Reviewed by

B. Lendl C. Huck G. Friedbacher

Development and Characterization of a Near-Field Infrared Microscope by the Coupling of AFM and QCL Spectroscopy

A thesis submitted for the degree of
Doctor of Technical Sciences
at

Vienna University of Technology

Faculty of Technical Chemistry
Institute of Chemical Technologies and Analytics

under supervision of
Univ. Prof. Dr. Bernhard Lendl
and
Dr. Johannes Ofner

defended by
Georg Ramer
Mat.Nr. 0625513
Enzersdorferstrasse 2 / 1, A-2345 Brunn am Gebirge

Place and Date Georg Ramer

Abstract

Infrared spectroscopic imaging is a non-destructive, molecular specific technique for chemical analysis. It has found wide application in material science, bio-medicine and quality control. However, infrared imaging has one major limitation: its spatial resolution is Rayleigh limited to the low micrometer range. To overcome this limitation and improve the spatial resolution of infrared imaging scanning probe based near-field imaging techniques can be used. Resonance enhanced - photothermal induced resonance (RE-PTIR) is a promising near-field imaging technique for the mid-infrared spectral range. It combines a tunable pulsed infrared laser and an atomic force microscope (AFM) to achieve infrared imaging at a resolution better than 50 nm.

In this work a RE-PTIR setup was designed and assembled from a commercially available AFM and a pulsed, broadly tunable external cavity - quantum cascade laser (EC-QCL). The laser tuning behavior was characterized by performing Fourier transform infrared (FTIR) step scan measurements. To optimize the laser spot size two different methods of focusing the laser were implemented and tested. Using physical optics propagation simulation the spot size of the laser beam on the sample was optimized. As the need for tracking the contact resonance frequency to achieve stable measurements became apparent, a controller for tracking the resonance was designed and constructed. The controller generated trigger pulses for the EC-QCL and digitized the resulting cantilever deflection signal. To allow flexible, parallel digital signal processing, the calculations needed for resonance tracking were performed on a field programmable gate array (FPGA). After evaluation, the results were output to the AFM controller via digital analog converters (DACs) to enable the acquisition of location specific photothermal induced resonance (PTIR) signals. The FPGA programming to control RE-PTIR measurements was developed in tandem with a test bench that simulated the photo-expansion signal for a given laser pulse train. The finished controller used the modulus of the cantilever deflection signal to determine the amplitude of the cantilever and swept a range of frequencies to ensure that the maximum amplitude of a resonance mode was detected.

In its implementation at the end of this work, the RE-PTIR setup covered a spectral range of 1039.6 cm^{-1} by using a source that combined the output of

four EC-QCLs . The controller electronics allowed to track shifts of the contact resonance across jumps >10 kHz while generating updated infrared near-field measurements fast enough to allow contact mode imaging at usual speeds of ≈ 1 line per second. Update rates of 350 Hz of the PTIR were easily possible without distorting the shape of the resonance curve. Each update included a full sweep of the selected range of the resonance, detection of the maximum amplitude and output of the amplitude value. The system also allowed to acquire single point spectra across the whole range of the EC-QCL source at a spectral resolution sufficient for solid state spectroscopy ($\approx$1 cm^{-1}). Using this setup, spectra of polymer films down to a thickness of 60 nm were collected. These were in excellent compliance with far field infrared reference spectra. For a polymer film as thin as 8 nm detection of strong absorption bands was demonstrated to be still possible. A ten fold improvement in the signal to noise ratio was achieved in comparison to a lock-in detector by using the controller.

Time resolved infrared near-field measurements recording the change in secondary structure of a poly-L-lysine polypeptide film were demonstrated using the PTIR setup. In order to detect the changes in the secondary structure PTIR spectra across the peptide amide I band had to be acquired, as the change manifests as a band shift in the infrared spectrum.

The controller has been designed in an open and flexible way, using open electronics and freely available software. This will allow facile reconfiguration for future improvements and replication by others.

Zusammenfassung

Infrarotspektroskopie und -imaging sind zerstörungfreie, molekülspezifische Techniken zur chemischen Analyse. Sie haben, beispielsweise in der Materialwissenschaft, der Biologie, Medizin und in der Qualitätskontrolle breite Anwendung gefunden. Ein wichtiger limitierender Faktor für die bildgebende Infrarotspektroskopie liegt in ihrer beschränkten Ortsauflösung. Diese liegt bedingt durch das Rayleigh Kriterium im unteren Mikrometerbereich. Um die Ortsauflösung zu verbessern, können rastersondenbasierte Nahfeldmethoden verwendet werden. Eine vielversprechende Nahfeld-Methode ist die resonanzverstärkte photothermisch induzierte Resonanz (RE-PTIR). Diese Methode kombiniert einen gepulsten, durchstimmbaren Infrarotlaser mit einem Rasterkraftmikroskop (AFM), um eine Ortsauflösung besser als 50 nm im mittleren Infrarot zu erreichen.

Im Zuge dieser Arbeit wurde ein RE-PTIR Aufbau implementiert. Dazu wurde ein kommerziell erhältliches AFM mit einem gepulsten, breit durchstimmbaren Quantenkaskadenlaser mit externer Kavität (EC-QCL) kombiniert. Das Durchstimmverhalten des Lasers wurde mittels step-scan Fouriertransform Infrarotspektroskopie charakterisiert. Für ein möglichst hohes RE-PTIR Signal muss der Anregungslaser möglichst eng auf die Probe fokussiert werden. Hierfür wurden zwei verschiedene optische Aufbauten zum Fokussieren des Laserstrahls realisiert und getestet. Der Durchmesser des Brennpunkts wurde mittels wellenoptischer Simulation optimiert.

Anhand erster Versuche wurde deutlich, dass Änderungen der Kontaktresonanzfrequenz die Stabilität und Richtigkeit der Messungen negativ beeinflussten. Um diese Änderungen zu kompensieren, wurde eine Kontrollelektronik entwickelt. Diese Schaltung erzeugt Triggerpulse für den EC-QCL und digitalisiert die dadurch erzeugten Schwingungen des AFM Hebels. Um flexible und parallele elektronische Datenverarbeitung zu ermöglichen, wurden die zur Auswertung und Steuerung notwendigen Rechenschritte auf einem FPGA (engl.: field programmable gate array; dt.: im Feld programmierbares Logikgatter) durchgeführt. Die erstellte Schaltung gab nach der Auswertung das PTIR Amplitudensignal mittels eines analog-digital Wandlers aus, um die ortsspezifische Aufnahme des Signals mit einem AFM Controller zu ermöglichen. Gleichzeitig mit der Programmierung der Kontrollelektronik wurde auch ein digitaler Prüfstand

geschrieben, der das Photoexpansionssignal für beliebige Pulsfolgen simulierte. Die fertige Elektronik verwendete den Betrag der Auslenkung des AFM Hebels um die Amplitude der Schwingung des AFM Hebels zu bestimmen. Ein vom Benutzer definierter Frequenzbereich wurde wiederholt rasch abgetastet um die maximale Amplitude zu finden.

Am Ende dieser Arbeit konnte mit dem RE-PTIR Aufbau durch Verwendung einer Lichtquelle, die vier EC-QCLs kombiniert, ein spektraler Bereich von 1039.6 cm^{-1} abgedeckt werden. Die Schaltung konnte Sprünge der Resonanzfrequenz größer als 10 kHz ausgleichen und gleichzeitig ihren Ausgabewert oft genug aktualisieren, sodass Kontakt-AFM Messungen mit einer Zeilengeschwindigkeit von einer Zeile pro Sekunde möglich waren. Die Aktualisierung des Ausgabewerts konnte 350 mal pro Sekunde durchgeführt werden ohne die gemessene Form der Resonanzkurve zu verändern. Die entwickelte Elektronik erlaubte auch, Einzelpunktspektren über den gesamten Durchstimmbereich des EC-QCLs mit einer spektralen Auflösung von etwa 1 cm^{-1} aufzunehmen.

Mit dem entwickelten Nahfeld-Infrarot Aufbau wurden Spektren von Polymerfilmen bis zu einer Dicke von 60 nm aufgenommen, die in guter Übereinstimmung mit im Fernfeld gemessenen Referenzspektren waren. Auch Filme mit einer Dicke von 8 nm konnten noch anhand ihrer stärksten Absorptionsbanden identifiziert werden. Durch Verwendung der Kontrollelektronik konnte die relative Standardabweichung des Signals im Vergleich zu lock-in Verstärkerbasierten Messungen ohne Kompensation der Änderung der Resonanzfrequenz um einen Faktor 10 verbessert werden.

Außerdem konnte gezeigt werden, dass mit RE-PTIR auch zeitaufgelöste Messungen möglich sind. Die Änderung der Sekundärstruktur eines Poly-L-Lysin Polypeptid-Films wurde im Nahfeld spektroskopisch verfolgt. Da die Änderung der Sekundärstruktur eines Polypeptids sich im mittleren Infrarotbereich als Verschiebung der Amid I Bande manifestiert, reicht es zur Detektion der Änderung der Sekundärstruktur nicht aus die Änderung der Absorption an einer einzelnen Wellenlänge zu verfolgen. Stattdessen ist es notwendig, den gesamten Amid I Bereich aufzuzeichnen.

Durch Verwendung offener, gut dokumentierter Elektronik und frei verfügbarer Software konnte die Steuerelektronik in einer offenen und flexiblen Bauweise erstellt werden. Dies wird in Zukunft die einfache Rekonfigurierung und Verbesserung der Elektronik erlauben, sowie den Nachbau durch Dritte ermöglichen.

Acknowledgments

It is said that any journey begins with a single step. However, it is hard to say when the first step for this journey was. Was it when the grant proposals that funded this work were accepted by the FFG? Was it the evening when Bernhard plotted out possibilities for funding this work on the white board in the conference room and we tried to figure out a way to acquire the significant investment needed for this work? Was it, when after hearing Alexandre Dazzi present his near-field infrared technique, at ICAVS-6 in 2011 I decided that I wanted to work with this method. Or maybe it was even earlier, when I first decided to do my bachelor thesis in Bernhard's working group. Perhaps, but this is very unlikely, it might also have been when I first met Bernhard as a student. I did so miserably on an exam that he ended it by exclaiming:

> Aber den Unterschied zwischen Atom und Molekül kennen Sie schon?

Whatever the first step was, it is the journey that matters. On my personal journey I had many companions to whom I owe great gratitude.

Ferry Kienberger and Christian Rankl of Keysight Labs Austria were valued contributors to this work. Keysight Labs Austria provided the AFM used in this work, Ferry and Christian provided vital AFM expertise. My thanks go to Markus Brandstetter, Markus Wenin and Peter Burgholzer of the Research Center for Non-destructive Testing (Linz, Austria) for fruitful discussions on thermal expansion and acoustic waves.

Gernot Friedbacher (TU Wien) was generous enough to let me use his AFM for several whole weeks to perform first tests of RE-PTIR. These preliminary results then allowed us to acquire funding for the rest of the project. Dieter Baurecht (University of Vienna) gave invaluable advice for working with poly-L-lysine films and crucial tips for correcting artifacts in step-scan measurements.

Mikhail Belkin, Feng Lu, Mingzhou Jin (UT Austin), Andrea Centrone, Aaron Katzenmeyer and Basudev Lahiri (CNST) gave me the opportunity to perform preliminary measurements with their PTIR instruments.

My gratitude goes to all members of the Lendl group for providing help, support and distractions, as needed. Special thanks go to Christoph Gasser and Christian Kristament, who were instrumental to figuring out many of the electronic parts. Anna Balbekova was patient enough to wait until the setup finally worked (I had promised it to be done by mid-2014). Florian Reisenbauer worked as a master student on this setup. He redesigned the focusing optics and even spent his Easter weekend to collect the last missing data sets for this work. Benedikt Steindl developed the C library for communicating with the FT2232H chip during his internship in the summer of 2015. Anna Balbekova, Andreas Genner, Jakob Hayden, Harald Moser and Paul Waclawek combed this work for mistakes in content, spelling and layout.

Wolfgang Tomischko designed many of the electronic components used in this work - the pre-amplifier and the controller box were based on his schematics, for the controller box he drew the PCBs. Wolfgang also put in a lot of additional effort in the last few months of this dissertation when suddenly everything stopped working.

Karin was stuck with me twenty-four hours a day for the last years and, yet, has not run away. She was there for me to discuss my problems, was my rock when I was desperate because nothing seemed to work and helped me focus back on the important things when I started to chase rainbows. Karin is the one point my life revolves around and I hope it remains that way permanently.

I'm not able to express the gratitude I feel towards my family in words. My parents, Andrea and Wolfgang are the kindest, most patient people I know. They supported me financially and emotionally throughout my student life. I deeply thank my sister Christine and my grandparents, Hertha and Felix, for their love and understanding.

And finally my sincere thanks go to Bernhard Lendl who served as my mentor and thesis adviser. For a long time I have not understood what drives Bernhard in his work. Why does he stay long evenings to work on tedious things like writing proposals, managing funds to achieve the maximum output or planning conferences. All of this does not seem fun. And it certainly does not look like doing science.

However, I believe that over the last year I've gained a deeper understanding of why Bernhard is doing what he is doing. The fact is, that the stereotypical image of a scientist - a lone genius burning the midnight oil, driven by his daemons to leave their mark in the world of science - does not correspond to reality anymore. While a century ago, C.V. Raman and his coworker Ashutosh Dey were working as a two man team in their lab in Calcutta [1], progress nowadays is made by connecting the dots across borders, by knowing and bringing together people whose combined expertise is needed to solve problems. And this is what Bernhard excels at, spinning a web that ties together the people and ideas that need to be together. Making science possible and making it happen.

Bernhard, I am very grateful that I could write my thesis in your group. Thank you for furthering my development as a researcher by offering exceptional support.

To succeed, planning alone is insufficient. One must improvise as well.

Issac Asimov

Contents

Glossary

E Young's modulus. xv, 18, 19, 32, 37

E* reduced Young's modulus, $\frac{1}{E^\star} = \frac{1-\nu_t^2}{E_t} + \frac{1-\nu_s^2}{E_s}$, where t and s indicate tip and sample [2]. 22, 32, 88, *see* E

H Hamaker constant, a proportionality constant to describe Van der Waals interactions between two bodies. 22, 88

I area moment of inertia. 18, 19

k spring constant. 3, 19, 20, 23, 34, 35, 37, 116, 141

λ wavelength. 53

ν Poisson's ratio, the ratio of strain in the axial and transversal direction. $\nu \approx -\frac{\Delta l_y}{\Delta l_x}$. xv, 32, 37

p parallel to the plane of incidence. 54

Q quality factor. 19, 33, 34, 36, 43, 88, 96

ρ density. 18, 19

s perpedendicular to the plane of incidence. 53, 54

Acronyms

ADC analog digital converter. 67, 77, 78, 80, 82, 91, 92, 99, 123

AFM atomic force microscope. iii, 15–18, 20, 21, 23, 26, 31, 33, 34, 38, 42, 54, 56, 57, 61, 64, 65, 69, 74, 76, 92, 99, 100, 103, 106, 122, 124, 125, 141

ATR attenuated total reflection. 12, 28–30, 32

CLI command line interface. 95

CW continous wave. 6, 27

DAC digital analog converter. iii, 42, 67, 77, 78, 84, 85, 92

DART dual AC resonance tracking. 41, 72–74, 142

DC direct current. 92, 98

DDS direct digital synthesis. 42

DIP dual inline package. 67

DLaTGS deuterated lanthanum α alanine doped triglycine sulphate. 6

DLL dynamic linked library. 48, 78, 87

DMT model Derjaguin-Müller-Toporov model. 22

DSP digital signal processing. 42

EC-QCL external cavity - quantum cascade laser. iii, iv, 6, 7, 27, 28, 47–49, 51, 54, 62, 63, 65, 70–72, 74–76, 88, 99, 100, 103, 106, 119, 122, 125, 143

FEL free electron laser. 26, 27

FEM finite element method. 34

FFT fast Fourier transform. 5, 39, 123

FIFO first in - first out. 85–87

FIR finite impulse response. 84, 121

FM-AFM frequency modulation atomic force microscopy. 74–76, 143

FPGA field programmable gate array. iii, xiii, 77–79, 82, 85, 86, 90, 91, 95, 121–123, 173, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238

FTIR Fourier transform infrared. iii, 4–7, 25, 49, 51, 105, 143

FWHM full width at half maximum. 6, 19, 26, 36, 62

GUI graphical user interface. 48, 67

HDL hardware description language. 78

IC integrated circuit. 42, 79, 82, 84–87, 91

IR infrared. 3–8, 12, 14, 16, 25–28, 30, 34, 54, 63, 69

IRRAS infrared reflection absorption spectroscopy. 54

ITO indium tin oxide. 63

MCT mercury cadmium telluride. 6, 26

OPG optical parametrical generator. 25

OPO optical parametric oscillator. 27, 28, 33

PAA poly-acrylic acid. 74, 75, 143

PALM photoactivatable localization microscopy. 14

PC personal computer. 48, 77, 78, 80, 85, 122

PCB printed circuit board. 78, 123

PEDOT poly(3,4-ethylenedioxythiophene). 29

PI proportional-integral controller. 21, 122

PID proportional-integral-derivative controller. 21

PLL phase locked loop. 42, 73, 74

PMMA poly(methyl methacrylate). 30

PSD position sensitive detector. 16, 19, 21, 22

PTIR photothermal induced resonance. iii, iv, xiii, 16, 18, 23, 27–30, 32–34, 36, 41, 48, 53, 54, 56, 67, 70, 74, 75, 99, 103–109, 111, 112, 119–122, 124, 125, 143, 144, 173, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238

PVD physical vapor deposition. 106, 108, 109, 114, 144

PWM pulse width modulation. 70–73, 143

QCL quantum cascade laser. 6, 28, 48, 49, 51, 52, 103, 125, 126, 142

RE-PTIR resonance enhanced - photothermal induced resonance. iii, 47, 70, 74, 76

RFL reflected focal length. 57

RMS root mean square. 42, 91

RTL register transfer level. 79

SNOM scanning near-field optical microscopy. xix, 25, 26

SNR signal to noise ratio. 5, 7, 19, 20, 25, 26, 36, 67, 68, 73, 103, 111, 113, 122, 143, 144

SPI serial peripheral interface. 77, 84, 85

s-SNOM scattering-SNOM. 26, 27

STED stimulated emission depletion. 14

SThM scanning thermal microscopy. 25

STM scanning tunneling microscope. 15

STORM stochastic optical reconstruction microscopy. 14

TTL transistor type logic. 49, 51, 76, 78

USB universal serial bus. 77, 91, 98

VCO voltage controlled oscillator. 42

VHDL very high-speed integrated circuit hardware description language. 78, 79, 89, 92

Part I.

Theoretical Background

1. Infrared Spectroscopy and Imaging

1.1. Infrared Spectroscopy - a Very Brief Introduction

For a thorough introduction the reader is advised to consult *Fourier Transform Infrared Spectrometry* by Griffiths and De Haseth [3], below only a succinct overview will be given.

The mid-infrared (IR) range of the electromagnetic spectrum is usually given as the range between 400 cm^{-1} to 4000 cm^{-1} (2.5 µm to 25 µm). Photons in this range do not have sufficient energy to excite electronic transitions, instead they exicte molecular vibrations. The location of an IR absorption band in the spectrum is defined by the structure of the molecule, its atoms and the way they are bound.

To get a feeling for the influence of the properties of a molecule on the frequencies of the vibration bands we imagine a two atomic molecule as two point masses connected by a spring. The masses are those of the bound atoms and the force constant k of the spring is set to the "bond strength", i.e. force needed to move the atoms slightly from their equilibrium position. The fundamental frequency of this oscillator can be calculated as

$$f = \frac{1}{2\pi}\frac{k}{m_r} \tag{1.1}$$

with m_r being the reduced mass, calculated from the masses of the balls m_1 and m_2 as $m_r = \frac{m_1 m_2}{m_1+m_2}$. Due to quantization of the vibrational energy states can be found at energies V_{iv} of

$$V_{iv} = hf(v_i + \frac{1}{2}) \tag{1.2}$$

Here h is Planck's constant and v_i is the vibrational quantum number ($v_i = 0, 1, \ldots$). Transitions between vibrational states are allowed at $v_i = \pm 1$, leading to energy differences of

$$\Delta V = hf \tag{1.3}$$

and the wavenumber of the band being

$$\tilde{\nu} = \frac{f}{2\pi c_0} \tag{1.4}$$

where c_0 is the vacuum speed of light. We can now make two observations about the influence of the properties of our molecule on the location of the vibrational bands:

1. When the reduced mass of the atoms increases, the wavenumber of the band decreases.

2. When the force between the atoms is increased, e.g. from single bond to double bond, the wavenumber increases as well.

This, of course, is not the full picture on mid-IR spectroscopy. Further important facts include that only vibrations during which the dipole moment μ changes lead to absorption bands. Infrared spectroscopy is polarization sensitive, i.e. the band intensity is proportional to the product of the electric field vector and the direction of the change of the dipole moment.

In addition to qualitative information, IR spectroscopy also allows quantitation of molecules. In transmission measurements, where the sample is placed inbetween an IR light source and an IR detector, the attenuation of the light through the sample is given by Beer's law:

$$A(\tilde{\nu}) = -\log_{10}\left(\frac{I(\tilde{\nu})}{I_0(\tilde{\nu})}\right) = \varepsilon(\tilde{\nu})cl \tag{1.5}$$

I_0 and I are the light intensity before and after the sample, c is the concentration, l is the sample thickness and ε is the molar extinction coefficient. While the extinction coefficients of some substances can be found in literature, the usual approach is to determine the factor εl experimentally for a given instrument by recording a calibration curve around the expected concentration.

1.1.1. Instrumentation

Currently the predominant instrument for mid-IR spectroscopy is still the Fourier transform infrared (FTIR) spectrometer. In such an instrument wavelength resolution is acquired indirectly via an interferogram of the light with itself. The light emitted by the light source is split into two parts at a beam splitter, one half the light is directed towards a fixed mirror positioned at a distance l the second one is directed towards a moving mirror positioned at $l + \Delta l/2$. Both mirrors reflect the light back towards the beam splitter (see fig. 1.1 for a sketch of the beam paths). Constructive and destructive interference of light in the output beam are defined by the ratio of the wavelength and the

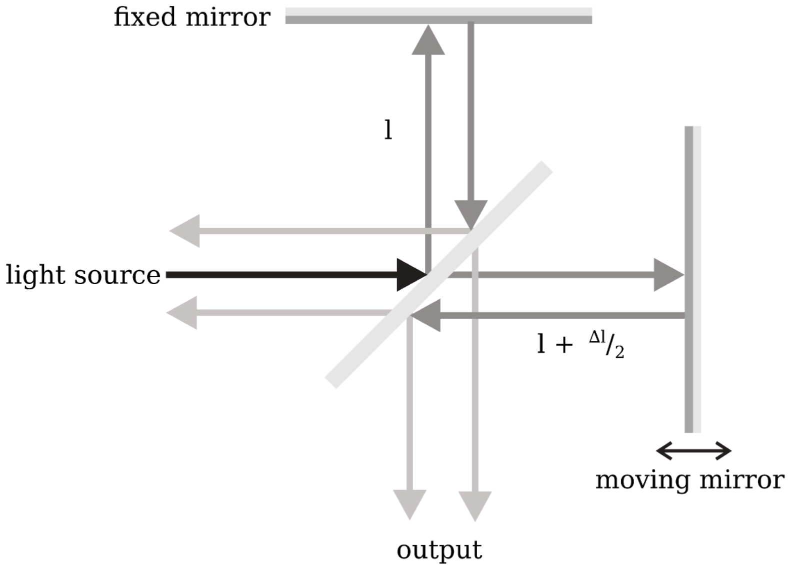


Figure 1.1.: Beam paths in one type of interferometer.

path length difference. As it turns out the intensity in the output arm of the interferometer is the cosine transform of the input light spectrum

$$S(\Delta l) = \int_{-\infty}^{\infty} I(\tilde{\nu}) \cos(2\pi\tilde{\nu}\Delta l) d\tilde{\nu} \tag{1.6}$$

Therefore, when the output intensity is recorded together with the path difference the original spectrum can be recovered by applying the inverse cosine transform. For real input the cosine transform is the real part of a Fourier transform, hence the fast, optimized algorithms for fast Fourier transform (FFT) can be used. Using the Fourier transform scheme for IR spectroscopy comes with three main advantages:

1. The multiplex (or Fellgett) advantage states that the signal to noise ratio (SNR) of a Fourier transform instrument recording M data points will be $\sqrt{M}$ higher than that of a dispersive instrument with the same parameters.

2. The throughput (or Jacquinot) advantage lies in the circular beam shape accepted by Fourier transform instrument. As the link between throughput and resolution is not as stringent as for dispersive instruments an FTIR instrument can operate at a higher SNR while achieving the same spectral resolution.

3. In the reference laser used to determine the mirror position, FTIR instruments have an inbuilt wavelength reference (Connes advantage).

In recent years, newly available broadly tunable laser light sources have somewhat negated these advantages of Fourier transform. These external cavity - quantum cascade lasers (EC-QCLs) provide high intensity (up to Watt range), collimated mid-IR light and can be tuned across several hundreds of wavenumbers [4]. EC-QCLs are electrically pumped semiconductor lasers that consist of a broad band gain medium, which is placed in an external cavity with a monochromator.

The broad band gain medium consists of a layer structure of different composition specifically designed to split its conduction band into several sub-bands [5]. Lasing in quantum cascade lasers (QCLs) happens via transitions between two sub-bands. By adjusting the energy differences between the sub-bands the wavelength range that can be amplified by the medium can be selected. If a current is applied to the laser medium electrons will collect in the higher energy sub-bands creating the population inversion needed for lasing. However, without an optical resonator only spontaneous emission across the range supported by the gain medium will happen. The optical resonator in an EC-QCL includes a diffraction grating that reflects one specific wavelength back into the gain medium. There the light is amplified and then emitted. Selection of the wavelength of the EC-QCL is performed by rotating the diffraction grating. In short, the gain medium defines the wavelengths the laser can emit and the grating is used to select one specific wavelength that is to be emitted.

EC-QCLs can be classified into pulsed and continous wave (CW) lasers. In general pulsed lasers have a broader full width at half maximum (FWHM) for each pulse than CW lasers due to thermal chirping of the laser. As to characteristic performances of such lasers: for the EC-QCL used for this work (see section 4.1.1 for a description) the vendor specifies a FWHM in pulsed mode of 1 cm^{-1} while for CW mode 0.001 cm^{-1} is specified.

Detectors used in IR spectrometers are either photonic detectors or thermal detectors. The read out signal of photonic detectors depends on the number of photons within a certain wavenumber range that hit the detector. Due to the low energy of IR photons, such detectors need cooling to liquid nitrogen temperatures to reduced thermal noise. Commonly used photonic detectors for the mid-IR are mercury cadmium telluride (MCT) detectors.

Thermal detectors give a read out proportional to the temperature or to the temperature change of the detector element. Deuterated lanthanum α alanine doped triglycine sulphate (DLaTGS) detectors are a type of pyroelectric detectors that are found in many FTIR instruments.

1.2. Infrared Microscopy

When the sample chamber in an IR spectrometer is replaced by imaging optics, spatially resolved IR spectra can be collected. By doing so, additional information about the spatial distribution of analytes in the sample is generated.

In recent years significant effort has been invested into IR microscopy, as this method promises to perform non-destructive, label free analysis of biological samples, such as microbes, tissue sections and liquid droplets [6]. Furthermore, microscopy has been used to analyze filamentous fungi where spatially resolved chemical information allowed to detect differences in the local composition of the hyphae [7] and to locate tissue types in sections of human tissue [8]. The assignment of tissue types in IR images is performed via algorithms applied to the raw spectra. Hence, by only changing the method of evaluating the data, plant sections can be "stained" as well [9].

However, biology and medicine are not the sole target for investigation using IR microscopy. In forensics, IR microscopy can be used to study residues left in fingerprints [10, 11]. In the analysis of artworks, the spatially resolved chemical information provided by IR spectroscopic images is used to study cross sections of paintings [12, 13]. Additionally, IR imaging has been used to analyze the composition of pharmaceutical formulations [14].

There are two types of detectors used for FTIR microscopy:

- single point detectors, as in IR spectrometers,
- focal plane array detectors, that record a spectroscopic image, i.e. multiple spectra corresponding to different parts of the sample at once.

Focal plane array detectors have the advantage of reducing measurement time relative to the time needed to acquire an image of the same resolution and SNR with a single point detector. The main drawback of focal plane array detectors is their significantly higher cost.

Light sources are predominantly thermal light sources as in conventional spectrometers with specialized applications using synchrotron sources. However, in recent years a commercial spectrometer employing an EC-QCL light source has been introduced. The advantage of the EC-QCL is again its higher intensity, which allows to use a room temperature focal plane array detector with a higher number of pixels [15]. When the necessary information in a sample is concentrated at a few wavenumbers in its spectrum, EC-QCL can speed up

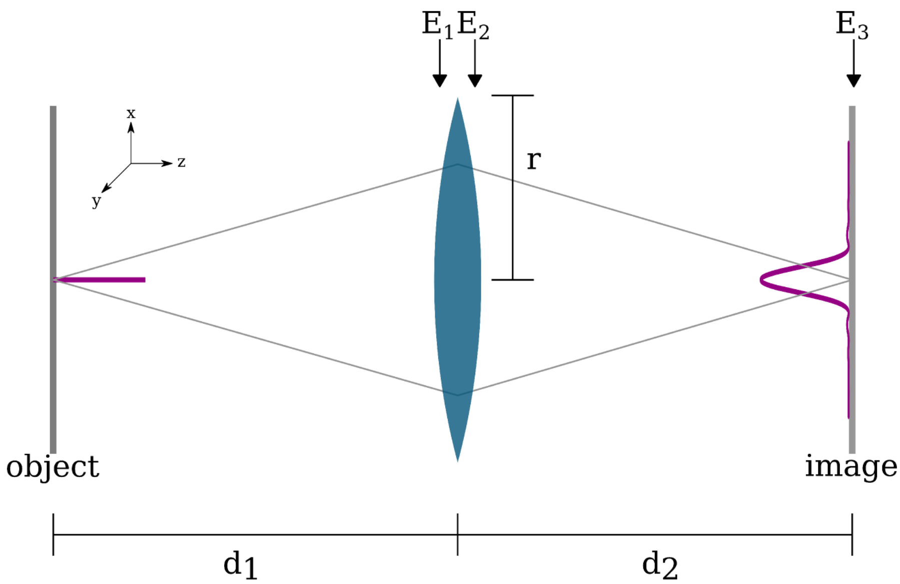


Figure 1.2.: Sketch used for deriving the Rayleigh limit.

data collection significantly as only the needed wavenumbers can be measured instead of a full interferogram [15].

There is however one important limit inherent to IR microscopic imaging: the best achievable spatial resolution in far field imaging in this wavelength range lies in the range of several micrometers.

1.3. Spatial Resolution

To develop an expression for the limit to spatial resolution in far-field optical techniques we are going to derive the impulse response function of a one lens imaging system. The impulse response function describes the output the system gives for an infinitely narrow input signal. The explanation roughly follows the explanation given in *Fundamentals of Photonics* [16].

Our system consists of a single point light source in the object plane described as a delta function ($\delta(x, y)$) which is placed at a distance d_1 along z from a thin lens with focal length f and aperture radius r. After passing the lens, the light coming from the source is focused onto the image plane placed at a distance d_2 from the source (see fig. 1.2). Object plane, image plane and lens are places parallel to the $x - y$ plane.

When using the Fresnel approximation the amplitude of the elecromagnetic field on a plane at a distance d_1 from the source can be written as

$$E_1(x,y) \propto \exp(-ik\frac{x^2+y^2}{2d_1}) \tag{1.7}$$

where $k = {}^{2\pi}/_{\lambda}$ is the length of the wavevector $\vec{k}$. The Fresnel approximation assumes that the wavevector components in x and y direction are small in comparison to k. Equation (1.7) is the impulse response of free space propagation. The impulse response function can be used to calculate the effect an optical system has on an input field. To determine the output of the system for a given input the convolution of the impulse response function is calculated.

After reaching the lens, only the part of light that is inside the lens radius r can pass through. This is described by multiplying E_1 with the circ($x/r, y/r$) function.

$$\mathrm{circ}(x,y) = \begin{cases} 1 & \sqrt{x^2+y^2} \leq 1 \\ 0 & elsewhere \end{cases} \tag{1.8}$$

The thin lens only influences the phase of the electromagnetic signal. It can be fully described by using the focal length f of the lens:

$$L(x,y) \propto \exp\left(ik\frac{x^2+y^2}{2f}\right) \tag{1.9}$$

The electromagnetic field after the lens thus reads

$$\begin{aligned} E_2(x,y) &= E_1(x,y)\,\mathrm{circ}(x/r, y/r)\,L(x,y) \\ &= \mathrm{circ}(x/r, y/r)\,\exp\left[ik\left(\frac{x^2+y^2}{2f} - \frac{x^2+y^2}{2d_1}\right)\right] \\ &= \mathrm{circ}(x/r, y/r)\exp\left[ik\frac{x^2+y^2}{2}\left(\frac{1}{f} - \frac{1}{d_1}\right)\right] \end{aligned} \tag{1.10}$$

To calculate the electromagnetic field E_3 in the image plane the convolution of E_2 and the impulse response of free space propagation (1.7) has to be calculated.

$$E_3(x,y) \propto \iint_{-\infty}^{\infty} E_2(x',y')\exp\left[-ik\frac{(x-x')^2+(y-y')^2}{2d_2}\right]dx'dy' \tag{1.11}$$

$$\begin{aligned} &\propto \exp\left(-ik\frac{x^2+y^2}{2d_2}\right)\iint_{-\infty}^{\infty} \mathrm{circ}(x'/r, y'/r) \\ &\quad \exp\left[ik\frac{x'^2+y'^2}{2}\left(\frac{1}{f} - \frac{1}{d_1} - \frac{1}{d_2}\right)\right]\exp\left(ik\frac{xx'+yy'}{d_2}\right)dx'dy' \end{aligned} \tag{1.12}$$

When the image and the object plane are positioned so that

$$\frac{1}{f} = \frac{1}{d_1} + \frac{1}{d_2} \tag{1.13}$$

(1.12) is simplified to

$$E_3(x, y) \propto \exp\left(-ik\frac{x^2 + y^2}{2d_2}\right) \iint_{-\infty}^{\infty} \mathrm{circ}(x'/r, y'/r) \exp\left(ik\frac{xx' + yy'}{d_2}\right) dx'dy' \tag{1.14}$$

Using the substitutions $x' = \chi\lambda d_2$ and $y' = \xi\lambda d_2$ (1.14) is converted to

$$E_3(x, y) \propto \exp\left(-ik\frac{x^2 + y^2}{2d_2}\right) \iint_{-\infty}^{\infty} \mathrm{circ}(\chi\lambda d_2/r, \xi\lambda d_2/r) \exp\left(i2\pi\left(x\chi + y\xi\right)\right) d\chi d\xi \tag{1.15}$$

The integral in (1.15) is now equivalent to the Fourier transform of $\mathrm{circ}(\chi\lambda d_2/r, \xi\lambda d_2/r)$ which can be found tabulated as

$$\mathcal{F}\left(\mathrm{circ}(x', y')\right) = \frac{J_1\left(2\pi\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}} \tag{1.16}$$

where J_1 is the first order Bessel function of the first kind. For the coefficients of x and y in this case the resulting function has the form of

$$E_3(x, y) \propto \exp\left(-ik\frac{x^2 + y^2}{2d_2}\right) \frac{J_1\left(\frac{2\pi r}{\lambda d_2}\sqrt{x^2 + y^2}\right)}{\frac{r}{\lambda d_2}\sqrt{x^2 + y^2}} \tag{1.17}$$

For calculating the intensity in the image plane the exponential function in (1.17) can be ignored, as its absolute value is 1. The shape of the remaining position dependent part of (1.17) is depicted in figure 1.3.

The intensity distribution in the image plane is calculated as $I = |E_3|^2$. The first minimum of the intensity distribution is equal to the position of the first root of J_1, which is found either in tables or provided by numerical mathematics software (e.g from the function scipy.special.jn_zeros) as $3.83171\ldots$. The radius $\rho = \sqrt{x^2 + y^2}$ at which the intensity reaches its minimum is

$$\rho = \frac{1.22\lambda d_2}{2r} \tag{1.18}$$

This radius is called the Rayleigh criterion after John William Strutt, the third lord of Rayleigh, who declared that two points spread at least $\Delta\rho = \frac{1.22\lambda d_2}{2\pi r}$ apart

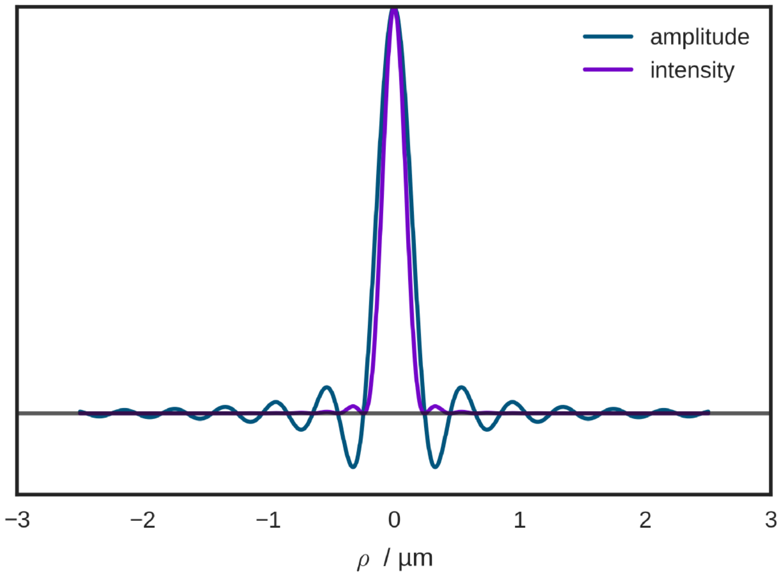


Figure 1.3.: Amplitude and intensity of the electromagnetic field in the image plane for $d_2 = 1\,\text{mm}$, $r = 5\,\text{mm}$ and $\lambda = 1\,\mu\text{m}$.

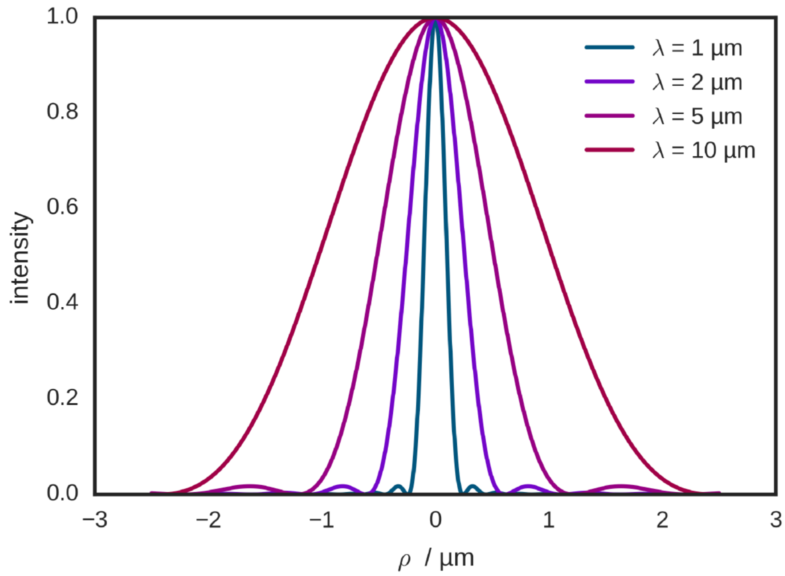


Figure 1.4.: Shape of the intensity distribution in the image plane for different wavelengths.

in the image plane as resolved [17]. To calculate the minimum distance of that two these points in the object plane need to have to allow resolution in the imaging plane $\Delta\rho$ can be projected from the image plane back to the object plane. This is done by dividing $\Delta\rho$ by the magnification factor of the lens $M = -\frac{d_2 n}{d_1}$. Here n is the refractive index of the medium between lens and object. The distance of two points in the object plane that can still be resolved according to the Rayleigh criterion is then made out to be

$$\Delta\rho = \frac{1.22\lambda d_1}{2nr} \tag{1.19}$$

For microcopes the fraction r/d_1 is often replaced by the sine of the acceptance angle of the objective α

$$\Delta\rho = \frac{1.22\lambda}{2n\sin(\alpha)} \tag{1.20}$$

It is important to note that the limit to spatial resolution is introduced by the aperture (in this case the lens diameter) and not by any other part of the setup.

Looking at the Rayleigh criterion there are three important factors that determine the spatial resolution in far-field optical imaging: the wavelength, the refractive index and the opening angle of the objective. To increase the spatial resolution either the wavelength has to be decreased (see fig. 1.5), the acceptance angle has to be increased or the refractive index of the surrounding medium has to be increased.

For IR spectroscopy, where the wanted information is tied to the wavelength, changing to a shorter wavelength is not an option. The acceptance angle of the objective can be increased by choosing a wider lens and a closer working distance, $\sin(\alpha)$ will, however, at most be 1.

Increasing the spatial resolution by increasing the refractive index around the sample is commonly performed in IR spectroscopy in the form of attenuated total reflection (ATR)-microscopy. For this technique a high refractice index material, usually Ge ($n \approx 4$) is pressed against the surface of the sample. The interface between sample and ATR element is then imaged using an IR microscope. For a short overview on state-of-the-art ATR microscopy see "Attenuated Total Reflection Fourier Transform Infrared Spectroscopy" [18]. As Ge is the IR transparent material with the highest refractive index available, only a fourfold improvement of the spatial resolution is possible through ATR. Hence, the best achievable resolution with an optical microscope in the mid-IR range is in the order of magnitude of the used wavelength, 1 µm to 10 µm. The spatial

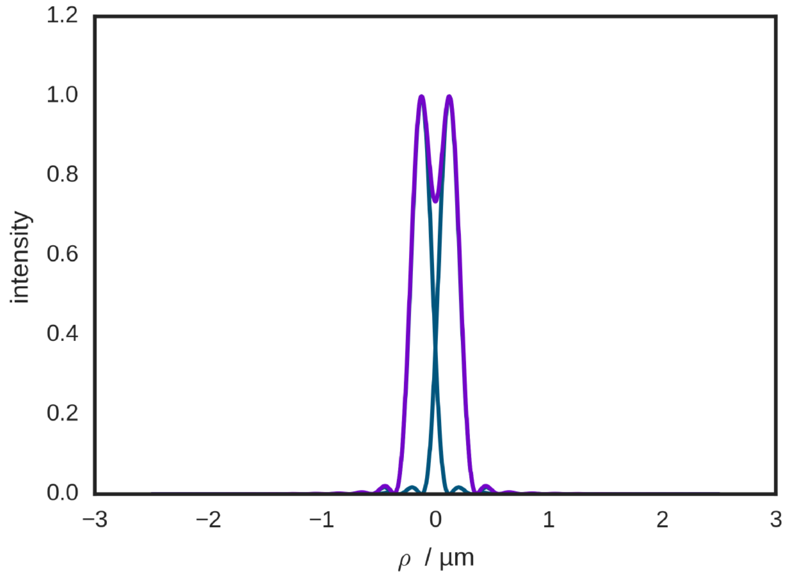


Figure 1.5.: Images of two points positioned at a distance sufficient for resolution according to the Rayleigh criterion. The two point spread functions are drawn in blue, the sum of the point spread functions, i.e. the signal that is actually detected, is in drawn in purple.

resolution in mid-IR is therefore not sufficient for many samples of interest, such as microorganisms (diameter $\approx 1\,\mu m$).

It should be noted, that the Rayleigh criterion is only one of several definitions of the spatial resolution in optical systems. Others are the Abbé limit and Sparrow's limit. The difference between these limits is, how far the intensity between two points has to dip before they are declared as resolved. For the Rayleigh criterion this is 26.4 % of the maximum value whereas for Sparrow's limit it is the smallest distance where a saddle point between two points is observed [19].

In an experimental investigation of the spatial resolution in commercial IR microscopes Lasch and Naumann [19] found a spatial resolution (Rayleigh criterion) of 3.9 µm at 2300 cm^{-1} and 2.4 µm at 4000 cm^{-1} for a Bruker Hyperion 3000 using a 36x beam condensor (the second value surpassing the Rayleigh criterion could be explained by the confocal configuration of the instrument, see below).

There are several ways to increase the spatial resolution in far-field optical techniques (see "Far-Field Optical Nanoscopy" [20] for an overview). Some of these techniques work by reducing the size of the point spread function. This is done in confocal microscopy, 4PI microscopy and stimulated emission depletion (STED). Other techniques, like photoactivatable localization microscopy (PALM) and stochastic optical reconstruction microscopy (STORM), ensure that the emitters in the sample are far enough apart that their point spread functions do not overlap. Through deconvolution the light source can than be localized at high spatial resolution. However, 4PI, STED, PALM and STORM can not be applied to IR microscopy as they need fluorescent dyes that are not available in the mid-IR wavelength range. Confocal microscopy is commonly used for the mid-IR range. Sadly, this technique only reduces the point spread function by $\sqrt{2}$ [20].

Another possibility to increase the spatial resolution is through near field detection. Near-field detection techniques work by either moving the detector and/or the light source in the immediate vicinity ($\approx \lambda$) of the sample. In this case limits to the spatial resolution due to far field effects do not apply and images at spatial resolutions surpassing the diffraction limit are possible [21]. In this work such a near field detection technique that is specifically adapted for the use in the mid-IR range was investigated. For further information about near-field techniques see section 3.1.

2. Atomic Force Microscopy

2.1. Atomic Force Microscopy

2.1.1. Historical Development

Atomic force microscopy is a technique for high resolution topographical imaging down to atomic resolution. The technique was introduced in 1986 by Binnig, Quate, and Gerber [22] as a follow-up to their scanning tunneling microscopy technique introduced in 1982 [23].

In a scanning tunneling microscope (STM) measurement a sharp metal tip is moved across the surface of a conductive sample using piezodrives for high resolution positioning. When the tunnel voltage between sample and surface is held constant the tunnel current J_T is $J_T \propto \exp(-\psi^{1/2}s)$, where s is the distance between tip and sample and ψ is the work function or the barrier height between tip and sample at the tip position [23]. Since the decay length of current is in the sub-Å range, STM can generate vertical resolutions in the range of 0.1 Å or better [24]. Due to the high vertical resolution the lateral resolution in an STM is increased as well as only the bottom most atom of the metal tip contributes to the tunneling current [23]. The main drawback of scanning tunneling microscopy is that it cannot be used for non-conductive samples.

In contrast, the atomic force microscope (AFM) is not restricted by the sample conductivity. Instead of the tunnel current between tip and sample the force exerted by the sample onto the tip is used for probing the surface. Binnig, Quate, and Gerber [22] used a gold foil - with diamond tip to interact with the sample - as force sensor. When the spring constant of this Au foil is known the tip sample force can be determined from the deflection of the foil. In the original implementation of AFM the deflection of the foil was detected using an STM on the backside of the gold foil.

2.1.2. Instrumentation

In modern AFM cantilever deflection detection via STM has now been mostly replaced by other methods. The beam-deflection method, which currently is the most widely used scheme in AFM, was introduced in 1988 by Meyer and Amer [25]. Beam deflection works by reflecting a weak laser beam off the backside of the AFM cantilever and onto the center of a position sensitive detector (PSD). When the slope of the cantilever at the point of incidence of the laser changes the laser beam is reflected in a slightly different direction. The deflection is detected through the difference in voltages of the upper and lower part of the PSD. Cantilever fabrication has also switched from bent wires or metal foils of earlier days to dedicated silicon and silicon nitride devices produced by using micro-fabrication techniques.

Atomic force microscopy has branched out into a wide variety of different operational modes all based on force detection via an elastic cantilever. However, for this work, the focus remains on contact mode AFM, one of the two techniques described in the original paper by Binnig, Quate, and Gerber. In contact mode AFM the force between sample and tip is kept repulsive and the vertical cantilever position is at most adjusted to keep a constant deflection, and thereby a constant force between tip and sample. The second common operational mode is tapping mode - where the tip is vibrated in the vicinity of the sample and only interacts with the sample in the bottom-most part of its oscillation. In tapping mode, the z position of the cantilever is adjusted to keep the amplitude constant. Tapping mode is generally less damaging for the sample and generates correcter images of the topography than contact mode, as contact mode not only exerts vertical forces on the sample but also horizontal shear forces (see for example [26]). As no implementation of photothermal near-field IR using tapping mode is currently known, this method will not be described in detail.

In AFM the interaction between sample and tip is ideally restricted to the bottom most atom of the tip. However, at ambient conditions long range forces such as van der Waals dispersion forces, dipole-dipole interactions, polarization forces, Coulomb forces due to charges of sample and cantilever and capillary forces due to thin liquid films can be encountered [27]. If any of these forces is attractive the apparent tip-sample force is reduced which can lead to sample damage [27]. Part of the reduction of the tip sample force can be seen in the force curve of the AFM approach and retraction (see figure 2.1 of a typical example).

Contact mode can be performed in constant height or in constant force mode. In constant height mode the z positioning actor is kept at a constant height and only the cantilever deflection signal is used to detect the sample height while in constant force mode the cantilever deflection is kept at a preset value by adjusting the z actor. While constant height mode allows faster imaging it can only be used for very flat samples as any steps in the sample height would increase the force on the sample and thus may damage it [26]. To the best of my knowledge, photothermal induced resonance (PTIR) appears to be used with

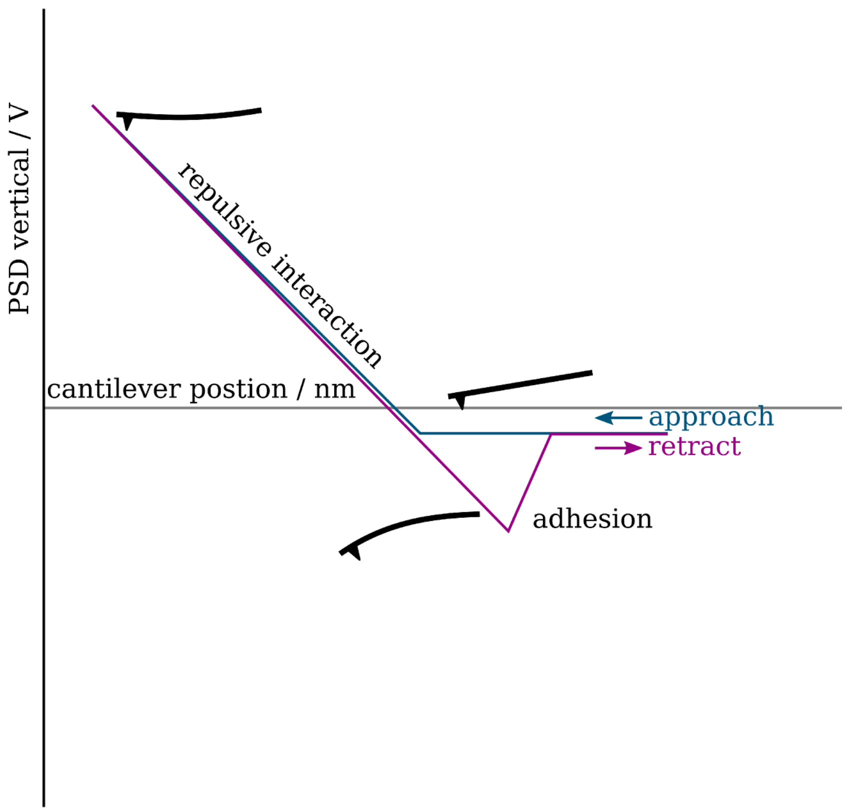


Figure 2.1.: Force curve for constact mode AFM.

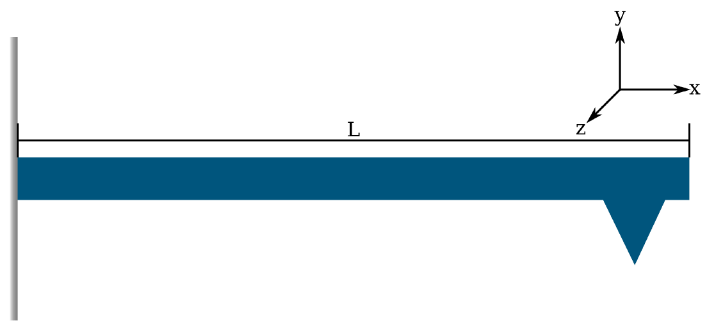


Figure 2.2.: Sketch of the beam layout described in this section.

constant force mode exclusively. While there is no fundamental reason that would preclude using the method with constant height mode the shape of the samples (i.e. with height difference of several hundreds of nanometers) together with the comparatively slow imaging speed of PTIR make constant force mode the better fit.

2.2. AFM Cantilevers

The main parameters of AFM cantilevers are their spring constant, their length, their material and their tip shape.

The behavior of a (beam-shaped) AFM cantilever can be modeled using the Euler-Bernoulli beam [28] (see fig. 2.2 for a sketch of the setup).

$$\mathrm{EI}\frac{\partial^4 y}{\partial x^4} + \rho A \frac{\partial^2 y}{\partial t^2} = 0 \tag{2.1}$$

Here the cantilever extends from $x = 0$ to L, y is the vertical position of the cantilever. A is the cross section area of the cantilever, ρ is the density, E is Young's modulus and I is the area moment of inertia. The general solution for this equation has the shape [2]

$$y(x,t) = (a_1 e^{kx} + a_2 e^{-kx} + a_3 e^{ikx} + a_4 e^{-ikx})e^{-i\omega t} \tag{2.2}$$

with $k = 2\pi/\lambda$ and $\omega = 2\pi f$. When the cantilever is clamped at the left end ($y = 0$ and $\frac{\partial y}{\partial x} = 0$ at $x = 0$) and free on the right end, meaning its moment is zero $\frac{\partial^2 y}{\partial x^2} = 0$ and it experiences no shear force $\frac{\partial^3 y}{\partial x^3} = 0$ at $x = L$ then only solutions fulfilling

$$\cos(k_n L)\cosh(k_n L) + 1 = 0 \tag{2.3}$$

are possible [2]. Through

$$\mathrm{EI}k^4 - \rho A \omega^2 = 0 \tag{2.4}$$

which is gained by inserting (2.2) into (2.1), k_n can be converted to resonant frequencies f_n [2]

$$f_n = \frac{(k_n L)^2}{L^2 2\pi \sqrt[2]{\frac{\rho A}{\mathrm{EI}}}} \tag{2.5}$$

and the spring constant equivalent to a point mass on a spring model

$$k_C = \frac{\mathrm{E} b^3 a}{4L^3} \tag{2.6}$$

The first free resonance of a cantilever is [2]

$$\omega_1 = k^2 \sqrt{\frac{\mathrm{EI}}{\rho A}} \tag{2.7}$$

with the area of the rectangular cross section of the beam being $A = ab$ and its area moment of inertia $\mathrm{I} = \frac{ab^3}{10}$.

Damping due to interaction with the surrounding medium (air, water,..) can be modeled as an additional term of

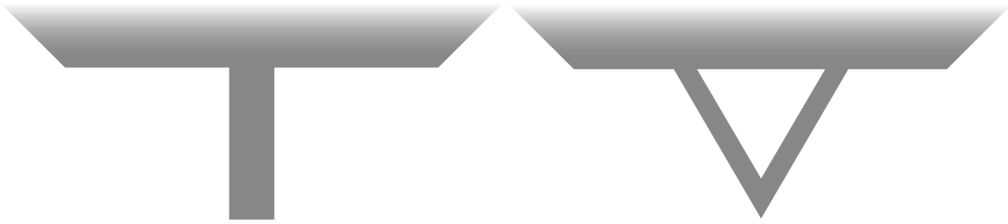
$$\rho \eta A \frac{\partial w}{\partial t} \tag{2.8}$$

added to the left side of (2.1) [29]. The value of the damping factor η is not easy to determine, as it is usually dependent on several factors, such as frequency and cantilever shape [29]. A common solution for modeling the cantilever is to use measurement values for the quality factor Q to calculate the damping factor instead of values calculated from a model [30]. Q, for a harmonic oscillator, can be determined as the ratio between the resonance frequency and the FWHM of the resonance. It is also related to the envelope of a decaying oscillation $\exp(-t/\tau)$ by $\tau = Q/\pi f_0$.

Several expressions for the SNR at a deflection Δz in a beam deflection setup can be found in literature (e.g. [31, 32, 25]). Generally, the SNR is proportional to the diameter of the laser beam and to the square roots of the laser power and the PSD sensitivity. It is inversely proportional to the length of the cantilever and the square root of the laser wavelength. If the laser spot diameter is larger than the diameter of the back side of the cantilever, the SNR is decreased by the ratio of spot size and cantilever back side [32].

These relations make clear that the cantilever selection has to balance several parameters:

- Shorter cantilevers result in higher deflection sensitivity.



(a) beam shaped (b) V-shaped

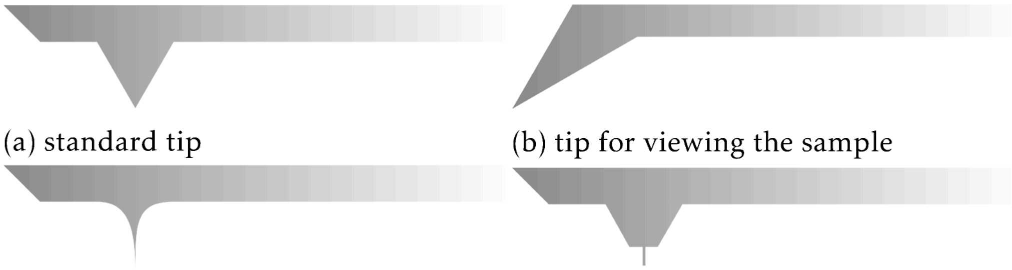
Figure 2.3.: Two common cantilever shapes.

- However, shorter cantilevers also have significantly higher force constants ($k \propto 1/L^3$).
- To compensate a change of the cantilever length to half while keeping k constant the width has to be decreased to $1/8$.
- Decreasing the width of the cantilever in turn decreases the sensitivity by reducing the amount of power on the detector and by increasing the laser spot size on the PSD through diffraction of the laser beam, once the laser diameter is larger than the cantilever backside.

Cantilever materials are currently dominated by silicon and silicon nitride [33]. Metal coatings of several tens of nanometers of Au and PtIr can be added to make the cantilevers chemically inert and conductive. Backside coatings of Al increase the reflectivity for higher SNR[1]. Currently available cantilevers for contact mode are predominantly beam shaped, with a few cantilevers having other shapes, such as V-shapes, available as well (see fig. 2.3). Most beam shaped cantilever have their tip slightly set back from their forward end, however, cantilevers with the tip extending past the cantilever are available as well (see fig. 2.4 for common types of tips). V-shaped cantilevers - counter-intuitively - seem to be more susceptible to lateral twisting beam shaped ones [34].

Finally, the tip of the cantilever is important for imaging quality of the AFM. The topography image is a convolution of the AFM tip shape and the actual sample topography [33]. Hence, important parameters of the AFM tip are not only its tip radius (usually in the range of $\approx$10 nm) but also its aspect ratio and shape. Low aspect ratio tips are not able to measure into deep and narrow grooves and will broaden steep sample features.

[1] Information taken from vendor descriptions at www.nanoandmore.com



(a) standard tip

(b) tip for viewing the sample

(c) symmetric tip

(d) high aspect ratio tip

Figure 2.4.: Sketches of several tip shapes found in commercially available AFM cantilevers

2.3. Contact Mode Measurement Parameters

For a constant force contact mode measurement, three important parameters have to be set. These are the deflection set point and the parameters P and I of the proportional-integral-derivative controller (PID).

In constant force contact mode positioning of the z actor - and thereby the generation of the topography image - is done by a PID. This type of controller compares the difference between deflection set point V_{set} and the current deflection V and updates the output of z actor according to

$$\frac{dz}{dt}(t) = P\,(V - V_{set})(t) + I \int\limits_0^t (V - V_{set}) + D\,\frac{d(V - V_{set})}{dt}(t) \tag{2.9}$$

The grayed out part of (2.9) is often omitted in AFM, as it is very sensitive to noise, resulting in a proportional-integral controller (PI). The coefficients P and I are tuned by the user during the run time of the experiment. Generally, higher coefficients mean faster reaction to the surface topography but also a higher proclivity for unstable behaviors, so called "ringing".

The deflection set point V_{set} is also a user selected parameter. Generally, lowering the deflection set point decreases the force with which the cantilever tip is pressed against the sample, while raising it does the opposite. If the set point is too low, the AFM is not operating in the repulsive domain and the contact between tip and sample might be lost at any time. High set points increase the tip wear and sample degradation. The deflection detected by the PSD is not a direct measure of the tip sample force, as a zero read out value does not mean that the cantilever is in a neutral position. Furthermore, the slope of the deflection read out is a function of the cantilever deflection and not the tip sample force.

In this work, the tuning of the measurement parameters was performed as in *Atomic Force Microscopy* [33, pg. 92ff]. With the exception that the PSD was positioned so that the free cantilever had a deflection read out in the range from −1 V to −0.5 V instead of 0 V, thus putting the deflection read out close to the most sensitive region around 0 V once the cantilever was approached.

2.4. Contact Resonance

A cantilever in contact with the sample can be modeled by (2.1) in a similar way as a free cantilever. However, the boundary conditions change have to be changed to include the tip sample force. A commonly used model for the forces between tip and sample is the Derjaguin-Müller-Toporov model (DMT model) [35] (printed here in the version found in [36] using the damping term from [37]):

$$F_{DMT}(d_n) = \begin{cases} -\frac{\mathrm{H}r_{tip}}{6a_0^2} + \frac{4}{3}\mathrm{E}^\star \sqrt{r_{tip}\left(a_0 - d_n\right)^3} - \frac{\pi r_{tip}\eta_n}{h}\sqrt{a_0 - d_n}\dot{d}_n & d_n < a_0 \\ -\frac{\mathrm{H}r_{tip}}{6d_n^2} & d_n \geq a_0 \end{cases} \tag{2.10}$$

(2.10) models the interaction between cantilever and sample as an interaction between a sphere of radius r_{tip} and a planar surface. Two cases are discerned by the normal distance d_n between tip and surface. When the distance is larger than the intermolecular distance a_0 then only (attractive) van der Waals interactions are considered, once d_n is smaller than the intermolecular distance repulsive mechanical interaction $\frac{4}{3}\mathrm{E}^\star \sqrt{r_{tip}\left(a_0 - d_n\right)^3}$ and damping $-\frac{\pi r_{tip}\eta_n}{h}\sqrt{a_0 - d_n}\dot{d}_n$ are acting as well.

For small variations in deflection around an equilibrium position d_0 (2.10) can be approximated by single spring as [36]

$$k^\star = -\frac{\partial F_{DMT}}{\partial d_n}\bigg|_{d_n = d_0} = \frac{4}{3}\mathrm{E}^\star \sqrt{R(a_0 - d_0)} \tag{2.11}$$

When the tip is mounted at the end of the cantilever the boundary condition describing the shear force at the formerly free end is changed from $\frac{\partial^3 y}{\partial x^3} = 0$ to $\frac{\partial^3 y}{\partial x^3} = \frac{3k^\star}{k_C L^3} y$ [2] to add the influence of the tip sample force on the cantilever to the description. The resonance frequencies of this new system are different to that of the free cantilever. For $k^\star < k_C$ the system can be approximated using a point mass model [2], resulting in a very simple expression for the frequency shift due to tip sample contact

$$\frac{f_{res}}{f_0} = \sqrt{\frac{k_C + k^\star}{k_C}} \tag{2.12}$$

The spring constants of higher order modes can be calculated from $\frac{f_i}{f_j} = \frac{k_i}{k_j}$ [38]. As higher vibrational modes should have higher frequencies, they are less susceptible to changes in the sample properties.

As most cantilevers are not mounted parallel to the surface but instead at an angle α of 10° to 15° the vertical spring constant of the sample has to be scaled by $\cos\alpha$ [39].

In addition to vertical bending modes, AFM cantilevers also have torsional modes, where the cantilever is twisted sideways and the tip moves left and right. Coupling between torsional and bending modes can lead to shifts in the resonance frequency [36]. This coupling is more likely in measurement schemes where vibrations are excited via the surface - as is the case for PTIR [36].

