *Stepper Motor Driver*. As a consequence of this test, a control module not dependent on the feedback time was chosen for the long term tests.

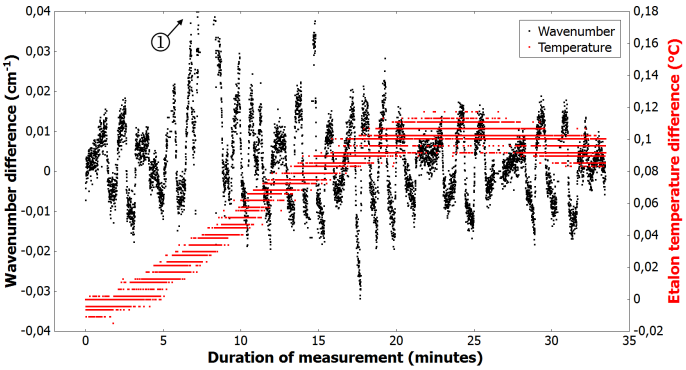


Figure 42: Wavenumber and temperature plot of the PID stabilized Ti:Sa laser.

All three loop implementations have shown the basic correction capability for wavenumber deviations. The best exemplary results were displayed by the two point controller with the exception of constant motor movement and spikes in the wavenumber. The choice for a long-term test was to use a purely proportional controller, as it represents a compromise between permanent correction and creating a dead zone as described in the implementation in section 6.4.2: *Stabilization Specific Control Loop Modules*. In the following section the long time performance of this controller will be discussed and quantified.

7.2 Long-Term Wavenumber Measurements

Following the short-term tests, the use of the proportional controller with a gain value of $K_P = 50$, as calculated in section 6.4.2: *Stabilization Specific Control Loop Modules*, was found to be the most promising choice for a long-term evaluation as it does not depend upon the feedback loop time. To determine the capability of the stabilization program to provide a stable wavelength over an extended period of time, for example an eight hour shift, a

long-term measurements of the wavenumber and temperature values were performed. To determine the use of the motor, the step count was also recorded during the use of the proportional stabilization. The resulting long-term plots are displayed in section 7.2.2: *Stabilized System*. To evaluate the system and to quantify the improvement, a relevant set of comparable data was created: The wavenumber of the Ti:Sa laser and the temperature at the RILIS laser room air conditioning unit (A/C) outlet was recorded as it is located directly above the system as described in section 4.7: *Air Temperature Sensors*. The stabilization was enabled and disabled, each over the course of more than eight hours. As it is located in a low temperature fluctuation zone at the opposite wall of the RILIS laser room the WS/7 wavemeter was used to minimize the influence of the systematic error. The WS/7 also yields higher accuracy than the WS/6 as described in the manual. The temperature data was acquired using the PT-100 temperature sensors read out by an Arduino microcontroller as described in section 4.7: *Air Temperature Sensors* and the respective readout VI described in section 6.3.5: *Arduino-Based RILIS Temperature Readout*

7.2.1 Unstabilized Reference

In the upper plot of figure 43 the values of the Ti:Sa wavenumber difference are displayed on the vertical axis. The lower plot shows the RILIS A/C outlet temperature values on the vertical axis. Both series were recorded simultaneously over the course of 13 h which is plotted on the horizontal axis. A similar oscillation indicating a drift of the wavenumber value in accordance to the measured temperature is clearly visible. To determine and to quantify a correlation between the temperature and wavenumber changes, the data was also plotted as wavenumber versus time shown in figure 44.

