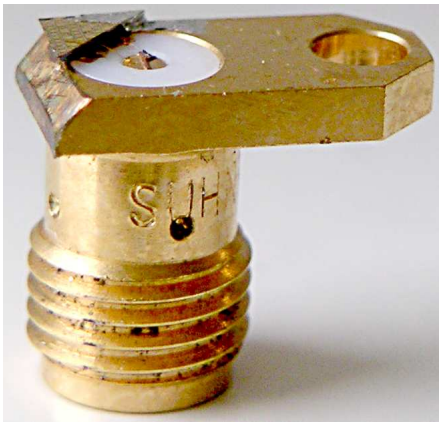
Université de Neuchâtel unine Institut de physique

Design and characterisation of far- and mid-infrared quantum cascade detectors



Thèse

présentée à la Faculté des Sciences de l'Université de Neuchâtel
pour obtenir le grade de Docteur ès Sciences

par

Marcel Graf

Soutenue le 28 septembre 2006, en présence du jury:

Prof. Daniel Hofstetter, directeur de thèse, Université de Neuchâtel
Prof. Vincent Berger, Université Paris 7 - Denis Diderot
Prof. Jérôme Faist, Université de Neuchâtel
PD Dr. Harald Schneider, Forschungszentrum Rossendorf

Order this book online at www.trafford.com/07-2403
or email orders@trafford.com

Most Trafford titles are also available at major online book retailers.

Note for Librarians: A cataloguing record for this book is available from Library and Archives Canada at www.collectionscanada.ca/amicus/index-e.html

Printed in Victoria, BC, Canada.

ISBN: 978-1-4251-5406-6

www.trafford.com

North America & international
toll-free: 1 888 232 4444 (USA & Canada)
phone: 250 383 6864 ♦ fax: 250 383 6804
email: info@trafford.com

The United Kingdom & Europe
phone: +44 (0)1865 722 113 ♦ local rate: 0845 230 9601
facsimile: +44 (0)1865 722 868 ♦ email: info.uk@trafford.com

10 9 8 7 6 5 4 3 2 1

Université de Neuchâtel

FACULTE DES SCIENCES
Secrétariat-Décanat de la faculté
- Rue Emile-Argand 11
- CP 158
- CH-2009 Neuchâtel

IMPRIMATUR POUR LA THESE

Design and optimization of far- and mid-infrared quantum cascade detectors

Marcel GRAF

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel,
sur le rapport des membres du jury

MM. D. Hofstetter (directeur de thèse), J. Faist,
H. Schneider (Dresden, D)
et V. Berger (Paris)

autorise l'impression de la présente thèse.

Neuchâtel, le 4 juin 2007

Le doyen :

T. Ward

UNIVERSITE DE NEUCHATEL
FACULTE DES SCIENCES
Secrétariat-décanat de la faculté
Rue Emile-Argand 11 - CP 158
CH-2009 Neuchâtel

■ Téléphone : +41 32 718 21 00 ■ Fax : +41 32 718 21 03 ■ E-mail : secretariat.sciences@unine.ch ■ www.unine.ch/sciences

Mots clés

photodétecteurs; détecteurs infrarouges; semiconducteurs III-V; puits quantiques; transitions inter-sous-bande; cascade quantique; rayonnement THz; détection hétérodyne optique;

Résumé

Cette thèse traite un type nouveau de photodétecteurs infrarouges, basés sur les transitions inter-sous-bande dans les matériaux semi-conducteurs III-V (AlAs/-GaAs sur GaAs et InAlAs/InGaAs sur InP). S'adressant à une des sources bruyantes principales de photodétecteurs photoconducteurs, le bruit de courant d'obscurité, ce design comprend un gradient potentiel intérieur en forme de cascade quantique, donc le nom détecteur à cascade quantique (QCD). Cette approche permet l'opération sans mise sous tension et sans courant d'obscurité, en conséquence. L'extraction d'électrons excités est fournie par cette cascade. La fabrication et l'opération de tels détecteurs ont été accomplis avec succès, dans la gamme de l'infrarouge moyen (détection aux longueurs d'ondes de 5.3 µm et de 9 µm) et dans l'infrarouge lointain (la région THz) à 84 µm, au-dessous de l'énergie de phonons optiques longitudinales (LO). Additionellement, il est démontré que ces QCDs sont rapides, comparables avec des photodétecteurs infrarouges à puits quantiques (QWIP). Jusqu'à la limite de fréquence superieure de 23 GHz, imposée par l'électronique de haute fréquence disponible pendant ces mesures, la diminution en performance observée est entièrement expliquée par l'amortissement à cause du circuit RLC, créé par la technique d'assemblage simple.

En plus du design, la fabrication et les mesures expérimentales, cette thèse essaie aussi d'expliquer deux propriétés importantes, variant en fonction de température de détecteur. Les changement fortes en résistance électrique et la

diminution en réponse des détecteurs à cascade quantique en infrarouge moyen sont numériquement calculés, basés sur un modèle de transitions d'électrons entre les différents états de la cascade quantique, assistées par des phonons optiques longitudinales.

Keywords

photodetectors; infrared detectors; III-V semiconductors; quantum wells; intersubband transitions; quantum cascade; THz radiation; optical heterodyne detection;

Abstract

This thesis covers a novel type of infrared photodetectors, based on intersubband transitions in III-V semiconductors (AlAs/GaAs on GaAs and InAlAs/InGaAs on InP). To address one of the main noise sources of photoconductive photodetectors, dark current noise, this design comprises an internal potential gradient in form of a quantum cascade, hence the name quantum cascade detector (QCD). This approach allows bias-less and therefore dark current-less operation, the extraction of excited electrons is provided by the cascade. Successful fabrication and operation of such detectors was achieved, both in the mid-infrared range (detectors at 5.3 µm and 9 µm wavelength) and in the far-infrared (THz region) at 84 µm, below the energy of LO-phonons. It is also shown that these QCDs are fast, comparable to quantum well infrared photodetectors (QWIP). Up to the limit of the high frequency electronics available for this measurement, 23 GHz, the observed decrease in performance is entirely due to the RLC damping, caused by the simple device mounting technique.

In addition to the design, fabrication and the experimental measurements, this thesis also attempts to explain two important properties as function of detector temperature. The strongly varying device resistance and the drop in responsivity of the mid-infrared quantum cascade detectors are numerically calculated based on a modell describing LO-phonon assisted transitions between the different involved states of the quantum cascade.

Contents

List of Figures

List of Tables

1 Introduction

1.1 History of intersubband transitions

Optical intersubband transitions in quantum wells were first observed in 1985 by two researchers at Stanford University, West and Eglash [1]. Owing to the theoretical description which involved the envelope function approximation, they called this novel optical phenomenon a 'quantum well envelope state transition' (QWEST). These transitions take place between electronic states located entirely in the conduction band. The quantum wells of West and Eglash consisted of thin GaAs layers sandwiched between AlGaAs barriers. In their pioneering work, they pointed out the possibility of tuning the narrow QWEST resonance by varying the well width; and they also immediately proposed its usage for fast optical digital logic devices.

This discovery opened the "era" of intersubband transitions for optical and optoelectronic applications which culminated in 1994 with the realisation of quantum cascade lasers at AT&T Bell Laboratories by Faist et al. [2].

On the detection side, intersubband transitions were extensively investigated by Levine et al. at AT&T Bell Laboratories, as well as by Harwit and Harris at Stanford University, who characterised both the strong absorption and the Stark shift of such intersubband transitions in AlAs/GaAs supperlattices [3, 4]. This resulted in the implementation of the first 10 µm quantum well infrared photodetector (QWIP) by Levine at al. in 1987 [5].

Since then, many other people have made important contributions to the understanding, physical mechanism, and performance of QWIPs; a substantial amount of this progress has been published by Levine in an excellent review article [6] and by Liu in a comprehensive book chapter [7].

In the context of this thesis, the work of Schneider at the Fraunhofer-Institut für Angewandte Festkörperphysik is of special interest. In 1993, he was the first

to propose the use of asymmetrical quantum well structures, effectively producing an internal bias field. Such a device can be operated in photovoltaic mode (i.e. without external bias). Schneider showed theoretically that this should lead to an optimised performance for operation under high photon fluxes or long integration times in the long wavelength range (8 µm - 12 µm) [8]. Schönbein, Schneider et al. reported also the successful operation of a photovoltaic QWIP in the GaAs/AlGaAs material system at 77 K with a maximal peak detectivity of 5.1×10^9 Jones [9]. In this device, the asymmetry was achieved by insertion of an additional, energetically higher, and very thin barrier layer on one side of the well. Later on, Schneider, Schönbein et al. performed high-speed characterisation of the same device at room temperature, which led to measured rise and decay times of 12 ps and 65 ps, respectively [10]

QWIPs have also been realised in other material systems: Hasnain, Levine et al. demonstrated successful detection at 3 µm - 5 µm with structures grown in lattice matched InGaAs/InAlAs on InP [11], where the higher conduction band offset of this material allowed to go for shorter wavelengths. Even QWIPs having transition energies higher than the conduction band offset have been shown by Leavitt and Little [12]; they used excited states above the barrier, being confined by one of the superlattice minigaps. The accessable wavelength range has also be extended to the "very long wavelength" range 14 µm - 20 µm by Sarusi et al. in 1994 [13].

Finally, the successful fabrication of focal plane arrays (FPA) by Gunapala et al. [14] and Schneider et al. [15] together with significantly raised absorption [16] and broadened detection spectra [17] made QWIPs a mature technolodgy which is used in several infrared cameras available on the market [18–21]

1.2 Motivation

Based on the experiments by Hofstetter et al. who investigated photocurrent generation in a quantum cascade laser [22], we recognised that a similar structure would be a promising approach for an intersubband detector in the THz range, which was quite recently boosted by the successful fabrication of emitters and quantum cascade lasers. On the emitter side, electronic devices like Gunn diodes or Schottky diode frequency multipliers approach this range from the low fre-

quency end[1], whereas optical devices like gas or semiconductor lasers are quickly moving into the THz range from the higher frequency side [23–26]. At low temperatures, they emit several milliwatts of continuous wave power. Gunn diodes and frequency mixers have also achieved several milliwatts of radiated power at frequencies on the order of hundreds of gigahertz [27]. Because of this increasing interest in the THz region, it was natural to look for a convenient photodetector in this wavelength region. Up to the year 2002, there were already several different detector types reported to work in the THz range: Coherent methods like Schottky diode mixers [28], nonlinear optical crystals like {110} ZnTe [29], and gated photoconductive antennas [30]; incoherent methods measuring the heat produced by absorption of incident radiaton like Si-bolometers or pyroelectric crystals, for example deuterated triglycerine-sulfate (DTGS); and finally, extrinsic photoconductors such as doped Ge detectors [31–33].

As the transition energy of such devices lies in the range of the thermal energy - 13 meV (95 µm) corresponds to a temperature of 150 K - the low barrier heights in a "normal" QWIP design would probably lead to a lot of thermally activated carriers, burying the photocurrent signal in detector noise. This problem could be reduced by "tailoring" a custom path for the electrons, i.e. by tunneling through barriers which are much higher than the transition energy. This concept was consequently realized in the first demonstration of a THz quantum cascade detector by ourselves [34]. It is noteworthy that at the beginning of this work, there was no THz QWIP reported, although Liu et al. showed a THz QWIP [35] shortly after our own THz detector.

Due to the absence of dark current, the photovoltaic operation mode of quantum cascade detectors has two important advantages: First, one gains more freedom in extending the integration time of a readout circuit to adapt to the actual illumination intensity, as no dark current is "filling" up the readout capacitance. Second, the thermal load of the detector is significantly reduced. This might be interesting for both space born applications, where passive cooling would be convenient and for terrestrial staring systems, which might then work with a smaller cooling system.

During this work, Gendron et al. at Thales Research and Technology, France published their work on a quantum cascade detector (QCD) [36] in the GaAs

[1] This means "low frequencies" compared to the optical frequencies of visible and infrared light, but in terms of electrical engineering, these are extremly high frequencies.

material system, where the term *quantum cascade detector* was publicly coined (apparently, the term has already been defined in a french patent application by Berger [37] in 2001). This motivated us to adapt our far-infrared detector[2] to the mid-infrared[3] region. In order to ensure an efficient vertical transport, an extraction cascade carefully adapted to the longitudinal optical (LO) phonon energy in the InGaAs/InAlAs material system was implemented. In addition, we intended to verify that this kind of detector was comparably fast as QWIPs [10, 38, 39].

1.3 Organisation of this thesis

This thesis is structured as follows. In chapter 2, I start with a couple of theoretical aspects, divided into two main sections. An introduction of the envelope function ansatz for the description of electronic states in quantum wells is given in section 2.1. It leads to the derivation of the absorption coefficient of an optical transition between two states in a quantum well, including the associated polarisation selection rule. In section 2.2, the equations describing general detector physics are presented. The relevant figures of merit describing detector performance are derived, taking into account different noise mechanisms.

Chapter 3 contains a detailed description of all measurement setups used and developed in the scope of this work. It contains also reference measurements and the description of correction calculations necessary for the spectral measurements.

The design of our far-IR quantum cascade detector is presented in Chapter 4. After the electrical and spectral measurements, an analysis of the Stark shift due to applied bias voltages is shown.

Chapter 5 contains the studies on a series of mid-IR QCDs on InP, detecting at 9 µm and 5.3 µm wavelength. Besides the electrical and spectral measurements - both as function of the accessible temperature range, the high frequency characterisation of the 5.3 µm structure is presented in section 5.4.

[2] In the text of this work, far infrared denotes the wavelength region above the Reststrahlen band - roughly congruent with the so called THz range, and not the 8 µm - 12 µm atmospherical window.

[3] Similarly, the term mid-IR stands for the whole region from roughly 3 µm up to the LO-phonon, and not only the window 3 µm - 5 µm

A summary of this work follows in chapter 6, followed by the acknowledgements. After the list containing the references cited herein, a list of soft- and hardware used for the measurements, data evaluation, and for writing this thesis is given.

In appendix B, notes from my labbook are written down. They might be of importance for future users of the measurement setups. Appendix C includes the relevant parts of the source code written to assist in the semi-automatic evaluation of temperature dependant spectral and $I - V$ measurements, as well as the code of the wrapper module around *sewself* {sewself}, which permits the simulations described in section 5.3 and the algorithms of the transport model used in section 5.2.

Finally, a list of work published during this work is appended.

2 Theoretical framework

The purpose of this chapter is to outlay the relevant physics of intersubband transitions in semiconductors and the description of photodetectors in general. These two subjects are the base of a successful description and characterisation of quantum cascade photodetectors.

2.1 Intersubband transitions in quantum wells

The description of the intersubband absorption is the first point in the modellisation of a photodetector based on intersubband transitions. In the framework of a single-particle description in one single band of the electronic structure of the bulk semiconductor, it is possible to calculate the intersubband absorption coefficient based on Fermi's golden rule for the induced transition rate. The description below roughly follows the description by Helm [40] with additional information and parts of the nomenclature as in Bastard [41].

2.1.1 Effective mass & envelope function approximation

According to the well known Bloch description of the electronic structure of a bulk crystal, we start by stating that the internal potential must be periodic with the lattice structure of the crystal. Therefore, the wavefunction $\psi_i(\boldsymbol{r})$ for a single electron in the conduction band of the semiconductor is written as product of an envelope function $f_i(\boldsymbol{r})$, which is supposed to vary slowly over the distance of a lattice period, and a function $u_\nu(\boldsymbol{r})$, which itself is periodic with the lattice.

$$\psi_i(\boldsymbol{r}) = f_i(\boldsymbol{r}) u_\nu(\boldsymbol{r}) \tag{2.1}$$

In the case of a bulk crystal, the envelope functions are simply plane waves $f_i(\boldsymbol{r}) = e^{i\boldsymbol{k_i}\cdot\boldsymbol{r}}$. In the presence of an external potential, like quantum wells due

to material composition modulation, or an applied electric field, the envelope function $f_i(\boldsymbol{r})$ needs some more degree of freedom. Assuming that the Bloch function $u_\nu(\boldsymbol{r})$ is identic in all material compositions, the following Schrödinger equation, written for the envelope function $f_i(\boldsymbol{r})$ only, applies:

$$\frac{-\hbar^2}{2m^*}\nabla^2 f_i(\boldsymbol{r}) + V(\boldsymbol{r}) f_i(\boldsymbol{r}) = E_i f_i(\boldsymbol{r}) \tag{2.2}$$

If we further assume that the materials are bulk in the two directions x and y of $\boldsymbol{r} = (x, y, z)$ with z being the growth direction, the envelope function can be further seperated with the ansatz:

$$f_{n\boldsymbol{k}_\perp}(\boldsymbol{r}) = \frac{1}{\sqrt{A}} e^{i\boldsymbol{k}_\perp \cdot \boldsymbol{r}} \varphi_n(z) \tag{2.3}$$

with $\boldsymbol{k}_\perp$ the vector $(k_x, k_y) = (k_x, k_y, 0)$ in the x-y plane and $V(\boldsymbol{r}) = V(z)$ only varying along z. As $\boldsymbol{k}_\perp \cdot \boldsymbol{r} = k_x x + k_y y$ does not depend on z, inserting equation (2.3) into equation (2.2)

$$\frac{-\hbar^2}{2m^*}\varphi_n(z)\left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2}\right) e^{i\boldsymbol{k}_\perp \cdot \boldsymbol{r}} + \frac{-\hbar^2}{2m^*} e^{i\boldsymbol{k}_\perp \cdot \boldsymbol{r}} \frac{d^2}{dz^2}\varphi_n(z) + e^{i\boldsymbol{k}_\perp \cdot \boldsymbol{r}} V(z)\varphi_n(z) = E_{n,\boldsymbol{k}_\perp} e^{i\boldsymbol{k}_\perp \cdot \boldsymbol{r}} \varphi_n(z) \tag{2.4}$$

leads to the one-dimensional Schrödinger equation along the z axis

$$\frac{-\hbar^2}{2m^*}\frac{d^2}{dz^2}\varphi_n(z) + V(z)\varphi_n(z) = E_n \varphi_n(z) \tag{2.5}$$

and free[1] motion in the x-y plane. The total energy eigenvalues are then

$$E_{n,\boldsymbol{k}_\perp} = E_n + \frac{\hbar^2 \boldsymbol{k}_\perp^2}{2m^*} \tag{2.6}$$

with only the subband energies E_n depending on the shape of $V(z)$. In general, the effective mass m^* will not be the same in two different materials A and B building an interface at $z = z_{AB}$. Therefore, equation (2.5) has to be solved in

[1] free in the sense of as in the bulk semiconductor

each material layer with the boundary conditions

$$\varphi_n^A(z_{AB}) = \varphi_n^B(z_{AB}) \qquad \text{and} \qquad \frac{1}{m^{*A}}\frac{d\varphi_n^A}{dz}(z_{AB}) = \frac{1}{m^{*B}}\frac{d\varphi_n^B}{dz}(z_{AB}) \tag{2.7}$$

to ensure that the wavefunction is contiguous and the probability current is conserved across the interface. Equation 2.5 can only be solved analytically for the most simple cases, like one finite, symmetric quantum well. Generally, numerical methods are used, e. g. the so-called transfer matrix method. Within the scope of this work, the software *sewself* {sewself} written by Jérôme Faist was used. *Sewself* is an implementation of the transfer matrix method mentioned before, combined with a Poisson solver to allow for a self-consistent Coulomb potential and the inclusion of nonparabolicity [42] for several material systems.

2.1.2 Absorption coefficient of a single quantum well

For the absorption coefficient, we use Fermi's golden rule to calculate the transition rate from an initial state i to a final state f with the interaction Hamiltonian $H' = \frac{e}{2m^*}(\boldsymbol{A}\cdot\boldsymbol{p} + \boldsymbol{p}\cdot\boldsymbol{A})$ under the influence of an external electromagnetic field.

$$W_{if} = \frac{2\pi}{\hbar}|\langle\psi_i|H'|\psi_f\rangle|^2\delta(E_f - E_i - \hbar\omega) \tag{2.8}$$

In our case of one-band effective-mass model, the interaction is correctly described by using the effective mass m^* in the expression above for H'. The incident electromagnetic wave is described by the plane wave

$$\boldsymbol{E}(\boldsymbol{r},t) = E_0\boldsymbol{e}\cos(\boldsymbol{q}\cdot\boldsymbol{r} - \boldsymbol{\omega t}) = \frac{E_0\boldsymbol{e}}{2}\left(e^{i(\boldsymbol{q}\cdot\boldsymbol{r}-\omega t)} + e^{-i(\boldsymbol{q}\cdot\boldsymbol{r}-\omega t)}\right) \tag{2.9}$$

where $\boldsymbol{E}(\boldsymbol{r},t)$ is the electric field as function of position and time, $\boldsymbol{q}$ the propagation vector, and $\boldsymbol{e}$ the linear polarisation vector perpendicular to $\boldsymbol{q}$. We choose a gauge in which the associated electrical potential ϕ is zero, producing the electric field by $\boldsymbol{E} = \frac{-\partial\boldsymbol{A}}{\partial t}$, and set the vector potential $\boldsymbol{A}$ as

$$\boldsymbol{A}(\boldsymbol{r},t) = \frac{E_0\boldsymbol{e}}{\omega}\sin(\boldsymbol{q}\cdot\boldsymbol{r} - \omega t) = \frac{iE_0\boldsymbol{e}}{2\omega}e^{i(\boldsymbol{q}\cdot\boldsymbol{r}-\omega t)} + \text{c.c.} \tag{2.10}$$

We can further apply the dipole approximation, which necessitates that the characteristic length of the electronic system is much smaller than the wavelength of

the radiation. This is very well fulfilled, as the wavelength is of the order of several micrometers and widths of quantum wells are typically tens of nanometers. Under this approximation, $\boldsymbol{A}$ and $\boldsymbol{p}$ commute, which simplifies the interaction Hamiltonian to $H' = \frac{e}{m^*}\boldsymbol{A}\cdot\boldsymbol{p}$. Inserting it into equation (2.8), we obtain

$$W_{if} = \frac{2\pi}{\hbar}\frac{e^2E_0^2}{4m^{*2}\omega^2}|\langle\psi_i|\boldsymbol{e}\cdot\boldsymbol{p}|\psi_f\rangle|^2\delta(E_f - E_i - \hbar\omega) \tag{2.11}$$

Inserting the envelope function expression (equation 2.1) into the matrix element $\langle\psi_i|\boldsymbol{e}\cdot\boldsymbol{p}|\psi_f\rangle$ and taking advantage of the fact that the Bloch functions $u_\nu(\boldsymbol{r})$ vary rapidly over $k_\perp^{-1}$ [41] leads to

$$\langle\psi_i|\boldsymbol{e}\cdot\boldsymbol{p}|\psi_f\rangle = \boldsymbol{e}\cdot\langle u_\nu|\boldsymbol{p}|u_{\nu'}\rangle\langle f_n|f_{n'}\rangle + \boldsymbol{e}\cdot\langle u_\nu|u_{\nu'}\rangle\langle f_n|\boldsymbol{p}|f_{n'}\rangle \tag{2.12}$$

which splits the optical transitions into two types. The first term describes interband transitions and the second intersubband transitions, which can be seen as follows: In the case of an interband transition, the band indices of the initial and final state ν and ν' are different, so the overlap integral of the Bloch functions, $\langle u_\nu|u_{\nu'}\rangle$ vanishes and only the first term remains. If the final and initial state are in the same band, $\nu = \nu'$ makes the first term vanish and $\langle u_\nu|u_{\nu'}\rangle = 1$. The remaining dipole matrix element of the envelope functions $\langle f_n|\boldsymbol{e}\cdot\boldsymbol{p}|f_{n'}\rangle$ describes the optical intersubband transition.

Writing f_n in the form of equation 2.3 and explicitly putting the dot product $\boldsymbol{e}\cdot\boldsymbol{p}$ evaluates to

$$\begin{aligned}
&\langle f_{n\boldsymbol{k}_\perp}|\boldsymbol{e}\cdot\boldsymbol{p}|f_{n'\boldsymbol{k}'_\perp}\rangle \\
&\qquad = \frac{1}{A}\int d^3r e^{-i\boldsymbol{k}_\perp\cdot\boldsymbol{r}}\varphi_n^*(z)\left[e_xp_x + e_yp_y + e_zp_z\right]e^{i\boldsymbol{k}'_\perp\cdot\boldsymbol{r}}\varphi_{n'}(z) \\
&\qquad = \left[e_x\hbar p_x + e_y\hbar p_y\right]\delta_{n',n}\delta_{\boldsymbol{k}'_\perp,\boldsymbol{k}_\perp} + e_z\delta_{\boldsymbol{k}'_\perp,\boldsymbol{k}_\perp}\int dz\varphi_n^*(z)p_z\varphi_{n'}(z)
\end{aligned} \tag{2.13}$$

From these three terms, only the term proportional to e_z does not vanish for different initial and final states ($n \neq n'$). This is the well known *polarisation selection rule* of intersubband transitions only allowing optical transitions if the electrical field has a component perpendicular to the layers. Reintroducing the

bra-ket notation for the relevant integral in equation 2.13

$$\langle n|p_z|n'\rangle = \int dz \varphi_n^*(z) p_z \varphi_{n'}(z) \tag{2.14}$$

one defines the dimensionless oscillator strength $f_{nn'}$, a commonly used quantity in optical spectroscopy, as

$$f_{nn'} = \frac{2}{m^* \hbar \omega_{n'n}} |\langle n|p_z|n'\rangle|^2 = \frac{2m^* \omega_{n'n}}{\hbar} |\langle n|z|n'\rangle|^2 \tag{2.15}$$

obeying the sum rule

$$\sum_{n'} f_{nn'} = 1 \tag{2.16}$$

for the sum running over all final states n'. Originally, West and Eglash [1] used the free electron mass m_0, which then means that equation 2.16 sums up to $\frac{m_0}{m^*}$ instead of 1. The same nomenclature has to be chosen if one takes the nonparabolicity into account, where it is no longer possible to completely separate interband and intersubband transitions [42].

The simple form of the matrix element of equation 2.14 allows to calculate the absorption coefficient. In general, the absorption coefficient α is defined as the ratio of electromagnetic energy being absorbed per *volume* and time, and the intensity of incident light I. In the case of a quasi-two-dimensional system like the 2D electron gas in the quantum well we are considering, it is more common to define a 2D absorption coefficient α_{2D} by calculating the ratio of the absorbed energy per *area* (A) and time, and the incident intensity. Therefore, α_{2D} is dimensionless, where the (3D) α has the dimension of inverse length. If we sum over all possible initial and final states n, n' and include stimulated emission in completion to absorption, the absorption coefficient is then written as

$$\alpha_{2D} = \frac{\hbar\omega}{IA} \sum_{n,n'} \sum_{\boldsymbol{k}_\perp} \frac{2\pi}{\hbar} |\langle n| \frac{e}{m^*} \boldsymbol{A} \cdot \boldsymbol{p} |n'\rangle|^2 \\ [f(E_n(\boldsymbol{k}_\perp)) - f(E_{n'}(\boldsymbol{k}_\perp))]\, \delta(E_{n'}(\boldsymbol{k}_\perp) - E_n(\boldsymbol{k}_\perp) - \hbar\omega) \tag{2.17}$$

where the $f(E_n)$ terms the states occupation by the Fermi-Dirac distribution function. Inserting the $I = \epsilon_0 c n \frac{E_0^2}{2}$ and $\boldsymbol{A}$ (equation 2.10) as well as the polarisation selection rule (equations 2.13,2.14) and changing the summation over k

into a 2D-integral leads to

$$\alpha_{2\mathrm{D}} = \frac{\pi e^2}{\epsilon_0 c n \omega m^{*2}} \sum_{n,n'} \frac{2}{(2\pi)^2} \int d^2k_\perp |\langle n|p_z|n'\rangle|^2 \left[f(E_n) - f(E_{n'})\right] \delta(E_{n'} - E_n - \hbar\omega) \tag{2.18}$$

Assuming a parabolic in-plane dispersion, this can be further simplified by performing the integration over the Fermi-Dirac distributions analytically - which is only possible in two dimensions. Additionally, we replace the δ function with a Lorentzian with a FWHM of 2Γ and express the matrix element by the introduced oscillator strength (equation 2.15) where we assume $\omega_{nn'} \approx \omega$ to get the final expression for the absorption coefficient:

$$\alpha_{2\mathrm{D}} = \frac{e^2 k_B T}{2\epsilon_0 c n \hbar} \sum_{n,n'} f_{nn'} \ln\left(\frac{1 + e^{\frac{E_F - E_n}{k_B T}}}{1 + e^{\frac{E_F - E_{n'}}{k_B T}}}\right) \frac{\Gamma/\pi}{(E_{n'} - E_n - \hbar\omega)^2 + \Gamma^2} \tag{2.19}$$

For zero temperature - where only the lowest state is occupied - and only considering the most important transition, say $1 \rightarrow 2$, a simple equation results, as the $ln\left(\frac{\cdot}{\cdot}\right)$ in equation 2.19 can be approximated by $\frac{E_F - E_1}{k_B T}$:

$$\alpha_{2\mathrm{D}} = \frac{n_s e^2 \hbar}{2\epsilon_0 c n m^*} f_{12} \frac{\Gamma}{(E_2 - E_1 - \hbar\omega)^2 + \Gamma^2} \qquad \text{at } T = 0 \tag{2.20}$$

For a multipass waveguide, as used in this work (section 3.3.1, figure 3.3), the transmission is then approximately

$$T \approx e^{-CMN\alpha_{2\mathrm{D}} \frac{\sin^2\theta}{\cos\theta}} \tag{2.21}$$

In equation 2.21, N denotes the number of periods, M is the number of passes through the active layer, Θ is the incident angle (between the growth axis and the propagation direction of the light) and C a factor between 0 and 2, including the effect of the standing wave pattern: 0 for the active zone being in a node, 2 for it being in a crest, or 1 if the active zone is thick compared to the wavelength in the material (the standing wave pattern is average out, then). Expressed in the geometrical dimensions of the sample (see figure 3.3), its length L and the total thickness d, including the substrate, equation 2.21 becomes

$$T \approx e^{-C\frac{L}{d}N\alpha_{2\mathrm{D}} \sin\theta} \tag{2.22}$$

2.2 Detector physics

Where not cited otherwise, the description in this section is based on the overview by Rogalski [43], the chapter by Liu [7] and the memo by Zogg [44].