3. Photothermal Induced Resonance

3.1. Historical Overview

The idea of using scanning probe techniques to increase spatial resolution in mid-IR spectroscopy has found several tentative implementations. Two main approaches to near field mid-IR imaging can be made out:

1. Scanning near-field optical microscopy (SNOM) based techniques, where light is detected directly after interaction with the sample and

2. techniques based on sample heating, where the effect of the interaction of light and sample is detected via the temperature change it generates in the sample.

Hammiche et al. proposed to use a scanning thermal probe for detecting local IR absorption. In this implementation, the sample was either illuminated with the output of an FTIR interferometer [40], of a CO_2 laser or the output of an optical parametrical generator (OPG) [41]. Local heating was detected using a scanning thermal microscopy (SThM) probe consisting of a bent Wollaston wire with a Pt/Rh core. It was shown that local IR spectra collected with this setup are comparable to those measured in transmission. The main drawback of the technique was due to the reduction of spatial resolution by thermal diffusion. The thermal diffusion length, the distance after which the modulation of the temperature has been decreased to $1/e$ of that at the maximum is given by $L_t = \sqrt{\frac{2\mu}{f}}$ where μ is the thermal diffusivity and f is the modulation frequency. Typical values of thermal diffusivity lie in the range of around 100 mm²/s for metals to around 0.1 mm²/s for polymers. For an interferometer based system the modulation frequency depends on the wavelength and the mirror speed. In their setup Hammiche et al. found thermal diffusion lengths in the range of 5.5 µm to 18 µm. Due to the low SNR of their system 1000 single scans of an FTIR spectrometer were averaged leading to acquisition times of 30 min for a single spectrum [42]. Using a higher powered OPG source, faster acquisition in the range of a few minutes was possible [42]. It is interesting to note that a comparison of the local heating signal with direct detection of sample expansion using a standard Si cantilever resulted in a decrease in SNR in comparison to thermal detection [43].

SNOM based techniques can be split into two types based on their near-field setup [21]:

- *Aperture* techniques, in which a probe with a small aperture, e.g. a tapered wave guide fiber, is moved across the sample to generate high spatial resolution, and
- *Scattering* techniques, which detect light by introducing a sub wavelength sized feature, such as a nano-particle or a scanning probe tip into the electromagnetic near-field of the sample.

Aperture SNOM in general suffers from two important drawbacks: as the aperture diameter is reduced below the wavelength of the light the transmittance decreases rapidly ($T \propto R^4$ for small diameters [44]) and the divergence of the transmitted light increases. Hence, very high resolution imaging is only possible for very thin layers. Due to the small transmittance of the fiber high powered light sources, such as free electron lasers (FELs) are necessary [45]. The strong divergence of the transmitted light entails that the sample layer has to be thinner than the radius of the aperture for the FWHM of the emitted wave to be not significantly larger than the aperture diameter [46].

$AgCl_xBr_{1-x}$ fibers are commonly used due to their broad transmissio range in the mid-IR [47]. Light is detected using MCT [44] or InSb detectors [45]. Of course, as a transmission based technique, aperture based SNOM suffers from the problem of measuring a small change on a high background, i.e. the maximum intensity is measured when the target signal - absorption - is zero.

In scattering-SNOM (s-SNOM) the change of the emitted radiation upon introduction of a scatterer in the near field of the sample is detected in the far-field. Such a scatterer can be a sharp metal tip, such as that of a metalized AFM cantilever. The challenge of s-SNOM is to extract the small change in the light intensity coming from the sample due to bringing the scatterer close to the sample. Current implementations of mid-IR s-SNOM are based on interferometric detection of the scattered light. The light scattered off the sample interfers with a reference beam in an interferometer and the local optical constants are determined from the interferogram. Through using a pseudoheterodyne detection scheme, in which the cantilever position and the interferometer oscillate at different frequencies, the real and imaginary part of the local refractive index of the sample can be detected background free [48].

s-SNOM can be performed either with thermal light sources [49], broad band laser sources [50] or monochromatic CW laser sources, such as mode-hop free EC-QCLs [51]. The spatial resolution (in all three dimensions) is defined by the region of field enhancement around the metallic tip of the cantilever, which is in the range of tens of nanometers [52]. By modifying the angle of incidence of the light, the dimensions of the field enhanced region can be modified to achieve depth resolution [52]. Through pump-probe experiments, picosecond time resolution has been demonstrated [53, 54]. Furthermore, s-SNOM has been shown to have the ability to analyze single protein complexes [51]. Through images of the electromagnetic field at different wavelengths the dependence of plasmon modes on the wavelength can be determined and compared to theoretical calculations [55, 56].

3.2. Photothermal Expansion Induced Resonance

Photothermal induced resonance (PTIR) works on the same premise as thermal near field IR imaging described in section 3.1: The sample is illuminated with an intensive light source and local sample heating upon absorption is detected. The main improvement in PTIR is circumvention of loss of spatial resolution through thermal diffusion. PTIR achieves this by using a light source that emits short pulses for excitation [57]. The local thermal expansion is not detected via the change in sample height but via the ring down motion of the cantilever. Upon being illuminated with the laser pulse the sample expands rapidly, leading to a spike in the tip-sample force. This short spike excites vertical vibrational contact modes of the cantilever. As Dazzi et al. [58] have shown, the amplitudes of the vibrations are proportional to the local IR absorption signal (see also section 3.2.3). Recent reviews on the topic of PTIR from researchers in the field are "Infrared Imaging and Spectroscopy Beyond the Diffraction Limit." by Centrone [59] and "AFMIR: Combining Atomic Force Microscopy and Infrared Spectroscopy for Nanoscale Chemical Characterization" by Dazzi et al. [60].

3.2.1. Instrumentation

In the original publication by Dazzi et al. CO_2 laser and FEL sources were used to excite a PTIR signal [61]. Both sources have in common that they offer high output power (Centre Infrarouge d'Orsay FEL:100 MW peak for 1 ps duration pulses, 1 W at 16 ns long pulses; optical parametric oscillator (OPO): 7 mJ). OPO sources where later introduced for PTIR due to their size and tunability - OPOs covering the mid-IR range are bench top size instruments while FELs need dedicated facilities and CO_2 lasers only emit at CO_2 lines. The commercial PTIR system nanoIR (Anasys Instruments) used an OPO source covering 1025 cm^{-1} to 4000 cm^{-1}, while custom setups using laser sources delivering 0.625 cm^{-1} to 6450 cm^{-1} were demonstrated[62]. Lu and Belkin introduced EC-QCL as light sources in PTIR in combination with resonant excitation of the cantilever [63, 38]. PTIR has also been shown to work with terahertz QCLs [64].

PTIR can be performed in top and in bottom illumination. Bottom illumination was used in Dazzi's original implementation [61] and in the nanoIR instrument. Illumination of the sample was implemented through a ZnSe ATR element. ATR is a phenomenon that can occur when light coming from a high refractive index material (here ZnSe or ZnS) with refractive index n_1 hits the interface to a material of lower refractive index n_2 (e.g. air or sample). When the angle of incidence θ_1 is larger than the critical angle

$$\theta_c = \arcsin(\frac{n_2}{n_1}) \tag{3.1}$$

light can not propagate into the medium of lower refractive index. Instead, the envelope of the electric field in the medium of lower refractive index takes the shape of

$$e^{-x\frac{2\pi}{\lambda_0}\sqrt{n_1^2\sin^2(\theta_1)-n_2^2}}. \tag{3.2}$$

Hence, the light intensity exponentially decays when moving further from the surface. How quickly the intensity decays is usually described by the depth of penetration

$$d_p = \frac{\lambda_0}{2\pi\sqrt{n_1^2\sin^2(\theta_1)-n_2^2}} \tag{3.3}$$

the distance at which the light intensity has decreased to $1/e^2$ of the intensity on the surface (for a more in depth treatment of ATR for spectroscopy see [65, 18]). Typically, the depth of penetration for IR wavelengths lies in the range of several micrometers. ATR illumination has two important properties for PTIR measurements:

1. While light is not propagating into the sample, it can still interact with it, thus allowing absorption and therefore PTIR measurements.

2. Due to the short depth of penetration most of the light intensity is concentrated in the sample and hardly any intensity reaches the cantilever. This helps to reduce the background contribution of light absorbed inside the cantilever.

Absorption by the cantilever can be further reduced by using gold coated cantilevers.

ATR illumination also places some restrictions on the samples that can be measured. Samples have to be placed on an ATR element and have to be thin enough to be in the intense part of the evanescent field. The maximum sample thickness is thus constrained to about 1 µm before the linearity between sample thickness and PTIR signal is lost [66].

Top illumination does not have these restrictions on sample dimensions and preparation. Instead of a prism, flat IR-transparent or reflective substrates can be used. This also allows taking advantage of near-field field enhancement taking place between tip and substrate to increase the signal intensity. By doing so even mono-layer samples have been measured successfully [38]. The update of the commercial PTIR instrument, the nanoIR 2 (Anasys Instruments), is designed to use top illumination.

3.2.2. Applications of PTIR

PTIR has found applications across many different disciplines including biology, medicine, material sciences and plasmonics.

Earliest application of PTIR are the analysis of single cells [67, 61, 58]. PTIR was used to detect the location of cell cores and viruses inside *E. Coli* [58] and to image the distribution of a chemotherapeuticum inside a human cell [68]. Further application on single or several cells include mapping of inclusion bodies in cells [69, 70, 71]. Sample preparation usually consists of simple deposition of the sample and drying before the measurement. Recent work demonstrates the possibility to map the distribution of proteins inside cells using resonance enhanced PTIR [72].

Structured photo-resist samples are commonly used as references for PTIR. The method is, however, also used to gain insights about polymers. The ability of PTIR to help in the reverse engineering of polymer samples has been repeatedly demonstrated [73]. High resolution chemical analysis also gives insights into the local differences in polymer composition [74] and in the polymerization [75]. PTIR has been used to analyze water uptake in polymers [76].

Several works have used PTIR to analyze the products of chemical reactions. Rosen et al. used the method to ascertain homogenous stripping of oleates from PbSe surfaces [77]. The combination of topographical and chemical information available in PTIR was used to compare different experimental parameters for the synthesis of PEDOT [78, 79] and to detect the difference in composition across a single crystal of a mixed-ligand metal-organic framework [80].

PTIR can be used to map the field distribution in plasmonic structures. Such measurements are of interest as without near-field imaging only the far-field effect of these structure can be measured while the actual field distribution around the structure can only be simulated. While PTIR can be used to directly image the local thermal expansions of resonators [81] in many cases a thin polymer layer (such as poly(methyl methacrylate) (PMMA)) is spun unto the resonator to increase the thermal expansion and thereby the PTIR signal [82, 83]. As the PMMA layer molds itself to the underlying substrate the location of the plasmonic structure can still be detected through the topography image.

3.2.3. The Photoexpansion Signal

An important factor to allow IR spectroscopy using PTIR is that the detected signal is in proportional to the imaginary part of the refractive index. Several researchers [39, 84, 38] have developed theoretical descriptions of the PTIR signal to analyze the relation between refractive index and measured amplitude.

Dazzi, Glotin, and Carminati [39] model the PTIR signal as the interaction between a spherical absorber in the evanescent field of an ATR setup and a two dimensional cantilever. The absorber is a weakly absorbing sphere of constant (complex) refractive index with uniform mechanical properties. The cantilever is modeled as a dampened Euler-Bernoulli beam without considering twisting. The system is excited with a single laser pulse, with a pulse duration in the nanosecond and tens of microsecond range being treated as two cases with slightly different properties.

The amplitudes of the cantilever modes in PTIR was determined to be proportional to the local IR absorption using several proportionality constants [39]:

$$\tilde{S}_n(\omega_n, \lambda) = H_m H_{AFM} H_{opt} H_{th} \frac{\Im(n(\lambda))}{\lambda} \tag{3.4}$$

$$H_m = k_z \alpha_{sph} a$$

$$H_{th} = \begin{cases} \frac{6\pi}{\rho_{sph} C_{sph}} t_p \left(\frac{t_p}{2} + \tau_{relax} \right) & t_p \ll \tau_{relax} \\ \frac{4\pi a^2}{\kappa} \left(\frac{\sin(\omega_n t_p/2)}{\omega_n} \right) & \tau_{relax} \ll t_p \end{cases}$$

$$H_{opt} = \frac{\Re(n)}{(\Re(n)^2 + 2)^2} c \epsilon_0 \left| E_{inc} \right|^2$$

$$H_{AFM} = \frac{1}{\Gamma \omega_n} \left[\cos(\alpha) \delta x + \sin(\alpha) H \right] \frac{D}{\rho A L} \left[\left. \frac{\partial g_n}{\partial x} \right|_{x=L} \right]^2$$

The four proportionality factors in (3.4) each describe the contribution of one part of the system to the amplitude $\tilde{S}_n(\omega_n, \lambda)$ at the resonance of a cantilever mode.

$$H_m = k_z \alpha_{sph} a \tag{3.5}$$

describes the contribution of the expanding sphere to the system, where k_z is the spring constant of the linearized force between tip and sample, α_{sph} is the thermal expansion coefficient of the sample, and a is the radius of the sphere.

$$H_{AFM} = \frac{1}{\eta\omega_n}\left[\cos(\alpha)\delta x + \sin(\alpha)H\right]\frac{D}{\rho AL}\left[\frac{\partial g_n}{\partial x}\Big|_{x=L}\right]^2 \tag{3.6}$$

depends on the properties of the AFM cantilever. $[\cos(\alpha)\delta x + \sin(\alpha)H]$ describes how the angle of the cantilever influences the signal, with α being the cantilever angle, H the tip height and δx the distance of the tip from the end of the cantilever. η is the damping of the cantilever, and ω_n the resonance frequency of the mode. D is the length of the cantilever that is illuminated by the laser spot and $\frac{\partial g_n}{\partial x}\Big|_{x=L}$ is the slope of the cantilever at its tip.

$$H_{opt} = \frac{\Re(n)}{(\Re(n)^2+2)^2} c\epsilon_0 |E_{inc}|^2 \tag{3.7}$$

describes the influence of the real part of the refractive index $\Re(n)$ and the incident optical power $|E_{inc}|^2$ on the signal. In [39] this part is derived from modeling the absorber as a single dipole, which means that the decay of the electromagnetic field across the absorber can be neglected.

Depending on the length of the pulse t_p relative to thermal relaxation time $\tau_{relax} = \rho_{sph}C_{sph}a^2/3\kappa$, two different thermal factors were discerned in [58]:

$$H_{th} = \begin{cases} \frac{6\pi}{\rho_{sph}C_{sph}} t_p \left(\frac{t_p}{2} + \tau_{relax}\right) & t_p \ll \tau_{relax} \\ \frac{4\pi a^2}{\kappa}\left(\frac{\sin\left(\frac{\omega_n t_p}{2}\right)}{\omega_n}\right) & \tau_{relax} \ll t_p \end{cases} \tag{3.8}$$

Here κ is the effective heat conductivity and C_{sph} is the thermal capacity of the sample. In the case of shorter pulse length the sample temperature never reaches an equilibrium temperature. Instead, the sample is - on the timescale of the cantilever mode - immediately heated up, the temperature then decays exponentially. On the other hand, for long pulse durations, the temperature quickly reaches an equilibrium temperature, determined by the absorbed power, the size of the absorber and the effective thermal conductivity κ. An interesting part of (3.8) is the factor $\sin(\omega_n t_p/2)$. It means, that for long pulses, the signal not only depends on the input intensity but also on the product of pulse length and resonance frequency. As the pulse length is in this case on the time scale of the cantilever vibrations, this can lead to an increase in the signal when $\omega_n t_p \approx (2N+1)\pi$ and no signal around $\omega_n t_p \approx 2N\pi$.

While (3.4) on the first glance shows a linear dependence of the amplitude on the imaginary part of the refractive index, a closer inspection of the proportionality constants show influences of additional sample parameters. As these should remain approximately constant at one sample position the band positions and relative intensities in the spectra measured in ATR based PTIR look similar to those measured in ATR absorption spectra, once they are normalized to the laser power $|E_{inc}|^2$. Things look different, however, when spectra taken of different materials are compared.

The sample properties influence H_m through all three parameters:

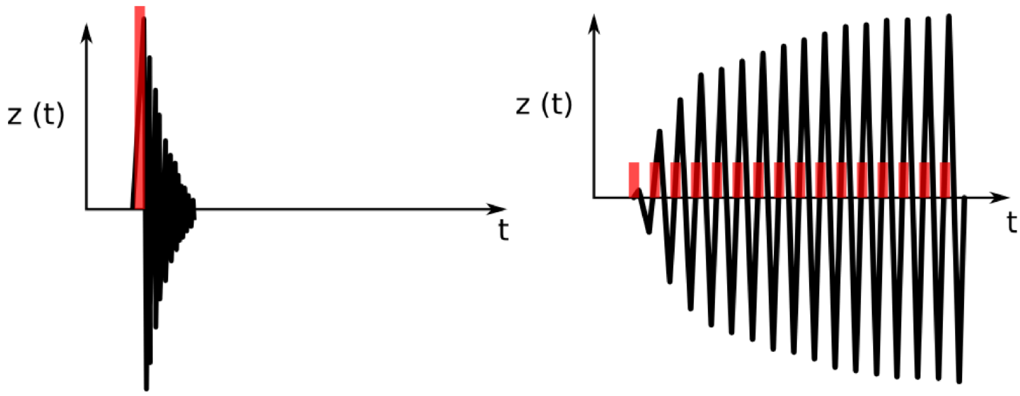
- k_z differs for different material depends on d_0 and $E^\star$. These values depend on the sample mechanical properties E and ν (see (2.11)).
- α_{sph} is a material parameter.
- Of course, the sample height a can also differ for different parts of the sample.

The resonance frequencies ω_n and the damping factor η in H_{AFM} are dependent on tip sample interactions. In H_{opt} $\Re(n)$ is a material parameter that for most organic samples is somewhere around 1.3 to 1.5, however, for some inorganic samples (e.g. GaAs $n \approx 3.3$), higher refractive indices can be encountered. Finally, the thermal conductivity and sample dimensions influence H_{th}. In the case of $t_p \ll \tau_{relax}$ density and sample thermal capacity also influence H_{th}, although the influence is small as t_p is much smaller than τ_{relax} and $\tau_{relax}/\rho_{sph}C_{sph} = a^2/3\kappa$. Finally, τ_{relax} depends on the size of the absorber and the thermal diffusivity μ:

$$\tau_{relax} = \frac{a^2}{\mu}. \tag{3.9}$$

As the most important material contributions to the scaling factors are α_{sph} and κ, Katzenmeyer, Aksyuk, and Centrone have defined a sensitivity factor α_{sph}/κ for PTIR that can be used to compare the signal intensities detected on different samples.

The laser intensity is directly proportional to the cantilever amplitude signal. This means that the PTIR signal can be normalized to laser intensity measurements from a power meter. It also means that higher powered lasers give better



(a) ring down (b) resonant

Figure 3.1.: Schematic depiction of the laser pulses (red) and cantilever deflection (black) in resonant and ring down excitation of the AFM cantilever. Laser pulse widths are not to scale.

signals as long as they don't change the sample itself. However, it should be noted that for very short pulses

$$H_{th} \approx \frac{6\pi}{\rho_{sph} C_{sph}} t_p \frac{a^2}{3\kappa}, \tag{3.10}$$

meaning that the laser pulse energy $t_p |E_{inc}|^2$ instead of just its intensity is the determining factor for the sensitivity. However, the signal enhancement achievable this way is limited by the thermal stability of the sample. Katzenmeyer, Aksyuk, and Centrone note, that the output of their OPO source is easily sufficient to melt polymer sample, thus restricting the maximum power that can be safely used for PTIR measurements of such samples [62].

A different approach to PTIR was taken by Lu and Belkin [63]. Instead of illuminating their sample with a single strong pulse, in their method the sample is illuminated with a series of weaker pulses. When the laser pulse repetition rate is tuned to one of the contact resonances of the cantilever, most of the energy transferred to the cantilever is stored between pulses, allowing to amplify the input energy over time. The difference in excitation methods is depicted in figure 3.1.

Lu, Jin, and Belkin provide a semi-empirical description of the signal in their resonant excitation scheme in [38]. The cantilever in contact with the sample is approximated as a harmonic oscillator by using experimental values to calculate Q-factor and spring constant. Excitation is simplified as a train of δ pulses of height $I_0 = F_{abs}\tau_{pulse}$. F_{abs} is determined using (2.11) as

$$F_{abs} = k^{\star}\Delta d_n. \tag{3.11}$$

Δd_n was determined by multiplying the sample heating during a laser pulse with literature values for the thermal expansion of the sample. Sample heating was calculated using a finite element method (FEM) simulation. The underlying assumption of calculating the energy input into the cantilever this way is that the cantilever movement is much slower than the sample expansions, thus during the pulse only the movement of sample is considered. Essentially the force curve in which the tip is positioned changes at the start of the pulse and at its end (see fig. 3.2). When the delta pulses are repeated at the resonance frequency f_0 of the cantilever, Lu, Jin, and Belkin calculate a sinusoid oscillation of the cantilever

$$z(t) = \frac{1}{t_p}\frac{2QI_0}{k}\sin(\omega_0 t) \tag{3.12}$$

The enhancement in resonant PTIR is due to the resonant excitation of the cantilver: the energy transferred to the mode corresponding to the laser repetition rate is stored in the cantilever leading to an oscillation that is by a factor of Q higher than the oscillation induced by a single pulse. For common cantilevers the Q-factor is in the range of ≈ 100.

3.3. Detection Schemes

In PTIR absorption of the IR laser pulse introduces cantilever oscillations which can be detected as oscillations of the deflection signal of the AFM cantilever. However, at the same time, scanning the cantilever across the sample also introduces changes in the deflection signal. To perform IR measurements the oscillations due to IR absorption have to be split from the deflection signal of the AFM cantilever. Retrieval of the IR signal is possible at all because it is usually found in a different frequency range than those induced by the topography of the sample. The frequency of IR information depends on the frequencies of the contact resonances of the cantilevers (usually found in the range of 10 kHz and above) while that caused by the topography depends on the sample shape and the scan speed and can be expected to lie in the range of 1 kHz. To recover the IR signal some sort of demodulation is needed. In the following sections demodulation schemes apt for different PTIR excitation modes are introduced and compared.

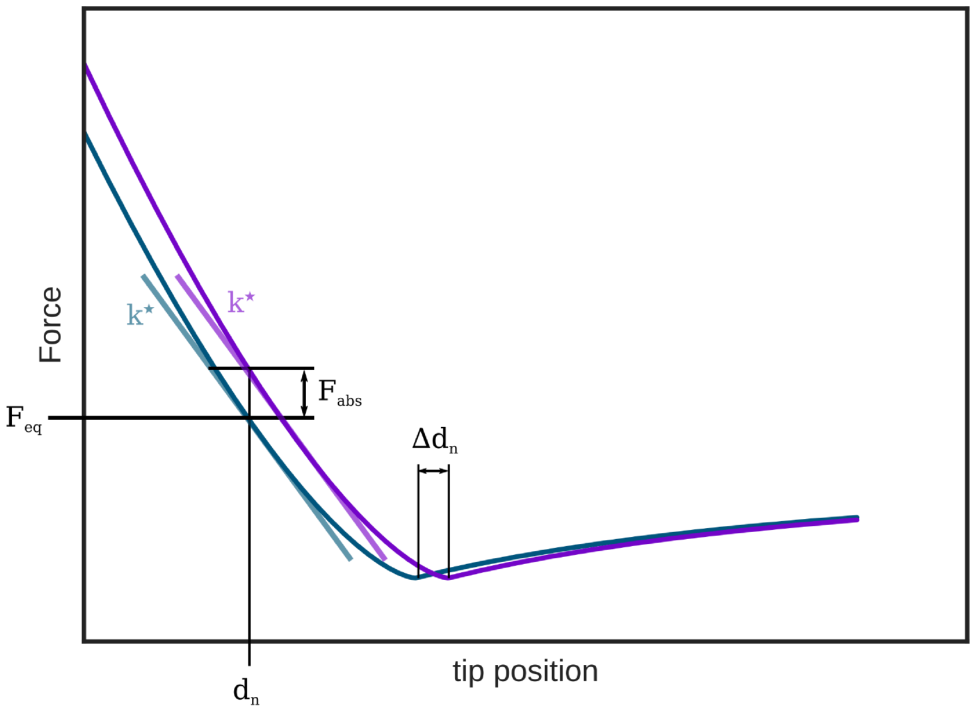


Figure 3.2.: Depiction of the tip sample force curve before (blue line) and during (purple line) a laser pulse. $k^\star$ is the linear approximation of the tip sample force around the equilibrium position d_n.

3.3.1. Ring Down Excitation

For ring down excitation the ring down motion of the cantilever is recorded and then split into the constituting frequencies. This can be done using a Fourier transform [57]. Fourier transform of the ring down signal ideally results in a spectrum of Lorentz peaks with center frequencies corresponding to the cantilever contact resonances and FWHM defined by the Q-factor. However, more complex methods based on wavelets were shown to improve the SNR of the detection significantly [85].

After having calculated a spectrum from the time domain signal the PTIR amplitude is determined as that corresponding to the highest peak within a given range. Other properties of the peaks, such their FWHM or their resonance frequency can be used to determine local material properties.

3.3.2. Resonant Excitation

In resonant excitation as introduced in [38], the laser pulse repetition rate is kept constant and therefore the cantilever is only excited at a single frequency. In this case signal demodulation can be performed using a lock-in amplifier. This instrument converts the oscillations of the input signal (in this case the cantilever deflection signal) to a slowly changing value corresponding to the amplitude.

A lock-in amplifier works by multiplying the input signal V_{in} with a sine and cosine at the frequency f_{ref} and then averaging the product over time to receive the in phase component X and the quadrature component Y:

$$X = \lim_{T \leftarrow \infty} \int_0^T V_{in} \cos\left(2\pi f_{ref} t\right) dt \qquad Y = \lim_{T \leftarrow \infty} \int_0^T V_{in} \sin\left(2\pi f_{ref} t\right) dt \tag{3.13}$$

From X and Y the magnitude of the signal can be calculated as

$$R = \sqrt{X^2 + Y^2} \tag{3.14}$$

and its phase relative to the reference pulse is

$$\phi = \arctan\left(\frac{Y}{X}\right) \tag{3.15}$$

A lock-in amplifier can be explained through the using the convolution theorem

$$\mathcal{F}\left(g(t)h(t)\right) = \mathcal{F}\left(g(t)\right) * \mathcal{F}\left(h(t)\right) \tag{3.16}$$

where $*$ denotes to convolution of two functions. For the in-phase component this means that Fourier transform of the signal is convolved with

$$\mathcal{F}\left(\cos(2\pi f_{ref} t)\right) = \frac{\delta\left(f - f_{ref}\right) + \delta\left(f + f_{ref}\right)}{2} \tag{3.17}$$

A cosine signal of frequency f_{sig} when convolved with the reference signal in the frequency domain

$$\begin{aligned}
&\mathcal{F}\left(V_{sig}\right) * \mathcal{F}\left(\cos(2\pi f_{ref} t)\right)(f) \\
&\quad = \int_{-\infty}^{\infty} A \frac{\delta\left(\xi - f_{sig}\right) + \delta\left(\xi + f_{sig}\right)}{2} \frac{\delta\left(f - \xi - f_{ref}\right) + \delta\left(f - \xi + f_{ref}\right)}{2} d\xi \\
&\quad = \frac{A}{4}\Big[\delta\left(f - (f_{sig} + f_{ref})\right) + \delta\left(f - (f_{sig} - f_{ref})\right) \\
&\qquad + \delta\left(f + (f_{sig} + f_{ref})\right) + \delta\left(f + (f_{sig} - f_{ref})\right)\Big]
\end{aligned} \tag{3.18}$$

leads to peaks at four different frequencies. For $f_{ref} = f_{sig}$ the Dirac delta peaks $\delta\left(f - (f_{sig} - f_{ref})\right)$ and $\delta\left(f + (f_{sig} - f_{ref})\right)$ both end up at $f = 0$. When using sines instead of cosines the same calculations can be performed for the quadrature signal. All peaks that have a non-zero frequency have a zero time average value and therefore don't contribute to the output of the lock-in amplifier. However, in an actual implementation of a lock-in amplifier averaging is only possible for a finite time. Instead of averaging rejection of other components of frequencies other than f_{ref} is then implemented using a low pass filter (see e.g. [86]). The finite width of the low-pass filter means that some components other than at f_{ref} can also pass, with narrower filter leading to a narrower range being passed. As very narrow settings of this low-pass filter also mean that faster changes in the oscillation at f_{ref} are being rejected and thus the output signal is distorted [87].

Using only the measurement at a single frequency has the significant drawback of making it impossible to ascertain during run time that the system is at maximum of the resonance. Changes in the resonance frequency happen as $k^\star$ changes across the sample (see (2.12)). For a horizontal cantilever, this entails mostly changes in the sample mechanical properties (E and ν). Additional changes, can be introduced by coupling between twisting and vertical bending modes see section 2.4. The effect of drifts in the resonance is depicted in figure 3.3.

Methods that allow to overcome this problem of resonance drifts are described in section 3.4.

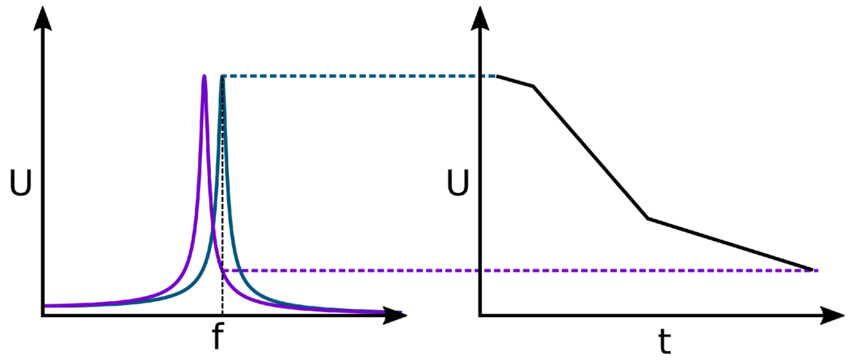


Figure 3.3.: Influence of the shift of the resonance curve on the signal recorded at a single frequency. In a single frequency based system, only the graph on the right side is accessible to the user.

3.4. Resonance Tracking

A multitude of methods have been published that can be used to track the contact resonance in AFM systems. In the following established methods will be discussed. It is important to note, that while no method to perform resonance tracking for resonance enhanced PTIR measurements has been published, a commercial supplier for PTIR instruments has implemented such method in their products[1].

Broad band excitation

To excite a system at a multitude of frequencies at once is to excite it with a single pulse of high intensity and low duration. The narrower a pulse is in the time domain, the broader it is in the frequency domain. This can be seen, for example, from the Fourier transform of the rect and the Gaussian function printed below:

$$\mathrm{rect}\left(\frac{t}{w}\right) \overset{\mathcal{F}}{\Longrightarrow} |w|\,\mathrm{sinc}\,(wf) \tag{3.19}$$

$$\exp\left(-\frac{t^2}{2w^2}\right) \overset{\mathcal{F}}{\Longrightarrow} \sqrt{2\pi w^2}\exp\left(-2\pi^2 f^2 w^2\right) \tag{3.20}$$

This type of excitation is used for ring-down PTIR, as introduced by Dazzi et al. [57]. However, broad excitation come with a draw-back as well: since the energy of the pulse is broadly spread across the frequency spectrum only a minute fraction ends up exciting each vibrational mode of the cantilever, while

[1] from personal correspondence with Dr. Kevin Kjoller, Anasys Instruments

the remaining part only heats up the sample. To determine the resonance curve of the cantilever the deflection signal is digitized and converted to the frequency domain using FFT.

An interesting alternative to pulsed excitation is band-excitation [88], where a precalculated waveform is used instead of a single pulse to excite the cantilever. The user chooses the frequency and phase spectrum of the excitation wave form, which is then Fourier transformed to a time domain signal. For analog excitation methods, such as an ultrasound transducer, the calculated waveform can be fed to the cantilever via an arbitrary function generator. As with single pulse excitation the frequency spectrum is again determined from the digitized cantilever oscillation via FFT. The full scheme is depicted in figure 3.4. The main advantage of this method is the possibility of concentrating the energy input on the frequency range of interest, while its main challenge for the application in PTIR is the translation of the input waveform to a laser pulse train or the need for a laser that allows arbitrary modulation of the output intensity.

With either method of broad-band excitation, further evaluations has to be performed on the recorded spectra before yielding data amplitude and Q factor.

Dual Frequency Resonance Tracking

In dual-frequency resonance tracking [89, 90] or dual AC resonance tracking (DART) the system is excited with sine waves of two different frequencies that are placed at a distance of $\pm\Delta f$ around a central frequency f_0. The cantilever deflection is then fed to two lock-in amplifiers set to $f_0+\Delta f$ and $f_0-\Delta f$, respectively. The amplitudes measured by these amplifiers is used to track the resonance of the system by continually shifting f_0 so that the amplitudes measured at $f_0 \pm \Delta f$ are equal, which is the case when the frequencies are placed symmetrically around the resonance (see fig. 3.5 for a visualization). Amplitude, phase and Q factor can be determined from the amplitudes and phases measured at the two excitation frequencies [89].

The advantage of DART is its simplicity. Lock-in amplifiers that already implement DART are commercially available (e.g. HF2LI, Zurich Instruments). However, for PTIR the main disadvantage is the fact that the maximum amplitude of the resonance is not accessible in a straight forward way and, as with any other technique that is based on exciting the system with a complex waveform, translation to a pulsed light source with a repetition rate that is not significantly higher than the resonance that is to be tracked is not straight forward.

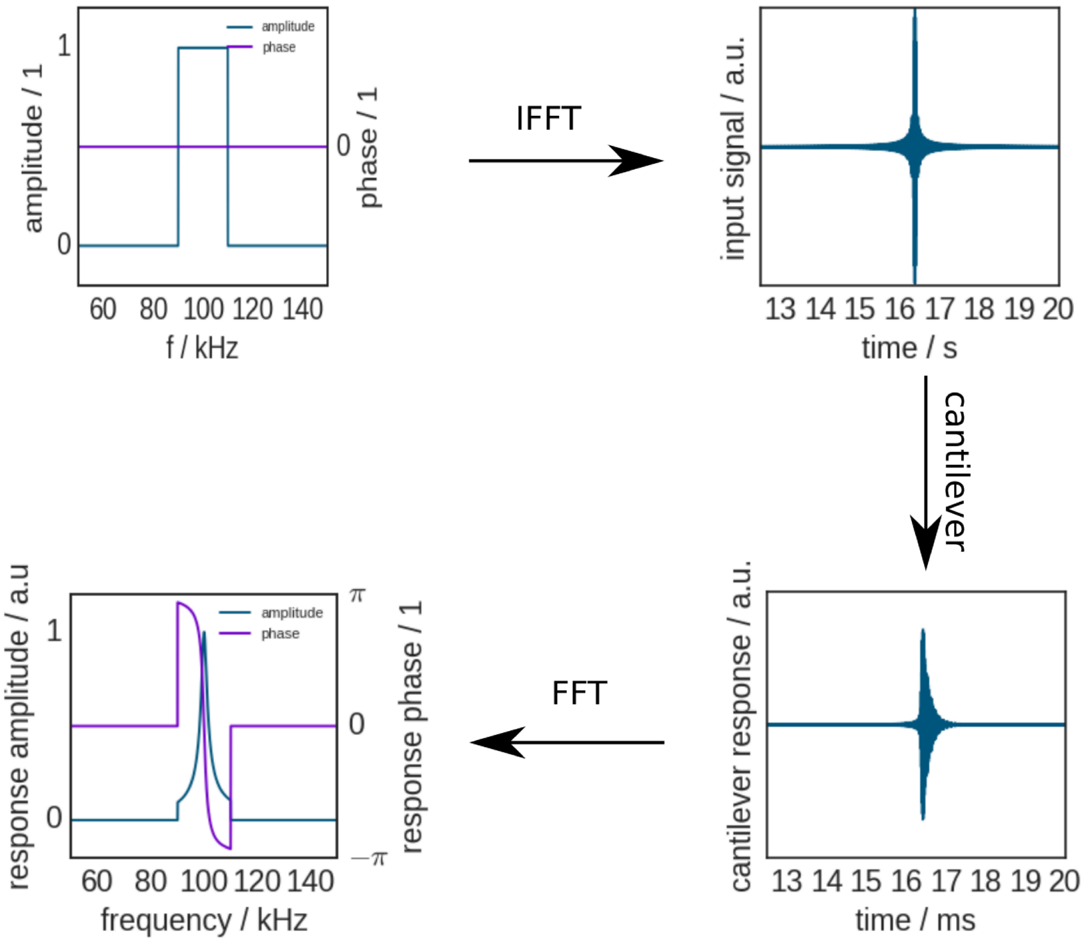


Figure 3.4.: Sketch of the signals in a band excitation scheme for AFM measurements.

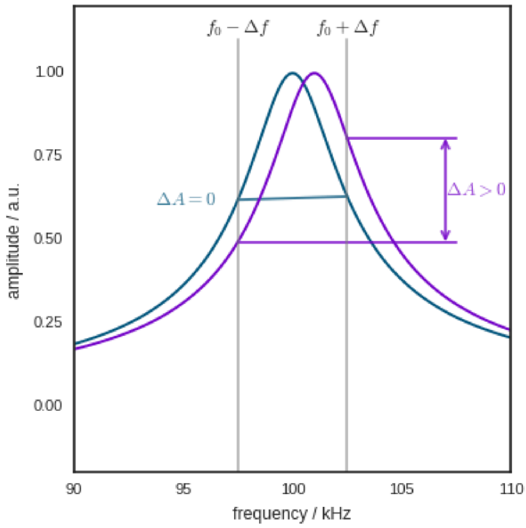


Figure 3.5.: Resonance detection in DART. If the resonance frequency is centered between the two excitation frequencies the difference of the amplitudes is zero (plotted in blue). When the resonance is closer to $f_0 + \Delta f$ the difference of amplitudes is positive (plotted in purple).

Phase Locked Loop

A phase locked loop (PLL) aims to keep the phase between an internal reference signal and an input signal constant. In an AFM this can be realized by using a voltage controlled oscillator (VCO) to drive the cantilever, a phase comparator that compares the output of the VCO to the cantilever deflection signal [91]. Since the phase at the resonance should ideally be 0°, a PLL can keep the excitation frequency at the resonance once it has locked onto it. PLL control is compatible with pulsed excitation by a laser. Techniques that exhibit large phase shifts between different points of the sample preclude the use of PLLs [90].

Frequency Sweep

Frequency sweep techniques detect the resonance of a system by rapidly probing the response of the system at a range of excitation frequencies. The resonance frequency is determined as the frequency corresponding to the highest amplitude. Since a complete sweep has to be performed for every measurement spot on the sample, sweep based techniques necessitate methods to rapidly - but predictably sweep the resonance frequencies, rapid amplitude detection schemes and rapid evaluation of the result.

In the work of Hurley et al. [92] as well as the follow-ups by Killgore and Hurley [93] and Kos, Killgore, and Hurley [94] excitation is either performed by a VCO coupled to a digital analog converter (DAC) controlled by a digital signal processing (DSP) chip or a by via direct digital synthesis (DDS) that was directly coupled to the DSP controller. For detection a root mean square (RMS) detector is used.

For a single sine wave the RMS value can be converted to the amplitude of the sine by multiplying with $\sqrt{2}$. In contrast to lock-in amplifiers which ideally detect the amplitude at a single frequency RMS detectors measure the RMS over a broad range of frequencies. However, by ensuring that the oscillations are predominantly at a single frequency, i.e. the frequency at which the system is currently excited, RMS can nevertheless be used for amplitude detection [94]. The upside of an RMS detector is its simpler implementation. RMS detectors are available as single integrated circuit (IC) components - such as the AD637 (Analog Devices) - while lock-in amplifiers are harder to construct needing either their own DSP or multiple ICs to generate a sine and cosine, multipliers, adders and so on.

Another reason for using a RMS detector is that for fast sweeps the narrow-band nature of the lock-in amplifier is somewhat negated by the need for a high cutoff-frequency for filtering the in-phase and quadrature components of the lock-in amplifier to not block fast changes in the amplitude during the sweep.

An important aspect of sweep techniques is that the Q-factor of the resonance is limiting the maximum possible sweep rate. The higher the Q-factor the longer energy that has been put into the oscillator can be stored in it. Hence, it takes longer for a high Q system to reach its maximum amplitude or for its amplitude to revert to zero ones the excitation has stopped. When the sweep rate is too high, the apparent maximum amplitude and Q-factor are decreased and the resonance frequency is shifted in the sweep direction [95].

Finally, as with single pulse or broad band excitation techniques, sweep techniques need a way to detect the location of the resonance. However, since the sweep range in sweep techniques also determines the sweep frequency, i.e. the number of completed sweeps within one second, it is beneficial to record just the frequencies around the resonance. To be able to do this the position of the resonance has to be determined in real time during the sweep. Known, working schemes are simple detection of the maximum amplitude [96] or more complex schemes to determine resonance, Q factor and amplitude through a polynomial fit [97, 94].

Part II.

Design of a PTIR Setup

4. Optical and Mechanical Considerations

4.1. Light Source

Due to the broad spectral features of solid and liquid phase Mid-IR spectroscopy single wavelength information is usually not sufficient for meaningful analysis[1]. Hence, a broadly tunable or broad band light source is necessary for PTIR. In the case of resonance enhanced - photothermal induced resonance (RE-PTIR) the repetition rate of the pulsed light source also has to be adjustable to the contact resonance of the cantilever. Finally, for imaging of single wavelengths the light source has to be able to emit monochromatic light (with a half-width corresponding to the wanted spectral resolution).

EC-QCLs combine all of these properties:

- As semiconductor lasers they are electronically pumped and their repetition rate can be quickly adjusted by changing the repetition rate of the input pulses. Typical repetition rates are in the kilohertz to low megahertz range.
- They are broadly tunable and are available at tuning ranges of several hundreds of wavenumbers across the whole fingerprint region of the mid-IR spectral range.
- EC-QCLs used for this work had a half-width around 1 cm^{-1} which is well below th width of solid and liquid mid-IR bands.

When multiple EC-QCLs are used together to achieve broader spectral coverage additional care has to be taken to ensure that the focal spot of all lasers are on the sample below the cantilever.

[1] A good example of this fact will be treated in chapter 8.

4.1.1. EC-QCL Sources Used in This Work

In this work two commercial EC-QCLs were used:

- A single element laser covering the range of 1729.30 cm^{-1} to 1565.06 cm^{-1} (Daylight Solutions Inc., San Diego, USA) at a duty cycle of up to 10 % and repetitions rates up to 350 kHz.
- A MIRcat (Daylight Solutions Inc.) source combining four EC-QCLs covering the ranges of 889.7 cm^{-1} to 1256.3 cm^{-1}, 1140.3 cm^{-1} to 1451.4 cm^{-1}, 1335.1 cm^{-1} to 1766.8 cm^{-1} and 2770.1 cm^{-1} to 2932.6 cm^{-1}.

Both lasers were chosen as light sources for the construction of the PTIR system due to their broad wavelength range, their high output power as well as their ease of use: selection of emitted wavelength, pulse length and repetition rate can be performed via a personal computer (PC) interface.

In the MIRcat the beams of all four lasers included in the system can be brought to the output of the source without any need for adjustment by the user. Grating positioning and QCL selection are performed automatically based on the wavelength selected by the user. The source can be controlled via a graphical user interface (GUI) or by calling the functions in a dynamic linked library (DLL).

In the MIRcat collinearity of the laser beams is achieved via a movable mirror, which means that the emission from only one QCL can be redirected to the output at a time. The MIRcat can be set to emit the light of any given wavelength in its tuning range or swept across parts or all of its range.

4.1.2. Fast Acquisition of EC-QCL Spectra

Spectral tuning in EC-QCLs is achieved via the rotatable grating inside the external cavity. This means that wavelength reproducibility and tuning speed in EC-QCLs are restricted by the accuracy and speed, respectively, at which the grating can be positioned. The lasers used in this work both have two different modes for grating positioning:

- In *tuning mode* the user selects a single wavelength to which the laser is then tuned and at which it is held,
- whereas in *scan mode* the user selects a wavelength range and an approximate scan speed at which the laser is scanned across the range. In scan mode no accurate information about the current wavelength is available to the user.

In tuning mode, moving the laser to a given wavenumber takes a few seconds, while scanning across the gain range of a single QCL chip depends on the size of the gain range but is in the range of several seconds, as well. Using tuning mode to acquire a spectrum across the gain range of a QCL chip would take several minutes. To achieve fast acquisition of spectra Brandstetter et al. [98] suggest characterizing the time-wavenumber relation in scan mode by performing a step scan measurement of the scan with an FTIR spectrometer. Once this characterization has been performed for a laser it is then possible to convert the time elapsed since the start of the scan to wavenumbers and thus record spectra in scan mode.

An introduction to step scan spectroscopy using an FTIR spectrometer is given in [3]. In brief: the step scan technique allows fast, time-resolved spectroscopy of repeatable experiments. In standard FTIR measurements the limiting factor for time resolution is the movement speed of the mirror of the interferometer: since the spectral resolution is defined by the path traveled by the mirror higher time resolution means higher mirror speed. The time needed to turn the mirror around at the edges of the interferometer is proportional to the square of the mirror travel speed.

The step scan technique removes the relation between mirror speed and time resolution. It does so by recording the signal at one interferogram position over time, then moving on to the next interferogram position and repeating the experiment. Once the trace at each interferogram position has been recorded, the data set can be Fourier transformed to time resolved intensity spectra (see fig. 4.1). The main requisite necessary to allow step-scan measurements is an experiment that can be reproduced multiple times (once for each interferogram position) and has a well defined starting point that can be made accessible to the spectrometer electronically. Single chip EC-QCLs as manufactured by Daylight Solutions fulfill both these conditions [99]: their sweeps are reproducible and they indicate the start of each sweep by raising a 5V transistor type logic (TTL) compatible output from low to high.

For the characterization of an EC-QCL scan the time-resolved spectra resulting from the step scan measurement are then evaluated to give a correlation between the time since the start of the scan and the wavenumber at which the laser is emitting the maximum intensity in a time slice. These data pairs are then fitted using a polynomial to be able to calculate the wavenumber for arbitrary scan times later on.

The step scan characterization of the single chip was determined at an earlier

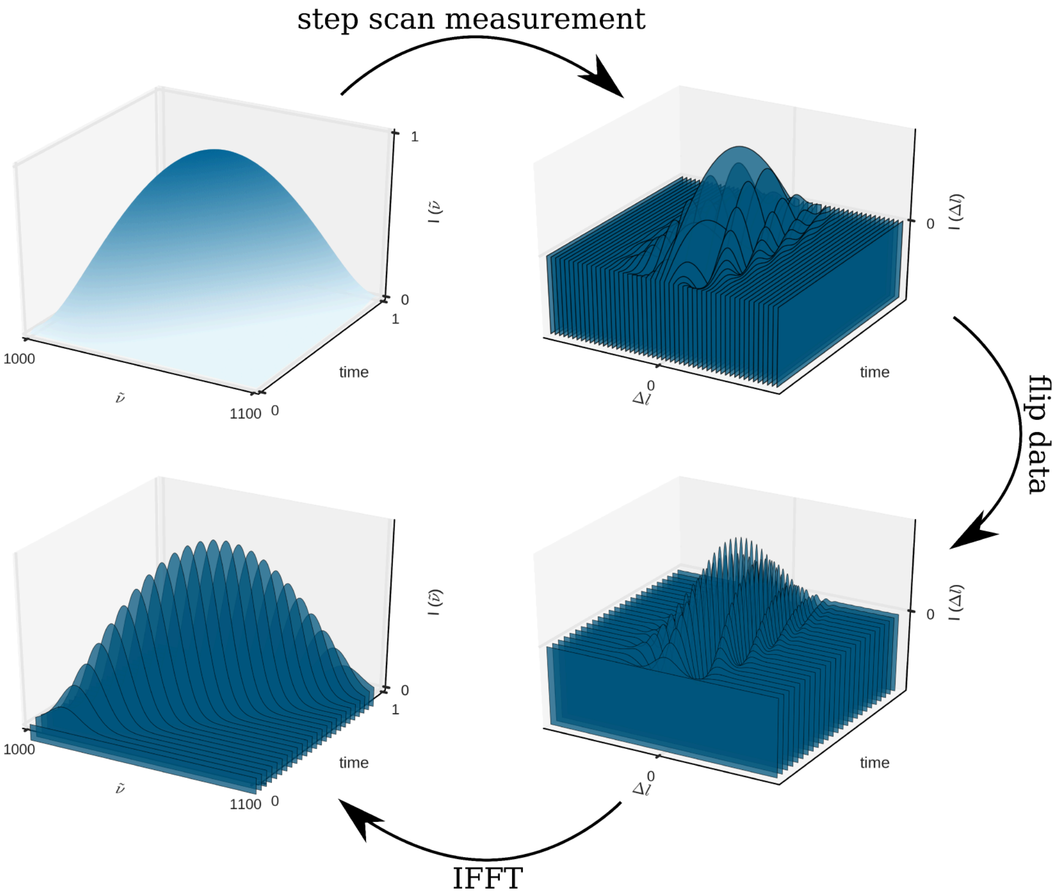


Figure 4.1.: Sketch of the signals in a step scan measurement. The input process is repeated multiple times, at least once for every mirror position. Once the traces at all the needed positions have been recorded, IFFT is applied along the Δl axis. The time resolution of the measured spectra is defined by the rate at which the traces were recorded.

time as:

$$\begin{aligned}\tilde{\nu}(t) =& 1.241 \times 10^{-9}\mathrm{cm}^{-1}\mathrm{ms}^{-3}t^3 + 7.468 \times 10^{-6}\mathrm{cm}^{-1}\mathrm{ms}^{-2}t^2 \\ &+ 0.09802\mathrm{cm}^{-1}\mathrm{ms}^{-1}t + 1564\mathrm{cm}^{-1}\end{aligned} \tag{4.1}$$

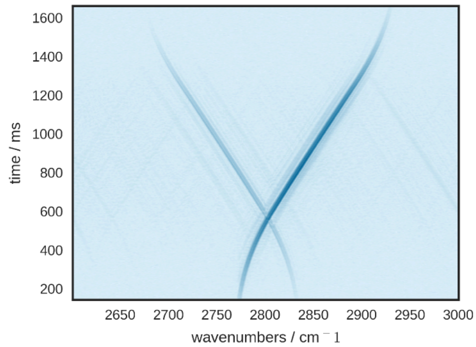
4.1.3. Step-Scan Characterization for Daylight Solutions MIRcat Sources

Step scan characterization of the MIRcat source adds some complications over those usually encountered when characterizing systems consisting just of a single chip. The three main complications are the longer duration of the scan of the laser across its full tuning range (close to one minute at full scan speed for the system used in this work), the interruptions in the laser emission caused by switching from one EC-QCL to the next and the broader spectral coverage of the laser.

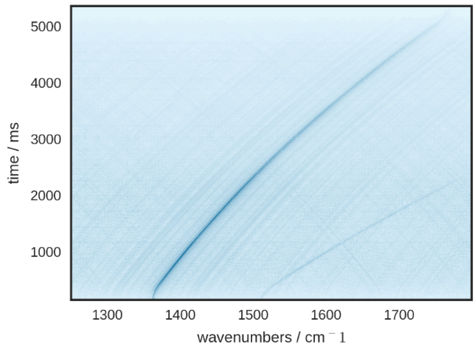
For a step scan measurement the laser scan has to be repeated several hundreds to several thousand times. The exact number of repetitions is given by the spectral range that is to be covered and by the wanted spectral resolution. To perform a step scan measurement of the full sweep of the MIRcat at 2 cm^{-1} spectral resolution 5700 steps of the interferometer would be necessary. At a scan duration of 1 minute the full characterisation measurement would take about 95 h. An experiment this long would entail that the detector of the FTIR spectrometer would have to be recooled several times during the measurement and would therefore be highly impractical - each time adding the chance of errors in the cooling and warming of the detector. In contrast, when each chip is measured by itself the time for the characterisation of the full chip can be shortened significantly because the scan time as well as the number of scans needed are both reduced.

When switching between different QCLs the MIRcat source waits until the upcoming QCL is at temperature. Hence, the delay between QCL sweeps is not constant, which poses a critical problem to step scan characterization. This problem can, too, be overcome by measuring each of the QCLs separately.

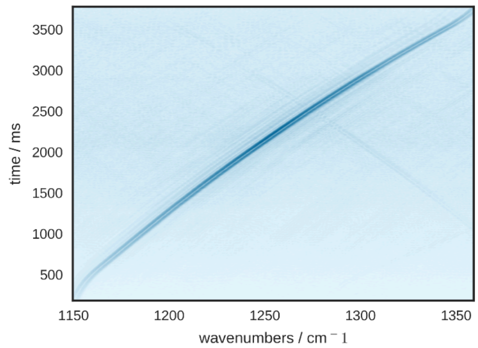
In order to make the conditions during the step scan measurement similar to those during a scan across all four lasers the step scan measurements included a small part of the spectral range of the preceding and the following laser as well as that of the laser itself. The positive edge of the Tuned signal output by the MIRcat was used as a reference for the start of a scan. Since the MIRcat outputs the same TTL signal on its Tuned output for each of the four chips, an additional circuit was needed that filtered out just the scan of the QCL that



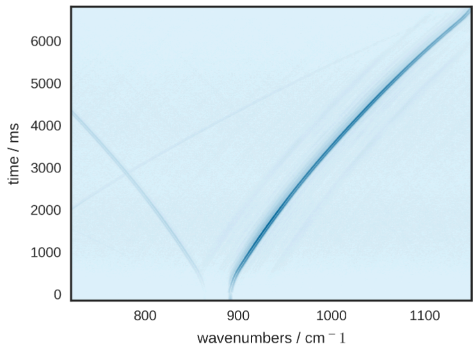
(a) QCL 1



(b) QCL 2



(c) QCL 3



(d) QCL 4

Figure 4.2.: Step scan measurements of the four QCL chips of the MIRcat source used in this work. The plotted values are the square roots of the intensity of the laser output. Darker colors denote higher output power. The square root has been plotted to make changes on the high and the low end of the intensity range visible next to each other.

was to be characterized. This was implemented using an Arduino Uno micro controller board. The Tuned signal of the EC-QCL was connected to a digital input of the board and a digital output of the board was connected to the step scan trigger input of the FTIR spectrometer. The code running on the micro controller is printed on page 167.

The collected step scan measurements are printed in figure 4.2. The resulting functions chosen to interpolate the relationship between scan time and wavelength were:

$$\begin{aligned}\tilde{\nu}_{\mathrm{QCL1}}(t) =& 2759\mathrm{cm}^{-1} + 187.3\mathrm{cm}^{-1}\mathrm{s}^{-1}t - 953.1\mathrm{cm}^{-1}\mathrm{s}^{-2}t^2 + 2813\mathrm{cm}^{-1}\mathrm{s}^{-3}t^3 \\ & - 3924\mathrm{cm}^{-1}\mathrm{s}^{-4}t^4 + 2921\mathrm{cm}^{-1}\mathrm{s}^{-5}t^5 - 1116\mathrm{cm}^{-1}\mathrm{s}^{-6}t^6 + 171\mathrm{cm}^{-1}\mathrm{s}^{-7}t^7\end{aligned} \tag{4.2}$$

$$\begin{aligned}\tilde{\nu}_{\mathrm{QCL2}}(t) =& 1364\mathrm{cm}^{-1} - 32.12\mathrm{cm}^{-1}\mathrm{s}^{-1}t + 157.5\mathrm{cm}^{-1}\mathrm{s}^{-2}t^2 - 130.8\mathrm{cm}^{-1}\mathrm{s}^{-3}t^3 \\ & + 60.26\mathrm{cm}^{-1}\mathrm{s}^{-4}t^4 - 15.24\mathrm{cm}^{-1}\mathrm{s}^{-5}t^5 + 1.992\mathrm{cm}^{-1}\mathrm{s}^{-6}t^6 - 0.105\mathrm{cm}^{-1}\mathrm{s}^{-7}t^7\end{aligned} \tag{4.3}$$

$$\begin{aligned}\tilde{\nu}_{\mathrm{QCL3}}(t) =& 11141\mathrm{cm}^{-1} + 33.04\mathrm{cm}^{-1}\mathrm{s}^{-1}t + 11.1\mathrm{cm}^{-1}\mathrm{s}^{-2}t^2 - 1.719\mathrm{cm}^{-1}\mathrm{s}^{-3}t^3 \\ & + 0.143\mathrm{cm}^{-1}\mathrm{s}^{-4}t^4\end{aligned} \tag{4.4}$$

$$\begin{aligned}\tilde{\nu}_{\mathrm{QCL4}}(t) =& 891.2\mathrm{cm}^{-1} - 2.467\mathrm{cm}^{-1}\mathrm{s}^{-1}t + 39.85\mathrm{cm}^{-1}\mathrm{s}^{-2}t^2 - 25.32\mathrm{cm}^{-1}\mathrm{s}^{-3}t^3 \\ & + 9.226\mathrm{cm}^{-1}\mathrm{s}^{-4}t^4 - 1.851\mathrm{cm}^{-1}\mathrm{s}^{-5}t^5 + 0.1914\mathrm{cm}^{-1}\mathrm{s}^{-6}t^6 \\ & - 0.007951\mathrm{cm}^{-1}\mathrm{s}^{-7}t^7\end{aligned} \tag{4.5}$$

4.2. Optomechanical Setup

The optomechanical part of the PTIR setup has to perform three tasks:

1. Setting the polarization of the light incident on the sample,
2. focusing the light into a tight spot on the sample and
3. allow for adjusting the position of the focal spot relative to the cantilever.

4.2.1. Polarization

For thin ($\approx \lambda$) samples on reflecting surfaces the polarization of the light relative to the plane of incidence strongly influences the interaction between light and sample, even if the optical properties of the sample are otherwise anisotropic [100].

Close to the surface the incident and reflected wave interfere causing either a field enhancement, when the field vectors of the incident and reflected wave align, or a field decrease, when the fields cancel each other out. This effect is strongest on metallic surfaces, where almost all the incident light is reflected. On metallic surfaces perpedendicular to the plane of incidence (s) polarized waves are phase shifted by close to 180° upon reflection, leading to destructive interference close to the surface. While the phase shift of parallel to the plane of incidence (p) polarized waves depends on the angle of incidence, the maximum field intensity close to the surface is reached when incident and reflected wave are 90° apart. This effect has been used in infrared reflection absorption spectroscopy (IRRAS) to perform surface selective IR spectroscopy (see for example [101]).

The light emitted by the lasers used for this work was already polarized 100:1 perpendicular to the base plate of the source. For the smaller single chip EC-QCL adjusting the polarization was possible by rotating the laser itself. While this would be inconvenient for adjusting the polarization during a measurement it can still be used to flip the polarization by 90° during the construction of the setup. For the MIRcat source even a rotation of the laser during setup time would have been impractical. Instead, polarization was adjusted using gold mirrors. Mirror based systems were preferred to the alternative of wave plates for polarization adjustment as they show no dispersion.

Since there is no coupling between p- and s-polarized light upon reflection on a plane surface, two mirrors can be used to change the polarization by 90°. The first mirror is placed at 45° to the beam propagation to reflect the beam onto a path perpendicular to the optical plane. The second mirror is placed at 45° angle to this perpendicular path and rotated so that it reflects the beam onto a path that is parallel to the original optical plane and 90° to the original beam path. Using such a design the polarization angle was adjusted in the PTIR setup as necessary. A possible future add-on to the PTIR setups would be a reflective, adjustable polarization rotator using three mirrors, that keeps input and output beams collinear [102], with which the polarization could be set to arbitrary angles.

4.2.2. Control of Focal Point Position

As the PTIR signal has a linear dependence on the light intensity below the cantilever tip the focal point of the EC-QCL beam has to be moved to the cantilever tip before each measurement. Hence, a mechanism for adjusting either the position of the focal point or the cantilever in three dimensions is necessary. As the Keysight 5400 AFM does not offer the possibility to position

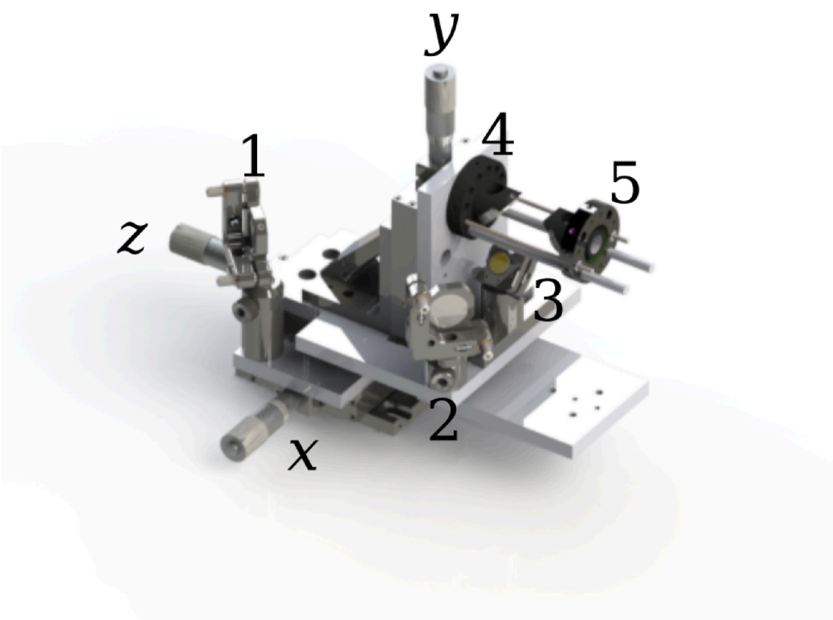


Figure 4.3.: Rendering of the three dimensional stage for adjusting the position of the focal spot. Numbers refer to the mirror numbers mentioned in the text, letters denote the knobs used to adjust the focal spot in the corresponding direction.

the cantilever (except for the scanning motion needed for imaging) and moving the whole AFM in three dimensions proved to introduce significant mechanical noise into the setup, a system to move the focal spot was designed instead. The system consisted of three mechanical stages (PT1-M, Thorlabs), custom made as well as commercially available mechanical parts and plane gold mirrors.

The main design idea of the system was to keep movements in each of the three spatial dimensions independent of each other. This was achieved by designing the system in such a way that two successive mirrors in the system can only move relative to each other along the beam path (a rendering of the design is depicted in figure 4.3).

The laser beam enters the stage along the x axis and is reflected at a right angle along the z axis (mirror 1). The beam is reflected by 90° along the z axis (mirror 2) and then flipped upwards to be parallel to the y axis (mirror 3). Two mirrors mounted in a cage system redirect the beam first along the z axis (mirror 4) and then downwards (mirror 5) towards the focusing optics.

When the focal spot is moved along the x axis of the system all mirrors move together, mirror 1 moves in parallel to the direction of the incident beam. When the focal spot is adjusted in z direction, mirror 1 remains fixed while mirror 2 to 5 are moved along z. The relative movement of mirror 1 and 2 is along z, which is also the direction of the beam path between them. Finally, for moving the focal spot mirror in y mirrors 1 to 3 remain fixed and mirror 4 and 5 as well as the focusing optics are lifted upwards. Again, the relative movement of mirror 3 and 4 is along the beam path connecting them.

Technical drawings of the custom made parts used to mount the mirrors on the translation stages are included in the appendix, on page 235ff.

4.3. Focusing

The job of the focusing optic in a PTIR setup is to concentrate the laser beam into a tight spot on the sample. Since the PTIR amplitude is proportional to the light intensity a tighter spot will result in a higher signal and a more sensitive measurement. For top illumination designs the geometry of the AFM can also add restrictions on the design of the focusing optic: the laser beam has to be directed to pass by all parts of the AFM except for the cantilever tip as other parts might either obscure the laser beam or - even worse - absorb the beam and thereby introduce an additional background signal into the measurement.

In this work two different options for focusing the laser beam were evaluated:

1. a fixed parabolic mirror in combination with a movable mirror to allow for adjustement of the angle of incidence without changing the position of the focal spot

2. a rotatable parabolic mirror with a shorter focal length

4.3.1. Focusing Using a Fixed Parabolic Mirror

In parabolic mirrors all incoming rays that are parallel to the rotation axis of the mirror are reflected towards the focal spot of the parabola. In this work this fact has been used to develop a focusing optic that allows adjusting the angle of incidence while keeping the position of the focal spot fixed. This is achieved in a design as sketched in figure 4.5. A plane mirror is used to shift a collimated laser beam towards or away from the rotation axis of the parabola. This changes the point of incidence of the laser beam on the parabolic mirror. Since the beam reflected off the parabolic mirror still has to go through the focal spot, its angle of incidence onto the sample is changed.