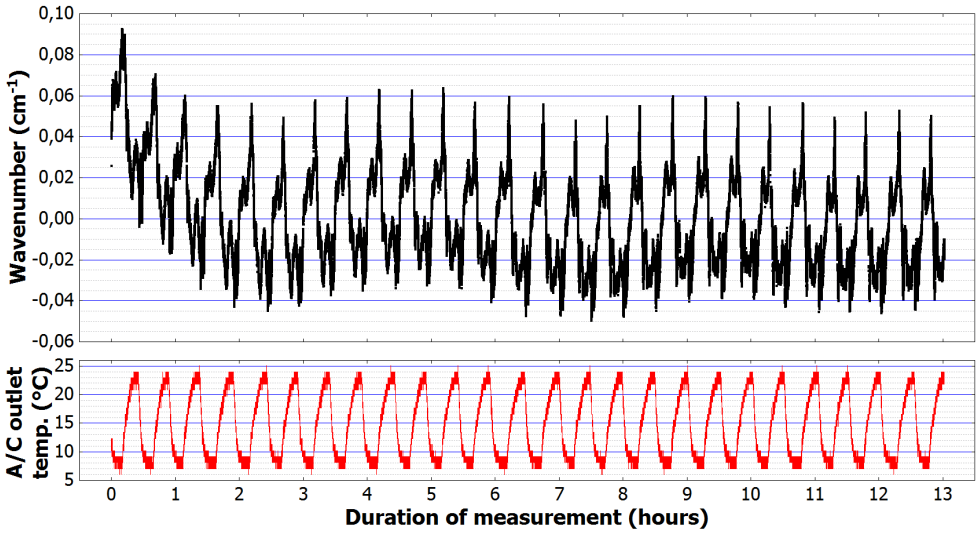


Figure 43: Plot of the Ti:Sa wavenumber and RILIS A/C outlet temperature when not stabilized. The upper plot shows the wavenumber values in black, the lower plot shows the temperature values in red. The horizontal axis represents the running time.

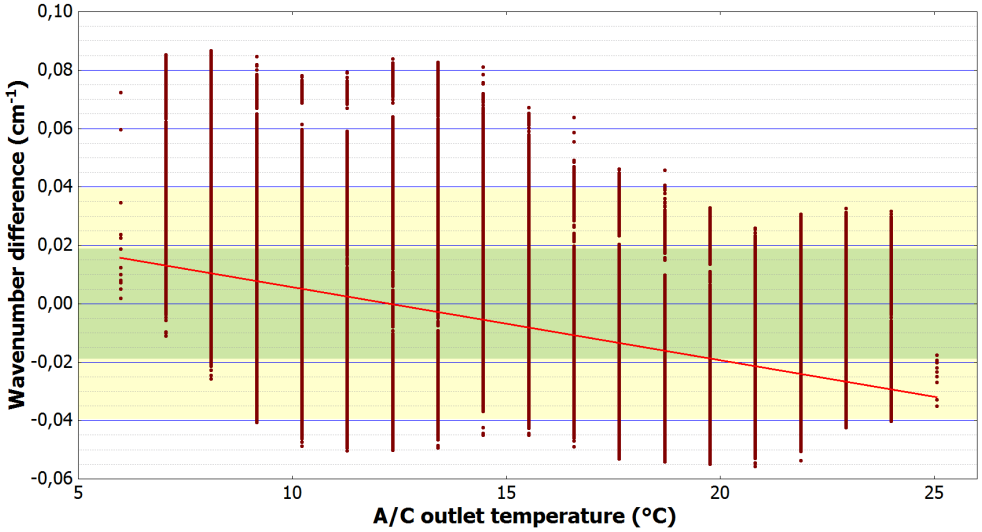


Figure 44: Plot of the Ti:Sa wavenumber versus the RILIS A/C outlet temperature when not stabilized. The wavenumber values are denoted on the vertical scale, the temperature values are denoted on the horizontal scale. Additionally, a linear fit was applied to the data set, resulting with the red line. The green and yellow overlays represent the zones of 99% and 95% resonance deviation zones defined in section 3.1: *Requirements Analysis*