Detecting (infrared) light means transforming the incident radiation into another physical quantity suitable for further processing. Or in other words, generating an electrical signal as function of the intensity of the incident light. One roughly divides detectors into two types based on their main working scheme:

Thermal detectors operate on the following simple principle. Incoming radiation heats the active part of the detector. The induced temperature change is determined by measuring any temperature dependant property such as pyroelectric voltage, thermoelectric voltage, or electrical resistance. This type of detection measures the incident radiation power and is largely wavelength independent besides the influence of the absorption coefficient of the surface coating, which is in general made as "black" as possible.

Photon detectors convert photons into electrons and then "count" the generated electrons passing between two electrical contacts of the device. The number of generated electrons per number of photons falling on the detector during the observation period determines the detector's efficiency. This quantity is called the overall quantum efficiency

$$\eta_{\text{tot}} = \frac{N_e}{N_p} \tag{2.23}$$

As it is often more practical to handle radiation power and electrical current instead of counting elementary particles, one generally uses the (current) spectral responsivity $\mathcal{R}$, measuring the electrical current at the detector's terminals per illumination power. The former being simply $I = \frac{eN_e}{\tau}$ where e is the electron charge and τ the observation time and the latter $P_i = \frac{h\nu N_p}{\tau} = \frac{hc}{\lambda\tau}N_p$ with Planck's constant h, the speed of light c, and the wavelength of the observed light λ, we can write

$$\mathcal{R} = \frac{I}{P_i} = \frac{\lambda e}{hc}\frac{N_e}{N_p} = \frac{\lambda e}{hc}\eta_{tot} \tag{2.24}$$

Contrary to the overall quantum efficiency η_{tot} which is directly comparable over the whole wavelength range, the responsivity $\mathcal{R}$ changes proportionally with the wavelength for a constant η_{tot}.

The overall quantum efficiency η_{tot} is generally the product of several specific quantum efficiencies or probabilities, depending on the concrete working principle

of the photon detector. These include the absorption probability of an electron passing the active zone $\eta_{\text{abs}} = 1 - e^{-\alpha d}$ where α is the absorption coefficient and d the thickness (in 3D units) or, counting per 2D layer, $\eta_{\text{abs}} = 1 - e^{-N\alpha_{\text{2D}}\frac{\sin^2\Theta}{\cos\Theta}}$. Another crucial factor is the fraction of electrons, once excited by absorbing incident photons, contributing to electrical current: the escape probability p_e. In the case of a multiple period structure, the factor $1/N$, with N being the number of periods, has to be included. If there is some gain mechanism like the photoconductive gain in QWIPs, $g = \frac{p_e}{Np_c}$ with p_c being the capture probability, this has to be included too. In that case, the factor $1/N$ is already included in g, so one has to be careful to include each factor exactly once.

Up to this point, the influence of noise on the performance of a detector has not been considered. As all measurement processes are inevitably affected by noise, the description of performance needs to account for noise, too. A detector, operated under given conditions like detector temperature, diffused background light, bandwidth or integration time of readout electronics, and others, produces a signal $I = I(t)$ correlated to the intensity of the light which should be detected $P = P(t)$. Additionally, there will also be a current which has no correlation with the incoming radiation. One commonly defines the so called "Signal-to-noise-ratio" S/N as the ratio of these two quantities:

$$\text{S/N} = \frac{I_{\text{signal}}}{I_{\text{noise}}} \tag{2.25}$$

In the context of photodetectors, the same information is expressed as "Noise Equivalent Power" *NEP* which is defined as the power of the light falling on the detector needed to produce the same signal strength as produced by noise sources.

$$NEP = \frac{P}{\text{S/N}} \qquad (I_{\text{noise}} = \mathcal{R} \cdot NEP) \tag{2.26}$$

To obtain a quantity which can be used to characterise a detector independently of its surface or the electrical measurement bandwidth, the inverse of the *NEP* is scaled with these two, which leads to the definition of the detectivity $\mathcal{D}^*$.

$$\mathcal{D}^* = \frac{\sqrt{A\Delta f}}{NEP} \tag{2.27}$$

A is the detector's surface and Δf the electrical bandwidth. Inserting equation

2.26 gives

$$\mathcal{D}^* = \frac{\mathcal{R}\sqrt{A\Delta f}}{I_{\text{noise}}} \tag{2.28}$$

the detectivity expressed in responsivity $\mathcal{R}$ and the noise current I_{noise}. Values of $\mathcal{D}^*$ are normally expressed in units of $(\text{cm}\sqrt{\text{Hz}}/\text{W})$ which is also called "Jones" in honour to R. Clark Jones, who proposed the inverse of the *NEP* as figure of merit for photodetectors in 1952 [45] and coined the term "normalised detectivity" for $\mathcal{D}^*$ as defined in equation 2.27 in 1959 [46].

2.2.1 Noise mechanisms

There are different origins of noise current that may limit the performance of photodetectors. In the context of photodetectors, one groups them in the 3 categories Johnson noise, $1/f$ noise and shot noise due to current flowing through a device. The latter is further subcategorised based on the origin of the current, either dark current noise due to the current produced by the applied bias voltage (without illumination) or photon noise associated with the current induced by incident photons.

Although $1/f$ noise is inherent to most physical and electronic processes, it is seldom the limiting noise contribution, as one normally manages to measure at characteristic frequencies which are high enough to sufficiently suppress $1/f$ noise. Therefore, it will be neglected in the following discussion.

In a resistive device, the Brownian motion of charged particles activated by the thermal energy leads to Johnson noise whose (flat) spectral power density is

$$p_{\text{noise}} = 4k_B T\Delta f \tag{2.29}$$

For an ohmic device with resistance R, or more generally in linear approximation at a working point with differential resistance R, this leads to the well known current and voltage noise spectral densities, respectively.

$$i^2_{\text{noise},J} = \frac{4k_B T\Delta f}{R}, \quad v^2_{\text{noise},J} = 4k_B TR\Delta f \tag{2.30}$$

Therefore, if Johnson noise is the dominant noise mechanism, detector perfor-

mance is described by the Johnson noise detectivity

$$\mathcal{D}^*_J(\nu) = \mathcal{R}(\nu)\sqrt{\frac{R_0 A}{4k_B T}} \tag{2.31}$$

In photoconducting devices like QWIPs, the contribution from Johnson noise is usually negligible compared to dark current noise and photon noise. In QCDs or other photovoltaic detectors, Johnson noise normally dominates over photon noise at elevated temperatures due to the absence of dark current.

The noise mean square current generated by a dark current I_{dark} can be described by

$$i^2_{\text{noise,dark}} = u\, e g_{\text{noise}} I_{\text{dark}} \Delta f \tag{2.32}$$

where e is the electron charge and u a constant describing the noise generation mechanism: 2 in the case of generation (photodiodes) or 4 for generation-recombination noise (QWIP, photoconductors). The noise gain g_{noise} is approximately the same as the photoconductive gain in conventional photoconductors, whereas for QWIPs they are different [47–50]. Hence, the detectivity in the dark current limited regime is

$$\mathcal{D}^*_{\text{dark}}(\nu) = \frac{\mathcal{R}(\nu)}{\sqrt{u\, e g_{\text{noise}} J_{\text{dark}}}} \tag{2.33}$$

The current induced by incident light, ambient light or blackbody radiation of the environment (or the surfaces of non-cooled optical elements!) also produces noise via the same mechanism as the dark current noise treated above. Therefore, $J_{\text{dark}} = I_{\text{dark}}/A$ is replaced with the current density due to the background illumination J_{BG} given by

$$J_{\text{BG}} = e\int \eta(\lambda)\frac{d\Phi_{\text{BG}}(\lambda)}{d\lambda}d\lambda = e\int \eta(\nu)\frac{d\Phi_{\text{BG}}(\nu)}{d\nu}d\nu \tag{2.34}$$

with $\eta = \eta_{\text{tot}}$ the quantum efficiency defined in 2.24 and $\frac{d\Phi_{\text{BG}}(\nu)}{d\nu}$ the background photon flux spectral density. For the background radiation emitted by the idealised blackbody at temperature T_{bb} and a full cone of view (FOV) angle Θ, it

is related to the blackbody emissivity

$$\begin{aligned} K(\nu, T_{\mathrm{bb}})d\nu &= \frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{k_B T_{\mathrm{bb}}}}-1}d\nu \\ K(\lambda, T_{\mathrm{bb}})d\lambda &= \frac{2hc^2}{\lambda^5}\frac{1}{e^{\frac{hc}{k_B T_{\mathrm{bb}}\lambda}}-1}d\lambda \end{aligned} \tag{2.35}$$

by

$$\begin{aligned} h\nu \cdot \frac{d\Phi_{\mathrm{BG}}(\nu)}{d\nu} &= \pi \sin^2\frac{\Theta}{2} \cdot K(\nu, T_{\mathrm{bb}}) \\ \frac{hc}{\lambda} \cdot \frac{d\Phi_{\mathrm{BG}}(\lambda)}{d\lambda} &= \pi \sin^2\frac{\Theta}{2} \cdot K(\lambda, T_{\mathrm{bb}}) \end{aligned} \tag{2.36}$$

Putting J_{dark} and 2.24 in 2.33 , we obtain

$$\mathcal{D}^*_{\mathrm{BLIP}}(\nu) = \frac{\mathcal{R}(\nu)}{\sqrt{ue^2 g_{\mathrm{noise}} \int \eta(\nu')\frac{d\Phi_{\mathrm{BG}}(\nu')}{d\nu'}d\nu'}} = \frac{\eta(\nu)}{h\nu\sqrt{u g_{\mathrm{noise}} \int \eta(\nu')\frac{d\Phi_{\mathrm{BG}}(\nu')}{d\nu'}d\nu'}} \tag{2.37}$$

2.2.2 BLIP regime

The different noise sources described above have different behaviour as function of device temperature T. Johnson noise rises at least proportional with $\sqrt{T}$ - under the assumption of a constant $\mathcal{R}$. Since $\mathcal{R}$ generally decreases with rising temperature in semiconductors or other structures where electrical conduction is thermally activated, Johnson noise rises even faster. The same holds for dark current in biased operation, so dark current noise is also supposed to rise with temperature. On the other hand, photon noise is considered to be roughly[2] constant as function of device temperature below the temperature of the (blackbody) environment T_{bb} (otherwise, the detector would start to see its own blackbody radiation).

As sketched in figure 2.1, there exists a temperature T_{BLIP} (background limited IR performance), below which the otherwise dominant noise source is superseded by photon noise. As the different noise sources are statistically independent, they add up quadratically: $i^2_{\mathrm{noise}} = \sum_\gamma i^2_{\mathrm{noise},\gamma}$. This results in the total detectivity

[2] Exactly constant under the assumption of constant $\mathcal{R}(\lambda)$ under varying temperature

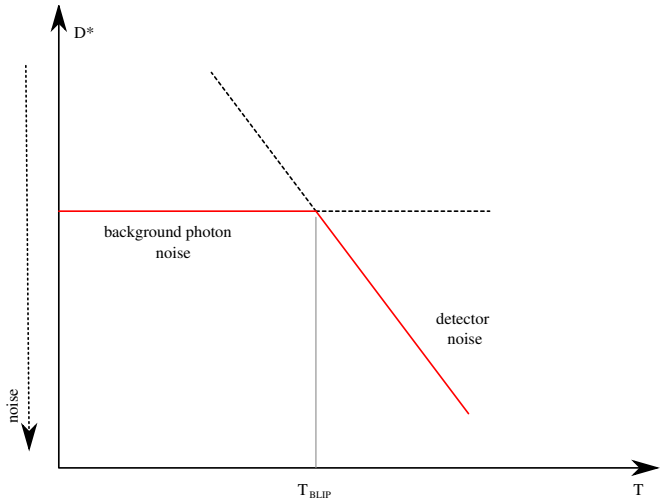


Figure 2.1: Definition of the BLIP temperature T_{BLIP} and detectivity $\mathcal{D}^*_{\mathrm{BLIP}}$ as the crossover from photon noise (background limited) to dark current or Johnson noise (temperature dependant).

including Johnson, dark current and photon noise

$$\mathcal{D}^*_{\mathrm{tot}} = \frac{\mathcal{R}}{\sqrt{\frac{4k_BT}{R_0A} + u\,e g_{\mathrm{noise}}\left(J_{\mathrm{dark}} + e\int \eta(\nu')\frac{d\Phi_{\mathrm{BG}}(\nu')}{d\nu'}d\nu'\right)}} \tag{2.38}$$

It is evident that cooling a detector much below the BLIP temperature is useless, as one can not further reduce the noise. Despite, of course, reducing the photon noise by using cooled optics and cooled apertures limiting the field of view (equation 2.36), which would shift T_{BLIP} downwards. As a result, it is necessary to specify under what assumptions of background temperature T_{bb} and field of view Θ background limited temperature T_{BLIP} and detectivity $\mathcal{D}^*_{\mathrm{BLIP}}$ have been measured or calculated[3].

2.2.3 Quantum cascade detectors (QCDs)

In a quantum cascade detector, an excited electron either falls back to its ground state, or tunnels through the extraction cascade into the next period (a more detailed describtion for a concrete structure is given in chapter 5). In analogy to the description of a photoconductive QWIP [7], the probability p_c to capture the electron in the ground state of the next period is unity and the escape probability

[3]Commonly used values are T_{bb} of 300 K or 500 K and Θ of 90° or 180°.

p_e is given by the fraction of excited electrons effectively travelling to the next period and not falling back.

Summarising the relevant equations of this section for this specific case leads to a responsivity of

$$\mathcal{R} = \frac{\lambda e}{hc}\eta_{\text{abs}}\frac{p_e}{N} \tag{2.39}$$

2.2.4 Variation of N and n_s

Let us consider the influence of changing the number of periods N for an otherwise unchanged structure with a constant doping concentration. Let us further assume, that η_{abs} is much smaller than 1, so e^{-x} can safely be approximated by $1 - x$. This leads to

$$\eta_{\text{abs}} = N\alpha_{\text{2D}}\frac{\sin^2\Theta}{\cos\Theta} \propto N \tag{2.40}$$

resulting in

$$\mathcal{R} \propto \eta_{\text{abs}}\frac{1}{N} \propto N \times \frac{1}{N} = \text{constant} \tag{2.41}$$

meaning that the responsivity does not change as function of the number of periods N as long as there is enough light throughout the structure (η_{abs} small). The responsivity being not influenced by changing the number of periods, the only source of change in the background photon limited detectivity $\mathcal{D}^*_{\text{BLIP}}$ (equation 2.37) is the noise gain $g_{\text{noise}} \propto 1/N$. $\mathcal{D}^*_{\text{BLIP}}$ then varies as

$$\mathcal{D}^*_{\text{BLIP}} = f(\mathcal{R}, g_{\text{noise}}) \propto \sqrt{N}. \tag{2.42}$$

The other relevant noise source in a QCD, Johnson noise, is accounted for in the Johnson noise detectivity $\mathcal{D}^*_J$ (equation 2.31). Again, $\mathcal{R}$ is constant but the device resistance R_0 will grow linearly in N. This leads then to

$$\mathcal{D}^*_J \propto \sqrt{N}, \tag{2.43}$$

showing the same behaviour as $\mathcal{D}^*_{\text{BLIP}}$. Considering the fact that stacking N periods of a detector structure together is the same thing as putting N current (noise) sources in series, which obviously reduces the total noise by a factor of $\sqrt{N}$[4]. Looking at figure 2.1 we conclude: raising the number of periods does not

[4] as would putting N voltage (noise) sources in parallel

change the BLIP temperature T_{BLIP}, as it just raises the total detectivity $\mathcal{D}^*_{\mathrm{tot}}$ by $\sqrt{N}$.

Doing a similar consideration for changing the doping density n_s under otherwise constant conditions and the same assumption of small absorption as above leads to

$$\mathcal{R} \propto n_s \tag{2.44}$$

as the absorption rises linearly in n_s. From equation 2.37 we see that

$$\mathcal{D}^*_{\mathrm{BLIP}} \propto \sqrt{n_s} \tag{2.45}$$

Contrary to a QWIP, where the dependence of dark current strength as function of doping density can be nicely derived [6, 51], the large number of states being involved in electrical transport through a quantum cascade detector makes it impossible to derive a simple expression for the device conductance $G = \frac{1}{R_0}$. Besides temperature and doping determining the population of the different states, the overlap of the states wavefunctions plays an important role [52]. For one given QCD structure, one could then use the model described in [52] - also used for the understanding of the temperature dependence of R_0A in section 5.2 - to numerically calculate the n_s dependence of R_0A and therefore $\mathcal{D}^*_J$ for a given temperature to optimise detector performance. Unfortunately, it was not possible anymore to apply this insight to the redesign of mid-IR QCDs in the scope of this work.

3 Experimental setups

3.1 Sample preparation

The samples studied during the work were all grown by means of molecular beam epitaxy (MBE). The structures for mid-infrared wavelength range were grown in the MBE facility at Neuchâtel. The material system used was lattice matched InAlAs-InGaAs on InP substrates. The sample for the terahertz range was obtained from the Cavendish Laboratory at the University of Cambridge, UK. This device was grown using the AlGaAs/GaAs material system on a semi-insulating GaAs substrate.

After verification of the layer thicknesses and periodicities by X-ray diffraction measurements, all samples were processed to square shaped mesas in the clean-room facilities of the Physics Institute in Neuchâtel. After removing the indium from the backside of the wafers[1] with mercuric chloride dissolved in dimethyl formamide ($HgCl_2$:DMF), the samples were thoroughly cleaned by rinsing them for several minutes with each of the following solvents: trichlorethylene, acetone, isopropanol, and de-ionised water. The actual structurisation consists of one etching step and one or more metal depositions.

Each of these processing steps is started by photolithographically transferring a mask pattern into a layer of photoresist. This mask serves then as either a protective mask for the etching or as sacrificial layer to enable the lift-off of deposited metal layers. In both cases, the sample is rinsed in isopropanol and DI water and then dried for one minute on a hotplate kept at 130 °C. After placing it on the head of a resist-spinner and letting cool down, a droplet of photoresist is placed on it and then spinned for 45 seconds. This results in an approximately 2 µm thick resist layer, which is dried once again for 1 minute at 85 °C. The sample is then brought in contact with the original mask under a mask aligner,

[1] with which the InP wafers are glued to the rotating holder during growth

and exposed to UV light. The development of the resist layer concludes the transfer from the Cr:quartz glass mask to the photoresist.

For the etching step, the resist is baked out at 120 °C for 5 minutes to make it harder. The chemical wet etch, aged etch[2] for the InP samples and sulfuric acid[3] for the GaAs samples, removes then the regions between the desired mesas. As a final step, the resist is removed with acetone.

The metalisations are done similarly, except for the fact that now the spinned-on photoresist is structured the other way around: After exposition and development, it stays everywhere where we do *not* want to have metal deposited. Then, a series of metals - Ti:1 − 2 nm, Au:12 nm, Ge:27 nm, Ag:50 nm, Au:3 00 − 500 nm - is deposited under high vacuum conditions. By swelling the subjacent photoresist with NMP[4] and then dissolving it with acetone, the unnecessary metal is lifted off.

Finally, the samples undergo rapid thermal annealing (RTA) to alloy ohmic contacts between the evaporated metal and the semiconductor underneath. The sample is rapidly heated up to 360 °C (InP) or 420 °C (GaAs), stabilised at this temperature for 1 minute and then rapidly cooled down again.

These samples are then further prepared according to the measurement to be done. First, the devices are cleaved into smaller rectangles to allow different kinds of measurements of the same device. Devices used in optical setups were either polished with 45° facets or equipped with a grating on top of the mesas during processing. This grating either replaces the full top metal with a metallic grating (sample A2879, chapter 4) or is etched into the semiconductor before deposition of the top metal (sample N657 and N658, chapter 5) as schematically shown in figure 3.2.

To allow for "macroscopic" handling, the samples are then soldered (In) or glued (varnish) onto copper submounts. Top contacts are connected to bigger contact pads via bond wires, whereas bottom contacts are directly connected to the copper block by a piece of indium foil. The exception is the high speed measurement, where stray inductances and capacitances are critical. In this case the sample was directly glued onto an SMA connector, where a very short bond wire between the middle contact of the coax connector and the top contact as

[2] $HBr : HNO_3 : H_2O$ in the ratio 1:1:10
[3] $H_2SO_4 : H_2O_2 : H_2O$ in the ratio 1:8:1
[4] NMP: N-Methyl-2-pyrolidone

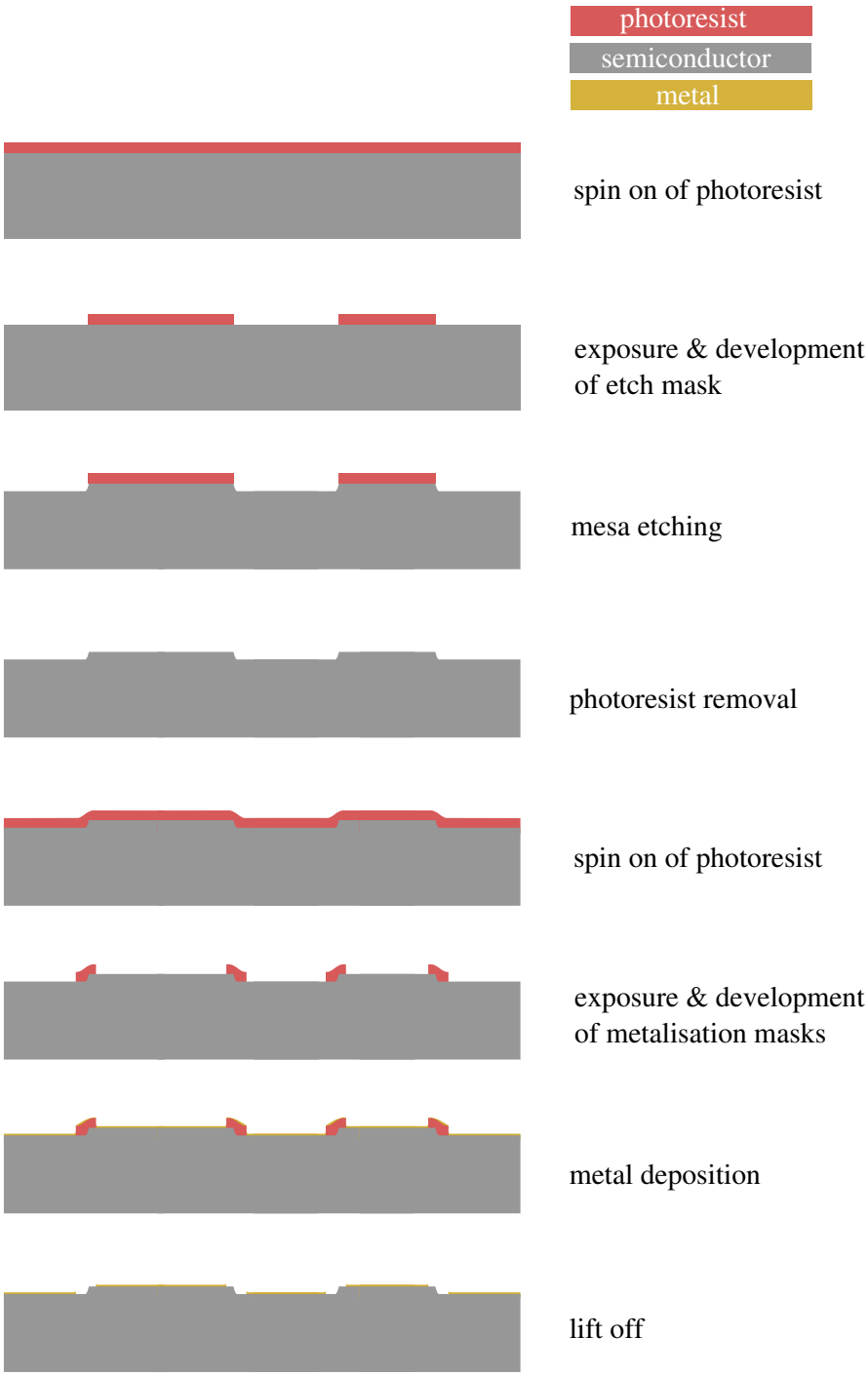


Figure 3.1: Processing steps to build mesas and contact them

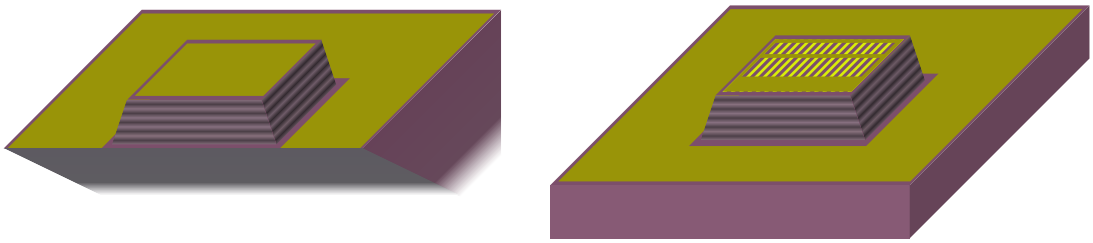


Figure 3.2: 45° and metal grating coupler

well as a piece of indium foil between the outer ring of the coax connector and the bottom contact of the sample provided electrical contact.

3.2 Electrical transport

Electrical transport was measured in form of $I - V$ curves in the temperature range between 6 K and 300 K. The copper submount with the soldered sample as described before is held on the cold finger of a flow cryostat. The device is cooled by either liquid helium or liquid nitrogen and the temperature is stabilised by a *Lakeshore 331 temperature controller* ⟨ls331⟩ driving the counterheating resistance on the cryostat finger.

To have conditions as dark as possible, the samples were optically and thermally shielded by an aluminium foil wrapped around the cryoshield of the cryostat. Additionally, $I-V$ curves have also been measured under the "illumination" of the blackbody radiation of the room background at approximately 300 K, entering through the cryostat window. If the device produces enough current to be directly seen in the "illuminated" $I - V$ curve, this allows one to estimate T_{BLIP} as the temperature below which the "illuminated" $I - V$ differs from the "dark" one.

For fast comparison between several mesas on one sample as well as to verify processing success, a needle setup has been used additionally. At temperatures between 10 K and room temperature, this allows fast access to the contacts on the chip with probe contacts positioned from outside the cryostat. This setup has the disadvantage that each change in temperature requires an additional lift and touch-down of the needles. Otherwise they may completely scratch the sample due to contraction and expansion of the needle holders and the cryostat window joints.

3.3 Spectral characterisation

The spectral characterisations were all done using Fourier transform infrared spectrometry (FTIR). The far infrared QCD and the QWIPs were characterised with a *Nicolet 800* ⟨nic800⟩ FTIR spectrometer, whereas for the mid-infrared QCDs all the measurements were done with a *Bruker IFS 66/S* ⟨ifs66⟩ FTIR spectrometer.

3.3.1 Absorption

To determine the absorption spectra of the different structures, the experimental setup as shown in figure 3.3 was used. The multipass transmission spectra through each sample is measured twice, with TE and TM polarised light. As the intersubband absorption takes only place in TM polarisation, the divison of the two spectra allows the extraction of the absorption spectra of the intersubband transitions. Therefore, the samples need to be prepared as follows: On each side of the chip, a 45° facet is polished, once from the upper side of the wafer and once from the backside, effectively building two parallel surfaces. Moreover, the backside of the wafer is also polished to roughnesses below the wavelength of the incident light. Due to the high refractive index of the semiconductor, total internal reflection takes place both on the polished backside and the upper side of the wafer. The light is zigzagging through this parallelepiped, passing N_{pass} times under an angle of 45° through the active zone, where the number of passes is simply the ratio between the sample length and thickness: $N_{\text{pass}} = L/d$.

For the THz device A2879 (see chapter 4), a different, in principle more sensitive measurement method was used: electromodulation. Depositing a metallic Schottky contact on top of the otherwise unprocessed chip and applying a negative bias voltage across the structure will deplete the active region, therefore cancelling the absorption. This allows one to use lock-in techniques together with the FTIRs step-scan capability: For each data point in the interferogram, the sample is periodically biased and the signal from the FTIR detector is fed to the lock-in amplifier, where it is demodulated in phase with the bias modulation and then fed back to the spectrometer.

All absorption measurements within the scope of this work were done at room temperature.