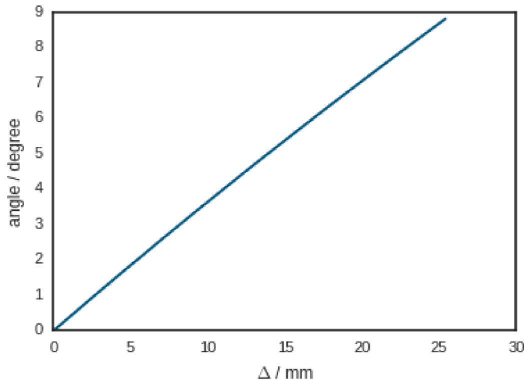


Figure 4.4.: Range of angles of incidence covered by the parabolic mirror.

For a vertical ray the point of incidence on the parabolic mirror is $\left(x|x^2/4f\right)$ relative to the origin and $\left(x|x^2/4f - f\right)$ relative to the focal spot. The angle α between the focal plane and the laser beam is $\tan^{-1}\left(x/4f - f/x\right)$.

A 90° off axis parabolic mirror is usually a circular segment taken from a paraboloid with a center at $2f$.Therefore, to make the positioning of the laser beam relative to the center of the mirror element, x can be replaced with $2f + \Delta$, where Δ is the distance the beam has been shifted from the center point of the mirror. The resulting formula for the angle is $\tan^{-1}\left(2f+\Delta/4f - f/2f+\Delta\right)$.

In this work a 90° off-axis parabolic mirror with a diameter of 50.8 mm and a focal length of 76.2 mm (MPD508762-90-M01, Thorlabs) was used for focusing. The beam was shifted via a plane gold mirror mounted on a cage system based linear stage (Edmund Optics) with a micrometer screw. The angle of incidence of the mirror could be set between 0° and 9° (see fig. 4.4).

In its original configuration the distance between the sample plane and the nose cone of the Keysight 5400 AFM was less than 2 mm. The angle between the most extreme ray that could still pass to the cantilever and the sample plane enclosed an angle less than 14°, restricting the angle of incidence to >76°. The low angle of incidence in this system lead to a broadening of the laser spot along the beam direction.

In order to reduce the spot size different methods of reducing the beam shape were evaluated in a physical optics propagation software (ZEMAX). A first idea was to use a 3 to 1 beam reducer to reduce the size of the opening angle of the cone of light incident on the sample. However, this system showed an increase in the size of the beam diameter over the bare system. The reason for this is most likely the link between diameter at the beam waist and the divergence of the

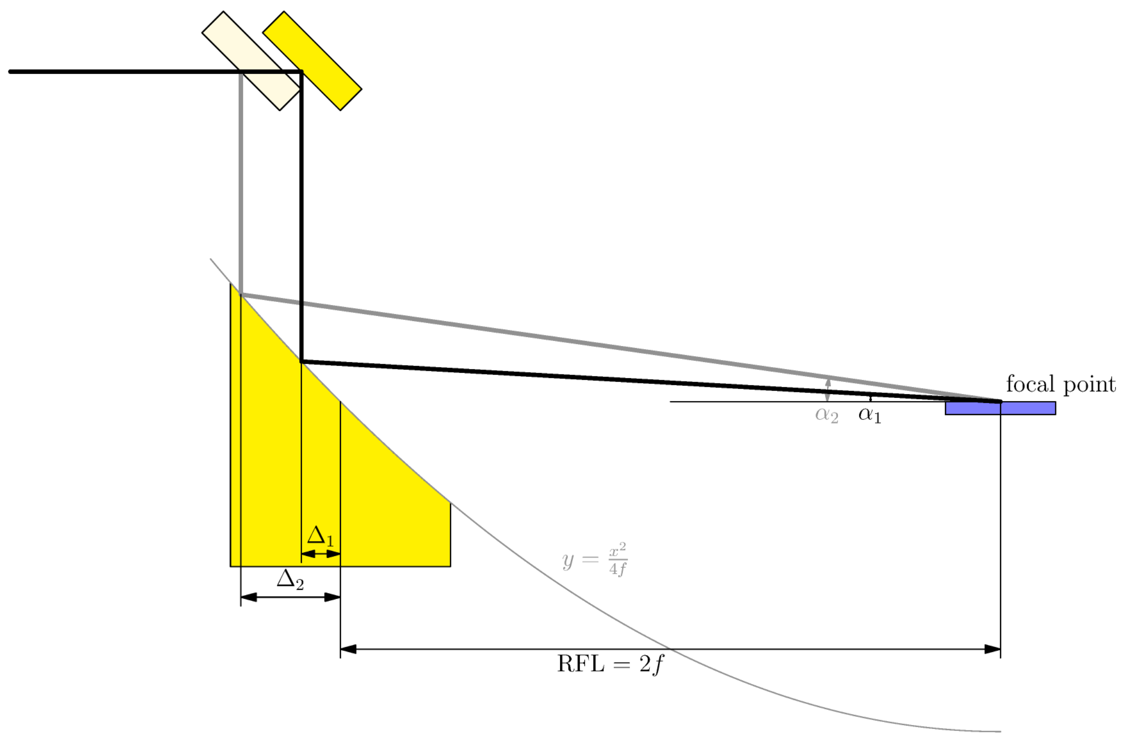


Figure 4.5.: Sketch of a design for changing the angle of incidence by shifting a plane mirror along the beam path. When the cantilever is placed in the focal spot of the parabolic mirror, changing the angle of incidence does not change the position of the focal spot in the sample plane.

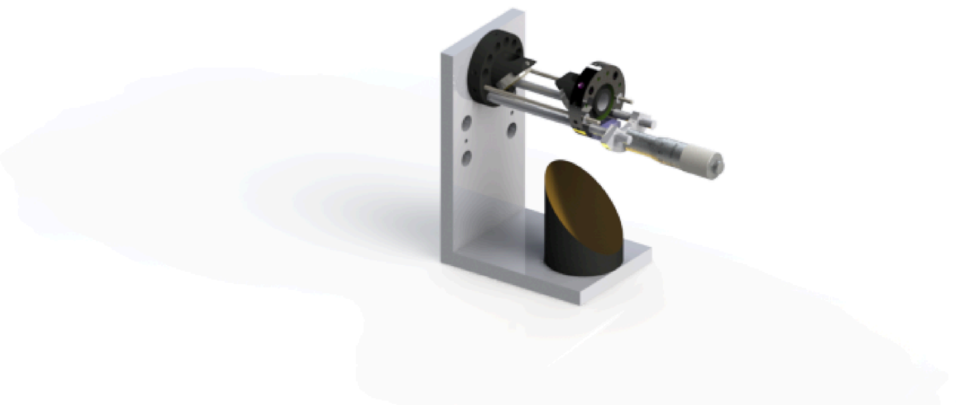


Figure 4.6.: Rendering of a design for changing the angle of incidence by shifting a plane mirror along the beam path.

beam found in Gaussian beams.

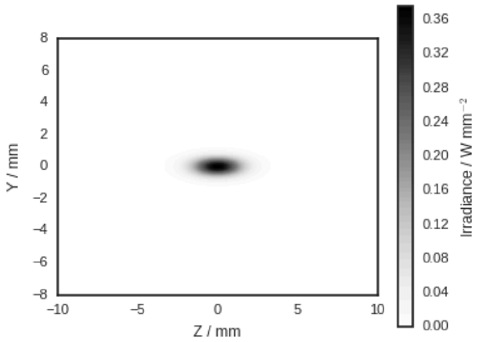
Refractive systems were also quickly discarded due to their chromatic aberration and absorption across the wavelength range of the lasers (e.g. CaF_2) or their high losses caused by reflections at the interfaces (e.g. ZnSe or Ge).

The remaining evaluated systems were reflectiv beam expanders built from off-the rack 25.4 mm diameter off-axis parabolic gold mirrors of reflected focal lengths (RFLs) of f = 25.4 mm, 50.8 mm and 76.2 mm. Leaving possible combinations of 1 to 2, 2 to 3 and 1 to 6 expansion. For a 1 to 6 expansion the beam width on the parabolic mirror turned out to be too wide for the remaining part of the optical setup, hence this design was also discarded.

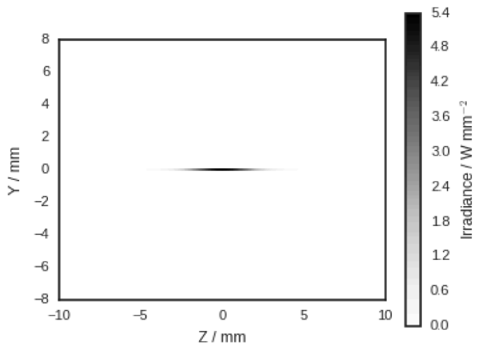
The systems were compared by the fraction of power emitted by the laser that was concentrated in a circle of 220 µm diameter on the sample plane. Before comparison, the systems were optimized using the ZEMAX optimization function to maximize the power inside the circle by adjusting the position of the sample plane and the distance of the beam expander mirrors.

The simulations show that the 1 to 3 beam expander leads to the highest fraction of the original power around the cantilever and the smallest spot size (see table 4.1).

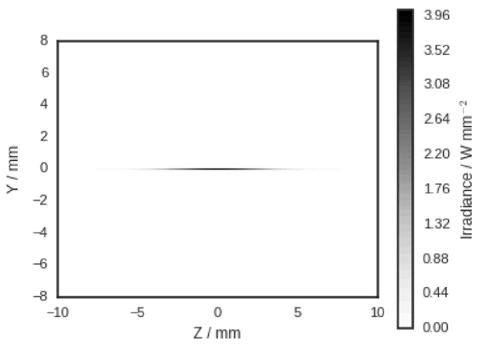
The actual intensity of the laser spot in the setup was measured using a Pyrocam III beam profiler (Spiricon). Since the sensor of the beam profiler was receded too far into its housing to be able to directly measure the beam intensity



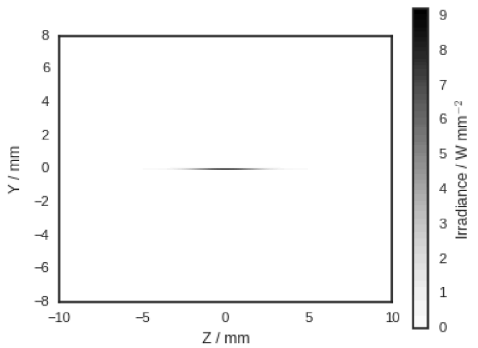
(a) Without beam expander.



(b) 1 to 2 beam expander.



(c) 2 to 3 beam expander.



(d) 1 to 3 beam expander.

Figure 4.7.: Simulated intensity distributions in the sample plane. Note that the color bars differ between plots.

Table 4.1.: Simulation results for different beam expanders.

configuration	fraction of power	width x / mm	width z / mm
no expander	0.014	0.560	3.026
2 to 3	0.137	0.196	2.712
1 to 2	0.177	0.163	2.844
1 to 3	0.265	0.115	2.612

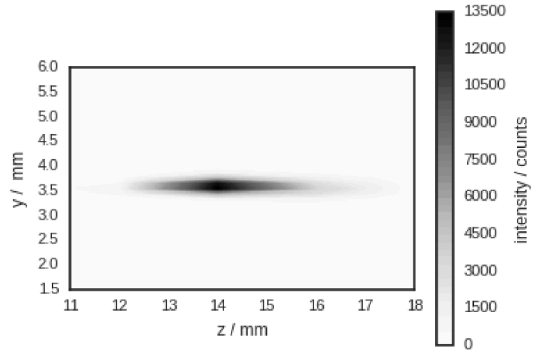


Figure 4.8.: Beam profile in the sample plane

in the sample plane, a series of vertical beam profiles was measured instead at different positions along the beam path. These profiles were assembled into a three dimensional recording of the beam intensity around the focal spot through which arbitrary sections could be calculated. The best focal spot achieved in these measurements is depicted in figure 4.8.

4.3.2. Focusing Using a Rotatable Mirror

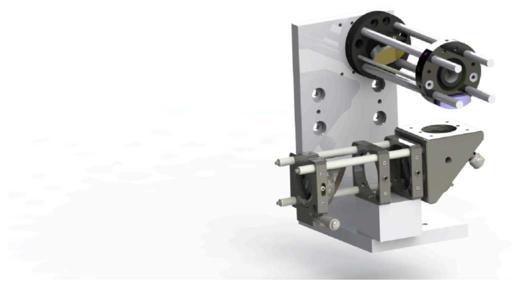


Figure 4.9.: 3D rendering of a focusing optic using a rotatable mirror.

Due to the broad spot size at the low angle of incidence an increase of the angle of incidence was considered necessary. This made it necessary to increase the distance between the sample plane and the AFM cantilever holder. A 1 mm aluminum spacer was inserted between the cantilever and the holder. In order to cover a broader range of angles of incidences than possible with the fixed parabolic mirror the focusing optics were redesigned to use an off axis parabolic mirror mounted in a cage system on a rotation stage (see fig. 4.9).

Focusing was performed with 90° off-axis parabolic mirror with a diameter of 25.4 mm and a focal length of 50.8 mm (MPD149-M01, Thorlabs). The mirror was mounted on a SM1MP holder. The holder was placed in a CRM10/M cage mounted rotation stage. The incoming vertical beam was reflected horizontally towards the parabolic mirror using a plane gold mirror in a KCBE1/M 45° mirror holder. The whole cage assembly was mounted to the stage using two CP03/M cage plates and a aluminum spacer. (All parts: Thorlabs). Technical drawings depicting the changes made to the custom made parts of the stage can be found in the appendix on 241ff.

The advantage of a rotatable mirror is that any angle of incidence can be set by rotating the mirror to the corresponding angle without restrictions due to the dimensions of the mirror. The drawback is however, that after rotating the mirror the focal spot has to be readjusted back to the position of the cantilever. It was determined that a setting of 65° angle of incidence still allowed the beam to pass below the cantilever holder.

Since the resolution of the Pyrocam III was not sufficient to resolve the focal spot, instead the amplitude signal of the cantilever induced by the laser beam was used for adjustment. The cantilever was approached onto a sample of known properties (e.g. a polymer film) and the laser repetition rate was adjusted as described in section 5.

Instead of adjusting for the highest intensity the system was adjusted to achieve a minimum spot size in the sample plane. While the maximum amplitude can slowly drift during the adjustment procedure (e.g. due to changing water vapor content in the air, or changes in the cantilever or sample propterties) the spot size should remain constant.

Finding the minimum spot size is a slightly involved process during which the focal spot is moved in steps of half a millimeter in the y direction and at each step the location of the x and z axis at which the measured amplitude reaches its maximum and the points at which the amplitude has sunk to half of the maximum is noted. From the noted positions the FWHM of the beam is then calculated. The y position at which the spot has its lowest extension marks the focal plane of the laser beam. As with the fixed mirror focusing setup, the beam expander can be readjusted to decrease the dimensions of the focal spot. The smallest beam dimensions that were found at 65° angle of incidence were FWHM: 200 µm × 55 µm.

To ascertain that the focal spots of all four EC-QCLs are close to each other the location of the cantilevers were tested by placing the cantilever on a 200 nm

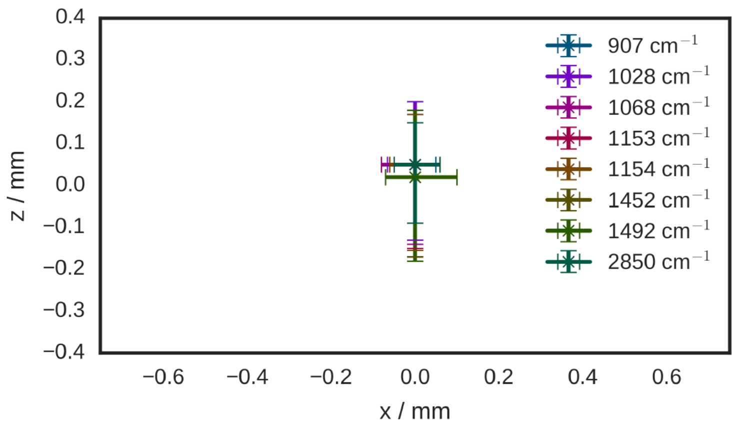


Figure 4.10.: Position of maximum intensity (cross) and the points at which the intensity was at the noise level (bars) for several absorption bands of polystyrene.

thick polystyrene film sample and then locating the stage position at which the maximum intensity was measured. These measurements were performed for selected polystyrene absorption bands in the tuning ranges of four chips of the MIRcat (see fig. 4.10). While the focal positions were not completely identical, the spots were nevertheless close in comparison to the spot size.

4.4. Assembly and Alignment

Assembly and alignment of the setup was done step wise, starting at the laser light source.

A red laser pointer was combined with the IR beam for easier alignment. For combining these two beams either a wedged CaF_2 window (25.4 mm diameter, 1.5° wedged, Crystran, UK) or glass covered with indium tin oxide (ITO) were used. The CaF_2 windows was used together with the single chip EC-QCL only, as CaF_2 absorbs in the lower wavenumber part of spectrum of the MIRcat emission. ITO has broad band reflection in the mid-IR [103] and is transparent in visible range.

After the beam combiner the EC-QCL beam was aligned with the laser pointer by checking their positions at several distances from the combiner with an IR detector card. From this step onward, further alignment was performed using the visible laser only. The stage for adjusting the position of the focal spot was assembled in place. Each mirror in the stage system was adjusted to reflect the laser beam perpendicular. Horizontal beam paths were checked by using the rows of the optical bread board as reference, vertical ones were checked using 0.5 m marking gauge. Finally, the AFM was placed so that the focal spot was approximately on the cantilever. All further adjustments were performed by moving the stage.

The whole setup was placed in a transparent housing that was flushed with dry air to remove water vapor. Initial test showed, that water vapor absorption could be reduced below the noise level of the laser within 10 min after closing the housing.

5. Signal Demodulation

5.1. Resonant Excitation - Lock In Amplifiers

For resonant excitation Lu, Jin, and Belkin [38] use a lock-in amplifier to detect the amplitude of the cantilever oscillations. The setup of Belkin and Lu uses the EC-QCL pulse trigger out signal as reference for the lock-in amplifier, their A-B (vertical deflection) signal is fed to an external lock-in amplifier (SR844, Stanford Research Systems, CA) and the filtered amplitude is then fed back to the AFM controller through one of its external inputs. Using the pulse rate of the EC-QCL as reference for the lock-in amplifier ensures that the lock-in amplifier and the EC-QCL pulses are synchronized.

The lock-in amplifier demodulation scheme used for this work was similar to that used by Lu and Belkin but differed in some important details. Instead of using an external lock-in amplifier one of the internal lock-in amplifiers of the MAC III box of the Keysight 5400 AFM was used. This lock-in amplifier can not be referenced to an external frequency, instead the lock-in amplifier reference frequency was selected in the PicoView software. The reference sine wave is output at the lock-in amplifier frequency on the drive out connectors of the MAC III controller. The lock-in amplifier amplitude and phase can be recorded as images or time traces in the PicoView software and ouput at the BNC 1 or 2 connectors of the MAC III controller box.

To tie together the EC-QCL repetition rate and the lock-in amplifier reference, the lock-in amplifier reference sine was passed through a Schmitt trigger that turned the sine-wave into a rectangular wave. This signal was then fed as pulse trigger to the EC-QCL. For an overview of the setup see figure 5.1.

When the laser is set to “external trigger” mode, pulses are emitted at each positive edge of the laser pulse trigger input, the pulse duration can however still be set via the controller of the laser.

Like the center frequency of the lock-in amplifier its cutoff frequency and a preamplifier can be selected in the softare. The preamplifier can be set in steps of powers of two from 1 tow 128 and the cutoff frequency for the low pass filter can be set to values of 80 Hz, 100 Hz, 500 Hz, 1000 Hz and 2000 Hz.

In the first implementation of this detection scheme the raw deflection signal

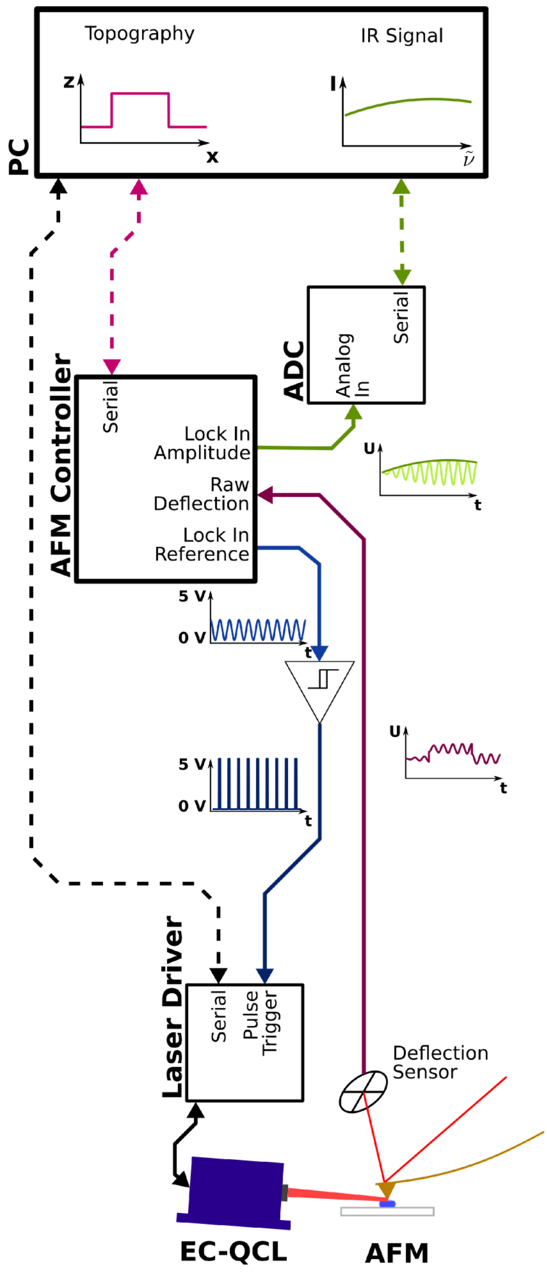


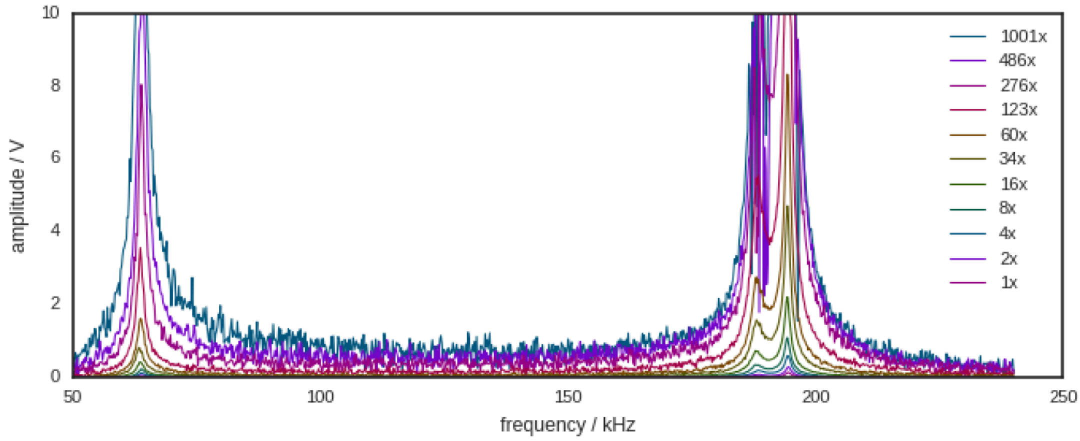
Figure 5.1.: Flow diagram showing the signal in the lock-in amplifier based setup. Reproduced with permission from [104]. Copyright 2015 American Chemical Society.

of the cantilever was used as input to the lock-in amplifier. However, due to the low level of the PTIR signal the raw deflection signal had to be amplified using the internal amplifier of the MAC III box. As the amplifier amplified low frequency components as well as those frequencies corresponding to the PTIR signal higher amplification decreased the dynamic range signal. This was especially troubling as for weak signals low amplification the signal was close to the resolution of the output DAC of the MAC III controller.

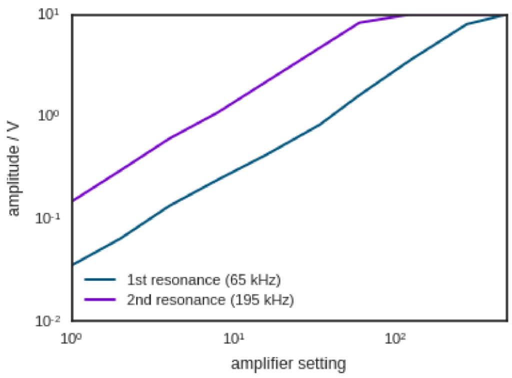
A preamplifier circuit was built to remove the low frequency part of the raw deflection signal. The circuit removes oscillations below 50 kHz and above 350 kHz. In addition to a band-pass filter this circuit also allows to select the amplification of the last stage of the amplifier via 10 dual inline package (DIP) switches. Starting at switch 1 the amplifications are 2x, 4x, 8x, 16x, 34x 60x, 123x, 276x, 486x and 1001x. If no switch is turned on, the amplification factor is 1x.

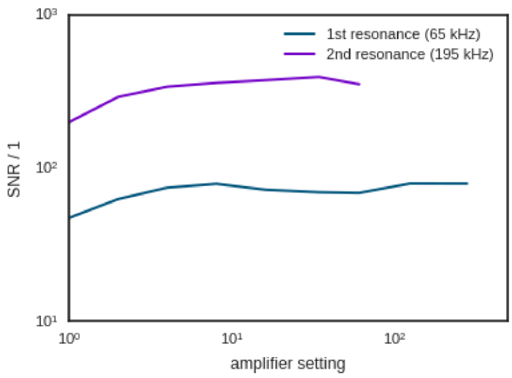
With higher amplification the signal as well as the noise are amplified (see figs. 5.2a and 5.2b), hence the amplifier does not give a large improvement in the SNR (see fig. 5.2c). The advantage of the pre-amplifier is rather that the low frequency components are removed from the signal, allowing a broader dynamic range for the lock-in amplifier, and that the signal range can be adapted to the input range of the analog digital converter (ADC).

Using the setup described above, PTIR images could be acquired by recording the lock-in amplitude setup together with the topography setup. To collect spectra, additional hardware was needed. The lock-in amplitude signal was output at one of the BNC connectors of the MAC III box. The analog signal was then digitized again by a NI9239 (National Instruments, US) 50 kSs^{-1} ADC. Data acquisition was triggered by Tuned signal of the laser source, which was digitized on one of the ports of a NI9401 digital I/O module. The modules were controlled via a LabView (National Instruments, US) program. The LabView program adjusted the number of datapoints to be acquired according to a preset scan time for each laser. Using sample rate and the time-wavenumber curves of the lasers, it then converted the recorded intensities to wavenumber dependent single channel spectra, which were displayed and saved to disk. The GUI is depicted in figure 5.3.



(a) Resonance spectra for amplifier settings from 0 to 10.





(b) Peak amplitude of first and seconds resonance.

(c) SNR for first and second resonance.

Figure 5.2.: Measurement of amplitude and SNR for the first two resonance of a polystyrene sample excited at $1430\,\mathrm{cm}^{-1}$.

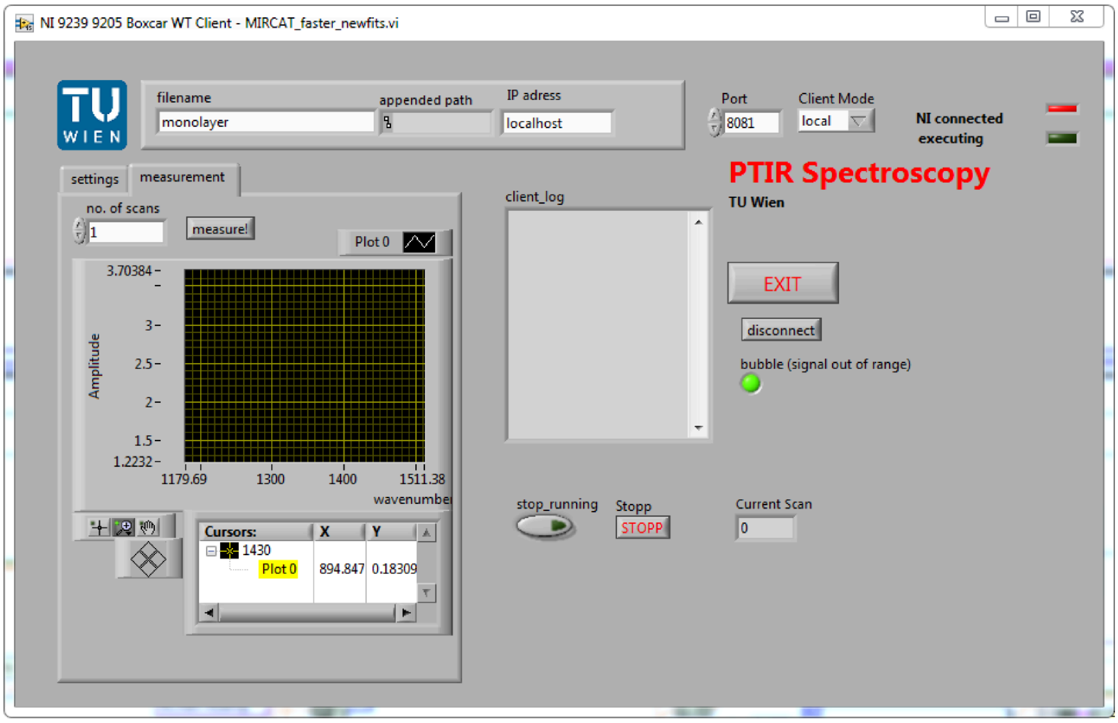


Figure 5.3.: GUI of the LabView program for acquiring spectra in a PTIR setup.

5.2. Implementation of a Resonance Tracking Method

5.3. Selection of a Resonance Tracking Method

An overview over resonance tracking methods that have been used with AFMs has been given in section 3.4. For selection of the optimum method for resonance tracking in PTIR several constraints in the system had to be considered. The most important constraints were

1. amplitude detection:

   The resonance frequency itself is of secondary importance to resonance tracking in PTIR. It is more important to measure a value that is proportional to the local IR absorption than to measure the exact amplitude or resonance frequency.

2. speed:

   The method has to be fast enough to be usable at common contact mode AFM scan rates. This means 100 or more updates of amplitude value per second.

3. compatibility with laser sources:

   While EC-QCLs that allow modulation of the output intensity are commercially available, pulsed EC-QCLs are currently much more common and offer broader wavelength coverage. Hence, a resonance tracking method has to be able to work with pulse trains whose repetition rate is not much higher than the contact resonance that is targeted.

4. ability to track jumps in the resonance frequency of several tens of kilohertz at frequencies of several hundreds of kilohertz:

   The force constant of the cantilever is indirectly proportional to the RE-PTIR sensitivity (see section 3.3.2, hence lower k cantilevers are used preferentially for RE-PTIR. However, lower force constants also imply a higher sensitivity of the contact resonance frequency to the mechanical properties of the sample. Jumps in resonance frequency of 10 kHz are commonly observed.

All methods described in section 3.4 fulfill condition 1 and 2, however broad band excitation via a single pulse will not be discussed further as it is essentially the original ring down method of PTIR. In the following the remaining methods will be evaluated for their compatibility with available lasers and their ability to track larger resonance jumps.

5.3.1. Compatibility with Selected EC-QCL Sources

The light source used in this work has a maximum repetition rate of 350 kHz and does not allow modulation of the peak pulse power in real time. These are, however, not restrictions that are valid for all current EC-QCL and even less so for future developments.

The band excitation method necessitates applying a complicated waveform to the cantilever. Without control over the output power of the laser the waveform can be approximated using pulse width modulation (PWM). This technique is commonly used as a form of digital analog conversion in electronics. In PWM a two level output (i.e. on/off) can be used to generate an analog waveform, as long as the maximum frequency of the signal is less than half that of the PWM pulse rate [105].

The spectrum of the raw PWM output contains strong components corresponding to the laser pulse rate and reflections of the wanted waveform around

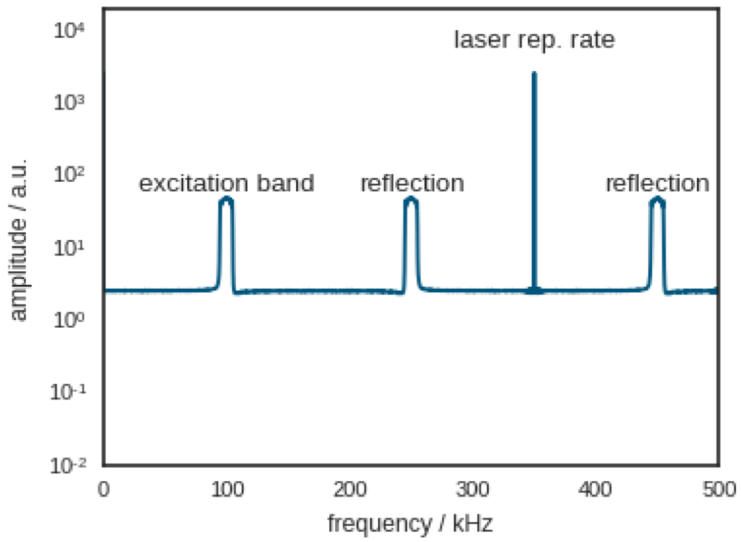


Figure 5.4.: Spectrum of a PWM based band-excitation using a Kaiser window ($\beta = 1.0$) for the range of 85 kHz to 115 kHz. The laser repetition rate peak and the reflection repeat at every full multiple of the laser repetition of 350 kHz: 700 kHz, 1050 kHz and 1400 kHz, ...

the laser pulse rate (see fig. 5.4). In electronic system these are removed by an analog low pass filter while this is not possible in the case of optical excitation.

PWM makes it necessary to accurately control the pulse width of the laser. The time resolution at which the pulse length can be set together with the maximum pulse length determines the intensity resolution at which the curve can be approximated. For the laser used in this work running at 350 kHz the maximum pulse length that all four chips can deliver is 140 ns (5 % duty cycle). A 1 ns time resolution would therefore grant an intensity resolution of 140. [1]

When using band excitation techniques with driver with a finite range the height of the excited band and its width become linked due to the fact that broad features in the frequency domain result in sharp features in the time domain. Hence, if broader frequency ranges are to be excited the whole waveform has to be scaled downwards to still fit inside the range of the driver.

As a rough estimate for the improvement to be expected with band excitation over ring-down techniques figure 5.5 shows the amplitude of the output signal

[1] However, for an EC-QCL that can run at 50 % duty cycle at the same repetition rate 1 ns time resolution would grant a ten times higher intensity resolution. Such lasers are currently commercially available albeit with a lower spectral coverage than the laser used in this work.

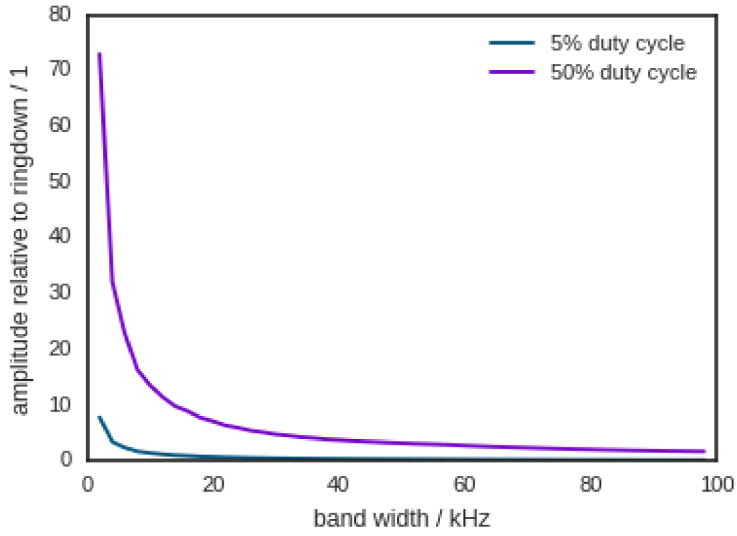


Figure 5.5.: Amplitude at 100 kHz for band excitation with 350 kHz laser repetition rate relative to the amplitude achieved with a broad band ring down excitation with a pulse width of 1 µs.

at 100 kHz for system parameters as delivered by the laser in this work, i.e. 5 % duty cycle at a maximum repetition rate of 350 kHz. Due to the low repetition rate needed for ring down measurement longer pulse lengths can be used in this mode (here 1 µs). The pulse intensity was in both cases arbitrarily set to one. Errors due to insufficient time resolution were ignored, but all band excitation wave-forms were scaled to use the full range that can be covered in the PWM method. In fig. 5.5 the reciprocal relationship between the pulse width and the amplitude is clearly visible. For the laser used in this work band excitation performs worse than ring down measurements at band width broader than 15 kHz. However, for systems with higher duty cycle (e.g. 50 % as in fig. 5.5), the advantage of band excitation remains for higher band widths.

To sum up, band excitation type measurements for EC-QCL driven systems need complex, high-speed electronics while limiting the maximum frequency to half of the maximum laser repetition rate and introducing several strong, unwanted frequencies into the system. When maximum duty cycles of the laser used in this work are considered the method performs better than pure ring down measurements for narrow band widths, only. For this reason the band excitation method was not pursued any further in this work.

To make DART compatible with pulsed excitation laser pulses have to be fired at the frequencies that are to be excited. Since two pulse trains have to be fired at the same time the maximum pulse frequency of each of them has to be less than or equal to 175 kHz. The one remaining challenge in using DART with a pulsed source is to resolve pulse "collision", i.e. when pulses of the two pulse trains would have to be emitted in close vicinity to each other. Possible solutions to this problem are to use a simplified PWM scheme by only using half the maximum pulse length for each pulse train and doubling the pulse length whenever the pulses coincide or by omitting one of the coinciding pulses.

PLL and sweep based systems can be used with a pulsed laser. The only change needed is to replace the sinusoidal excitation with a pulsed one.

5.3.2. Ability to Track Resonance Jumps

Band excitation techniques are only sensitive to the position of the resonance when it lies within the excited band. Since the band width is indirectly proportional to the height of the excited band the ability for tracking across a broad range has to be balanced against the increase in amplitude.

In DART the user also has to strike a compromise between the ability to track large changes in the resonance frequency and the sensitivity [89]: Sensitivity considerations alone would lead to excitation frequencies at $f_{1,2} = f_0\left(1 \pm \frac{1}{2\sqrt{2}Q}\right)$ as the change in the amplitude difference is highest at this spacing as long as the resonance is between these frequencies. However, should the resonance jump out-side the range between f_1 and f_2 the amplitude difference quickly goes towards zero. When f_1 and f_2 are more broadly spaced they can track the resonance across a larger range. The exact range at which tracking is no longer possible depends on the SNR of the signal, as well.

Similarly, PLL can have trouble tracking across larger jumps in noisy systems. When the phase is determined as the arc-tangent of the ratio of X and Y component of a lock in amplifier the noise will contribute significantly to the phase signal once the magnitude of X and Y are low which is the case further from the resonance.

In sweep based techniques the tracking range is linked to the swept range. A feed-back mechanism can be implemented to move the sweep range towards the maximum detected signal [96, 94].

5.3.3. Selected Tracking Method

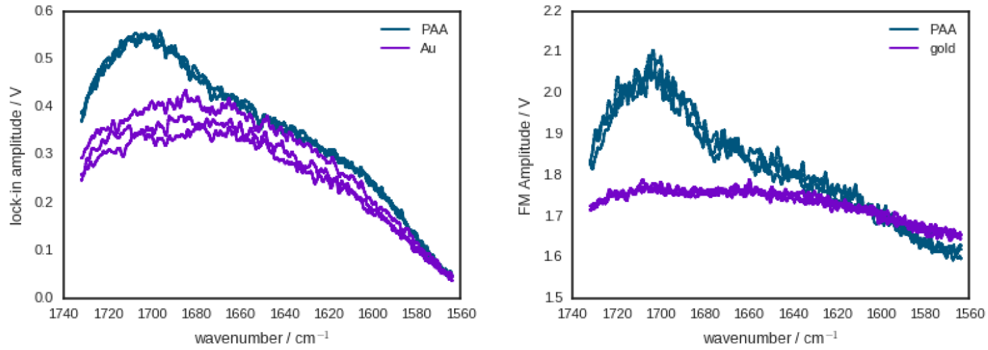
From the methods discussed above two were selected for further examination: PLL based resonance tracking was chosen as it was already available in the frequency modulation atomic force microscopy (FM-AFM) module for the MAC III controller box and needed only slight adjustments to be usable with the existing RE-PTIR setup. Sweep based resonance tracking was chosen as it would give access to at least the first two contact resonance of the cantilever with the laser sources used in this work. This would not be possible with the other remaining technique, DART.

Use of a Keysight frequency modulation atomic force microscopy (FM-AFM) controller for PTIR

Setup Resonance tracking using a PLL was implemented using the optional FM-AFM module of the Keysight MAC III controller. The FM-AFM module was originally designed for using the tapping piezo for excitation, therefore, some steps taken in the Keysight manual for setting up an FM-AFM measurement had to modified. The add-on provides additional outputs to the FM-AFM measurement: FM Amplitude and FM Frequency. The FM Amplitude signal was digitized together with the Tuned signal of the EC-QCL using the setup described in section 5.1 to acquire spectra. Alternatively it was directly recorded in the AFM controller together with the cantilever position for imaging.

Setting up an FM-AFM measurement is a multi-step process, wherein first a resonance frequency of the system has to be determined using the manual sweep mode of the lock-in amplifier. Gain and bandwidth of the internal lock-in amplifier were set to 8x and 80 Hz to ensure a smooth signal. This resonance frequency is then subtracted from the internal frequency (4.5 MHz) of the FM-AFM module to calculate the working frequency. The FM-AFM adds an additional reference signal output to the MAC III controller that is connected to the laser via a Schmitt trigger (as in the lock-in amplifier measurements). Once the reference signal is connected, the driver phase is adjusted until the FM Amplitude signal reaches a maximum.

Evaluations The FM-AFM controller was tested using gold slides which were spin coated with poly-acrylic acid (PAA). For these measurements the single EC-QCL system described in section 4.1.3 was used. The laser was tuned to the first contact resonances of a ContGB-G cantilever ($\approx$90 kHz) using the lock-in amplifier of the MAC III box. Then the measurement parameters were set as



(a) Lock-in signal (b) FM-AFM signal

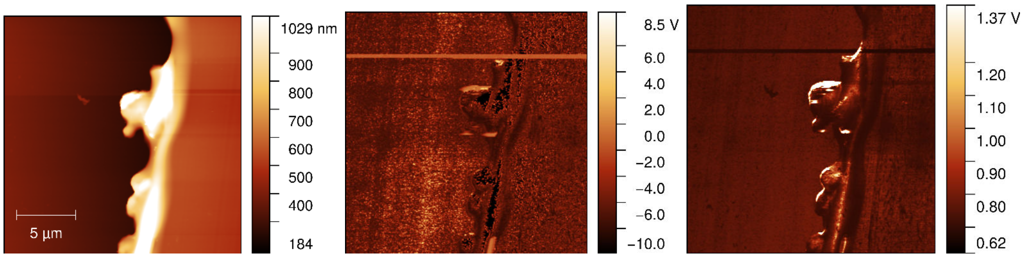
Figure 5.6.: EC-QCL sweeps performed on the substrate (gold) and a 300 nm film of PAA

described as above.

As a first test of the FM-AFM module, spectra of a PAA film of a thickness of 300 nm were acquired. Since the cantilever is not moved while a spectrum is recorded the contact resonance should only slightly shift, hence only the ability to record the amplitude of the cantilever is tested.

Figure 5.6 shows a comparision of background and sample spectra of PAA on gold taken using the built in lock-in amplifier (a) and the FM-AFM module (b). Both measurements look similar in shape. However, the FM-AFM spectrum has a constant offset off the amplitude, which should be subtracted from the single channel measurements before the PTIR signal is calculated. The background single channel of the lock-in amplifier measurement shows some drift in the amplitude value, suggesting a slight shift of the contact resonance. A similar effect - but smaller - is seen in the FM-AFM amplitude signal. In general, the noise in the FM-AFM signal is higher than in the lock-in amplifier signal single channel.

To test the ability of the FM-AFM module to track shift in the resonance frequency the edge between a PAA film and gold was imaged with the EC-QCL wavelength set to $1725\,\mathrm{cm}^{-1}$. Judging from the foreground and background spectra (see fig. 5.6b) a difference in amplitude of about 0.25 V is to be expected. However, no such difference is seen in the FM Amplitude image (see fig. 5.7c). Both gold and PAA have similar, high noise levels and average values. The reason for this difference could be problems in tracking the resonance changes. The FM-AFM seems to cross its whole dynamic range (±10 V) to adjust for the difference in the resonance frequency (see fig. 5.7b).



(a) Topography (b) FM Frequency channel (c) FM Amplitude channel

Figure 5.7.: Outputs of an FM-AFM measurement of a 300 nm high poly acrylic acid film on gold. While recording the horizontal brown stripe in the upper half of b and c the EC-QCL was turned off.

Further experiments performed on a film of polystyrene on gold yielded similar results. Therefore, it was decided to discard the idea of using the FM-AFM module in favor a different resonance tracking scheme.

5.4. PTIR Controller Design

In planning the RE-PTIR controller for sweep based measurements the following properties were targeted:

- Ability to trigger laser pulses at 5 V TTL compatible level
- Generation of laser trigger pulses at repetition rates at least up to 350 kHz
- Sweep across user defined repetition rate ranges
- On-board data evaluation (to remove lag due to communication with PC)
- Analog output of results in the ±10 V range accepted by the AFM controller used to locate the RE-PTIR signal on the sample
- user selected sweep ranges and sweep rates that can be changed in real time
- fast evaluation that allows acquisition of RE-PTIR spectra at the fastest wavelength sweep rate of the EC-QCL

In addition, since no RE-PTIR controller has been published in literature, at the time of writing, the controller was designed with an eye towards easy replication by others. This entails using standard electronic components and use of well documented prototype boards for components that are harder to prototype. It also entails using freely available software for programming. A further consideration in the development of the controller was a mostly digital approach to allow fast reconfiguration during experimental development and therefore having the option of comparing different evaluation methods.

5.4.1. Controller Components

As main controller Spartan 9LX (Xilinx), a small field programmable gate array (FPGA) was chosen. An FPGA is electronic component that can be reprogrammed to different types of logical circuits. The advantage of this type of element to microprocessors is their ability to work highly parallel and perform multiple operations at every clock cycle.

The FPGA was used in the form of a Mojo Board (`https://embeddedmicro.com/`), that includes the FPGA as well as all peripherals needed for driving and programming it.

For digitization of the AFM deflection signal an AD7760 (Analog Devices) Σ-Δ-ADC was chosen. This component has 24 bit resolution at a rate of $5\,\mathrm{MS\,s^{-1}}$. The chip can be digitally reconfigured to down sample and filter the data as needed. The AD7760 was used in the form of a prototype board (AD7760-EVAL), that provides a reference voltage and quiet supplies for the ADC. It is connected to the FPGA via a level-shifter board that is described below. To control the ADC and to read its digitized values a data bus consisting of 16 data lines and five control lines is used. The control lines are $\overline{\mathrm{DRDY}}$[2], which is low whenever a new sample is ready to be read, $\overline{\mathrm{RD}}\mathrm{WR}$ which is held high by the controller for writing data to the ADC registers or low for reading data and finally $\overline{\mathrm{CS}}$ that is brought low to actually perform a read or write. The two additional control signals are $\overline{\mathrm{SYNC}}$, which can be used to reset the digitization and to synchronize several ADCs and $\overline{\mathrm{RESET}}$, which resets all configurations.

A DAC 8555 (Texas Instruments) 16 bit quad-DAC is used. The component can update its output at 100 kHz, new values are set via a serial peripheral interface (SPI) interface. Output can be synchronized via the LDAC input signal. The output of the DAC 8555 in this circuit is in the range 0 V to 3.3 V. In order to generate output in the ±10 V range amplifiers have been connected to its analog outputs.

For rapid communication between controller and PC a FT2232H (FTDI) parallel to universal serial bus (USB) communication chip was used. This chip allows

[2] overlines for signals denote active low signals, i.e. signals that are logical true when their voltage level is low

communication at $480\,\mathrm{Mbit\,s^{-1}}$ and comes with a DLL that allows integrating the chip into the control software running on the PC.

5.4.2. Circuit and Schematics

The schematics of the printed circuit boards (PCBs) used in this work are printed on page 161ff.

The first (AFMIR01) of the two circuits described in this section mainly contains level shifters between the logic level of the FPGA (3.3 V) to that of the ADC (2.5 V). All logic and control signals are shifted using 74AVC8T245 (NXP Semiconductors). These components can translate digital signals from either of their two input voltages to the other. A single DIR pin determines which of the pins serve as inputs and which serve as outputs. For unidirectional signals - such as $\overline{\mathrm{RD}}$WR, $\overline{\mathrm{SYNC}}$, $\overline{\mathrm{RESET}}$, $\overline{\mathrm{CS}}$ and $\overline{\mathrm{RDWR}}$ the direction signal can be hard wired. For the bidrectional data bus DB0-15 the direction signal has been connected to the $\overline{\mathrm{RD}}$WR so that the direction of the level shifter changes as necessary. The connectors on this PCB connect to one of the connectors of the FPGA board on one side and to the AD7760-EVAL board on the other side.

The second circuit (AFMIR02) contains additional level shifters for TTL compatible input and output. It also contains the DAC as and its output amplifiers. Level-shifting is in this case performed by 74LVC1T45 that are 5 V tolerant. A symmetric ± 15 V DC-DC converter provides the supplies for the AD8421 (Analog Devices) amplifiers that convert the output of the DAC to ± 10 V range. A second DC-DC converter provides 5 V for the level-shifters.

5.4.3. Basics of FPGA Design

Due to their parallel nature FPGAs cannot be programmed in common programming languages like C, instead hardware description languages (HDLs) are used to describe the logic circuit. The circuit is then converted by a synthesizer into actual connections between the different elements of the FPGA[3]. The most important difference between "normal" programming languages and HDLs is, that in“normal” programming languages all commands are expected to be treated sequentially, except for special cases. For HDLs it is exactly the other way around: every assignment of a value happens not only at the same time

[3] A good introduction to very high-speed integrated circuit hardware description language (VHDL), the HDL used in this work can be found in the book *Free Range VHDL*[106], available online at `http://www.freerangefactory.org`.

but continuously, except when the programmer specifically tells the program otherwise.

When designing for an actual FPGA most signals are set to change at the edge of a clock signal - a rectangular signal of a fixed frequency and pulse length. The clock signal is usually generated by an external frequency reference, however, it is possible to use elements of the FPGA to devide or multiply this external frequency to achieve higher or lower clock rates as needed.

An important part of FPGA design workflow is the simulation of the developed hardware design before synthesizing it and transferring it to the chip. For each part of the design, usually each entity[4], a test bench is written that provides the inputs to the part and records and evaluates its outputs. During development the test bench is run together with the part in the form of a register transfer level (RTL) simulation. This type of simulation only considers the change of values of the signals in the program but not signal delays due to the length of connections or maximum switching speed of elements. It also does not consider the available space on the FPGA. Hence, a design that works in RTL simulations does not necessarily work on an FPGA. Nevertheless, since RTL simulations allow direct viewing of all signals within the entity at the same time they can help to eliminate many errors that would be much harder to track down in the circuit itself.

Once the top level entity, i.e. that one which contains all other entities and connects to the pins on the package of the FPGA has been tested and is working it can be synthesized into a programming file for the FPGA. During synthesis unnecessary signals that either are always constant or don't result in any change to the output are removed, the remaining design is mapped to components of the FPGA. At this stage the timings of the signal inside and outside the FPGA are determined and it is ascertained that all signals arrive at their sink in time. At the end of synthesis a programming file is produced that can be loaded onto the FPGA.

In designing the controller a modular approach was chosen that allows re-designing parts of the controller without much hassle. Therefore, instead of one monolithic entity that performs the complete evaluation, multiple small entities were created each performing part of the data processing, i.e. one performs data acquisition, one low pass filtering one output and so on. For passing data between these processes a simple data bus protocol (loosely based on the Wish-bone protocol described later) was implemented. This protocol consisted of two

[4] entities are small units in a VHDL program that have defined in- and outputs; a real world counter part would be an IC

control signals and one data bus of an unspecified length. The control signals are STB, which is strobed high whenever the data bus contains new and valid data and CYC, which is intended to denote data packets that belong together by going low at the end of a package. The data bus is of the type standard_logic_vector, which means that the exact meaning of the bits (signed, unsigned, fix point data) is not defined. The programmer has to take care that the data is correctly interpreted by the receiving entity.

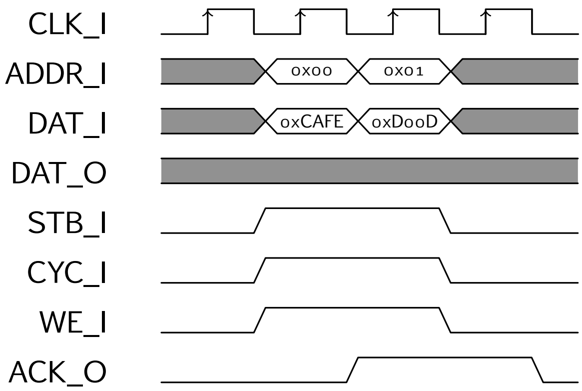
For readability's sake, the input data bus can be split into multiple ports, however, even when the input data ports are connected to different preceding entities still only a single STB signal is used, meaning that the programmer has to ascertain that the input values at the input bus arrive at the same time. The data bus, in the remainder of this work, is also called "pipe" as data flows through it from source to sink.

In all entities the control input ports are prefixed with PIPEIN_ and the output ports with PIPEOUT_. The data buses are either called DAT if their content is obvious or with a more descriptive name if there are multiple in- or outputs (e.g. SIN or COS).

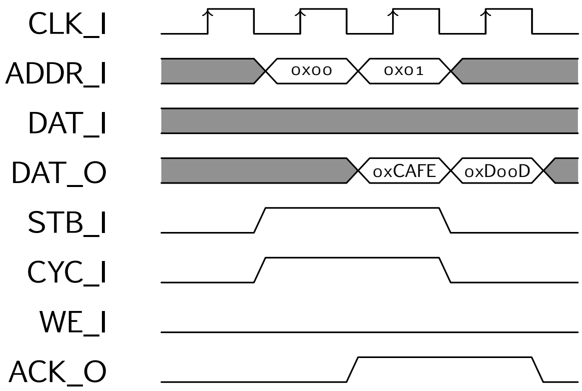
If an entity has to be reconfigured during run time, e.g. to change the repetition rate of a pulse generator or to change the output rate of an ADC it has configuration registers 16 bit deep with 8 bit for addressing the individual registers in which the current configuration is saved. The configuration can be changed by writing to a register via a Wishbone interface. The Wishbone bus [107] is an open standard for communication between different parts of an integrated circuit.

For this work a version consisting of data in and ouput buses DAT_I and DAT_O with 16 bit was chosen. An 8 bit ADDR_I input was used to select the register to be written to or read from. As control signals CYC_I, STB_I, ACK_O and WE_I were used. WE_I is used to switch between reading and writing data to the selected register. STB_I is used to perform a read or write and CYC_I is used to tie several reads or writes together. ACK_O is used signal that a message has been received. The timing diagram for the operations is depicted in figure 5.8.

A central control entity is used to (re)configure all entities as needed during run time. This entity also takes user input from the PC. User input is given by writing to a configuration register (16 bit wide, 8 address bit individual registers). Bit 0 in register 0 is set to 1 to start the data acquisition and output and to 0 to stop output for all versions of the controller treated hereafter. The meaning of the other registers is described in table 5.1 on page 90.



(a) Write operation. First register 0x00 is written to then register 0x01 is written to. Each write is acknowledged with by the recipient by raising the ACK_O signal.



(b) Read operation. First register 0x00 is read from, then register 0x01 is read from. The value of each register is output on *DAT_O* and *ACK_O* is raised to signal that the value is valid.

Figure 5.8.: Timing diagrams for read and write operation via data bus. All operations are performed on the rising edge.

5.4.4. Common Components

In this section general components that can be reused different hardware designs will be briefly described.

AD7760 Interface

The interface to AD7760 consists of two entities. AD7760_control connects to the AD7760 chip, while pipe_source_AD7760 is a wrapper that translates the register writes inside the FPGA to inputs for AD7760_control and outputs new samples into a pipe.

AD7760_control is implemented as a state machine with states called IDLE, SET, MEASURE and SYNCHRONIZE. After a reset the component is in state IDLE. State changes can be requested by putting the selected state in STATE_IN input and setting CHANGE_STATE to '1'. A successful change of state is acquitted by setting STATE_CHANGE_ACK to '1'. State SET writes the settings in the inputs GAIN, OFFSET, FILTER1, FILTER2_DEC and FILTER3 to the corresponding registers of the AD7760 and then returns to IDLE. SYNCHRONIZE waits until SYNC_IN is pulled low and then changes to state MEASURE, which will begin measuring when SYNC_IN is high again. State MEASURE ouputs data until a new state change is requested, upon which it will return to IDLE. For a state diagram see figure 5.9.

pipe_source_AD7760 has three configuration registers which can be read to and written from using the Wishbone protocol. The registers are for gain (register 0), offset (register 1) and digital filter settings (register 2) of the ADC. Registers 0 and 1 are written to the corresponding registers of the AD7760. Bit 15 of register 2 is passed on as filter 1 flag to AD7760, bits 14 to 12 are passed on as DEC2 to DEC0 and bit 11 is passed on as filter 3. All 24 bits of data output by AD7760_control are passed on by pipe_source_AD7760 together with an STB signal. Additionally, the flags denoting out of range input and valid data are output as well.

The AD7760 IC has an internal digital filter (that is initially configured as a low pass filter) and optional decimation to reduce the output data rate and remove high frequency signals. Down sampling and filter calculations introduce a delay, given as the delay in response to a step change in the data sheet. The actual delay varies due to the enabled filters. When filter one and three are enabled this leads to delay of approximately 30 ADC samples. To synchronize signals with the ADC output they can be sent to an AD7760_DELAY_SYNCHRONIZER entity. This entity will delay all input signals by a user selected number of AD7760

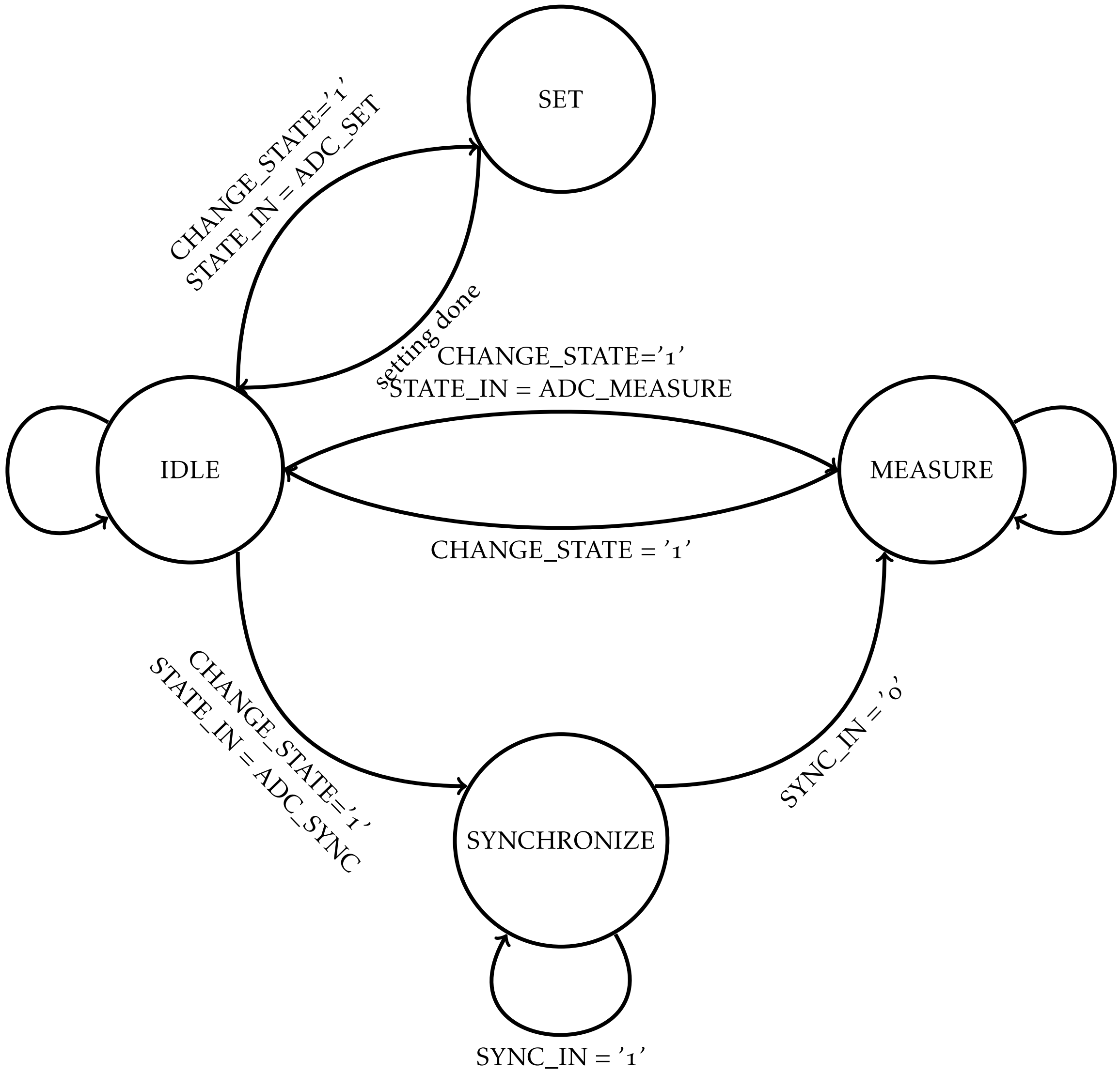


Figure 5.9.: State diagram of AD7760_control.

samples.

Digital Filters

To implement finite impulse response (FIR) filters in this work Xilinx IP Cores as generated by Xilinx FIR_compiler 6.3 are used. FIR_compiler converts user selected filter coefficients into fix point representation and optimizes the filter for maximum dynamic range. IP Core FIR filters also allow down sampling of data and keeping unfiltered data synchronized with filtered data.

The interface of the IP Cores follows the Xilinx AXI4-Stream [108] protocol. Data to be filtered is input on the port s_axis_tdata and is read when s_axis_tvalid is pulsed high. Synchronized unfiltered data is input on s_axis_tuser at the same time. The port s_axis_tready is high whenever the IP core is ready for the next sample. Filtered and synchronized data are output on m_axis_tdata and m_axis_tuser respectively, valid data is denoted by pulling m_axis_tvalid high. When the input data rate is not so high that the IP Core has to pull s_axis_tready low, AXI4-Stream can be used directly with the pipe connections used for the other components in this work.

For designing FIR filters the scipy functions signal.kaiserord and signal.firwin were used. These functions allow specifying pass band and stop band ripple as well as the width of the transition region and return the FIR filter coefficients. The filter coefficients are saved in a .coe file formatted for use with the FIR_compiler. When the filter is to be used for down sampling then the filter coefficients are rounded up to the next multiple of the decimation factor to achieve a more efficient design.

Analog Output via DAC8555

DAC8555 (Texas Instruments) is a 16 bit DAC that has four independent analog outputs. The output voltages of the DAC are set via a SPI compatible interface.

Like the interface of the AD7760 the interface of the DAC8555 consists of two components: one component - wb_DAC_sink - that has input pipes for updating the output voltages and a wishbone interface for configuration and one component - DAC8555 - that interfaces with the IC itself. wb_DAC_sink has a fairly simple design. It has four 16 bit inputs for setting voltages and an STB_I for each of them. There is only a single configuration register, the bits 0 to 3 of which are used to enable (1) or disable (0) each the outputs 1 to 4 individually.

The component DAC8555 has a single process with a state machine that continuously writes one analog value after the other to the IC and after having written

0x12	address	value[0]	value[1]	0x13

(a) From PC to controller.

0x12	index	cycle[0]	cycle[1]	X[0]	X[1]	X[2]	X[3]	0x13

(b) From controller to PC.

Figure 5.10.: Message protocol between controller and PC. Each box is a single byte.

all enabled values, pushing them onto the outputs of the DAC by asserting the LDAC pin of the IC. The clock signal necessary for the SPI interface of the IC is slow enough that it can be generated in the main process of the DAC8555 component.