7.2.2 Stabilized System

For the stabilized system, the wavenumber of the Ti:Sa laser was measured as well as the RILIS A/C outlet temperature and the rotational position of the stepper motor. The accordant data is shown in the plots in figure 45. The measurement was conducted over the course of 18,5 h. At the time indicated by the the marker ①, the measured temperature settled to a stable value. Prior to this time a similar oscillation can be observed in all tree plots. After settlement of the temperature, a correlating oscillation can be observed for the wavenumber and the stepper position values. Overall the long time plot with the stabilization activated shows a linear horizontal structure in the wavenumber during the whole measurement. Similar to the correlation plot figure 44 in section 7.2.1: *Unstabilized Reference*, the values of the measured wavenumbers and temperatures have been plotted in figure 46 corresponding to the y and x axis respectively to allow for a comparison of both measurements. The evaluating discussion can be found in section 7.3: *Discussion* quantifying the differences between both measurements. Figure 47 displays a screenshot of the main graph found in the Ti:Sa value display described in section 6.3.8: *Ti:Sa Value Display*. The white line displays the acquired wavenumber values while the red line displays the setpoint with the corresponding scale on the vertical axis. Both values are displayed in respect to the time denoted on the horizontal axis. As a test of the required **go to** ability of the system defined in section 3.1: *Requirements Analysis* the setpoint was changed from 12571.300 cm^{-1} to 12571.350 cm^{-1} at the time marked with ① and set back at the time marked with ②. With the proportional controller configured to apply a gain value of 50 it takes $\approx$ 4 s to compensate for the mechanical hysteresis and $\approx$ 3 s to execute the etalon movement needed. This adds up to a total of $\approx$ 7 s to reach the desired setpoint.

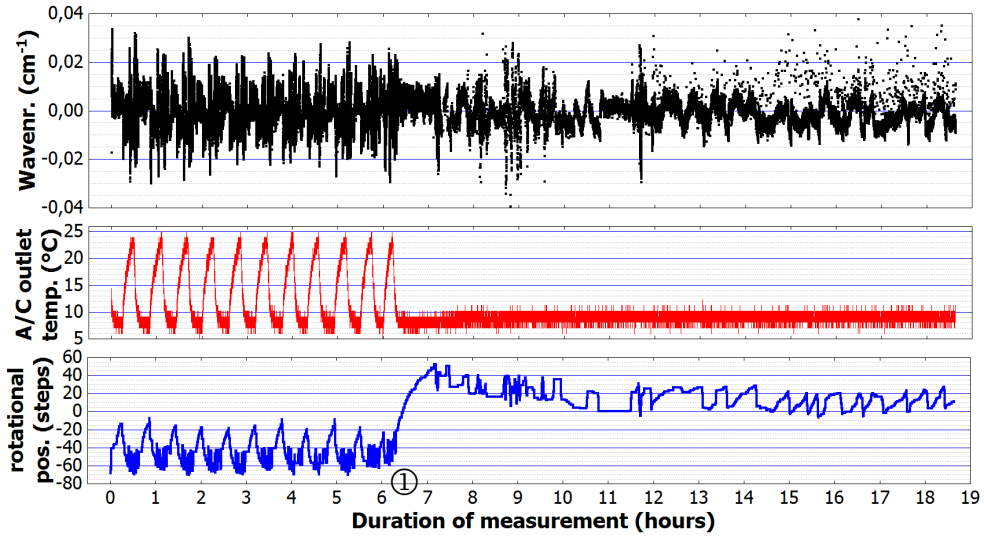


Figure 45: Plot of the Ti:Sa wavenumber, RILIS A/C outlet temperature and rotational stepper motor position when stabilized. The upper plot shows the wavenumber values in black, the middle plot shows the temperature values in red. The lower plot shows the rotational position in blue. The horizontal axis represents the running time.