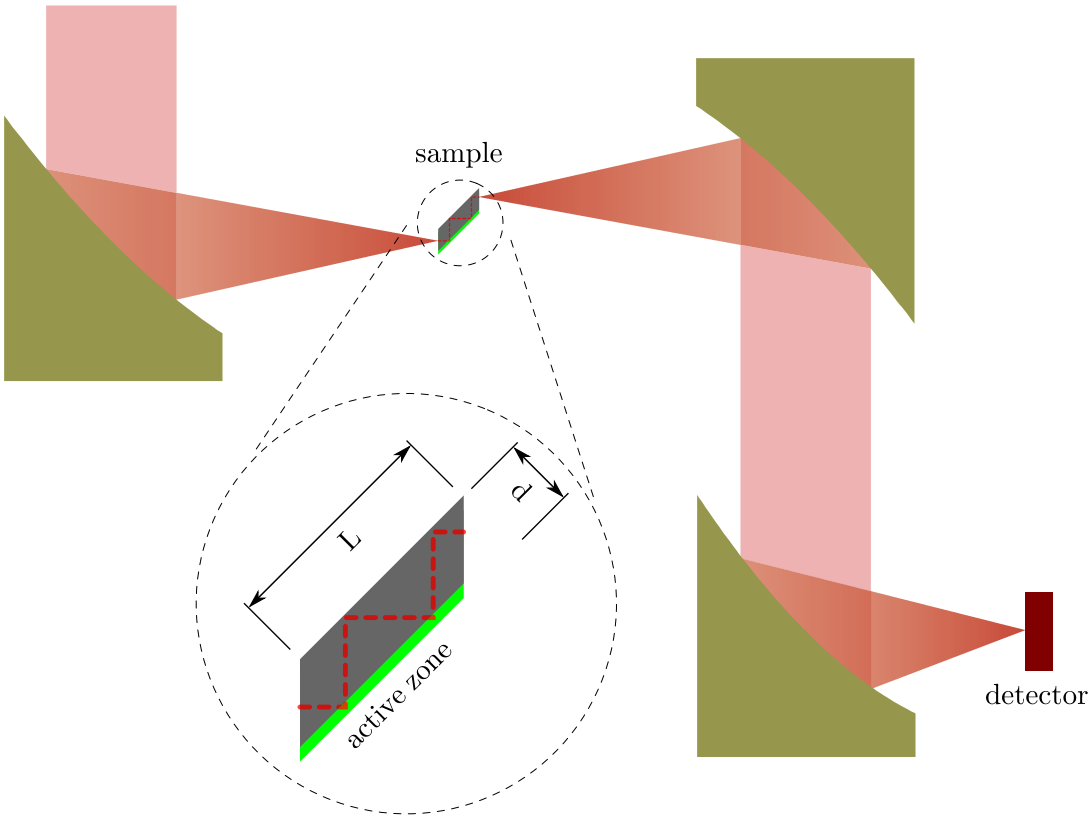


Figure 3.3: Absorption measurement setup

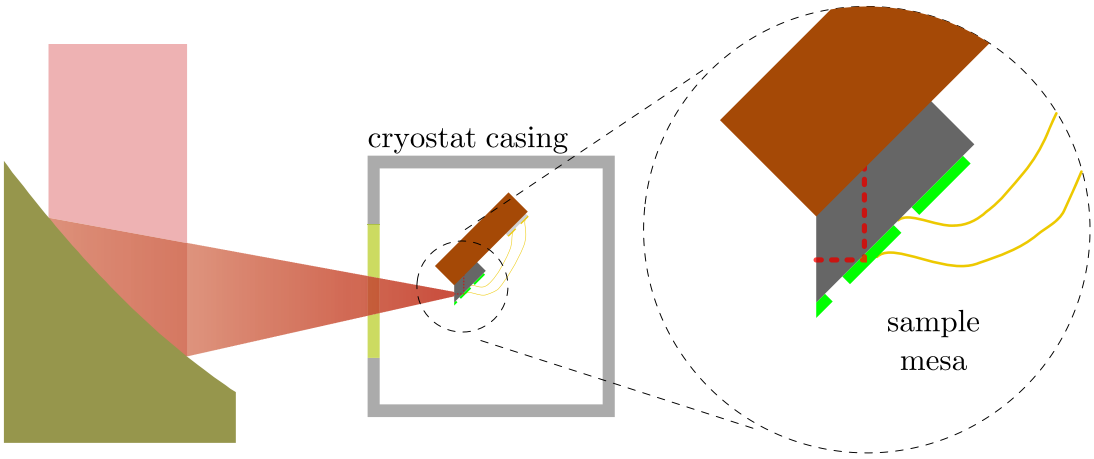


Figure 3.4: Photocurrent measurement setup

3.3.2 Photocurrent

Detection spectra of the different devices were measured in a setup similar to the one shown for the absorption measurements and are shown in figure 3.4. Instead of looking at the light passing through the device, the electrical signal at the terminals of the observed sample is amplified and fed back to the spectrometer, providing the signal otherwise coming from the built-in detector. The sample on its copper submount is attached to the cold finger of the cryostat and stabilised at temperatures between 10 K and 300 K. Depending on the observed wavelength range, we used different cryostat windows whose transmission spectra have to be taken into account. Thin coax cables provide contact for two mesas to the outside of the cryostat. These are connected with a pre-amplifier whose output is then fed to the analog-digital-converter (ADC) of the spectrometer. As mainly the photocurrent - which corresponds to the number of charges per time unit - is of interest, the ideal case is the use of a transimpedance amplifier. This provides a current sink with a low input impedance, thus effectively measuring current producing a voltage which is proportional to the current flowing into its input terminals. As we had no such amplifier during the first time of this work, I used a "normal" preamplifier with high input impedance, just providing voltage amplification. The photovoltage spectra obtained in this way have then to be corrected to be comparable between different devices, device temperatures or wavelengths, as under these conditions the conversion between current and voltage takes place in the impedance composed of the input impedance of the amplifier parallel to

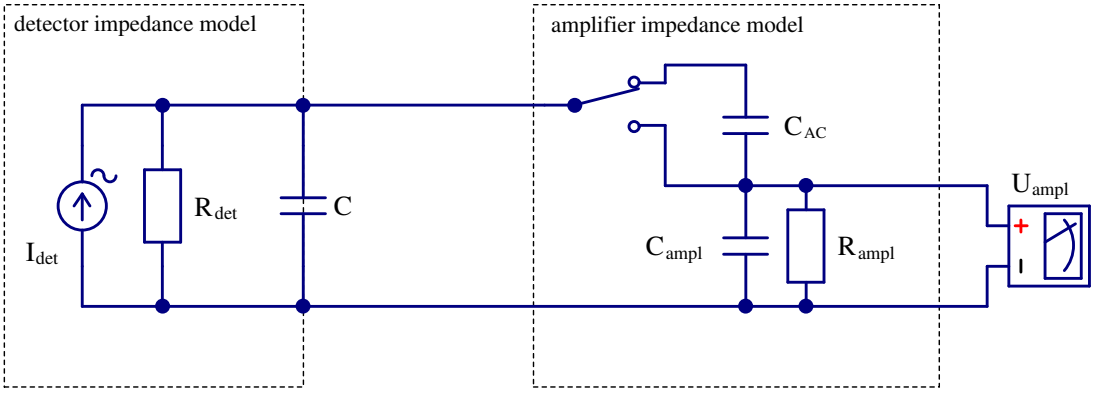


Figure 3.5: Equivalent circuit used in modelling the transimpedance behaviour of connecting the detectors directly to a voltage preamplifier.

the device resistance, device capacitance and cable capacitance as shown in the equivalent circuit of figure 3.5. Different wavelengths correspond to different spatial frequencies or wavenumbers[5] w of the interferogram and these get directly converted to temporal frequencies in the electrical signal due to spectrometer scanner velocity f_{scanner}[6] via

$$f = \mathsf{w} \times \frac{f_{\text{scanner}}}{15798.2} \tag{3.1}$$

Therefore, the obtained photovoltage spectrum is already multiplied by the amplification factor of the voltage amplifier and the magnitude of the (complex) transimpedance $\mathcal{Z}_{\text{trans}}$ describing the frequency response of the RC-circuit shown in figure 3.5

$$\begin{gathered} \mathcal{Z}_{\text{trans}}^{-1} = \mathcal{Z}_{\text{ampl}}^{-1} + \mathcal{Z}_{\text{det}}^{-1}\left(\mathcal{Z}_{\text{AC}}\mathcal{Z}_{\text{ampl}}^{-1} + 1\right) \\ \mathcal{Z}_{\text{det}}^{-1} = R_{\text{det}}^{-1} + i\omega C, \quad \mathcal{Z}_{\text{ampl}}^{-1} = R_{\text{ampl}}^{-1} + i\omega C_{\text{ampl}}, \quad \mathcal{Z}_{\text{AC}} = \frac{1}{i\omega C_{\text{AC}}} \end{gathered} \tag{3.2}$$

Where $\mathcal{Z}_{\text{det}}$ is the complex impedance of the detector, including cable capacitance, $\mathcal{Z}_{\text{ampl}}$ is the complex input impedance of the voltage amplifier, $\mathcal{Z}_{\text{AC}}$ describes the capacitor which cuts DC signal in AC-coupling mode of the amplifier, and as usual $\omega = 2\pi f$. In DC-coupling mode, this capacitor is bypassed

[5] classically measured in cm^{-1}

[6] The scanner velocity of FTIR spectrometers is expressed as the frequency of the interferogram signal from the reference laser, commonly a HeNe-laser emitting at $15\,798.2\,\text{cm}^{-1}$.

Table 3.1: Electrical values of the input circuit components of the *EG&G Model 5113 PRE-AMP* ⟨egg5113⟩

settings		values		
input time constant	coupling	C_{ampl}	R_{ampl}	C_{AC}
1 s	DC	25 pF	10 MΩ	
10 s	DC	25 pF	100 MΩ	
1 s	AC	25 pF	10 MΩ	0.1 µF
10 s	AC	25 pF	100 MΩ	0.1 µF

($\mathcal{Z}_{\mathrm{AC}} = 0$), thus equation 3.2 simplifies to

$$\mathcal{Z}_{\mathrm{trans}}^{-1} = \mathcal{Z}_{\mathrm{ampl}}^{-1} + \mathcal{Z}_{\mathrm{det}}^{-1} = \frac{1}{R_{\mathrm{ampl}}} + \frac{1}{R_{\mathrm{det}}} + i\omega\,(C_{\mathrm{ampl}} + C) \tag{3.3}$$

Even in the case of AC-coupling, equation 3.3 is valid approximately, as the term $|\mathcal{Z}_{\mathrm{AC}}\mathcal{Z}_{\mathrm{ampl}}^{-1}|$ is much smaller than 1 for typical frequency ranges $f > 10\,\mathrm{Hz}$ and values of C_{AC}, $C_{\mathrm{det}}, R_{\mathrm{det}}$ (Table 3.1).

Depending on the internal resistance of the detector, which itself is generally strongly varying with device temperature, this might have a rather large impact, as can be seen in figure 3.6. A sample at high temperatures (say room temperature) with an internal resistance far lower than the input impedance of the amplifier, as symbolised by the green curve, is entirely determined by its own resistance, resulting in a low but spectrally flat response. On the other hand, samples at cryogenic temperatures with resistances equal or higher than the input of the amplifier (blue, red & yellow curve), are strongly influenced by the spectral behaviour of the (stray) capacitances. The data in figure 3.6 has been calculated with equation 3.2 and also been verified with the electric circuit simulator *Quite Universal Circuit Simulator* {qucs}.

Even using a transimpedance amplifier, some spectral "distortion" may take place, as one normally sets filters (electrical high-pass, low-pass) to limit the spectral range in the interesting region which allows for a higher dynamic reserve in the analog-digital-conversion and the following numerical Fourier transformation[7]. This has been accounted for by dividing the photocurrent spectra with the electrical response of a series of simple $R - C$ terms, modelling the

[7] One does not want to "push" the signal into the numerical precision limiting range by "filling" the numerical "space" with a lot of noise.

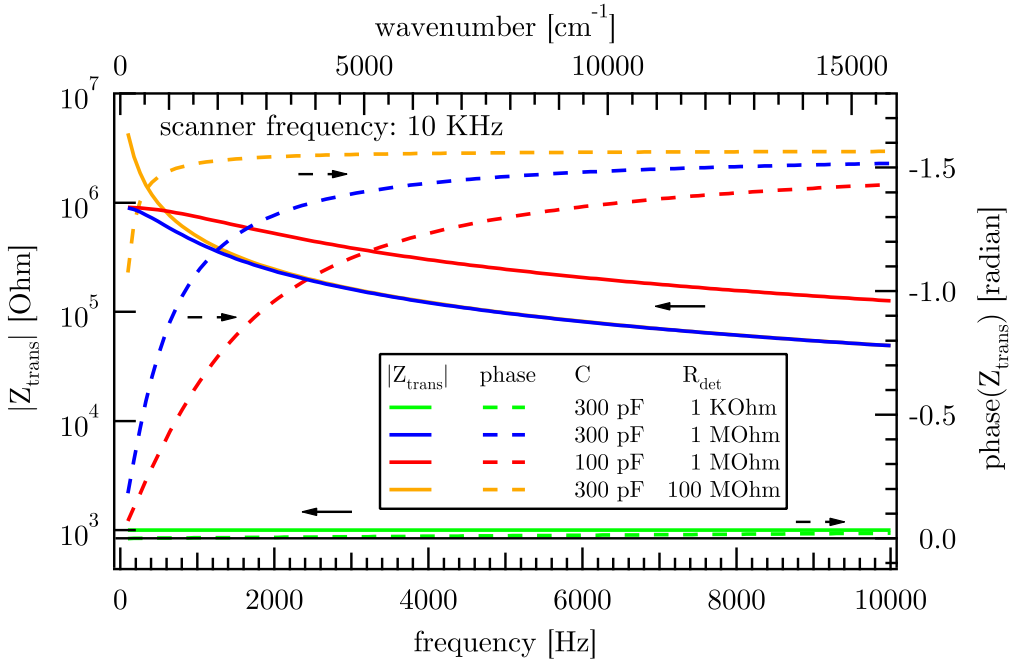


Figure 3.6: Calculated frequency dependence of phase and magnitude of $|Z_{\text{trans}}|$ according to equation 3.2. These curves are calculated for $C_{\text{AC}} = 0.1\,\mu\text{F}$, $R_{\text{amp}} = 10\,\text{M}\Omega$ and $C_{\text{ampl}} = 25\,\text{pF}$, the standard amplifier settings *AC-coupling* and *input time constant* = 1 s. On the top axis, the corresponding energy (in wavenumbers) for a spectrometer scanner velocity of 10 KHz is shown.

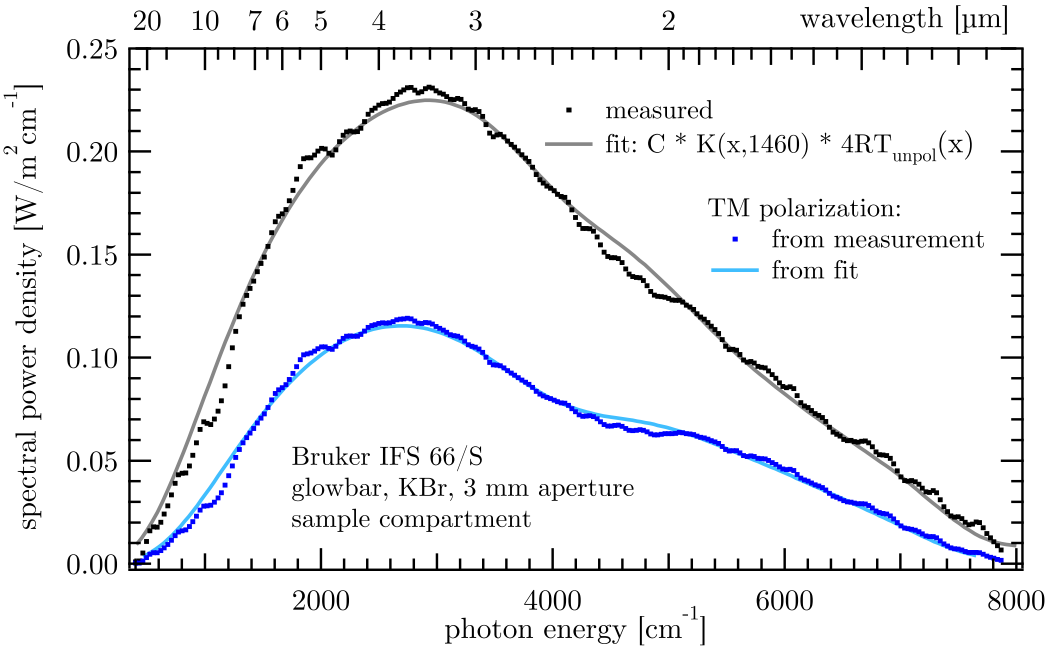


Figure 3.7: Effective spectra in the sample compartment of the *Bruker IFS 66/S* spectrometer with the glowbar as light source and the KBr beamsplitter, together with a fit of blackbody emissivity × beamsplitter efficiency $4RT$.

filter setting of the amplifier used, either the *Model 5113* ⟨egg5113⟩ or the *SR 570* ⟨sr570⟩. For the former, this was applied in addition to the correction of an equivalent input circuit as described above (equation 3.3, figure 3.6). The numerical implementation thereof is found in appendix C.1.

To obtain spectral responsivity, one has to divide the photocurrent spectra by the power spectrum of the specific illumination source in the spectrometer at the position of the cryostat. This includes the emission spectrum of that light source multiplied with the modulation depth spectrum $4RT$ of the beamsplitter and other losses from mirrors and other non perfectly transmitting objects in the optical path as, for instance, the cryostat window.

Figures 3.7 and 3.8 show reference measurements of the effective illumination spectra produced by the internal glowbar of the spectrometer together with the KBr beamsplitter. These measurements were done in step-scan mode with an *Ophir LaserStar* ⟨laserstar⟩ powermeter as detector, either at the focus of the (internal) beam path in the sample compartment or at the focus of our external

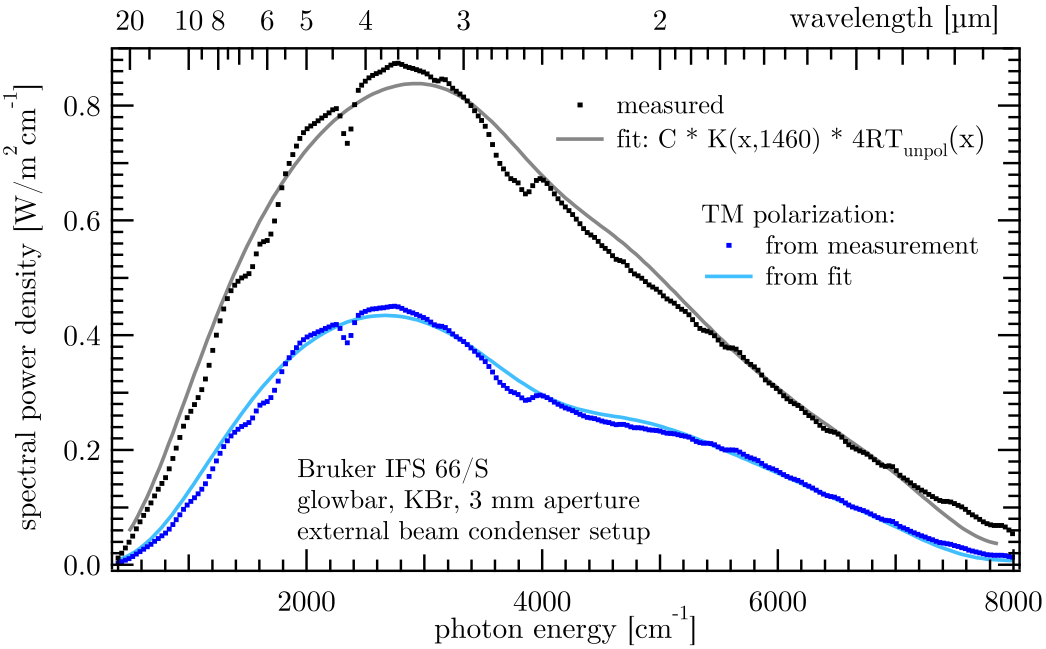


Figure 3.8: Effective spectra in our external beam condenser set at the *Bruker IFS 66/S* spectrometer with the glowbar as light source and the KBr beamsplitter, together with a fit of blackbody emissivity × beamsplitter efficiency $4RT$.

measurement setup with a shorter focal length, resulting in a 1:2 image of the aperture instead of 1:1 as in the sample compartment. Fitting the product of the blackbody emissivity $K(\nu, T_{\mathrm{bb}})$ (equation 2.35) and the KBr beamsplitters efficiency $4RT(\nu)$[8] to these measurements leads to glowbar temperatures T_{bb} of

measurement position	aperture diameter	T_{bb} (K)
sample compartment	3 mm	1459.6 ± 3.6
sample compartment	1 mm	1473.0 ± 7.3
external setup	3 mm	1442.6 ± 4.1

which agree reasonably well to a previously obtained value of $T_{\mathrm{bb}} = 1423\,\mathrm{K}$ by targeting the externally visible part of the glowbar with an *Ircon Ultimax UX-20* ⟨ux20⟩ pyrometer. Multiplying the blackbody emissivity with $\frac{1}{2} \cdot 4RT_{\mathrm{TM}}(\nu)$ or correcting the measured, unpolarised spectrum with $\frac{4RT_{\mathrm{TM}}(\nu)}{2 \cdot 4RT_{\mathrm{unpol}}(\nu)}$ finally leads to the effective spectrum (blue dots and curves in figures 3.7, 3.8) for quantum well based photodetectors due to the polarisation selection rule (equation 2.13). The TM polarisation with respect to the beamsplitter's optical plane corresponds to the TM polarisation with respect to the optical plane of the characterised device in both, the sample compartment as well as the external beam condenser setup.

The ZnSe window used as cryostat entrance window in all the spectral measurements in the mid-infrared has a transmission spectrum as shown in the measurement of figure 3.9.

3.4 Monochromatic responsivity

Another method to determine the absolute value of the responsivity is based on a direct illumination with a quantum cascade laser emitting at a wavelength in the detection range of the device. Its power at the detector's position can be measured exactly with a pyroelectric powermeter, for example. The detection signal produced by switching on or off the illumination by a chopper wheel can also be measured. Moving around the detector laterally allows to record an intensity profile in the detection plane. Together with the known effective mesa surface, the response as the ratio of produced current and illumination intensity

[8] These beamsplitter data has been provided by Bruker, calculated based on the layer structure of the beamsplitter.

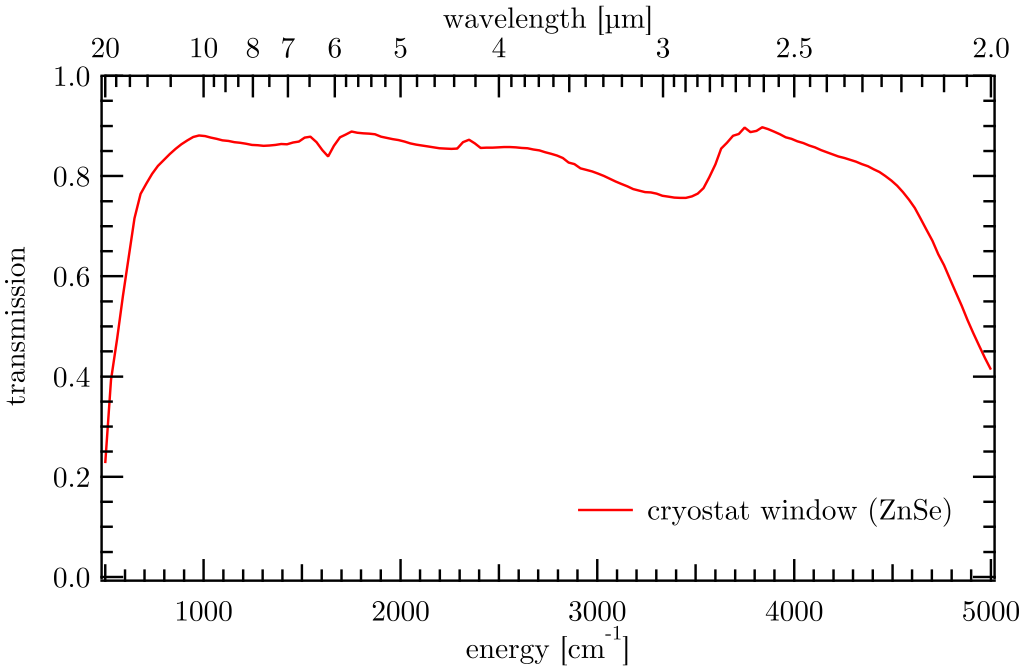


Figure 3.9: Measured transmission of the ZnSe entrance window of the cryostat. It is evident that measurements below 600 cm^{-1} and above 5000 cm^{-1} are seriously hampered by this transmission.

can be determined for one wavelength. As there were many unknowns in the spectrometer setup for the far infrared wavelength range, this method was applied in the case of the device A2879 (THz detector, chapter 4).

3.5 High speed characterisation

To verify that quantum cascade detectors are intrinsically really fast, another type of experiment has been performed. As we were not able to modulate suitable quantum cascade lasers with frequencies higher than 3 GHz, the modulation was generated optically by means of interference between the light of two different quantum cascade lasers operated at nearly the same wavelength in the detection range of the detector. Therefore, the beams of these two lasers have to be carefully aligned and parallelised to evenly overlap so the total light falling onto the sample is modulated with the difference of the optical frequencies of the two lasers (figure 3.10). From the detector, which is mounted on a 3-axis xyz-stage for precise alignment, the signal was fed into a low noise high frequency amplifier *Miteq*

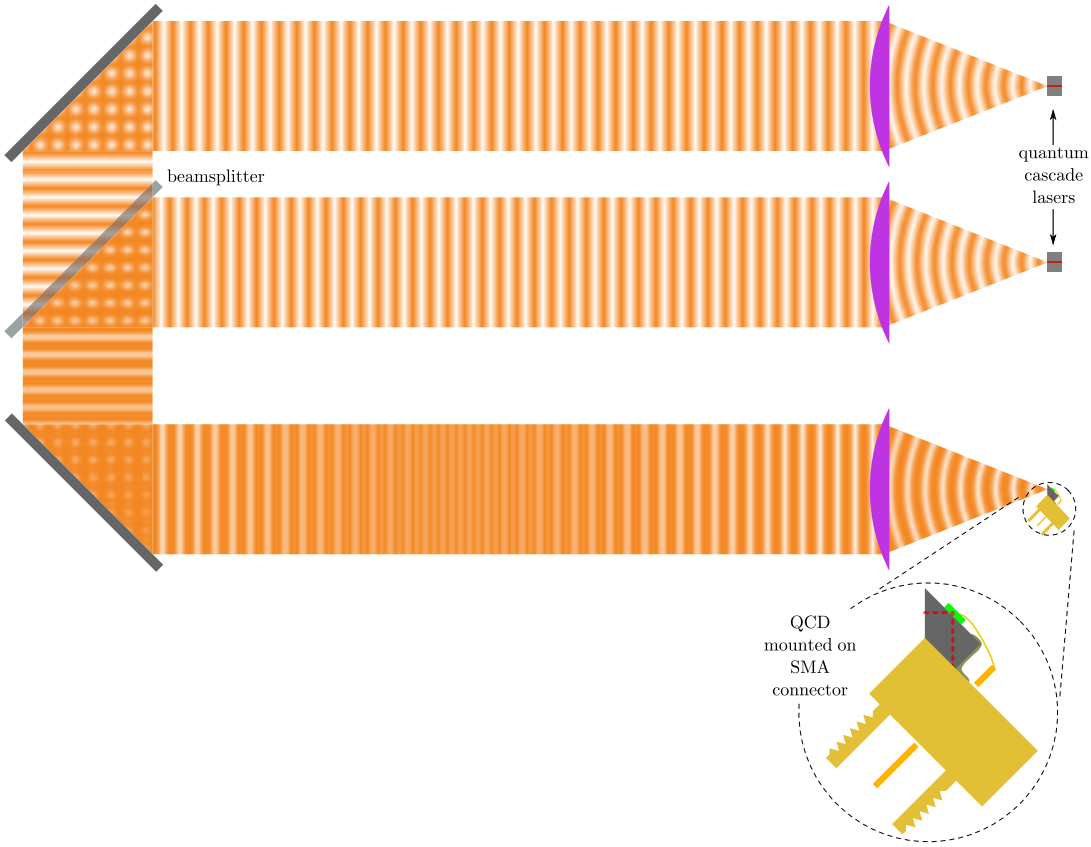


Figure 3.10: Schematic drawing of the optical heterodyning setup used for the highspeed measurements. The two QCL beams are overlaid from behind the beam splitter to finally reach the QCD.

ASF-5 ⟨miteq⟩ with a specified gain of 26 dB up to a cut-off frequency of 21 GHz, followed by a second one (*HP83006* ⟨hp83006⟩, 26 dB, 26 GHz) and finally visualised on a spectrum analysor (*Agilent E4407B* ⟨e4407b⟩, see figure 3.11). This allowed the measurement of the signal strength as function of frequency. The gain curves of both amplifiers are shown in figure 3.12. Alternatively, a similar setup with only one amplifier, a *Sonoma Model 317* ⟨s317⟩ (38 dB up to 2.5 GHz) and the "slower version" of the spectrum analyser (*E4402B* ⟨e4402b⟩, 3 GHz) were used.

Although the device was mounted as close as possible on an SMA coaxial connector to reduce the stray inductance of the bond wire, the impedance matching between the high quality coax cable and the detector mesa connected by a wire bond is far from perfect. I modelled this connection with the equivalent circuit shown in the inset of figure 3.11. This circuit corresponds to a second order

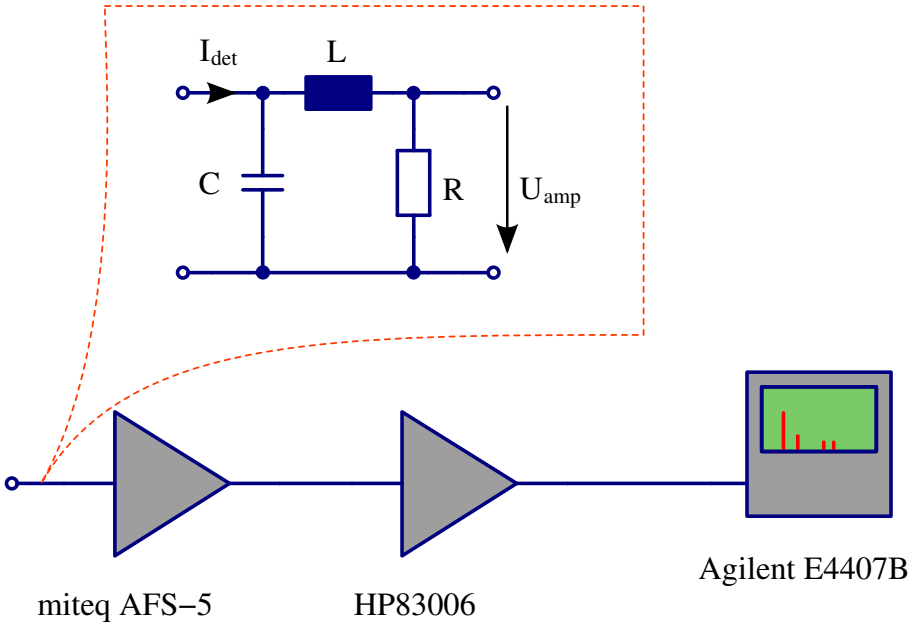


Figure 3.11: Schematic signal path from the detector to the spectrum analyser. The inset shows the equivalent circuit modeling the matching between the detector, taking into account the stray inductance L of the bond wire, the mesa capacitance C, and the coax cable with a perfectly matched impedance to the first amplifier of $R = 50\,\Omega$.