The code for wb_DAC_sink and DAC8555 are printed on page 179ff.

USB Communication via FT2232H

The USB inferface between the controller FPGA and the PC is performed via the synchronous first in - first out (FIFO) mode of the FT2232H (FTDI) IC, this allows communication at a data rate of up to 480 Mbit s^{-1}.

Development of the interface included development of an FPGA design for sending and receiving messages via the FT2232H IC, as well as a library for encoding and decoding messages on the PC.

Message Protocol Messages in the FT2232H are transmitted as a series of bytes without any user settable context information. Hence, for any type of message larger than a single byte, a control protocol that is used by both ends of the communication has to be implemented.

Since settings are entered on the controller via writing to a register, messages from the PC to the controller consist of a register address (8 bit) and a register value (16 bit), in total 24 bit. In addition to the message bytes, three control bytes are used. Start (0x12) and stop(0x13) bytes denote the begin and end of a message. An escape (0x7D) byte precedes any message byte that has the same value as a control byte. The full message protocol is depicted in figure 5.10.

The communication from controller to PC is not as well defined as in the opposite direction, as different control schemes might output different kinds of data to the PC. For the mean modulus controller a scheme consisting of one index byte, two bytes containing the cycle count, and four bytes containing an amplitude data point were used. However, as described below, the message format sent by the controller can be quickly adjusted by changing the length of the input array in the hardware design (and the logic decoding the data on the PC).

Hardware Design In synchronous FIFO mode the FT2232H has a data bus consisiting of 8 pins, six control pins (RD, WR, TXE, RXF, OE, SIWU) and one clock output that runs at 60 MHz [109]. The protocol needed for interfacing with the FT2232H in synchronous FIFO mode is documented in the part data sheet [109] and in the tech note for the synchronous FIFO mode [110]. TXE and RXF are held low by the FT2232H whenever data can be read or written from the chip, respectively. A read operation is performed by first switching the direction of the data bus by pulling low the OE pin and at the next clock edge pulling low the RD pin, upon which the FT2232H starts to output one byte after the other to the data bus. As long as the RXF pin is low, additional bytes can be read. For writing to the FT2232H, the TXE pin has to be low. As long as the TXE pin is low, data can be writting by setting the bits of the data bus and pulling WR low.

The hardware design for interfacing with the FT2232H consists of 3 parts: one hardware interface that writes and reads bytes from and to the chip, one that encodes messages into strings of bytes and one that decodes the incoming string and writes data to the control registers.

FT2232H_FASTER is the entity that is connected to the FT2232H IC (code see 3). The important detail of the implementation of this part of the controller is that due to the strict timing constraints of the FT2232H, the entity has to use the clock signal of the FT2232H chip instead of that of the rest of the FPGA. To ensure the integrity of signals across the clock domains, the message bytes are passed from and to the entity via two FIFOs generated using the Xilinx fifo_generator 9.3. The FIFOs are configured for input and output in two different clock domains. In addition to the message bytes only the reset signal crosses the clock domain. FT2232H_FASTER checks if data is available to be read from FT2232H, and if it is starts reading it, if no data is available, FT2232H_FASTER checks if there is data to be written in the input FIFO and starts writing if the TXE pin is low.

The entities that encode data for transmission (WRITE_FT2232H_PROC) and decode received data (READ_FT2232H_PROC) are in the same clock domain as the rest of the FPGA design. WRITE_FT2232H_PROC is agnostic to the meaning of the data that it encodes. It takes number of input bytes (the actual number can be configured in the generic map section of the instantiation) prepends a start byte and then iterates over all input bytes to escape all bytes of the value of signal bytes. At the end a stop byte is appended. When encoding a message, WRITE_FT2232H_PROC is generating bytes faster (1 byte per cycle of a 120 MHz clock) than can be written to the FT2232H (slightly less than 1 byte per 60 MHz cycle). However, since the input of FT2232H_FASTER is buffered by a FIFO only the average data rate has to be lower than 60 MHz.

Bytes that have been received from the FT2232H are then decoded by the entity READ_FT2232H_PROC. After detecting a start byte the entity reads one address byte and two data bytes. If the byte following the second data byte is a stop byte the data is written to the register, otherwise it is discarded. If connected accordingly in the hardware design, READ_FT2232H_PROC can also be used to write data to pipes in the controller - e.g. to an analog output. Four pipes are provided that are updated with the value in the data bytes if the highest bit of the address is set to 1, the pipe is selected through the value of the seven lower adress bits.

DLL for Communication via FT2232H

FTDI provides the D2XX library for configuring the FT2232H and communicating with it. Using this library a wrapper was developed that performs the correct settings to put the IC into synchronous FIFO mode and automatically decodes incoming bytes into a structure data type. The operations to decode incoming messages are simple (e.g. type casting, copying of raw bytes, checking for message length,...) but need to be performed as fast as possible, the wrapper was written as a C DLL. Once compiled the wrapper can either be used in other C programs, or imported into higher languages such as Python.

Using open MojoConnectToolBox first the connection to a FT2232H IC connected to the PC is opened by calling the function OpenConnection(). To make sure that the controller connects to the correct FT2232H, OpenConnection will only connect to FT2232H chips that have been named "RE-PTIR A". Next the chip is put into synchronous FIFO mode by calling ProgramFIFOChip(). Data can be transmitted to the FT2232H using the TransmitData and TransmitDataPackage functions. Raw byte data can be received from the FT2232H using either ReceiveData or ReceiveDataPackage. Decoded messages can be received using the DecodeMessages function, wich returns readily decoded messages as an array of type rx_message:

```
typedef struct rx_message {
    unsigned char index;
    unsigned short tau;
    int32_t amplitude;
```

```
} rx_message;
```

Test Bench for Full Controller Designs

For integration of full controller designs a test bench simulating the response of the AFM signal to laser excitation as well as parts of the AD7760 and FT2232H chips was implemented.

Response of the cantilever to laser pulses is simulated using a finite time difference simulation. The cantilever is modeled as a point mass on a spring while with a Derjaguin-Muller-Toporov model for tip sample interactions (for both see e.g. [36]).

The point mass model of the cantilever is given by

$$\ddot{z} + \frac{\omega_0}{Q}\dot{z} + \omega_0^2 z = \frac{F_{DMT}(z)}{m} \tag{5.1}$$

The model is discretized using the finite difference method:

$$\ddot{z} \rightarrow \frac{z(t) - 2z(t-\Delta t) + z(t-2\Delta t)}{\Delta t^2} \tag{5.2}$$

$$\dot{z} \rightarrow \frac{z(t) - z(t-2\Delta t)}{\Delta t} \tag{5.3}$$

$$z \rightarrow z(t-\Delta t) \tag{5.4}$$

Sample heating is simplified as a constant deviation ΔT from the sample equilibrium temperature, while the laser is emitting and equilibrium temperature while the laser is off. While this is a simplified model, Lu, Jin, and Belkin[38] have shown that the sample temperature stays virtually constant during an EC-QCL pulse for thin samples. Therefore, the sample height was increased by a constant value of 5 pm, whenever the laser input indicated that the controller was triggering a laser input signal.

d_0 is the initial tip sample separation and z is the vertical position of the tip. The current tip sample force is calculated using a user set value for the pull off force and value of the atomic distance set a_0 set to an arbitrary value of 0.4 nm [111]. Hamaker constant (H) of polymers and gold were taken from [112].

$$F_{DMT}(d_n) = \begin{cases} -\frac{\mathrm{H}r_{tip}}{6a_0^2} + \frac{4}{3}\mathrm{E}^{\star}\sqrt{r_{tip}\,(a_0 - d_n)^3} & d_n < a_0 \\ -\frac{\mathrm{H}r_{tip}}{6d_n^2} & d_n \geq a_0 \end{cases} \tag{5.5}$$

Viscosity of the sample is currently neglected, but can be added as an additional term of $-\frac{\pi r_{tip}\eta_n}{h}\sqrt{a_0 - d_n}\dot{d}_n$ [37], where η_n is the vertical tip sample viscosity.

The updated deflection value z is high-pass filtered ($f_C = 25\,\mathrm{kHz}$) and scaled with sensitivity scaling factor that is used to convert the vertical position to a deflection read out. The VHDL code implementing the cantilever simulation is printed on page 190ff.

To give access to the signal at different stages of the signal processing pipe the pipe_scope_int component (see page 207) can be connected to the data, STB and CYC signals. When STB and CYC are asserted, the current value of data is printed to a file, together with a time stamp in nanoseconds. For further processing the data can be read into a numpy array using the following python code:

```python

from numpy import loadtxt
from bitstring import BitArray

def bits2int(val):
    if "X" in val:
        return np.nan
    if "U" in val:
        return np.nan
    bs = BitArray(bin=val[1:])
    if val[0] == "s":
        return bs.int
    else:
        return bs.uint

data = loadtxt("<filepath>.txt",
               delimiter=";",
               converters={0:float, 1:bits2int})
```

To ensure that entities of PIPE_SCOPE_INT are not synthesized, the instantiation of the component can be surrounded with

```
--pragma synthesis_off
    <instantiation of pipe_scope_int>
--pragma synthesis_on
```

5.4.5. Sweep Controller

Working Principle

For sweep based controllers the repetition rate of the laser has to be chirped across the user selected frequency range, cantilever amplitudes have to be recorded and the maximum amplitude during the chirp has to be determined.

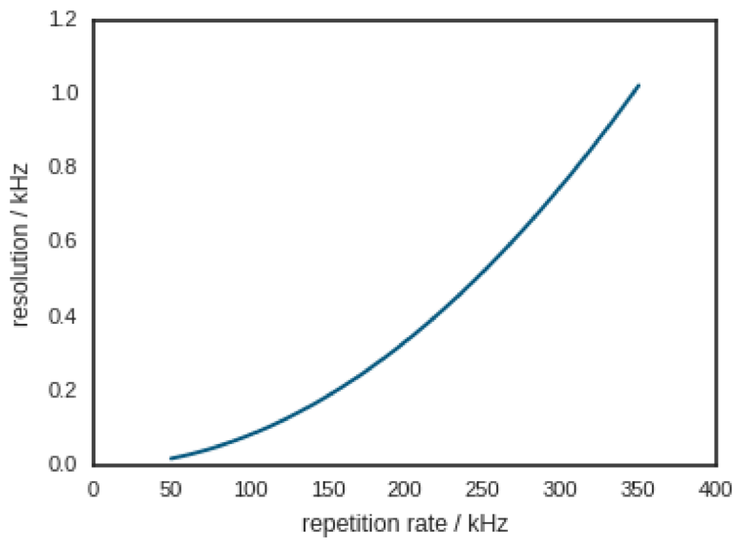


Figure 5.11.: Change of the resolution of the repetition rate with increasing repetition rate.

Therefore, in addition to the interface components and filter described above further components were needed.

Pulse Generator For generating pulse trains at different duty cycles and repetition rates a pulse generator was developed (see code on page 192). In this component length and repetitions rate can be set as a number of cycles of the FPGA clock signal. At the clock frequency of 120 MHz that was used for the FPGA for most of this work this results in a resolution of 8.33 ns for the pulse length. Since the count of clock cycles is indirectly proportional to the repetition rate the resolution of the repetition rate decreases when the repetition rate increases (see fig. 5.11).

The pulse generator outputs the pulse as a digital signal that can be directly connected to one of the pins of the FPGA. It also generates additional information for use by other components. The maximum count that is currently set and the current count are output into pipes. From these signals the current phase can be determined, which can be used to calculate the phase for reference signals for a lock-in amplifier.

Maximum Detector To detect the maximum amplitude during the chirp (i.e. the PTIR signal that will be output) all amplitudes that are recorded during the chirp are compared and the largest one is output together with the corresponding amplitude at the beginning of the next sweep.

The component determines the start of a chirp via a start cycle length that is written to its register. Therefore, delays in signal processing do not pose a problem to the component. (for the source code see page 196).

Downsampling for USB Output The USB interface of the controller is not fast enough to handle transmitting measurement results at the maximum data rate of the ADC. To reduce the amount of data that needs to be sent only the last value collected during a cycle is sent to the FT2232H IC. The design of the downsampling component is printed on page 198.

Mean Modulus Controller

In section 3.4 the RMS detector was introduced as a simple method for measuring the amplitude of an electric signal. However, for the implementation of the controller, the mean modulus $|x|$ was chosen instead for detecting the amplitude.

Both the mean modulus and the RMS are proportional to the amplitude of a sine wave, but with differing proportionality constants. The mean modulus was chosen over a RMS detector based on its easier implementation in a fix-point math based FPGA design. Calculating the modulus is an operation that can be performed within a single cycle and reduces the number of bits that are sent to the low pass filter by one relative to the input while calculating the RMS includes squaring the input (which doubles the number of needed bits), then low pass filtering and finally calculating the square root. Hence, the low pass filter has to work at twice the number of bits in order to keep the precision of the input and needs more complicated logic.

A flow diagram of the mean modulus-based controller is depicted in figure 5.12. The laser pulses which are used to excite the sample are triggered by the pulse generator component. The current cycle length is output by the pulse generator delayed by 30 ADC samples and then kept synchronized with the digitized deflection signal throughout all other stages.

The AFM deflection signal is amplified by the analog amplifier described in section 5.1. The amplifier removes low frequency changes (<50 kHz) from the deflection signal and does therefore not transmit the direct current (DC) offset of the deflection signal. However, at a high amplification factor, the amplifier itself introduces a DC offset. Since the mean modulus does not reject any modulation frequency, the offset would be added to the output signal and lead to an offset in the detected amplitude.

Table 5.1.: Registers of mean modulus controller

<table>
<tr><th>register</th><th>content</th></tr>
<tr><td>0</td><td>bit
0 enable
2 bi-directional sweep
3 start increasing pulse length</td></tr>
<tr><td>1</td><td>sweep start (number of clock cycles)</td></tr>
<tr><td>2</td><td>sweep step size (number of clock cycles)</td></tr>
<tr><td>3</td><td>number of steps</td></tr>
<tr><td>4</td><td>number of repetitions of each sweep step</td></tr>
<tr><td>5</td><td>pulse generator pulse length</td></tr>
</table>

To remove the constant offset a digital high pass filter is applied to the ADC samples. The filter has a broad transition range (30 kHz) to decrease the number of filter coefficients that are needed. The cut off frequency of 25 kHz was chosen so that frequencies above 50 kHz are passed.

The high pass filtered signal is then converted to its modulus. This operation was performed inline using the ABS function from the IEEE.NUMERIC_STD VHDL library after type casting the output of the lowpass filter to SIGNED.

A low pass filter is used to convert the modulus into the signal corresponding to the amplitude. The PTIR amplitude that is output to the AFM controller via the DAC 8555 is determined by the max detector (see pg. 88). To allow viewing the current resonance curve each sweep is output to the PC after downsampling as described in section 5.4.5. The flow diagram of the data processing in the controller is depicted in fig. 5.12.

The controller has several registers that can be set by the user via the FT2232H as described in section 5.4.4 (see table 5.1 for a description of the registers). The values of these registers are read by the central control component. Changes of the parameters in register 1 to 5 are only read and applied when the enable bit goes from low to high. This mechanism was chosen so that the order in which the registers 1 to 5 are written to and the timing in writing are not critical for the measurement.

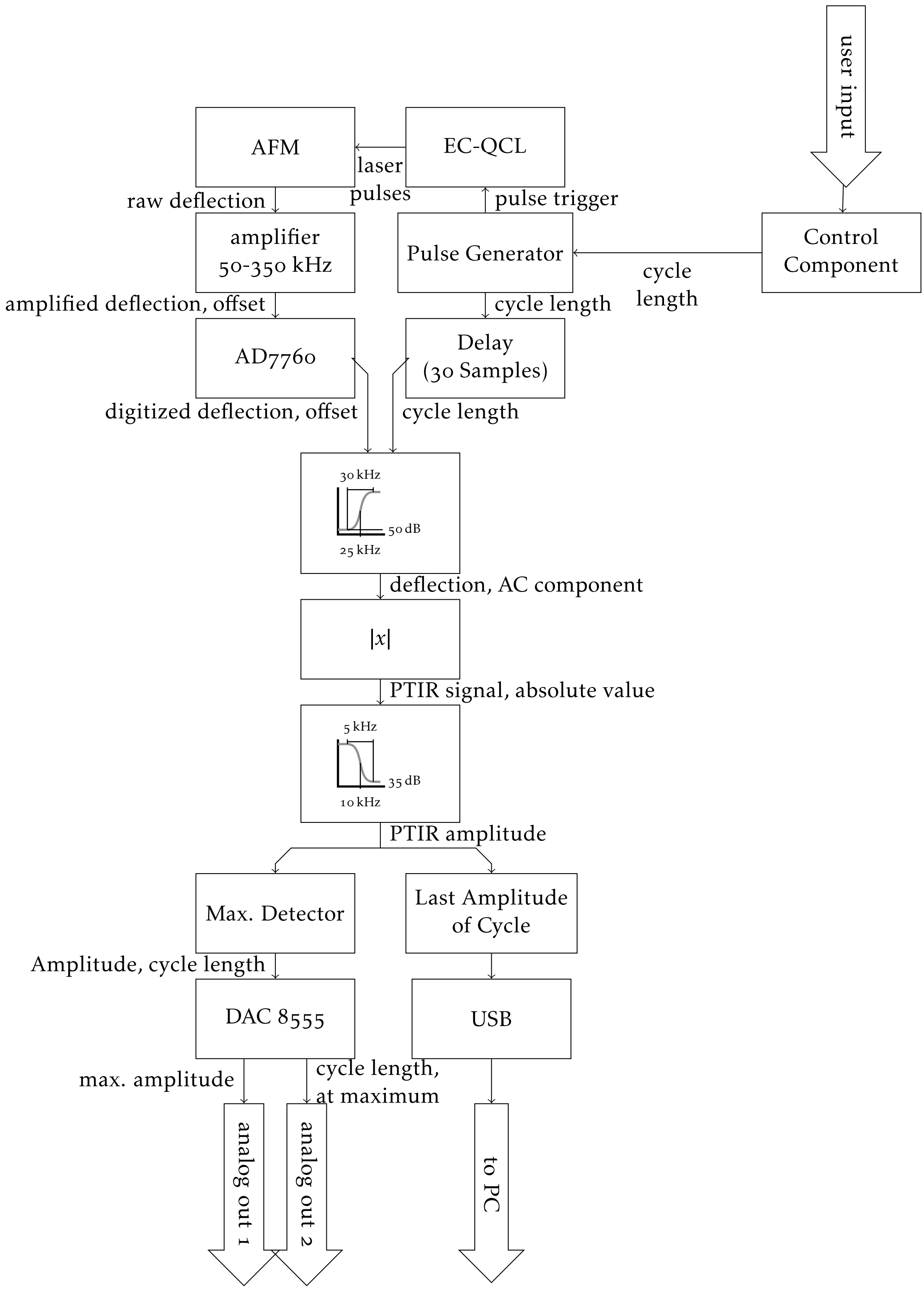


Figure 5.12.: Flow diagram of signals in the controller for the PTIR controller based on an mean modulus detector.

The control component has a fairly simple design (code see page 8). It has four different states

1. MAIN_STARTUP,
2. MAIN_WAIT,
3. MAIN_START_MEASURE and
4. MAIN_MEASURE.

MAIN_WAIT is the default state after a reset. If the controller has not been initialized, i.e. the initial configurations have not been written to the configurable components, the next state is MAIN_STARTUP, otherwise the control component remains in MAIN_WAIT until the user asserts the enable bit. The initial configurations for pulse generator, AD7760_pipe_source and DAC controller are saved in an array of TYP_WB_MESSAGE, a record type that contains the component the configuration is for, the register address and the data to be written. In state MAIN_STARTUP in each clock cycle one item from the configuration array is written to the corresponding component. After writing the last configuration, an additional wait time has been added to ensure that the AD7760 is fully initialized before moving on to the next state.

When the enable bit has been asserted by the user during start up, the next state is MAIN_START_MEASURE, otherwise, the controller returns to MAIN_WAIT. MAIN_START_MEASURE writes to the registers of the pulse generator component to select the initial cycle length and the user selected pulse length. The initial cycle length is also written to the register of the maximum detector as the beginning of a new sweep. The step following MAIN_START_MEASURE is MAIN_MEASURE.

In MAIN_MEASURE the controller waits the number of cycles set by the user and then switches to the next cycle length by writing to the corresponding register of the pulse generator. The beginning of a pulse is determined via the PULSE_OUT signal of the pulse generator. When the controller has changed the cycle length as often as set by the user in register 4, the controller either moves back to the initial cycle length (if the bi-directional sweep bit is set to 0), or switches from increasing to decreasing the pulse length or vice versa. The controller remains in MAIN_MEASURE until the user sets the enable bit to 0.

User Interface for the Mean Modulus Controller

To simplify adjusting the settings of the mean modulus controller, a python program has been developed that sends commands to the controller and displays the incoming measurement data (code see pages 223 and 228).

This program is written in Python (for easier prototyping). The ctypes library is used to call functions from the MojoConnectToolBox to communicate with the controller. When started, controller.py opens a plot window and a command line interface (CLI). The CLI is used to send commands to the controller, while the plot window is used to display the amplitudes of the last 10 sweeps.

When controller.py is executed, it creates an instance of the fasterController class. fasterController itself connects to the FPGA board via the MojoConnectToolBox via the Python wrapper FT2232H.py. Updating the plot window with new data (in _display_process) and calls to the MojoConnectToolBox (_comm_thread) can block the execution of their thread hence they have been moved to separate threads. Plotting itself is performed in a second process using the pyqtgraph.multiprocess library. The threads communicate with eachother via queues.

_comm_thread checks for new data in comm_queue. If the new data contains measurement parameters, they are encoded and sent to the controller using the fasterController.send_message. comm_queue can also contain the command "stop", upon which _comm_thread writes zeros to register 0 of the Controller, to stop the sweep, or "shutdown", which _comm_thread passes on to _display_process and then shuts down itself. If there is no new data in comm_queue _comm_thread tries to receive data from the controller (via FT2232H.readMessages_ctypes). If data is available, it is parsed into separate sweeps and each sweep is put into display_queue.

When started the _display_process thread starts a second process for plotting. It then waits for new messages in display_queue. New messages can either contain sweep data or "shutdown". Sweep data is transferred to the plotting process if the time since the last time plot data was sent is longer than 250 ms. This is done because transferring the plot data and plotting adds some delay displaying all sweeps at higher sweep rates would lead to a high lag between the plot curves and the actual measurement. If the "shutdown" command is received _display_process shuts down the plotting process and then returns.

The two most important functions in the CLI are start_sweep() and stop(). These two are calls to the methods of fasterController of the same name.

start_sweep is called with the start and stop frequency of the sweep, the number of steps into which this frequency interval should be split and the number of cycles the controller should pause at each step. Finally, the bidir arguments can be set to True for bidirectional sweeps. In start_sweep the input frequencies are converted into cycle counts and their difference is divided into integer steps. The number of cycles corresponding to the the start frequency as well as the step size, step number and hold value are then put into comm_queue to be sent to the controller. Calling stop puts "stop" into the comm_queue.

Test Bench Results: Mean Modulus Controller

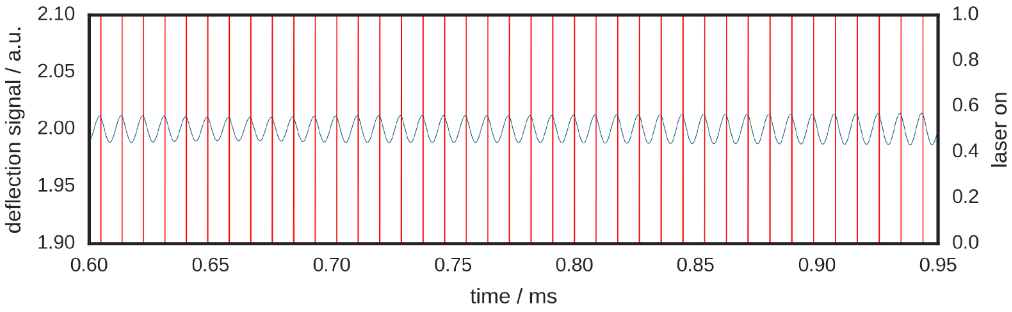
The test bench was configured using a sensitivity factor of 10^9 and a Q of 50. The cantilever parameters were set to k=0.2 N m^{-1} and r_{tip}=25 nm as in the CONTGB-G (BudgetSensors) cantilever used in most of the measurements in this work.

To test the capability of the system to sweep a single resonance the controller was set to a start at 1044 cycles (114.943 kHz) and then perform 100 steps of 2 cycles, waiting for 2 cycles after each step. As expected, the cantilever oscillations are low when the repetition rate of the laser pulses does not coincide with the contact resonance and when the cantilever is excited close to its resonance (see fig. 5.13.

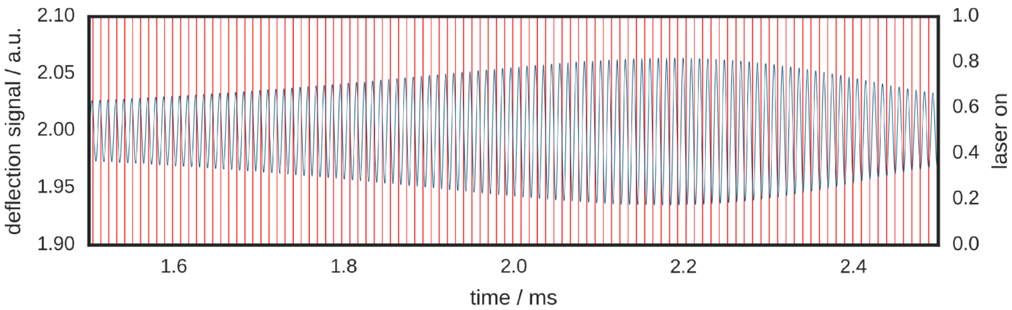
The signals inside the mean modulus controller are depicted in figure 5.14. The 2 V offset in the deflection signal is removed by the high pass filter (see fig. 5.14a), leaving only the oscillation induced by the laser pulses centered around the zero line. After converting to the absolute value (see fig. 5.14c), the signal is low pass filtered resulting in a smooth envelope proportional to the amplitude of the oscillations (see fig. 5.14d). Finally, the cycle length is output via USB (see fig. 5.14e).

Of course, the linearity of the output of the controller at different excitation frequencies is also of interest. To test the linearity, the test bench was run with sample expansion parameters from 1 pm to 40 pm.

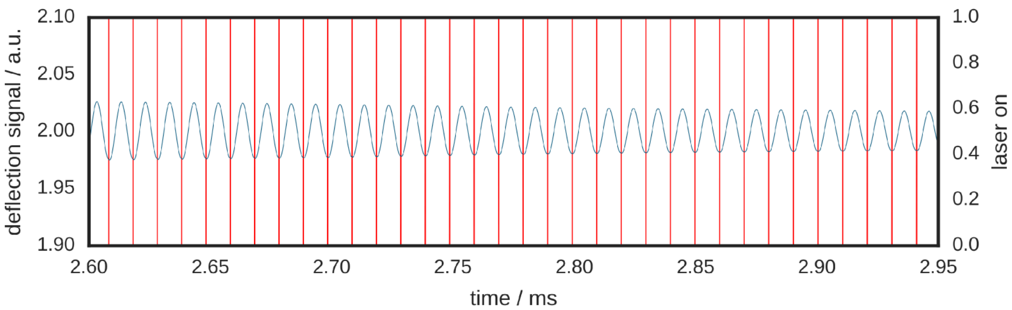
At the selected parameters the maximum amplitude shows an almost linear dependence on the expansion, with a slight decrease of the slope towards higher expansions (see fig. 5.15a). The detected frequency of the maximum is not independent of the sample expansion, instead, at higher frequencies the maximum is slightly shifted in the direction of the sweep (i.e. in this case towards lower frequencies). The reason for this is found in the slower reaction to external changes close to its resonance (as stated in section 3.4). Hence, when the sweep rate is reduced to half of the original speed, the linearity of the measurement is improved and the frequency at which the resonance is detected moves against the excitation frequency.



(a) Excitation frequency above resonance.

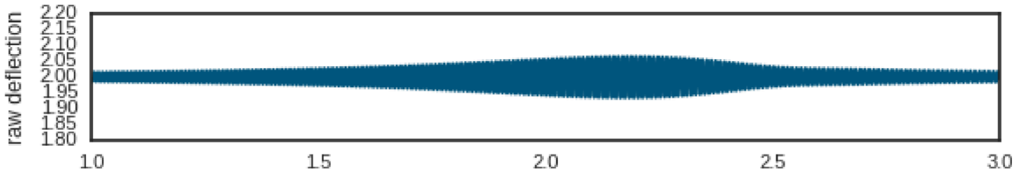


(b) Excitation frequency across resonance.

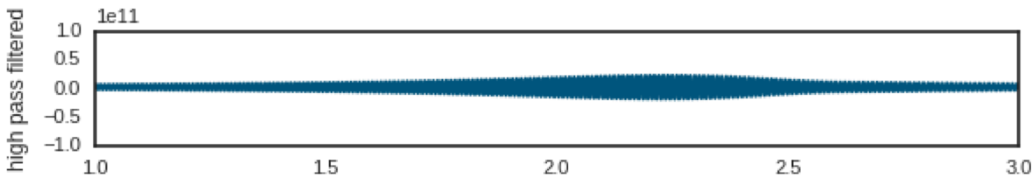


(c) Excitation frequency below resonance.

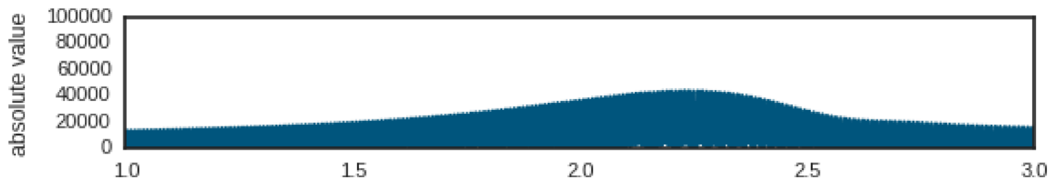
Figure 5.13.: test bench: cantilever response and laser pulse input.



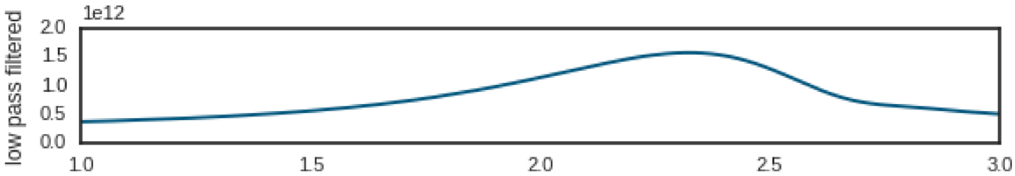
(a) Raw deflection signal.



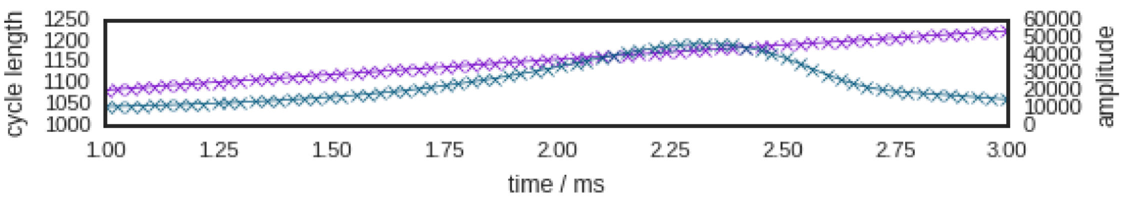
(b) Digitized, high-pass filtered signal. DC offset has been removed.



(c) Modulus of the high-pass filtered signal.

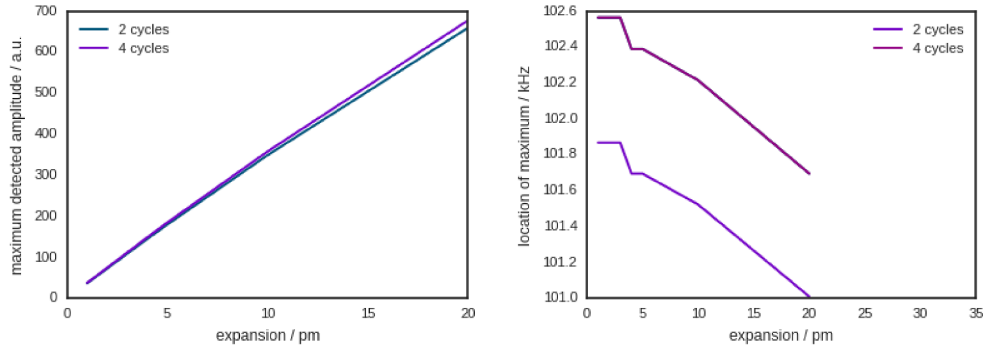


(d) Through low-pass filtering of the mean modulus, the amplitude of the oscillation are recovered.



(e) USB output of the controller.

Figure 5.14.: Signals inside the mean modulus controller.



(a) Amplitude read-out of the controller at a range of sample expansions.

(b) Maximum frequency read-out of the controller at a range of sample expansions.

Figure 5.15.: Amplitude and frequency output of the mean modulus controller at different sample expansions.

Measurements using the Mean Modulus Controller

In order for the AFM controller to be able to record the amplitude and frequency output of the mean modulus controller together with the cantilever position the amplitude and frequency analog outputs have to be recorded via an analog input of the AFM controller. For the Keysight 5400 this can be done by using the Aux In connectors of the MAC III box. For spectroscopy, the amplitude output has to be recorded together with the Tuned wavelength signal of the EC-QCL. In this work this was done using the ADC setup described in section 5.1.

The first step in a PTIR measurement using the setup described in this work is adjusting the laser focal spot back to the location of the cantilever. The cantilever is brought in contact with the sample and the set point is optimized (decrease set-point until contact is lost, slowly increase set-point until contact is reestablished) and the EC-QCL is tuned to a wavelength that is strongly absorbed by the sample. If no absorption band of the sample is known, the laser is set to the wavelength at which it has the maximum power ($\approx 1450\,\mathrm{cm}^{-1}$). The mean modulus controller is set to sweep the range of 50 kHz to 250 kHz. For the given controller clock settings this range can be divided into 1920 steps. With hold set to 1 this frequency range would be covered in 23 ms. A hold value of 5 gives still result in more than 8 sweeps per second.

The focal spot adjustement is now performed by moving the focusing mirror vertically (y direction) until the controller amplitude value reaches a maximum (in Keysight PicoView, the controller amplitude ouput can be displayed using the Signal vs. Time window). The alignment of the spot can also be slightly improved by moving the cantilever left and right (x direction) by a few tens of micrometers, again adjusting for a maximum signal. Once the focal spot is adjusted, the frequency range of the sweep can be reduced by honing in on one of the contact resonances of the system. Since the location of the resonances can be shifted at high sweep rates a multi step process for adjusting the sweep parameters has proven to be advantageous: first the sweep range is reduced to half or a third while keeping the resonance in the center, then the position of the resonance is reevaluated in the plot window of the controller. If the resonance is not close to the center of the sweep range, the borders of the range are changed accordingly.

When measuring the first time with a newly mounted cantilever, the z direction also has to be adjusted. This can be done as described in section 4.3.2.

In its current implementation the mean modulus controller does change the swept range without user input. The user has to define the area that is to be swept beforehand. This is done by placing the cantilever on different parts of the sample (usually the regions exhibiting different contact resonance frequencies can be seen in the topography image of the sample) and noting the frequency of the maximum amplitude in the plotting window of controller.py. The sweep range is then set to encompass all found frequencies and a little bit of "wiggle room" below and above the minimum and maximum found frequency, respectively.

Steps and hold are then again increased until the amplitude signal reaches its maximum while keeping the sweep range constant. The start_sweep function returns the cycles that are currently covered and prints the frequency at which sweeps are performed. The system is now ready for imaging or spectroscopy.

For imaging the laser is set to a single wavelength and then a scan of the cantilever is performed. If the sampling rate of the AFM, i.e. lines per second times samples per line, is higher than the sweep frequency output by start_sweep then the maximum amplitude and the frequency images will look pixelated. In this case the scan rate or the sampling rate have to be decreased.

For spectroscopy the cantilever is placed on the sample and the amplitude and as well as the Tuned signal of the EC-QCL are recorded. The wavelength at which the EC-QCL is currently emitting is determined as described in section 4.1.3. It is important to note that the sweep rate of the controller should again be high enough so that the wanted spectral resolution is achieved.

The spectroscopy and imaging capabilities of the mean modulus controller are tested in section 6 and compared to those of a lock-in amplifier based setup in section 7.

Part III.

Results

6. Measurement Results with the Developed Controller

6.1. Spectroscopy

Spectroscopy of polymer films was performed as described in section 5.4.5. In short, the AFM tip was placed on one spot on the sample and held in position. Then the EC-QCL was swept across its wavelength range and the PTIR amplitude was recorded. Four scans were averaged.

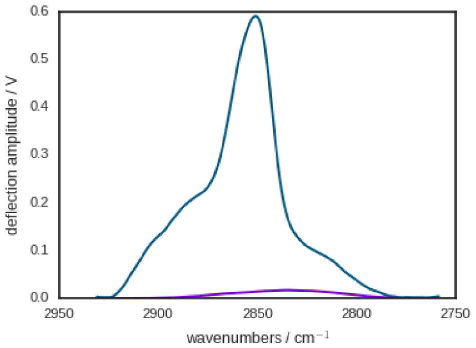
In first data treatment step, the constant offset (≈60 mV) was removed from the single channel measurements. The thus corrected spectra were then low-pass filtered to reduce noise. The cut off frequency for filtering was selected according to the wanted spectral resolution of $8\,cm^{-1}$. Single channel spectra taken on a polystyrene film are depicted in figure 6.1.

After low-pass filtering the PTIR signal was converted to a spectrum by calculating the ratio of the signal measured on the polymer film and that collected on bare gold. By calculating the ratio changes in the wavelength dependent laser intensity cancel out. However, differences in the thermal conductivity and thermal expansion of substrate and sample and in their mechanical properties are not cancelled. Hence, spectra are always scaled by an unknown factor depending on mechanical ad thermal properties of the sample and the substrate. The calculated spectrum is depicted in figure 6.2.

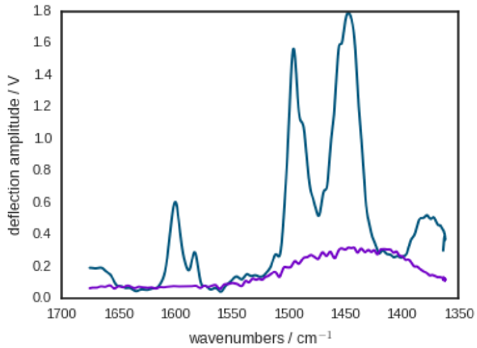
As can be seen the reproduction of the sample is best in the spectral range corresponding to QCL 2, here a very high laser power in conjunction with strong absorption bands lead to a high SNR signal. In addition, as QCL2 is the most intensive of the four QCLs it was also the one used for aligning the setup. The relative band intensities change somewhat between near-field and far field spectrum. I expect the reason to lie in a combination of optical (standing wave at the gold substrate) and mechanical reasons (cantilever deflection enters nonlinear region).

Some artifacts are seen at the edges of the gain range of the individual QCLs. The best example for this is seen in QCL 1, where the absorption band a $2925\,cm^{-1}$ seems to be shifted to lower wavenumbers because of this effect.

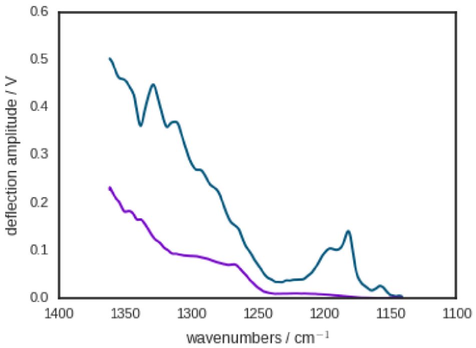
Spectroscopy of polystyrene was tested down to a film thickness of 8 nm with



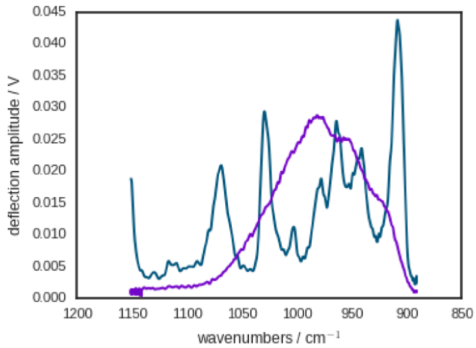
(a) QCL 1



(b) QCL 2

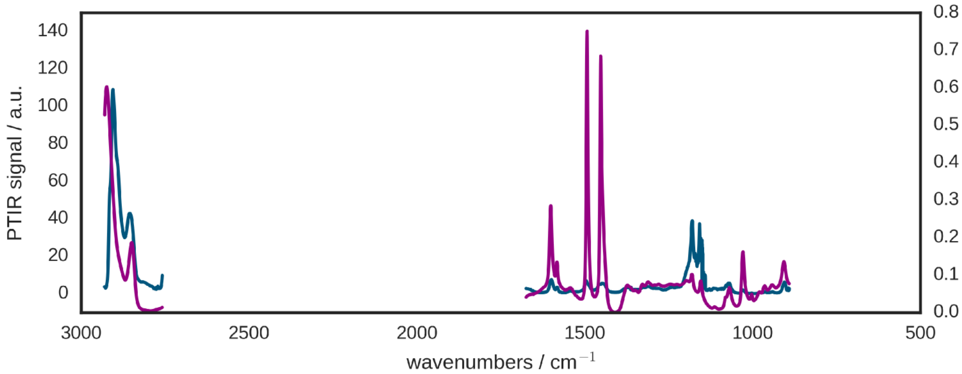


(c) QCL 3

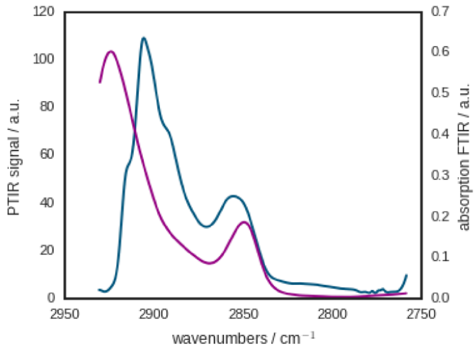


(d) QCL 4

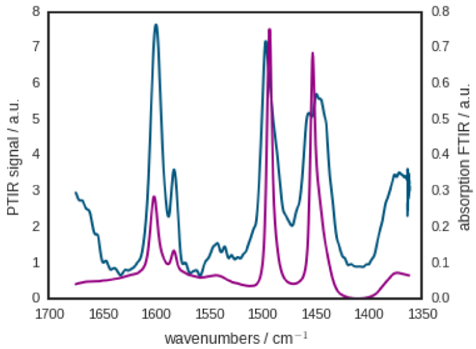
Figure 6.1.: Single channel of a polystyrene film collected with the PTIR controller. Blue lines correspond to measurements taken on the polymer film, purple were taken on the gold substrate. Differences in the intensities of the lasers lead to differences in the scaling of the signal.



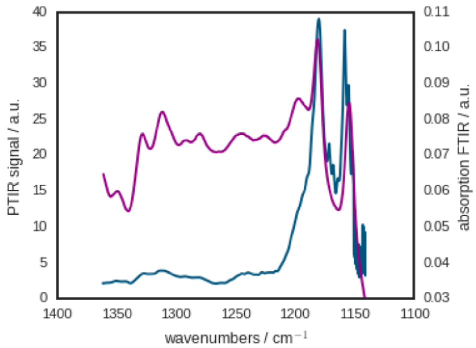
(a) Full Spectrum



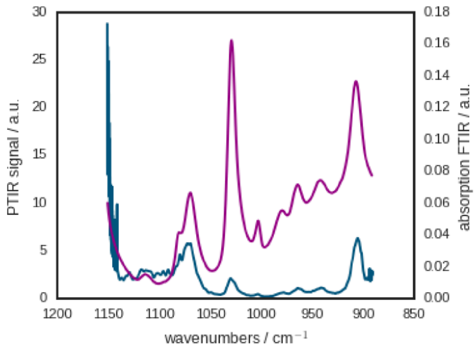
(b) QCL 1



(c) QCL 2



(d) QCL 3



(e) QCL 4

Figure 6.2.: Spectra of a polystyrene film collected with the PTIR controller. The ratio of background and sample spectrum are depicted in blue lines correspond to measurements taken on the polymer film, in red a far field FTIR spectrum of a free-standing polystyrene film is depicted.

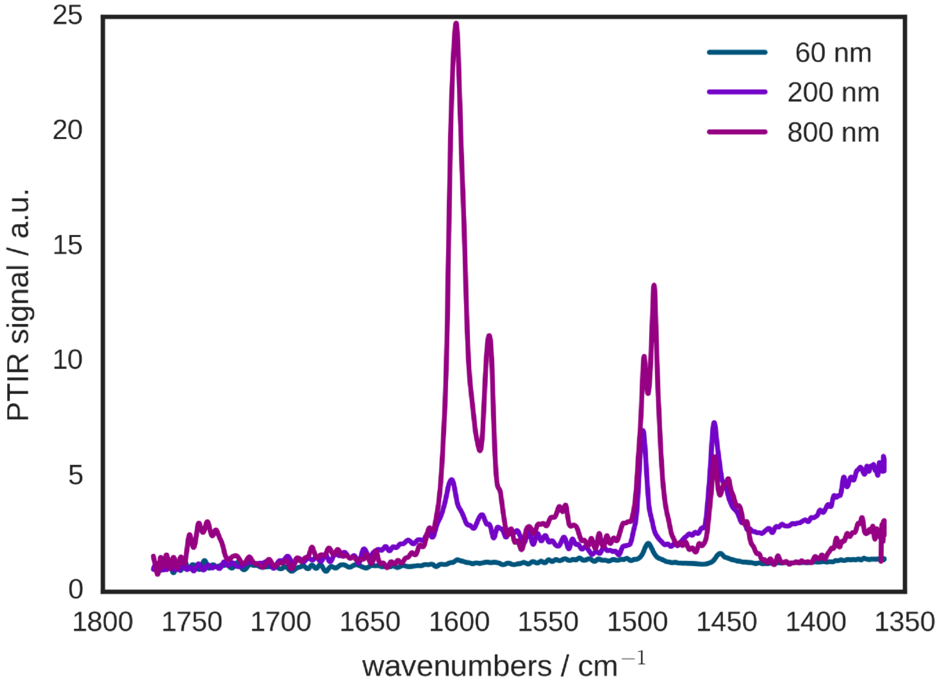


Figure 6.3.: Spectrum of 60 nm and 800 nm polystyrene thin films prepared on a gold surface.

films of 60 nm still resulting in strong bands for QCL 2 (see fig. 6.3) and films of thicknesses of 8 nm still being detectable (see fig. 6.4)[1]. It is interesting to note that the relative intensities of bands change depending on the thickness of the film. Possible explanations for this effect could be a wavenumber dependent contribution of background absorption in the cantilever to the signal which becomes more dominant as the PTIR signal of the sample becomes weaker.

6.2. Imaging

Spectral imaging was performed as described in section 5.4.5. A 60 nm polystyrene film was prepared on a physical vapor deposition (PVD) gold layer by spin coating. The step at an edge in the film was imaged with the AFM and height and IR signal were recorded. The EC-QCL was set to 1450 cm^{-1} which corresponds to the band giving the highest PTIR signal (high absorption and high laser power).

[1] All heights in this section were determined via AFM topography measurements. Hence, their accuracy is implied to be that of the AFM height measurement, ±10 %.

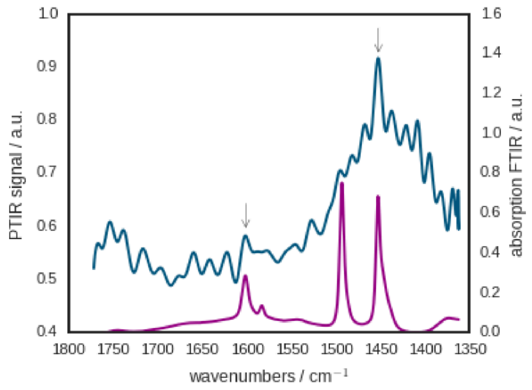
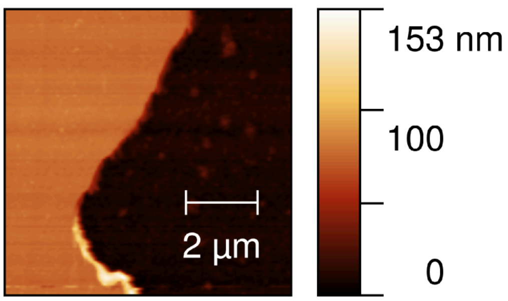
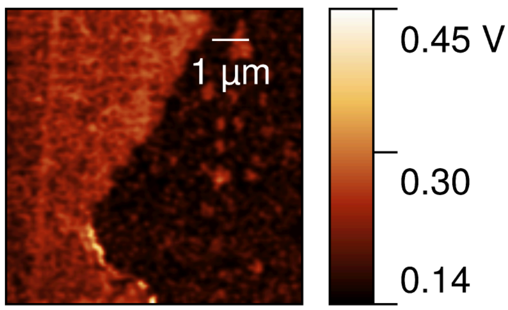


Figure 6.4.: Spectrum of a polystyrene film of 8 nm thickness prepared on a gold surface. While most bands are below the noise level, at 1450 nm a band is still visible.



(a) topography



(b) PTIR at 1450 cm^{-1}

Figure 6.5.: Image of the 60 nm step in the polystyrene film.

The controller was set to sweep a range from 80 kHz to 140 kHz in 120 steps holding 4 cycles at each frequency step. The sample topography and the PTIR image are depicted in figure 6.5.

The step response was evaluated by looking at a single line in figure 6.5 (see fig. 6.6). The IR signal follows the height of the sample, however, there is a slight shift in the location of the edge, possibly due to the edge expanding against the side of the cantilever.

As the “step” in the 60 nm film was quite wide, a thinner film was tested as well. A 8 nm polystyrene film was prepared on PVD gold by spin coating and imaged at 1450 cm^{-1}. At a film thickness this low, the roughness of the gold surface begins to contribute noticeably to the topography and the PTIR signal (see fig. 6.7). However, still a noticeable increase in the signal height is detected

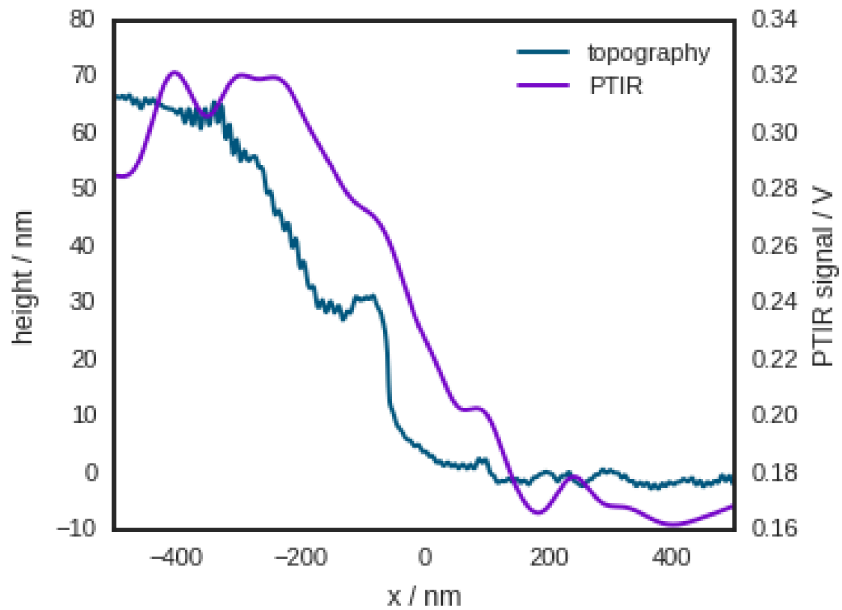
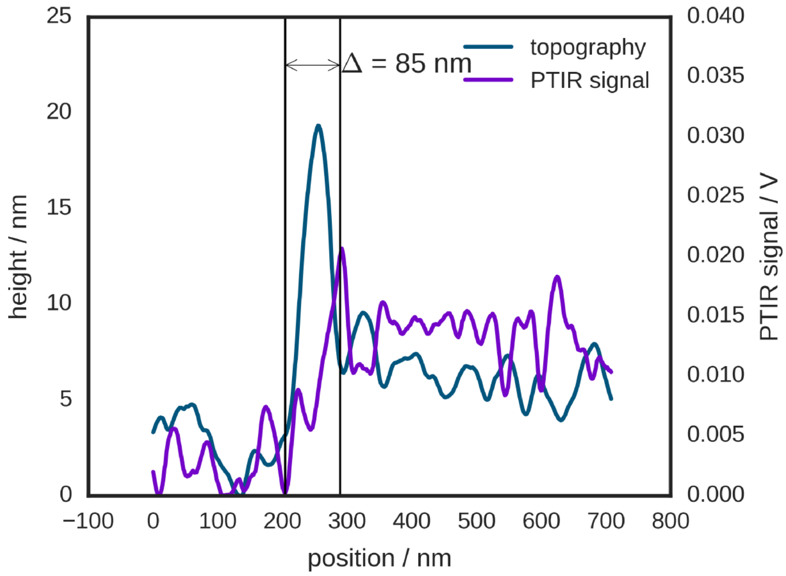


Figure 6.6.: Step in a 60 nm polystyrene film. The PTIR signal was recorded at 1450 cm^{-1}.

on the polystyrene film. However, in contrast to the measurement of the 60 nm film the measurement of the 8 nm film shows no offset in the measurement positions of topography and PTIR signal, giving credence to the idea, that the offset stems from an interaction between tip side and the edge.

(a) Overview



(b) Cross section topography and PTIR signal at 1450 cm^{-1}

Figure 6.7.: 8 nm polystyrene film on PVD gold. The blue line in (a) shows the approximate location of the crosssection in (b). The cross section was recorded in a second measurement at a higher resolution.

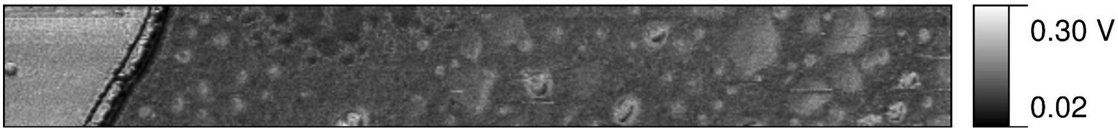
7. Comparison Lock-In Amplifier vs. Resonance Tracking

A comparison of the lock-in amplifier and a resonance tracking detector in the PTIR setup is not straight forward, as the resonance tracking scheme does not improve the figures of merit of the signal. That means, while the signal is more correct when using the mean modulus controller, the SNR might actually be lower with a lock-in detector, when it is perfectly adjusted to the resonance and the system does not change.

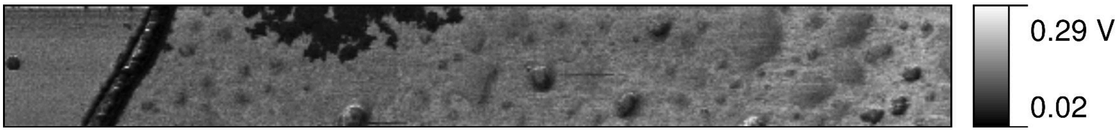
The advantage in using a resonance tracking scheme versus a bare lock-in detector for imaging can be seen in figures 7.1, 7.2 and 7.4. While the relative signal intensities of substrate and sample change significantly with the selected frequency of the lock-in amplifier, ranging from high intensity on the sample (see fig. 7.1b and 7.2a), and equal intensity on sample and substrate (see fig. 7.1c and 7.2b) to some parts of the substrate generating a higher amplitude than the sample (see fig. 7.1d and 7.2c). This effect is not seen in the mean modulus controller as it sweeps a broad range of possible resonances (see fig. 7.4).

The mean modulus controller applies arbitrary scaling to the digitized signal, hence the bare comparison of the magnitudes of lock in and mean modulus outputs does not give usable information (see fig. 5.2). The comparison of the signal's SNRs (see fig. 7.5b) shows similar values for both detection methods. However, it has to be noted, that even if the lock in detector would result in more precise measurements, it certainly results in less correct measurements.

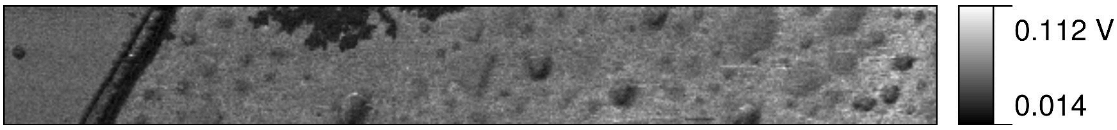
(a) Sample Topography



(b) PTIR Amplitude, 57.0 kHz



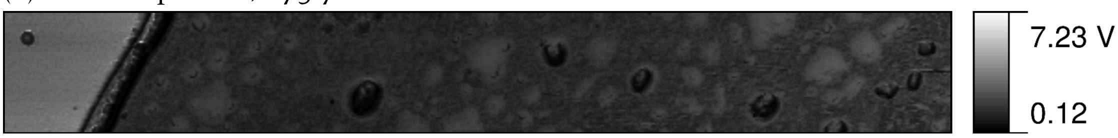
(c) PTIR Amplitude, 58.9 kHz



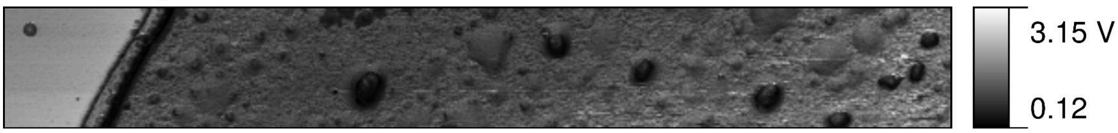
(d) PTIR Amplitude, 61.3 kHz

Figure 7.1.: PTIR image of a polystyrene film on gold using the first resonance recorded with a lock-in amplifier. The heigher region in a on the left side is the polystyrene film, the lower region is gold substrate.

(a) PTIR amplitude, 175.7 kHz



(b) PTIR amplitude, 179.5 kHz



(c) PTIR amplitude, 182.0 kHz

Figure 7.2.: PTIR signal of the second resonance recorded with a lock-in amplifier.

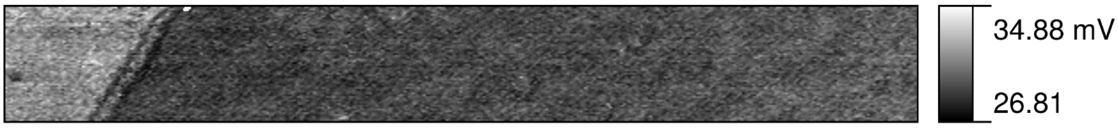


Figure 7.3.: Mean modulus controller amplitude of the polystyrene film (see fig. 7.1a). A frequency range of 50 kHz to 70 kHz was swept.

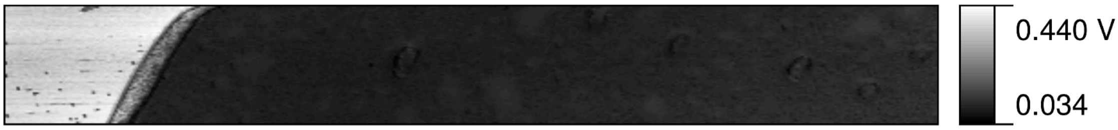
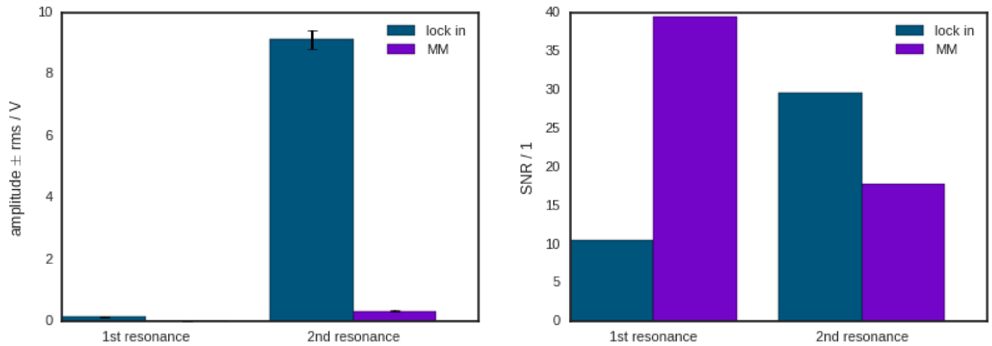


Figure 7.4.: Mean modulus controller amplitude of the polystyrene film (see fig. 7.1a). A frequency range of 160 kHz to 200 kHz was swept.



(a) Comparison of signal magnitudes. (b) Comparison of SNR.

Figure 7.5.: Comparison of signal magnitude and SNR of lock in and mean modulus controller for the film in figures 7.1 to 7.4

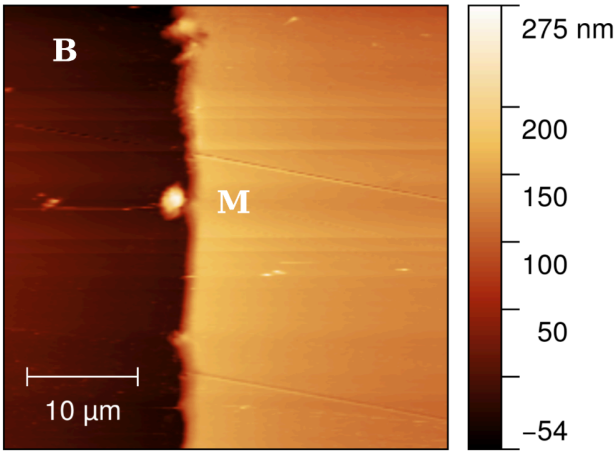
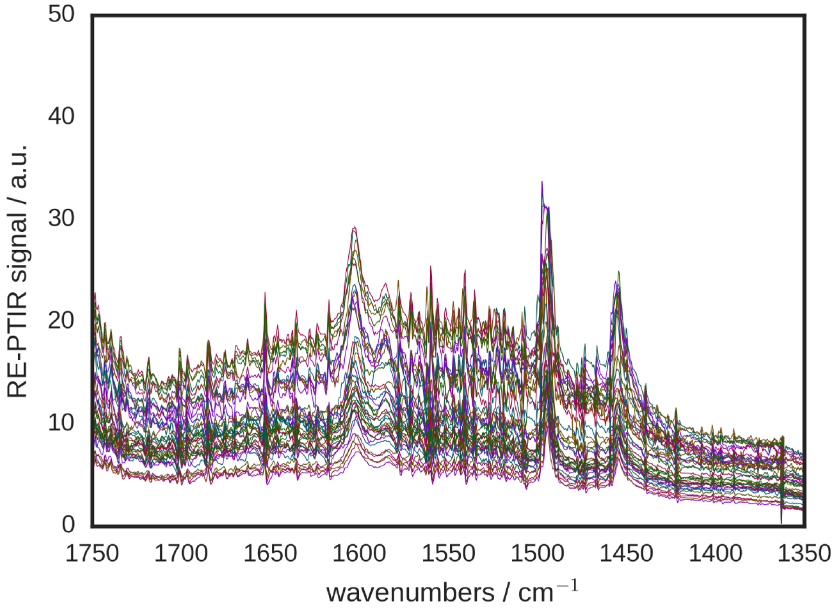


Figure 7.6.: Measurement positions on the polystyrene sample for comparison of lock-in and mean modulus controller. “B” marks the background position, “M” marks the measurement position.