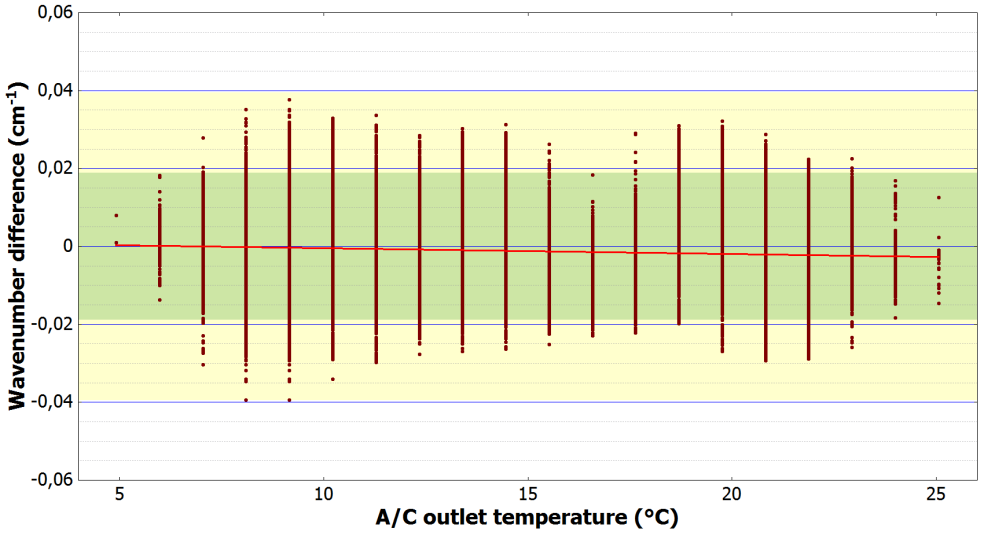


Figure 46: Plot of the Ti:Sa wavenumber versus the RILIS A/C outlet temperature when stabilized. The wavenumber values are denoted on the vertical scale, the temperature values are denoted on the horizontal scale. Additionally, a linear fit was applied to the data set, resulting in the red line. The green and yellow overlays represent the zones of 99% and 95% resonance deviation zones defined in section 3.1: *Requirements Analysis*