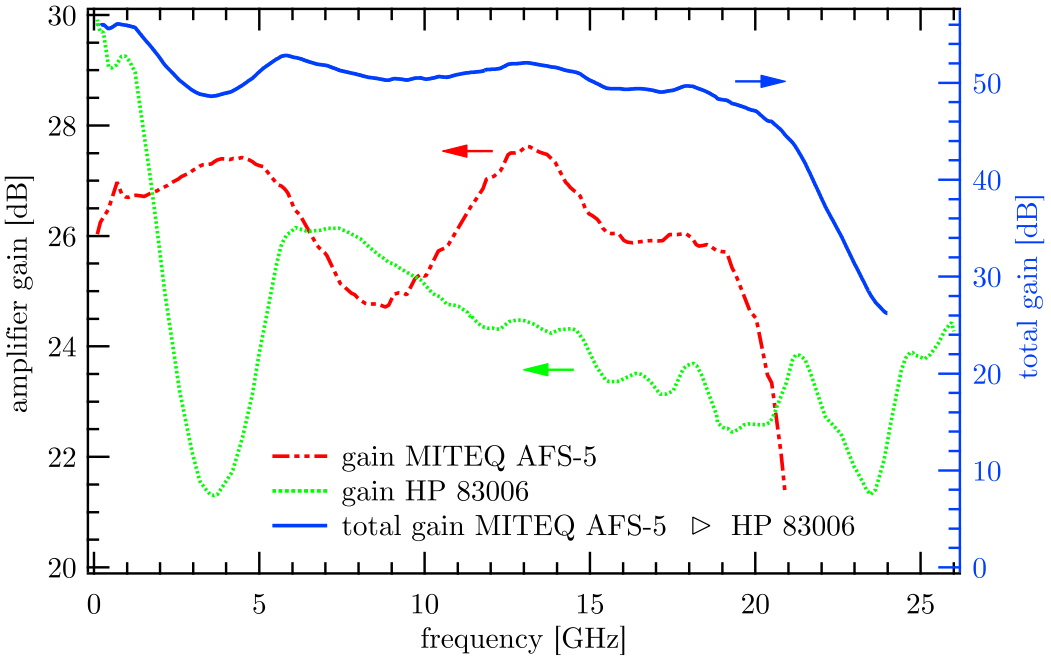


Figure 3.12: Frequency response (gain) of the two used highspeed 26 dB amplifiers, the low noise *Miteq AFS-5* ⟨miteq⟩ as well as the broadband *HP 83006* ⟨hp83006⟩, together with the total gain of the two in series (blue curve).

low-pass filter, whose frequency dependant complex impedance Z is given by

$$Z(\omega) = \frac{R}{1 + i\omega C\,(R + i\omega L)} \tag{3.4}$$

while the voltage at the amplifier input as function of the detector current reads

$$U_{\mathrm{amp}}(\omega) = I_{\mathrm{det}}|Z(\omega)| = \frac{IR}{|1 + i\omega C\,(R + i\omega L)\,|} \tag{3.5}$$

4 Far infrared QCD

4.1 Design

As there was no reported detection of THz radiation using intersubband transitions at that time, we tried to exploit the possibility to use structures similar to quantum cascade lasers as detectors, as "detecting" QCLs had been reported by the *Mesoscopic Physics* group here in Neuchâtel [22]. Therefore, we designed a structure specifically as detector for THz radiation which was then MBE-grown in the Cavendish Laboratory at the University of Cambridge. The growth sheet and the corresponding band structure simulation (self-consistently calculated with *Sewself* {sewself}) are shown in table 4.1 and figure 4.1, respectively.

The basic idea of the detection process is as follows: Once an electron is transferred from the ground state 0 to one of the excited states 4 or 5 by absorption of an incident photon with the corresponding energy, there are two possible paths of relaxation. Either the electron hops directly back to the ground state, not producing any electrically detectable quantity. Or it tunnels to state 3 and then further on through the other states in grey ($\rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$) to finally arrive in the ground state of the next period. In this case, an electrical charge (the electron) has been displaced laterally, which is the microscopic equivalent of an electrical current - the sought-after photocurrent. As the transitions from the ground state up to states 6 and 7 have also been identified in the spectral measurements, they are plotted in the graph, too. Their calculated transition energies are 27.5 meV and 34.2 meV, respectively.

The overall composition and the period of the superlattice were confirmed by X-ray diffraction. After growth, the samples were processed into square shaped mesas of different sizes (900, 450, 220 µm) using standard photolithography and wet etching as described in section 3.1.

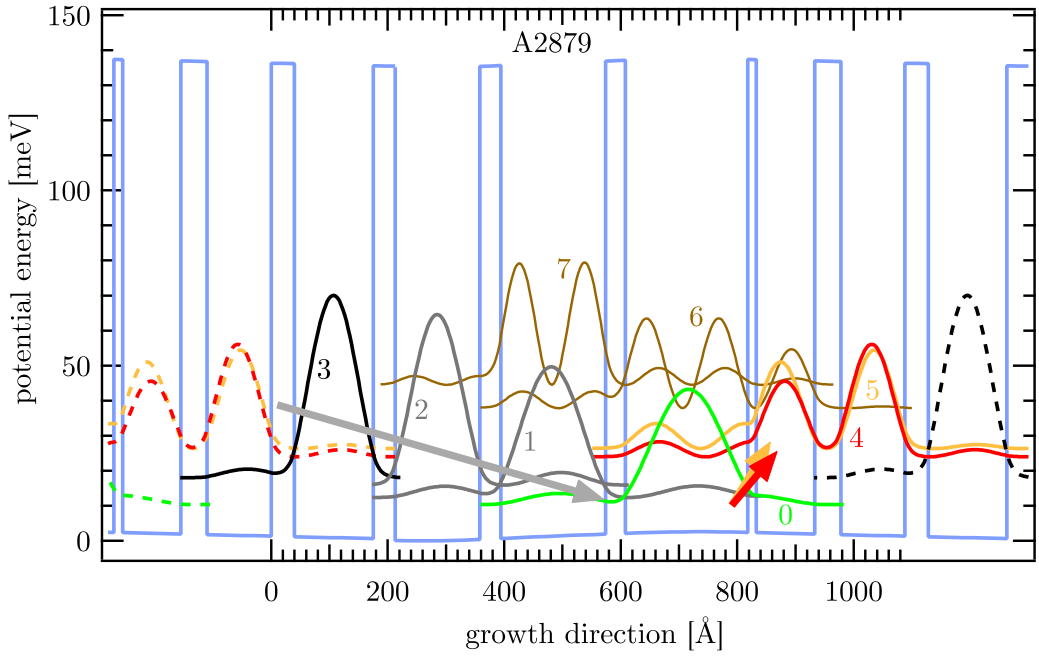


Figure 4.1: Self-consistently calculated conduction band structure of A2879 at 10 K. One full period, surrounded by adjacent parts of the previous and next period, is shown. The observed transitions take place between the ground state 0 (in green) and the upper states 4 and 5 (yellow, red) and have computed energies of 13.7 meV and 16.0 meV. The states 3,2,1 (grey tones) provide the relaxation channel for the electrons to the ground state of the next period.

Table 4.1: Parameters for the molecular beam epitaxial growth of sample A2879.

Material	thickness (Å)	Si-Doping (cm^{-3})	Rep.
GaAs	1000	n 5×10^{18}	
GaAs	200		
$Al_{0.15}Ga_{0.85}As$	45		
GaAs	100		60×
$Al_{0.15}Ga_{0.85}As$	15		↑
GaAs	210		·
$Al_{0.15}Ga_{0.85}As$	34		·
GaAs	180		·
$Al_{0.15}Ga_{0.85}As$	36		·
GaAs	145	n 6×10^{15}	·
$Al_{0.15}Ga_{0.85}As$	38		·
GaAs	135		·
$Al_{0.15}Ga_{0.85}As$	40		·
GaAs	110		↓
$Al_{0.15}Ga_{0.85}As$	45		60×
GaAs	200		
GaAs	6000	n 2×10^{18}	
GaAs	2500		
GaAs semi-insulating substrate			

4.2 Electrical transport

Figure 4.2 shows dark current measurements carried out at different temperatures. The region with a substantially reduced differential resistance around −0.65 V bias corresponds to the situation where the upper state of the transition (4, red coloured in figure 4.1) is resonantly aligned with the ground state of the next period ("lasing"-type of band structure). From the Arrhenius plot displaying logarithmic dark current at a constant voltage (50 mV) versus inverse temperature (inset of figure 4.2), an activation energy of 13.2 meV can be extracted in the "high temperature" range; this agrees well with the smallest calculated transition energy (0→ 4), 13.7 meV). For temperatures below 25 K however, a temperature independent component of the dark current gets dominant. As it has become more evident in the analysis of the electrical transport for the mid-IR QCDs (see section 5.2), one simple activation energy is not sufficient to describe the transport involving transitions between several states of the ladder. Indeed, the transport is activated with the "matching" energy only for temperatures above 25 K, where the device is barely usable as detector (sections 4.3.1, 4.3.2). In this temperature range, the mean thermal energy is already quite comparable to the optical transition energy (30 K corresponding to a thermal energy of 2.6 meV).

4.3 Optical characterisation

4.3.1 Spectral measurements

Spectral photocurrent measurements, as shown in figure 4.3, prove successful detection of THz radiation. The large peaks at 119 cm^{-1} (84 µm) and 135 cm^{-1} (74 µm) are at the expected energies (110 cm^{-1} and 129 cm^{-1} from simulation with *Sewself* {sewself}). In this measurement configuration, detection of the glowbar is possible up to a detector temperature of 30 K. Additional peaks at 219 cm^{-1} and 297 cm^{-1} are likewise identified as transitions into states 6 and 7 (brown in figure 4.1). With rising device temperature from 10 K to 30 K, the constant photocurrent signal gets more and more buried in the rising noise floor until it is completely concealed above 30 K.

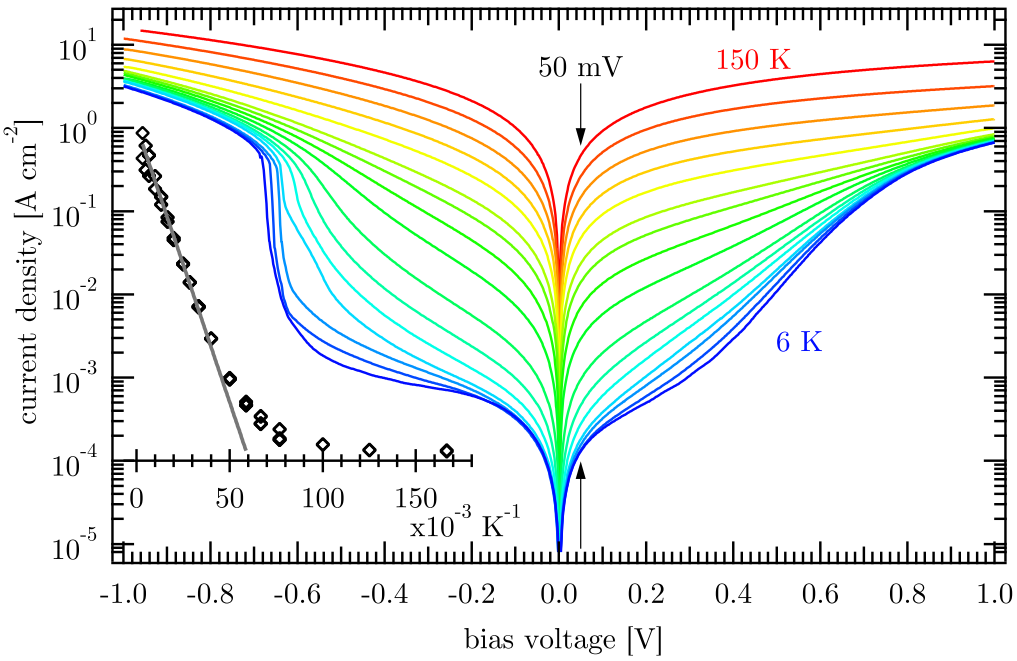


Figure 4.2: Measurements of $I - V$ characteristics at temperatures of 6, 8, 10, 13, 15, 17, 20, 25, 30, 35, 40, 50, 60, 75, 100, and 150 K in different colours. The inset shows an Arrhenius-style plot ($\diamond$) of current densities at 50 mV bias. The solid grey line is a fit resulting in an activation energy of 13.2 meV.

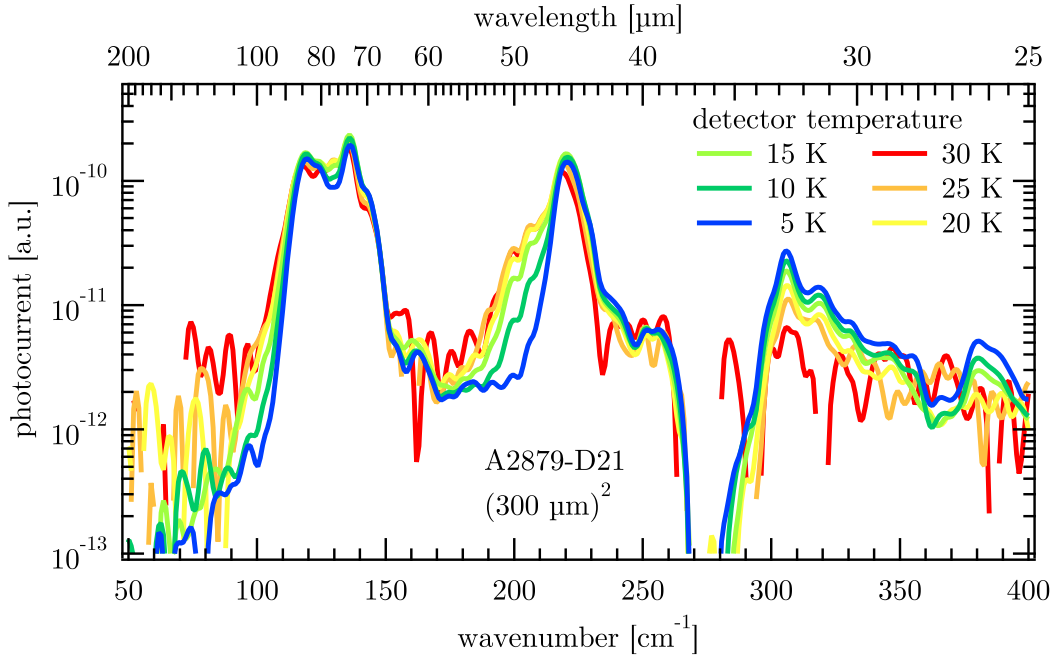


Figure 4.3: Photocurrent spectra of the device, illuminated with blackbody radiation from the internal glowbar of the FTIR. Different peaks are identifiable with calculated transition energies.

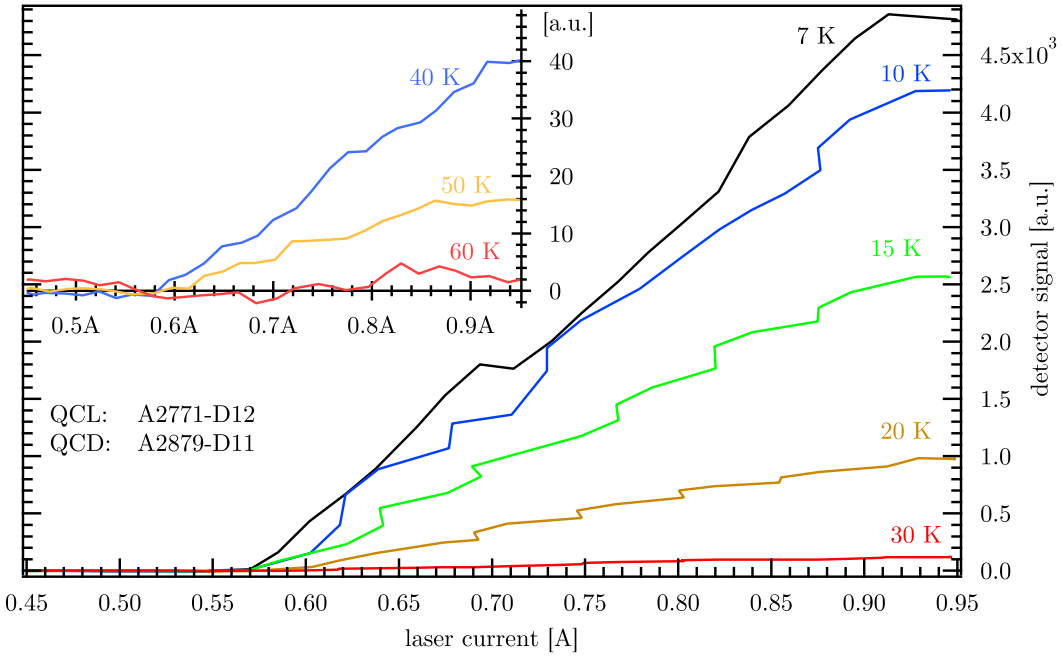


Figure 4.4: $L - I$ characterisation of a 87 µm quantum cascade laser, using the far infrared QCD A2879 as detector. Different detector temperatures are shown, the inset is a blow-up for the higher temperatures, with matching values of the arbitrary units between inset and main graph.

4.3.2 Monochromatic responsivity

Determination of the detector responsivity was then done using a bound-to-continuum QCL at 87 µm [53]. The light from the laser was picked up directly on the outside of the laser cryostat window and guided to the detector cryostat through a gold-coated lightpipe. For a total mean output power of 350 µW at this place (measured with an *Ophir Laserstar* ⟨laserstar⟩ powermeter), we first performed a lateral scan by moving the detector cryostat. Assuming circular symmetry and integrating the radial intensity profile over the entire aperture of the lightpipe allows to calculate the total power (in the center of the pipe) illuminating the sample surface. Together with the photocurrent from the device, which was measured with the current input of an *EG&G 7265* ⟨egg7265⟩ lock-in amplifier triggered by the laser pulser, a responsivity $\mathcal{R} = 8.6\,\mathrm{mA/W}$ at 10 K detector temperature was determined. To estimate the noise equivalent power (NEP), we performed an $L-I$ characterisation of the same laser, using our device as detector (shown in figure 4.4). Calibration was done again with the power-

meter, using exactly the same setup as before. At a NEP of 7 nW, the growing signal during switch-on of the laser rose above noise level, therefore a detectivity $\mathcal{D}^*$ of 5×10^7 Jones was determined. Compared to the performance (at 84 µm) of a heterojunction terahertz detector based on internal photoemission [54], responsivity and detectivity of the device are roughly two orders of magnitude lower. On the other hand, a 60 period photodetector with 100 % internal quantum efficiency would theoretically result in a responsivity of 1.15 A/W (one electron has to absorb 60 photons to pass the whole structure).

Based on dipole matrix elements from the numerical simulation, the device is expected to have an intersubband absorption of about $\eta_{\text{abs}} = 3\,\%$ for both relevant transitions (0→4,5). From the fact that both upper states are strongly coupled, which is confirmed by the observed anticrossing behaviour under applied bias voltage (figure 4.5), and due to the coherent tunneling of the electrons between these two states, we estimate an escape probability of $p_e \approx 0.5$[1]. Together with a capturing probability $p_c = 1$ due to the QCD design, one would expect a maximum responsivity $\mathcal{R}_i = (e/h\nu)\eta_{\text{abs}} p_e/(N p_c) = 18\,\text{mA/W}$ [7].

Despite the confirmation of the intended doping level using $C{-}V$ (capacitance-voltage) measurements of the structure, absorption was too weak to be reliably measurable ($< 2\,\%$), even in a multipass waveguide geometry. One possibility is that a substantial fraction of the n-type carriers is trapped by impurities below the ground state and therefore not available for absorption, as the overall doping concentration of $8 \times 10^{14}\,\text{cm}^{-3}$ is in the same order of magnitude as the background doping. Furthermore, the coupling through the top-illuminated diffraction grating might be quite inefficient. Yet, using a QCL as the light source enabled detection of radiation up to a detector temperature of 50 K; mainly thanks to the higher brightness of the laser compared to the glowbar.

4.4 Stark shift

As mentioned before, this detector was designed for operation under zero bias conditions. Accordingly, applying bias voltages did not improve its performance compared to the unbiased situation. Nevertheless, it allows to verify whether the device shows a Stark shift as expected from theoretical predictions. Figure

[1] An electron in one of the upper states will escape to next period with a probability of roughly 50 %.

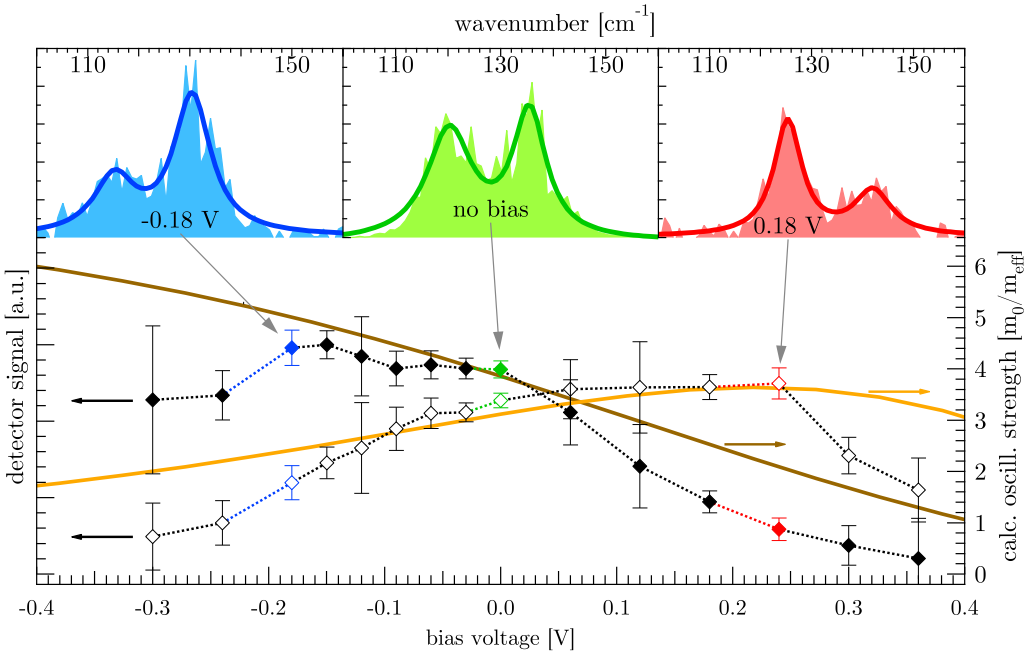


Figure 4.5: Bias voltage dependant ratio between the first two transitions ($0 \rightarrow 4$ and $0 \rightarrow 5$) . The solid curves were obtained from band structure calculations, the dashed curves are the result of spectral photocurrent measurements at different bias voltages. The insets show three of these measurements at $-0.18\,\mathrm{V}$, $0\,\mathrm{V}$, and $+0.18\,\mathrm{V}$ bias together with the fitted double peak Lorentzians.

4.5 shows calculated oscillator strengths for the main transitions ($0 \rightarrow 4$ and $0 \rightarrow 5$) as function of applied voltage (solid) and in comparison with measured values (dashed). The measured points were obtained by fitting the double peaks in the photocurrent spectra by two Lorentzians; three examples at different bias voltages are shown as insets. When the field changes from negative to positive values, the oscillator strength of the $0 \rightarrow 4$ transition increases at the cost of the decreasing $0 \rightarrow 5$ transition. As presented in figure 4.5, this Stark effect is clearly visible in both experiment and simulation.

5 Mid infrared QCDs

Based on the success of the far-infrared detector A2879, an adaptation to the mid-infrared wavelength range seemed to be an interesting topic. As the energy of the optical transitions is bigger than the energy of the LO-phonon, one could try to design the extraction cascade in a way that one could profit from the highly efficient electron-phonon scattering process.

5.1 Bandstructure design

Figure 5.1 shows the design chosen for such a detector targeting an optical transition energy of 235 meV (wavelength of 5.3 µm). Given incident light of the corresponding energy, electrons living in the ground state 0 can absorb the incident photon and be excited into one of the two upper states 7 and 8 (red and yellow states in figure 5.1). From there, they have basically two possibilities to "choose" from:

As before in the FIR detector, they either jump back to the ground state, which produces neither a net charge displacement nor any photocurrent.

Or they jump to the next state on the right (6) by emission of an LO-phonon, then to next one (5), emitting a LO-phonon again and so on until they reach the ground state of the following period. This second process leads to a displacement of these electrons along the growth directon (lateral in figure 5.1), which can finally be measured as current flowing between the two terminals of such a device. As all states of the extraction cascade are arranged with an energy difference of 34 meV, the energy of the LO-phonon in InP, the total extraction should be efficient (fast) and potentially result in a high escape probability p_e.

A similar design for a detection wavelength of 9.1 µm (135 meV) is shown in figure 5.2. The working principle is the same as for the 5.3 µm device. The extraction cascade consists of fewer states to allow again a next-neighbour energy

difference equal to that of the LO-phonon. These two structures were grown on semi-insulating InP substrates in the MBE in Neuchâtel with layer structures described in table 5.1. On top of a 6000 Å thick n doped $In_{0.533}Ga_{0.467}As$ contact layer, 50 respectively 30 repetitions of the detector period were grown, "closed" by a single 20 Å InAlAs barrier. On top of this active region, an InGaAs contact layer was grown. The 3D-doping level n_{3D} of the contacts was adjusted to the value for which the resulting Fermi level $E_{F,c}$ equals the quasi-Fermi level $E_{F,w}$ in the ground state well, defined by its 2D-doping density n_s:

$$\begin{gathered} E_{F,c} \stackrel{!}{=} E_{F,w} \\ n_{3D} = \frac{1}{3\pi^2}\sqrt{\left(\frac{2m_e^*\left(E_{F,c}-E_C\right)}{\hbar^2}\right)^3}\,,\; n_s = \frac{m^*}{\pi\hbar^2}\left(E_{F,w}-E_0\right) \end{gathered} \tag{5.1}$$

To provide an enhanced ohmic contact, this was finally overgrown with an alloyed contact (see table 5.2) consisting of an InAs-InGaAs chirped superlattice with continuously rising doping density. To prevent structure cracking due to lattice mismatch because of increasing In content, the substrate temperature during this last growth step is continuously decreased.

The overall composition and the period of the active region superlattice were confirmed by X-ray diffraction. The measured superlattice period deviated from design by $\approx 3\,\%$ for N538 and $\approx 0.7\,\%$ for N516 (table A.1). The samples were then processed to mesas of three different sizes (100 µm, 200 µm, and 300 µm) as described in section 3.1. The metallic contact layer was as follows: Ti:1.5 nm, Au:12 nm, Ge:27 nm, Ag:50 nm, Au:350 nm.

5.2 Electrical transport

For both structures, N516 and N538, and different mesa sizes, $I-V$-curves have been taken at different temperatures between 10 K and room temperature. On one hand, these were measured under illumination of the background from the laboratory environment, on the other hand under dark conditions. Therefore, the openings in the cryo-shield for optical access were carefully closed with aluminium foil to provide an optical background illumination as low as possible.

This data allows to extract the differential resistances R_0 or conductances

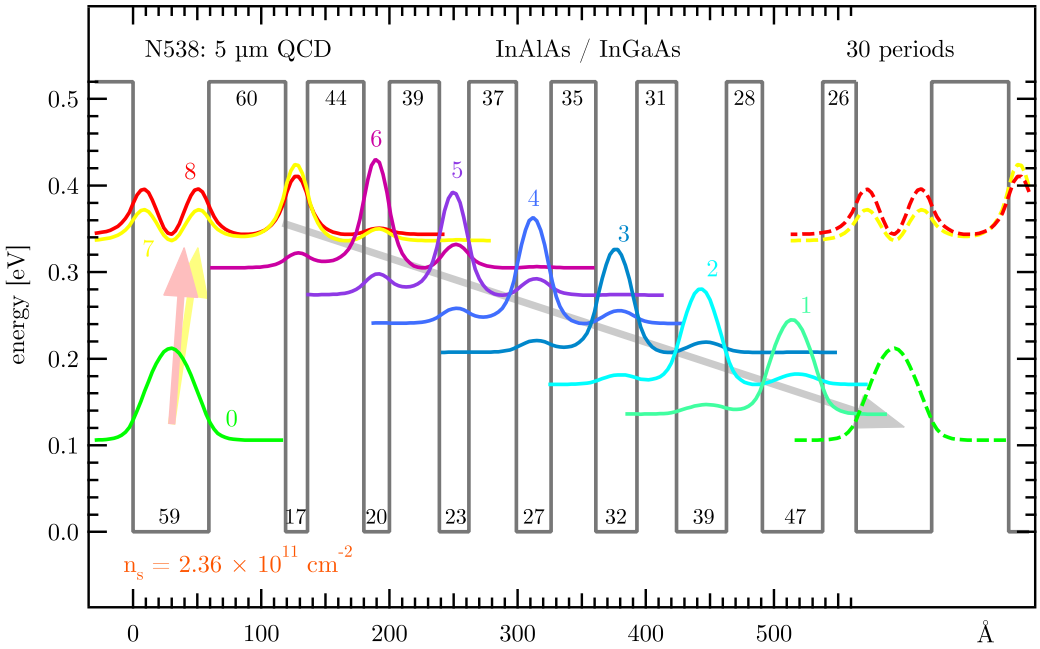


Figure 5.1: Designed conduction band offset structure for N538, a QCD targeting 5.3 µm. The modulated potential built up by the different conduction band edge of the alternating material layers - lattice matched InGaAs and InAlAs on InP - as function of growth direction in grey. The coloured curves show the moduli squared of the relevant wavefunctions. The arrows show the transport of the electrons through the structure.