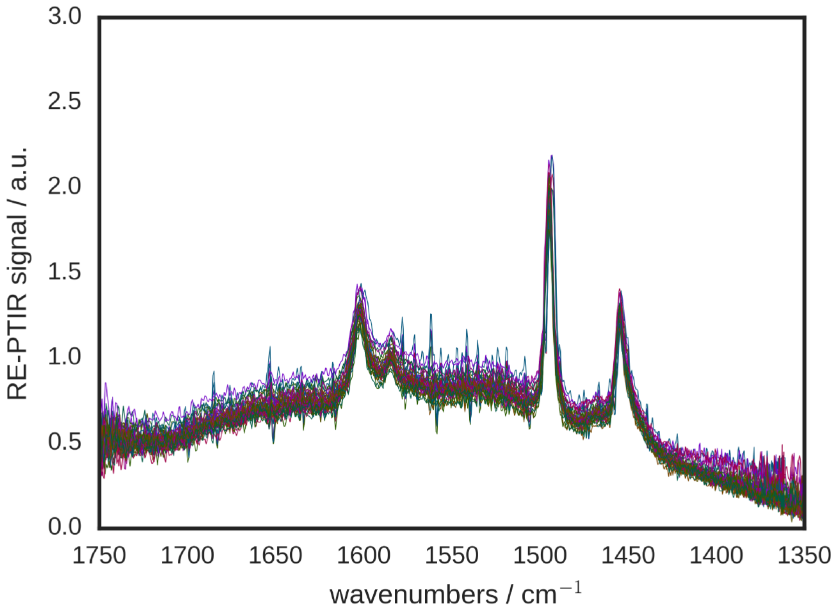
Spectroscopy

As the position of the cantilever does not change during spectroscopy the contact resonance should not change, either. Hence, mean modulus controller and lock-in amplifier should show similar performance. To test this idea a series of spectra was recorded with the mean modulus controller and the lock-in amplifier on the same spot of a polystyrene sample. The sample was 180 nm thick and was prepared on a PVD gold substrate. Spectra were background corrected with measurements taken on the gold substrate (see fig. 7.6 for measurement positions).

The mean modulus controller was set to sweep a range from 70 kHz to 130 kHz in 100 steps, waiting for 2 cycles at each step. The spectra were smoothed to a cut-off frequency of 1.5 cm^{-1} and a tenfold downsampling was performed to remove superfluous datapoints. The resulting spectra are depicted in figure 7.7.



(a) lock-in amplifier



(b) mean modulus controller

Figure 7.7.: Comparison of spectra of 180 nm polystyrene on gold taken at the same position as with the lock-in amplifier (a) and the mean modulus controller (b). (Spikes in the spectra stem from insufficient removal of water vapor.)

Table 7.1.: Comparison of relative standard deviations of the integrated bands of a polystyrene film (see fig. 7.8 for the distribution of the measurement).

band	relative standard deviation	
	lock-in	controller
1456 cm^{-1}	0.5421	0.0694
1500 cm^{-1}	0.4947	0.0502
1603 cm^{-1}	0.4823	0.0552

Looking at the distribution of integrals of three bands in the spectra (see fig. 7.8) it is clear, that the drift of the signal is far higher when measured with the lock-in amplifier than with the mean modulus controller. For all three bands the relative standard deviation of the integral was approximately 10 times higher in the lock-in amplifier measurement than in the mean modulus measurement (see table 7.1). We have to conclude that other factors, in addition to tip, sample and cantilever properties influence the location of the resonance maximum. One possible factor could be the equilibrium position of the cantilever relative to the sample surface. When the deflection read out drifts during the measurement the tip-sample force will change accordingly, leading to a change in the equilibrium position, which in turn influences the contact resonance frequency via $k^{\star}$.

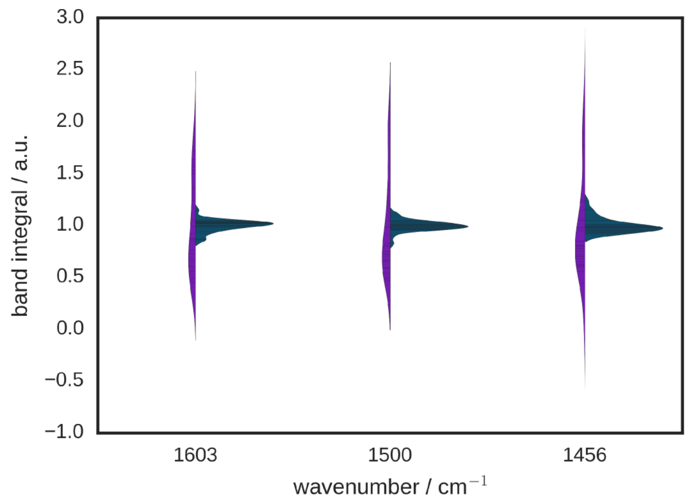


Figure 7.8.: Comparison of distribution of integrals of bands measured with the mean modulus controller and the lock-in amplifier. To make the distributions comparable, they were normalized to their mean value. Blue denotes data for the mean modulus controller, purple denotes data measured with the lock in amplifier.

8. Time-Resolved PTIR Spectroscopy

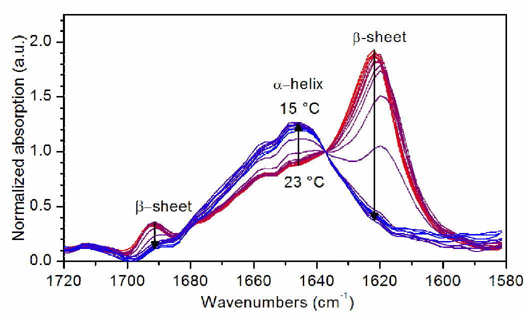
The following experiment is also described in publication I (see page 147ff.).

The secondary structure of a poly-peptide has a strong influence on its IR spectrum. For α-helix secondary structure the peptide amide I band has a single maximum at $\approx$1650 cm^{-1}. For β-sheet conformation the amide I band has two maxima, one at $\approx$1630 cm^{-1} and one at $\approx$1680 cm^{-1} [113]. For films of the polypeptide of poly-L-lysine the secondary structure changes in dependence on their hydration [114]. This fact was used to evaluate the possibility for time-resolved PTIR measurements. As the wanted information (i.e. secondary structure) is in this case not encoded in the absorption at a single wavenumber but in relative band positions, information at a broad range of wavenumbers is needed.

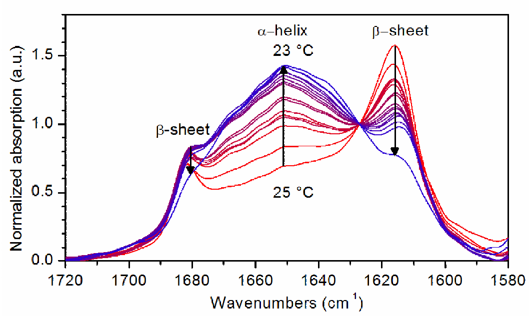
A poly-L-lysine film was prepared on a CaF_2 window. The window was placed on a heating stage in the PTIR setup. The setup was assembled as in section 5.1, with the addition of an air inlet that was used to fill the housing of the setup with air that had been bubbled through ^2H_2O. ^2H_2O was used instead of ^1H_2O in this experiment as the heavier hydrogen isotope in this molecule leads to a shift of the HOH deformation band to lower wavenumbers. For water this band is located at 1640 cm^{-1} where it overlaps with the polypeptide amide I band. For ^2H_2O the HOH deformation band is found at 1280 cm^{-1} [115].

For the measurements first the cantilever was placed on one spot of the poly-L-lysine film. Then the EC-QCL was tuned repeatedly across its tuning range and a temperature ramp was applied to the sample via the heating stage. Since changing the temperature of the sample also changed the mechanical properties of sample and cantilever the location of the cantilever resonance continuously changes as well. To compensate before each sweep a resonance curve around the expected resonance was recorded, then the current contact resonance was determined from this curve and the laser repetition rate adjusted to collect the spectrum at the frequency corresponding to the maximum amplitude in the sweep.

Using this method time resolved spectra could be collected of the secondary structure change in poly-peptide films (see fig. 8.1). A time resolution of 1.5 s was achieved, when no compensation for changes in the contact resonance



(a) FTIR transmission



(b) RE-PTIR

Figure 8.1.: Time resolved measurements of the secondary structure change of poly-L-lysine. Reprinted with permission from [104]. Copyright 2015 American Chemical Society.

.

frequency was performed. However, as the limiting factor in the time resolution was EC-QCL tuning speed, advances in laser technology will most likely lead to resolutions in the millisecond range. It is also possible to trade the width of the wavenumber range that measured for a better time resolution. If the time trace of the absorption at a single wavelength is sufficient a time resolution in the microsecond range could be achieved with current laser technology. In this case the measurement can be performed just as PTIR imaging currently is with the sole difference being that the cantilever is kept in place.

9. Concluding Remarks

9.1. Possible Further Improvements

While the PTIR setup is now in working order and can be routinely used for analyzing samples, there are still several possible improvements that could be added. A list of such improvements is given below.

9.1.1. Electronics

Lock-in Detection

The detection method used for this work does not block any frequency inside the pass band defined by the pre-amplifier. That means, that the oscillations at all multiples of the repetition rate f_{ref}, which are excited together with the wanted oscillation at f_{ref} can reach the output.

A lock-in amplifier based detector would not suffer from these problems as its pass band is defined around the reference frequency . While a full lock-in amplifier might lead to problems regarding space on the FPGA used for the controller, a lock-in amplifier based on rectangular reference signals would likely fit on the controller. The lock-in amplifier based on a rectangular reference signal does have the disadvantage of passing signals around f_{ref} as well as $(1 + 2N)f_{ref}$. However, as the amplitudes that are passed are scaled by a factor of $1/1+2N$ higher order modes can mostly be ignored. Furthermore, with the pass band of the pre-amplifier ending at 350 kHz higher order modes passed by the amplifier are only a problem below 117 kHz.

The pulse generator used in the mean modulus controller already has been programmed to output the needed reference signals. The first high pass filter could be removed, as the lock-in amplifier removes low-frequency components by itself. XILINX FIR_compiler FIR filters can be set to have parallel input values or pipe-lining the the filtering of multiple values. Some additional logic would be needed to convert the in-phase and quadrature components to a magnitude value. The space needed to perform this operation could be kept at a minimum by using a slow but small multiplier. Calculation of the square root can be implemented using a XILINX CORDIC IP Core. The magnitude can then be used instead of the low passed filtered mean modulus signal in the design.

Resonance Tracking

As the size of the swept range is proportional to the time needed to complete a sweep reducing the sweep range would increase the update rate of the output. This would allow either faster measurements or an increase in the SNR as several successive data points could be averaged. However, as the sweep range is decreased the chance that resonance shifts outside the range increases. One possible solution is to adjust the swept range to keep the resonance in its center. Several works [96, 94] describe adaptive methods for resonance tracking in AFM measurements. In the simplest case this could be sweeping across a defined range and moving the sweep range after each sweep so that the maximum frequency remains in the center. In a more sophisticated implementation, the sweep range could be adjusted by PI controller and some additional logic could be added to ensure that the center of the swept range is not moved outside a predefined range. While the excitation - detection - output part of the controller would not have to be changed to implement this method, the control entity would have to be modified. The sweep that is currently implemented would have to be replaced by a more complex function that awaits the output of the maximum detector at the end of each sweep and subsequently adapts the sweep range. The communication protocol between FPGA and PC would have to be adapted to allow entering sweep ranges, center frequencies and borders within which the center frequency is to be kept. More importantly, this modification would entail the need to characterize the setup thoroughly with regards to optimizing the parameters for this new controller mode. While the parameters for the currently used implementation without adjusted ranges can be eyeballed by looking at the resonance spectrum, sweep ranges, borders for the center frequency and possible PI parameters require a more detailed investigation for selecting optimum parameters.

Ring-down Measurement

While the power of EC-QCLs is lower than the power of ligh sources commonly used for ring-down PTIR measurements, such measurements can nevertheless be performed using EC-QCLs[1]. The implementation of ring-down measurements is straight forward: a pulse generator is used to trigger the start of both the laser

[1] from personal correspondence with Andrea Centrone

pulse and the data acquisition, then a set number of datapoints are acquired. FFT is applied to the data points to receive a frequency spectrum. From the resulting spectrum the maximum in a defined range is detected and output.

A preliminary version of such a ring-down controller was implemented on the controller before the dead-line for handing in this thesis but not tested thoroughly enough to warrant inclusion in this work. However, it should be noted, that due to the very flexible nature of the FPGA switching the controller from resonance enhanced detection to ring-down detection is a matter of seconds, mainly defined by the time it takes to send the ring-down design to the FPGA.

PCB Design

While the schematics printed in the appendix to this work correspond to the state of the controller as it currently is, as some parts had to be rewired repeatedly during the evaluation stage. Hence the original PCBs designs can not be used to replicate the controller.

If a copy of this controller is to be manufactured, the PCB has to be reworked to incorporate all changes. Of course, this could also be seen as an opportunity to exchange the FPGA for a larger model which would more make more complex evaluation possible. The pre-amplifier could be integrated together with the ADC onto one board and extended by a programmable gain amplifier (PGA) to reduce the need for the user to set DIP switches.

9.1.2. Optomechanics and AFM

Beam Expander

The beam expander was originally built into the setup to decrease the spot size for focusing with a fixed mirror. However, after the mirror had been replaced by the off-axis mirror, the expander was left in place without re-evaluating its benefit. As the expander incurs a decrease in total laser power on the sample of roughly 8.75 %, the need for it should be re-considered. Removing the expander would not only increase the laser power on the sample but also reduce the beam path and thereby reduce the water vapor absorption.

Automatic Stages

In the current state of the setup, positioning of the focal spot and the sample are performed manually. That means, that any adjustment of the focal spot also entails opening the housing. Every opening of the housing leads to an increase in water vapor concentration inside, incurring additional wait time until the setup can be used for highly sensitive measurements. If the stages for positioning the sample and the focus spot could be operated without opening the box then water vapor could only get into the enclosure during sample exchange. A motorized stage would also allow to readjust the position of the focal spot as needed during the measurement. This would make it possible to compensate pointing errors of the laser during the measurement.

Beam Profiling

In this work two techniques for determining the laser spot size were used:

1. a laser beam profiler camera with a pixel pitch of 125 µm

2. the translation stage used for positioning the laser focal spot in combination with the PTIR signal for measuring the signal intensity

Neither of these techniques allowed fast determination of the diameter of the focal spot in the sample plane. Furthermore, the resolution provided by the beam profiler camera was not sufficient to resolve the tight focal spot after the focusing optics had been improved. This means, that optimizing the focal spot of the setup was always a tedious affair of taking several measurements, comparing them and then further adjusting the setup.

A laser beam profiler with a higher resolution and a lower sensor set back of the sensor would have made adjustment of the spot size significantly faster and more efficient.

Sample Preparation

Fabrication of reference samples for PTIR measurements turned out to be a major challenge. While polymer films are easily fabricated using spin coating, referencing the measurement against the background signal is not trivial, as distance between the edge and the plateau of the film turned out to be too large to fit into a single AFM image. Without being able to measure the substrate and the sample in one AFM image, it is not possible to know the exact height of the film at the measurement position.

In this work, this problem was overcome by first spin coating the substrate and then removing a small strip of the film with a soft tip, such as a tooth pick. This generated sharp edges in the film. However, this method had several drawbacks. Often debris was left in place of the removed part of the film, that made AFM imaging challenging. The edge of film close to the scratch was sometimes lifted up from the surface. In such lifted areas hardly any PTIR signal could be detected. Finally, scratching the film off the sample also meant that generating any structure more complicated than an edge was just a matter of luck.

By using e-beam lithography 2.5 dimensional, nano-meter sized structures can be written into photo resist. The most commonly used photo resist for e-beam lithography, PMMA, has strong IR bands and can be easily detected in PTIR. Hence, using samples structures in this way would be advantageous for fabricating references for PTIR measurements.

9.2. Achievements

Before the start of this work, the Lendl group possessed ample experience in the use of EC-QCLs and in IR spectroscopy. AFM experience was available at the same institute, from Prof. Friedbacher but not in the Lendl group itself. At the inception of this work PTIR measurements using QCLs had been demonstrated by the Belkin group, but no commercial instrument was available, yet.

During this work the prior available knowledge in the characterization of EC-QCLs was extended to multi-chip lasers. The most important step in this improvement was to use a micro-controller to filter the TUNED signal of the MIRcat source so that only a single chip was characterized at once. Further improvements on the already available laser knowledge included the adaption of previously available data acquisition software to allow working with a multi-chip EC-QCL.

Two different types of focusing optics were implemented. Significant improvement in the spot size was achieved by using a rotatable off axis mirror. Additionally, a beam expander was added to the setup to improve the spot size.

Using an early stage of the PTIR setup and the EC-QCL knowledge of the Lendl group, it was possible to demonstrate time resolved PTIR measurements. These results were published in Analytical Chemistry (publication I) and presented at CEITEC 2014 (Brno, Czech Republic) and SCIX 2014 (Reno, Nevada). However, during the work on time resolved measurement, the influence of the contact resonance drift even when the cantilever was not moved across the sample became apparent. The remaining time of this work was spent on working on solving this contact resonance drift problem.

Control electronics to implement a scheme to track the contact resonance drift for PTIR measurements were designed, implemented and programmed. Development and programming turned out to be a major part of this work. A working prototype could be developed and implemented. Intermediate stages of the development were presented at VSS 2015 (Vienna, Austria), ICAVS 8 (Vienna, Austria) and SCIX 2015 (Providence, Rhodes Island).

The controller is built from well documented components to allow easy replication of the system by others. PC software for setting up measurements and adjusting the measurement parameters were developed.

Finally, using the working prototype, spectroscopy across all four QCLs of the MIRcat system was achieved. Through resonance tracking it was possible to significantly improve the stability of the system, decreasing the standard deviation of the band height by a factor of 10. Using this controller, it was possible to detect a 8 nm thin film of polystyrene on gold substrate in spectroscopy and imaging mode.

Bibliography

[1] Foil A Miller and George B. Kauffman. "C . V . Raman and the Discovery of the Raman Effect". In: *Journal of Chemical Education* 66.10 (1989), pp. 795–801.

[2] U Rabe, K Janser, and W Arnold. "Vibrations of free and surfacecoupled atomic force microscope cantilevers: theory and experiment". In: *Review of Scientific Instruments* 67.9 (1996), pp. 3281–3293. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=4994282.

[3] Peter R. Griffiths and James A. De Haseth. *Fourier Transform Infrared Spectrometry*. Hoboken, NJ: John Wiley & Sons, Inc, 2007. ISBN: 978-0-471-19404-0.

[4] Andreas Hugi, Richard Maulini, and Jérôme Faist. "External cavity quantum cascade laser". In: *Semiconductor Science and Technology* 25.8 (Aug. 2010), p. 083001. ISSN: 0268-1242. DOI: 10.1088/0268-1242/25/8/083001. URL: http://stacks.iop.org/0268-1242/25/i=8/a=083001?key=crossref.9d8c66b0f83eedb83e0bf4ae9240cf15.

[5] J. Faist et al. "Quantum cascade laser". In: *Science* 264.5158 (1994), p. 553. URL: http://www.sciencemag.org/content/264/5158/553.short.

[6] Matthew J Baker et al. "Using Fourier transform IR spectroscopy to analyze biological materials." In: *Nature protocols* 9.8 (Aug. 2014), pp. 1771–91. ISSN: 1750-2799. DOI: 10.1038/nprot.2014.110. URL: http://www.ncbi.nlm.nih.gov/pubmed/24992094.

[7] Susan Kaminskyj et al. "High spatial resolution analysis of fungal cell biochemistry–bridging the analytical gap using synchrotron FTIR spectromicroscopy." In: *FEMS microbiology letters* 284.1 (July 2008), pp. 1–8. ISSN: 0378-1097. DOI: 10.1111/j.1574-6968.2008.01162.x. URL: http://www.ncbi.nlm.nih.gov/pubmed/18422624.

[8] Daniel C Fernandez et al. "Infrared spectroscopic imaging for histopathologic recognition". In: *Nature Biotechnology* 23.4 (Apr. 2005), pp. 469–474. ISSN: 1087-0156. DOI: 10.1038/nbt1080. URL: http://www.ncbi.nlm.nih.gov/pubmed/15793574%20http://www.nature.com/nbt/journal/v23/n4/abs/nbt1080.html%20http://www.nature.com/doifinder/10.1038/nbt1080.

[9] CW Huck et al. "Advances of Infrared Spectroscopic Imaging and Mapping Technologies of Plant Material". In: *Planta Medica* 77.12 (2011), SL56. ISSN: 0032-0943. DOI: 10.1055/s-0031-1282179. URL: https://www.thieme-connect.com/ejournals/abstract/10.1055/s-0031-1282179.

[10] Tsoching Chen, Zachary D Schultz, and Ira W Levin. "Infrared spectroscopic imaging of latent fingerprints and associated forensic evidence." In: *The Analyst* 134.9 (Sept. 2009), pp. 1902–4. ISSN: 1364-5528. DOI: 10.1039/b908228j. URL: http://www.ncbi.nlm.nih.gov/pubmed/19684917.

[11] Pompi Hazarika et al. "Multiplexed detection of metabolites of narcotic drugs from a single latent fingermark". In: *Analytical chemistry* 82.22 (Nov. 2010), pp. 9150–4. ISSN: 1520-6882. DOI: 10.1021/ac1023205. URL: http://www.ncbi.nlm.nih.gov/pubmed/20968301.

[12] Edith Joseph et al. "Performance evaluation of mapping and linear imaging FTIR microspectroscopy for the characterisation of paint cross sections". In: *Analytical and Bioanalytical Chemistry* 396.2 (2010), pp. 899–910. ISSN: 16182642. DOI: 10.1007/s00216-009-3269-8.

[13] Adriana Rizzo. "Progress in the application of ATR-FTIR microscopy to the study of multi-layered cross-sections from works of art". In: *Analytical and Bioanalytical Chemistry* 392.1-2 (2008), pp. 47–55. ISSN: 16182642. DOI: 10.1007/s00216-008-2064-2.

[14] K L A Chan and S G Kazarian. "Fourier transform infrared imaging for high-throughput analysis of pharmaceutical formulations". In: *Journal of Combinatorial Chemistry* 7 (2005), pp. 185–189. URL: %7B%3C%7DGo%20to%20ISI%7B%3E%7D://000227708100005.

[15] Paul Bassan et al. "Large scale infrared imaging of tissue micro arrays (TMAs) using a tunable Quantum Cascade Laser (QCL) based microscope." In: *The Analyst* 139 (2014), pp. 3856–3859. ISSN: 1364-5528. DOI: 10.1039/c4an00638k. URL: http://pubs.rsc.org/en/content/articlehtml/2014/an/c4an00638k.

[16] Bahaa E. A. Saleh and Marvin Carl Teich. *Fundamentals of Photonics*. 2nd Editio. WILEY and Sons, 2007.

[17] John William Strutt (Third Lord of Rayleigh). "XXXI. Investigations in optics, with special reference to the spectroscope". In: *Philosophical Magazine and Journal of Science (fifth series)* 8.49 (Oct. 1879), pp. 261–274. ISSN: 1941-5982. DOI: 10.1080/14786447908639684. URL: http://www.tandfonline.com/doi/abs/10.1080/14786447908639684.

[18] Georg Ramer and Bernhard Lendl. "Attenuated Total Reflection Fourier Transform Infrared Spectroscopy". In: *Encyclopedia of Analytical Chemistry*. Chichester, UK: John Wiley & Sons, Ltd, Mar. 2013. ISBN: 9780470027318. DOI: 10.1002/9780470027318.a9287. URL: http://doi.wiley.com/10.1002/9780470027318.a9287.

[19] Peter Lasch and Dieter Naumann. "Spatial resolution in infrared microspectroscopic imaging of tissues." In: *Biochimica et biophysica acta* 1758.7 (July 2006), pp. 814–29. ISSN: 0006-3002. DOI: 10.1016/j.bbamem.2006.06.008. URL: http://www.ncbi.nlm.nih.gov/pubmed/16875659.

[20] Stefan W. Hell. "Far-Field Optical Nanoscopy". In: *Science* 316.5828 (May 2007), pp. 1153–1158. ISSN: 0036-8075. DOI: 10.1126/science.1137395. URL: http://www.sciencemag.org/cgi/doi/10.1126/science.1137395.

[21] Marc Richter and Volker Deckert. "Scanning Near-Field Optical Microscopy". In: *Surface and Thin Film Analysis*. Ed. by Gernot Friedbacher and Henning Bubert. Viley-VCH, 2011.

[22] G. Binnig, C. F. Quate, and Ch. Gerber. "Atomic Force Microscope". In: *Physical Review Letters* 56.9 (Mar. 1986), pp. 930–933. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.56.930. URL: http://link.aps.org/doi/10.1103/PhysRevLett.56.930.

[23] Gerd Binnig et al. *Surface Studies by Scanning Tunneling Microscopy*. 1982. DOI: 10.1103/PhysRevLett.49.57. URL: http://link.aps.org/doi/10.1103/PhysRevLett.49.57.

[24] J. Tersoff and D. R. Hamann. *Theory of the scanning tunneling microscope*. 1985. DOI: 10.1103/PhysRevB.31.805. URL: http://link.aps.org/doi/10.1103/PhysRevB.31.805.

[25] Gerhard Meyer and Nabil M. Amer. "Novel optical approach to atomic force microscopy". In: *Applied Physics Letters* 53.12 (1988), pp. 1045–1047. ISSN: 00036951. DOI: 10.1063/1.100061. URL: http://link.aip.org/link/APPLAB/v53/i12/p1045/s1%7B%5C&%7DAgg=doi$%5Cbackslash$nhttp://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=4857553.

[26] Nader Jalili and Karthik Laxminarayana. "A review of atomic force microscopy imaging systems: Application to molecular metrology and biological sciences". In: *Mechatronics* 14.8 (2004), pp. 907–945. ISSN: 09574158. DOI: 10.1016/j.mechatronics.2004.04.005.

[27] Gernot Friedbacher. "Atomic Force Microscopy". In: *Surface and Thin Film Analysis*. Ed. by Gernot Friedbacher and Henning Bubert. Wiley-VCH, 2011.

[28] David-A Mendels et al. "Dynamic properties of AFM cantilevers and the calibration of their spring constants". In: *Journal of Micromechanics and Microengineering* 16.8 (Aug. 2006), pp. 1720–1733. ISSN: 0960-1317. DOI: 10.1088/0960-1317/16/8/037. URL: http://stacks.iop.org/0960-1317/16/i=8/a=037?key=crossref.1b364320f52f71776fa319f096eb5cb1.

[29] U Rabe, J Turner, and W Arnold. "Analysis of the high-frequency response of atomic force microscope cantilevers". In: *Applied Physics A: Materials Science and Processing* 66.SUPPL. 1 (1998), pp. 277–282. ISSN: 09478396. DOI: 10.1007/s003390051145.

[30] Yaxin Song and Bharat Bhushan. "Simulation of dynamic modes of atomic force microscopy using a 3D finite element model". In: *Ultramicroscopy* 106.8-9 (June 2006), pp. 847–873. ISSN: 03043991. DOI: 10.1016/j.ultramic.2005.12.019. URL: http://linkinghub.elsevier.com/retrieve/pii/S030439910600057X.

[31] Gerhard Meyer and Nabil M. Amer. "Optical-beam-deflection atomic force microscopy: The NaCl (001) surface". In: *Applied Physics Letters* 56.21 (1990), p. 2100. ISSN: 00036951. DOI: 10.1063/1.102985. URL: http://scitation.aip.org/content/aip/journal/apl/56/21/10.1063/1.102985.

[32] Constant A J Putman et al. "A detailed analysis of the optical beam deflection technique for use in atomic force microscopy". In: *Journal of Applied Physics* 72.1 (1992), pp. 6–12. ISSN: 00218979. DOI: 10.1063/1.352149.

[33] Peter Eaton and Paul West. *Atomic Force Microscopy*. Oxford University Press, 2010.

[34] John Elle Sader. "Susceptibility of atomic force microscope cantilevers to lateral forces: Experimental verification". In: *Applied Physics Letters* 83.15 (2003), pp. 3195–3197. ISSN: 00036951. DOI: 10.1063/1.1616657.

[35] B.V Derjaguin, V.M Muller, and Yu.P Toporov. "Effect of contact deformations on the adhesion of particles". In: *Journal of Colloid and Interface Science* 53.2 (Nov. 1975), pp. 314–326. ISSN: 00219797. DOI: 10.1016/0021-9797(75)90018-1. URL: http://linkinghub.elsevier.com/retrieve/pii/0021979775900181.

[36] Yaxin Song and Bharat Bhushan. "Atomic force microscopy dynamic modes: modeling and applications". In: *Journal of Physics: Condensed Matter* 20.22 (June 2008), p. 225012. ISSN: 0953-8984. DOI: 10.1088/0953-8984/20/22/225012. URL: http://stacks.iop.org/0953-8984/20/i=22/a=225012?key=crossref.06b7fe3df375dd6668e93717fcc05b15.

[37] J Tamayo and R Garcia. "Deformation, Contact Time, and Phase Contrast in Tapping Mode Scanning Force Microscopy - Langmuir (ACS Publications)". In: *Langmuir* 7463.13 (1996), pp. 4430–4435. URL: http://pubs.acs.org/doi/abs/10.1021/la960189l.

[38] Feng Lu, Mingzhou Jin, and Mikhail A. Belkin. "Tip-enhanced infrared nanospectroscopy via molecular expansion force detection". In: *Nature Photonics* 8.4 (Jan. 2014), pp. 307–312. ISSN: 1749-4885. DOI: 10.1038/nphoton.2013.373. URL: http://www.nature.com/doifinder/10.1038/nphoton.2013.373.

[39] Alexandre Dazzi, F. Glotin, and R. Carminati. "Theory of infrared nanospectroscopy by photothermal induced resonance". In: *Journal of Applied Physics* 107.12 (2010), p. 124519. ISSN: 00218979. DOI: 10.1063/1.3429214. URL: http://link.aip.org/link/JAPIAU/v107/i12/p124519/s1%7B%5C&%7DAgg=doi.

[40] A Hammiche et al. "Photothermal FT-IR Spectroscopy : A Step Towards FT-IR Microscopy at a Resolution Better Than the Diffraction Limit". In: *Applied Spectroscopy* 53.7 (1999), pp. 810–815.

[41] A. Hammiche et al. "Progress in near-field photothermal infra-red microspectroscopy". In: *Journal of Microscopy* 213.2 (Jan. 2004), pp. 129–134. ISSN: 00222720. DOI: 10.1111/j.1365-2818.2004.01292.x. URL: http://doi.wiley.com/10.1111/j.1365-2818.2004.01292.x.

[42] L Bozec et al. "Localized photothermal infrared spectroscopy using a proximal probe". In: *Journal of Applied Physics* 90.10 (2001), p. 5159. ISSN: 00218979. DOI: 10.1063/1.1403671. URL: http://link.aip.org/link/JAPIAU/v90/i10/p5159/s1%7B%5C&%7DAgg=doi.

[43] Azzedine Hammiche et al. "Mid-infrared microspectroscopy of difficult samples using near-field photothermal microspectroscopy". In: *Spectroscopy (Amsterdam)* 19.2 (2004), pp. 20–+. ISSN: 0887-6703. URL: http://www.spectroscopymag.com/spectroscopy/data/articlestandard//spectroscopy/132005/152794/article.pdf.

[44] P. Ephrat et al. "Scanning near-field infrared microscopy based on tapered silver-halide probes". In: *Applied Physics Letters* 84.4 (2004), pp. 637–639. ISSN: 00036951. DOI: 10.1063/1.1644032.

[45] Jasbinder S. Sanghera et al. "Infrared Scanning Near-Field Optical Microscopy Below the Diffraction Limit". In: *IEEE Journal of Selected Topics in Quantum Electronics* 14.5 (2008), pp. 1343–1352. ISSN: 1077-260X. DOI: 10.1109/JSTQE.2008.928166. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4636777.

[46] E Betzig et al. "Near-field diffraction by a slit: implications for superresolution microscopy." In: *Applied optics* 25.12 (1986), p. 1890. ISSN: 0003-6935. DOI: 10.1364/AO.25.001890.

[47] Max Platkov et al. "Development of tapered silver-halide fiber tips for a scanning near-field microscope operating in the middle infrared". In: *Review of Scientific Instruments* 77.12 (2006). ISSN: 00346748. DOI: 10.1063/1.2403935.

[48] Nenad Ocelic, Andreas Huber, and Rainer Hillenbrand. "Pseudoheterodyne detection for background-free near-field spectroscopy". In: *Applied Physics Letters* 89.10 (2006), p. 101124. ISSN: 00036951. DOI: 10.1063/1.2348781. URL: http://link.aip.org/link/APPLAB/v89/i10/p101124/s1%7B%5C&%7DAgg=doi.

[49] F Huth et al. "Infrared-spectroscopic nanoimaging with a thermal source." In: *Nature materials* 10.5 (May 2011), pp. 352–6. ISSN: 1476-1122. DOI: 10.1038/nmat3006. URL: http://www.ncbi.nlm.nih.gov/pubmed/21499314.

[50] Florian Huth et al. "Nano-FTIR Absorption Spectroscopy of Molecular Fingerprints at 20 nm Spatial Resolution." In: *Nano letters* 12.8 (Aug. 2012), pp. 3973–8. ISSN: 1530-6992. DOI: 10.1021/nl301159v. URL: http://www.ncbi.nlm.nih.gov/pubmed/22703339.

[51] Iban Amenabar et al. "Structural analysis and mapping of individual protein complexes by infrared nanospectroscopy." In: *Nature communications* 4 (Jan. 2013), p. 2890. ISSN: 2041-1723. DOI: 10.1038/ncomms3890. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3863900%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract.

[52] Götz Wollny et al. "Nanoscale depth resolution in scanning near-field infrared microscopy." In: *Optics express* 16.10 (2008), pp. 7453–7459. ISSN: 1094-4087. DOI: 10.1364/OE.16.007453.

[53] Martin Wagner et al. "Ultrafast dynamics of surface plasmons in InAs by time-resolved infrared nanospectroscopy". In: *Nano Letters* 14 (2014), pp. 4529–4534. ISSN: 15306992. DOI: 10.1021/nl501558t.

[54] Martin Wagner et al. "Ultrafast and nanoscale plasmonic phenomena in exfoliated graphene revealed by infrared pump-probe nanoscopy". In: *Nano Letters* 14 (2014), pp. 894–900. ISSN: 15306984. DOI: 10.1021/nl4042577. arXiv: 1402.6003.

[55] Jianing Chen et al. *Optical nano-imaging of gate-tunable graphene plasmons*. 2012. DOI: 10.1038/nature11254. arXiv: 1202.4996.

[56] Z Fei et al. "Electronic and plasmonic phenomena at graphene grain boundaries." In: *Nature nanotechnology* 8.11 (Nov. 2013), pp. 821–5. ISSN: 1748-3395. DOI: 10.1038/nnano.2013.197. URL: http://www.ncbi.nlm.nih.gov/pubmed/24122082.

[57] Alexandre Dazzi et al. "Subwavelength infrared spectromicroscopy using an AFM as a local absorption sensor". In: *Infrared physics & technology* 49.1-2 (2006), pp. 113–121. URL: http://www.sciencedirect.com/science/article/pii/S1350449506000119.

[58] Alexandre Dazzi et al. "Chemical mapping of the distribution of viruses into infected bacteria with a photothermal method." In: *Ultramicroscopy* 108.7 (June 2008), pp. 635–41. ISSN: 0304-3991. DOI: 10.1016/j.ultramic.2007.10.008. URL: http://www.ncbi.nlm.nih.gov/pubmed/18037564.

[59] Andrea Centrone. "Infrared Imaging and Spectroscopy Beyond the Diffraction Limit." In: *Annual review of analytical chemistry (Palo Alto, Calif.)* 8.1 (2015), pp. 101–26. ISSN: 1936-1335. DOI: 10.1146/annurev-anchem-071114-040435. URL: http://www.ncbi.nlm.nih.gov/pubmed/26001952.

[60] Alexandre Dazzi et al. "AFMIR: Combining Atomic Force Microscopy and Infrared Spectroscopy for Nanoscale Chemical Characterization". In: *Applied Spectroscopy* 66.12 (2012), pp. 1366–1384.

[61] A. Dazzi et al. "Local infrared microspectroscopy with subwavelength spatial resolution with an atomic force microscope tip used as a photothermal sensor". In: *Optics Letters* 30.18 (Sept. 2005), p. 2388. ISSN: 0146-9592. DOI: 10.1364/OL.30.002388. URL: http://ol.osa.org/abstract.cfm?URI=ol-30-18-2388%20https://www.osapublishing.org/abstract.cfm?URI=ol-30-18-2388.

[62] Aaron M Katzenmeyer, Vladimir Aksyuk, and Andrea Centrone. "Nanoscale infrared spectroscopy: improving the spectral range of the photothermal induced resonance technique." In: *Analytical chemistry* 85.4 (Feb. 2013), pp. 1972–9. ISSN: 1520-6882. DOI: 10.1021/ac303620y. URL: http://www.ncbi.nlm.nih.gov/pubmed/23363013.

[63] Feng Lu and Mikhail A Belkin. "Infrared absorption nano-spectroscopy using sample photoexpansion induced by tunable quantum cascade lasers". In: *Optics Express* 19.21 (Sept. 2011), p. 19942. ISSN: 1094-4087. DOI: 10.1364/OE.19.019942. URL: http://www.ncbi.nlm.nih.gov/pubmed/21997003%20http://www.opticsinfobase.org/abstract.cfm?URI=oe-19-21-19942.

[64] F Lu and M. A. Belkin. "Quantum-cascade laser-based nanoscale photoexpansion micro-spectroscopy in mid-infrared and terahertz". In: *IRMMW-THz 2011 - 36th International Conference on Infrared, Millimeter, and Terahertz Waves*. 2011. ISBN: 9781457705090. DOI: 10.1109/irmmw-THz.2011.6104773. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6104773.

[65] Milan Milosevic. *Internal Reflection and ATR Spectroscopy*. John Wiley & Sons, 2012, p. 264. ISBN: 0470278323.

[66] Basudev Lahiri, Glenn Holland, and Andrea Centrone. "Chemical Imaging Beyond the Diffraction Limit: Experimental Validation of the PTIR Technique". In: *Small* 9.3 (Feb. 2013), pp. 439–445. ISSN: 16136810. DOI: 10.1002/smll.201200788. URL: http://www.ncbi.nlm.nih.

gov/pubmed/23034929%20http://doi.wiley.com/10.1002/smll.201200788.

[67] C Mayet et al. “Sub-100 nm IR spectromicroscopy of living cells.” In: *Optics letters* 33.14 (July 2008), pp. 1611–3. ISSN: 0146-9592. URL: http://www.ncbi.nlm.nih.gov/pubmed/18628814.

[68] Clotilde Policar et al. “Subcellular IR imaging of a metal-carbonyl moiety using photothermally induced resonance”. In: *Angewandte Chemie - International Edition* 50.4 (Jan. 2011), pp. 860–864. ISSN: 14337851. DOI: 10.1002/anie.201003161. URL: http://www.ncbi.nlm.nih.gov/pubmed/20941714.

[69] Celine Mayet et al. “In situ identification and imaging of bacterial polymer nanogranules by infrared nanospectroscopy.” In: *The Analyst* 135.10 (Oct. 2010), pp. 2540–5. ISSN: 1364-5528. DOI: 10.1039/c0an00290a. URL: http://www.ncbi.nlm.nih.gov/pubmed/20820491.

[70] Céline Mayet et al. “Analysis of bacterial polyhydroxybutyrate production by multimodal nanoimaging.” In: *Biotechnology advances* 31.3 (May 2013), pp. 369–74. ISSN: 1873-1899. DOI: 10.1016/j.biotechadv.2012.05.003. URL: http://dx.doi.org/10.1016/j.biotechadv.2012.05.003.

[71] Ariane Deniset-Besseau et al. “Monitoring TriAcylGlycerols Accumulation by Atomic Force Microscopy Based Infrared Spectroscopy in Streptomyces Species for Biodiesel Applications”. In: *The Journal of Physical Chemistry Letters* 5.4 (Feb. 2014), pp. 654–658. ISSN: 1948-7185. DOI: 10.1021/jz402393a. URL: http://pubs.acs.org/doi/abs/10.1021/jz402393a.

[72] L Baldassarre et al. “Mapping the amide I absorption in single bacteria and mammalian cells with resonant infrared nanospectroscopy”. In: *Nanotechnology* 27.7 (2016), p. 075101. ISSN: 0957-4484. DOI: 10.1088/0957-4484/27/7/075101. URL: http://stacks.iop.org/0957-4484/27/i=7/a=075101?key=crossref.a013af06961ac8c548768cdc3c25235e.

[73] Mauritz Kelchtermans et al. “Characterization of a polyethylenepolyamide multilayer film using nanoscale infrared spectroscopy and imaging”. In: *Vibrational Spectroscopy* 82 (2016), pp. 10–15. ISSN: 09242031. DOI: 10.1016/j.vibspec.2015.11.004. URL: http://linkinghub.elsevier.com/retrieve/pii/S0924203115300308.

[74] Curtis Marcott et al. "Spatial differentiation of sub-micrometer domains in a poly(hydroxyalkanoate) copolymer using instrumentation that combines atomic force microscopy (AFM) and infrared (IR) spectroscopy." In: *Applied spectroscopy* 65.10 (Oct. 2011), pp. 1145–50. ISSN: 1943-3530. DOI: 10.1366/11-06341. URL: http://www.ncbi.nlm.nih.gov/pubmed/21986074.

[75] Suzanne Morsch et al. "Insights into Epoxy Network Nanostructural Heterogeneity Using AFM-IR". In: *ACS Applied Materials & Interfaces* 8.1 (Jan. 2016), pp. 959–966. ISSN: 1944-8244. DOI: 10.1021/acsami.5b10767. URL: http://pubs.acs.org/doi/10.1021/acsami.5b10767%20http://pubs.acs.org/doi/abs/10.1021/acsami.5b10767.

[76] S. Morsch et al. "Mapping water uptake in an epoxy-phenolic coating". In: *Progress in Organic Coatings* 86 (2015), pp. 173–180. ISSN: 03009440. DOI: 10.1016/j.porgcoat.2015.05.017. URL: http://dx.doi.org/10.1039/C4FD00229F.

[77] Evelyn L Rosen et al. "Exceptionally mild reactive stripping of native ligands from nanocrystal surfaces by using Meerwein's salt." In: *Angewandte Chemie (International ed. in English)* 51.3 (Jan. 2012), pp. 684–689. ISSN: 1521-3773. DOI: 10.1002/anie.201105996. URL: http://www.ncbi.nlm.nih.gov/pubmed/22147424.

[78] Srabanti Ghosh et al. "PEDOT nanostructures synthesized in hexagonal mesophases". In: *New Journal of Chemistry* 38.3 (2014), p. 1106. ISSN: 1144-0546. DOI: 10.1039/c3nj01349a. URL: http://xlink.rsc.org/?DOI=c3nj01349a.

[79] Zhenpeng Cui et al. "Radiation-induced reduction-polymerization route for the synthesis of PEDOT conducting polymers". In: *Radiation Physics and Chemistry* 119 (2016), pp. 157–166. ISSN: 18790895. DOI: 10.1016/j.radphyschem.2015.10.011. URL: http://dx.doi.org/10.1016/j.radphyschem.2015.10.011.

[80] Aaron M. Katzenmeyer et al. "Assessing chemical heterogeneity at the nanoscale in mixed-ligand metal-organic frameworks with the PTIR technique". In: *Angewandte Chemie - International Edition* 53 (2014), pp. 2852–2856. ISSN: 14337851. DOI: 10.1002/anie.201309295.

[81] Aaron M. Katzenmeyer et al. "Nanoscale Imaging and Spectroscopy of Plasmonic Modes with the PTIR Technique". In: *Advanced Optical Materials* 2.8 (Aug. 2014), pp. 718–722. ISSN: 21951071. DOI: 10.1002/adom.201400005. URL: http://doi.wiley.com/10.1002/adom.201400005.

[82] Basudev Lahiri et al. "Nanoscale Imaging of Plasmonic Hot Spots and Dark Modes with the Photothermal-Induced Resonance Technique." In: *Nano letters* 13.7 (June 2013), pp. 3218–3224. ISSN: 1530-6992. DOI: 10.1021/nl401284m. URL: http://www.ncbi.nlm.nih.gov/pubmed/23777547.

[83] Jungseok Chae, Basudev Lahiri, and Andrea Centrone. "Engineering Near-Field SEIRA Enhancements in Plasmonic Resonators". In: *ACS Photonics* 22.iii (2015), acsphotonics.5b00466. ISSN: 2330-4022. DOI: 10.1021/acsphotonics.5b00466. URL: http://pubs.acs.org/doi/abs/10.1021/acsphotonics.5b00466%7B%5C#%7D.VoRv78J9zi4.mendeley.

[84] Alexandre Dazzi. "PhotoThermal Induced Resonance. Application to Infrared Spectromicroscopy". In: *Topics in Applied Physics: Thermal Nanosystems and Nanomaterials*. Ed. by Sebastian Volz. Topics in. Vol. 118. Springer, 2009. Chap. 16. ISBN: 9783642042577. DOI: 10.1007/978-3-642-04258-4.

[85] Hanna Cho et al. "Improved atomic force microscope infrared spectroscopy for rapid nanometer-scale chemical identification". In: *Nanotechnology* 24.44 (Nov. 2013), p. 444007. ISSN: 0957-4484. DOI: 10.1088/0957-4484/24/44/444007. URL: http://www.ncbi.nlm.nih.gov/pubmed/24113150%20http://stacks.iop.org/0957-4484/24/i=44/a=444007?key=crossref.078039c1748d078d2dec4e0bff1d2c29.

[86] Maximiliano Osvaldo Sonnaillon and Fabián Jose Bonetto. "A low-cost, high-performance, digital signal processor-based lock-in amplifier capable of measuring multiple frequency sweeps simultaneously". In: *Review of Scientific Instruments* 76.2 (2005), p. 024703. ISSN: 00346748. DOI: 10.1063/1.1854196. URL: http://scitation.aip.org/content/aip/journal/rsi/76/2/10.1063/1.1854196.

[87] Maximiliano Osvaldo Sonnaillon and Fabian Jose Bonetto. "Lock-in amplifier error prediction and correction in frequency sweep measurements". In: *Review of Scientific Instruments* 78.1 (2007). ISSN: 00346748. DOI: 10.1063/1.2428269.

[88] Stephen Jesse et al. "The Band Excitation Method in Scanning Probe Microscopy for Rapid Mapping of Energy Dissipation on the Nanoscale". In: *Nanotechnology* 18 (2007), p. 435503. ISSN: 0957-4484. DOI: 10.1088/0957-4484/18/43/435503. arXiv: 0708.4248. URL: http://arxiv.org/abs/0708.4248.

[89] a Gannepalli et al. "Mapping nanoscale elasticity and dissipation using dual frequency contact resonance AFM." In: *Nanotechnology* 22.35 (2011), p. 355705. ISSN: 0957-4484. DOI: 10.1088/0957-4484/22/35/355705.

[90] Brian J Rodriguez et al. "Dual-frequency resonance-tracking atomic force microscopy". In: *Nanotechnology* 18.47 (Nov. 2007), p. 475504. ISSN: 0957-4484. DOI: 10.1088/0957-4484/18/47/475504. URL: http://stacks.iop.org/0957-4484/18/i=47/a=475504?key=crossref.0dfdbac53357759ce5c78faaa9a2ee31.

[91] Kazushi Yamanaka et al. "Resonance frequency and Q factor mapping by ultrasonic atomic force microscopy". In: *Applied Physics Letters* 78.13 (2001), pp. 1939–1941. ISSN: 00036951. DOI: 10.1063/1.1357540.

[92] D C Hurley et al. "Nanoscale elastic-property measurements and mapping using atomic force acoustic microscopy methods". In: *Measurement Science and Technology* 16.11 (2005), pp. 2167–2172. ISSN: 0957-0233. DOI: 10.1088/0957-0233/16/11/006.

[93] Jason P Killgore and Donna C Hurley. "Low-force AFM nanomechanics with higher-eigenmode contact resonance spectroscopy". In: *Nanotechnology* 23.5 (2012), p. 055702. ISSN: 0957-4484. DOI: 10.1088/0957-4484/23/5/055702.

[94] A B Kos, J P Killgore, and D C Hurley. "SPRITE: a modern approach to scanning probe contact resonance imaging". In: *Measurement Science and Technology* 25.2 (Feb. 2014), p. 025405. ISSN: 0957-0233. DOI: 10.1088/0957-0233/25/2/025405. URL: http://stacks.iop.org/0957-0233/25/i=2/a=025405?key=crossref.dd11d5e29215cdff720d327ac2e3fa2c.

[95] C I Enriquez-Flores et al. "Fast frequency sweeping in resonance-tracking SPM for high-resolution AFAM and PFM imaging." In: *Nanotechnology* 23.49 (2012), p. 495705. ISSN: 1361-6528. DOI: 10.1088/0957-4484/23/49/495705. URL: http://www.ncbi.nlm.nih.gov/pubmed/23149480.

[96] a B Kos and D C Hurley. "Nanomechanical mapping with resonance tracking scanned probe microscope". In: *Measurement Science and Technology* 19.1 (2007), p. 015504. ISSN: 0957-0233. DOI: 10.1088/0957-0233/19/1/015504.

[97] M.P. Robinson and J. Clegg. "Improved Determination of<tex>Q</tex>-Factor and Resonant Frequency by a Quadratic Curve-Fitting Method". In: *IEEE Transactions on Electromagnetic Compatibility* 47.2 (May 2005), pp. 399–402. ISSN: 0018-9375. DOI: 10.1109/TEMC.2005.847411. arXiv:

arXiv:1011.1669v3. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1433067.

[98] Markus Brandstetter et al. "Tunable Mid-IR lasers: A new avenue to robust and versatile physical chemosensors". In: *Procedia Engineering* 5 (2010), pp. 1001–1004. ISSN: 18777058. DOI: 10.1016/j.proeng.2010.09.278. URL: http://www.sciencedirect.com/science/article/pii/S1877705810008258.

[99] Markus Brandstetter. "Quantitative chemical analysis of aqueous solutions using broadly tunable mid-infrared quantum cascade lasers". PhD thesis. TU Wien, 2014, XIII, 215 pg.

[100] Paul Bassan et al. "The inherent problem of transflection-mode infrared spectroscopic microscopy and the ramifications for biomedical single point and imaging applications." In: *The Analyst* 138.1 (Jan. 2013), pp. 144–57. ISSN: 1364-5528. DOI: 10.1039/c2an36090j. URL: http://www.ncbi.nlm.nih.gov/pubmed/23099638.

[101] Junzo Umemura. "ReflectionAbsorption Spectroscopy of Thin Films on Metallic Substrates". In: *Handbook of Vibrational Spectroscopy*. John Wiley & Sons, Ltd, 2006. ISBN: 9780470027325. DOI: 10.1002/0470027320.s2202. URL: http://dx.doi.org/10.1002/0470027320.s2202.

[102] L H Johnston. "Broadband polarization rotator for the infrared." In: *Applied optics* 16.4 (1977), pp. 1082–1084. ISSN: 0003-6935. DOI: 10.1364/AO.16.001082.

[103] Scott H. Brewer and Stefan Franzen. "Optical properties of indium tin oxide and fluorine-doped tin oxide surfaces: Correlation of reflectivity, skin depth, and plasmon frequency with conductivity". In: *Journal of Alloys and Compounds* 338.1-2 SPEC. (2002), pp. 73–79. ISSN: 09258388. DOI: 10.1016/S0925-8388(02)00217-7.

[104] Georg Ramer et al. "Method for Time-Resolved Monitoring of a Solid State Biological Film Using Photothermal Infrared Nanoscopy on the Example of Poly- l -lysine". In: *Analytical Chemistry* 87.8 (2015), pp. 4415–4420. ISSN: 0003-2700. DOI: 10.1021/acs.analchem.5b00241. URL: http://pubs.acs.org/doi/abs/10.1021/acs.analchem.5b00241.

[105] Jing Huang, Krishnan Padmanabhan, and Oliver M. Collins. "The sampling theorem with constant amplitude variable width pulses". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.6 (2011), pp. 1178–1190. ISSN: 15498328. DOI: 10.1109/TCSI.2010.2094350.

[106] Bryan Mealy and Fabrizio Tappero. *Free Range VHDL*. 2013, p. 200. URL: http://www.freerangefactory.org.

[107] Wade D. Peterson and Richard Herveille. *Wishbone B4 WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*. Tech. rep. 2010, p. 128.

[108] Xilinx Inc. *AXI Reference Guide*. 2012. URL: http://www.xilinx.com/support/documentation/ip%7B%5C_%7Ddocumentation/ug761%7B%5C_%7Daxi%7B%5C_%7Dreference%7B%5C_%7Dguide.pdf.

[109] *FT2232H: Dual High Speed USB to Multipurpose UART/FIFO IC*. 2012.

[110] United Kingdom. *Application Note AN_130 FT2232H Used In An FT245 Style Synchronous FIFO Mode*. 2010.

[111] Lior Kogut and Izhak Etsion. "Adhesion in elastic-plastic spherical microcontact". In: *Journal of Colloid and Interface Science* 261.2 (2003), pp. 372–378. ISSN: 00219797. DOI: 10.1016/S0021-9797(03)00071-7.

[112] K Feldman et al. "Toward a force spectroscopy of polymer surfaces". In: *Langmuir* 14.2 (1998), pp. 372–378. ISSN: 0743-7463. DOI: 10.1021/la9703353. URL: C:$%5Cbackslash$Dokumente%20und%20Einstellungen$%5Cbackslash$fery$%5Cbackslash$Eigene%20Dateien$%5Cbackslash$%7B%5C_%7DLesen$%5Cbackslash$feldman98.pdf.

[113] Andreas Barth. "Infrared spectroscopy of proteins." In: *Biochimica et Biophysica Acta* 1767.9 (Sept. 2007), pp. 1073–101. ISSN: 0006-3002. DOI: 10.1016/j.bbabio.2007.06.004. URL: http://www.ncbi.nlm.nih.gov/pubmed/17692815.

[114] S J Prestrelski et al. "Dehydration-induced conformational transitions in proteins and their inhibition by stabilizers." In: *Biophysical journal* 65.2 (Aug. 1993), pp. 661–71. ISSN: 0006-3495. DOI: 10.1016/S0006-3495(93)81120-2.

[115] John E. Bertie, M. Khalique Ahmed, and Hans H. Eysel. "Infrared Intensities of Liquids. 5. Optical and Dielectric Constants, Integrated Intensities, and Dipole Moment Derivatives of H2o and D2o at 22řC". In: *Journal Physical Chemistry* 93.16 (1989), pp. 2210–2218.

[116] Georg Ramer and Bernhard Lendl. "Nanoskopie im mittleren Infrarot". In: *Nachrichten aus der Chemie* 62.7-8 (July 2014), pp. 780–781. ISSN: 14399598. DOI: 10.1002/nadc.201490262. URL: http://doi.wiley.com/10.1002/nadc.201490262.

List of Figures

List of Tables

Part IV.

Publications

Publication I

Reprinted with permission from Georg Ramer et al. "Method for Time-Resolved Monitoring of a Solid State Biological Film Using Photothermal Infrared Nanoscopy on the Example of Poly- l -lysine". In: *Analytical Chemistry* 87.8 (2015), pp. 4415–4420. ISSN: 0003-2700. DOI: 10.1021/acs.analchem.5b00241. URL: http://pubs.acs.org/doi/abs/10.1021/acs.analchem.5b00241

analytical chemistry

Article

pubs.acs.org/ac

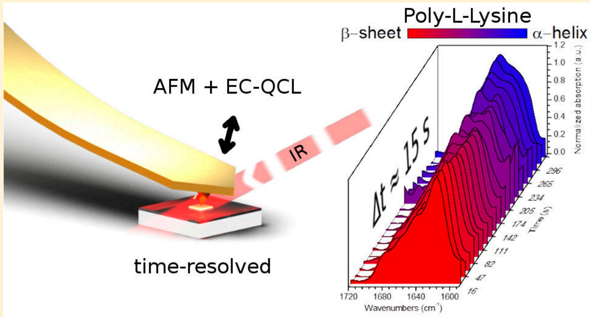
Method for Time-Resolved Monitoring of a Solid State Biological Film Using Photothermal Infrared Nanoscopy on the Example of Poly-L-lysine

Georg Ramer, Anna Balbekova, Andreas Schwaighofer, and Bernhard Lendl*

Vienna University of Technology, Institute for Chemical Technologies and Analytics, Getreidemarkt 9/164 UPA, 1060 Vienna, Austria

Supporting Information

ABSTRACT: We report time-resolved photothermal infrared nanoscopy measurements across a spectral range of more than 100 cm^{-1} (1565 cm^{-1} to 1729 cm^{-1}) at nanoscale spatial resolution. This is achieved through a custom-built system using broadly tunable external cavity quantum cascade lasers in combination with a commercially available atomic force microscope. The new system is applied to the analysis of conformational changes of a polypeptide (poly-L-lysine) film upon temperature-induced changes of the humidity in the film. Changes of the secondary structure from β-sheet to α-helix could be monitored at a time resolution of 15 s per spectrum. The time-resolved spectra are well comparable to reference measurements acquired with conventional Fourier transform infrared microscopy.



Infrared (IR) spectroscopy is an advantageous, nondestructive and label-free technique for chemical analysis. Fourier transform infrared (FTIR) spectroscopy is a commonly used method for the analysis of biological samples,[1] in particular for proteins and peptides.[2,3] Recently, IR microscopy showed great promise in the medical field for the analysis of human tissue and cell films.[1,4] However, the spatial resolution of IR optical microscopes is limited to the scale of several microns,[5] when using far-field techniques, where either the detector (aperture) or the light source is placed at distances of more than one wavelength from the sample. This practically precludes imaging of structures smaller than the employed wavelengths. That limitation can be overcome with near-field techniques—either detector (aperture) or light source placed at distances less than one wavelength from the sample—and the spatial resolution can be greatly enhanced.

Recent developments in the area of near-field imaging have made it possible to record IR spectra and images at a nanoscale spatial resolution. Currently, there are two well-established techniques allowing performance of such high-resolution measurements: one is a scattering scanning near-field technique based on pseudoheterodyne detection in the far-field (scattering scanning near-field optical microscopy, sSNOM);[6,7] the other uses thermal expansion of the sample upon illumination with pulsed IR light.[8,9] The latter method is often called AFMIR (atomic force microscope−infrared or induced resonance) or PTIR (photothermal induced resonance). Both methods have in common that an AFM cantilever is used to obtain high lateral spatial resolution. Both of them are capable of spatial resolution in IR imaging down to about 20 nm,[10,11] while still providing absorption spectra very similar to those acquired in conventional, far-field IR spectroscopy. The least sample amount detected for both methods is a single monolayer.[11,12] A main difference in performance of the near-field techniques is that sSNOM only detects analytes in the enhanced electromagnetic field around the cantilever tip, whereas AFMIR also perceives analytes at a distance of one micrometer or more beneath the sample surface.[13] PTIR has been applied for the analysis of micro-organisms, cells and other biological materials[14,15] and for the analysis of polymer films.[16] Other applications of PTIR include imaging of the modes of plasmonic resonators[17] and the analysis of micron-sized crystals.[18] sSNOM has been used for analyzing plasmonic modes in graphene[19−21] and in metallic infrared antennas.[22] It has been established as a technique for the analysis of polymer films[23] and is also able to perform spectroscopy on single protein complexes.[12]

In recent years imaging resolution and sensitivity have been improved at a rapid pace for sSNOM and PTIR; however, the majority of works addressed static systems only, i.e. systems that do not exhibit any induced changes during the measurement. Wagner et al.[21,24] demonstrated time-resolved sSNOM in repeatable events, i.e. events that can be repeated and reproduced many times without exchanging the sample. In

Received: January 19, 2015
Accepted: March 26, 2015
Published: March 26, 2015

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

these pump−probe experiments, time resolutions down to 200 fs via changing the delay line of the pump pulse were achieved at a spectral range of 400 cm^{-1}. Spectral resolution was obtained via an interferometer and Fourier transform. In order to be able to apply this technique, the observed event has to be repeatable in exactly the same way for each interferogram position at each time step.

For nonrepeatable events, time-resolved measurements of the local IR absorption at a single wavelength can be achieved with both currently commercially available sSNOM and PTIR instruments. For such measurements, the cantilever is kept in place while the sample is irradiated by IR light pulses. However, single wavelength information is often not sufficient for IR spectroscopy. A graphic example is the investigation of the secondary structure changes of proteins. Via exclusive monitoring of the amide I band at either 1650 cm^{-1} (α-helix) or 1633 cm^{-1} (β-sheet),[25] the change from α-helix to β-sheet is easily mistaken for a decrease or increase, respectively, in the total protein concentration. Monitoring of the entire amide I region provides a complete picture of the ongoing transformations of the protein—as will be demonstrated in this work.

Instead of collecting the absorbance at a single wavenumber, in this study we present the use of an external cavity quantum cascade laser (EC-QCL) in combination with a PTIR setup to measure the IR absorption across a wavenumber range of more than 100 cm^{-1}. This new method only needs a single repetition of the event to acquire time-resolved spectra. To achieve this objective, a Daylight solutions EC-QCL is used in its scan mode. Here the grating is swept in one continuous motion to cover the whole tuning range of the source in a short time. When using the scan mode, it is crucial to know the relation between the time that has passed since the start of the scan and the wavelength emitted by the laser. This relation was determined for the laser used in this work through characterization with a step-scan FTIR spectrometer.[26,27]

In this paper, we demonstrate the viability of our measurement scheme by monitoring the secondary structure change of a poly-L-lysine (PLL) thin film. This particular polypeptide was chosen, because its secondary structure changes depending on the hydration level of the film.[28] The conformational changes can therefore be introduced at comparatively soft conditions: at ambient pressure, without aggressive chemicals and at temperatures near room temperature. Since the secondary structure change is reversible, a single sample could be reused for several experiments. This, however, is not a necessity for the method introduced in this work. During experiments, the hydration was changed via the relative humidity above the film. The humidity was controlled by changing the temperature of the film in a humid environment. PLL is widely used as a model substance for proteins, and just as these more complex biomolecules, it exhibits amide bands in the IR region.[29] By monitoring the amide I band position, it is possible to determine the secondary structure (α-helix, β-sheet, or random coil).[25,29,30] Using time-resolved IR nanoscopy, the transition of the amide I band of one point on the PLL film can be followed during the change of its secondary structure.

■ EXPERIMENTAL SECTION

Sample Preparation. Poly-L-lysine hydrobromide (MW 15.000−30.000) was purchased from Sigma-Aldrich. PLL films were prepared by spin-coating. Prior to film deposition, a CaF_2 substrate (Sigma-Aldrich) was subjected to cleaning by means of subsequent 10 min rinses in acetone, ethanol, and distilled water ultrasonic baths. An aqueous solution of PLL with concentration 4.6% (w/v) was used for the spin-coating procedure. One drop (3 μL) of the PLL solution was casted at the spinning substrate with rotation speed 1500 rpm and rotation time of 1 min. The topography of the prepared film was characterized by AFM. For all dynamic measurements, the CaF_2 substrate was placed on a thermo electric heater/cooler which was connected to a temperature controller.

FTIR Spectroscopy. FTIR spectra were collected in transmission mode at 4 cm^{-1} resolution on a FTIR microscope (Hyperion 3000, BRUKER) with a liquid nitrogen cooled MCT detector. Every spectrum was averaged from 128 scans, with a sampling area of about 100 μm by 100 μm. Measurements were performed in homogeneous regions of the film with a thickness of about 200 nm. The absolute height of the film was obtained by introducing a scratch into the film down to the substrate and measuring the step height at the wall of the scratch with an AFM. Homogeneity of the film was checked by evaluation of the intensity of the amide I band. The absorption at arbitrary areas of measurement was nearly the same and the standard deviation was not larger than 7%.

The presence of water creates several artifacts in the IR spectra of proteins. The most prominent of these artifacts are water vapor bands and liquid water absorptions across the amide I band originating from a water HOH deformation band at ~1640 cm^{-1}.[3] Usually, the water vapor bands are reduced by evacuating the path of the IR beam or flushing the measurement chamber with dry air. However, our experiments require a humid environment to perform the secondary structure change of the PLL film. This problem was overcome by using D_2O instead of H_2O to create humid air. The DOD deformation band is located at lower wavenumbers than the HOH deformation band (1208 cm^{-1} as opposed to 1640 cm^{-1}),[31] allowing unperturbed protein spectroscopy in the amide I region.

Dynamic measurements were performed in a two-step process. First, the HD exchange was initiated by blowing dry air from an adsorption dryer (K-MT 2 MS/TE, AGRE, Austria) at a flow rate of 90 L/h through a washing flask with D_2O over the sample. Then, with the air still running, the secondary structure change was initiated by changing the substrate temperature.