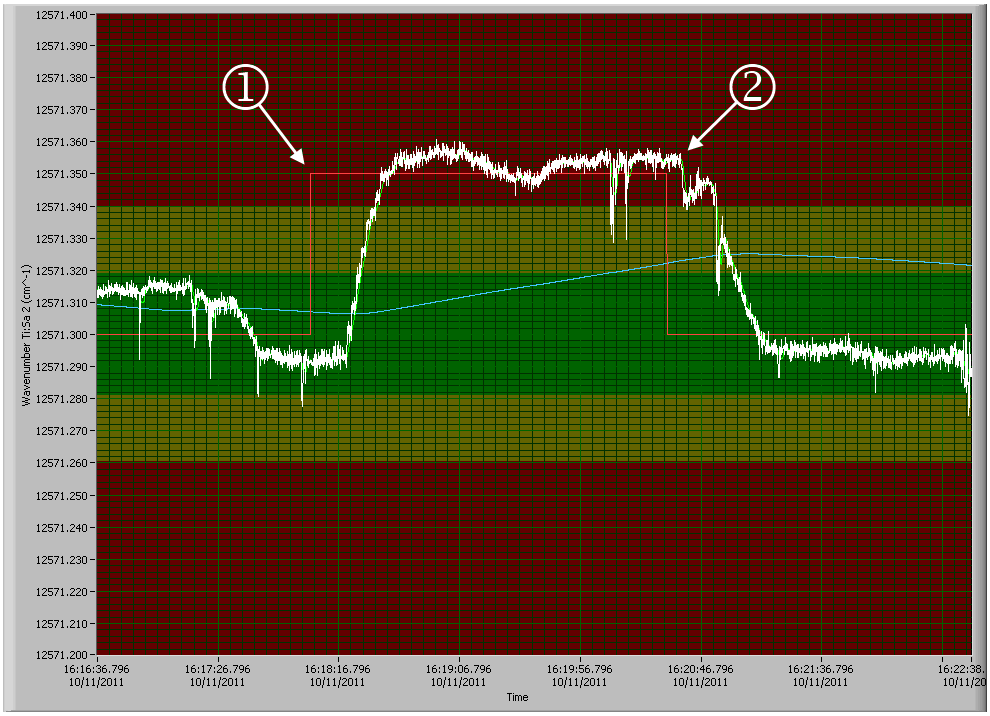


Figure 47: Ti:Sa value display main graph screenshot. The white line represents the acquired wavenumber values. The green, yellow and red zones represent the evaluation zoned defined in section 3.1: *Requirements Analysis*. The red line represents the setpoint value while the blue line visualizes the customizable average mean value described in section 6.3.8: *Ti:Sa Value Display*.

7.3 Discussion

Evaluation of the unstabilized system determined the average mean value of the wavenumber to 12571.934 cm^{-1} with a standard deviation of $\pm$ 0.0244 cm^{-1}. Applying a linear fit to the data plotted in figure 43 yields a linear dependency of -0.00250 $cm^{-1} \cdot K^{-1}$. For the stabilization the evaluation of the complete data set revealed that 0.22% of the acquired wavenumber values showed a difference of more than 43 cm^{-1} from the setpoint. As determined in section 3.1: *Requirements Analysis*, the stabilization system must not correct deviations of more than 100 steps i.e more than 1 cm^{-1} according to section 5.2: *Hysteresis Measurements*. These deviations can be accounted to mode jumps of the Ti:Sa laser. Taking this into account the evaluation was based on the remaining 99.78% of acquired values while the stabilization was activated. Evaluation of the Ti:Sa laser stabilized at a setpoint of 12571.300 cm^{-1} determined the average mean value of the wavenumber to 12571,2995 cm^{-1}

with a standard deviation of ± 0,00642 cm^{-1}. Applying a linear fit to the data plotted in figure 43 yields a linear dependency of -0.0001498 $cm^{-1} \cdot K^{-1}$. Comparison of the evaluated standard deviations of the unstabilized and stabilized Ti:Sa laser leads to a quantified stability improvement by a factor of 3.807. Comparison of the temperature dependency slopes of the unstabilized and stabilized Ti:Sa laser leads to a quantified reduction factor of temperature dependency of 16.707. Remaining spikes in the wavenumber data can be attributed to the mechanical hysteresis as no change in etalon position takes place if the direction of the motor changes in two consecutive corrections as shown in section 5.2: *Hysteresis Measurements* and section 7.1: *Test of Stabilizing Loop Modules*. When choosing a stabilization algorithm, a trade-off between drift correction requirements and constant motor movement had to be found. At 6 h measurement intervals the rotational position values clearly show a reciprocal correspondence to the error value temperature, while the wavenumber value remains within the allowed value range representing 99% signal strength of the reachable maximum established in section 3.1: *Requirements Analysis*. This corridor has to be re-evaluated according to the respective desired element resonance.

	average mean	standard deviation	temperature dependency
unstabilized	12571.934 cm^{-1}	± 0.0244 cm^{-1}	-0.00250 $cm^{-1} \cdot K^{-1}$
stabilized	12571,2995 cm^{-1}	± 0,00642 cm^{-1}	-0.0001498 $cm^{-1} \cdot K^{-1}$

Table 1: Table of measurement results for unstabilized and stabilized Ti:Sa laser

8 Conclusion

During the commissioning of the ISOLDE RILIS Ti:Sa laser system, the requirements for the adaptable wavelength stabilization have been defined. The corresponding documentation is outlined in section 3.1: *Requirements Analysis*. Based on subsequent measurements, the emitted wavelength was recognized as being dependent upon the corresponding temperature and has been documented in section 5.1: *Temperature Measurements*, as well as section 7.2.1: *Unstabilized Reference*. Following the installation and setup of computerized actuators into the Ti:Sa, which is described in section 4: *Hardware Setup*, the wavelength stabilization routine was programmed by implementing a modular feedback control loop program using the integrated software development platform LabVIEW. The resulting software design process and the Virtual Instruments created are described in section 6: *Software Setup*. To estimate the performance of the stabilization, a system evaluation was conducted and is documented in section 7. According to the evaluation results gathered, the Ti:Sa wavelength stabilization, at its current state of hard- and software development, meets the requirements of both use cases **go to** and **stabilize** which were defined in section 3.1: *Requirements Analysis*. Also, the stepper driver control program described in section 6.3.7 was used to perform manual resonance **scans** for use in RILIS spectroscopy.