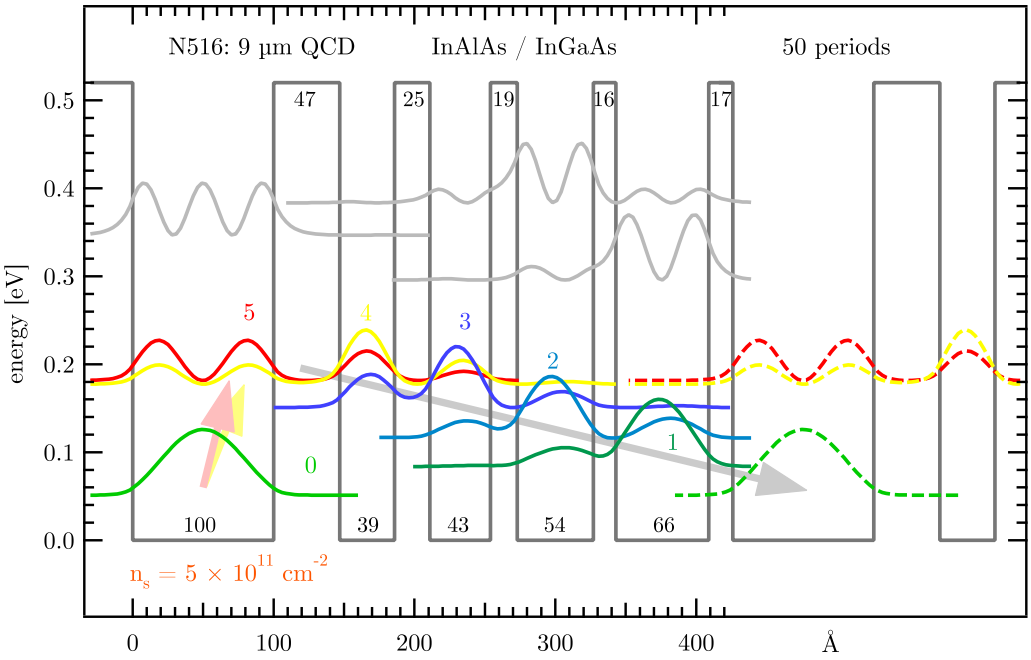


Figure 5.2: Designed conduction band offset structure for N516, a QCD targeting 9 µm. The modulated potential built up by the different conduction band edge of the alternating material layers - lattice matched InGaAs and InAlAs on InP - as function of growth direction in grey. The coloured curves show the moduli squared of the relevant wavefunctions. The arrows show the transport of the electrons through the structure.

Table 5.1: Growth parameters for N516 and N538

N516			N538		
Material thickness (Å)	Si-doping (cm^{-3})	Rep.	Material thickness (Å)	Si-doping (cm^{-3})	Rep.
Alloyed contact, see table 5.2			*Alloyed contact, see table 5.2*		
InGaAs 2500	n 2.0×10^{18}		InGaAs 2000	n 1.6×10^{18}	
InAlAs 20			InAlAs 20		
InGaAs 100	n 5.0×10^{17}	50×	InGaAs 59	n 4.0×10^{17}	30×
InAlAs 47		↑	InAlAs 60		↑
InGaAs 39		·	InGaAs 17		·
InAlAs 25		·	InAlAs 44		·
InGaAs 43		·	InGaAs 20		·
InAlAs 19		·	InAlAs 39		·
InGaAs 54		·	InGaAs 23		·
InAlAs 16		·	InAlAs 37		·
InGaAs 66		↓	InGaAs 27		·
InAlAs 17		50×	InAlAs 35		·
			InGaAs 32		·
			InAlAs 31		·
			InGaAs 39		·
			InAlAs 28		·
			InGaAs 47		↓
			InAlAs 26		30×
InGaAs 6000	n 1.0×10^{18}		InGaAs 6000	n 1.6×10^{18}	
InP:Fe semi-insulating substrate			InP:Fe semi-insulating substrate		

Table 5.2: Standard alloyed contact grown over the top contact of the MIR QCDs. The doping concentration continuously raised from the concentration of the latest doped layer, normally the top contact layer, to $2 \times 10^{19}\,\mathrm{cm}^{-3}$. In parallel, the overall In-concentration is raised together with a decreasing substrate temperature to prevent cracking due to lattice mismatch induced by the former.

Material	thickness (Å)	Si-doping	(cm^{-3})
InGaAs	9	n	2×10^{19}
InAs	3	n	↑
InGaAs	9	n	·
InAs	3	n	·
InGaAs	12	n	↑
InAs	3	n	·
InGaAs	15	n	·
InAs	3	n	↑
InGaAs	18	n	·
InAs	3	n	·
InGaAs	21	n	↑
InAs	3	n	·
InGaAs	24	n	·
InAs	3	n	↑
InGaAs	27	n	·
InAs	3	n	·
InGaAs	27	n	↑
InAs	3	n	x

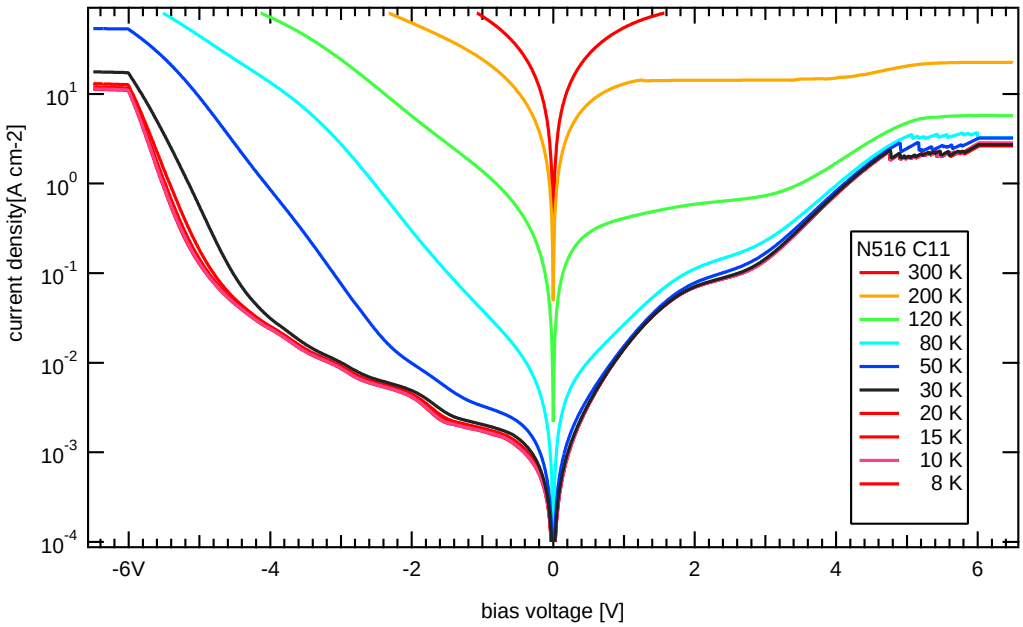


Figure 5.3: $I-V$ curves for N516 (9 µm) at different device temperatures between 8 K and 300 K. The kinks correspond to special alignment conditions between states of the extraction cascade and the ground state.

$G_{\mathrm{det}} = \frac{1}{R_0}$ of the devices[1], which are important for several purposes: First, they are needed for the transformation of the voltage-spectra measured using a voltage pre-amplifier to photocurrent spectra as described in section 3.3.2 (equation 3.2). Second, $R_0(T)A$ with A being the detector's surface is a figure of merit for QCDs[2], as they are Johnson-noise limited above T_{BLIP} (equations 2.31,2.37).

To provide finer data over the temperature range, additional $I-V$ measurements under background illumination as well as under dark conditions have been performed. A *LabVIEW* {labview} program continuously recorded $I-V$ curves and device temperature while the temperature of the cryostat cold finger was slowly increased by the temperature controller. This data allows to average several $I-V$-curves and therefore reduce the noise at low temperatures, where for the sample N538 the currents were at the inferior measurement limit of the used *Keithley SourceMeter 2410* ⟨sm2410⟩.

Figures 5.3 and 5.4 show a selection of $I-V$ curves for different temperatures of the devices N516 and N538 respectively.

[1] By doing a linear fit in the vicinity of zero bias voltage
[2] and other types of photovoltaic detectors [36, 55–57]

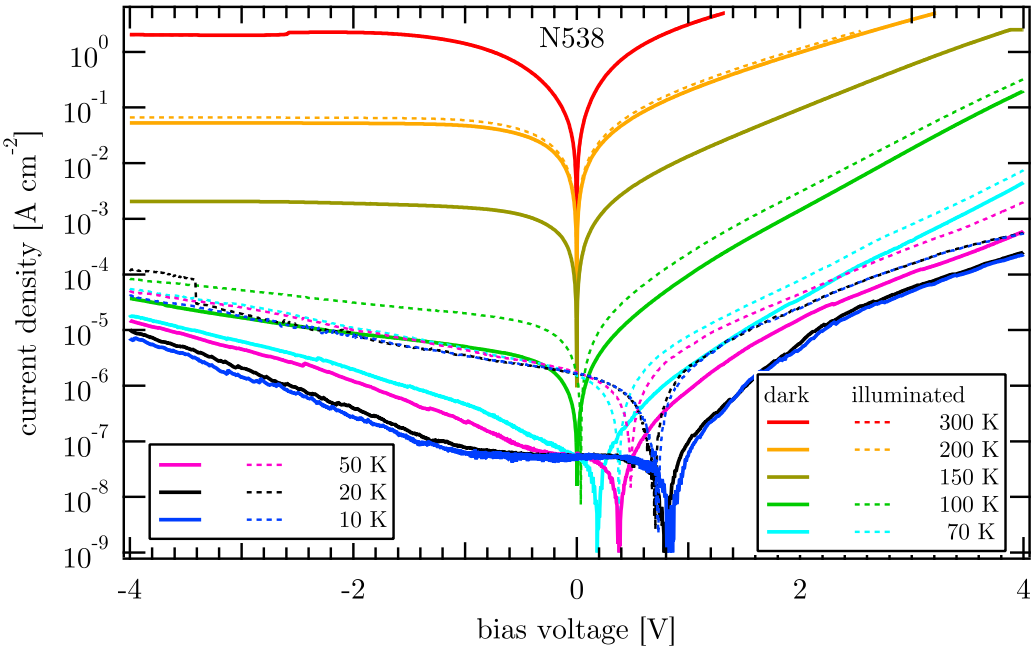


Figure 5.4: $I-V$ of N538

The kinks seen in N516's $I-V$ curves for both positive and negative voltages at lower device temperatures are due to special alignment configurations in the band structure: Since each period contains a quantum stair, there are always certain external fields for which the ground state of one period's stair is in resonance with the first or a higher excited state of the next period's stair. Under these conditions, transport efficiency is "remarkably" better, which leads to increased conductivity.

Plotting the differential device resistivity logarithmically as function of inverse device temperature, as shown in figure 5.5, is equivalent to usual Arrhenius plots[3]. Evidently it is not possible to find exactly one activation energy from these data. At first glance, this is rather unsatisfactory, as I expected to find an activation energy which corresponds to the optical transition energy. But on a closer look, it becomes clear that a "two state" model is way too simple to explain electrical transport, as there is a series of states at energies situated between the lower and upper states of the optical transition which also contribute to electrical

[3] As generally the logarithm of the *activated* quantity (the electrical current) is plotted, the graph is just "upside down", but additionally shows the Johnson noise-limited figure of merit R_0A (equation 2.31).

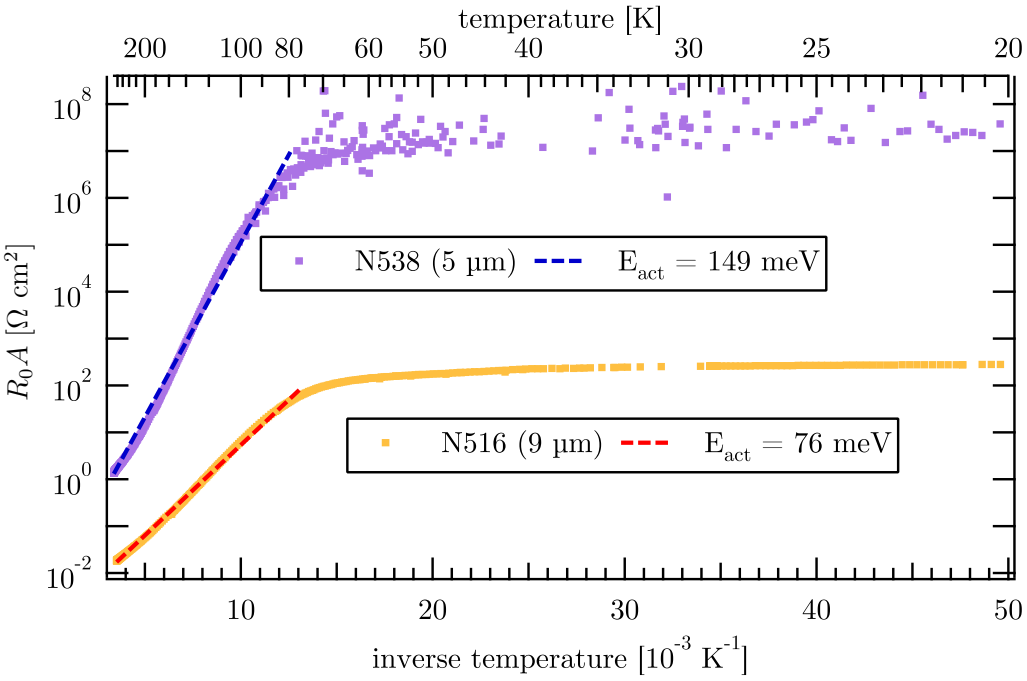


Figure 5.5: R_0A plot for N516 and N538 as function of inverse temperature: "Arrhenius"

transport.

Following the model described by Koeniguer in [52] for a very similar QCD structure in AlGaAs/GaAs, it is possible to better understand the behaviour of the differential resistance at zero bias as function of device temperature. The model distinguishes between two classes of transitions: those taking place between states inside the same period (cascade) and those connecting states of adjacent periods of a repeated quantum cascade structure. The assumption that the former (intracascade transitions) are much stronger than the latter (intercascade transitions) then allows for the following description. Each period is considered to stay at thermodynamical equilibrium internally, but not necessarily to share the chemical potential with adjacent periods. Introducing a bias voltage between two periods as a perturbation and expressing the current due to that voltage leads to the main result of that model: the electrical resistance as function of intercascade transition rates, evaluated at thermal equilibrium.

$$R_0A = \frac{k_BT}{e^2\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}}G_{ij}} \quad \text{(per period)} \tag{5.2}$$

Here, $G_{ij} = G^a_{ij} + G^e_{ij}$ denotes the total transition rate from state i in period $\mathcal{A}$ to state j in the next period $\mathcal{B}$, covering both the absorption and emission of LO-phonons. $G^{a,e}_{ij}$ are obtained based on the (energy dependant) transition rates $S^{a,e}_{ij}(E)$ derived by Ferreira and Bastard [58][4] like

$$G^a_{ij} = \int_{\max(E_i, E_j - \hbar\omega_{\mathrm{LO}})}^{\infty} S^a_{ij}(E) f(E)[1 - f(E + \hbar\omega_{\mathrm{LO}})] n_{\mathrm{LO}} D(E) dE \tag{5.3}$$

and

$$G^e_{ij} = \int_{\max(E_i, E_j + \hbar\omega_{\mathrm{LO}})}^{\infty} S^e_{ij}(E) f(E)[1 - f(E - \hbar\omega_{\mathrm{LO}})][1 + n_{\mathrm{LO}}] D(E) dE, \tag{5.4}$$

taking into account the thermal phonon population n_{LO}, the thermal filling of the electronic states by the Fermi-Dirac factor f as well as the 2D density of states $D(E)$ and the (minimum) energies of initial and final states E_i and E_j.

$$n_{\mathrm{LO}} = \frac{1}{e^{\frac{\hbar\omega_{\mathrm{LO}}}{k_B T}} - 1}, \quad f(E) = \frac{1}{e^{\frac{E - E_f}{k_B T}} + 1}, \quad D(E) = \frac{m^*}{\pi\hbar^2}$$

Figure 5.6 shows the comparison of this calculaton[5] with the measured data for the both structures. The R_0A calculation as described above is based on the wavefunctions and energies as calculated by *sewself* {sewself}. For the structure N538 it is not possible to calculate the wavefunctions for two full periods, as *sewself* unfortunately shows numerical unstabilities above a certain structure thickness.[6] Thus, only transitions between the two lowermost states in period $\mathcal{A}$ and the full set of states in period $\mathcal{B}$ are contained in the simulation of N538. Similar calculations for "rotated" layer sequences[7] have been done and show indeed, that the set chosen contains the dominant transitions.

5.3 Optical characterisation

Like described in sections 3.1 and 3.3.1, the samples N516 and N538 were polished to 45° facets and then mounted on the cold finger of the cryostat for the

[4] $S^{a,e}_{ij}(E)$ is named $\frac{1}{\tau_i}$ there.
[5] Implemented as shown in appendix C.2
[6] Approximately 10 quantum wells in the case of N538.
[7] Three lowermost states in $\mathcal{A}$ together with 8 uppermost states in $\mathcal{B}$, 4 lowermost in $\mathcal{A}$ and 7 uppermost in $\mathcal{B}$, ...

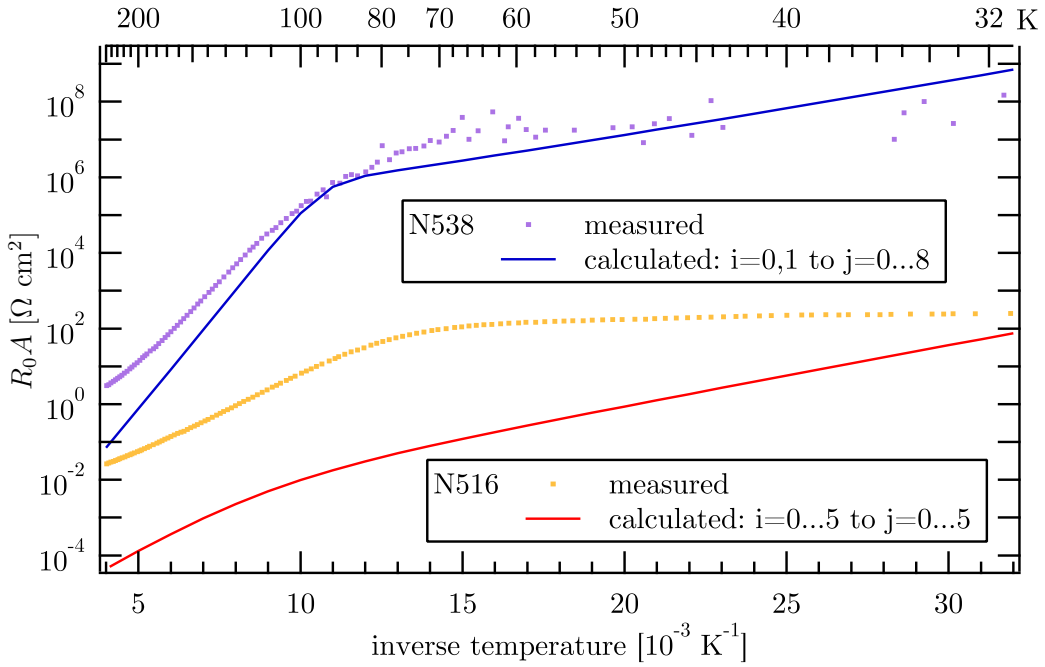


Figure 5.6: Transport simulation as described in equations 5.2-5.4 together with experimental data for N516 and N538.

optical measurements. For several temperatures from 10 K to room temperature, photocurrent spectra were determined following the description in section 3.3.2 by applying the transimpedance correction as in equation 3.2 to measured photovoltage spectra. Later, after we had acquired the transimpedance amplifier ⟨sr570⟩, these measurements were repeated, directly measuring the photocurrent for the whole temperature range in case of the 5.3 µm detector N538 and at some selected temperatures for the 9 µm device N516.

Figures 5.7 and 5.8 show responsivity spectra obtained from these photocurrent spectra by dividing them by the effective illumination spectrum in the spectrometer as obtained by the reference measurement (shown in figures 3.7 and 3.8 in section 3.3.2), including the transmittance of the ZnSe cryostat window and the transmission of the vacuum-semiconductor interface as well as a factor of $\sqrt{\frac{1}{2}}$ due to the effective surface shortening in the case of a 45° sample. Additionally, measured absorption (measured as described in section 3.3.1) at room temperature is added to these figures as "hanging" from the top axis. For both devices, there is a good match between the (narrow band) detection current and the absorption peak on one hand and the designed transition energy on the other hand.

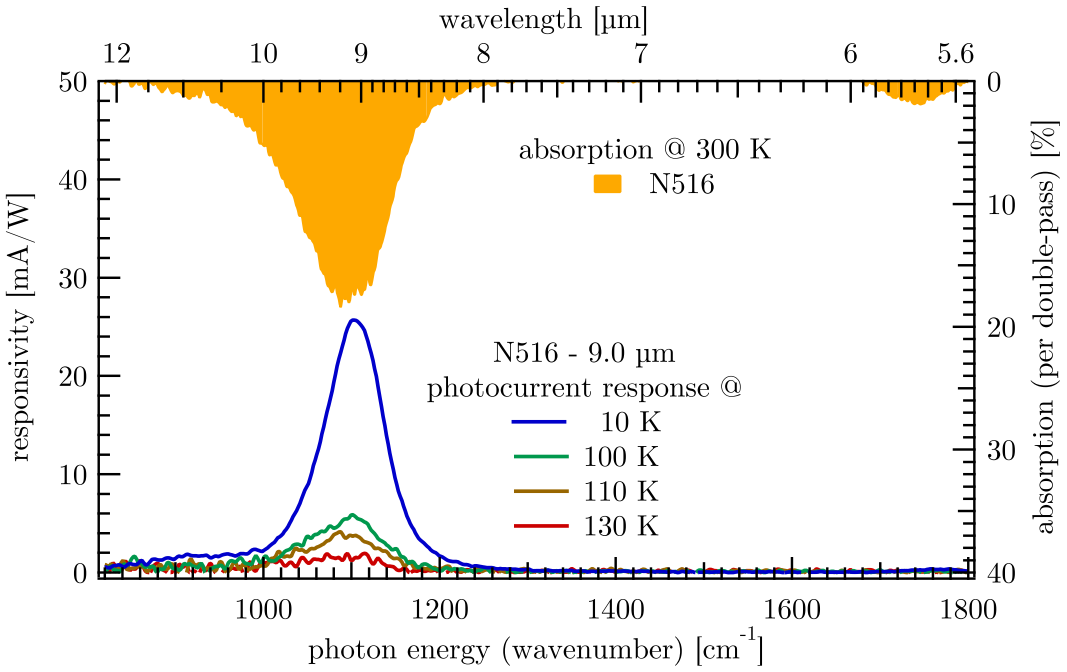


Figure 5.7: Responsivity spectra of N516

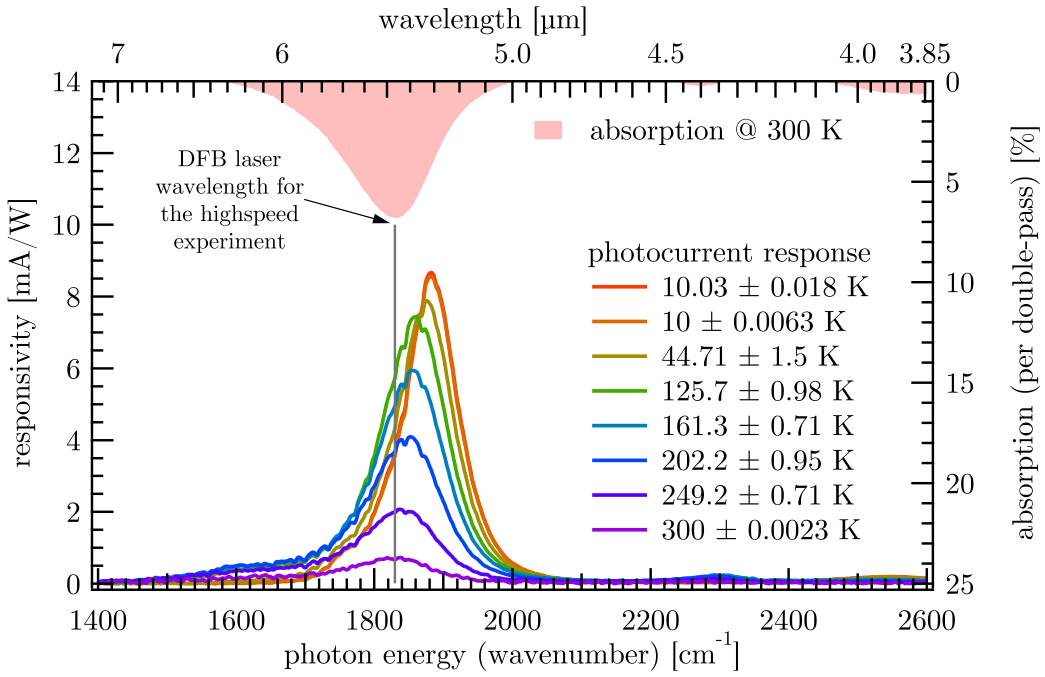


Figure 5.8: Responsivity spectra of N538

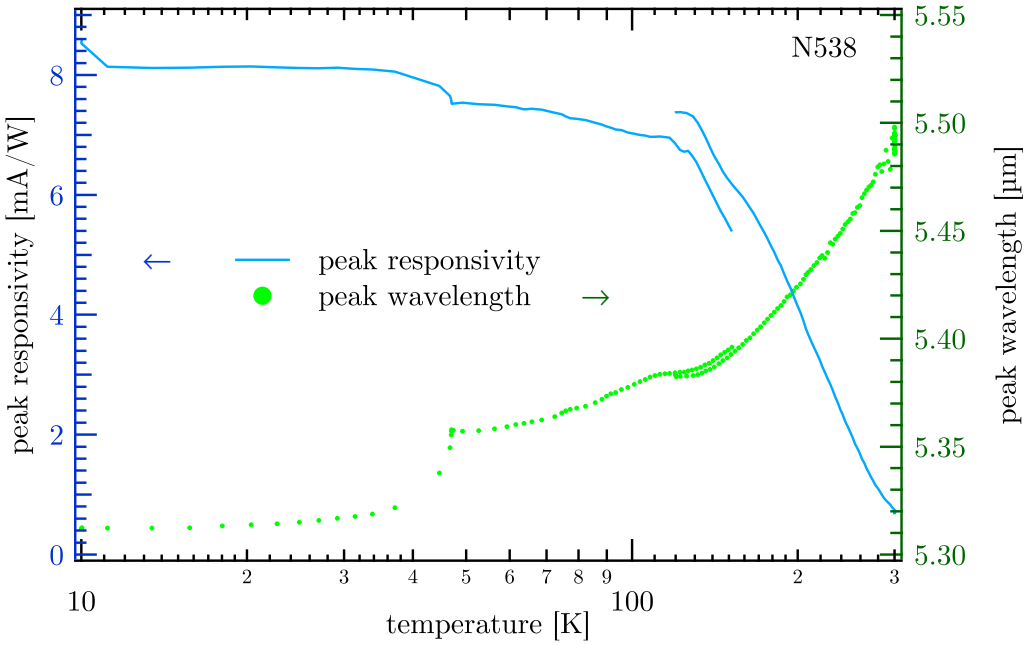


Figure 5.9: Peak responsivity $\mathcal{R}$ of N538 as function of device temperature (red curve on left axis) together with the corresponding position of the peak (orange dots on right axis).

The photocurrent peak for the device N538 has a FWHM (full width at half maximum) varying between 11 meV at a device temperature of 10K and 18 meV at room temperature. Given the central transition energy of 232 meV (225 meV at room temperature), these correspond to relative linewidths of 4.8 % up to 8.1 % over the measured temperature range. For the device at 9 µm (N516), the peak widths lie between 9.9 meV and 13.6 meV (10 K and 130 K) or relative linewidths of 7.3 % to 10.0 %, as the transition energy in that case peaks at 136 meV.

Plotting the peak responsivities and their corresponding central transition energies results in figure 5.9. These values were obtained by numerically fitting a Lorentzian to all photocurrent spectra between 10 K and room temperature and reporting the corresponding fit coefficients as function of device temperatures. The overlapping (but differing) data in the range of 120 K - 150 K is the result of a mechanical measurement difficulty: As the cold finger of the cryostat gets warmer, it also gets longer, which results in an optical misalignment when the detector mesa moves out of the illumination spot (given as the image of the spectrometers aperture). These measurements were done semi-automatically contin-

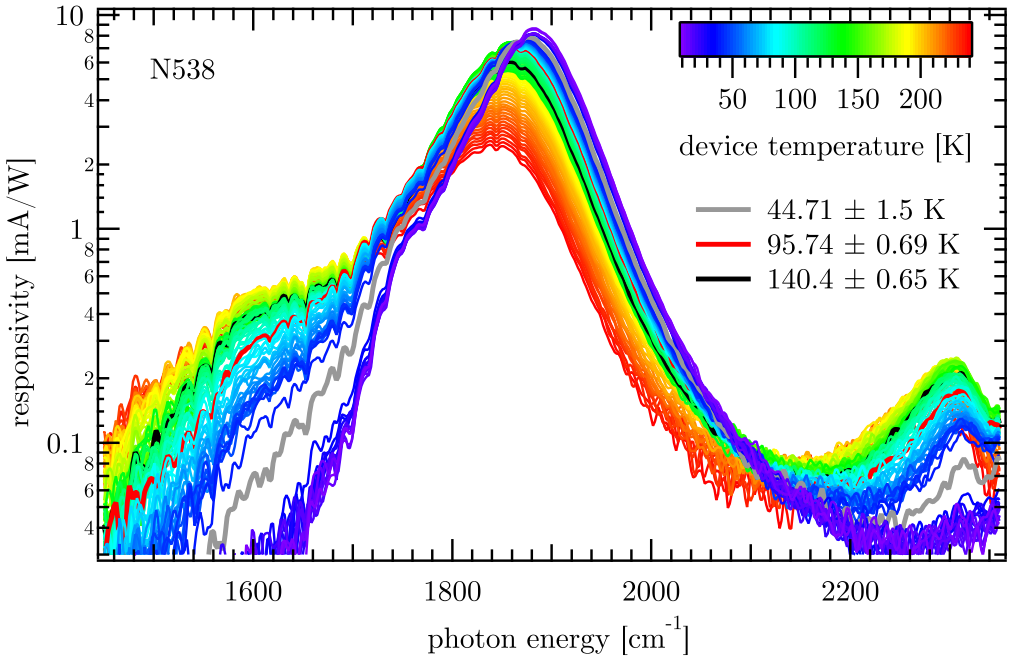


Figure 5.10: Detailed temperature scan of N538 responsivities

uously recording spectra after spectra while the device temperature was slowly driven upwards under the control of the temperature controller[8]. When the misalignment became obvious (the signal interferogram significantly dropped), the measurement had to be interrupted and restarted[9] at a reasonable, lower temperature.