PTIR Setup. As outlined above, the setup consists of an IR light source for excitation of thermal expansion of the sample and a scanning probe microscope for spatially resolved detection of the thermal expansion. In this work, a Daylight solution EC-QCL with a peak power of 800 mW and tuning range from 1565 to 1720 cm^{-1} was used as a light source in combination with an Agilent 5400 AFM with a MAC III controller as a detector. The manufacturer states a spatial drift in the range of 0.5 nm min^{-1} when the instrument is at thermal equilibrium. A sketch of the setup is depicted in Figure 1. The system was placed onto an air dampened vibration isolation table to reduce the mechanical noise in the system.

A reflective optic consisting of gold mirrors (Thorlabs) was used to direct the laser beam from the source to the sample below the AFM cantilever. In addition to plane mirrors for redirecting the beam, a beam expander was added to reduce the divergence of the beam. An off-axis parabolic mirror was used to focus the light onto the sample. The beam expander consists of one parabolic mirror with 2″ and one with 6″ reflected focal

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

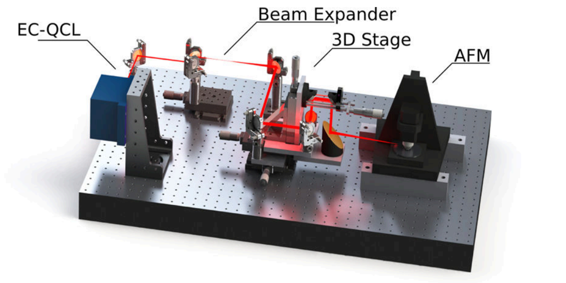


Figure 1. Sketch of the optical setup used for performing fast IR nanoscopy. The IR laser beam is shown in red. Not depicted: PE foil housing to allow controlling the atmosphere in the beam path and around the AFM (e.g., create water-vapor free, but D_2O-enriched environment).

length. A combination of three translation stages was used to move the position of the focal spot on the sample in three dimensions independently.

The AFM was placed slightly (50 mm) elevated from the optical table to allow for better optical access. Due to the small distance of about 2 mm between the sample and the AFM nose cone, the IR beam had to be directed onto the sample at an angle of incidence above 77°.

All IR nanoscopy measurements were performed using gold covered cantilevers HQ:CSC38/CR-AU (MikroMasch, US).

To reduce the influence of water vapor on the laser intensity, the setup was placed in a housing of polyethylene foil and constantly flushed with dry air. The sample was prepared as described above. Dry air humidified with deuterium oxide was blown across the sample as described for the FTIR experiments.

Fast Spectra Acquisition. For fast acquisition of the local IR absorption of the sample across the emission range of the EC-QCL, the sweep mode of the Daylight solution laser was used. In this mode, the grating of the laser is sweeped across a range of wavenumbers set at an approximately constant rate of wavenumbers per second. However, since the grating has to be accelerated at the starting wavenumber and decelerated at the stopping wavenumber, this rate is not constant across one sweep. To correct for deviations from the linear behavior, the laser sweep was characterized in step-scan measurement on an FTIR spectrometer (VERTEX 80v, BRUKER, Germany) as described elsewhere.[27,32]

The IR absorption was filtered from AFM deflection signal through the Agilent 5400 built-in lock-in amplifier. To trigger the emission of the EC-QCL at the frequency of the AFM 5400 lock-in amplifier the lock-in amplifier's sinusoidal reference signal had to be converted to a 5 V rectangular signal. The reference signal was set to an offset of 1.3 V and peak to peak amplitude of 2 V. A Schmidt trigger was used to convert the sinusoidal signal into rectangular pulses at the needed level which were fed to the TRIGGER input of the EC-QCL driver. The EC-QCL was operated in the external trigger mode in which the start of a pulse is determined by the positive edge on the TRIGGER input while the length of the pulse is set via the serial interface from a PC to the laser driver. These electronics suffice to perform single wavelength PTIR measurements and imaging.

However, to acquire local IR spectra the lock-in signal has to be recorded starting at the positive edge of the SCAN ENABLE pulse of the EC-QCL. Since external triggering of data collection is currently not possible with the Agilent hardware, the lock-in amplitude signal was instead output as an analog signal on one of the BNC connectors of the MAC III box and then digitized with a national instruments analog digital converter (ADC). The NI9401 digital I/O (National Instruments, US) was used to trigger the acquisition from the SCAN ENABLE signal. The lock-in filtered signal was output as an analog signal from the AFM controller and recorded using a NI9239 $50 kSs^{-1}$ADC (National Instruments, US). A sketch of all electronic connections between the individual parts of the used setup is depicted in Figure 2.

To speed up the acquisition of spectra and the retuning of the repetition rate, a Python script was implemented that automatically measured spectra at given points across the sample. The interface to the national instruments card was done via PyDAQmx library,[33] the Daylight laser was controlled

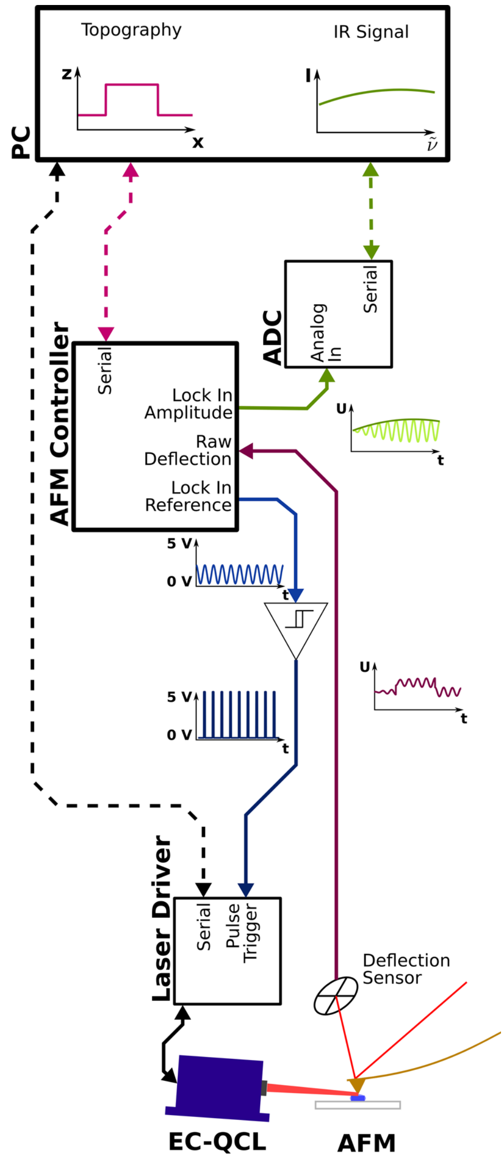


Figure 2. Sketch of the electronic connections in the setup used in this work. Solid lines are analog signals while dashed lines symbolize digital connections. The graphs in the sketch show the shape of the signals in the analog connections next to them (corresponding colors) and the final output displayed on the PC user interface.

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

via an in-house written Daylight laser controller which communicated with the laser via PyVISA. The AFM was controlled using the Python version of Picoscript provided by Agilent.

To remove high frequency modulations and to reduce noise in the raw data a Fourier transform based low-pass filter was applied to each collected sweep. To further improve the signal-to-noise ratio (SNR) several laser sweeps can be averaged at the cost of reducing time resolution.

Resonance Tracking. The signal in resonant PTIR spectroscopy strongly depends on the proximity of the repetition rate of the laser and the contact resonance, i.e. the mechanical resonance of the cantilever - sample system. The highest signal can be achieved, if the repetition rate of the EC-QCL is close to the resonance frequency of the system. In cases, when the repetition rate is far - in our system about 10 kHz or more - from the cantilever resonance, no photo-expansion signal can be detected at all. In systems with changing parameters (e.g., temperature) that lead to changing mechanical properties and resonance positions, the laser repetition rate has to be readjusted to the contact resonance in order to achieve the highest signal. In this system, the readjustment was achieved in a two step process. First, the amplitude of the cantilever was measured for a series of repetition rates across the frequency range wherein the resonance was expected to lie. From this measurement set, the repetition rate corresponding to the maximum amplitude was determined, which was then used for the laser scan. This procedure was performed prior to recording of each spectrum.

RESULTS AND DISCUSSION

In a preliminary examination of the dynamic behavior of the sample, far-field FTIR transmission measurements of the PLL film prepared as described above were performed in a dry air environment.

FTIR measurements of the PLL thin film revealed the well-known amide I band in the region between 1600 and 1700 cm^{-1} with the maximum at 1650 cm^{-1} (α-helix).[25,34] After adding D_2O vapor to the air blown across the sample, a small shift in the position of the maximum of the band toward lower wavenumbers was observed (see Figure 3). This shift of the amide I band detected after HD exchange is due to replacement of hydrogen atoms in the polypeptide film by heavier deuterium atoms. The amide I band of a protein or polypeptide after HD exchange is usually called amide I′.

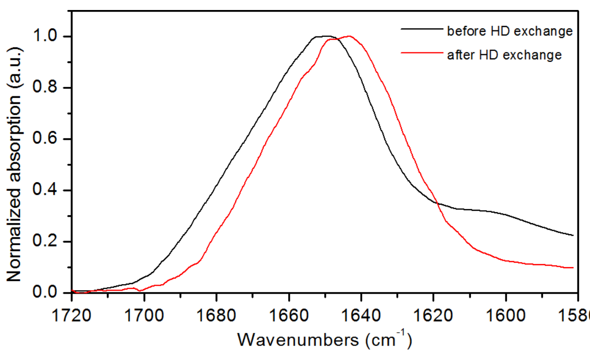


Figure 3. FTIR spectra of the PLL thin film deposited on CaF_2 substrate before and after HD exchange. Replacement of hydrogen by the heavier deuterium atoms leads to a shift of the amide I band to lower wavenumbers (amide I′).

The secondary structure change in the deuterated polypeptide was initiated by increasing the relative content of gaseous deuterium oxide in the PLL film (by increasing the degree of film hydration through decreasing the film temperature). As the (heavy) water content of the film is increased, the intramolecular hydrogen bonds in PLL are replaced by intermolecular hydrogen bonds between D_2O and PLL, which leads to a change of the PLL secondary structure from β-sheet to α-helix.[28] Figure 4 shows the time-resolved transfer from β-sheet to α-helix secondary conformers measured in the FTIR experiment.

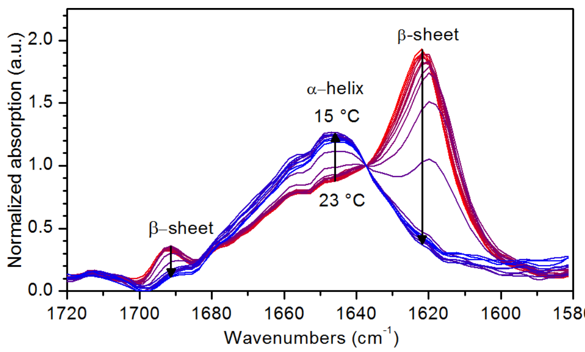


Figure 4. Time-resolved FTIR spectra of secondary structure change of the PLL thin film deposited on CaF_2 substrate during the HD exchange. When the temperature is decreased from 23 °C (red) to 15 °C (blue), the amide I′ bands corresponding to β-sheet secondary structure decrease and the one corresponding to the α-helix arises instead.

While the substrate was cooled down from 23 to 15 °C over the course of 22 min, the amide I′ bands at 1614 and 1690 cm^{-1}, corresponding to β-sheet secondary structure, gradually decreased, concurring with the increase of a single band at 1650 cm^{-1}, indicating the emergence of α-helix as the dominating secondary structure element.

The same transition was registered by PTIR nanoscopy: For this time-resolved near-field measurement, the PLL film sample was placed in the PTIR setup under a stream of dry air enriched with D_2O vapor. The secondary structure change from β-sheet to α-helix was again triggered by slowly decreasing the temperature of the sample from 25 to 23 °C over a period of 19 min. Time-resolved spectra of this transition are depicted in Figure 5.

During the temperature change the bands at ~1618 and 1680 cm^{-1} corresponding to the β-sheet decreased and the band at ~1650 cm^{-1} assigned to the α-helix increased, as was also observed in the far-field measurements (Figure 4). While the same spectral features arising from the change in the polypeptide were observed (shown in Figure 5), PTIR exhibited a nonstructured but curved baseline that is not present in the FTIR spectra. We attribute this feature to a direct excitation of the cantilever by the laser. This direct excitation is proportional to the laser intensity and is independent of the infrared spectrum of the sample. The curve in this contribution stems from the wavelength dependence of the emitted intensity of the EC-QCL.

A time resolution of 67 s per measurement (average of 3 consecutive sweeps of the laser) proved to be sufficiently fast to

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

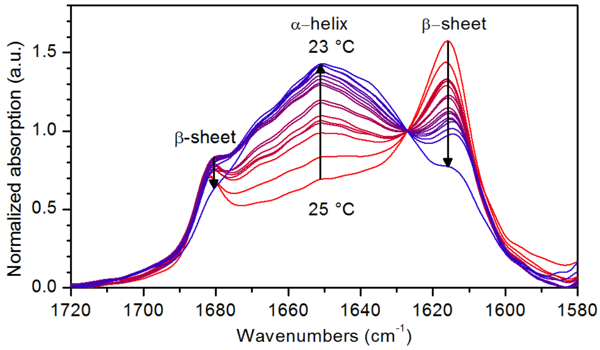


Figure 5. Time-resolved PTIR spectra of secondary structure change of the PLL thin film deposited on CaF_2. The full measurement series took 19 min. At the end of the series, the change from β-sheet to α-helix secondary structure is mostly completed.

accurately resolve the spectral change in the amide I′ band. However, 67 s per spectrum is not the limit of the temporal resolution of the instrument. In a further experiment, a steeper temperature ramp was used to change the conformation of the PLL thin film in a faster manner (see Figure 6). In this

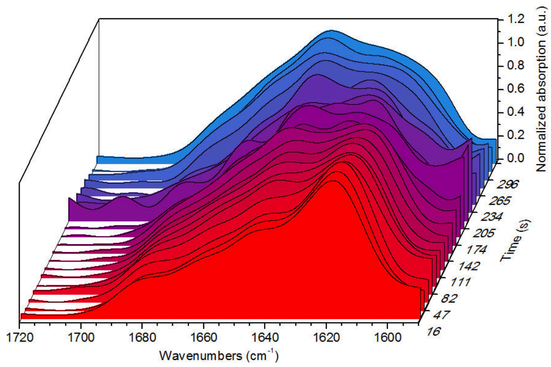


Figure 6. Time-resolved IR nanoscopy spectra of a PLL thin film taken every 15 s during a temperature ramp. The z-axis shows the time in seconds at which the measurement was taken. A change from β-sheet to α-helix secondary structure occurs within 300 s. At an acquisition rate of 15 s per spectrum, this change can be clearly detected.

measurement series, the time difference between two spectra was set to the current maximum acquisition speed of the instrument of 15 s per measurement. These faster changes in the spectra were well-resolved using this higher acquisition rate (see Figure 6). The laser sweep by itself is completed within 1.5 s.

The remaining time needed for a single measurement is spent on retuning the lock-in amplifier to the contact resonance of the system, as described in the Experimental Section. This step is necessary, because alterations of the temperature in the system lead to changes of the mechanical properties. Figure 7 shows the resulting shift of the cantilever contact resonance frequency at different temperatures.

In this measurement setup, where the alteration of the temperature is necessary to initiate a reaction, retuning the lock-in is indispensable. To achieve higher temporal resolution, a better retuning method could be used,[35] which would limit the time resolution to the time of a single laser scan. For samples that do not undergo changes in the mechanical

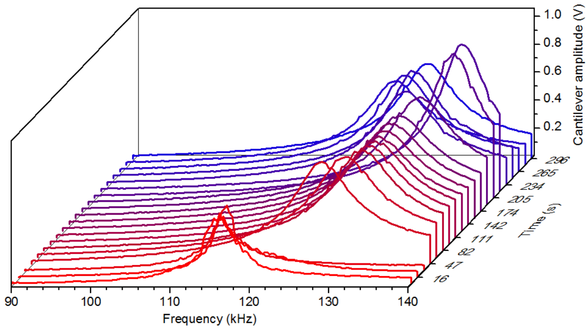


Figure 7. Lock-in amplitudes recorded during the tuning step of the measurement series in Figure 6. Differences in the resonance frequency make necessary retuning of the cantilever after each measurement.

properties during the reaction, retuning the lock-in amplifier is not necessary. In these cases we expect that an even better time resolution can be achieved with an approximate acquisition rate of one spectrum per 1.5 s with our setup. After removing this lock-in tuning procedure as a limiting factor, the laser scan time is a remaining limit on the time resolution. The laser scan time is a function of the EC-QCL scanning speed and the desired wavenumber range—which is often not open for discussion. Currently, commercially available EC-QCLs boast a scanning rate up to 1000 cm^{-1} s^{-1} which implies a scan time of the entire amide I region of about 100 ms. Of course, for the use of this fast tuning laser, other instrument parameters such as the frequency of the contact resonance and accordingly the laser repetition rate would also have to be adapted.

In addition to changes in the mechanical properties of the sample, the temperature change is also expected to lead to a spatial drift of the cantilever tip in excess of the rate of 0.5 nm min^{-1} given for a system in thermal equilibrium. This drift was not considered relevant for this work for two reasons: first because only a small temperature change was needed to perform the experiment and second because the film was assumed to be homogeneous. If a higher lateral stability of the cantilever is needed, a closed-loop scanner could be used instead the open-loop scanner used in this work.

■ CONCLUSION

PTIR is a promising technique for nondestructive chemical imaging of a multitude of samples. In this work, we demonstrated the feasibility of time-resolved PTIR measurements of nonrepeatable events at a wavenumber range of more than 100 cm^{-1}. Using a custom-made setup of commercially available parts, we followed the conformational change of PLL, initiated by changing the degree of hydration in the biological film. The spectra acquired with PTIR at a time resolution of 67 s per spectrum (3 scans) agree well with the ones taken with conventional FTIR microscopy. The maximum acquisition speed for this setup was determined to be 15 s per spectrum. A limiting factor is retuning of the lock-in amplifier. This procedure is necessary, because the temperature change used to induce the conformational change of the polypeptide alters the mechanical properties, causing the mechanical resonance to drift. It takes approximately 13.5 s to retune the lock-in. Without this tuning step, i.e. for measurements that do not involve changes in the sample temperature, acquisition speeds as high as 1.5 s per spectrum can be achieved.

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

In conclusion, the presented approach of time-resolved PTIR of nonrepeatable events at an expanded wavenumber range provides a missing piece in the toolbox of near-field imaging techniques, that complements the available subpicosecond time resolution achieved for repeatable events[21,24] and the well-established imaging of static samples. In the future, time-resolved IR nanoscopy may thus be of particular interest for a wide range of problems where chemical information is needed at a high spatial resolution as well as with temporal resolution. Potential systems of interest are the detection of chemical changes in a polymer during heat degradation,[16] the monitoring of hydration of polymer electrolyte membranes used for fuel cells[36] and analyzing chemical changes of wood under UV irradiation.[37]

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

Second derivative spectra of the first and last spectra in Figures 4 and 5 showing the band positions without background contributions. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: bernhard.lendl@tuwien.ac.at.

Author Contributions

G.R. designed and built the PTIR setup used in this work. A.B. performed FTIR and PTIR measurements and sample preparation. G.R. evaluated the measurements. The paper was written jointly by G.R. and A.B. A.S. supported and advised G.R. and A.B. in composing this paper. B.L. served as supervisor for G.R. and A.B. for all parts of the work. The manuscript was written through contributions of all authors. All authors have read the final version of the manuscript.

■ ACKNOWLEDGMENTS

The authors would like to thank Dieter Baurecht (University of Vienna) for initial help with use of PLL as a model substance for proteins. Ferry Kienberger and Christian Rankl of Keysight Laboratories, Linz, Austria (formerly Agilent Laboratories), provided the AFM used in this work and provided help and advice throughout the experiments. Johannes Frank (TU Vienna) fabricated mechanical parts for the setup. The thermocontroller was built by Wolfgang Tomischko (TU Vienna). The work was supported financially by the Austrian Science Fund (FFG) projects No. 478599 and 465271 as well as the Ph.D. school MEIBio at TU Vienna.

■ REFERENCES

(1) Baker, M. J.; Trevisan, J.; Bassan, P.; Bhargava, R.; Butler, H. J.; Dorling, K. M.; Fielden, P. R.; Fogarty, S. W.; Fullwood, N. J.; Heys, K. a; Hughes, C.; Lasch, P.; Martin-Hirsch, P. L.; Obinaju, B.; Sockalingum, G. D.; Sulé-Suso, J.; Strong, R. J.; Walsh, M. J.; Wood, B. R.; Gardner, P.; Martin, F. L. *Nat. Protoc.* **2014**, *9*, 1771−1791.

(2) Surewicz, W. K.; Mantsch, J. H. H.; Chapman, D. *Perspect. Biochem.* **1993**, *32*, 329−394.

(3) Kong, J.; Yu, S. *Acta Biochim. Biophys. Sin. (Shanghai)* **2007**, *39*, 549−559.

(4) Sahu, R. K.; Mordechai, S. *Future Oncol.* **2005**, *1*, 635−647.

(5) Lasch, P.; Naumann, D. *Biochim. Biophys. Acta* **2006**, *1758*, 814−829.

(6) Ocelic, N.; Huber, A.; Hillenbrand, R. *Appl. Phys. Lett.* **2006**, *89*, 101124.

(7) Brehm, M.; Taubner, T.; Hillenbrand, R.; Keilmann, F. *Nano Lett.* **2006**, *6*, 1307−1310.

(8) Dazzi, A.; Prazeres, R.; Glotin, F.; Ortega, J. M. *Opt. Lett.* **2005**, *30*, 2388.

(9) Dazzi, A.; Prater, C. B.; Hu, Q.; Chase, D. B.; Rabolt, J. F.; Marcott, C. *Appl. Spectrosc.* **2012**, *66*, 1366−1384.

(10) Huth, F.; Govyadinov, A.; Amarie, S.; Nuansing, W.; Keilmann, F.; Hillenbrand, R. *Nano Lett.* **2012**, *12*, 3973−3978.

(11) Lu, F.; Jin, M.; Belkin, M. A. *Nat. Photonics* **2014**, *8*, 307−312.

(12) Amenabar, I.; Poly, S.; Nuansing, W.; Hubrich, E. H.; Govyadinov, A. a; Huth, F.; Krutokhvostov, R.; Zhang, L.; Knez, M.; Heberle, J.; Bittner, A. M.; Hillenbrand, R. *Nat. Commun.* **2013**, *4*, 2890.

(13) Lahiri, B.; Holland, G.; Centrone, A. *Small* **2013**, *9*, 488.

(14) Deniset-Besseau, A.; Prater, C. B.; Virolle, M.-J.; Dazzi, A. *J. Phys. Chem. Lett.* **2014**, *5*, 654−658.

(15) Policar, C.; Waern, J. B.; Plamont, M. A.; Clède, S.; Mayet, C.; Prazeres, R.; Ortega, J. M.; Vessières, A.; Dazzi, A. *Angew. Chem., Int. Ed.* **2011**, *50*, 860−864.

(16) Marcott, C.; Lo, M.; Kjoller, K.; Prater, C.; Noda, I. *Appl. Spectrosc.* **2011**, *65*, 1145−1150.

(17) Katzenmeyer, A. M.; Chae, J.; Kasica, R.; Holland, G.; Lahiri, B.; Centrone, A. *Adv. Opt. Mater.* **2014**, *2* (8), 718−722 DOI: 10.1002/adom.201400005.

(18) Katzenmeyer, A. M.; Canivet, J.; Holland, G.; Farrusseng, D.; Centrone, A. *Angew. Chem., Int. Ed.* **2014**, *53*, 2852−2856.

(19) Chen, J.; Badioli, M.; Alonso-González, P.; Thongrattanasiri, S.; Huth, F.; Osmond, J.; Spasenović, M.; Centeno, A.; Pesquera, A.; Godignon, P.; Zurutuza Elorza, A.; Camara, N.; de Abajo, F. J. G.; Hillenbrand, R.; Koppens, F. H. L. *Nature* **2012**, 3−7.

(20) Fei, Z.; Rodin, a. S.; Andreev, G. O.; Bao, W.; McLeod, a. S.; Wagner, M.; Zhang, L. M.; Zhao, Z.; Thiemens, M.; Dominguez, G.; Fogler, M. M.; Neto, a. H. C.; Lau, C. N.; Keilmann, F.; Basov, D. N. *Nature* **2012**, *487*, 82−85.

(21) Wagner, M.; Fei, Z.; McLeod, A. S.; Rodin, A. S.; Bao, W.; Iwinski, E. G.; Zhao, Z.; Goldflam, M.; Liu, M.; Dominguez, G.; Thiemens, M.; Fogler, M. M.; Castro Neto, A. H.; Lau, C. N.; Amarie, S.; Keilmann, F.; Basov, D. N. *Nano Lett.* **2014**, *14*, 894−900.

(22) Schnell, M.; Garcia-Etxarri, A.; Huber, A. J.; Crozier, K. B.; Borisov, A.; Aizpurua, J.; Hillenbrand, R. *J. Phys. Chem. C* **2010**, *114*, 7341−7345.

(23) Taubner, T.; Hillenbrand, R.; Keilmann, F. *Appl. Phys. Lett.* **2004**, *85*, 5064−5066.

(24) Wagner, M.; McLeod, A. S.; Maddox, S. J.; Fei, Z.; Liu, M.; Averitt, R. D.; Fogler, M. M.; Bank, S. R.; Keilmann, F.; Basov, D. N. *Nano Lett.* **2014**, *14*, 4529−4534.

(25) Barth, A. *Biochim. Biophys. Acta* **2007**, *1767*, 1073−1101.

(26) Brandstetter, M.; Genner, A.; Anic, K.; Lendl, B. *Analyst* **2010**, *135*, 3260−3265.

(27) Ramer, G.; Kasberger, J.; Brandstetter, M.; Saeed, A.; Jakoby, B.; Lendl, B. *Appl. Phys. B: Laser Opt.* **2013**, *116*, 325−332.

(28) Prestrelski, S. J.; Tedeschi, N.; Arakawa, T.; Carpenter, J. F. *Biophys. J.* **1993**, *65*, 661−671.

(29) Szyc, Ł.; Pilorz, S.; Czarnik-Matusewicz, B. *J. Mol. Liq.* **2008**, *141*, 155−159.

(30) Kong, J.; Yu, S. *Biochemistry* **2007**, *39*, 549−559.

(31) Bertie, J. E.; Ahmed, M. K.; Eysel, H. H. *J. Phys. Chem.* **1989**, *93*, 2210−2218.

(32) Brandstetter, M.; Lendl, B. *Sensors Actuators B Chem.* **2012**, *170*, 189−195.

(33) Cladé, P. PyDAQmx: a Python interface to the National Instruments DAQmx driver. http://pythonhosted.org/PyDAQmx/.

(34) Dzwolak, W.; Smirnovas, V. *Biophys. Chem.* **2005**, *115*, 49−54.

(35) Sonnaillon, M. O.; Bonetto, F. J. *Rev. Sci. Instrum.* **2007**, *78*, 014701.

(36) Kunimatsu, K.; Bae, B.; Miyatake, K.; Uchida, H.; Watanabe, M. *J. Phys. Chem. B* **2011**, *115*, 4315−4321.

(37) Müller, U.; Rätzsch, M.; Schwanninger, M.; Steiner, M.; Zöbl, H. *J. Photochem. Photobiol. B Biol.* **2003**, *69*, 97−105.

DOI: 10.1021/acs.analchem.5b00241
Anal. Chem. 2015, 87, 4415−4420

Publication II

Georg Ramer and Bernhard Lendl. “Nanoskopie im mittleren Infrarot”. In: *Nachrichten aus der Chemie* 62.7-8 (July 2014), pp. 780–781. ISSN: 14399598. DOI: 10.1002/nadc.201490262. URL: http://doi.wiley.com/10.1002/nadc.201490262

Nanoskopie im mittleren Infrarot

Georg Ramer, Bernhard Lendl

Mit neuen Nahfeldtechniken umgeht die IR-Mikroskopie das Beugungslimit und erreicht eine Ortsauflösung zwischen 20 und 50 nm. Dabei hat die Rasterkraftmikroskopie eine entscheidende Funktion.

● Die klassische FTIR-Mikroskopie und die daraus abgeleiteten Techniken der bildgebenden Analyse (FTIR-imaging) arbeiten im Fernfeld. Deshalb ist ihre räumliche Auflösung beugungslimitiert. Strukturen unterhalb der Wellenlänge des eingesetzten Lichts können sie also nicht auflösen, im mittleren Infrarot beschränkt sich die erzielbare räumliche Auflösung auf wenige Mikrometer.

Eine Möglichkeit dieses Beugungslimit zu umgehen, sind Nahfeldtechniken. Im sichtbaren Wellenlängenbereich funktioniert dies schon länger, etwa mit faserbasierter, nahfeldoptischer Rastermikroskopietechnik (SNOM). Eine analoge Geräteentwicklung im mittleren Infrarot ist jedoch aufgrund technischer und physikalischer Beschränkungen bis heute nicht durchführbar.

Ansätze für höhere Ortsauflösung

● In den letzten Jahren wurden zwei alternative, von ihrem Ansatz her jedoch grundverschiedene Nahfeldtechniken entwickelt, mit denen heute Infrarotnanoskopie mit einer Ortsauflösung in der Größenordnung von 20 bis 50 nm machbar ist. Es handelt sich hierbei zum einen um die von Dazzi et al. erstmals gezeigte photothermische Infrarotnanoskopie (PTIR oder AFMIR)[1] und zum anderen um die von Hillenbrand et al. entwickelte pseudoheterodyne Methode zur Detektion der Streuung des Infrarotstrahls am Cantilever (Hebel) eines Rasterkraftmikroskops (s-SNOM).[2]

Bei der PTIR-Technik (Abbildung 1) beleuchten periodische Infrarotlaserpulse die Probe in einem Rasterkraftmikroskop (AFM). Wenn ein Teil der Probe Licht absorbiert, erwärmt sich dieser Probenausschnitt periodisch und dehnt sich dabei aus. Diese Ausdehnung nimmt die Spitze des AFM-Cantilevers auf, sodass das AFM die Ausdehnung im Nahfeld detektiert.

Für s-SNOM (Abbildung 2) wird der AFM-Cantilever über der Probe zum Schwingen gebracht und die an der AFM-Spitze gestreute Strahlung im Fernfeld detektiert. Durch Modulierung des eingestrahlten Infrarotlichts in einem Interferometer kann die lokale Absorption störungsfrei bei einer Modulationsfrequenz, die einer Kombination aus Cantileverschwingung und Interferometerschwingung entspricht, detektiert werden (pseudoheterodyne Detektion).

Bei beiden Methoden definiert nicht der Fokus des Lasterstrahls auf der Probe die Ortsauflösung, sondern der Radius der eingesetzten AFM-Spitze.

Die Techniken haben Hersteller bereits in Spektrometer umgesetzt:

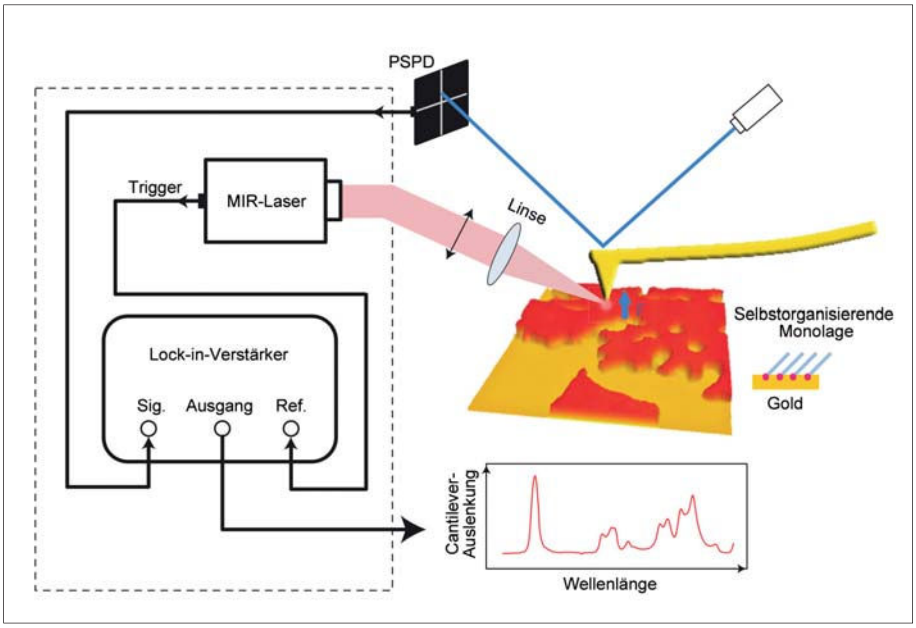


Abb. 1. Photothermische Infrarotnanoskopie (PTIR), Messaufbau, hier verwendet zur Messung einer Monolage.[12]
PSPD: positionsempfindliche Photodiode (position sensitive photodiode)

Geräte mit PTIR gibt es von Anasys Instruments[3)] und mit s-SNOM von neaspec[4)].

Anwendungen

Anwendungen für die Nahfeldtechniken finden sich in der traditionellen Infrarotanalytik, etwa der Polymeranalytik[5)] und der pharmazeutischen Forschung.[6)] Die beiden Nanoskopiemethoden machen aber auch zahlreiche neue Anwendungsgebiete erstmals dem Infrarotimaging zugänglich. So gewannen mehrere Arbeitsgruppen Infrarotspektren von Einzelzellen mit subzellulärer Auflösung und bestimmten so zum Beispiel die Verteilung eines Chemotherapeutikums im Inneren einer einzelnen Krebszelle, die Anwesenheit und Position von Viren in *E. Coli* oder die Größe und Form von Polyhydroxybuttersäure-Vesikeln in *Rhodobacter capsulatus*.[7,8)] Die Methode eignet sich auch dazu, lebende Organismen zu untersuchen: Mayet et al. spektroskopierten mit hoher Ortsauflösung lebende Hyphen, das sind fadenförmige Zellen, des Hefepilzes *Candica albicans*.[9)]

Lahiri et al.[10)] verwendeten PTIR, um die Bildung von Hotspots im elektromagnetischen Feld um asymmetrische Ringresonatoren aus Gold zu untersuchen. Im Gegensatz zu herkömmlichen IR-Methoden, die nur das Fernfeld der Resonatoren vermessen können, detektiert PTIR die Intensität des elektromagnetischen Feldes um den Resonator. Aufgrund der an Hotspots stark erhöhten Intensität des elektromagnetischen Feldes können Infrarot und Ramansignale um mehrere Größenordnungen verstärkt werden. Dies lässt sich zum Design hochempfindlicher molekülspezifischer Sensoren nutzen.

Chen et al.[11)] analysierten mit s-SNOM-Infrarotnanoskopie die Ausbreitung von Plasmonen in Graphen und wiesen nach, dass schon Unterbrechungen von wenigen Nanometern Breite im Graphen – mehrere Größenordnungen unter der Wellenlänge der Plasmonen – genügen, um Plasmonen effizient zu reflektieren. Die Forscher schlagen vor, dies zur Konstruktion von plasmonischen Schaltkreisen zu verwenden.

Beide Techniken sind empfindlich genug, um Monolagen zu detektieren.[12–14)] Die Reife der Messgeräte und Anwendungsbeispiele aus Biologie, Medizin, Pharmazie, Physik und Materialwissenschaften zeigen, dass mit der Infrarotnanoskopie heute eine weitere experimentelle Technik zur Verfügung steht, um nanostrukturierte Proben direkt, zerstörungs- und markierungsfrei sowie bildgebend zu untersuchen.

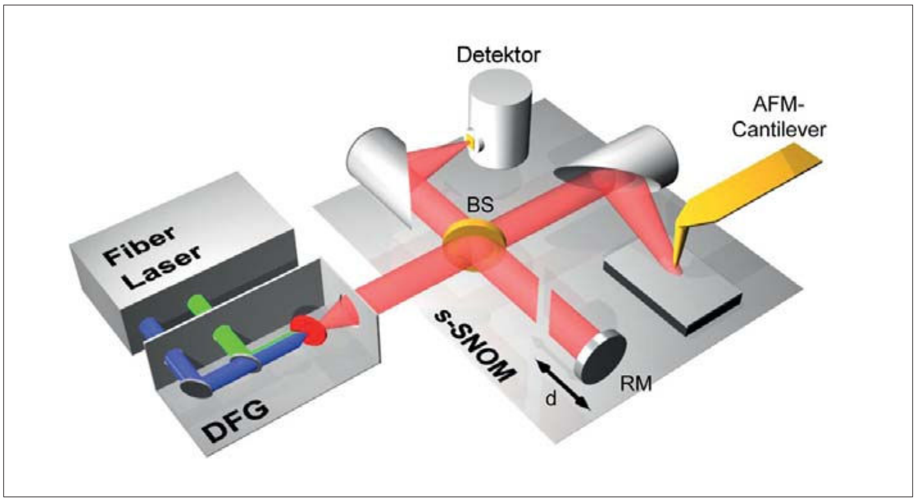


Abb. 2. Messaufbau der Infrarotnanoskopie über die Streuung des Infrarotstrahls am AFM-Cantilever (s-SNOM).[15)]
BS: Strahlteiler (beam splitter), RM: Referenzspiegel (reference monitor)

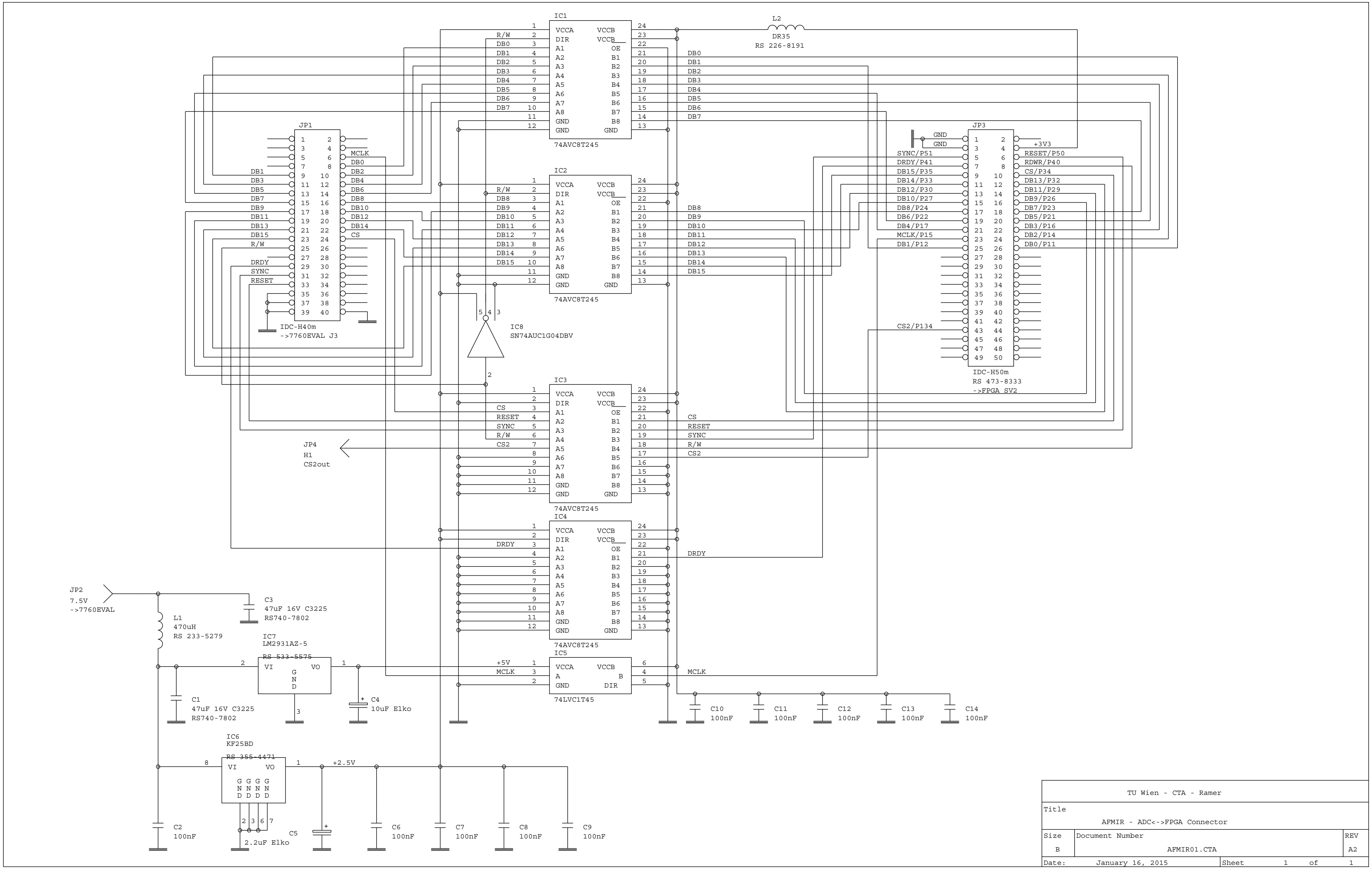
Georg Ramer ist wissenschaftlicher Mitarbeiter der Arbeitsgruppe Prozessanalytik und Schwingungsspektroskopie an der TU Wien. **Bernhard Lendl** ist Professor für analytische Chemie an der TU Wien und leitet die Arbeitsgruppe. bernhard.lendl@tuwien.a.at

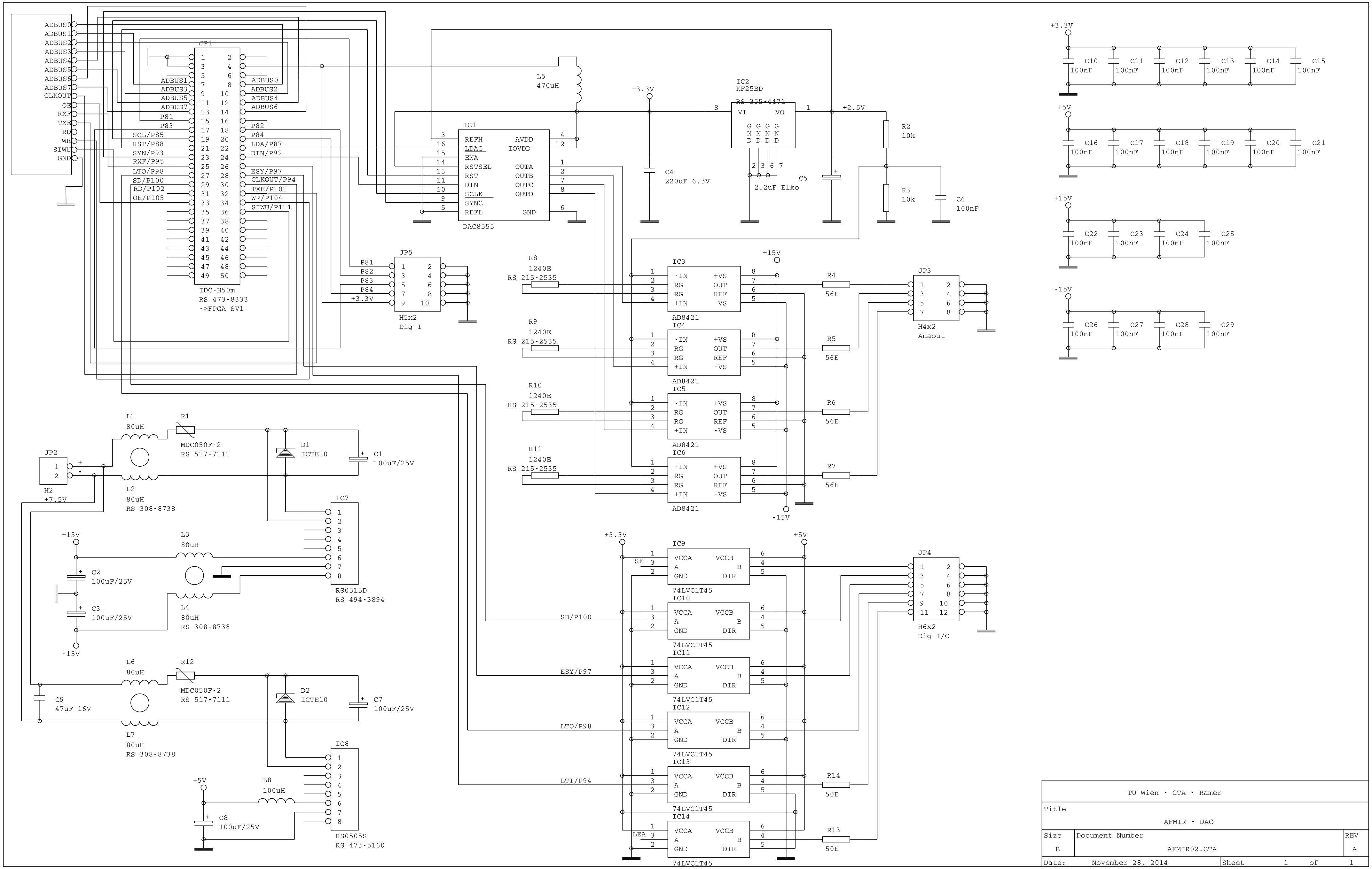
Literatur

1) *A. Dazzi, R. Prazeres, F. Glotin, J. Ortega,* Infrared Phys. Technol. *2006*, 49, *113*.
2) *N. Ocelic, A. Huber, and R. Hillenbrand,* Appl. Phys. Lett. *2006*, 89, *101124*.
3) *www.anasysinstruments.com*
4) *www.neaspec.com*
5) *C. Marcott, M. Lo, K. Kjoller, C. Prater, I. Noda,* Appl. Spectrosc. *2011*, 65, *1145*.
6) *B. V. A. N. Eerdenbrugh, M. Lo, K. Kjoller, C. Marcott, L. S. Taylor,* J. Pharm. Sci. *2012*, 101, *2066*.
7) *C. Mayet, A. Deniset-Besseau, R. Prazeres, J.-M. Ortega, A. Dazzi,* Biotechnol. Adv. *2013* 31, *369*.
8) *H. Cho, J. R. Felts, M.-F. Yu, L. A. Bergman, A. F. Vakakis, W. P. King,* Nanotechnology *2013*, 24, *444007*.
9) *C. Mayet, a Dazzi, R. Prazeres, F. Allot, F. Glotin, J. M. Ortega,* Opt. Lett. *2008*, 33, *1611*.
10) *B. Lahiri, G. Holland, V. Aksyuk, A. Centrone,* Nano Lett. *2013*, 13, *3218*.
11) *J. Chen, M. L. Nesterov, A. Y. Nikitin et al.,* Nano Lett. *2013*, 13, *6210*.
12) *F. Lu, M. Jin, M. A. Belkin,* Nat. Photonics *2014*, 8, *307*.
13) *I. Amenabar, S. Poly, W. Nuansing et al.,* Nat. Commun. *2013*, 4, *2890*.
14) *X. G. Xu, M. Rang, I. M. Craig, M. B. Raschke,* J. Phys. Chem. Lett. *2012*, 3, *1836*.
15) *F. Huth, A. Govyadinov, S. Amarie, W. Nuansing, F. Keilmann, R. Hillenbrand,* Nano Lett. *2012*, 12, *3973*.

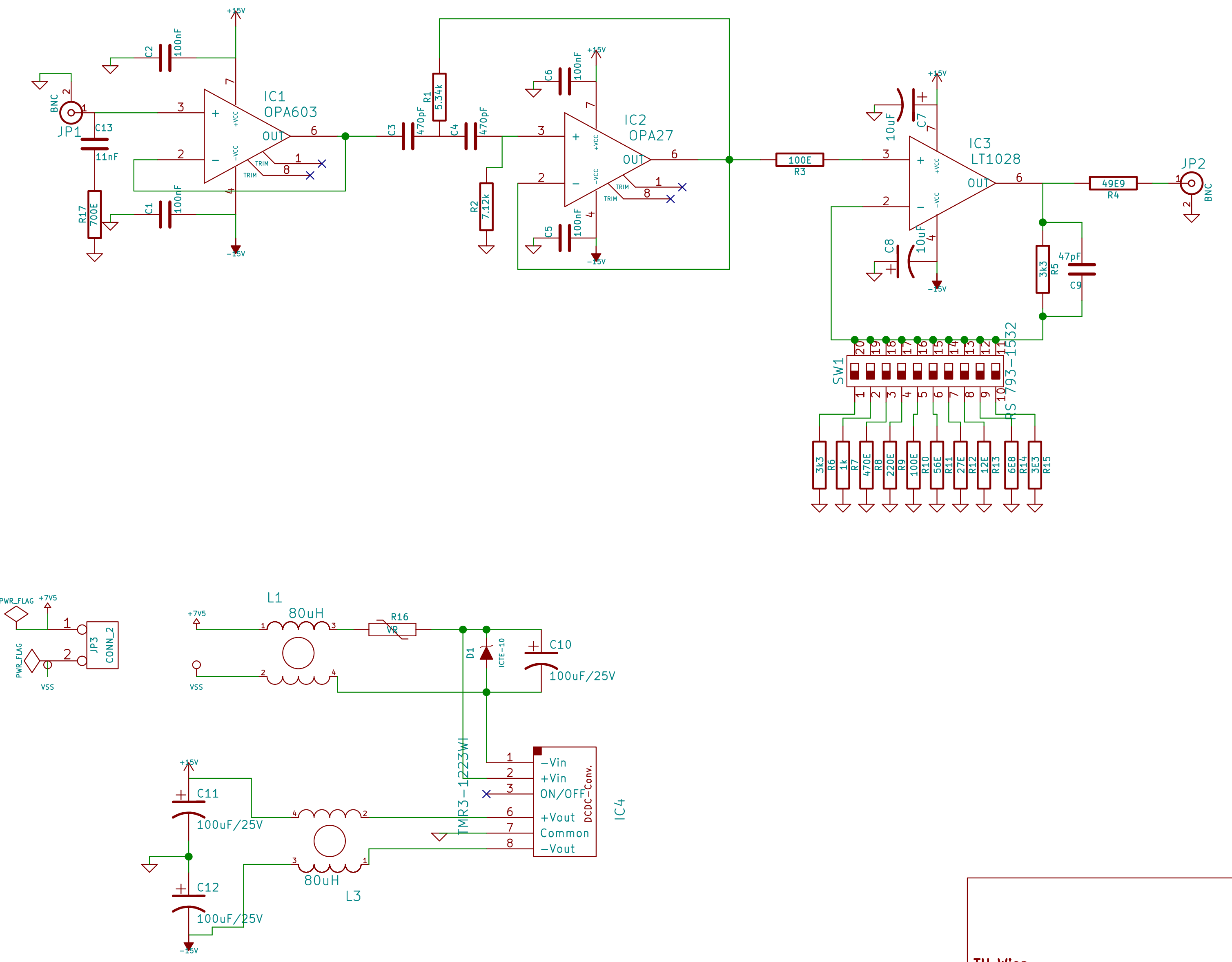
Part V.

Electronics


IC1
74AVC8T245
VCCA VCCB DIR VCCB OE GND
A1 A2 A3 A4 A5 A6 A7 A8
B1 B2 B3 B4 B5 B6 B7 B8
R/W DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
L2
DR35
RS 226-8191
IC2
74AVC8T245
DB8 DB9 DB10 DB11 DB12 DB13 DB14 DB15
IC3
74AVC8T245
CS RESET SYNC R/W CS2
IC4
74AVC8T245
DRDY
IC5
74LVC1T45
+5V MCLK
VCCA VCCB A B GND DIR
IC8
SN74AUC1G04DBV
JP1
IDC-H40m
->7760EVAL J3
MCLK DB0 DB2 DB4 DB6 DB8 DB10 DB12 DB14 CS
DB1 DB3 DB5 DB7 DB9 DB11 DB13 DB15 R/W DRDY SYNC RESET
JP3
IDC-H50m
RS 473-8333
->FPGA SV2
GND GND +3V3
SYNC/P51 DRDY/P41 DB15/P35 DB14/P33 DB12/P30 DB10/P27 DB8/P24 DB6/P22 DB4/P17 MCLK/P15 DB1/P12
RESET/P50 RDWR/P40 CS/P34 DB13/P32 DB11/P29 DB9/P26 DB7/P23 DB5/P21 DB3/P16 DB2/P14 DB0/P11
CS2/P134
JP4
H1
CS2out
JP2
7.5V
->7760EVAL
L1
470uH
RS 233-5279
C3
47uF 16V C3225
RS740-7802
IC7
LM2931AZ-5
RS 533-5575
VI VO GND
C1
47uF 16V C3225
RS740-7802
C4
10uF Elko
C10 100nF
C11 100nF
C12 100nF
C13 100nF
C14 100nF
IC6
KF25BD
RS 355-4471
VI VO GND GND GND GND
+2.5V
C2 100nF
C5 2.2uF Elko
C6 100nF
C7 100nF
C8 100nF
C9 100nF
TU Wien - CTA - Ramer
Title
AFMIR - ADC<->FPGA Connector
Size B
Document Number AFMIR01.CTA
REV A2
Date: January 16, 2015
Sheet 1 of 1



ADBUS0
ADBUS1
ADBUS2
ADBUS3
ADBUS4
ADBUS5
ADBUS6
ADBUS7
CLKOUT
OE
RXF
TXE
RD
WR
SIWU
GND
JP1
SCL/P85
RST/P88
SYN/P93
RXF/P95
LTO/P98
SD/P100
RD/P102
OE/P105
P81
P83
P82
P84
LDA/P87
DIN/P92
ESY/P97
CLKOUT/P94
TXE/P101
WR/P104
SIWU/P111
IDC-H50m
RS 473-8333
->FPGA SV1
JP5
P81
P82
P83
P84
+3.3V
H5x2
Dig I
L5
470uH
IC1
REFH
LDAC
ENA
RSTSEL
RST
DIN
SCLK
SYNC
REFL
AVDD
IOVDD
OUTA
OUTB
OUTC
OUTD
GND
DAC8555
+3.3V
C4
220uF 6.3V
IC2
KF25BD
RS 355-4471
VI
VO
GND
C5
2.2uF Elko
+2.5V
R2
10k
R3
10k
C6
100nF
R8
1240E
RS 215-2535
R9
1240E
RS 215-2535
R10
1240E
RS 215-2535
R11
1240E
RS 215-2535
IC3
IC4
IC5
IC6
-IN
RG
+IN
+VS
OUT
REF
-VS
AD8421
+15V
-15V
R4
56E
R5
56E
R6
56E
R7
56E
JP3
H4x2
Anaout
+3.3V
C10
C11
C12
C13
C14
C15
100nF
+5V
C16
C17
C18
C19
C20
C21
100nF
+15V
C22
C23
C24
C25
100nF
-15V
C26
C27
C28
C29
100nF
JP2
H2
+7.5V
L1
80uH
R1
MDC050F-2
RS 517-7111
D1
ICTE10
C1
100uF/25V
L2
80uH
RS 308-8738
IC7
RS0515D
RS 494-3894
+15V
L3
80uH
C2
100uF/25V
C3
100uF/25V
L4
80uH
RS 308-8738
-15V
L6
80uH
R12
MDC050F-2
RS 517-7111
D2
ICTE10
C7
100uF/25V
C9
47uF 16V
L7
80uH
RS 308-8738
IC8
RS0505S
RS 473-5160
+5V
L8
100uH
C8
100uF/25V
+3.3V
+5V
IC9
IC10
IC11
IC12
IC13
IC14
VCCA
VCCB
A
B
GND
DIR
74LVC1T45
SE
SD/P100
ESY/P97
LTO/P98
LTI/P94
LEA
R14
50E
R13
50E
JP4
H6x2
Dig I/O
TU Wien - CTA - Ramer
Title
AFMIR - DAC
Size
B
Document Number
AFMIR02.CTA
REV
A
Date: November 28, 2014
Sheet 1 of 1



TU Wien

Sheet: /

File: filter.sch

Title: AFM Deflection Amplifier Circuit

Size: A4 Date: 6 aug 2015	Rev:
KiCad E.D.A. kicad 4.0.1+dfsg1-stable	Id: 1/1

Part VI.