The results obtained and discussed in section 7.3 show, that the use of the stabilization is of vital importance as the average stability as defined in section 3.1: *Requirements Analysis* can be increased by a factor of 3.8 and the temperature correlation of the system reduced by a factor of 16.7. Due to these enhancements, the ionization efficiency of the Ti:Sa laser beam is optimized. This permits RILIS, and other experiments conducted in scientific cooperation, to operate at significantly improved cost efficiency due to higher yield rates and reduced manual intervention. Additionally, the current system opens the possibility for developing largely automated data acquisition programs and control routines allowing for the next steps towards an extended RILIS monitoring, alert and operation system. The development of a LabVIEW-based data acquisition program, interfacing to both RILIS infrastructure and the ISOLDE sensor network, has already been used to determine the inonization potential of astatine, which is subject to the thesis of S. Rothe. Also the combined usage of the ISOLDE infrastructure interface VIs and legacy systems requiring dedicated serial connections has been successfully established, proving the flexibility of the concept outlined in section 2: *Motivation*. In the following section 9: *On-going Projects and Future Prospects*, the next goals towards improved maintenance of RILIS, as well as enhanced user support and feedback, will be described.

The design concepts outlined, hardware installations performed and VIs created during this project will continue to assist ISOLDE operators, machine protection and users alike by establishing a foundation towards an largely automated *dual mode RILIS*.

9 On-going Projects and Future Prospects

The developed stabilization system is designed to correct laser wavelength drifts due to gradual temperature changes. Short term fluctuations caused by changing air flow or vibrations create possible instabilities in laser beam pointing. The speed and sensitivity of the stepper motor is not suitable for the required position stabilization. In this case, an upgrade of the mirror mount with piezo actuators can be considered. Such actuators would be able to quickly react to minute position changes and would supplement the spatial component to the spectral stabilization as was already envisaged in section 2.
The existing system is designed to provide a modular framework and a starting basis for further developments and improvements. Changes to the hardware and software can be realized in small steps by altering just singular components at different time periods. Software engineering methods can be applied for RILIS automatization project.

9.1 Individual Ti:Sa Microcontroller

One of the main goals is the provision of independent and simultaneous operation of each Ti:Sa laser whilst using the stepper driver program for different tasks such as stabilization on different wavelengths, resonance scanning or manual wavelength manipulation. To achieve this, one microcontroller should exist per Ti:Sa laser. This should display the current wavelength and report status messages such as internal temperature measurements over the RS-232 interface. A dedicated software control interface running on any computer in the CERN technical computing network would enable each Ti:Sa laser to be operated remotely, therefore limiting the need to physically access the laser room for routine adjustments of the Ti:Sa lasers.

9.2 Interchangeable Software Framework Modules

The software is designed to use different control modules for the stabilization which can be exchanged when the stabilization is running. This option allows for the possibility to test different algorithms, as well as check and improve the overall long term performance. To further increase the reaction time of the stabilization as well as to improve independent operation of serveral Ti:Sa lasers, different driver modules can be written to communicate with different driver types.

9.3 Software Driver for Narrow Band Laser Mode

For narrow band operation, two concerted etalons of differnet thicknesses are combined to produce a narrow laser linewidth. Both etalons have to be kept in a stable relative position

since even a small aberration would result in unwanted multi mode operation and severe power loss. For this mode of operation the respective driver modules will have to be precisely coordinated, and a dedicated narrow band scanning routine developed.