There is a kink at 45 K device temperature in the responsivity for the sample N538 (figure 5.9). At this temperature, the "preferred" transition switches from the higher energy one ($0 \rightarrow 8$) to the lower energy one ($0 \rightarrow 7$), which is also clearly recognised in the wavelength hop at the same temperature (green dots). Figure 5.10 shows the whole series of responsivity spectra between 10 K and 240 K, logarithmically scaled. The highlighted spectra at 45 K nicely shows this switch in wavelength. Additionally, at higher temperatures an increase of the responsivity at 1600 cm^{-1} becomes significant. This is explained by the thermally activated carrier concentration in state 1, which leads to absorptions $1 \rightarrow 7, 8$ with the corresponding lower energy.

[8] as well as occasionally manual adjustments of the liquid helium flow rate
[9] after stabilising the temperature at the restart value and adjusting the optical alignment by maximising the signal

Peak responsivities at low temperatures are $\mathcal{R} = 26\,\mathrm{mA/W}$ for the 9 µm detector and $\mathcal{R} = 8.4\,\mathrm{mA/W}$ in the case of the the 5.3 µm device, which correspond to overall quantum efficiencies of $\eta_{\mathrm{tot}} = 3.6 \times 10^{-3}$ and $\eta_{\mathrm{tot}} = 1.9 \times 10^{-3}$, respectively. As the maximal achievable values of η_{tot} in a QCD includes the factor $1/N$ due to the capture probability of p_c=1, with N being the number of periods, the former is

$$\eta_{\mathrm{tot}} = \eta_{\mathrm{abs}} p_e \frac{1}{N} \tag{5.5}$$

Comparing the measured values of η_{tot} with the maximal achievable quantum efficiency η_{tot} of 3.8×10^{-3} (2.33×10^{-3}) for the the 9 µm (5.3 µm) device based on the measured absorption of 19 % (7 %) and 50 (30) periods, escape probabilities p_e of 0.94 and 0.82 can be determined. This is somewhat more than the expected 0.5 due to the following rough estimation: Since the upper state is a duplet, there is only about a 50 % chance for the excited electrons to escape from the main well and to contribute to the current transport.

Nevertheless, the values agree reasonable well, as this calculation compared quantum efficiency at low temperature ($< 100\,\mathrm{K}$) with absorption data measured at room temperature. The thermal occupation of higher states reduces the electrons available for absorption with increasing temperature, so it is safe to assume that the effective absorption at 10 K is bigger than at room temperature.

The most prominent observation of figures 5.9 and 5.10 is certainly that the responsivity, which is supposed to be temperature independent, drops significantly from an almost constant value of $\approx 8\,\mathrm{mA/W}$ below 130 K to 0.7 mA/W at 300 K. This behaviour has several possible reasons: First, the already mentioned thermal backfilling of the first step level (only 32 meV above the ground state) reduces the charges "available" in the ground state, which results in a smaller absorption from the ground state; second, the escape probability p_e will be reduced due to a phonon scattering induced shortening of the upper state relaxation time; and third, the smaller device resistance at elevated temperatures (see figure 5.5) severely hampers unidirectional transport of electrons. Using the results obtained by calculating the band structure of N538 with *sewself* {sewself} for different temperatures, in particular the occupation density of the ground state, the absorption rates from the ground state to the upper states of the optical transition, and the transition rates between all the states, I tried to find the temperature dependence of the escape probability. The latter could

explain the drop in responsivity, using numerical calculations described below[10]. Starting from one of the upper states, all paths leading either to the ground state of the next period (escape) or back to the original ground state (fallback) are identified. For each of these paths going through nodes (states) $s_1, s_2, \ldots, s_n$, its probability p_{path} to be traversed by an electron is the product of the probabilities of each involved transition

$$p_{\text{path}} = \prod_{i=1}^{n-1} p_{s_i \to s_{i+1}} \tag{5.6}$$

where $p_{s_i \to s_{i+1}} = r_{s_i \to s_{i+1}} / \sum_k r_{s_i \to k}$ is the probability of exiting from state s_i to state s_{i+1} based on all transition rates $r_{a \to b}$ from the band structure calculation. Summing the probabilities of all paths leading to the next ground state results in the simulated escape probability

$$p_e = \sum_l p_{\text{path}_l} \tag{5.7}$$

Additionally, summing up the probabilities of all paths going back to the original ground state defines the probability to stay, which must be equal to $1 - p_e$. This allows to numerically verify that in the rather big sum of products along paths of different lengths no transition was forgotten or counted double.

This calculation has to be done for both upper states, multiplied by the corresponding absorption probability $\alpha_{0 \to u_1}$ and by the occupation density of the ground state ρ_0. One then obtains a quantity[11] which - as function of temperature - should be proportional to the quantum efficiency, or for an almost constant wavelength also proportional to the measured peak responsivity.

$$\rho_0 \left(\alpha_{0 \to u} \times p_{e,0 \to u} + \alpha_{0 \to d} \times p_{e,0 \to d} \right) \propto \mathcal{R}_{\text{peak}} \tag{5.8}$$

Figure 5.11 shows the values for the quantum efficiency obtained by these

[10] These calculations were implemented in a small program package called "sewcalc", which wraps around a slightly modified version of *sewself*, runs it for a series of temperatures, let it calculate the relevant data, parses its output, allows to store all these values in an intermediate file, does all the calculations described here depending on a variety of commandline options, and finally allows to write these out to text files, formatted as "Igor Text Files", which allows fast and convenient loading into *Igor Pro* {igor}. The sources thereof are printed in appendix C.3.

[11] all these values are taken from the simulation done with *sewself*

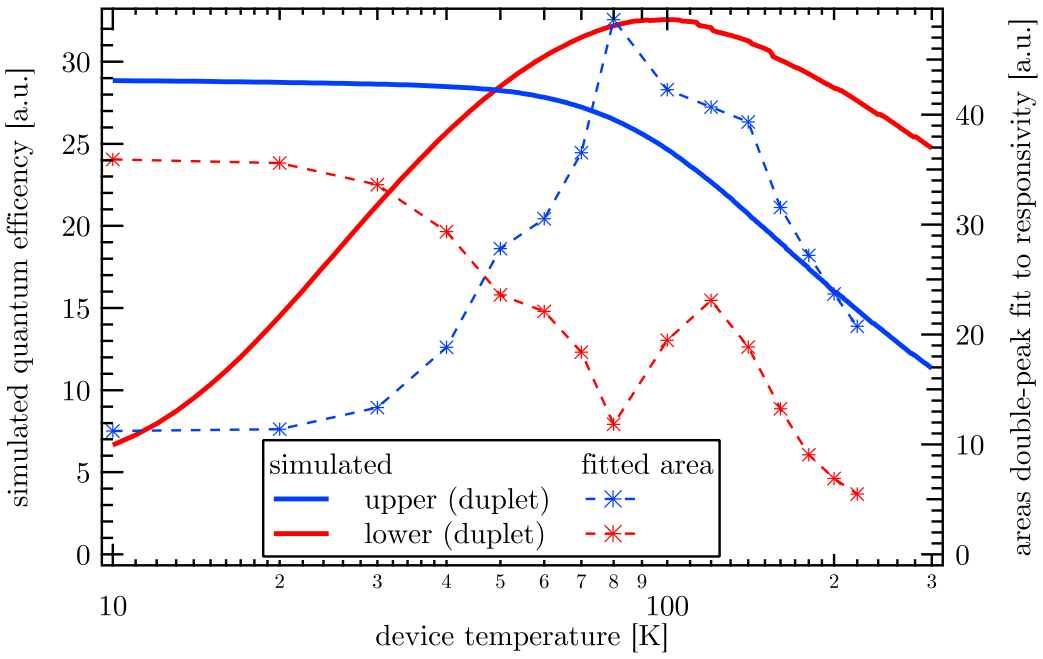


Figure 5.11: Responsivity as function of device temperature, independently plotted for both states of the upper duplet. The star-shaped markers show the values obtained by fitting a double Lorentzian to the measured responsivity spectra, the solid lines show the result of the simulation (equations 5.6-5.8 / appendix C.3).

calculations, seperately plotted for both states of the excited duplet of the optical transition. By only looking at the upper values from the upper state, the measured drop in responsivity (see figure 5.9) at roughly around 120 K could be explained by the simulation, although it does not completely explain the strong decrease for temperatures over 250 K. Additionally, the process via the lower state takes over at 50 K and should keep the responsivity almost constant up to room temperature, which is not at all the case in the measured data, where the responsivity drops by more than a factor of 10 between 10 K and 300 K. Therefore, I believe that the third reason mentioned above, the reduced electrical resistance for elevated temperatures cancelling out the directed transport of electrons across the structure, is the dominant reason for the loss in responsivity at rising temperature.

The data from these simulations does, on the other hand, rather well predict the temperature at which the duplet states change their dominant role, as can clearly be seen in figure 5.11. As the higher-energy transition is dominant below 50 K in the simulation and the exactly opposite is observed in the measurement where the lower-energy transition is dominant for lower temperatures, the perfect agreement on the crossing temperature might probably be just a "lucky" coincidence.

For both detectors, Johnson noise limited detectivity $\mathcal{D}^*_J$ and the background photon noise detectivity $\mathcal{D}^*_{\mathrm{BLIP}}$ as well as the detectivity due to both noise sources are shown as function of device temperature in figure 5.12. These were calculated based on the measured responsivity data and the differential device resistance at zero bias, extracted from the electrical measurements (see section 5.2) and under the assumption of a full cone field of view (2π opening angle) illumination with a 300 K blackbody. As described in section 2.2.2, the temperature at which $\mathcal{D}^*_J$ and $\mathcal{D}^*_{\mathrm{BLIP}}$ intersect defines the BLIP temperature T_{BLIP}, below which the detector's performance is limited by background photon noise and where one does not gain anymore by further reducing the device temperature.

A background limited detectivity of 8×10^{10} Jones below and 5.4×10^{10} Jones exactly at $T_{\mathrm{BLIP}} = 80\,\mathrm{K}$ is achieved for the 5.3 µm detector N538. For the 9 µm device N516, the Johnson noise dominates over the whole temperature range and no BLIP operation can be achieved. Although the decrease in detectivity is relatively moderate up to 65 K with $\mathcal{D}^*_{\mathrm{tot}} = \mathcal{D}^*_J = 0.7 \times 10^{10}$ Jones, it drops significantly above this temperature.

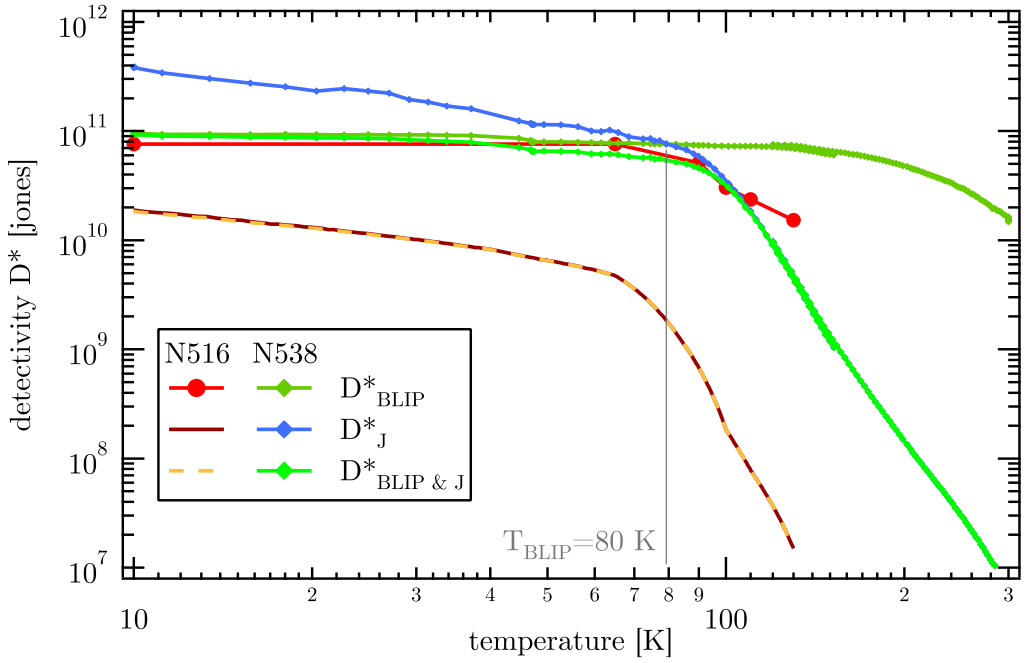


Figure 5.12: Detectivity plot N516, N538

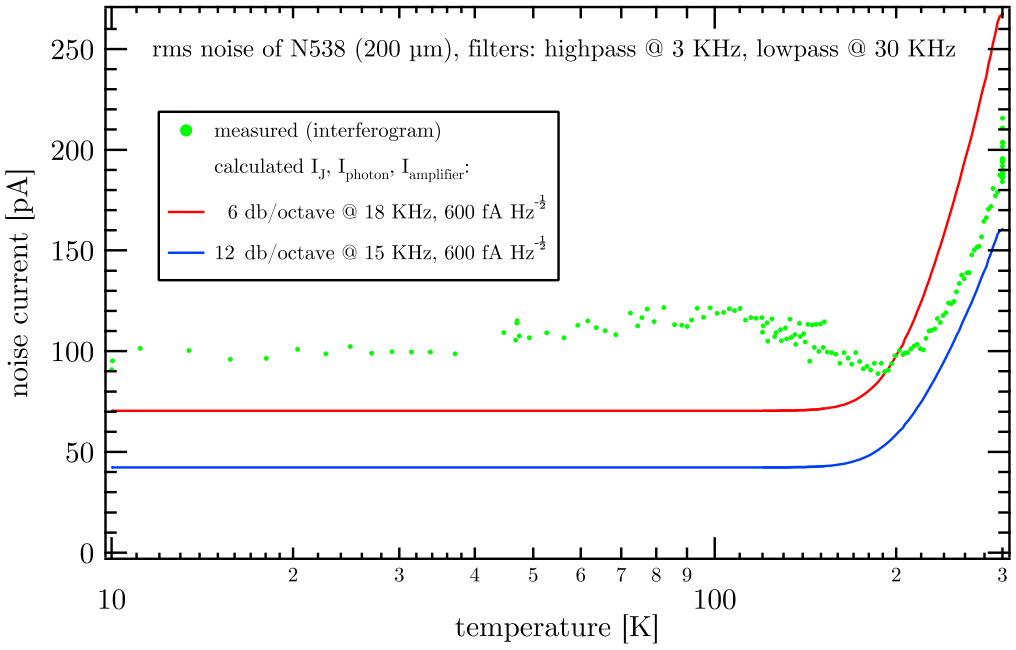


Figure 5.13: Noise comparison N538

The analysis of the RMS noise on the sides of the interferograms[12] allows to directly verify the noise, which is shown in figure 5.13 for the device N538. As function of device temperature, the noise measured directly from the interferogram for the given bandwidth is compared with the expected noise based on

- Johnson noise due to the electrical resistance of the device,
- photonic noise due to a 300 K blackbody environment (as in the detectivity calculations),
- additional photonic noise due to the average illumination in the spectrometer (half the intensity as given in figure 3.8),
- electrical noise added by the transimpedance amplifier at the amplification factor used.

All these four noise contributions are added spectrally, taking into account the spectral photon flux density of the light sources as well as the spectral quantum efficiency of the detector for both photonic noises. This noise spectrum is then transformed to a temporal spectra with the well known relation 3.1, multiplied by the response function of a series of electrical high- and low-pass filters modelling the used filter settings as well as the intrinsic cut-off behaviour of the amplifier. Finally, the result of the preceding calculation is (numerically) integrated. The numerical details are found in appendix C.1. The comparison between the measured noise and the calculated values show quite a good agreement, considering a number of uncertainties like using simple first order $R-C$ terms as approximation of the filters, the assumption of a flat amplifier noise spectrum[13], and most significantly, the approximation of the intrinsic cut-off of the amplifier[14].

Apparently, the amplifier noise dominates up to a device temperature of 200 K, which is the drawback of the usage of an otherwise very convenient multi-purpose amplifier with a large gain range between 1×10^3 V/A and 1×10^{12} V/A [sr570].

[12] Far away from the centerburst, so the interferogram signal is negligible compared to the noise.

[13] As specified in the manual of transimpedance amplifier [sr570] for the given amplification

[14] Either a first order $R-C$ low-pass with a 3 dB frequency at 18 KHz or a doubled low-pass at 15 KHz, two choices being reasonable for the used amplification of 50 nA/V (High BW mode), see appendix B.1.3.

5.4 Highspeed operation

As mentioned in the introduction, one of the motivations of using intersubband transitions for detecting devices is the possible high speed operation due to the short lifetimes of the upper states. For several reasons, the device N538 appeared to be very suitable for the verification of that assumption:

- Several single mode distributed feedback (DFB) quantum cascade lasers (QCL) in the same wavelength range were accessible at the *Mesoscopic Physics* group and *Alpes Lasers SA* here in Neuchâtel. Working in continuous wave (CW) mode at temperatures reachable by Peltier cooling (−30 °C - 30 °C), these allow either for electrically modulated operation limited to frequencies up to roughly 1 GHz with the electrical equipment at hand, or the use of two of these QCLs in a heterodyne mixing setup, resulting in beating frequencies up to several tens of gigahertz.

- There was already some experience with heterodyne measurements with exactly these lasers in the *Mesoscopic Physics* group and the remains of an existing setup could relatively easily be adapted to the needs of our highspeed characterisation.

- The device N538 works relatively well up to room temperature, eliminating the need of a bulky cryostat, which would have required heavy modifications to one of the existing cryostats. Their presently used wiring did not allow to carry (weak) signals at frequencies far above some hundreds megahertz.

We started with the simpler experiment first, directly modulating one laser with different frequencies between 100 KHz and 3 GHz to verify that our detector is capable to detect higher frequencies and that it is possible to detect the signal with the installed amplifier chain (see section 3.5 for details of the optical and electrical setup). Indeed the signal was visible on the spectrum analyser over the whole range, although with a significant decrease at higher frequencies. Figure 5.14 shows this measurement obtained by modulation of one of the lasers with the signal from a 3.3 GHz function generator [sml03] we could borrow. The detector's signal was visualised with the spectrum analyser [e4402b](3 GHz) after its amplification by the 2.5 GHz *Sonoma 317* ⟨s317⟩ amplifier. As we were also

able to find selected lower frequencies at which the signal almost completely vanished and reappeared for frequencies above, I supposed the decrease in signal above 1.5 GHz was at least partly due to reflections and standing wave effects on the electrical wiring between the laser chip and the bias-T. It was clear that the impedances of the output of the bias-T, the connectors passing into the laser's box, and finally the wire bond to the laser ridge were most probably not perfectly adapted. There was also a slight, but non negligible part of electrical pickup from the laser driving circuit observed by blocking the optical path.

Thus, we went on to the more interesting experiment, the modulation of the incident light by means of the beating obtained from two collinearly aligned laser beams with slightly detuned wavelengths. Creating the modulation in an optical way automatically resolves the two problems mentioned above: First, the modulation depth of the incident light is not influenced by the electrical wiring of the lasers, so it is almost constant[15] for the whole range of beating frequencies and second, there is no electrical pickup possible as there is no other electrical signal at the same frequency around. On the other hand, it took us quite some time to align both laser beams to finally observe a signal. One laser was operated at a temperature of −18 °C and DC currents between 450 mS - 500 mA (25 mW output power), whereas the other was held at −33 °C and operated with 320 mA (8 mW output power). These parameters resulted in an emission wavelength of 1832 cm^{-1} (5.46 µm) for both lasers, hitting almost perfectly the center of the responsivity peak at room temperature (indicated in the responsivity graph in figure 5.8). The current tuning of the first laser is sufficiently large to generate a beating frequency in excess of 20 GHz (0.7 cm^{-1}) between the two emission wavelengths.

A first series of heterodyne measurements done with the same amplifier and spectrum analyser confirmed the results obtained by the direct modulation setup. Obviously, we were still limited by the 2.5 GHz bandwidth of the amplifier. We could then borrow a spectrum analyser with an upper frequency limit of 26.5 GHz [e4407b] and faster amplifiers [miteq,hp83006], so we repeated the measurements, as shown in figure 5.15. The brown curve is data measured first, before we

[15] Almost constant, as the shift in the wavelength of one of the lasers is obtained by slightly changing its driving current, which has a small influence on the modulation depths. The measures with the laser shifted to the opposite direction in wavelength shows that the effect is minor - the curves in figure 5.15 contain data from measurements in both directions.

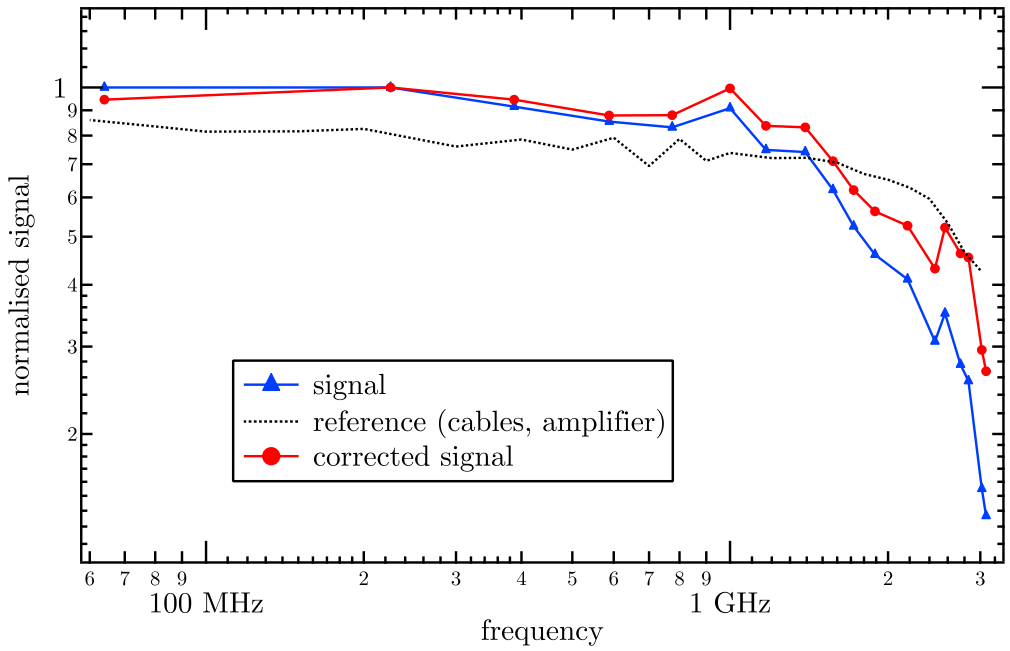


Figure 5.14: Normalised frequency responses of N538 operated at room temperature. The blue curve shows the measured values, the red curve is corrected by the reference measurement (grey) directly on the cables and the amplifier.

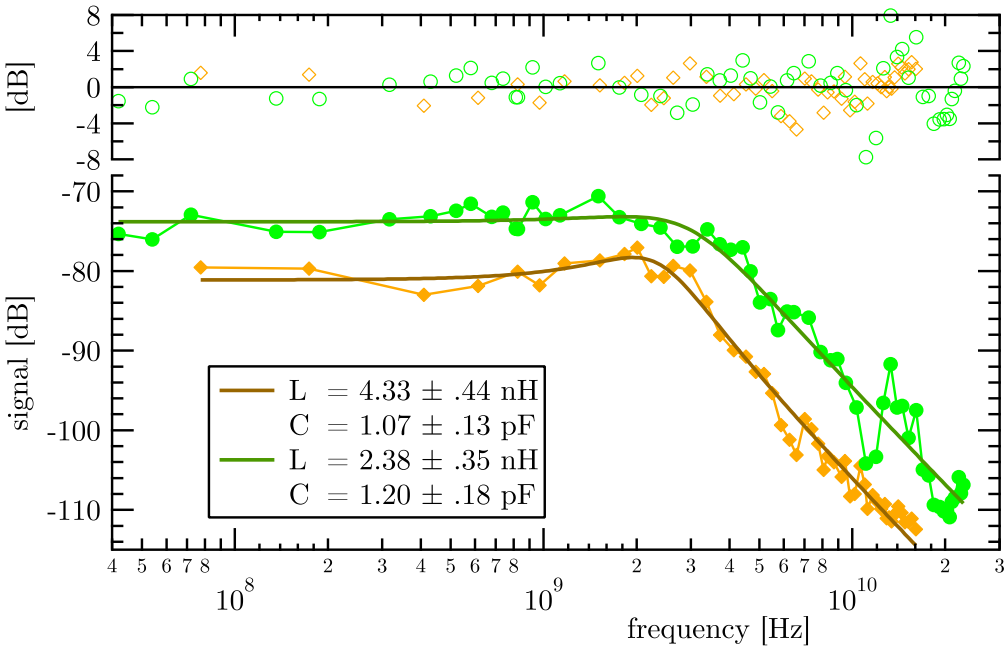


Figure 5.15: Frequency responses of the 5.3 µm QCD N538 operated at room temperature and for two differently connected devices. The green curve corresponds to a device with a 2.5 mm long bond wire, whereas the brown curve describes the previous mounting with a longer bond. The dots are measured data points (corrected by the amplifier gain data from figure 3.12), the solid lines are fits using the RLC model of equation 3.5. The data in the upper part of the graphs are differences between the measurement and fitted values, expressed in dB.

realised that the roll-off behaviour above 2 GHz is mainly due to the influence of the device's capacitance and the inductance of the bondwire (see the equivalent circuit model in figure 3.11, equation 3.5 in section 3.5). Therefore, I remounted the sample to achieve a minimum of stray inductance. With this simple mounting technique, the main target was to find an arrangement with the shortest possible bond wire together with some mechanical modification of the SMA connector. The final mount is shown on the photograph in figure 5.16. Together with a slightly better optical alignment, we were able to observe the signal up to 23 GHz, which already exceeds the frequency for which we have the gain curve data of the *Miteq AFS-5* ⟨miteq⟩ amplifier (figure 3.12). In addidion, it also exceeds the specified maximal frequency of the used SMA coax cables, which is 15 GHz.

Figure 5.16: Mount photograph

Numerically fitting equation 3.5 to both of these data sets results in values of the fitting parameters L and C which correspond well to the values of these quantities estimated from their physical dimensions: An inductor made of a long, straight, round wire with radius r and length l has a high frequency self-inductance given by

$$L = \frac{\mu_0 l}{2\pi} \ln\left(\frac{2l}{r} - 1\right) \tag{5.9}$$

For the 2.5 mm long bond wire (4 mm in the first mounting), equation 5.9 results in an inductance of $L = 2.5\,\mathrm{nH}$ (4.4 nH), which is in good agreement with the values obtained from the fit. The same is true for a capacitor with an area of $A = 100 \times 100\,\mu\mathrm{m}^2$, a thickness of $d = 1.65\,\mu\mathrm{m}$ and using InAlAs/InGaAs as dielectric ($\epsilon = 15$, the dimensions correspond to the actual device). In this case, we get $C = \epsilon\epsilon_0 \frac{A}{d} = 0.82\,\mathrm{pF}$ which agrees reasonably well to the fitted values, too. Looking at the upper part of figure 5.15 showing the fit residuals, or in other words the data corrected by the response function of the second order low-pass filter from equation 3.5, it is thus clear that the roll-off of the QCD signal is not due to the internal mechanism of the device, but rather due to its size and (inappropriate) connection type.

An estimation of the transport time from the excited state to the ground state by subsequent electron-phonon scattering - using the transition times obtained by the simulations done to explain the decrease in responsivity shown in figure 5.11

- yields a value on the order of 5 ps. This corresponds to a 3 dB frequency $f_{3\,\mathrm{dB}} = \frac{1}{2\pi\tau}$ of roughly 32 GHz, a region which might be explored in future experiments by

- either processing the device to smaller sizes (in the order of 50 µm diameter) and carefully trying to match the impedances using coplanar micro-strip lines [59] and finally use cables, amplifiers, and spectrum analysers specified up to this frequency (this is probably too expensive to be done in a small lab).

- or trying to down-convert the frequency on the detector using a second difference frequency generation exploiting a nonlinearity on the $I-V$ curve by "illuminating" the device with a microwave beam [38] at 30 GHz, for example, although in this case one has to carefully protect the sensible amplifier from the direct microwave power.

5.5 Optimised design

Based on the experience with the 5.3 µm detector N538, I tried to optimise its design to achieve better performance in terms of higher responsivity $\mathcal{R}$ and less Johnson noise at elevated temperatures.

The conduction band structure of the redesigned structures N657 and N658 are listed in table 5.3, in comparison with those of N538. I made three changes:

- To achieve a higher responsivity, the doping concentration in the ground state well is raised by a factor of four to allow more light to be absorbed.

- To obtain a higher R_0 together with compensating the fact that the first point would certainly decrease the electrical resistance, the thickness of the barriers in the extraction cascade is significantly enhanced.

- As another way to enhance the absorption, I planned to process one series of the structures with an etched grating coupler instead of the 45° wedge coupler. As this ideally would redirect the incoming light parallel to the layers, the absorption should be significantly enhanced. So I asked for exactly the same structure as the redesigned but with only one third

of the period repetitions (ten instead of 30) - so the maximal achievable responsivity rises by the same factor of three.

Both samples were processed as device N538 before (sections 5.1 and 3.1 respectively), each of them once with a "standard" 45° polished edge and once with a grating coupler. These gratings were done as the very first step in the mesa fabrication. Using an Ar-ion laser, a grating with a period of 1.72 µm was holographically transferred to a layer of photoresist deposited on the wafer. After developing the resist, the lattice period was verfied by optical diffraction and then transferred into the top contact layer by an RIE etch with a depth of approximately 100 nm. After that, the samples were processed together with the other pieces of the wafers designated for the 45° coupling scheme.