Source Code

Step Scan Characterization

```
 // Use digital pin 5 as output, connect to DI3 on 80v trigger box
int outPin = 5;
// connect to mircat tuned connector
int inPin = 3;
boolean was_high = false;
volatile boolean outPinold = false;
byte selected_tune = 1;
volatile byte max_tune = 2;
volatile byte current_tune = 2;
int ledPin = 13;


void setup()
{
  pinMode(outPin, OUTPUT);      // sets the digital pin as output
  pinMode(inPin,INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  noInterrupts();
}

void loop()
{
  boolean is_high = digitalRead(inPin);
  if (is_high){
    if ((not was_high) and is_high ){
      current_tune += 1;
    }
    if (current_tune> max_tune){
      current_tune = 0;
    }
    if (current_tune == selected_tune){
      digitalWrite(outPin, HIGH);
      digitalWrite(ledPin, HIGH);
    }else
      {
    digitalWrite(outPin,LOW);
    digitalWrite(ledPin,LOW);
      }
```

```
  }else{
    digitalWrite(outPin,LOW);
    digitalWrite(ledPin,LOW);
  }
  was_high = is_high;


}
```

PTIR FPGA Controller

1. AD7760 Interface

```
-- addr description
-- 0           gain (unsigned)
-- 1           offset (signed)
-- 2           15: filter1
--             14 to 12: filter2
--             11: filter3

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

library pipes;
use pipes.types.all;

library work;
use work.definitions.all;



entity PIPE_SOURCE_AD7760 is
  port (CLK_I           : in  std_logic;
        WE_I            : in  std_logic;
        STB_I           : in  std_logic;
        CYC_I           : in  std_logic;
        DAT_I           : in  std_logic_vector (15 downto 0);
        DAT_O           : out std_logic_vector (15 downto 0);
        ADDR_I          : in  std_logic_vector (7 downto 0);
        ERROR_O         : out std_logic;
        ACK_O           : out std_logic;
        RES_I           : in  std_logic;
        PIPEOUT_DATA_O  : out std_logic_vector (23 downto 0);
        PIPEOUT_VALID_O : out std_logic;
        PIPEOUT_STB_O   : out std_logic;
        PIPEOUT_CYC_O   : out std_logic;

        -- IO Ports AD7760
```

```
        MCLK      : in    std_logic;
        DRDY      : in    std_logic;
        DB        : inout std_logic_vector(15 downto 0);
        CS        : out   std_logic;
        RDWR      : out   std_logic;
        ADC_RESET : out   std_logic;
        SYNC      : out   std_logic;

        -- controls
        SYNC_IN : in std_logic
        );
end PIPE_SOURCE_AD7760;


architecture Behavioral of PIPE_SOURCE_AD7760 is

  constant DATA_DEPTH : integer := 3;
  constant RAM_INITIAL : pipe_settings_array(0 to DATA_DEPTH-1) :=
    (0 => std_logic_vector(to_unsigned(16#A000#, 16)),  --gain
     1 => std_logic_vector(to_signed(0, 16)),           -- offset
     2 => (15 => '1', 14 => '0', 13 => '0', 12 => '0', 11 => '1',
        ↪ others => '0'));


  signal RAM : pipe_settings_array(0 to DATA_DEPTH-1) := RAM_INITIAL;
  signal SET_DATA_SIG : boolean;
  signal INVALIDATE : std_logic;
  signal DATA_INT : std_logic_vector(24 downto 0);
  signal AD7760_STATE_INPUT      : AD7760_STATES := ADC_IDLE;
  signal AD7760_CHANGE_STATE     : std_logic     := '0';
  signal AD7760_CHANGE_STATE_ACK : std_logic;

  component AD7760_CONTROL
    port(
      MCLK             : in    std_logic;
      DRDY             : in    std_logic;
      RESET            : in    std_logic;
      CLK              : in    std_logic;
      SYNC_IN          : in    std_logic;
      STATE_IN         : in    AD7760_STATES;
      CHANGE_STATE     : in    std_logic;
      GAIN             : in    unsigned(15 downto 0);
      OFFSET           : in    signed(15 downto 0);
      FILTER1          : in    std_logic;
      FILTER2_DEC      : in    unsigned(2 downto 0);
      FILTER3          : in    std_logic;
```

```
      DB                : inout std_logic_vector(15 downto 0);
      CS                : out   std_logic;
      RDWR              : out   std_logic;
      ADC_RESET         : out   std_logic;
      SYNC              : out   std_logic;
      CHANGE_STATE_ACK  : out   std_logic;
      DATA_OUT          : out   signed(23 downto 0);
      DATA_VALID        : out   std_logic;
      DATA_NEW          : out   std_logic;
      DATA_OVR             : out   std_logic
      );
  end component;

begin

  INST_AD7760_CONTROL : AD7760_CONTROL port map(
    DB                         => DB,
    MCLK                       => MCLK,
    CS                         => CS,
    RDWR                       => RDWR,
    DRDY                       => DRDY,
    ADC_RESET                  => ADC_RESET,
    RESET                      => RES_I,
    CLK                        => CLK_I,
    SYNC                       => SYNC,
    SYNC_IN                    => SYNC_IN,
    STATE_IN                   => AD7760_STATE_INPUT,
    CHANGE_STATE               => AD7760_CHANGE_STATE,
    CHANGE_STATE_ACK           => AD7760_CHANGE_STATE_ACK,
    GAIN                       => unsigned(RAM(0)),
    OFFSET                     => signed(RAM(1)),
    FILTER1                    => RAM(2)(15),
    FILTER2_DEC                => unsigned(RAM(2)(14 downto 12)),
    FILTER3                    => RAM(2)(11),
    std_logic_vector(DATA_OUT) => PIPEOUT_DATA_O,
    DATA_VALID                 => PIPEOUT_VALID_O,
    DATA_NEW                   => PIPEOUT_STB_O
    );

  process (CLK_I)
    variable SET_DATA     : boolean := false;
    variable SYNC_IN_OLD : std_logic;
  begin

    if rising_edge(clk_i) then
      ACK_O <= '0';
```

```
      if AD7760_CHANGE_STATE_ACK = '1' then
        AD7760_CHANGE_STATE <= '0';
      end if;
      if RES_I = '1' then
        ERROR_O              <= '0';
        RAM                  <= RAM_INITIAL;
        AD7760_CHANGE_STATE  <= '0';
      else
        PIPEOUT_CYC_O <= '1';
        if AD7760_CHANGE_STATE_ACK = '1' then
          AD7760_CHANGE_STATE <= '0';
        end if;
        if CYC_I = '1' then
          if WE_I = '1' then
            SET_DATA := true;
            if STB_I = '1' then
              if unsigned(ADDR_I) < DATA_DEPTH then
                RAM(to_integer(unsigned(ADDR_I))) <= DAT_I;
                ACK_O                             <= '1';
              else
                ERROR_O <= '1';
              end if;
            end if;
          end if;
          if WE_I = '1'then
            if unsigned(ADDR_I) < DATA_DEPTH then
              DAT_O <= RAM(to_integer(unsigned(ADDR_I)));
              ACK_O <= '1';
            else
              ERROR_O <= '1';
            end if;
          end if;
        end if;
        if CYC_I = '0' then
          if SET_DATA then
            AD7760_STATE_INPUT  <= ADC_SET;
            AD7760_CHANGE_STATE <= '1';
            PIPEOUT_CYC_O       <= '0';
            SET_DATA            := false;
          end if;
        end if;
      end if;
      SET_DATA_SIG <= SET_DATA;
    end if;
  end process;

```

```
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

library work;
use work.definitions.all;

entity AD7760_control is
  port (DB               : inout std_logic_vector (15 downto 0);
        MCLK             : in    std_logic;
        CS               : out   std_logic;
        RDWR             : out   std_logic;
        DRDY             : in    std_logic;
        ADC_RESET        : out   std_logic;
        RESET            : in    std_logic;
        CLK              : in    std_logic;
        SYNC             : out   std_logic;
        SYNC_IN          : in    std_logic             := '1';
        STATE_IN         : in    AD7760_STATES         := ADC_IDLE;
        CHANGE_STATE     : in    std_logic             := '0';
        CHANGE_STATE_ACK : out   std_logic;
        GAIN             : in    unsigned(15 downto 0) := to_unsigned (16#A000#, 16);
        OFFSET           : in    signed(15 downto 0)   := to_signed (0, 16);
        FILTER1          : in    std_logic             := '1';
        FILTER2_DEC      : in    unsigned(2 downto 0)  := to_unsigned (2, 3);
        FILTER3          : in    std_logic             := '1';
        DATA_OUT         : out   ADC_MEASUREMENT;
        DATA_VALID       : out   std_logic;
        DATA_NEW         : out   std_logic;
        DATA_OVR         : out   std_logic
        );

end AD7760_control;

architecture Behavioral of AD7760_control is

  signal DB_IN       : std_logic_vector(15 downto 0);
  signal DB_OUT      : std_logic_vector(15 downto 0) := (others => '0');
  signal RDWR_INT    : std_logic                     := '1';
  signal INITIALIZED : std_logic                     := '0';
```

```

  signal BOOT_OLD : boolean := false;
  signal BOOT_NOW : boolean := false;

  signal SET_COUNTER : natural range 0 to 500 := 0;

  type FSM_STATES_INTERNAL is (IDLE, MEASURE, SET, SYNCHRONIZE);
  signal STATE : FSM_STATES_INTERNAL := IDLE;

-- AD7760 registers

  signal REG_1 : std_logic_vector(15 downto 0);
  signal REG_2 : std_logic_vector(15 downto 0);
  signal PD    : std_logic := '1';

  signal RECEIVE_MEASUREMENT : boolean              := false;
  signal MEASUREMENT_COUNTER : natural range 0 to 48 := 0;

  signal ARMED         : boolean              := true;
  signal FIRST_BYTES   : std_logic_vector (15 downto 0);
  signal SYNC_COUNTER : natural range 0 to 20 := 0;

  signal CHANGE_STATE_ARMED : boolean := false;

  signal ADC_RESET_INT : std_logic := '0';


begin
  DB     <= DB_out when RDWR_int = '1' else (others => 'Z');
  DB_in  <= DB;
  RDWR   <= RDWR_int;


--registers
  reg_1 <= (15          => '0',          --DL_FILT
            14          => '0',          --RDOVR
            13          => '0',          --RDGAIN
            12          => '0',          --RDOFF
            11          => '0',          --RDSTAT
            10          => '0',          -- always 0
            9           => '0',          -- SYNC
            8 downto 5  => '0',          --FLEN
            4           => filter3,
            3           => filter1,
            2           => filter2_dec(2),
            1           => filter2_dec(1),
```

```
            0          => filter2_dec(0));

reg_2 <= (5      => '0',                --CDIV
          3      => PD,                 --PD
          2      => '0',                --LPWR
          1      => '1',                --always '1'
          0      => '0',                --D1PD
          others => '0');


RESET_PROC : process(MCLK)

begin
  if rising_edge(MCLK) then
    ADC_RESET <= ADC_RESET_INT;
  end if;
end process;



interface_proc : process(CLK)
begin
  if rising_edge(clk) then
    DATA_NEW <= '0';
    if RESET = '1' or not INITIALIZED = '1' then
      ADC_RESET_INT      <= '0';
      STATE              <= IDLE;
      RDWR_INT           <= '1';
      SYNC               <= '1';
      CS                 <= '1';
      SET_COUNTER        <= 0;
      INITIALIZED        <= '1';
      BOOT_OLD           <= false;
      BOOT_NOW           <= false;
      CHANGE_STATE_ARMED <= true;
      DATA_OVR           <= '0';
      -- reset PD to the state on the ADC
      PD                 <= '1';
    else
      CHANGE_STATE_ACK <= '0';
      if CHANGE_STATE = '0' then
        CHANGE_STATE_ARMED <= true;
      end if;
      case STATE is

        when IDLE =>
```

```
            CHANGE_STATE_ACK    <= '0';
            RDWR_INT            <= '1';
            ADC_RESET_INT       <= '1';
            DATA_VALID          <= '0';
            SYNC                <= '1';
            RECEIVE_MEASUREMENT <= false;
            if SYNC_IN = '0' then
              STATE <= SYNCHRONIZE;
            else
              if CHANGE_STATE_ARMED and CHANGE_STATE = '1' then
                case STATE_IN is
                  when ADC_IDLE =>
                    STATE <= IDLE;
                  when ADC_SET =>
                    STATE <= SET;
                  when ADC_MEASURE =>
                    STATE <= MEASURE;
                  when ADC_SYNC =>
                    STATE <= SYNCHRONIZE;
                end case;
                CHANGE_STATE_ARMED <= false;
                CHANGE_STATE_ACK   <= '1';
              end if;
            end if;

          when SET =>
            PD                 <= '0';
            SET_COUNTER        <= SET_COUNTER + 1;
            CHANGE_STATE_ACK <= '0';
            case SET_COUNTER is
              when 0 to 15 =>
                ADC_RESET_INT <= '0';
                RDWR_INT      <= '1';
              when 16 to 36 =>
                ADC_RESET_INT <= '1';
              when 37 =>
                -- select Power register (register 2)
                DB_OUT <= std_logic_vector(to_unsigned(2, DB_OUT'
                    ↪ length));
              when 38 to 78 =>
                CS <= '0';
              when 79 to 129 =>
                CS <= '1';
              when 130 =>
                DB_OUT <= REG_2;
              when 131 to 166 =>
```

```
            CS <= '0';
          when 167 to 182 =>
            CS <= '1';
          when 183 =>
            -- select filter and settings register (register 1)
            DB_OUT <= std_logic_vector(to_unsigned(1, DB'length))
               ↪ ;
          when 184 to 219 =>
            CS <= '0';
          when 220 to 250 =>
            CS <= '1';
          when 251 =>
            DB_OUT <= REG_1;
          when 252 to 292 =>
            CS <= '0';
          when 293 =>
            CS <= '1';
          when 294 to 318 =>
            --select offset register
            DB_OUT <= std_logic_vector(to_unsigned(3, DB_OUT'
               ↪ length));
          when 319 to 359 =>
            CS <= '0';
          when 360 to 380 =>
            CS <= '1';
          when 381 =>
            DB_OUT <= std_logic_vector(OFFSET);
          when 382 to 399 =>
            CS <= '0';
          when 400 to 402 =>
            CS <= '1';
          when 403 to 423 =>          --increased this and all
             ↪ following +1
            -- select gain register
            DB_OUT <= std_logic_vector(to_unsigned(4, DB'length))
               ↪ ;
          when 424 to 444 =>
            CS <= '0';
          when 445 to 464 =>
            CS <= '1';
          when 465 =>
            DB_OUT <= std_logic_vector(GAIN);
          when 466 to 496 =>
            CS <= '0';
          when others =>
            STATE       <= IDLE;
```

```
                CS          <= '1';
                SET_COUNTER <= 0;
            end case;

          when MEASURE =>
            CHANGE_STATE_ACK <= '0';
            if SYNC_IN = '0' then
              STATE <= SYNCHRONIZE;
            else
              if CHANGE_STATE_ARMED and CHANGE_STATE = '1' then
                STATE               <= IDLE;
                RECEIVE_MEASUREMENT <= false;
                ARMED               <= true;
              else
                if ARMED and DRDY = '0' then
                  -- DRDY has gone low, we begin a new measurement
                  RECEIVE_MEASUREMENT <= true;
                  RDWR_INT            <= '0';
                  CS                  <= '1';
                  MEASUREMENT_COUNTER <= 1;
                end if;
                if RECEIVE_MEASUREMENT then
                  MEASUREMENT_COUNTER <= MEASUREMENT_COUNTER + 1;
                end if;
                SYNC <= '1';
                case MEASUREMENT_COUNTER is
                  when 0 =>
                  when 1 =>
                    RDWR_INT <= '0';
                  when 2 to 6 =>
                    CS    <= '0';
                    ARMED <= true;
                  when 7 =>
                    -- read first two bytes into word_in
                    FIRST_BYTES <= DB_IN;
                    DB_OUT      <= DB_IN;
                  when 8 =>
                    CS <= '1';
                  when 9 to 13 =>
                    RDWR_INT <= '1';
                  when 14 =>
                    RDWR_INT <= '0';
                  when 15 to 19 =>
                    CS <= '0';
                  --data_valid <= '0';
                  when 20 =>
```

```
                    DATA_OUT <= signed(FIRST_BYTES & DB_IN(15 downto
                        ↪ 8));
                    if DB_IN(7) = '1' then
                      -- only announce valid data if DVALID = 1
                      DATA_VALID <= '1';
                    else
                      DATA_VALID <= '0';
                    end if;
                    DATA_OVR <= DB_IN(6);
                    DATA_NEW <= '1';

                    DB_OUT <= DB_IN;
                  when 21 =>
                    CS <= '1';
                  when others =>
                    RDWR_INT            <= '1';
                    MEASUREMENT_COUNTER <= 0;
                    RECEIVE_MEASUREMENT <= false;
                end case;
              end if;
            end if;

          when SYNCHRONIZE =>
            -- measurement starts four cycles after change_state_ack
            if SYNC_IN = '1' then
              SYNC                <= '1';
              RECEIVE_MEASUREMENT <= false;
              CHANGE_STATE_ACK    <= '1';
              STATE               <= MEASURE;
            else
              SYNC <= '0';
            end if;

        end case;
      end if;
    end if;

  end process;

end Behavioral;
```

2. DAC8555

```
-- addr description
-- 0                    0 DAC 0 enable
```

```
--                                                  1 DAC 1 enable
--                                                  2 DAC 2 enable
--                                                  3 DAC 3 enable


library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.Numeric_std.all;


entity wb_DAC_sink is
  port (pipein_data_i : in  std_logic_vector (63 downto 0);
        pipein_cyc_i  : in  std_logic_vector (3 downto 0);
        pipein_stb_i  : in  std_logic_vector (3 downto 0);
        clk_i         : in  std_logic;
        rst_i         : in  std_logic;
        error_o       : out std_logic;
        stb_i         : in  std_logic;
        cyc_i         : in  std_logic;
        ack_o         : out std_logic;
        addr_i        : in  std_logic_vector(7 downto 0);
        data_i        : in  std_logic_vector(15 downto 0);
        data_o        : out std_logic_vector(15 downto 0);
        we_i          : in  std_logic;
        DAC_SCLK      : out std_logic;
        DAC_SYNC      : out std_logic;
        DAC_DIN       : out std_logic;
        DAC_RST       : out std_logic;
        DAC_LDAC      : out std_logic);
end wb_DAC_sink;

architecture Behavioral of wb_DAC_sink is

  component DAC8555
    port(
      clk             : in  std_logic;
      DAC0            : in  signed(15 downto 0);
      DAC1            : in  signed(15 downto 0);
      DAC2            : in  signed(15 downto 0);
      DAC3            : in  signed(15 downto 0);
      res             : in  std_logic;
      active_channels : in  std_logic_vector(3 downto 0);
      DAC_SCLK        : out std_logic;
      DAC_SYNC        : out std_logic;
      DAC_DIN         : out std_logic;
      DAC_RST         : out std_logic;
```

```
      DAC_LDAC         : out std_logic
      );
  end component;

  signal active_channels : std_logic_vector (3 downto 0);
  type DAC_values_array is array (0 to 3) of std_logic_vector(15 downto 0);
  signal DAC_values      : DAC_values_array := (others => (others => '0'));

begin

  Inst_DAC8555 : DAC8555 port map(
    clk             => clk_i,
    DAC0            => signed(DAC_values(0)),
    DAC1            => signed(DAC_values(1)),
    DAC2            => signed(DAC_values(2)),
    DAC3            => signed(DAC_values(3)),
    res             => rst_i,
    DAC_SCLK        => DAC_SCLK,
    DAC_SYNC        => DAC_SYNC,
    DAC_DIN         => DAC_DIN,
    DAC_RST         => DAC_RST,
    DAC_LDAC        => DAC_LDAC,
    active_channels => active_channels
    );

  process (clk_i)
  begin
    if rising_edge(clk_i) then
      if rst_i = '1' then
        data_o <= (others => '0');
      else
        for channel in 0 to 3 loop
          if pipein_cyc_i(channel) = '1' then
            if pipein_stb_i(channel) = '1' then
              DAC_values(channel) <= pipein_data_i((16 * (channel + 1)) - 1 downto (16 * (channel)));
            end if;
          end if;
        end loop;
        ack_o <= '0';
        if cyc_i = '1' and stb_i = '1' then
          if we_i = '1' then
            active_channels <= data_i(3 downto 0);
            ack_o           <= '1';
```

```
          else
            data_o (3 downto 0) <= active_channels;
          end if;
        end if;
      end if;
    end if;
  end process;

end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;


entity DAC8555 is

  port (clk             : in  std_logic;
        DAC0            : in  signed (15 downto 0);
        DAC1            : in  signed (15 downto 0);
        DAC2            : in  signed (15 downto 0);
        DAC3            : in  signed (15 downto 0);
        res             : in  std_logic;
        DAC_SCLK        : out std_logic;
        DAC_SYNC        : out std_logic;
        DAC_DIN         : out std_logic;
        DAC_RST         : out std_logic;
        DAC_LDAC        : out std_logic;
        active_channels : in  std_logic_vector (3 downto 0) := "1111"
        );
end DAC8555;

architecture Behavioral of DAC8555 is

  constant CLK_per_half_SCLK : integer := 3;  --
  signal DAC0_hold : signed (15 downto 0) := (others => '0');
  signal DAC1_hold : signed (15 downto 0) := (others => '0');
  signal DAC2_hold : signed (15 downto 0) := (others => '0');
  signal DAC3_hold : signed (15 downto 0) := (others => '0');
  signal active_channels_hold : std_logic_vector (3 downto 0);

  type fsm_states is (start_cycle, CHANGE_CHANNEL, SYNC, writing,
     ↪ OUTPUT);
  signal state           : fsm_states                    :=
     ↪ start_cycle;
  signal channel         : unsigned(2 downto 0)          :=
```

```
    to_unsigned(0, 3);
  signal out_data       : std_logic_vector(23 downto 0) := (others => '0');
  subtype DAC_select_type is integer range 0 to 3;
  signal DAC_select     : DAC_select_type               := 3;
  signal DAC_select_val : std_logic_vector (15 downto 0);
  subtype state_counter_type is integer range 0 to 2*CLK_per_half_SCLK;
  signal state_counter  : state_counter_type            := 0;

  subtype DAC_clk_counter_type is integer range 0 to CLK_per_half_SCLK;
  signal DAC_clk_counter   : DAC_clk_counter_type := 0;
  subtype out_bit_index_type is integer range 23 downto 0;
  signal out_bit_index     : out_bit_index_type   := 23;
  signal DAC_SCLK_internal : std_logic            := '0';
  signal DAC_LDAC_int : std_logic;

begin



  with DAC_select select DAC_select_val <=
    std_logic_vector(DAC0_hold) when 0,
    std_logic_vector(DAC1_hold) when 1,
    std_logic_vector(DAC2_hold) when 2,
    std_logic_vector(DAC3_hold) when 3,
    (others => '0')             when others;

  out_data (21) <= '0';                 --LD1
  out_data (20) <= '1';                 --LD0

  out_data (23 downto 22) <= (others => '0');
  out_data (19)           <= '0';
  out_data (18 downto 17) <= std_logic_vector(to_unsigned(DAC_select, 2));
  out_data (16)           <= '0';       -- power down
  out_data (15 downto 0)  <= DAC_select_val;

  DAC_SCLK <= DAC_SCLK_internal;
  DAC_LDAC <= DAC_LDAC_int;

  process (clk)
    variable last_channel : boolean := true;
  begin
    if rising_edge(clk) then
```

```
      if res = '1' then
        DAC_RST          <= '0';
        DAC_SYNC         <= '1';
        DAC_clk_counter  <= 0;
        state_counter    <= 0;
        DAC_LDAC_int     <= '0';
      else
        DAC_LDAC_int <= '0';
        DAC_SYNC     <= '1';
        DAC_RST <= '1';
        case state is

          when START_CYCLE =>
            DAC0_hold            <= DAC0;
            DAC1_hold            <= DAC1;
            DAC2_hold            <= DAC2;
            DAC3_hold            <= DAC3;
            active_channels_hold <= active_channels;
            last_channel         := true;
            for I in 3 downto 0 loop
              if active_channels(I) = '1' then
                last_channel := false;
                DAC_select   <= I;
                exit;
              end if;
            end loop;
            out_bit_index <= 23;
            if last_channel then
              state <= START_CYCLE;
            else
              state <= WRITING;
            end if;
            state_counter <= 0;

          when CHANGE_CHANNEL =>
            if DAC_select = 0 then
              state <= OUTPUT;
            else
              last_channel := true;
              for I in 3 downto 0 loop
                if not(I >= DAC_select) then
                  if active_channels_hold(I) = '1' then
                    last_channel := false;
                    DAC_select   <= I;
                    exit;
                  end if;
```

```
          end if;
        end loop;
        if last_channel then
          state <= OUTPUT;
        else
          state <= SYNC;
        end if;
      end if;
      state_counter <= 0;

    when SYNC =>
      state_counter <= state_counter + 1;
      if state_counter = 2 * CLK_per_half_SCLK -1 then
        state         <= writing;
        state_counter <= 0;
        out_bit_index <= 23;
      end if;

    when writing =>
      state_counter <= state_counter + 1;
      DAC_SYNC      <= '0';

      case state_counter is
        when 0 to CLK_per_half_SCLK -1 =>
          DAC_DIN           <= out_data(out_bit_index);
          DAC_SCLK_internal <= '1';
        when CLK_per_half_SCLK to 2*CLK_per_half_SCLK-2 =>
          DAC_SCLK_internal <= '0';
        when 2*clk_per_half_SCLK -1 =>
          state_counter <= 0;
          if out_bit_index = 0 then
            state <= change_channel;
          else
            out_bit_index <= out_bit_index - 1;
            state_counter <= 0;
          end if;
        when 2*CLK_per_half_SCLK =>
                                      -- should not happen
      end case;

    when OUTPUT =>
      DAC_LDAC_int <= '1';
      if DAC_LDAC_int = '1' then
        state        <= start_cycle;
        DAC_LDAC_int <= '0';
      end if;
```

```
        end case;
      end if;
    end if;
  end process;



end Behavioral;
```

3. FT2232H interface

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;


entity FT2232H_faster is
  port (CLK_I         : in  std_logic;
        RST_I         : in  std_logic;
        write_DAT_I   : in  std_logic_vector (7 downto 0);
        write_STB_I   : in  std_logic;
        write_READY_O : out std_logic;
        write_STALL_O : out std_logic;
        write_ACK_O   : out std_logic := '1';
        read_DAT_O    : out std_logic_vector (7 downto 0);
        read_STB_O    : out std_logic;

        --off chip connections
        FT2232h_CLKOUT : in    std_logic;
        FT2232h_RXF    : in    std_logic;
        FT2232h_TXE    : in    std_logic;
        FT2232h_OE     : out   std_logic;
        FT2232h_RD     : out   std_logic;
        FT2232h_WR     : out   std_logic;
        FT2232h_ADBUS  : inout std_logic_vector(7 downto 0));
end FT2232H_faster;

architecture Behavioral of FT2232H_faster is


-- CLKOUT domain
  signal write_fifo_valid : std_logic;
  signal write_fifo_empty : std_logic;
  signal write_fifo_rd_en : std_logic;
  signal fifo_read_wr_en  : std_logic;
```

```
  signal read_fifo_din    : std_logic_vector(7 downto 0);
  signal write_fifo_dout  : std_logic_vector(7 downto 0);
  signal have_old_byte    : boolean;
  signal old_byte         : std_logic_vector (7 downto 0);
  signal FT2232h_OE_int   : std_logic;
  signal write_ERR_O : std_logic;
  signal read_fifo_almost_full : std_logic;

  type state_type is (state_idle, state_read, state_start_read,
     ↪ state_start_read_2, state_write, state_start_write);
  signal state : state_type := state_idle;

  signal ADBUS_read  : std_logic_vector(7 downto 0);
  signal ADBUS_write : std_logic_vector(7 downto 0);

--CLK_I domain
  signal read_fifo_rd_en : std_logic := '1';

begin

  FT2232h_OE <= FT2232h_OE_int;
  FT2232h_ADBUS <= ADBUS_write when FT2232h_OE_int = '1' else (others
     ↪  => 'Z');
  ADBUS_read    <= FT2232h_ADBUS;

  communication : process (FT2232H_CLKOUT)
  begin
    if (rising_edge(FT2232H_CLKOUT)) then
      FT2232h_rd       <= '1';
      FT2232h_wr       <= '1';
      FT2232h_oe_int   <= '1';
      write_fifo_rd_en <= '0';
      fifo_read_wr_en  <= '0';
      if RST_I = '1' then
        state <= state_idle;
      else
        case state is

          when state_idle =>
            FT2232h_RD <= '1';
            FT2232h_WR <= '1';
            if FT2232h_TXE = '0' and not (write_fifo_empty = '1')
               ↪ then
              state <= state_start_write;
            end if;
```

```
          if FT2232h_RXF = '0' and not (read_fifo_almost_full = '1') then
            state <= state_start_read;
          end if;

        when state_start_read =>
          FT2232h_oe_int <= '0';
          state          <= state_start_read_2;

        when state_start_read_2 =>
          FT2232h_rd     <= '0';
          FT2232h_oe_int <= '0';
          state          <= state_read;

        when state_read =>
          FT2232h_rd     <= '0';
          FT2232h_oe_int <= '0';
          if FT2232h_rxf = '0' then
            read_fifo_din   <= ADBUS_read;
            fifo_read_wr_en <= '1';
          else
            state      <= state_idle;
            FT2232h_rd <= '0';
          end if;
          if (read_fifo_almost_full = '1') then
            state      <= state_idle;
            FT2232h_rd <= '0';
          end if;

        when state_start_write =>
          if not have_old_byte then
            write_fifo_rd_en <= '1';
          else
            FT2232h_wr <= '0';
          end if;
          state <= state_write;

        when state_write =>
          if FT2232h_TXE = '0' and not (FT2232h_RXF = '0') then
            if write_fifo_valid = '1' or have_old_byte then
              write_fifo_rd_en <= '1';
              FT2232h_wr       <= '0';
              if have_old_byte then
                ADBUS_write   <= old_byte;
                have_old_byte <= false;
              else
```

```
                ADBUS_write <= write_fifo_dout;
              end if;
            else
              state <= state_idle;
            end if;
          else
            state <= state_idle;
            if write_fifo_valid = '1' then
              old_byte      <= write_fifo_dout;
              have_old_byte <= true;
            end if;
          end if;
      end case;
    end if;
  end if;
end process;


FIFO_to_write : entity work.FIFO_FT2232h
  port map (
    rst         => RST_I,
    wr_clk      => CLK_I,
    rd_clk      => FT2232h_CLKOUT,
    din         => write_DAT_I,
    wr_en       => write_STB_I,
    rd_en       => write_fifo_rd_en,
    dout        => write_fifo_dout,
    full        => write_ERR_O,
    almost_full => write_STALL_O,
    empty       => write_fifo_empty,
    valid       => write_fifo_valid
    );

FIFO_from_read : entity work.FIFO_FT2232h
  port map (
    rst         => RST_I,
    wr_clk      => FT2232h_CLKOUT,
    rd_clk      => CLK_I,
    din         => read_fifo_din,
    wr_en       => fifo_read_wr_en,
    rd_en       => read_fifo_rd_en,
    dout        => read_DAT_O,
    almost_full => read_fifo_almost_full,
    valid       => read_STB_O
    );
```

```
end Behavioral;
```

4. Cantilever Simulation

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.MATH_REAL.all;



entity oscillator is
  generic(stepsize          : time := 1 ns;
          sensitivity       : real := 1.0;
          mass              : real := 0.2/(2.0*2.0*3.1415*3.1415*13.0
             ↪ E3*13.0E3);
          force_constant    : real := 0.2;
          offset            : real := 0.0;
          height            : real := 2.0E-9;     -- sample height
          expansion         : real := 20.0E-12;   --5.0E-12; -- from
             ↪ Belkin
          pulloff_force     : real := 10.0E-9;    --2*pi*R*w
          tip_radius        : real := 25.0E-9;
          eta_n             : real := 200.0;      --PS (nach Rademacher
             ↪ )
          a0                : real := 0.40E-9;    -- intermolecular
             ↪ distance
          Hamaker_constant : real := 18.21E-21   --Hamaker constant PS
          );
  port (heating    : in  boolean;
        Q          : in  real := 100.0;
        reduced_E  : in  real := 5.0E9; -- reduced youngs modulus
        d0         : in  real := 0.0;    -- depth of penetration into
           ↪ material for straight cantilver
        deflection : out real);
end oscillator;


architecture Behavioral of oscillator is

  type typ_values_mem is array (0 to 1) of real;
  constant dt               : real := real(stepsize / 1 ps) /1.0e12;
  constant om0              : real := sqrt(force_constant/mass);
  constant pi               : real := 3.141593;
```

```
  constant pull_off_factor : real := Hamaker_constant * tip_radius
     ↪ /6.0;
  constant highpass_RC     : real := 1.0/25.0E3 /2.0/pi;  --fc / 2pi
  constant highpass_factor : real := highpass_RC/(highpass_RC+dt);

begin
  process

    variable highpass_yi : real := 0.0;
    variable zs          : typ_values_mem := (others => 0.0);
    variable z           : real           := 0.0;
    variable input_force : real;
    variable DMT_force   : real           := 0.0;
    variable delta       : real;
    variable delta_old   : real           := d0+z;
    variable Temp        : real           := 0.0;

  begin
    if heating then
      delta := d0 + z + expansion;
    else
      delta := d0 + z;
    end if;
    if delta < a0 then
      DMT_force := 4.0/3.0*reduced_E * sqrt(tip_radius) * sqrt((a0-
         ↪ delta)**3.0) - pull_off_factor/(a0*a0)-eta_n*pi*
         ↪ tip_radius/height*(a0-delta)*(delta-delta_old)/dt;
    else
      DMT_force := - pull_off_factor/(delta*delta);
    end if;

    delta_old := delta;
    z         :=dt*dt/mass/(1.0+omo/Q*dt)*DMT_force-zs(0)*(omo*dt*omo
       ↪ *dt-2.0)/(1.0+omo/Q*dt)-zs(1)*(1.0-omo/Q*dt)/(1.0+omo/Q*dt)
       ↪ ;

    zs(1) := zs(0);
    zs(0) := z;

    highpass_yi := highpass_factor * highpass_yi + highpass_factor*(
       ↪ zs(0) - zs(1));
    deflection  <= sensitivity * highpass_yi+offset;
    wait for stepsize;
  end process;

```

```
end Behavioral;
```

5. Pulse Generator

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.Numeric_std.all;

library work;
use work.types.all;


entity wb_pulsegen is
  port (clk_i                  : in  std_logic;
        res_i                  : in  std_logic;
        stb_i                  : in  std_logic;
        dat_i                  : in  std_logic_vector(15 downto 0);
        dat_o                  : out std_logic_vector(15 downto 0);
        ack_o                  : out std_logic;
        we_i                   : in  std_logic;
        error_o                : out std_logic;
        addr_i                 : in  std_logic_vector (7 downto 0);
        cyc_i                  : in  std_logic;
        pulse_out              : out std_logic;
        reference_out          : out std_logic;
        reference_quad_out     : out std_logic;
        pipeout_stb_o          : out std_logic;
        pipeout_cyc_o          : out std_logic;
        pipeout_phase_count_o : out std_logic_vector (15 downto 0);
        pipeout_full_circle_o : out std_logic_vector (15 downto 0));
end wb_pulsegen;

architecture Behavioral of wb_pulsegen is

  constant data_depth : integer := 3;
  signal pulse_int : std_logic := '0';
  signal ram           : pipe_settings_array(0 to data_depth-1);
  constant ram_initial : pipe_settings_array(0 to data_depth-1) := (
     ↪ others => (others => '0'));
  signal invalidate : std_logic;
  signal data_int         : std_logic_vector(63 downto 0);
  signal pulse_counter    : unsigned(15 downto 0) := (others => '0');
     ↪  -- counts the length of the emission
  signal cycle_counter    : unsigned(15 downto 0) := (others => '0');
     ↪  -- counts the length of a full cycle
```

```
  signal delay           : unsigned(15 downto 0) := (others => '0');
  signal delay_plus_half : unsigned(15 downto 0) := (others => '0');
  signal half            : unsigned(15 downto 0) := (others => '0');
  signal params_not_changed : std_logic := '0';

  component wb_filter_template
    generic(data_depth : integer;
            ram_intial : pipe_settings_array);
    port(
      clk_i      : in  std_logic;
      we_i       : in  std_logic;
      stb_i      : in  std_logic;
      dat_i      : in  std_logic_vector(15 downto 0);
      dat_o      : out std_logic_vector(15 downto 0);
      addr_i     : in  std_logic_vector(7 downto 0);
      cyc_i      : in  std_logic;
      error_o    : out std_logic;
      res_i      : in  std_logic;
      ack_o      : out std_logic;
      invalidate : out std_logic;
      ram        : out pipe_settings_array(0 to data_depth-1)
      );
  end component;

begin

  wb_filter_control : wb_filter_template
    generic map(data_depth => data_depth,
                ram_intial => ram_initial
                )
    port map(
      clk_i      => clk_i,
      we_i       => we_i,
      stb_i      => stb_i,
      dat_i      => dat_i,
      dat_o      => dat_o,
      addr_i     => addr_i,
      ack_o      => ack_o,
      cyc_i      => cyc_i,
      error_o    => error_o,
      res_i      => res_i,
      invalidate => invalidate,
      ram        => ram
      );

  pulse_out <= pulse_int;
```

```

process (clk_i)
begin

  if rising_edge(clk_i) then
    if (res_i = '1') then
      pulse_int       <= '0';
      cycle_counter   <= unsigned(ram(0));
      pulse_counter   <= unsigned(ram(1));
      delay           <= SHIFT_RIGHT(unsigned(ram(0)), 2);
      half            <= SHIFT_RIGHT(unsigned(ram(0)), 1);
      delay_plus_half <= SHIFT_RIGHT(unsigned(ram(0)), 2) + SHIFT_RIGHT(unsigned(ram(0)), 1);
      pipeout_cyc_o <= '0';
      pipeout_stb_o <= '0';
      pipeout_phase_count_o <= (others => '0');
      pipeout_full_circle_o <= (others => '0');
      pipeout_cyc_o         <= '0';
      pipeout_stb_o         <= '0';
      cycle_counter         <= (others => '0');
      pulse_counter         <= (others => '0');
      params_not_changed    <= '0';
    else
      if invalidate = '1' then
        params_not_changed <= '1';
      end if;
      if cycle_counter = half then
        reference_quad_out <= '1';
      end if;
      if cycle_counter = delay_plus_half then
        reference_quad_out <= '0';
      end if;
      if cycle_counter = delay then
        reference_out <= '0';
      end if;
      if (invalidate = '1' or params_not_changed = '1') and cycle_counter = 0 then
        params_not_changed <= '0';
        pulse_int          <= '0';
        cycle_counter      <= unsigned(ram(0));
        pulse_counter      <= unsigned(ram(1));
        pipeout_cyc_o      <= '0';
        pipeout_stb_o      <= '0';
        if cyc_i = '1' and stb_i = '1' and we_i = '1' then
          if unsigned(addr_i) = 0 then
            cycle_counter   <= unsigned(dat_i);
```

```
            delay           <= SHIFT_RIGHT(unsigned(dat_i), 2);
            half            <= SHIFT_RIGHT(unsigned(dat_i), 1);
            delay_plus_half <= SHIFT_RIGHT(unsigned(dat_i), 2) +
               ↪ SHIFT_RIGHT(unsigned(dat_i), 1);
          end if;
          if unsigned(addr_i) = 1 then
            pulse_counter <= unsigned(dat_i);
          end if;
        end if;

      else
        if unsigned(ram(0)) /= 0 then
          pipeout_cyc_o <= '1';
          pipeout_stb_o <= '1';
          if pulse_counter = 0 then
            pulse_int <= '0';
          else
            pulse_int     <= '1';
            pulse_counter <= pulse_counter - 1;
          end if;
          if cycle_counter = 1 then
            reference_out         <= '1';
            cycle_counter         <= unsigned(ram(0));
            pulse_counter         <= unsigned(ram(1));
            pipeout_phase_count_o <= std_logic_vector(unsigned(ram
               ↪ (0)) - cycle_counter);
            pipeout_full_circle_o <= ram(0);
            delay                 <= SHIFT_RIGHT(unsigned(ram(0)),
               ↪ 2);
            half                  <= SHIFT_RIGHT(unsigned(ram(0)),
               ↪ 1);
            delay_plus_half       <= SHIFT_RIGHT(unsigned(ram(0)),
               ↪ 2) + SHIFT_RIGHT(unsigned(ram(0)), 1);
          else
            cycle_counter <= cycle_counter - 1;
            pipeout_phase_count_o <= std_logic_vector(unsigned(ram
               ↪ (0)) - cycle_counter);
            pipeout_full_circle_o <= ram(0);
          end if;
        else
          pulse_int <= '0';
        end if;
      end if;
    end if;
  end if;
end process;
```

```


end Behavioral;
```

6. Maximum Detector

```
--register
--0             start cycle counts


library IEEE;
use IEEE.STD_LOGIC_1164.all;
use ieee.numeric_std.all;

library pipes;
use pipes.all;
use pipes.types.all;

entity max_detector_sweep is
  generic (amplitude_length : integer := 24;
           frequency_length : integer := 10);
  port (clk                  : in  std_logic;
        pipein_amplitude      : in  std_logic_vector (
           ↪ amplitude_length-1 downto 0);
        pipein_cycles         : in  std_logic_vector (
           ↪ frequency_length-1 downto 0);
        pipein_stb            : in  std_logic;
        pipein_cyc            : in  std_logic;
        pipeout_max_amplitude : out std_logic_vector (
           ↪ amplitude_length - 1 downto 0);
        pipeout_max_cycles    : out std_logic_vector (
           ↪ frequency_length-1 downto 0);
        pipeout_stb           : out std_logic;
        pipeout_cyc           : out std_logic;
        reset                 : in  std_logic;

        we_i    : in  std_logic;
        stb_i   : in  std_logic;
        dat_i   : in  std_logic_vector (15 downto 0);
        dat_o   : out std_logic_vector (15 downto 0);
        addr_i  : in  std_logic_vector (7 downto 0);
        ack_o   : out std_logic;
        error_o : out std_logic;
        cyc_i   : in  std_logic
        );
```

```
end max_detector_sweep;

architecture Behavioral of max_detector_sweep is

  signal is_starting : boolean := false;
  signal initial_run : boolean := true;
  signal invalidate : std_logic;
  signal settings : pipe_settings_array(0 to 0);
  signal cur_max        : std_logic_vector(amplitude_length -1 downto 0);
  signal cur_max_cycles : std_logic_vector(frequency_length -1 downto 0);

begin

  configs : entity pipes.wb_filter_template
    generic map(data_depth => 1,
                ram_intial => (0 => (others => '0')))
    port map(clk_i      => clk,
             we_i       => we_i,
             stb_i      => stb_i,
             dat_i      => dat_i,
             dat_o      => dat_o,
             addr_i     => addr_i,
             ack_o      => ack_o,
             error_o    => error_o,
             cyc_i      => cyc_i,
             res_i      => reset,
             invalidate => invalidate,
             ram        => settings
             );


  process (clk)
  begin
    if rising_edge(clk) then
      pipeout_stb <= '0';
      pipeout_cyc <= '0';
      if reset = '1' or invalidate = '1' then
        is_starting    <= true;
        cur_max        <= (others => '0');
        cur_max_cycles <= (others => '0');
        initial_run    <= true;
      else
        if pipein_stb = '1' and pipein_cyc = '1' then
          if initial_run then
```

```
            if pipein_cycles = settings(0)(frequency_length - 1
              ↪ downto 0) then
              initial_run <= false;
            end if;
            cur_max        <= pipein_amplitude;
            cur_max_cycles <= pipein_cycles;
          else
            if unsigned(pipein_amplitude) > unsigned(cur_max) then
              cur_max        <= pipein_amplitude;
              cur_max_cycles <= pipein_cycles;
            end if;

            if is_starting then
              if pipein_cycles /= settings(0)(frequency_length - 1
                ↪ downto 0) then
                is_starting <= false;
              end if;
            else
              if pipein_cycles = settings(0)(frequency_length - 1
                ↪ downto 0) then
                is_starting <= true;
                pipeout_max_amplitude <= cur_max;
                pipeout_max_cycles    <= cur_max_cycles;
                pipeout_stb           <= '1';
                pipeout_cyc           <= '1';
                cur_max               <= pipein_amplitude;
                cur_max_cycles        <= pipein_cycles;
              end if;
            end if;
          end if;
        end if;
      end if;
    end if;
  end process;
end Behavioral;
```

7. Maximum Detector

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity pipe_last_in_cyc is
  generic(amplitude_length : integer := 24;
          cycle_length     : integer := 16);
  port (CLK              : in  std_logic;
```

```
        RESET             : in  std_logic;
        pipein_stb        : in  std_logic;
        pipein_cyc        : in  std_logic;
        pipein_amplitude  : in  std_logic_vector (amplitude_length - 1 downto 0);
        pipein_cycle      : in  std_logic_vector (cycle_length - 1 downto 0);
        pipeout_stb       : out std_logic;
        pipeout_cyc       : out std_logic;
        pipeout_cycle     : out std_logic_vector (cycle_length - 1 downto 0);
        pipeout_amplitude : out std_logic_vector (amplitude_length - 1 downto 0));
end pipe_last_in_cyc;

architecture Behavioral of pipe_last_in_cyc is

  signal last_cycle     : std_logic_vector (cycle_length -1 downto 0);
  signal last_amplitude : std_logic_vector (amplitude_length -1 downto 0);
  signal first_run : boolean := true;

begin

  process(clk)
  begin
    if rising_edge(clk) then
      pipeout_stb <= '0';
      pipeout_cyc <= '0';
      if reset = '1' then
        first_run <= true;
      else
        if pipein_stb = '1' and pipein_cyc = '1' then
          first_run      <= false;
          last_cycle     <= pipein_cycle;
          last_amplitude <= pipein_amplitude;
          if not first_run then
            if last_cycle /= pipein_cycle then
              pipeout_amplitude <= last_amplitude;
              pipeout_cycle     <= last_cycle;
              pipeout_stb       <= '1';
              pipeout_cyc       <= '1';
            end if;
          end if;
        end if;
```

```
      end if;
    end if;
  end process;
end Behavioral;
```

8. Control Component for Sweep Based Measurements

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

library work;
use work.typedefs.all;

entity lockin_control is
  port (CLK120             : in  std_logic;
        reset              : in  std_logic;
        config_mem         : in  typ_config_mem(0 to 5);
        led                : out std_logic_vector(4 downto 0);
        output_enable      : out boolean;
        measure_enable     : out boolean;
        ad7760_sync_in     : out std_logic;
        pulsegen_pulse_out : in  std_logic;

        -- wb_AD7760
        wb_AD7760_stb_i    : out std_logic;
        wb_AD7760_ack_o    : in  std_logic;
        wb_AD7760_dat_i    : out std_logic_vector(15 downto 0);
        wb_AD7760_dat_o    : in  std_logic_vector(15 downto 0);
        wb_AD7760_addr_i   : out std_logic_vector(7 downto 0);
        wb_AD7760_we_i     : out std_logic;
        wb_AD7760_cyc_i    : out std_logic;
        -- wb_DAC
        wb_DAC_stb_i       : out std_logic;
        wb_DAC_ack_o       : in  std_logic;
        wb_DAC_dat_i       : out std_logic_vector(15 downto 0);
        wb_DAC_dat_o       : in  std_logic_vector(15 downto 0);
        wb_DAC_addr_i      : out std_logic_vector(7 downto 0);
        wb_DAC_we_i        : out std_logic;
        wb_DAC_cyc_i       : out std_logic;
        -- wb_pulsegen
        wb_pulsegen_stb_i  : out std_logic;
        wb_pulsegen_ack_o  : in  std_logic;
        wb_pulsegen_dat_i  : out std_logic_vector(15 downto 0);
        wb_pulsegen_dat_o  : in  std_logic_vector(15 downto 0);
```

```
        wb_pulsegen_addr_i  : out std_logic_vector(7 downto 0);
        wb_pulsegen_we_i    : out std_logic;
        wb_pulsegen_cyc_i   : out std_logic;
        -- wb_max
        wb_max_stb_i        : out std_logic;
        wb_max_ack_o        : in  std_logic;
        wb_max_dat_i        : out std_logic_vector(15 downto 0);
        wb_max_dat_o        : in  std_logic_vector(15 downto 0);
        wb_max_addr_i       : out std_logic_vector(7 downto 0);
        wb_max_we_i         : out std_logic;
        wb_max_cyc_i        : out std_logic
   --writing
   --serial_data_to_write : in serial_data(0 to 4);
   --write_data           : in boolean;
   --serial_data_to_write_out : out serial_data(0 to 4);
   --write_data_out                            : out boolean;
   --writer_busy: in boolean;
   --DAC_output_dat_o: out std_logic_vector ( 15 downto 0);
   --DAC_output_stb_o: out std_logic;
   --DAC_output_cyc_o: out std_logic
        );
end lockin_control;






architecture Behavioral of control is

  type rec_wb_connections is
  record
    stb_i   : std_logic;
    ack_o   : std_logic;
    dat_i   : std_logic_vector(15 downto 0);
    dat_o   : std_logic_vector(15 downto 0);
    addr_i  : std_logic_vector(7 downto 0);
    we_i    : std_logic;
    cyc_i   : std_logic;
  end record;

  type typ_wb_cores is (wb_AD7760, wb_DAC, wb_pulsegen, wb_max,
     ↪ wb_none);
  type typ_wb_conn_array is array(typ_wb_cores) of rec_wb_connections
     ↪ ;
  signal wb : typ_wb_conn_array;
```

```vhdl
type typ_wb_message is
record
  core   : typ_wb_cores;
  addr_i : std_logic_vector(7 downto 0);
  dat_i  : std_logic_vector(15 downto 0);
end record;

signal hold_mem : typ_config_mem(1 to 5);

type typ_message_list is array (integer range <>) of typ_wb_message ;
constant messages_setup : typ_message_list(0 to 5) :=
  (0          => (core => wb_AD7760,
         addr_i => std_logic_vector(to_unsigned(0, 8)),
         dat_i  => std_logic_vector(to_unsigned(16#A000#, 16))),
   1          => (core => wb_AD7760,
         addr_i => std_logic_vector(to_unsigned(1, 8)),
         dat_i  => std_logic_vector(to_signed(0, 16))),

   2          => (core => wb_AD7760,
         addr_i => std_logic_vector(to_unsigned(2, 8)),
         dat_i  => (15 => '1', 14 => '0', 13 => '0', 12 => '1', 11 => '1', others => '0')),
   3          => (core => wb_pulsegen,
         addr_i => std_logic_vector(to_unsigned(0, 8)),
         dat_i  => std_logic_vector(to_unsigned(1200, 16))),
   4          => (core => wb_pulsegen,
         addr_i => std_logic_vector(to_unsigned(1, 8)),
         dat_i  => std_logic_vector(to_unsigned(20, 16))),
   5          => (core => wb_DAC,
         addr_i => (others => '0'),
         dat_i  => (0 => '1', 1 => '1', 2 => '1', 3 => '1', others => '0')));

signal messages_start_meas : typ_message_list(0 to 2) :=
  (0          => (core => wb_pulsegen,
         addr_i => std_logic_vector(to_unsigned(0, 8)),
         dat_i  => config_mem(1)),
   1          => (core => wb_pulsegen,
         addr_i => std_logic_vector(to_unsigned(1, 8)),
         dat_i  => config_mem(5)),
   2          => (core => wb_max,
         addr_i => std_logic_vector(to_unsigned(0, 8)),
         dat_i  => config_mem(1)));
```

```
  signal start_measure_counter    : integer range 0 to
     ↪ messages_start_meas'length + 1;
  signal measure_counter          : integer range 0 to 2**15;
  signal next_pulse_rate          : unsigned(15 downto 0)   := (
     ↪ others => '0');
  signal frequency_step_counter   : integer range 0 to 2047 := 0;
  signal sweep_counter            : integer range 0 to 255;
  signal startup_counter          : integer range 0 to 1100 := 0;
  signal started                  : boolean                 := true;
  signal enabled                  : boolean                 := false;
  signal sweep                    : boolean                 := false;
  signal sweep_bidirectional      : boolean;
  signal sweep_going_up           : boolean;
  signal pos_edge_pulse           : boolean;
  signal old_pulse                : std_logic;
  type typ_main_states is (main_startup, main_measure, main_wait,
     ↪ main_start_measure);
  signal main_state               : typ_main_states         :=
     ↪ main_wait;
  signal decim_counter            : unsigned(2 downto 0)    := (
     ↪ others => '0');
  signal write_again              : boolean                 := false;
  signal serial_data_to_write_reg : serial_data(0 to 4);
  signal write_data_out_int       : boolean;
  signal current_maximum          : std_logic_vector(23 downto 0);
  signal reset_maximum            : std_logic               := '0';
  signal max_data_valid           : std_logic               := '0';


begin

  wb(wb_AD7760).ack_o   <= wb_AD7760_ack_o;
  wb(wb_AD7760).dat_o   <= wb_AD7760_dat_o;
  wb(wb_DAC).ack_o      <= wb_DAC_ack_o;
  wb(wb_DAC).dat_o      <= wb_DAC_dat_o;
  wb(wb_pulsegen).ack_o <= wb_pulsegen_ack_o;
  wb(wb_pulsegen).dat_o <= wb_pulsegen_dat_o;
  wb_AD7760_stb_i       <= wb(wb_AD7760).stb_i;
  wb_AD7760_cyc_i       <= wb(wb_AD7760).cyc_i;
  wb_AD7760_addr_i      <= wb(wb_AD7760).addr_i;
  wb_AD7760_we_i        <= wb(wb_AD7760).we_i;
  wb_AD7760_dat_i       <= wb(wb_AD7760).dat_i;
  wb_DAC_stb_i          <= wb(wb_DAC).stb_i;
  wb_DAC_cyc_i          <= wb(wb_DAC).cyc_i;
  wb_DAC_addr_i         <= wb(wb_DAC).addr_i;
  wb_DAC_we_i           <= wb(wb_DAC).we_i;
```

```
wb_DAC_dat_i            <= wb(wb_DAC).dat_i;
wb_pulsegen_stb_i       <= wb(wb_pulsegen).stb_i;
wb_pulsegen_cyc_i       <= wb(wb_pulsegen).cyc_i;
wb_pulsegen_addr_i      <= wb(wb_pulsegen).addr_i;
wb_pulsegen_we_i        <= wb(wb_pulsegen).we_i;
wb_pulsegen_dat_i       <= wb(wb_pulsegen).dat_i;
wb_max_dat_i            <= wb(wb_max).dat_i;
wb_max_stb_i            <= wb(wb_max).stb_i;
wb_max_cyc_i            <= wb(wb_max).cyc_i;
wb_max_addr_i           <= wb(wb_max).addr_i;
wb_max_we_i             <= wb(wb_max).we_i;
wb_max_dat_i            <= wb(wb_max).dat_i;


enabled                        <= true when config_mem(0)(0) = '1'
   ↪ else false;
sweep                          <= true when config_mem(0)(1) = '1'
   ↪ else false;
LED(4)                         <= '1'  when sweep_bidirectional
   ↪ else '0';
messages_start_meas(0).dat_i <= config_mem(1);
messages_start_meas(1).dat_i <= config_mem(5);
messages_start_meas(2).dat_i <= config_mem(1);
pos_edge_pulse                 <= old_pulse = '0' and
   ↪ pulsegen_pulse_out = '1';

main_proc : process(CLK120)
  variable cur_mes        : typ_wb_message;
  variable cur_core       : rec_wb_connections;
  variable slv_next_pulse : std_logic_vector(15 downto 0);
begin
  if rising_edge(CLK120) then

    led(3 downto 0) <= (others => '0');
    measure_enable  <= false;
    for wb_core in typ_wb_cores loop
      wb(wb_core).stb_i  <= '0';
      wb(wb_core).dat_i  <= (others => '0');
      wb(wb_core).we_i   <= '0';
      wb(wb_core).cyc_i  <= '0';
      wb(wb_core).addr_i <= (others => '0');
    end loop;
    startup_counter       <= 0;
    start_measure_counter <= 0;
    output_enable         <= false;
    if RESET = '1' then
```

```
      main_state      <= main_wait;
      ad7760_sync_in <= '1';
      started         <= false;
      old_pulse       <= '0';
      sweep_counter   <= 0;
      sweep_going_up <= true;
    else
      old_pulse <= pulsegen_pulse_out;
      case main_state is  -- (main_startup, main_measure, main_wait
         ↪ )
        when main_startup =>
          led(0)          <= '1';
          startup_counter <= startup_counter + 1;
          if startup_counter < messages_setup'length then
            cur_mes                  := messages_setup(
               ↪ startup_counter);
            wb(cur_mes.core).cyc_i  <= '1';
            wb(cur_mes.core).stb_i  <= '1';
            wb(cur_mes.core).addr_i <= cur_mes.addr_i;
            wb(cur_mes.core).dat_i  <= cur_mes.dat_i;
            wb(cur_mes.core).we_i   <= '1';
          end if;
          if startup_counter = 1000 then
            ad7760_sync_in <= '0';
          end if;
          if startup_counter = 1001 then
            ad7760_sync_in <= '1';
            started        <= true;
            if enabled then
              main_state <= main_start_measure;
            else
              main_state <= main_wait;
            end if;
          end if;

        when main_wait =>
          led(1) <= '1';
          if enabled then
            if started then
              main_state <= main_start_measure;
            else
              main_state <= main_startup;
            end if;
          end if;

        when main_start_measure =>
```

```
          sweep_counter          <= 0;
          frequency_step_counter <= 0;
          led(2)                 <= '1';
          start_measure_counter  <= start_measure_counter+1;
          sweep_going_up         <= config_mem(0)(3) = '1';
          sweep_bidirectional    <= config_mem(0)(2) = '1';
          if start_measure_counter = 0 then
            for i in 1 to config_mem'length -1 loop
              hold_mem(i) <= config_mem(i);
            end loop;
          else
            next_pulse_rate <= unsigned(config_mem(1));
          end if;
          if start_measure_counter /= messages_start_meas'length
             ↪ then
            cur_mes                   := messages_start_meas(
               ↪ start_measure_counter);
            wb(cur_mes.core).cyc_i  <= '1';
            wb(cur_mes.core).stb_i  <= '1';
            wb(cur_mes.core).addr_i <= cur_mes.addr_i;
            wb(cur_mes.core).dat_i  <= cur_mes.dat_i;
            wb(cur_mes.core).we_i   <= '1';
          else
            main_state     <= main_measure;
            ad7760_sync_in <= '1';
          end if;

        when main_measure =>
          measure_enable <= true;
          ad7760_sync_in <= '1';
          led(3)         <= '1';
          output_enable  <= true;
          if not enabled then
            main_state <= main_wait;
          end if;
          if pos_edge_pulse then
            measure_counter <= 0;
            sweep_counter   <= sweep_counter + 1;
            if sweep_counter = unsigned(hold_mem(4)) then
              if sweep then
                if not sweep_going_up then
                  next_pulse_rate <= next_pulse_rate - unsigned(
                     ↪ hold_mem(2));
                else
                  next_pulse_rate <= next_pulse_rate + unsigned(
                     ↪ hold_mem(2));
```

```
                    end if;
                    frequency_step_counter <= frequency_step_counter +
                       ↪ 1;
                  end if;
                  sweep_counter          <= 0;
                  measure_counter        <= 0;
                  wb(wb_pulsegen).cyc_i  <= '1';
                  wb(wb_pulsegen).stb_i  <= '1';
                  wb(wb_pulsegen).addr_i <= std_logic_vector(
                     ↪ to_unsigned(0, 8));
                  wb(wb_pulsegen).dat_i  <= std_logic_vector(
                     ↪ next_pulse_rate);
                  wb(wb_pulsegen).we_i   <= '1';
                  if frequency_step_counter = unsigned(hold_mem(3))
                     ↪ then
                    frequency_step_counter <= 0;
                    if not sweep_bidirectional then
                      next_pulse_rate <= unsigned(config_mem(1));
                    else
                      sweep_going_up  <= not sweep_going_up;
                      next_pulse_rate <= next_pulse_rate;
                      if sweep_going_up then
                      end if;
                    end if;
                  end if;
                end if;
              end if;
          end case;
        end if;
      end if;
  end process;
end Behavioral;
```

9. Scope for Pipes in Test Bench

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

use std.textio.all;

entity pipe_scope_int is
  generic (Name : string  := "unnamed";
           size : integer := 24;
```

```
          sign : boolean := false);
  port (CLK  : in std_logic;
        stb  : in std_logic;
        cyc  : in std_logic;
        data : in std_logic_vector(size-1 downto 0));
end pipe_scope_int;

architecture Behavioral of pipe_scope_int is

  type textfile is file of string;      -- file of text
  file myfile : text open write_mode is Name&".txt";

  function chr(sl : std_logic) return character is
    variable c : character;
  begin
    case sl is
      when 'U' => c := 'U';
      when 'X' => c := 'X';
      when '0' => c := '0';
      when '1' => c := '1';
      when 'Z' => c := 'Z';
      when 'W' => c := 'W';
      when 'L' => c := 'L';
      when 'H' => c := 'H';
      when '-' => c := '-';
    end case;
    return c;
  end chr;

  function str(slv : std_logic_vector) return string is
    variable result : string (1 to slv'length);
    variable r      : integer;
  begin
    r := 1;
    for i in slv'range loop
      result(r) := chr(slv(i));
      r         := r + 1;
    end loop;
    return result;
  end str;

begin

--pragma synthesis_off
  process (CLK)
    variable vDataoutline : line;
```

```
  begin
    if rising_edge(CLK) then
      if (cyc = '1' and stb = '1') then
        write(vDataoutline, real'image(real(now/ 1 ns))&";");
        if sign then
          write(vDataoutline, "s" & str(data));
        else
          write(vDataoutline, "u" & str(data));
        end if;
        writeline(myfile, vDataoutline);
      end if;
    end if;
  end process;
--pragma synthesis_on
end Behavioral;
```

10. MojoConnectToolBox

10.1. MojoConnectToolBox.h

```
/*-------------------------------------------------------------*/
/*                                                             */
/*       .....::::: MojoConnectToolBox.h :::::.....            */
/*                         C version                           */
/*                                                             */
/*  Copyright (c) 2015 Benedikt Steindl, All rights reserved.  */
/*                                                             */
/*             mailaddress : benedikt.steindl@gmail.com        */
/*          postal address : Reisenbauer-Ring 3/1/15           */
/*                           2351-Wr.Neudorf                   */
/*                           Austria                           */
/*                                                             */
/*-------------------------------------------------------------*/

#ifndef MOJOCONNECTTOOLBOX_H
#define MOJOCONNECTTOOLBOX_H

#ifdef __gnu_linux__
#   define DLLEXPORT            /* empty definition */
#else /* MS Windows */
#   include <windows.h>
#   include "ftd2xx_win.h"
#   define DLLEXPORT __declspec(dllexport)
#endif
```

```
#include "ftd2xx.h"
#include <stdint.h>


typedef struct rx_message {
  unsigned char index;
  unsigned short tau;
  int32_t amplitude;
} rx_message;




DLLEXPORT extern FT_HANDLE ftHandle ;        /* handle to the device */
DLLEXPORT extern FT_STATUS ftStatus ;        /* status info of the device */
DLLEXPORT extern int stdTxBufferSize;        /* [bytes] : size of the
    ↪ transmitter buffer */
DLLEXPORT extern int stdRxBufferSize;        /* [bytes] : size of the
    ↪ receiver buffer */
DLLEXPORT extern int MaxTxBufferSize;        /* [bytes] : maximum size of the
    ↪  transmitter buffer */
DLLEXPORT extern int MaxRxBufferSize;        /* [bytes] : maximum size of the
    ↪  receiver buffer */
DLLEXPORT extern char* DataBuffer;           /* intermediate buffer for the
    ↪ received data set */
DLLEXPORT extern int DataBufferSize;         /* size of the intermediate
    ↪ buffer */
DLLEXPORT extern int PopIndex;               /* index for the pop */




DLLEXPORT int SetStdTxBufferSize(int TxSize);                              /*
    ↪  sets the standard size of the transmitter buffer */
DLLEXPORT int GetStdTxBufferSize();                                        /*
    ↪  gets the standard size of the transmitter buffer */
DLLEXPORT int SetStdRxBufferSize(int RxSize);                              /*
    ↪  sets the standard size of the receiver buffer */
DLLEXPORT int GetStdRxBufferSize();                                        /*
    ↪  gets the standard size of the transmitter buffer */
DLLEXPORT int OpenConnection();                                            /*
    ↪  opens the connection to the ft2232h chip */
```

```
DLLEXPORT int CloseConnection();                                        /*
    ↪  closes the connection to the ft2232h chip */
DLLEXPORT int ProgramFIFOChip();                                        /*
    ↪  reprograms the ft2232h to ft245 synchronous FIFO mode */
DLLEXPORT int FIFOStatus(unsigned int * RxBytes, unsigned int * TxBytes); /*
    ↪  gives over the number of bytes in the receiver and transmitter buffer
    ↪  */
DLLEXPORT int TransmitData(char * TxData, int lengthTxData);            /*
    ↪  transmitts a data set of the given size */
DLLEXPORT char* ReceiveData(char * RxData, int * lengthRxData);         /*
    ↪  receives a data set and gives over its size */
DLLEXPORT int TransmitDataPackage(char TxBuffer[], int TxBufferSize);   /*
    ↪  transmitts a data package of the given size */
DLLEXPORT int ReceiveDataPackage(char* RxBuffer, int RxBufferSize);     /*
    ↪  receives a data package and gives over its size */
DLLEXPORT int PopFromBuffer(char* DataByte);                            /*
    ↪  pops a byte of the data in the internal buffer */
DLLEXPORT int PurgeBuffers();                                           /*
    ↪  purge the buffers of the FIFO chip */
DLLEXPORT int PurgeTxBuffer();                                          /*
    ↪  purge the TxBuffer of the FIFO chip */
DLLEXPORT int PurgeRxBuffer();                                          /*
    ↪  purge the RxBuffer of the FIFO chip */
DLLEXPORT int DecodeSingleMessage(char InputBuffer[], rx_message *
    ↪ DecodedMessage); /* decode a single message*/
DLLEXPORT int DecodeMessages(char InputBuffer[],int InputBufferSize,
    ↪ rx_message** OutputBuffer, int* OutputBufferSize, char*
    ↪ RemainingBuffer, int* RemainingBufferSize);/*Decodes messages into
    ↪ index, tau, X, Y signal. RemainingBufferSize must be at least 31 bytes
    ↪  large*/
DLLEXPORT int ReceiveMessages(rx_message ** Messages, int * lengthMessages);
    ↪ /*receives messages. If no other rx function is called in between
    ↪ successive calls, than the few leftover bytes that don't make a full
    ↪ message are handled correctly*/


#endif
```