9.4 Software Optimized Ti:Sa Autotuning

By also motorizing the resonator component mounts in two dimensions it will be possible to supplement the intensity stabilization with a laser optimization algorithm. Without automatization the resonator needs to be manually optimized by manipulating up to 13 thumbscrews located on the end mirrors, resonator mirrors, pump beam mirrors, and the focussing lens. Adding computerized actuators to these elements would help to automatically adjust for maximum output power and greatly reduce setup and optimization time. As a future prospect, the implementation of an optimization algorithm based on artificial intelligence research will ensure that the ionization efficiency is continuously at the optimal level.

9.5 Interchangeable Hardware and Mechanical Components

Future work on possible improvements of the overall reliability of the system should include the testing of different mirror mounts and stepper driver modules. For example, some drivers have a microstepping function which may improve the correction and scanning ability of the system.
Furthermore, improved laser beam coupling to the wavemeter using different fibers and modifyed optical fiber routing within the laser laboratory has indicated an increase of wavelength acquisition accuracy which has yet to be systematically studied.

9.6 RILIS Monitoring

The stabilization itself represents one small building block towards the larger task of a RILIS monitoring and a future remote control project. For future developments all system values can be monitored such as temperature, ethanol concentration, water leaks and laser diode currents. For the users, crucial operational values of interest could be exported to provide better assistance during the time constrained online runs, as well as some limited interaction for simple tasks, such as blocking and unblocking the lasers through the use of a remotely controllable shutter. These developments would ensure for the users **optimal ionization conditions** to maximize beam usage, **safe remote operation** for all operators, as well as **enhanced machine protection** to safeguard sensitive and crucial equipment.

Acknowledgments

In my experience, a project is successful only when all participants concert their efforts towards a common goal, keep channels of communication open and the exchange of ideas flowing.

I gratefully tip my hat to my congenial colleague Sebastian Rothe for our creative conversations, his dedication and getting-things-done attitude.

I am also very grateful for my supervisors Prof. Dr. Detlef Richter and Prof. Dr. Klaus Wendt and their advice. Their assistance and support paved the way to the successful completion of this project.

Working in the fields of Natural Science and Engineering on any scale requires an integral amount of team effort. I express my deepest gratitude towards the whole RILIS team, ISOLDE and CERN Control Center operators, Sven Richter and the whole LARISSA team at Universität Mainz for their open-minded support and great teamwork.

I thank the staff and fellow students of Hochschule RheinMain for the past three years and for the information provided that has helped with this project.

References

[1] *CERN ISOLDE website*: "The ISOLDE facility" http://isolde.web.cern.ch/isolde/default2.php?index=index/facilityindex.htm&main=facility/facility.php, date referenced: 10th of December 2011

[2] *Erich Kugler*: "The ISOLDE facility", Hyperfine Interactions, Geneva, 2000

[3] *CERN EN-STI-LP section website*: "Engineering : Sources, Targets and Interactions : Lasers and Photocathodes : Activities" https://en-sti-lp.web.cern.ch/en-sti-lp/, date referenced: 18th of December 2011

[4] *Peter A. Blume*: "The LabVIEW Style Book", Prentice Hall, 2007, ISBN: 9780131458352

[5] *T. Gockel, R. Dillmann*: "Embedded Robotics - Das Praxisbuch", Elektor-Verlag, Aachen, 2005, ISBN: 9783895761553

[6] *Sebastian Rothe*: "Aufbau eines Chrom: Forsterit-Lasers und Resonanzionisationsspektroskopie an Strontium, Titan, Nickel, Scandium und Silicium", Diploma Thesis, Mainz, 2009

[7] *B.E.A. Saleh, M.C. Teich*: "Fundamentals of Photonics", Wiley Series in Pure and Applied Optics, 1991, ISBN: 9780471839651

[8] *NI Developer Zone*: "Changing the Face of Design Patterns with LabVIEW 7 Express Event Structure" http://zone.ni.com/devzone/cda/tut/p/id/3085, date referenced: 5th of December 2011