5.5.1 Electrical characterisation

For device temperatures between 10 K and room temperature, $I-V$ curves were measured for both devices under dark conditions as described in section 3.2 and the device conductance extracted by numerically fitting the $I-V$ curves in the proximity of zero bias voltage. The resulting R_0A are shown in figure 5.17 together with the values of N538. Two main observations are obvious:

- The device resistance of N657 is not higher than that of N538, but it is even smaller in the temperature range of 20 K to 100 K as well as above 200 K. Apparently, the thicker barriers in the extraction stage do just roughly compensate the elevated charge concentration due to the quadrupled doping.
- The resistances of N658 compared with N657 is three times smaller (as expected due to the reduced number of periods with otherwise identical structure) only at temperatures above 150 K. Below that, the device N658 (10 periods) is roughly two orders of magnitude more conductive than its 30 periods brother.

5.5.2 Optical measurements

For all four samples, N657 and N658 in both 45° and grating configuration, photocurrent spectra were taken in our external beam condenser setup of the

Table 5.3: Growth parameters for N657 and N658 in comparison with N538

N657 & N657				N538		
Material thickness (Å)	Si-doping n: (cm^{-3})	Rep. N657	Rep. N658	Material thickness (Å)	Si-doping n: (cm^{-3})	Rep.
Alloyed contact (table 5.2)				*Alloyed contact (table 5.2)*		
InGaAs 2000	2.5×10^{18}			InGaAs 2000	1.6×10^{18}	
InAlAs 20				InAlAs 20		
InGaAs 59	1.6×10^{18}	30×	10×	InGaAs 59	4.0×10^{17}	30×
InAlAs 60		↑	↑	InAlAs 60		↑
InGaAs 17		·	·	InGaAs 17		·
InAlAs 64		·	·	InAlAs 44		·
InGaAs 20		·	·	InGaAs 20		·
InAlAs 79		·	·	InAlAs 39		·
InGaAs 23		·	·	InGaAs 23		·
InAlAs 77		·	·	InAlAs 37		·
InGaAs 27		·	·	InGaAs 27		·
InAlAs 75		·	·	InAlAs 35		·
InGaAs 32		·	·	InGaAs 32		·
InAlAs 71		·	·	InAlAs 31		·
InGaAs 39		·	·	InGaAs 39		·
InAlAs 68		·	·	InAlAs 28		·
InGaAs 47		↓	↓	InGaAs 47		↓
InAlAs 66		30×	10×	InAlAs 26		30×
InGaAs 6000	2.5×10^{18}			InGaAs 6000	1.6×10^{18}	
InP:Fe semi-insulating substrate				InP:Fe semi-insulating substrate		

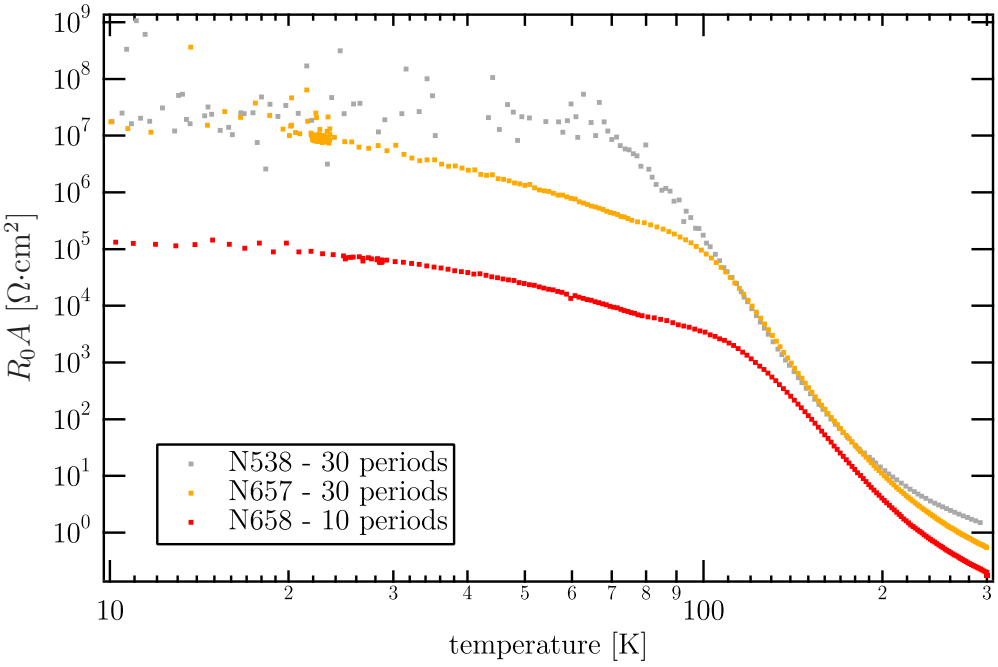


Figure 5.17: Electrical resistance of N657 and N658 in comparison with N538

Bruker IFS 66/S ⟨ifs66⟩ FTIR spectrometer for the whole temperature range[16] from 10 K to 300 K. All these spectra were then processed in the same way as N538 and N516 before. The functions in the script of appendix C.1 were then used to obtain spectra of their responsivity, overall quantum efficiency, different detectivities and extracted from these, peak values as function of temperature. A selection of spectral responsivities of the 30 period device N657 at different temperatures and in both configurations (45° and top grating) are shown in figure 5.19.

Extracted peak responsivities over the whole temperature range are shown in figure 5.18 together with the "reference" of N538. As in the electrical measurements, there is not much of an enhancement compared with the first structure. The responsivities of the grating-coupled samples (dotted lines) are reduced by

[16] As already happend in the case of N538, the continuous spectra recording under slowly varying temperature had to be interrupted because of the mechanical deformation of the cryostat cold finger and the resulting optical misalignment. As I "learned" during these measurements, this problem can easily be resolved by choosing a larger aperture on the spectrometer - which reduces the achievable spectrum resolution, but that is not an issue in measuring photocurrent, whose spectral features are always larger than even the worst resolution of this instrument made for transmission spectroscopy in the infrared.

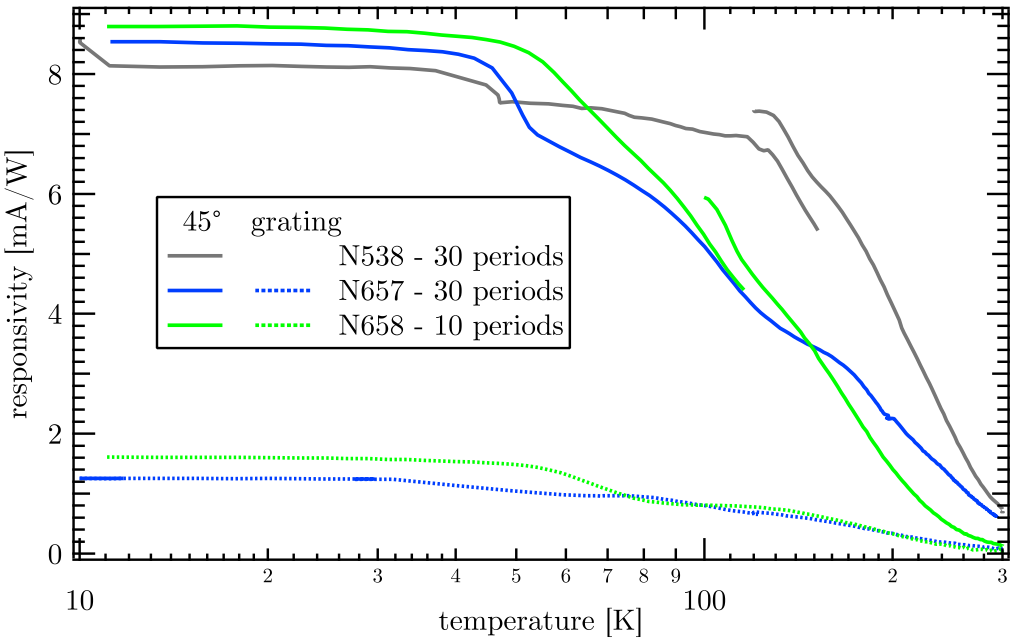


Figure 5.18: Responsivity $\mathcal{R}_{\text{peak}}$ as function of device temperature for N657, N658 in relation to the structure N538. The "holes" and "overlaps" in the curves for N538 and N658 (45°) are due to changed optical alignment and restarted measurements.

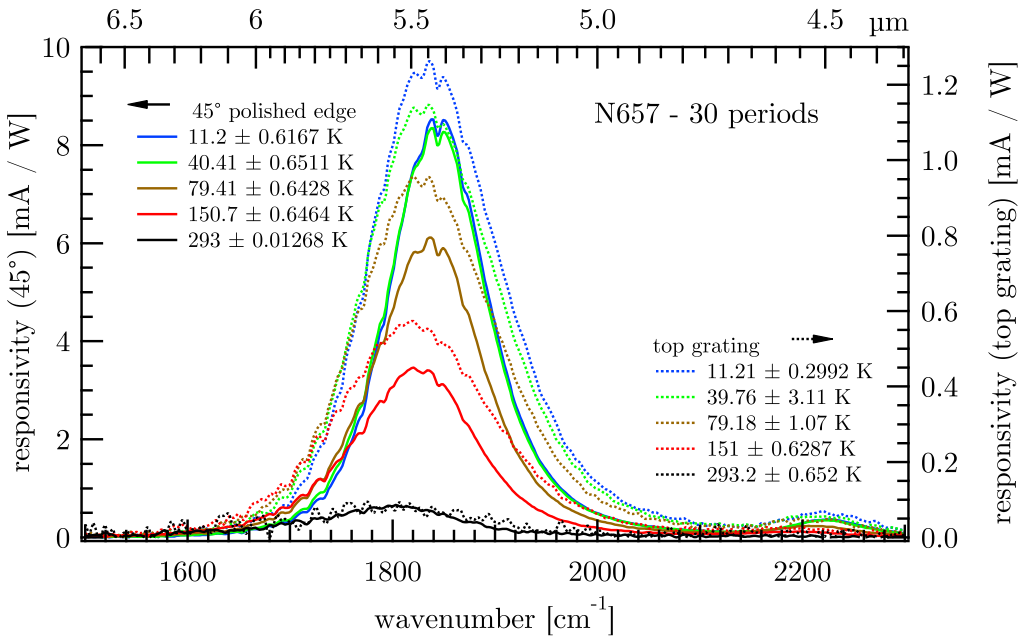


Figure 5.19: Responsivity spectra of N657 for the 45° configuration (left axis) and the grating configuration (right axis).

more than a factor of five in comparison with the 45° wedges, which indicates that the efficiency of the etched grating is rather small. One possible explanation is that the grating, which is only etched into the upper contact layer, is not "seen" very well by the incident light. This remains to be verfied in a future processing run with a grating partially etched into the active zone. The responsivities of the samples with a 45° polished edge are slightly enhanced in comparison with N538, nevertheless, there is no factor four between them, as one would roughly expect due to the raised doping concentration. I suspect the enhanced absorption (which I did not measure on these samples) is cancelled by a reduced escape probability due to the thicker barriers in the extractor. This might also explain the faster degrade in responsivites as function of rising temperature, with the drop-off at around 50 K compared with 130 K of N538.

On the other hand, the fact that the responsivities of N657 and N658 are essentially the same, verifies the prediction from equation 2.41 in section 2.2.4 that under otherwise unchanged parameters - especially the doping concentration - the responsivity should not change, as the loss of $1/N$ in transport is compensated by a gain in absorption of N as long as the total absorption is significantly

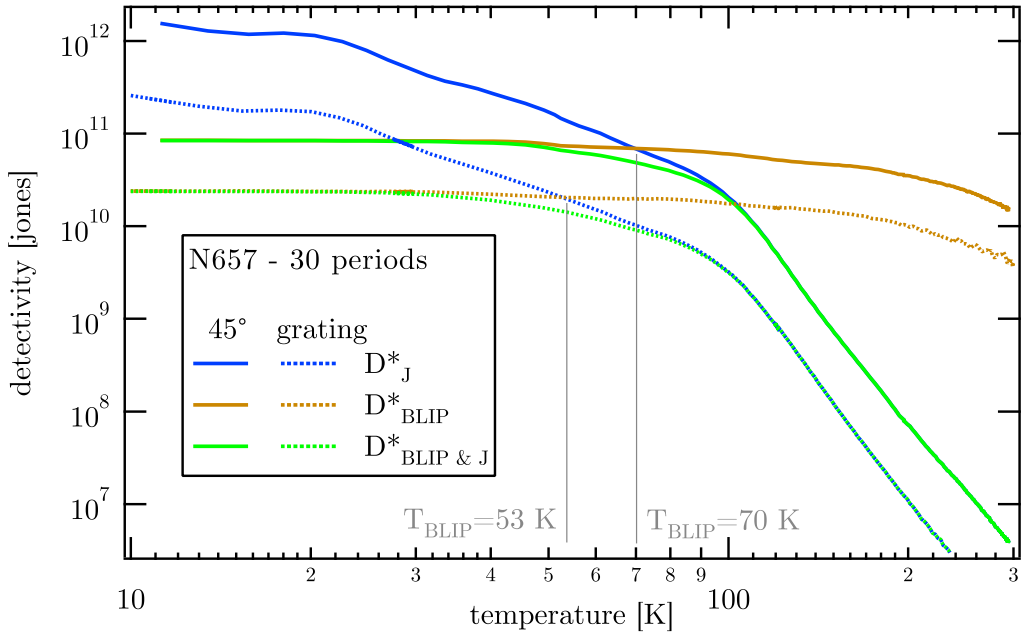


Figure 5.20: Johnson noise and background photon noise limited detectivities for both optical coupling devices of the 30 period QCD N657.

smaller than one.

Treating both main noise sources in a QCD, background photon noise and Johnson noise, with the acquired electrical and optical data, allows the derivation of the detectivity comparison graph, shown in figures 5.20 and 5.20 for N657 and N658, respectively.

The 30 periods structure N657 shows a detectivity behaviour very similar to its predecessor N538. The BLIP temperature of the 45° device, $T_{\mathrm{BLIP}} = 70\,\mathrm{K}$ is almost the same as 80 K for N538 (concerning uncertainties and errors of the underlying measurements). The total detectivity $\mathcal{D}^*_{\mathrm{tot}} = 4.8 \times 10^{10}$ Jones at T_{BLIP} is slightly less than its counterpart for N538, mainly due to the reduced electrical device resistance at that temperature, whereas for low temperatures (completely inside the BLIP regime) $\mathcal{D}^*$ has essentially the same value as for N538. Not astonishing, the device with the grating coupler achieves BLIP figures below that ($\mathcal{D}^*_{\mathrm{tot}} = 1.4 \times 10^{10}$ Jones, $T_{\mathrm{BLIP}} = 53\,\mathrm{K}$) due to its smaller $\mathcal{R}$.

Structure N658, which has only 10 repetitions of the basic period, has an electrical resistance which is accordingly smaller, and even much smaller for low temperatures (figure 5.17). It is therefore no surprise that Johnson noise limited

detectivity $\mathcal{D}_J^*$ intersects with the background photon noise limited $\mathcal{D}_{\mathrm{BLIP}}^*$ at much lower temperatures of $T_{\mathrm{BLIP}} = 29\,\mathrm{K}$ and $16.7\,\mathrm{K}$ for the 45° and grating coupled sample, respectively.

Based on these results, it was too optimistic to change two parameters at the same time (doping, barrier widths) to enhance the performance. Therefore, I suggest to investigate the other two possible structures in the 2-by-2 matrix spawn by the doping concentration and barrier well thickness: A redo of N538 with the higher doping concentration of $1.6 \times 10^{18}\,\mathrm{cm}^{-3}$ and a redo of N657 with the lower concentration of $4 \times 10^{17}\,\mathrm{cm}^{-3}$. This will allow to verify the influence of the barrier thicknesses to the escape probability p_e. Additionally, one might regrow N538 with 100 or 200 periods to enhance the detectivity at high temperatures: The absorption of $\approx 7\,\%$ leaves room for more repetitions without loosing in responsivity.

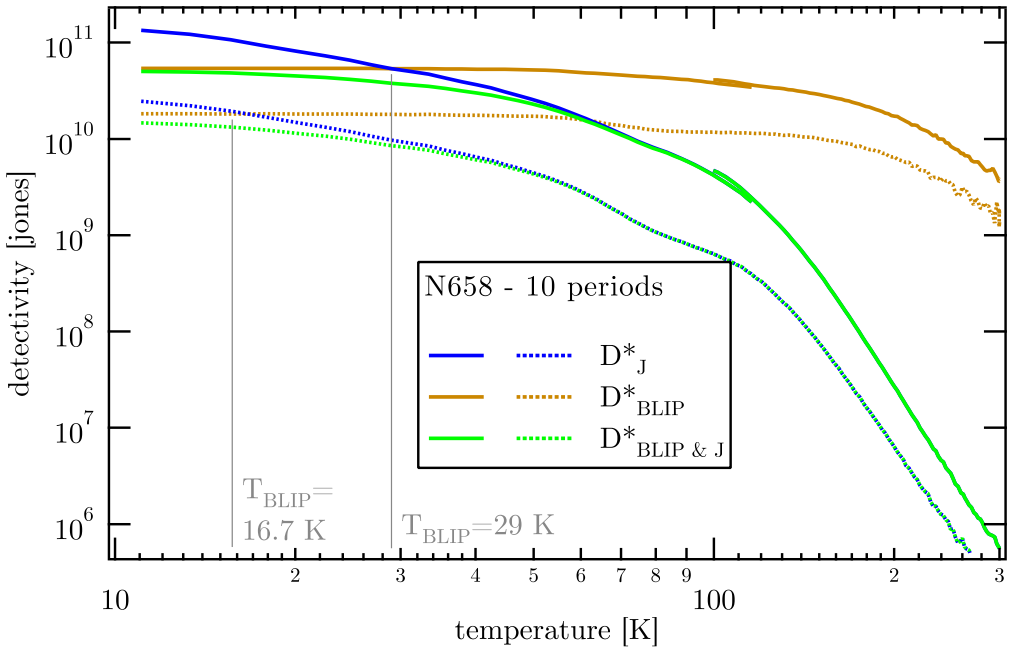


Figure 5.21: Johnson noise and background photon noise limited detectivities for both optical coupling devices of the 10 period QCD N658.

6 Conclusions

In this work, a novel type of semiconductor-based infrared detector has been developed. Thanks to its photovoltaic operation scheme, it eliminates in a most efficient way one of the dominating noise sources of QWIPs, namely dark current noise. In order to operate such a detector, an internal potential gradient rather than an external voltage bias is responsible for the vertical transport of the electrons. In the case of quantum cascade detectors, as explored in this work, this internal potential gradient is provided by a carefully designed series of quantum wells with increasing width, forming a chirped superlattice. The latter provides an efficient electron extraction cascade towards the ground state of the following active region period.

In the scope of the present work, this interesting concept was first used to successfully demonstrate a quantum cascade detector working at 84 µm (3.5 THz) up to 50 K. Later, we also adapted it to the mid-infrared region where the extraction cascade can be made of steps whose height corresponds to the LO-phonon energy. This allows an ultra-fast vertical transport of the electrons and thus high frequency operation of the detector.

Besides the design, fabrication, and experimental characterisation, another main point of this work consisted in the theoretical understanding of some of the physical mechanisms of these mid-infrared detectors. For this purpose, two relatively simple models were used. They allowed a qualitative understanding of two important detector properties as a function of temperature. First, the strong dependence of the device resistance as a function of temperature is well explained by the description of LO-phonon assisted transitions between different states of the cascade, presented in the section about the electrical characterisation of the mid-IR QCDs (equations 5.2-5.4 in section 5.2). Second, a set of numerical calculations was done in order to explain the responsivity drop of these detectors; for this purpose, it used the results of the self-consistent band structure simulations performed by the Schrödinger-Poisson equation solver *sewself* {sewself}.

For both types of QCDs, the measured responsivities $\mathcal{R}$ are in good agreement with the theoretically expected values. Thanks to the wide temperature range in which the 5.3 µm detectors function, and due to the semi-automatic type of measurement and data evaluation, a huge amount of data with a remarkably high confidence level could be recorded. Specifically designed evaluation procedures allowed us to perform noise measurements over the entire investigated temperature range; they agreed well with estimated total values including photonic and thermal noise sources as well as the modeled filtering behaviour and residual noise of the amplifier used.

In addition, it has been shown that quantum cascade detectors are indeed fast. Up to the upper limit of the available high frequency equipment, namely 23 GHz, the observed performance decrease could be explained entirely by the RLC term built up by the parasitic device capacitance and the stray inductance of the "primitive" mounting technique.

The main advantage of the photovoltaic operation mode, namely the absence of dark current, makes these devices attractive for other wavelengths as well. At this moment, work is carried out in our group to implement a quantum cascade detector at $\lambda = 17$ µm for the heterodyne detection of atomic hydrogen in stellar objects. Additional projects exist for the demonstration of such detectors in the near-infrared wavelength range at 1.6 µm. The envisioned applications there range from glucose sensing to time-of-flight measurements for eye-safe high-precision 3D imaging systems. For these detectors, a different material system (InGaAs/AlAsSb) with a larger conduction band discontinuity must be used. Overall, a substantial amount of research has been carried out to understand the basics of such detectors. The accumulated know-how permits to go further in the direction of potential applications as the ones mentioned before.

Acknowledgements

In the first place I wish to express my gratitude to Prof. Daniel Hofstetter, my PhD thesis director, for having provided the opportunity of doing this work in his group, which initially consisted just of the two of us. I thank Daniel for his support and guidance during my time in Neuchâtel as well as the good personal relationship we were able to establish and maintain during this period.

I would like to thank Prof. Vincent Berger (University Denis Diderot - Paris 7, France), PD Dr. Harald Schneider (Forschungszentrum Rossendorf, Deutschland) and Prof. Jérôme Faist (Université de Neuchâtel) for acting as co-examiners of the jury and for critically reading this work. Jérôme has also been a valuable source of knowledge for me, in many discussions as well as providing a lot of ideas in the last four years.

I warmly acknowledge Thierry Aellen for all the details in processing technology I learned from him, a lot of discussions about all the world and his brother and finally (chronologically) for his help in getting the heterodyne setup working again. I would like to thank Dr. Giacomo Scalari with whom I had much enjoyable exchange of ideas in general and a very fruitful collaboration in the characterisation of the THz QCD, specifically.

My thanks go out to Tobias Gresch and Romain Terazzi who stood with me for most of my time in Neuchâtel as office-mate, additionally to Tobias for his partnership in the "meso" administration team and to Romain for lots of theoretical insights. I acknowledge my "young" PhD colleagues Esther Baumann and Fabrizio Giorgetta for planning and equipping our group's lab together with me and the good collaboration. A special thank goes to Dr. Stéphane Blaser for his unfatiguing planning of different social and cultural activities, and to Dr. Yargo Bonetti for the RF and scripting knowledge he shared with me and also for his unbeaten, fine sarcasm. I also thank Dr. Mattias Beck and Dr. Marcella Giovannini together with Nicolas Hoyler for mastering and maintaining the MBE facility and growing the structures. Similarly, I would like to thank the members

of the group in Cambridge, Dr. Harvey Beere, Prof. Giles Davies, Dr. Edmund Linfield and Prof. David Ritchie for the growth of the THz QCD.

The members of the different workshops, namely Daniel Varidel and Urs Bart of the electronics department, Jean-Pierre "Kouni" Bourquin and Isidro Fernandez of the mechanical department and Thierry Delhove maintaining cleanroom and cryogenic facilities, have all my gratitude for their technical support, often needed very fast. The same is very true for all members of the secretariat, providing prompt administrative support.

Sincere thanks go to Hansruedi Benedickter (Microwave Electronics Group, ETH Zürich) and Dr. Alexis Bögli (IMT, Université de Neuchâtel) for technical assistance and for lending high frequency equipment to us.

Finally, I would like to thank my girlfriend Sandra, for her patience and love.

This work was supported by the Swiss National Science Foundation.

Bibliography

[1] L. C. West and S. J. Eglash. First observation of an extremely large-dipole infrared transition within the conduction band of a GaAs quantum well. *Applied Physics Letters*, 46(12):1156–1158, 1985. doi:10.1063/1.95742.

[2] J. Faist, F. Capasso, D. L. Sivco, C. Sirtori, A. L. Hutchinson, and A. Y. Cho. Quantum Cascade Laser. *Science*, 264:553–556, April 1994. http://www.sciencemag.org/cgi/content/abstract/264/5158/553

[3] B. F. Levine, R. J. Malik, J. Walker, K. K. Choi, C. G. Bethea, D. A. Kleinman, and J. M. Vandenberg. Strong 8.2 μm infrared intersubband absorption in doped GaAs/AlAs quantum well waveguides. *Applied Physics Letters*, 50(5):273–275, 1987. doi:10.1063/1.98223.

[4] Alex Harwit and J. S. Harris, Jr. Observation of Stark shifts in quantum well intersubband transitions. *Applied Physics Letters*, 50(11):685–687, 1987. doi:10.1063/1.98066.

[5] B. F. Levine, K. K. Choi, C. G. Bethea, J. Walker, and R. J. Malik. New 10 μm infrared detector using intersubband absorption in resonant tunneling GaAlAs superlattices. *Applied Physics Letters*, 50:1092, 1987. doi:10.1063/1.97928.

[6] B. F. Levine. Quantum-well infrared photodetectors. *Journal of Applied Physics*, 74(8):R1–R81, 1993. doi:10.1063/1.354252.

[7] H. C. Liu. *Quantum well infrared photodetector physics and novel devices*, volume 62 of *Semiconductors and semimetals*, chapter 3. Academic Press, San Diego, 2000.

[8] H. Schneider. Optimized performance of quantum well intersubband infrared detectors: Photovoltaic versus photoconductive operation. *Journal of Applied Physics*, 74(7):4789–4791, 1993. doi:10.1063/1.354352.

[9] C. Schönbein, H. Schneider, G. Bihlmann, K. Schwarz, and P. Koidl. A 10 μm GaAs/$Al_xGa_{1-x}As$ intersubband photodetector operating at zero bias voltage. *Applied Physics Letters*, 68:973–975, 1996. doi:10.1063/1.116116.

[10] H. Schneider, Schönbein C., Bihlmann G., van Son P., and Sigg. H. High-speed infrared detection by uncooled photovoltaic quantum well

infrared photodetectors. *Applied Physics Letters*, 70:1602–1604, 1997. doi:10.1063/1.118628.

[11] G. Hasnain, B. F. Levine, D. L. Sivco, and A. Y. Cho. Mid-infrared detectors in the 3–5 μm band using bound to continuum state absorption in InGaAs/InAlAs multiquantum well structures. *Applied Physics Letters*, 56(8):770–772, 1990. doi:10.1063/1.103186.

[12] R. P. Leavitt and J. W. Little. Infrared photodetector based on intersubband transitions to minigap-confined states in doped quantum wells. *Applied Physics Letters*, 79(13):2091–2093, 2001. doi:10.1063/1.1374482.

[13] G. Sarusi, S. D. Gunapala, J. S. Park, and B. F. Levine. Design and performance of very long-wavelength GaAs/$Al_xGa_{1-x}As$ quantum-well infrared photodetectors. *Journal of Applied Physics*, 76(10):6001–6008, 1994. doi:10.1063/1.358351.

[14] S.D. Gunapala, S.V. Bundara, J.K. Liu, Winn Hong, Mani Sundaram, P.D. Maker, R.E. Muller, C.A. Shott, and R.; Carralejo. Long-wavelength 640 $\times$ 486 GaAs-AlGaAs quantum well infraredphotodetector snap-shot camera. *IEEE Transactions on Electron Devices*, 45(9):1890–1895, 1998. doi:10.1109/16.711352.

[15] H. Schneider, P. Koidl, M. Walther, J. Fleissner, R. Rehm, E. Diwo, K. Schwarz, and G. Weimann. Ten years of QWIP development at Fraunhofer IAF. *Infrared Physics & Technology*, 42:283–289, 2001. doi:10.1016/S1350-4495(01)00086-X.

[16] H. C. Liu, R. Dudek, A. Shen, E. Dupont, C. Y. Song, Z. R. Wasilewski, and M. Buchanan. High absorption ($> 90\%$) quantum-well infrared photodetectors. *Applied Physics Letters*, 79(25):4237–4239, 2001. doi:10.1063/1.1425066.

[17] H.C. Liu, T. Oogarah, E. Dupont, Z.R. Wasilewski, M. Byloos, M. Buchanan, F. Szmulowicz, J. Ehret, and G.J. Brown. p-type quantum well infrared photodetectors covering wide spectrum. *Electronics Letters*, 38(16):909–911, 2002. doi:10.1049/el:20020644.

[18] AIM Infrarot-Module GmbH. Infrared cameras. http://www.aim-ir.com/pages/products/

[19] FLIR Systems. ThermaCAM SC 3000. http://www.flirthermography.com/cameras/camera/1008/

[20] IRcam. Geminis series. http://www.ircam.de/produkte/geminis_serie_e.php

[21] Sofradir. Sirius.
http://www.sofradir.com/_pdf/2006_04_sirius_sampling.pdf

[22] Daniel Hofstetter, Mattias Beck, and Jérôme Faist. Quantum-cascade-laser structures as photodetectors. *Applied Physics Letters*, 81:2683, 2002. doi:10.1063/1.1512954.

[23] Rüdeger Köhler, Alessandro Tredicucci, Fabio Beltram, Harvey E. Beere, Edmund H. Linfield, A. Giles Davies, David A. Ritchie, Rita C. Iotti, and Fausto Rossi. Terahertz semiconductor-heterostructure laser. *Nature*, 417:156–159, 2002. doi:10.1038/417156a.

[24] L. Ajili, G. Scalari, D. Hofstetter, M. Beck, J. Faist, H. Beere, G. Davies, E. Linfield, and D. Ritchie. Continuous-wave operation of far-infrared quantum cascade lasers. *Electronics Letters*, 38:1675–1676, 2002. doi:10.1049/el:20021143.