10.2. MojoConnectToolBox.c

```
/*-----------------------------------------------------------*/
/*                                                           */
/*        .....:::::: MojoConnectToolBox.c ::::::.....       */
/*                         C version                         */
/*                                                           */
```

```
/*  Copyright (c) 2015 Benedikt Steindl, All rights reserved.   */
/*                                                              */
/*             mailaddress : benedikt.steindl@gmail.com         */
/*          postal address : Reisenbauer-Ring 3/1/15            */
/*                           2351-Wr.Neudorf                    */
/*                           Austria                            */
/*                                                              */
/*--------------------------------------------------------------*/

#ifdef __gnu_linux__
#   include <unistd.h>
#   include <pthread.h>
#   define SLEEP(x) (sleep(x))  /* sleep() on LINUX and UNIX */
#   define DLLEXPORT            /* empty definition */
#else /* MS Windows */
#   include <windows.h>
#   define SLEEP(x) (Sleep(x))  /* Sleep() on Windows */
#   define DLLEXPORT __declspec(dllexport)
#endif

#include "MojoConnectToolBox.h"
#include "ftd2xx.h"
#include <math.h>
#include <stdio.h>      /* printf, scanf, NULL */
#include <stdlib.h>     /* malloc, free, rand */
#include <string.h>
#include <stdbool.h>
#include <stdint.h>


const unsigned char START_BYTE = 0x12;
const unsigned char STOP_BYTE = 0x13;
const unsigned char ESC_BYTE = 0x7D;




DLLEXPORT FT_HANDLE ftHandle  =   0;    /* handle to the device */
DLLEXPORT FT_STATUS ftStatus  =   0;    /* status info of the device
    ↪ */
DLLEXPORT int stdTxBufferSize = 510;    /* [bytes] : size of the
    ↪ transmitter buffer */
DLLEXPORT int stdRxBufferSize = 510;    /* [bytes] : size of the
    ↪ receiver buffer */
```

```
DLLEXPORT int MaxTxBufferSize = 510;     /* [bytes] : maximum size of the transmitter buffer */
DLLEXPORT int MaxRxBufferSize = 510;     /* [bytes] : maximum size of the receiver buffer */
DLLEXPORT char* DataBuffer = NULL;       /* intermediate buffer for the received data set */
DLLEXPORT int DataBufferSize = 0;        /* size of the intermediate buffer */
DLLEXPORT int PopIndex = 0;              /* index for the pop */

DLLEXPORT int SetStdTxBufferSize(int TxSize) {
  if (TxSize <= MaxTxBufferSize) {
    stdTxBufferSize = TxSize;
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int GetStdTxBufferSize() {
  return stdTxBufferSize;
}

DLLEXPORT int SetStdRxBufferSize(int RxSize) {
  if (RxSize <= MaxRxBufferSize) {
    stdRxBufferSize = RxSize;
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int GetStdRxBufferSize() {
  return stdRxBufferSize;
}

DLLEXPORT int OpenConnection() {

  ftStatus = FT_OpenEx("RE-PTIR A", FT_OPEN_BY_DESCRIPTION,&ftHandle);
  if (ftStatus == FT_OK) {
    return 1;
  } else {
    return 0;
  }
}
```

```
DLLEXPORT int CloseConnection() {
  ftStatus = FT_Close(ftHandle);
  if (ftStatus == FT_OK) {
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int ProgramFIFOChip() {
  UCHAR Mask = 0xff;
  UCHAR Mode;
  UCHAR LatencyTimer = 16; /* default setting is 16 */
  Mode = 0x00; /* reset mode */
  ftStatus = FT_SetBitMode(ftHandle, Mask, Mode);
  SLEEP(1);
  Mode = 0x40; /* 245 synchronous FIFO mode */
  ftStatus = FT_SetBitMode(ftHandle, Mask, Mode);
  if (ftStatus == FT_OK) {
    ftStatus = FT_SetLatencyTimer(ftHandle, LatencyTimer);
    ftStatus = FT_SetUSBParameters(ftHandle,0x10000,0x10000);
    ftStatus = FT_SetFlowControl(ftHandle,FT_FLOW_RTS_CTS,0,0);
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int FIFOStatus(unsigned int * RxBytes, unsigned int *
    ↪ TxBytes) {
  DWORD EventDWord;
  DWORD rBytes;
  DWORD tBytes;
  ftStatus = FT_GetStatus(ftHandle,&rBytes,&tBytes,&EventDWord);
  *RxBytes = rBytes;
  *TxBytes = tBytes;
  if (ftStatus == FT_OK) {
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int TransmitData(char * TxData, int lengthTxData) {
  int CompTrans = 1;
```

```
  int SpecTrans = 1;
  int TransSuccess;
  int NrOfCompleteBuffers = trunc(lengthTxData / stdTxBufferSize);
  int SpecialTxBufferSize = lengthTxData % stdTxBufferSize;
  int dataIndex = 0;
  int extractIndex = 0;
  char* CompleteTxBuffer = (char*) malloc(stdTxBufferSize);
  for (int i = 0; i < NrOfCompleteBuffers; i++) { /* transmitts
      ↪ packages of size stdTxBufferSize */
    extractIndex = i * stdTxBufferSize;
    for (dataIndex = extractIndex; dataIndex < extractIndex +
        ↪ stdTxBufferSize; dataIndex++)
      CompleteTxBuffer[dataIndex - extractIndex] = TxData[dataIndex];
    CompTrans = TransmitDataPackage(CompleteTxBuffer, stdTxBufferSize
        ↪ );
  }
  free(CompleteTxBuffer);
  if (SpecialTxBufferSize > 0) { /* transmitts a package of size <
      ↪ stdTxBufferSize */
    char* SpecialTxBuffer;
    SpecialTxBuffer = (char*) malloc(SpecialTxBufferSize);
    for (int i = 0; i < SpecialTxBufferSize; i++)
      SpecialTxBuffer[i] = TxData[i + NrOfCompleteBuffers *
          ↪ stdTxBufferSize];
    SpecTrans = TransmitDataPackage(SpecialTxBuffer,
        ↪ SpecialTxBufferSize);
    free(SpecialTxBuffer);
    SpecialTxBuffer = 0;
  }
  if (CompTrans == 1 && SpecTrans == 1)
    return 1;
  else
    return 0;
}

DLLEXPORT int TransmitDataPackage(char TxBuffer[], int TxBufferSize)
    ↪ {
  DWORD EventDWord;
  DWORD RxBytes;
  DWORD TxBytes;
  DWORD BytesWritten;
  if (TxBufferSize <= stdTxBufferSize) {
    while (1) {
      ftStatus = FT_GetStatus(ftHandle,&RxBytes,&TxBytes,&EventDWord)
          ↪ ;
      if ((ftStatus == FT_OK) && (TxBytes == 0)) {
```

```
    ftStatus = FT_Write(ftHandle, TxBuffer, TxBufferSize, &
       ↪ BytesWritten);
    if (ftStatus == FT_OK && BytesWritten == TxBufferSize) {
      return 1;
    } else {
      return 0;
    }
      }
    }
  } else {
    return 0;
  }
}

short reverse_tau(char *c)
{
  short new_tau;
  char * p_new_tau = (char*) & new_tau;
  //printf("\n pointer: %p, value: ", c);
  //for (int k = 0; k<2; k++){
  //    printf("%d ", c[k]);
  //}
  //printf("\n");
  p_new_tau[0] = c[1];
  p_new_tau[1] = c[0];
  return new_tau;

}

long long reverse_XY(char *c){
  long long new_XY = 0;
  char * p_newXY = (char *) & new_XY;
  //printf("\n pointer: %p, value: ", c);
  //for (int k = 0; k<6; k++){
  //    printf("%d ", c[k]);
  //}

  for (int i = 0; i<6; i++){
    p_newXY[i] = c[5-i];
  }
  //printf("\nbefore: %llx \n", new_XY);
  if (c[0] & 0x80){ // if leading bit is 1, then this has to be a
      ↪ negative number
    unsigned char ones = 0xFF;
    p_newXY[7] = (char) ones;
    p_newXY[6] = (char) ones;
```

```
      //printf("\n smaller than 0\n");
    }else{
      p_newXY[7] = 0;
      p_newXY[6] = 0;
    }
    //printf("\nafter %llx \n", new_XY);
    //printf("\n %x \n", p_newXY[7]);
    //printf("\n %x \n", p_newXY[6]);
    return new_XY;
}


int32_t reverse_to_long(char *c){
    long new_long = 0;
    char * p_newlong = (char *) & new_long;
    //printf("\n pointer: %p, value: ", c);
    //for (int k = 0; k<6; k++){
    //    printf("%d ", c[k]);
    //}

    for (int i = 0; i<3; i++){
      p_newlong[i] = c[2-i];
    }
    //printf("\nbefore: %llx \n", new_XY);
    if (c[0] & 0x80){ // if leading bit is 1, then this has to be a negative number
      unsigned char ones = 0xFF;
      p_newlong[3] = (char) ones;
      //printf("\n smaller than 0\n");
    }else{
      p_newlong[3] = 0;
    }
    //printf("\nafter %llx \n", new_XY);
    //printf("\n %x \n", p_newXY[7]);
    //printf("\n %x \n", p_newXY[6]);
    return new_long;
}




DLLEXPORT int DecodeSingleMessage(char InputBuffer[], rx_message * DecodedMessage){
    memcpy (&( (*DecodedMessage).index),InputBuffer, 1);
    (*DecodedMessage).tau = reverse_tau(InputBuffer + 1);
    int32_t amplitude = reverse_to_long(InputBuffer + 3);
```

```
  (*DecodedMessage).amplitude = amplitude;
  return 1;
}




DLLEXPORT int DecodeMessages(char * InputBuffer, int InputBufferSize, rx_message** OutputBuffer, int* OutputBufferSize, char* RemainingBuffer, int* RemainingBufferSize){
  int byte_counter = 0;
  bool esced = false;
  bool inmessage = false;
  char cur_byte;
  int message_count = -1;
  char  cur_message[15];
  rx_message output_messages[(InputBufferSize / 6)];
  for (int i=0; i<InputBufferSize; i++){
    cur_byte = InputBuffer[i];
    if (! esced){
      // first handle all the special bytes, only data bytes get past these 4 ifs
      if (cur_byte == START_BYTE) {
    byte_counter = 0;
    if (! inmessage) {
      message_count += 1;
    }
    inmessage = true;
    continue;
      }
      if (cur_byte == STOP_BYTE) {
    if (inmessage && byte_counter == 6) {
      DecodeSingleMessage(cur_message, output_messages + message_count);
      rx_message mes;
      mes = *(output_messages +message_count);
      inmessage = false;
    };
    continue;
      }
      if (cur_byte == ESC_BYTE){
    esced = true;
    continue;
      }
    }
```

```
    if (byte_counter == 6) {
      inmessage = false;
      byte_counter = 0;
    }
    if (inmessage){
      cur_message[byte_counter] = cur_byte;
      byte_counter += 1;
    }
    esced = false;
  }
  rx_message *DataOutputBuffer;
  char* RemainingBufferData;
  if (message_count > 0 || (message_count==0 && ! inmessage)){
    if (inmessage){
      *OutputBufferSize = message_count;
      DataOutputBuffer = (rx_message*) malloc(*OutputBufferSize *
          ↪ sizeof(rx_message));
      *RemainingBufferSize = byte_counter;
      memcpy(RemainingBuffer, &START_BYTE,1);
      memcpy(RemainingBuffer +1, cur_message, byte_counter);
    }else
      {
    *OutputBufferSize = message_count + 1;
    DataOutputBuffer = (rx_message*) malloc(*OutputBufferSize *
        ↪ sizeof(rx_message));
    *RemainingBufferSize = 0;
      }
    memcpy(DataOutputBuffer, output_messages, *OutputBufferSize *
        ↪ sizeof(rx_message));
    *OutputBuffer = DataOutputBuffer;
    return 1;
  }
  return 0;
}







DLLEXPORT char* ReceiveData(char * RxData, int * lengthRxData) {
  int RecSucc = 0;
  int RxBufferSize;
  char* interBuffer = NULL;
```

```
  unsigned int rxBytes, txBytes;
  if (FIFOStatus(&rxBytes, &txBytes)) {
    if (rxBytes > 0) {
      if (rxBytes >= stdRxBufferSize) {
    RxBufferSize = stdRxBufferSize;
      } else {
    RxBufferSize = rxBytes;
      }
      if (*lengthRxData != RxBufferSize) {
    interBuffer = (char*) realloc(RxData, RxBufferSize);
    *lengthRxData = RxBufferSize;
    if (interBuffer != NULL) {
      RxData = interBuffer;
    } else {
      printf("Memory␣(re)allocation␣FAILED!\n");
      return NULL;
    }
      }
      if (ReceiveDataPackage(RxData, RxBufferSize)) {
    return RxData;
      } else {
    return NULL;
      }
    } else {
      return NULL;
    }
  } else {
    return NULL;
  }
}

DLLEXPORT int ReceiveDataPackage(char* RxBuffer, int RxBufferSize) {
  DWORD EventDWord;
  DWORD RxBytes = 0;
  DWORD TxBytes = 0;
  DWORD BytesReceived;
  if (RxBufferSize <= stdRxBufferSize) {
    ftStatus = FT_GetStatus(ftHandle,&RxBytes,&TxBytes,&EventDWord);
    if ((ftStatus == FT_OK) && (RxBytes >= RxBufferSize))
      ftStatus = FT_Read(ftHandle,RxBuffer,RxBufferSize,&
          ↪ BytesReceived);
    if ((ftStatus == FT_OK) && (BytesReceived == RxBufferSize)) {
      return 1;
    } else {
      return 0;
    }
```

```
    } else {
      return 0;
    }
}

DLLEXPORT int PopFromBuffer(char* DataByte) {
    unsigned int rxBytes;
    unsigned int txBytes;
    if ((DataBuffer == NULL) && (FIFOStatus(&rxBytes, &txBytes))) {
      if (rxBytes > 0) {
        DataBufferSize = rxBytes;
        DataBuffer = (char*) malloc(DataBufferSize);
        DataBuffer = ReceiveData(DataBuffer, &DataBufferSize);
        if (DataBuffer == NULL)
      return 0;
        PopIndex = 0;
        *DataByte = DataBuffer[PopIndex];
        PopIndex++;
        if (PopIndex == DataBufferSize) {
      free(DataBuffer);
      DataBuffer = NULL;
        }
        return 1;
      } else {
        return 0;
      }
    } else {
      *DataByte = DataBuffer[PopIndex];
      PopIndex++;
      if (PopIndex == DataBufferSize) {
        free(DataBuffer);
        DataBuffer = NULL;
      }
      return 1;
    }
}

DLLEXPORT int PurgeBuffers() {
    if (PurgeTxBuffer() && PurgeRxBuffer()) {
      return 1;
    } else {
      return 0;
    }
}

DLLEXPORT int PurgeTxBuffer() {
```

```
  ftStatus = FT_Purge(ftHandle, FT_PURGE_TX);
  if (ftStatus == FT_OK) {
    //sleep(1);
    return 1;
  } else {
    return 0;
  }
}

DLLEXPORT int PurgeRxBuffer() {
  ftStatus = FT_Purge(ftHandle, FT_PURGE_RX);
  if (ftStatus == FT_OK) {
    //sleep(1);
    return 1;
  } else {
    return 0;
  }
}




char remaining_chars[31] = ""; //one message can be at most 30 bytes
    ↪ long (+ start_byte: 31)
int remaining_char_count = 0;

DLLEXPORT int ReceiveMessages(rx_message ** Messages, int *
    ↪ lengthMessages){
  unsigned int rxBytes, txBytes;
  if (FIFOStatus(&rxBytes, &txBytes)){
    if (rxBytes > 0) {
      int totalBytes = rxBytes + remaining_char_count;
      char * alldata = (char *) malloc(totalBytes);
      memcpy(alldata, remaining_chars, remaining_char_count);
      //printf("total length bytes= %d\n", totalBytes);
      int ftread_status;
      DWORD BytesReceived;
      ftread_status  =FT_Read(ftHandle,alldata + remaining_char_count
          ↪ ,rxBytes,&BytesReceived);
      //printf("FT_Read Status = %d, expected = %d, got = %d\n",
          ↪ ftread_status, rxBytes, BytesReceived);
      /*printf("----------------- all data");
    for (int k = 0; k< totalBytes; k++){
    printf("%d ", (unsigned char ) alldata[k]);
    }
```

```
    printf("\n");*/
      rx_message * DecodedMessages;
      int lengthDecodedMessages = 0;
      DecodeMessages(alldata, totalBytes, &DecodedMessages, &lengthDecodedMessages, remaining_chars, &remaining_char_count);
      *Messages = DecodedMessages;
      *lengthMessages = lengthDecodedMessages;
      return 1;
    }else{
      return 0;
    }
  }else
    {
      return 0;
    }
}
```

10.3. FT2232H.py

```
from __future__ import print_function

import ctypes
from ctypes import cdll, Structure, c_ubyte, c_ushort, c_long, POINTER, c_int, cast,c_int32, byref
from sys import platform
import os
from time import sleep

from collections import namedtuple

from multiprocessing import Process, Pipe, Value, Queue, Array, TimeoutError
from multiprocessing.queues import Empty

from enum import Enum

dllpath =os.path.dirname(os.path.abspath(__file__))+'/../dll/MojoConnectToolBox'

import logging


class rx_message(Structure):
```

```
    _fields_ =[("index", c_ubyte), ("tau", c_ushort), ("amplitude", c_int32)]
    def __repr__(self):
        return "rx_message(index=%d, tau=%d, amplitude=%d)"%(self.index, self.tau,self.amplitude)


rx_message_python = namedtuple("rx_message_python", "index tau amplitude")



class FT2232HError(Exception):
    def __init__(self, *args, **kwargs):
        super(FT2232HError, self).__init__(*args, **kwargs)



class FT2232H (object):

    def __init__(self, *args, **kwargs):
        #TODO: allow selecting specific FT2232H
        #TODO: add 64 bit versions of windows dll and version check
        if platform.startswith("linux"):
            self.MojoConnectToolBox = cdll.LoadLibrary(dllpath+".so")
        elif platform.startswith("win"):
            self.MojoConnectToolBox = cdll.LoadLibrary(dllpath+".dll")
        else:
            raise Exception("No dll for os type: " + platform)
        self.MojoConnectToolBox.ReceiveMessages.argtypes = [POINTER(POINTER(rx_message)), POINTER(c_int)]
        self.connected = False



    def connect(self):
        """
        connect to FT2232H. If this doesn't work, there might be permission problems (under Linux).
        Try running with "sudo".
        """
        if not (self.MojoConnectToolBox.OpenConnection()==1):
            raise FT2232HError("Could not connect.")
        if not(self.MojoConnectToolBox.ProgramFIFOChip() == 1):
```

```
            raise FT2232HError("Programming Chip to synchronous FIFO was unsuccessful.")
        self.connected = True

    def purge_buffer(self):
        if self.connected:
            self.MojoConnectToolBox.PurgeRxBuffer()

    def purge_buffers(self):
        if self.connected:
            self.MojoConnectToolBox.PurgeBuffers()

    def write(self, bytes):
        """
        writes bytes in 'bytes' to the chip.
        'bytes' is a byte string (e.g. python 2.x str())
        """
        TxData = ctypes.c_char_p(bytes)
        TxDataSize = ctypes.c_int(len(bytes) + 1)
        if not (self.MojoConnectToolBox.TransmitData(TxData, TxDataSize) == 1):
            raise FT2232HError("Unable to transmit data" + repr(bytes))

    def read(self):
        """
        reads all available date from FT2232H and returns it as byte string.
        """
        rxBytes = ctypes.c_uint()
        txBytes = ctypes.c_uint()
        self.MojoConnectToolBox.FIFOStatus(ctypes.byref(rxBytes), ctypes.byref(txBytes))

        DataSetSize = ctypes.c_int(510)
        retstr = ""
        while (rxBytes.value != 0) :
            self.MojoConnectToolBox.FIFOStatus(ctypes.byref(rxBytes),  ctypes.byref(txBytes))
            if (rxBytes.value > 510) :
                DataSetSize.value = 510
            else :
                DataSetSize.value = rxBytes.value
            DataSet = ctypes.create_string_buffer(DataSetSize.value+2) # allocate necessary memory
```

```
            if (self.MojoConnectToolBox.ReceiveDataPackage(ctypes.byref(DataSet), DataSetSize) == 0): # int ReceiveDataPackage(char RxBuffer[], int RxBufferSize);
                raise FT2232HError("Unable to read data!")
            #print (len(DataSet.raw), repr(DataSet.raw))
            retstr = retstr + DataSet.raw[:-2]
        if len(retstr) > 0:
            pass
            #print ("retstr:" , len(retstr))
        return retstr

    def readMessages(self):
        """
        reads all available messages from FT2232H and returns them as array of rx_message
        """
        mes = POINTER(rx_message)()
        lenmes = c_int()
        ok = self.MojoConnectToolBox.ReceiveMessages(byref(mes), byref(lenmes))
        if ok:
            return map(lambda m: rx_message_python(m.index, m.tau, m.X, m.Y),cast( mes, POINTER(rx_message * lenmes.value)).contents)
            #return Array(rx_message, cast( mes, POINTER(rx_message * lenmes.value)).contents)
        else:
            return []

    def readMessages_ctypes(self):
        """
        reads all available messages from FT2232H and returns them as array of rx_message
        """
        mes = POINTER(rx_message)()
        lenmes = c_int()
        ok = self.MojoConnectToolBox.ReceiveMessages(byref(mes), byref(lenmes))
        if ok:
            #return map(lambda m: rx_message_python(m.index, m.tau, m.X, m.Y),cast( mes, POINTER(rx_message * lenmes.value)).contents)
            return Array(rx_message, cast( mes, POINTER(rx_message * lenmes.value)).contents), lenmes.value
        else:
```

```
            return [],0


    def disconnect(self):
        """
        disconnect from FT2232H
        """
        if not (self.MojoConnectToolBox.CloseConnection() == 1):
            raise FT2232HError("Unable␣to␣close␣connection")
        self.connected=False



class FT2232H_Command(Enum):
    disconnect = 10
    purge_buffer = 20


class FT2232H_multiprocess(object):

    def __init__(self, *args, **kwargs):
        self.queue_results, self.queue_commands = Queue(), Queue()
        self.read = self.queue_results.get
        self.recv = self.queue_results.get
        self.write = self.queue_commands.put
        self.send = self.queue_commands.put

    def connect(self):
        self.ft2232h_process = Process(target=self._process,
                                        args=(self.queue_results,self.
                                           ↪ queue_commands))
        self.ft2232h_process.start()

    def disconnect(self):
        self.queue_commands.put(FT2232H_Command.disconnect)
        self.ft2232h_process.join()

    def purge_buffer(self):
        self.queue_commands.put(FT2232H_Command.purge_buffer)

    @staticmethod
    def _process(queue_results, queue_commands):
        ft2232h = FT2232H()
        ft2232h.connect()
        logging.info("FT2232H␣connected")
        while True:
```

```
            if not queue_commands.empty():
                try:
                    wr = queue_commands.get_nowait()
                except TimeoutError:
                    pass
                else:
                    logging.info("new command :"+str(wr))
                    if wr is FT2232H_Command.purge_buffer:
                        try: #consume everything in result queue. pretty dumb.
                            while True:
                                queue_results.get_nowait()
                        except Empty:
                            pass
                        ft2232h.purge_buffer()
                        continue
                    if wr is FT2232H_Command.disconnect:
                        ft2232h.disconnect()
                        return
                    logging.info("sent command %s"%wr)
                    ft2232h.write(wr)
            else:
                try:
                    queue_results.put(ft2232h.readMessages())
                except TimeoutError:
                    logging.warn("TimeoutError when putting read data into queue")
            #logging.info ("pipe read write: %d"% (queue_results.qsize()))

if __name__ == "__main__":
    f = FT2232H()
    f.connect()
```

controller.py

```
import ctypes
from warnings import warn
from FT2232H import rx_message, FT2232H
try:
    from Queue import Queue as ThreadQueue
    from Queue import Full
except ImportError:
    # is lowercase since py3
    from queue import Queue as ThreadQueue
```

```
    from queue import Full

from multiprocessing import Process, Pipe, Value, Queue, Array,
    ↪ TimeoutError
from multiprocessing.queues import Empty
from threading import Thread
import time
import struct
import sys
import numpy as np
import scipy.linalg
from collections import namedtuple

plot_data = namedtuple("plot_data", "tau␣amplitude␣bidir")
proc_message = namedtuple("proc_message", "message␣data")
sweep_params = namedtuple("sweep_params", "cyc_start␣steps␣step_size␣
    ↪ reverse␣bidir␣hold_cycles")

import pyqtgraph as pg

class fasterController(object):
    start_byte = "\x12"
    stop_byte = "\x13"
    esc_byte ="\x7D"
    base_clock = 120E6
    plot_update_time = 0.25
    frequency_upper_limit = 250E3
    frequency_lower_limit = 45E3

    def __init__(self):
        self.ft2232h = FT2232H()
        self.ft2232h.connect()
        self.max_remember = 10
        self.start_threads()

    def send_message(self, address, value):
            value = max(min(32767, value), -32768)
            mes = struct.pack(">B", address) + struct.pack(">h",value
                ↪ )
            esced = ""
            for byte in mes:
                if byte == self.start_byte or byte == self.stop_byte
                    ↪ or byte == self.esc_byte:
                     esced+=(self.esc_byte)
                esced+=(byte)
```

```
        self.ft2232h.write(self.start_byte +  esced + self.stop_byte)

    def start_sweep(self, frequency_start, frequency_stop=None, steps=None, hold_cycles = None, bidir=False):
        #TODO: check if this fits the new fpga program, check config registers!!!
        if frequency_start > self.frequency_upper_limit or frequency_start > self.frequency_upper_limit:
            raise ValueError("All repetition rates have to be smaller than %d"%self.frequency_upper_limit)
        start, stop = frequency_start, frequency_stop
        cyc_start = int(np.round(self.base_clock/start))
        config_message = 1
        config_message += 2
        if bidir:
            config_message += 4
        cyc_stop = int(np.round(self.base_clock/stop))
        reverse = cyc_stop < cyc_start

        step_size =int(np.round( np.abs (cyc_stop - cyc_start) / steps))
        if step_size == 0:
            warn("too many steps. stepsize set to 1")
            step_size = 1
        if reverse:

            expected_steps = cyc_start - np.arange(steps + 1) * step_size
        else:
            config_message += 8 # going_up flag in controller set to 1
            expected_steps = cyc_start + np.arange(steps + 1) * step_size
        msg = proc_message("sweep",
                 sweep_params(cyc_start = cyc_start, steps=steps, step_size=step_size, hold_cycles = hold_cycles, bidir = bidir, reverse = reverse))

        if (np.sum(expected_steps < 120E6/self.frequency_upper_limit) > 0) or( np.sum(expected_steps> 120E6/self.frequency_lower_limit) > 0):
            raise ValueError("Rep. rate out of range(%f,%f). Please adjust lower and upper end of sweep or 
```

```
                   ↪ step␣number"%(self.frequency_lower_limit, self.
                   ↪ frequency_upper_limit))
            self.comm_queue.put(msg)
            print ("sweep␣frequency:␣%f"%( self.base_clock/(np.sum(
                ↪ expected_steps) * hold_cycles * (1 + bidir))))
            return expected_steps

    def _display_process(self):
        # set up display
        import pyqtgraph as pg
        pg.mkQApp()
        import pyqtgraph.multiprocess as mp
        proc = mp.QtProcess()
        rpg = proc._import("pyqtgraph")
        plotwin = rpg.plot()
        curve_list = []
        last_update = 0
        while True:
            t = time.clock()
            new_plot = self.display_queue.get()
            if type(new_plot) is plot_data:
                if t - last_update >= self.plot_update_time:
                    last_update = t
                    dataY = proc.transfer(new_plot.amplitude[:,0].
                        ↪ tolist(), _callSync="off")
                    dataX = proc.transfer(self.base_clock/new_plot.
                        ↪ tau,  _callSync="off")
                    curve = plotwin.plot(x=dataX, y=dataY)
                    curve_list.append(curve)
                    if new_plot.bidir:
                        plotwin.plot(x=self.base_clock/new_plot.tau,
                            ↪ y=new_plot.amplitude[:,1], pen="r",
                            ↪ _callSync="off")
                    while len(curve_list) > self.max_remember:
                        c = curve_list.pop(0)
                        c.clear()
            elif type(new_plot) is proc_message:
                if new_plot.message == "shutdown":
                    proc.close()
                    return
            else:
                pass

    def start_threads(self):
        self.display_queue = ThreadQueue(self.max_remember*2)
        self.plot_thread = Thread(target=self._display_process)
```

```
        self.plot_thread.setDaemon(True)
        self.plot_thread.start()
        self.comm_queue = ThreadQueue()
        self.comm_thread = Thread(target=self._comm_thread)
        self.comm_thread.setDaemon(True)
        self.comm_thread.start()

    def calc_bins(frequency_start, frequency_stop, steps):
        cyc_start = int(np.round(self.base_clock/start))
        cyc_stop = int(np.round(self.base_clock/start))

    def _comm_thread(self):
        bins = None
        data = []
        data_idx = 0
        while True:
            t_meas = time.clock()
            while not self.comm_queue.empty():
                new_message = self.comm_queue.get_nowait()
                if new_message.message == "sweep":
                    self.send_message(0,0)
                    self.send_message(1, new_message.data.cyc_start)
                    self.send_message(2, new_message.data.step_size)
                    self.send_message(3, new_message.data.steps)
                    self.send_message(4, new_message.data.hold_cycles
                       ↪ )
                    config_message = 1 + 2 + new_message.data.bidir
                       ↪ *4 + (not new_message.data.reverse)*8
                    if not new_message.data.reverse:
                        bins = new_message.data.cyc_start + np.arange
                            ↪ (new_message.data.steps +1) *
                            ↪ new_message.data.step_size
                    else:
                        bins = new_message.data.cyc_start - np.arange
                            ↪ (new_message.data.steps +1) *
                            ↪ new_message.data.step_size
                    data = []
                    amplitudes = np.zeros((len(bins), 2))
                    data_idx = 0
                    cur_b = bins[0]
                    cur_bi = 0
                    if  new_message.data.reverse:
                        going_up = True
                    bidir = new_message.data.bidir
                    self.send_message(0,config_message)
                    self.ft2232h.purge_buffer()
```

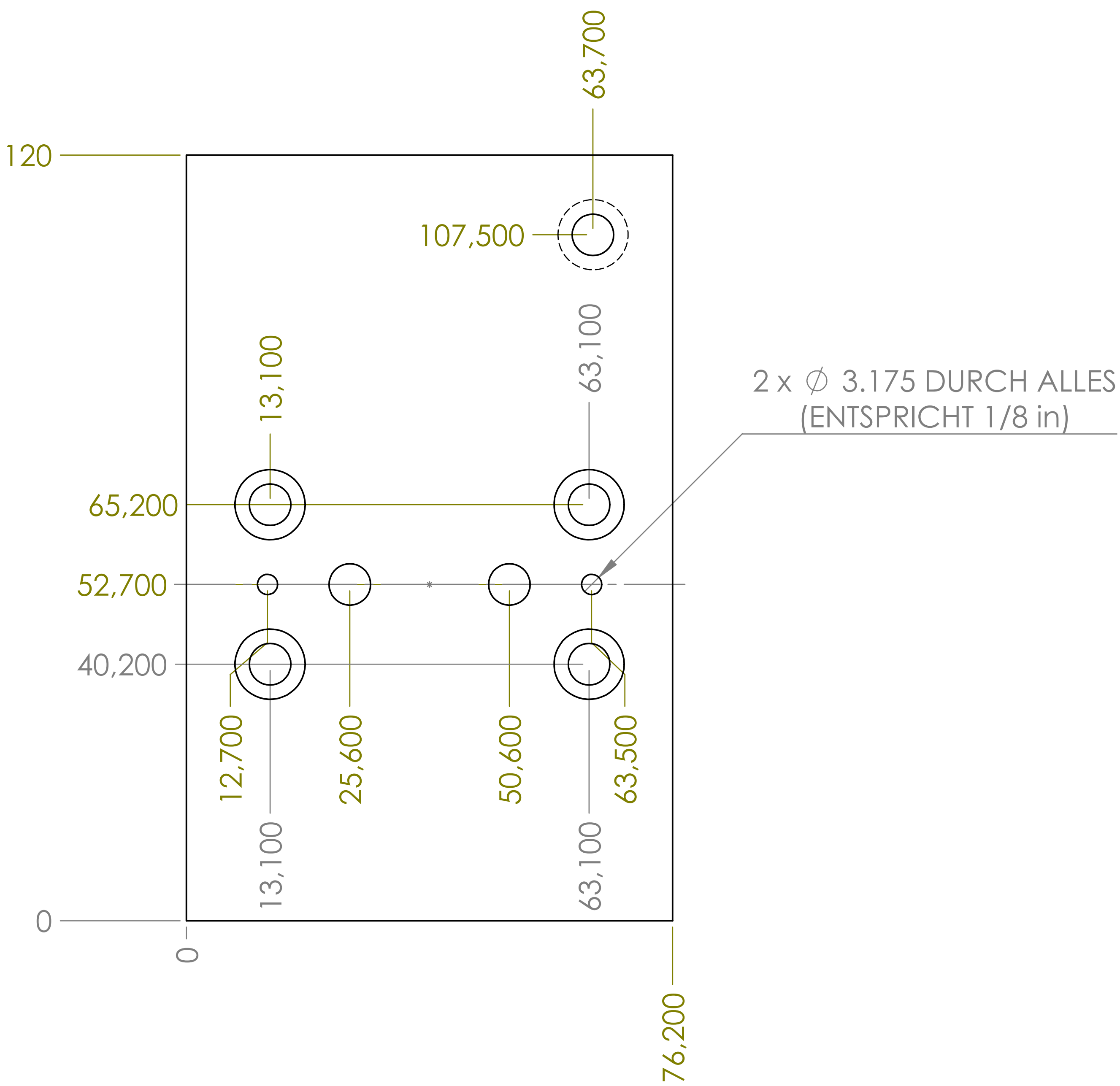
```
        if new_message.message == "stop":
            self.send_message(0,0)
        if new_message.message == "shutdown":
            self.display_queue.put(proc_message("shutdown",
                ↪ None))
            return
    if data_idx == len(data) or len(data)==0:
        try:
            data, _ = self.ft2232h.readMessages_ctypes()
            data_idx = 0
        except ctypes.WindowsError:
            warn("Null␣Pointer␣Exception␣while␣reading␣new␣
                ↪ data")
    while bins is not None and data_idx < len(data):
        sample = data[data_idx]
        data_idx += 1
        if cur_b == sample.tau:
            if going_up:
                amplitudes[cur_bi,0] = sample.amplitude
                cur_bi += 1
                if cur_bi == len(bins):
                    if bidir:
                        going_up = False
                        cur_bi = cur_bi - 2
                    else:
                        cur_bi = 0
                        try:
                            self.display_queue.put_nowait(
                                ↪ plot_data(bins, amplitudes.
                                ↪ copy()/2.0**23, bidir))
                        except Full:
                            pass
            else:
                amplitudes[cur_bi,1] = sample.amplitude
                cur_bi -= 1
                if cur_bi == 0: # bin[0] belongs to the next
                    ↪ sweep
                    going_up = True
                    amplitudes[0,1] = amplitudes[0,0]
                    try:
                        self.display_queue.put_nowait(
                            ↪ plot_data(bins, amplitudes.copy
                            ↪ ()/(2.0**23), bidir))
                    except Full:
                        pass
        cur_b = bins[cur_bi]
```

```
    def stop(self):
        self.comm_queue.put(proc_message("stop", None))
    def quit(self):
        self.comm_queue.put(proc_message("shutdown",None))



if __name__ == "__main__":
    c = fasterController()
    start_sweep = c.start_sweep
    stop = c.stop
    #stop = c.stop
    def _in_ipython():
        try:
            __IPYTHON__
            return True
        except NameError:
            return False
    def quit():
        c.quit()
        if _in_ipython():
            exit
        else:
            sys.exit()
```

Technical Drawings

1. Positioning of the Focal Spot

Alle unbemaßten Bhrg 6.5/Skg. 11 6.2 tief



WENN NICHT ANDERS DEFINIERT:
BEMASSUNGEN SIND IN MILLIMETER
OBERFLÄCHENBESCHAFFENHEIT:
TOLERANZEN:
LINEAR:
WINKEL:

OBERFLÄCHENGÜTE:

ENTGRATEN UND SCHARFE KANTEN BRECHEN

ZEICHNUNG NICHT SKALIEREN

ÄNDERUNG

	NAME	SIGNATUR	DATUM
GEZEICHNET			
GEPRÜFT			
GENEHMIGT			
PRODUKTION			
QUALITÄT			

BENENNUNG:

WERKSTOFF:

ZEICHNUNGSNR.

Part1

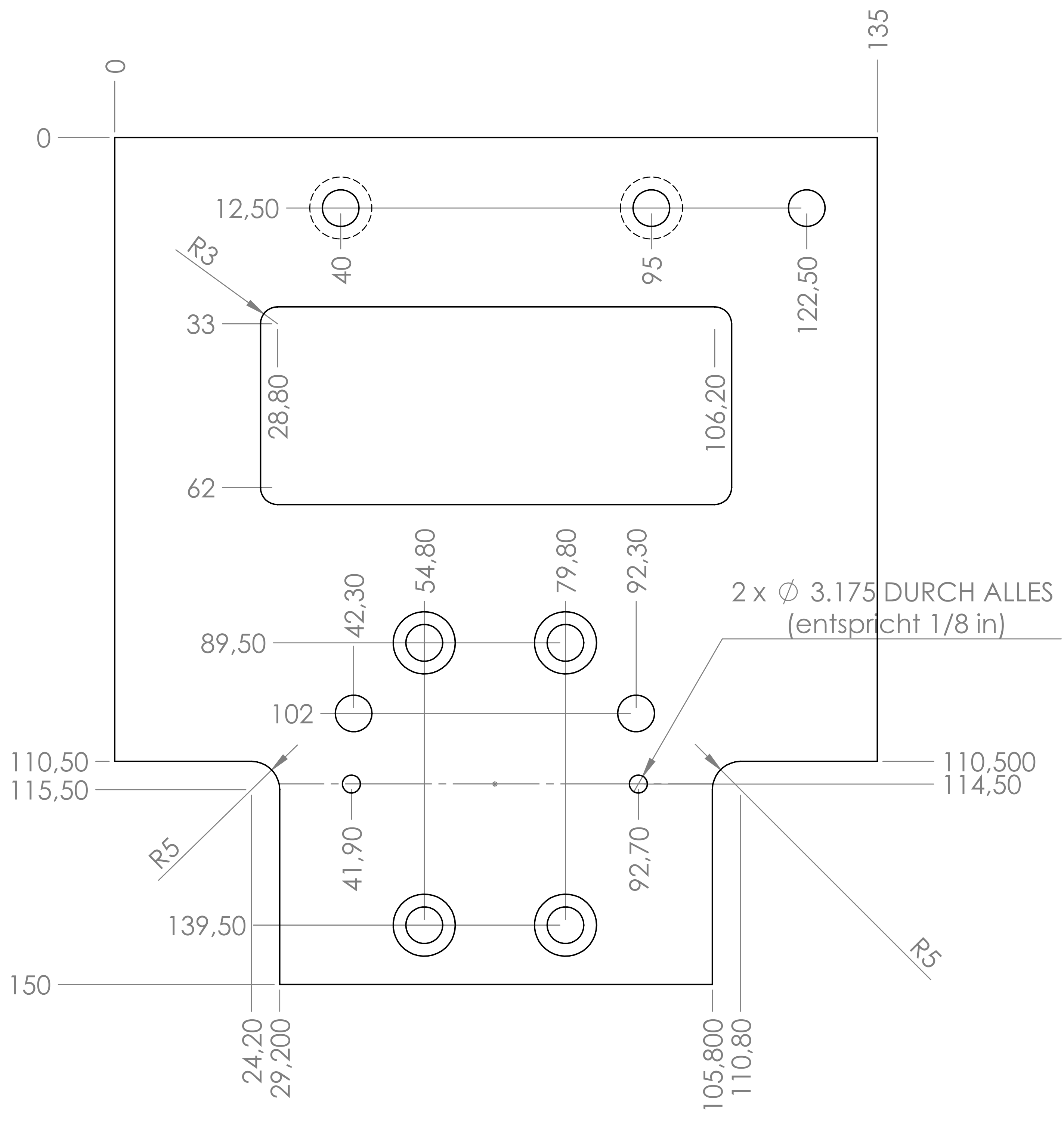
A4

GEWICHT:

MASSSTAB:1:1

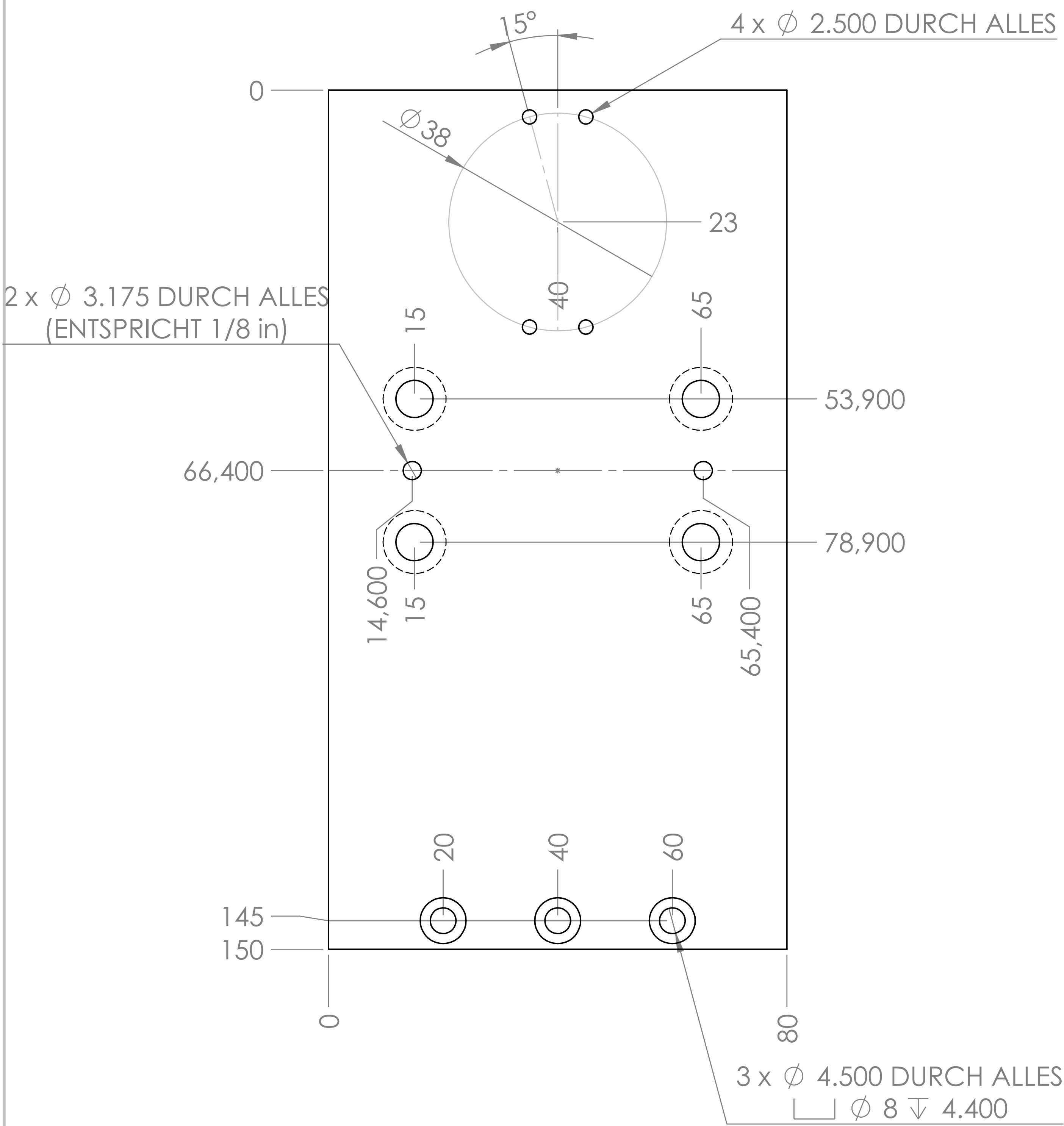
BLATT 1 VON 1

Alle unbemaßten Bhrg 6.5/Skg. 11 6.2 tief



WENN NICHT ANDERS DEFINIERT: BEMASSUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:	OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG	
	NAME	SIGNATUR	DATUM	BENENNUNG:		
GEZEICHNET						
GEPRÜFT						
GENEHMIGT						
PRODUKTION						
QUALITÄT			WERKSTOFF:	ZEICHNUNGSNR.	Part4	A4
			GEWICHT:	MASSSTAB:1:1	BLATT 1 VON 1	

1.1. Design for Fixed Parabolic Mirror



Alle unbemaßten Bhrg 6.5/Skg. 11 6.2 tief

WENN NICHT ANDERS DEFINIERT:
BEMASSUNGEN SIND IN MILLIMETER
OBERFLÄCHENBESCHAFFENHEIT:
TOLERANZEN:
LINEAR:
WINKEL:

OBERFLÄCHENGÜTE:

ENTGRATEN UND SCHARFE KANTEN BRECHEN

ZEICHNUNG NICHT SKALIEREN

ÄNDERUNG

	NAME	SIGNATUR	DATUM
GEZEICHNET			
GEPRÜFT			
GENEHMIGT			
PRODUKTION			
QUALITÄT			

BENENNUNG:

WERKSTOFF:

GEWICHT:

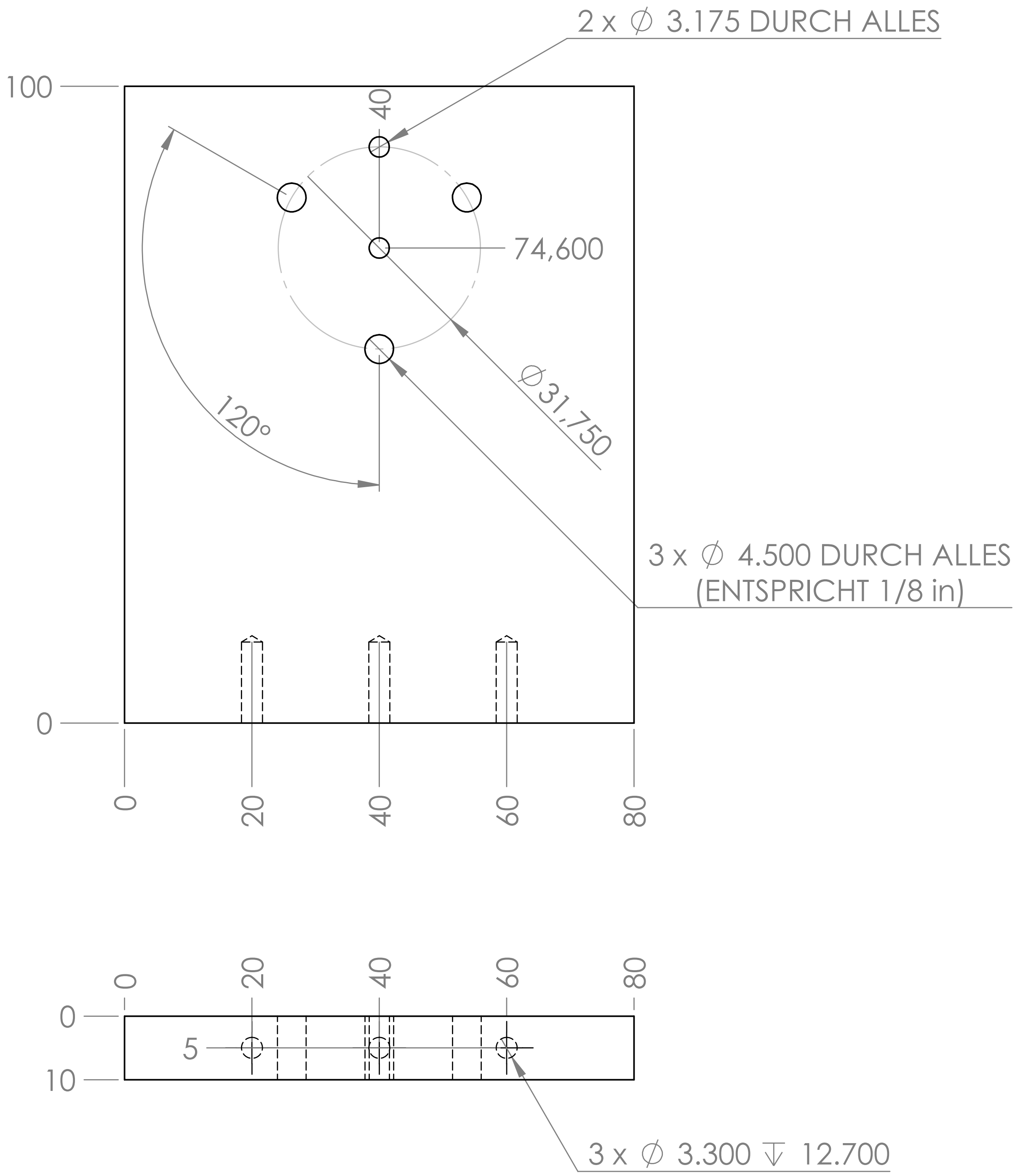
ZEICHNUNGSNR.

Part5

A4

MASSSTAB:1:1

BLATT 1 VON 1



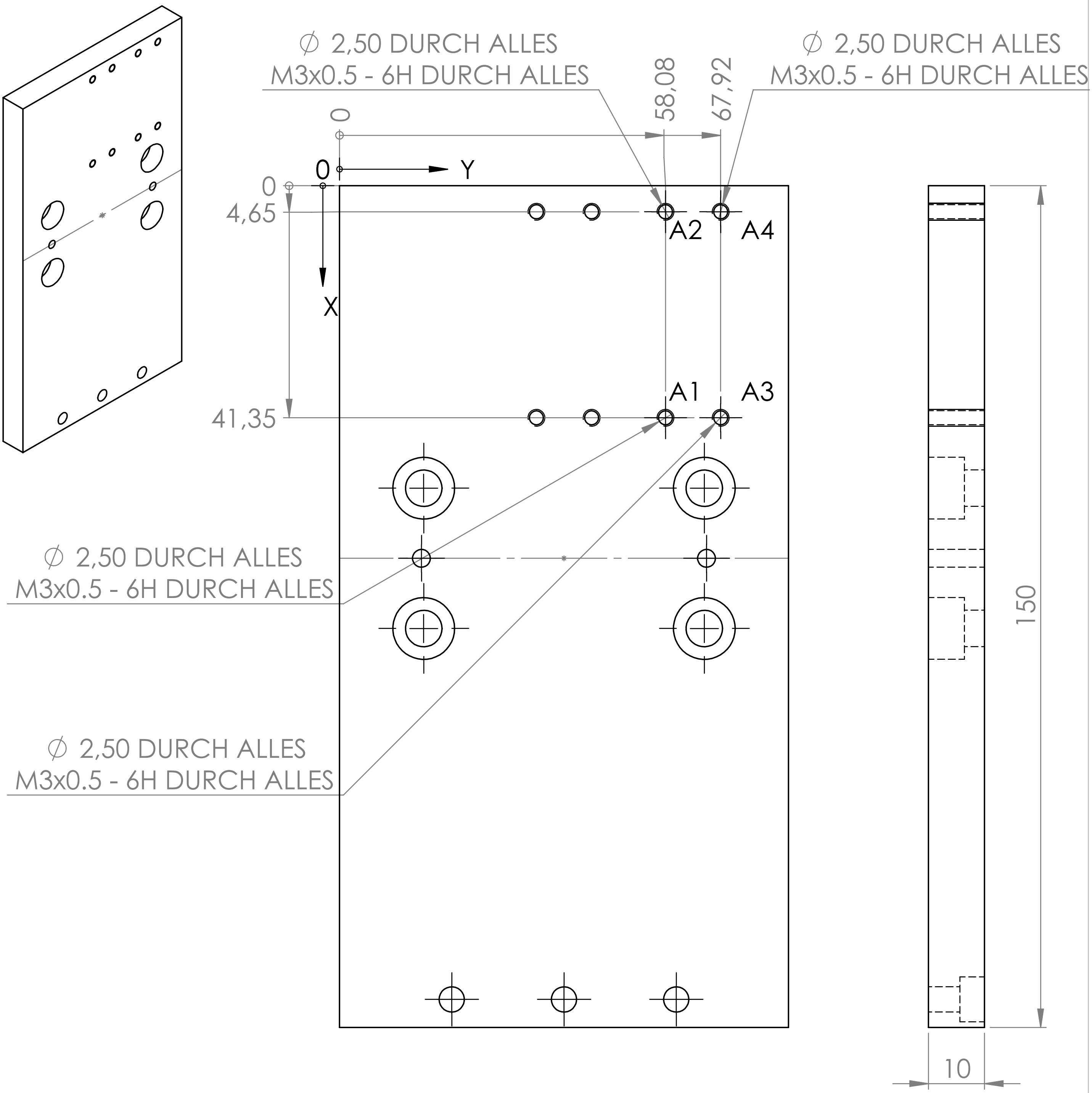
WENN NICHT ANDERS DEFINIERT: BEMASSUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:	OBERFLÄCHENGÜTE:	ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG

	NAME	SIGNATUR	DATUM			
GEZEICHNET						
GEPRÜFT						
GENEHMIGT						
PRODUKTION						
QUALITÄT				WERKSTOFF:		
				GEWICHT:		

BENENNUNG:

ZEICHNUNGSNR.	Part6	A4
MASSSTAB:1:1	BLATT 1 VON 1	

1.2. Design for Rotatable Parabolic Mirror

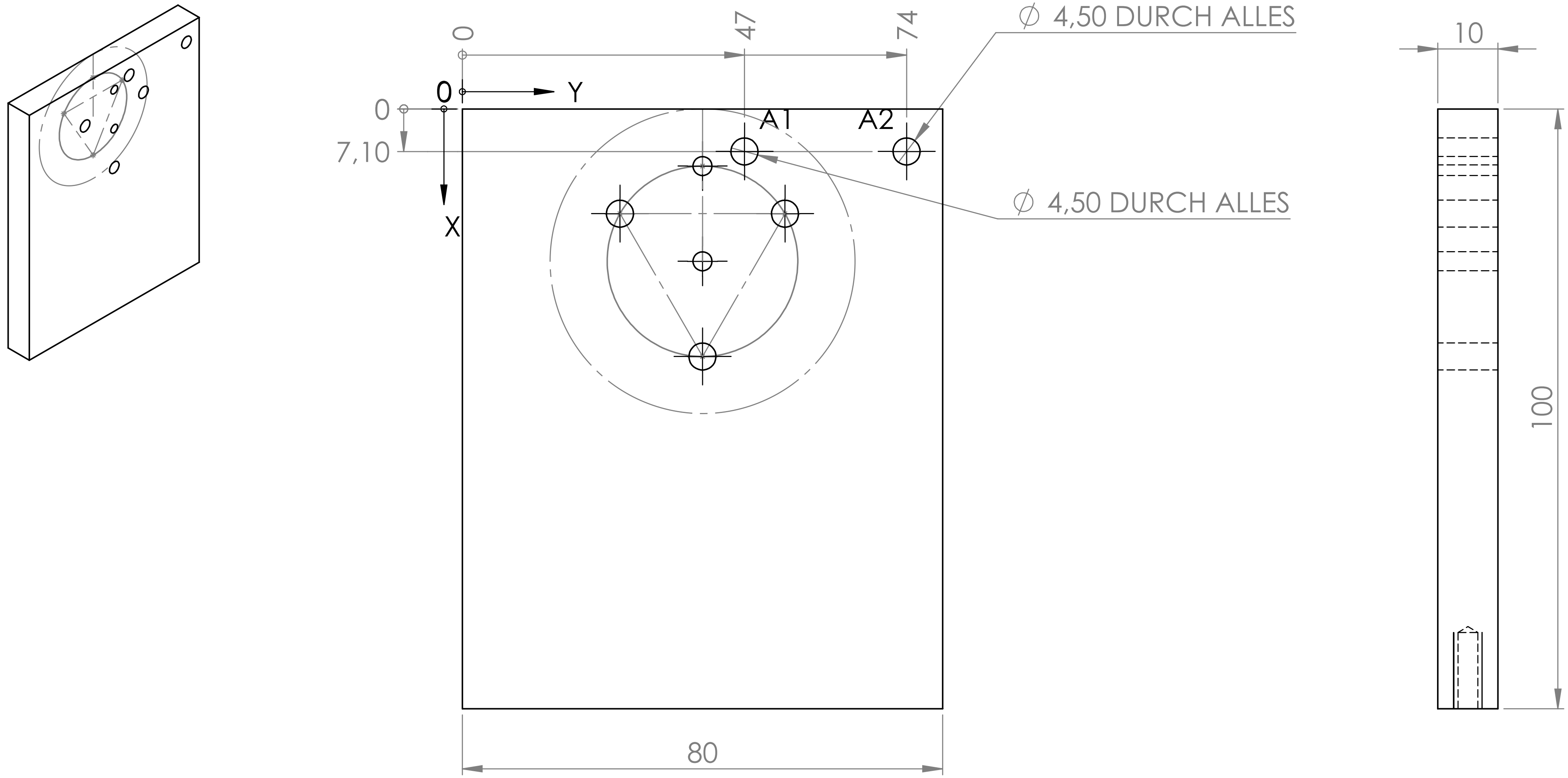


ETIKETT	X-POSITION	Y-POSITION	GRÖSSE
A1	41,35	58,08	∅ 2,50 DURCH ALLES M3x0.5 - 6H DURCH ALLES
A2	4,65	58,08	∅ 2,50 DURCH ALLES M3x0.5 - 6H DURCH ALLES
A3	41,35	67,92	∅ 2,50 DURCH ALLES M3x0.5 - 6H DURCH ALLES
A4	4,65	67,92	∅ 2,50 DURCH ALLES M3x0.5 - 6H DURCH ALLES

GRÖSSE	ZEICHNUNGSNR.	ÄND.
A	a)Part5	
MASSSTAB:1:2	GEWICHT:	BLATT 1 VON 1

ANWENDUNG

ZEICHNUNG NICHT SKALIEREN



ETIKETT	X-POSITION	Y-POSITION	GRÖSSE
A1	7,10	47	Ø 4,50 DURCH ALLES
A2	7,10	74	Ø 4,50 DURCH ALLES

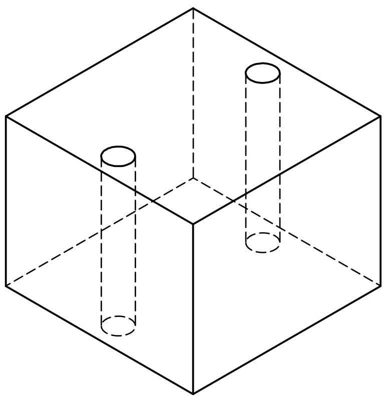
ZEICHNUNGSNR. a)Part6

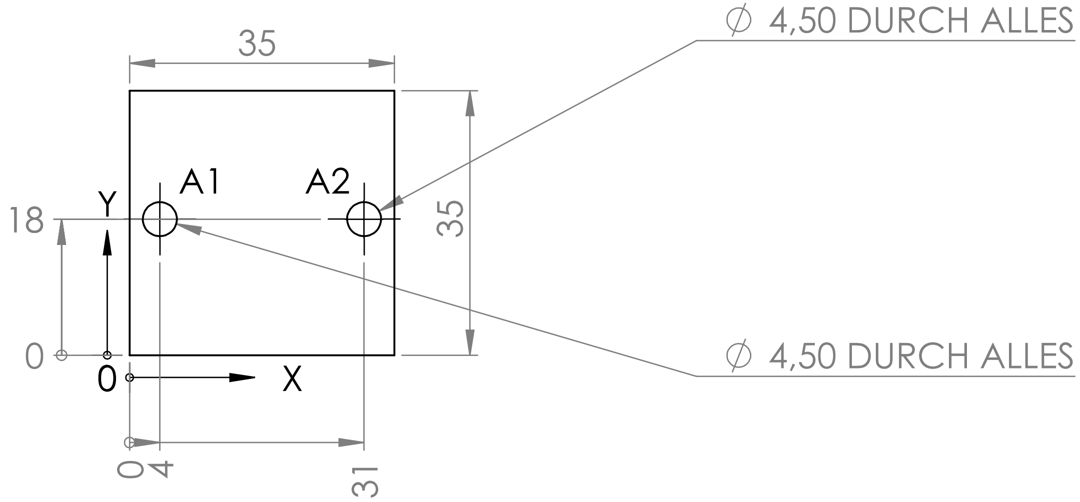
A4

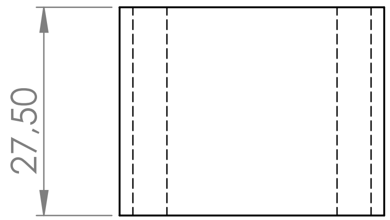
GEWICHT:

MASSSTAB:1:2

BLATT 1 VON 1







ETIKETT	X-POSITION	Y-POSITION	GRÖSSE
A1	4	18	Ø 4,50 DURCH ALLES
A2	31	18	Ø 4,50 DURCH ALLES

Teil1

A4

GEWICHT:

MASSSTAB:1:1

BLATT 1 VON 1

Curriculum Vitae

Georg Ramer

CV

Enzersdorferstrasse 2 / 1
2345 Brunn am Gebirge
Austria
☎ +43 1 58801 15144
+43 664 63 44 931
✉ georg@ramer.at
Date of birth: 19.2.1987

Education

Since 2012 **Doctoral programme in Engineering Sciences, Diploma programme: Technical Chemistry**, TU Wien, Austria.

25.07.2012 **Academic Degree Master of Science**.

2010 – 2012 **Master's programme Technical Chemistry - Materials Technology and Materials Analytics**, TU Wien, Austria.

18.05.2010 **Academic Degree Bachelor of Science**.

2006 – 2010 **Bachelor's programme Technical Chemistry**, TU Wien, Austria.

Theses, Internships

Since 2012 **Development and characterization of a near-field infrared microscope by combining Atomic Force Microscopy and QCL spectroscopy**, *Ph.D. Thesis*, Supervisor: Bernhard Lendl, TU Wien, Austria.

- development of an EC-QCL AFMIR/PTIR system out of discrete components
  - design and assembly of an optical/optomechanical setup
  - development of control software using Python and Labview
  - development of a dedicated PTIR controller
- application of AFMIR/PTIR to biological and microelectronic samples.

2011 – 2012 **Characterization of Planar SiN Waveguides for a MIR Laser Based Sensor**, *Master Thesis*, Supervisor: Bernhard Lendl and Markus Brandstetter, TU Wien, Austria.

- development of a mid-IR optical setup for coupling the beam of an EC-QCL into grating coupled waveguides
- design of a flow cell for integration of waveguide into flow injection analysis (FIA) setups
- simulation of waveguide spectra using Mathematica and Python
- development of a Labview-FPGA based system for using a pyroelectric detector with a pulsed EC-QCL

Juli 2010 **Explosives Detection Using Stand-Off Raman Spectroscopy**, *Bacchelor Thesis*, Supervisor: Bernhard Lendl and Bernhard Zachhuber, TU Wien, Austria..

- collection of reference spectra using a confocal Raman microscope
- determination of calibration curves and LODs for different explosives using a stand-off Raman setup
- implementation of a ranging technique with stand-off Raman

Juli 2009 **Analysis of Sugar-Water Interaction**, *Internship*, Supervisor: Bernhard Lendl and Christoph Wagner, TU Wien, Austria..
- implementation of a flow injection analysis system using Labview control software
- chemometric data pretreatment and evaluation (MCR-ALS, 2D-CoS)

Skills

Lab Skills

Mid-IR Spectroscopy including familiarity with the following advanced techniques:
- attenuated total reflection and waveguide sensing
- time resolved step scan
- FTIR microscopic imaging using single point and focal plane array detectors
- infrared nanoscopy (AFMIR/PTIR)

Optics with a focus on mid-IR lasers
- design and assembly of optical lab setups from discrete components
- simulation using Zemax
- experience with Daylight Solutions EC-QCLs
- certified as laser safety officer
- use of infrared detectors

Electronics
- use of common lab electronics (oscilloscope, signal/pulse generator, lock-in amplifier, logic analyzer)
- programming microcontrollers for automating lab setups
- use of National Instruments C-series modules including cDAQ and cRIO based systems

Atomic Force Microscopy contact mode, tapping mode and AFMIR/PTIR

Chemistry student labs: organic synthesis, inorganic synthesis, (instrumental) analytical chemistry, metallurgy, biochemistry, physical chemistry, chemical engineering, polymer sciences

Programming Languages

Python development of scripts and GUI programs for data evaluation and instrument control. Familiar with the following packages: NumPy, SciPy, Matplotlib, SymPy, PyVISA, PyDAQmx, PyQt, PyQtGraph

MATLAB development of scripts for data evaluation

LabView data acquisition and instrument control. Familiar with the following additional modules: DAQmx, VISA, Labview FPGA

Mathematica solving equations, scripting

VHDL basics in developing for FPGAs

Latex typesetting of academic publications

Computer Programs

IR Bruker OPUS

AFM Keysight PicoView, Gwyddion

Scientific OriginPro, ZEMAX

CAD	Solidworks, KiCAD
Chemometrics	Cytospec, ImageLab, PLSToolbox
Office	MS Office, LibreOffice
OS	MS Windows, Linux
Graphics	Inkscape, Scribus, Gimp, CorelDraw
	Business/Management
ECBL	European Business Competence License Level A

Awards and Achievements

Leistungsstipendium	TU Wien award for special performance in studies and examinations, 2010 and 2012
TUtheTop	the high potential programme at TU Wien
Student Poster Award	at ICAVS-6

Languages

German	native speaker
English	fluent
French	high school level

Publications

Articles

G. Ramer and L. Ashton, “Two-dimensional codistribution spectroscopy applied to UVRR and ROA investigations of biomolecular transitions,” *Journal of Molecular Structure*, vol. In press, feb 2016.

G. Ramer, A. Balbekova, A. Schwaighofer, and B. Lendl, “Method for Time-Resolved Monitoring of a Solid State Biological Film Using Photothermal Infrared Nanoscopy on the Example of Poly- l -lysine,” *Analytical Chemistry*, vol. 87, no. 8, pp. 4415–4420, 2015.

M. R. Alcaráz, A. Schwaighofer, C. Kristament, G. Ramer, M. Brandstetter, H. Goicoechea, and B. Lendl, “External-Cavity Quantum Cascade Laser Spectroscopy for Mid-IR Transmission Measurements of Proteins in Aqueous Solution,” *Analytical Chemistry*, vol. 87, no. 13, pp. 6980–6987, 2015.

G. Ramer, A. Balbekova, and B. Lendl, “Resonant Photoexpansion Infrared Nanoscopy - towards robust and time-resolved measurements,” in *SCIX*, (Reno, Nevada), 2014.

G. Ramer, J. Kasberger, M. Brandstetter, A. Saeed, B. Jakoby, and B. Lendl, “A broadband grating-coupled silicon nitride waveguide for the mid-IR: characterization and sensitive measurements using an external cavity quantum cascade laser,” *Applied Physics B*, vol. 116, pp. 325–332, Oct. 2013.

J. Srajer, A. Schwaighofer, G. Ramer, S. Rotter, B. Guenay, A. Kriegner, W. Knoll, B. Lendl, and C. Nowak, “Double-layered nanoparticle stacks for surface enhanced infrared absorption spectroscopy.,” *Nanoscale*, vol. 6, no. 1, pp. 127–31, 2013.

D. Lumpi, C. Wagner, M. Schöpf, E. Horkel, G. Ramer, B. Lendl, and J. Fröhlich, “Fibre optic ATR-IR spectroscopy at cryogenic temperatures: in-line reaction monitoring on organolithium compounds,” *Chemical Communications*, vol. 48, no. 18, p. 2451, 2012.

B. Zachhuber, G. Ramer, A. Hobro, E. t. H. Chrysostom, and B. Lendl, “Stand-off Raman spectroscopy: a powerful technique for qualitative and quantitative analysis of inorganic and organic compounds including explosives.,” *Analytical and bioanalytical chemistry*, vol. 400, pp. 2439–47, June 2011.

B. Zachhuber, C. Gasser, G. Ramer, E. t. H. Chrysostom, and B. Lendl, “Depth profiling for the identification of unknown substances and concealed content at remote distances using time resolved stand off Raman spectroscopy,” *Applied Spectroscopy*, vol. 66, no. 8, pp. 875–881, 2012.

Book Chapters

G. Ramer and B. Lendl, *Spectroscopic Techniques for Characterization of Gold Nanoparticles*, ch. 7, pp. 301 – 325. Elsevier, 2014.

G. Ramer and B. Lendl, *Attenuated Total Reflection Fourier Transform Infrared Spectroscopy*. John Wiley & Sons, Ltd, 2013.

C. Wagner, A. Genner, G. Ramer, and B. Lendl, "Advanced Total Lab Automation System (ATLAS)," in *Modeling, Programming and Simulations Using LabVIEW Software* (R. de Asmundis, ed.), InTech, 2010.

Presentations

G. Ramer, A. Balbekova, A. Schwaighofer, and B. Lendl, "Resonance Tracking in Resonance Enhanced Infrared Nanoscopy," in *SCIX*, (Providence, Rhodes Island), 2015.

G. Ramer, A. Balbekova, A. Schwaighofer, and B. Lendl, "Developments in mid-infrared nanoscopy instrumentation," in *ICAVS 8*, 2015.

G. Ramer, A. Balbekova, and B. Lendl, "Resonant Photoexpansion Infrared Nanoscopy - towards robust and time-resolved measurements," in *SCIX*, (Reno, Nevada), 2014.

G. Ramer, A. Balbekova, and B. Lendl, "Time Resolved Photothermal Expansion Infrared Nanoscopy," in *CEITEC Annual Conference "Frontiers in Material and Life Sciences"*, (Brno, Czech Republic), 2014.

G. Ramer, M. Brandstetter, J. Kasberger, A. Saeed, and B. Lendl, "Highly Sensitive Waveguide Sensor Facilitated By A Broadly Tunable Quantum Cascade Laser," in *Transducerst̓13 & Eurosensors XXVII, Barcelona*, 2013.

G. Ramer, M. Brandstetter, J. Kasberger, and B. Lendl, "Tunable Quantum Cascade Laser based waveguide sensors for mid-infrared spectroscopy," in *ICAVS VI, Sonoma County. California*, 2011.