[9] *National Instruments*: "National Instruments VISA" http://www.ni.com/visa/, date referenced: 5th of December 2011

[10] *CERN Engineering Department*: "LabVIEW Support Homepage" http://j2eeps.cern.ch/wikis/display/EN/LabVIEW+support, date referenced: 5th of December 2011

[11] *National Instruments*: "Product Information: What is NI LabVIEW?" http://www.ni.com/labview/whatis/, date referenced: 5th of December 2011

[12] *National Instruments*: "The Benefits of Programming Graphically in NI LabVIEW" http://www.ni.com/labview/whatis/graphical-programming/, date referenced: 5th of December 2011

[13] *CERN BE and ICE groups*: "FESA" http://wikis/display/fesa/Fesa+3, http://accelconf.web.cern.ch/accelconf/ica07/TALKS/WOPA04_TALK.PDF, date referenced: 7th of December 2011

[14] *CERN BE and ICE groups*: "The RADE project" http://en-dep-ice-mta.web.cern.ch/en-dep-ice-mta/Projects/Rade.html, date referenced: 7th of December 2011

[15] *National Instruments*: "Using the LabVIEW Run-Time Engine" http://zone.ni.com/reference/en-XX/help/371361B-01/lvhowto/using_the_lv_run_time_eng/, date referenced: 19th of December 2011

[16] *LabJack Corporation*: "Melexis MLX90614 IR Temperature Sensor I2C" http://labjack.com/support/app-notes/mlx90614-ir-temperature-sensor-i2c, date referenced: 19th of December 2011

[17] *NI Developer Zone*: "LabVIEW Interface for Arduino" https://decibel.ni.com/content/groups/labview-interface-for-arduino, date referenced: 20th of December 2011

[18] *Arduino homepage*: "Arduino overview" http://arduino.cc/, date referenced: 20th of December 2011

List of Figures

List of Tables

A Glossary of Terms

Ti:Sa Laser	Titanium Sapphire Laser
Etalon	Thin plate of transmissive material
RS-232	Recommended Standard 232, a computer communications interface
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
Wavenumber	The Wavenumber of a Laser is the reciprocal value of the Wavelength. It is usually denoted in cm^{-1} (inverse centimeters.)
Baud rate	Symbols per second i.e. line state changes per second. In binary transmissions, as used in the RS-232 interface, the bit rate corresponds to the baud rate.

B Abbreviation Index

CERN	**C**onseil **E**uropéen pour la **R**echerche **N**ucléaire
ISOLDE	**I**sotope **S**eparator **O**n **L**ine **DE**tector
RILIS	**R**esonance **I**onization **L**aser **I**on **S**ource
PSB	**P**roton **S**ynchrotron **B**ooster
Ti:Sa	**Ti**tanium:**Sa**pphire
ISST	**I**ntensity **S**patial **S**pectral **T**emporal
Nd:YAG	**N**eodymium-**d**oped **Y**ttrium **A**luminium **G**arnet
RS-232	**R**ecommended **S**tandard **232**
LabVIEW	**Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench
LARISSA	**LA**ser **R**esonanz **I**onisation für **S**pektroskopie in **S**elektiven **A**nwendungen
NI-DSM	**N**ational **I**nstruments **D**istributed **S**ystem **M**anager
IDE	**I**ntegrated **D**evelopment **E**nvironment
VI	**V**irtual **I**nstrument

C Translation Index

Regelkreis	-	Control Loop
Zweipunktregler	-	Two-point Controller
Dreipunktregler	-	Three-point Controller
Sollwert	-	Setpoint Value (Reference)
Istwert	-	Actual Value (System Output)
Messwert	-	Measured Value (Measured Output)
Abweichung	-	Difference Value
Abweichungseffekt	-	Correction Value
Störgröße	-	Error Value
Wellenzahl	-	Wavenumber