[25] R. Köhler, A. Tredicucci, F. Beltram, H.E. Beere, E.H. Linfield, D.A. Ritchie, and A.G. Davies. Quantum cascade lasers emitting at lambda greater than 100 μm. *Electronics Letters*, 39(17):1254–1255, 2003. doi:10.1049/el:20030779.

[26] L. Ajili, J. Faist, H. Beere, D. Ritchie, G. Davies, and E. Linfield. Loss-coupled distributed feedback far-infrared quantum cascade lasers. *Electronics Letters*, 41(7):419–421, 2005. doi:10.1049/el:20050128.

[27] W. H. Haydl, S. Smith, and R. Bosch. 100-GHz gunn diodes fabricated by molecular beam epitaxy. *Applied Physics Letters*, 37(6):556–557, 1980. doi:10.1063/1.91985.

[28] D. T. Hodges and M. McColl. Extension of the schottky barrier detector to 70 μm (4.3 THz) using submicron-dimensional contacts. *Applied Physics Letters*, 30(1):5–7, 1977. doi:10.1063/1.89208.

[29] Zhiping Jiang and X.-C. Zhang. Electro-optic measurement of THz field pulses with a chirped optical beam. *Applied Physics Letters*, 72(16):1945–1947, 1998. doi:10.1063/1.121231.

[30] Y. Cai, I. Brener, J. Lopata, J. Wynn, L. Pfeiffer, J. B. Stark, Q. Wu, X. C. Zhang, and J. F. Federici. Coherent terahertz radiation detection: Direct comparison between free-space electro-optic sampling and antenna detection. *Applied Physics Letters*, 73(4):444–446, 1998. doi:10.1063/1.121894.

[31] A. G. Kazanskii, P. L. Richards, and E. E. Haller. Far-infrared photoconductivity of uniaxially stressed germanium. *Applied Physics Letters*, 31(8):496–497, 1977. doi:10.1063/1.89755.

[32] E. E. Haller, M. R. Hueschen, and P. L. Richards. Ge : Ga photoconductors in low infrared backgrounds. *Applied Physics Letters*, 34(8):495–497, 1979. doi:10.1063/1.90861.

[33] G. J. Stacey, J. W. Beeman, E. E. Haller, N. Geis, A. Poglitsch, and M. Rumitz. Stressed and unstressed ge:ga detector arrays for airborne astronomy. *International Journal of Infrared and Millimeter Waves*, 13(11):1689–1707, November 1992. doi:10.1007/BF01010739.

[34] Marcel Graf, Giacomo Scalari, Daniel Hofstetter, Jérôme Faist, Harvey Beere, Edmund Linfield, David Ritchie, and Giles Davies. Terahertz range quantum well infrared photodetector. *Applied Physics Letters*, 84(4):475–477, 2004. doi:10.1063/1.1641165.

[35] H. C. Liu, C. Y. Song, A. J. SpringThorpe, and J. C. Ca. Terahertz quantum-well photodetector. *Applied Physics Letters*, 84(20):4068–4070, 2004. doi:10.1063/1.1751620.

[36] Laure Gendron, M. Carras, A. Huynh, V. Ortiz, C. Koeniguer, and V. Berger. Quantum cascade photodetector. *Applied Physics Letters*, 85:2824, 2004. doi:10.1063/1.1781731.

[37] Vincent Berger. French patent No. 0109754, 2001.

[38] H. C. Liu, Jianmeng Li, E. R. Brown, K. A. McIntosh, K. B. Nichols, and M. J. Manfra. Quantum well intersubband heterodyne infrared detection up to 82 GHz. *Applied Physics Letters*, 67(11):1594–1596, 1995. doi:10.1063/1.114950.

[39] S. Ehret, H. Schneider, J. Fleissner, P. Koidl, and G. Bohm. Ultrafast intersubband photocurrent response in quantum-well infrared photodetectors. *Applied Physics Letters*, 71(5):641–643, 1997. doi:10.1063/1.119815.

[40] Manfred Helm. *The Basic Physics of Intersubband Transitions*, volume 62 of *Semiconductors and semimetals*, chapter 1. Academic Press, San Diego, 2000.

[41] Gerald Bastard. *Wave mechanics applied to semiconductor heterostructures*. Monographies de physiques. Les editions de physique, Les Ulis, France, 1996.

[42] Carlo Sirtori, Federico Capasso, Jérôme Faist, and Sandro Scandolo. Nonparabolicity and a sum rule associated with bound-to-bound and bound-to-continuum intersubband transitions in quantum wells. *Physical Review B*, 50(12):8663–8674, 1994. doi:10.1103/PhysRevB.50.8663.

[43] A. Rogalski. Quantum well photoconductors in infrared detector technology. *Journal of Applied Physics*, 93(8):4355–4391, 2003. doi:10.1063/1.1558224.

[44] Hans Zogg. Optronische Sensoren. Memo, 4 2004.

[45] R. Clark Jones. 'Detectivity': the reciprocal of noise equivalent input of radiation. *Nature*, 170(4335):937–938, 11 1952. doi:10.1038/170937b0.

[46] R. Clark Jones. Phenomenological description of the response and detecting ability of radiation detectors. *Proc. IRE*, 47(9):1495–1502, 1959.

[47] H. C. Liu. Noise gain and operating temperature of quantum well infrared photodetectors. *Applied Physics Letters*, 61(22):2703–2705, 1992. doi:10.1063/1.108115.

[48] B. F. Levine, A. Zussman, J. M. Kuo, and J. de Jong. 19 μm cutoff long-wavelength GaAs/$Al_xGa_{1-x}As$ quantum-well infrared photodetectors. *Journal of Applied Physics*, 71(10):5130–5135, 1992. doi:10.1063/1.350618.

[49] B. F. Levine, A. Zussman, S. D. Gunapala, M. T. Asom, J. M. Kuo, and W. S. Hobson. Photoexcited escape probability, optical gain, and noise in quantum well infrared photodetectors. *Journal of Applied Physics*, 72(9):4429–4443, 1992. doi:10.1063/1.352210.

[50] B. Xing, H. C. Liu, P. H. Wilson, M. Buchanan, Z. R. Wasilewski, and J. G. Simmons. Noise and photoconductive gain in AlGaAs/GaAs quantum well intersubband infrared photodetectors. *Journal of Applied Physics*, 76(3):1889–1894, 1994. doi:10.1063/1.357713.

[51] S. D. Gunapala, B. F. Levine, L. Pfeiffer, and K. West. Dependence of the performance of gaas/algaas quantum well infrared photodetectors on doping and bias. *Journal of Applied Physics*, 69(9):6517–6520, 1991. doi:10.1063/1.348861.

[52] C. Koeniguer, G. Dubois, A. Gomez, and V. Berger. Electronic transport in quantum cascade structures at equilibrium. *Physical Review B*, 74(23):235325, 2006. doi:10.1103/PhysRevB.74.235325.

[53] Giacomo Scalari, Lassaad Ajili, Jérôme Faist, Harvey Beere, Edmund Linfield, David Ritchie, and Giles Davies. Far-infrared 87 μm bound-to-continuum quantum-cascade lasers operating up to 90 K. *Applied Physics Letters*, 82(19):3165–3167, 2003. doi:10.1063/1.1571653.

[54] S. G. Matsik, M. B. M. Rinzan, A. G. U. Perera, H. C. Liu, Z. R. Wasilewski, and M. Buchanan. Cutoff tailorability of heterojunction terahertz detectors. *Applied Physics Letters*, 82(1):139–141, 2003. doi:10.1063/1.1534409.

[55] Laure Gendron, C. Koeniguer, V. Berger, and X. Marcadet. High resistance narrow band quantum cascade photodetectors. *Applied Physics Letters*, 86:121116, 2005. doi:10.1063/1.1884257.

[56] Laure Gendron, C. Koeniguer, V. Berger, and X. Marcadet. Quantum cascade detectors. *Infrared Physics and Technology*, 47:175–181, 2005. doi:10.1016/j.infrared.2005.02.023.

[57] Laure Gendron. *Transport electronique dans les detecteurs infrarouge à puits quantiques*. Ph.D. thesis, Université de Paris 7 - Dennis Diderot, 2005.

[58] R. Ferreira and Gerald Bastard. Evaluation of some scattering times for electrons in unbiased and biased single- and multiple-quantum-well structures. *Physical Review B*, 40(2):1074–1086, 1989. doi:10.1103/PhysRevB.40.1074.

[59] P. D. Grant, R. Dudek, L. Wolfson, M. Buchanan, and H.C. Liu. Ultra-high frequency monolithically integrated quantum well infrared photodetector up to 75 Hz. *Electronics Letters*, 41(4):214–215, 2005. doi:10.1049/el:20057428.

Software

{jabref} Morten O. Alver, Nizar N. Batada, Michel Baylac, Guillaume Gardey, Cyrille d'Haese, Raik Nagel, Ellen Reitmayr, Andreas Rudert, Michael Spiegel, Ulrik Stervbo, Dominik Waßenhoven, Joerg K. Wegner, Michael Wrighton, Egon Willighagen, and Jörg Zieren. JabRef. An open source bibliography reference manager. http://jabref.sourceforge.net

{opus} Bruker Optik GmbH. OPUS. OPtics Users Software. http://www.brukeroptics.com/opus

{extlib} Nicolas Cannasse, Brian Hurt, and Yamagata Yoriyuki. OCaml ExtLib. Extended standard library for OCaml. http://ocaml-lib.sourceforge.net

{eclipse} Eclipse Foundation Inc. Eclipse IDE. http://www.eclipse.org/

{tetex} Thomas Esser. teTEX. A complete TeX distribution for UNIX compatible systems. http://www.tug.org/teTeX

{sewself} Jérôme Faist. Sewself. Self-consistent Schrödinger-Poisson solver.

{bash} Free Software Foundation. Bourne again Shell. GNU command language interpreter. http://www.gnu.org/software/bash/bash.html

{inkscape} Bryce Harrington et al. Inkscape. SVG compliant open source vector graphics editor. http://www.inkscape.org

{json-wheel} Mika Illouz and Martin Jambon. JSON-wheel. JSON library for OCaml. http://martin.jambon.free.fr/json-wheel.html

{json-static} Martin Jambon. JSON-static. A JSON validator and converter for OCaml. http://martin.jambon.free.fr/json-static.html

{komascript} Markus Kohm et al. KOMA-Script. A versatile LaTeX 2ε bundle. http://www.komascript.de/

{latex} Leslie Lamport. LaTeX: A Document Preparation System, 1986.

{ocaml} Xavier Leroy, Didier Rémy, Jérôme Vouillon, and Jacques Garrigue. Objective Caml. A general-purpose programming language, designed with program safety and reliability in mind, supporting functional, imperative, and object-oriented programming styles. http://caml.inria.fr/ocaml/

{qucs} Michael Margraf, Stefan Jahn, Jens Flucke, Raimund Jacob, Vincent Habchi, and Toyoyuki Ishikawa. Qucs, 2006. Quite Universal Circuit Simulator. http://qucs.sourceforge.net/

{labview} National Instruments. LabVIEW 8. A graphical development environment, for signal acquisition, measurement analysis, and data presentation. http://www.ni.com/labview

{texlipse} Oskar Ojala, Kimmo Karlsson, Boris von Loesch, and Tor Arne Vestbø. TeXlipse 1.1. LaTeX support plugin for the Eclipse IDE. http://texlipse.sourceforge.net/

{bibtex} Oren Patashnik and Leslie Lamport. BibTeX, 1988.

{unison} Benjamin C. Pierce et al. Unison. A cross-platform file-synchronization tool for Unix and Windows. http://www.cis.upenn.edu/~bcpierce/unison/

{linux} Linus Benedict Torvalds. Linux. An open source kernel. http://www.kernel.org/

{igor} WaveMetrics, Inc. Igor Pro 5. An interactive software environment for experimentation with scientific and engineering data and for the production of publication-quality graphs and page layouts. http://www.wavemetrics.com/products/igorpro/igorpro.htm

Hardware

⟨e4402b⟩ Agilent. E4402B spectrum analyser. 3 GHz.
http://www.home.agilent.com/agilent/product.jspx?pid=413310

⟨e4407b⟩ Agilent. E4407B spectrum analyser. 26.5 GHz.
http://www.home.agilent.com/agilent/product.jspx?pid=413558

⟨ifs66⟩ Bruker Optik GmbH. IFS 66/S FTIR, 2005.
http://www.brukeroptics.com/ftir/ifs66.html

⟨egg7265⟩ EG&G Instruments Corporation. DSP lock-in amplifier 7265.
http://www.signalrecovery.com/7265.htm

⟨egg5113⟩ EG&G Instruments Corporation. Low noise pre-amplifier Model 5113.
http://www.signalrecovery.com/5113page.htm

⟨hp83006⟩ Hewlett Packard. HP 83006 highspeed amplifier. 24 dB, 26 GHz.
http://www.home.agilent.com/agilent/facet.jspx?k=83006

⟨ux20⟩ Ircon. Ultimax UX-20. Non contact infra-red pyrometer.
http://www.irconindia.co.in/web/prod/ultimax.php

⟨mjb3⟩ Karl Süss. MJB3 mask aligner.
http://www.suss.com

⟨sm2410⟩ Keithley Instruments Inc. Source Meter 2410.
http://www.keithley.com/products/currentvoltage/?mn=2410

⟨ls331⟩ Lake Shore Cryotronics, Inc. Model 331 temperature controller.
http://www.lakeshore.com/temp/cn/331po.html

⟨miteq⟩ Miteq. AFS-5 highspeed amplifier. 26 dB, 20 GHz.
http://www.miteq.com/micro/amps/afsamps/c24c/index.html

⟨nic800⟩ Nicolet Instrument Corporation. Nicolet 800 FTIR.

⟨laserstar⟩ Ophir Optronics. LaserStar powermeter with AN/2 monitor.
http://www.ophiropt.com/laser/products.htm

⟨sml03⟩ Rhode & Schwarz. SML 03 signal generator. 9 kHz - 3.3 GHz.
http://www.rohde-schwarz.com/product/sml.html

⟨s317⟩ Sonoma Instrument Co. Model 317 highspeed amplifier. 38 dB, 10 KHz - 2.5 GHz.
http://www.sonoma-instrument.com/products.htm

⟨sr570⟩ Stanford Research Systems. Low noise current pre-amplifier SR570.
http://thinksrs.com/products/SR570.htm

⟨utp1100⟩ UniTemp GmbH. UTP-1100 rapid thermal annealing oven.
http://www.unitemp.de/index.php?id=36&showdetail=true

A Samples overview

Table A.1 gives on overview of all samples investigated in the scope of this work.

Table A.1: Sample list

Sample			Period thickness (Å)		
#	Comments	Rep.	designed	measured	rel. err.
N45	AlGaAs/GaAs QWIP		316	298.9	−5.41 %
N64	Dito, delta-doped		316	288.85	−8.59 %
N77	InGaAs bound-to-quasicontinuum QWIP		346	352.43	+1.86 %
N128	Redo N77		346	328.5	−5.06 %
N129	Redo N77		346	348.5	+0.72 %
A2879	AlGaAs/GaAs QCD FIR		1088	1108	+1.84 %
N298	redo A2879, modulation doped in 3 barriers, modified stair	50	1063	1075	+1.13 %
N303	Dito, modulation doped in 4 wells	50	1063	1103.4	+3.8 %
N317	Redo A2879 *as close as possible*, one barrier thicker		1123	1138.2	+1.35 %
N318	Trial at 160 um		846	846.18	+0.02 %
N491	InP QWIP at 5.2 µm	60	334	334.4	+0.12 %
N516	InP QCD at 9 µm	50	426	423	−0.7 %
N538	InP QCD at 5.3 µm	30	564	580.64	+2.95 %
N657	Redo 538 with thicker barriers, more doping	30	824	848.57	+2.98 %
N658	Dito	10	824	837.68	+1.66 %

B Labbook extracts

This appendix contains several snippets of information which might be usefull for me and others in measuring or maintaining lab equipment.

B.1 Instrument particularities

B.1.1 Keithley SourceMeter 2410

Integration times as functions of instrument settings.

mode	NPLC (number of power line cycles)	integration time τ (ms)	corresponding bandwidth $1/(2\tau)$ (Hz)
fast	0.01	0.2	2500
medium	1	20	25
slow	10	200	2.5

B.1.2 Bruker - OPUS

The absolute scale of signals throughout the *IFS 66/S* ⟨ifs66⟩ FTIR spectrometer and the controller software *OPUS* {opus} is as follows:

Interferogram (IgSm)

The full range of $\pm 10\,\mathrm{V}$ at the input of the ADC converter (or $\pm 10\,\mathrm{V}/G_{\mathrm{ADC}}$ for an ADC-gain $G_{\mathrm{ADC}} > 1$) is mapped to $\pm 32\,768\,\mathrm{counts}$[1] in the signal setup dialog window. The same signal corresponds to values of ± 1 in the main display window as well as to ± 1 in exported *JCAMP-DX* `.dx`,`.jdx` files.

Spectrum (ScSm)

The (single channel) spectra displayed in the main window are either scaled $\times 1$ or $\times 100$ the power spectrum of the electrical input signal (generally $\times 1$ directly after the fourier transformation, $\times 100$ after manual calculations in the *Spectrum Calculator*, probably because *OPUS* "thinks" in percents), that time a possible ADC-gain G_{ADC} already accounted for!

Values in exported *JCAMP-DX* `.dx`,`.jdx` files are always $\times 100$, so if one wants absolute values in *Igor Pro* {igor}, for example, one has to divide by 100.

[1] probably $-32\,768$ .. $32\,767$ corresponding to the range defined by 16 bit signed integers

B.1.3 SR570 bandwidth & noise specifications

The manual for [sr570] current amplifier only specifies maximum bandwidth and noise figures for full decade steps (1, 10, ...) of amplifier gain. For the intermediate gain settings (2, 5, 20, 50, ...), the specifications can only by estimated; the experience shows, that generally these values change rather slowly, but sudden dramatic changes occur between certain gain values, indicated with horizontal lines in table B.1. This behaviour might be due to changes of distribution of the overall amplification to different amplifier stages, as switching gains between these values emits some clicking noises from the amplifier casing.

B.1.4 Cryostat capacitances

Stray capacitance of the cryostat with some different configuration of BNC-cables and on both sample positions.

capacitance (pF @ 1 KHz)	meas.	corr.
Offset:		
LCR-meter + 4 mm/BNC adapter	3.9	-
green BNC cable (1 m)	85.3	81.4
green BNC + LEMO adapter	87.5	83.6
green BND, LEMO adapter, sample "up"	285.4	281.5
green BNC, LEMO adapter, sample "down"	308.5	304.6
blue BNC cable	160.8	156.9
black BNC cable	198.6	194.7

B.1.5 Nicolet 800

This section is dedicated to various bits of knowledge I acquired in using and repairing the *Nicolet 800 FTIR* ⟨nic800⟩ and its controlling computer system. I assume these might be important for future users of this system, and I hope this is one of the places the information will "survive" and also will be found.

Usage

Slow computer

If the controlling computer is extremly slow (approximately one second per key press), it lost its date and time settings. Use the `DATE` command of the `NICOS:` environement (`MON` to exit from `sx>`) to set its date and time (of the form as `24 AUG 06 14:38`), and it will behave normal again.

Table B.1: Bandwidths and noise figures of the SR570 transimpedance preamplifier as specified by the manual. Horizontal lines indicate sudden changes in bandwidth and probably noise as well.

Mode:	*High Bandwidth*		*Low noise*	
Sensitivity (A/V)	Bandwidth (Hz)	Noise ($A/\sqrt{Hz}$)	Bandwidth (Hz)	Noise ($A/\sqrt{Hz}$)
1 m	1 MHz	150 p	1 MHz	150 p
500 µ	·	·	·	·
200 µ	·	·	·	·
100 µ	1 MHz	100 p	500 KHz	60 p
50 µ	·	·	·	·
20 µ	·	·	·	·
10 µ	800 KHz	60 p	200 KHz	2 p
5 µ	·	·	·	·
2 µ	·	·	·	·
1 µ	200 KHz	2 p	20 KHz	600 f
500 n	·	·	·	·
200 n	·	·	·	·
100 n	20 KHz	600 f	2 KHz	100 f
50 n	·	·	·	·
20 n	·	·	·	·
10 n	2 KHz	100 f	200 Hz	60 f
5 n	·	·	·	·
2 n	·	·	·	·
1 n	200 Hz	60 f	15 Hz	10 f
500 p	·	·	·	·
200 p	·	·	·	·
100 p	100 Hz	10 f	10 Hz	5 f
50 p	·	·	·	·
20 p	·	·	·	·
10 p	20 Hz	10 f	10 Hz	5 f
5 p	·	·	·	·
2 p	·	·	·	·
1 p	10 Hz	5 f	10 Hz	5 f

Humidity in detector chamber

The title already indicates it: Take care keeping the detector chamber dry, as the flatband cable providing the detector bus runs at the bottom of the chamber, including the open connectors.

What actually happend after a period of intense usage of the internal MCT detector (liquid nitrogen cooled) is that the accumulated humidity in the chamber lead to corrosion, probably together with a short circuit between the detector bus' power supply lines ($\pm 15\,\mathrm{V}$) and the electrical ground. The apparent problem was that no detector worked on the bus, although a check on the lines showed the required voltage values.

After a thorough inspection of the electronic board providing the detector bus, we identified the problem: on of the current limiting resistors of the power supply circuits burnt, but not completely. It just had a (residual) resistance of $\sim 20\,\mathrm{K}\Omega$ instead of its specified value ($10\,\Omega$), which was sufficient to provide the correct voltage under open conditions, but could no longer provide enough current for the preamplifier on the detector module.

The solution was simple, I just soldered another resistance with the correct value on top of the burnt one and replace the corroded cables and connectors. But as in another case something more delicate (and much more complicated to find and replace) might be destroyed, I strongly suggest to regularly check the detector chamber and dry it, if needed.

HD replacement

Another weak point is the controlling computer, and apparently its hard disk is the weakest. The original harddisk had already been replaced before, laboriously setup from the floppies by Michel Rochat. As a provident measure, he also prepared and installed a second, "spare" harddisk *just in case*. Luckily, the spare one died (as it was constantly energised by mistake), so I decided to take a snapshot of the working disk to be prepared for future failures.

The working harddisk is a "Quantum LP240S G" model (SCSI, 240 Mb), originally from an Apple Macintosh computer. The snapshot (or dump) was created on a linux system (kernel 2.6.3) where this disk was temporarely attached to, with the command

```
$ dd if=/dev/scsi/host0/bus0/target6/lun0/disc of=dump
479350+0 records in
479350+0 records out
```

The resulting file 'dump' is stored on a CD attached to the documentation of the spectrometer. To get it onto another SCSI disk (>= 240 Mb), one needs to put it back with

```
$ dd if=/path/to/dumpfile of=/dev/of/the/new/harddisk
```

or its SCSI-specific counterpart `sg_dd` from the *sg3_utils*[2] toolkit

```
$ sg_dd if=/path/to/dumpfile of=/dev/of/the/new/harddisk
```

Then, the disk must be set to SCSI ID 6 and attached to the controlling computer, and with a bit of luck, it should just boot. This was succesfully tested with a "Quantum LPS270S" (270 Mb), which is ready inside the computer to be used as "spare" part, although the recognised size is smaller than that of the 240 Mb disk. This might be due to a (nowadays) unusual SCSI blocksize of 2560, as the dump corresponds to 95 744 sectors according to `DST` on `NICOS:`, whereas the 270 Mb disk is reported with only 31 232 sectors! *Setblocksize*[3] is a small tool to set blocksizes to arbitrary values on SCSI disks, which might be usefull but was not needed in this case. One needs a C compiler to get it working, *gcc* just workes fine.

In the meantime, we could acquire some more harddisk of the model "Quantum LP240S", to which I also copied the snapshot, but did not test them up to now.

B.1.6 MJB3 mask aligner

Figure B.1 show a sketch of the differential micrometers screws on the *MJB3* ⟨mjb3⟩ mask aligner, just in case someone took one of them apart - which happens by loosening the "fine" adjustment knob too much. The procedure to put it together is as follows:

1. Screw the the "coarse" handle completely into the loose "inner screw" (brass) and put them back, fitting the the lobe into the knotch (see left part of the sketch) so the "inner screw" can not turn, but move in and out.

2. Screw the "fine" knob between the fixed shell and the "inner screw" by carefully aligning the "inner screw" with the aid of the "coarse" handle. Be extremly careful to not destroy the thread, do not force!

[2] http://sg.torque.net/sg/sg3_utils.html, contains a whole lot of SCSI-specific utilities like `sg_readcap /dev/sg0` (gets size info), `sginfo -d /dev/sg0` (shows drive defects lists), ...

[3] http://io.debian.net/ tar/debian/setblocksize/setblocksize-V0.1.tar.bz2

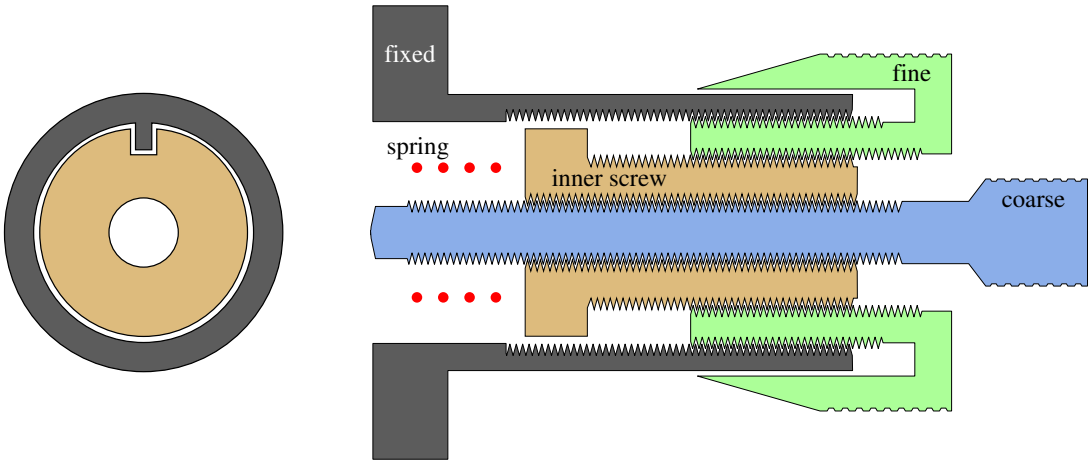


Figure B.1: Alignment handle of the MJB3 mask aligner

C Scripts, software

A selection of the scripts and other software written within the scope of this work.

C.1 Detector calculations

Igor {igor} procedures used for the responsivity and detectivity determination as described in section 3.3.2, based on FTIR spectra and interferograms as well as beamsplitter and blackbody references.

Listing C.1: MG_detectorcalc.ipf: main file

```
#pragma rtGlobals=1 // Use modern global access method.
#include "MG_KBr4RT"
#include "MG_Planck"
static strconstant packagesbase="root:Packages:MG_detectorcalc"
static constant HeNe=15798.2 // [cm-1]
static constant maxfreq=200000 // [Hz]
static constant freqpnts=32768
constant C100=3.04e-10, C200=2.81e-10 // [F]
function one_argument_func(x)
  variable x
  return NaN
end // one_argument_func
function defaultRange(x) // returns bitmap: bit 0 (=1): include in BLIP photons, bit 1 (=2): search peak here
  variable x
  return 1
end // defaultRange
function rN538(x) // range function for N538
  variable x
  variable ret=0
  if (x > 1200 && x < 6000)
    ret+=1
  endif
  if (x > 1700 && x < 2100)
    ret+=2
  endif
  return ret
end // rN538
Structure Results
  double R_max
  double R_maxloc
  double eta_max
  double eta_maxloc
  double DsBLIP_max
  double DsBLIP_maxloc
  double DsJ_max
  double DsJ_maxloc
  double NoiseEL_BG_2
  double NoiseEL_J_2
```

```
39    double NoiseEL_FTIR_2
40    double SumNoise
41    double MeasNoise
42  EndStructure
43  static function printer(s)
44    string s
45    Struct results res
46    structget/S res s
47    printf "peak R: %g -> eta: %g @ %g cm-1\r", res.R_max, res.eta_max, res.R_maxloc
48    printf "peak D*: BLIP %g, J %g -> %g\r", res.DsBLIP_max, res.DsJ_max, 1/sqrt( 1/(res.
          DsBLIP_max)^2 + 1/(res.DsJ_max)^2 )
49    printf "Noise electrons ^2: BG %g, J %g, FTIR %g\r", res.NoiseEL_BG_2, res.NoiseEL_J_2, res.
          NoiseEL_FTIR_2
50    printf "Noise at ampli input: summed %g, measured %g\r", res.SumNoise, res.MeasNoise
51  end // printer
52  function oneDetector(spectra, igram, rangeF, Temp, diffRes, dC, dSize, velocity, ampli, gain,
        filterL, lightF, nscans, nperiods, nBulk)
53    wave/D spectra, igram // parameters as in doDetector ...
54    string rangeF
55    variable Temp, diffRes, dC, dSize, velocity
56    string ampli
57    variable gain
58    string filterL, lightF
59    variable nscans, nperiods, nBulk
60    STRUCT Results res
61    string s
62    doDetector(spectra, igram, rangeF, Temp, diffRes, dC, dSize, velocity, ampli, gain, filterL,
          lightF, nscans, nperiods, nBulk, res)
63    StructPut/S res, s
64    print Temp, nameofwave(spectra), nameofwave(igram)
65    printer(s)
66  end // oneDetector
67  function doDetector(spectra, igram, rangeS, Temp, diffRes, dC, dSize, velocity, ampli, gain,
        filterList, lightS, nscans, nperiods, nBulk, res, [lightFac])
68    wave/D spectra, igram // from jdx (bruker)
69    string rangeS
70    variable Temp, diffRes, dC, dSize, velocity // [K], [ohm], [F], [um], [Hz]
71    string ampli // "EGG5113" or "SR570"
72    variable gain // gain or 'transimpedance' SR570
73    string filterList, lightS // stringlist with filter freqs |Z(nu)|
74    // and function "giving" light spectra as f(wavenumber) in [W m^-2 /cm^-1]
75    variable nscans, nperiods, nBulk // number of scans, refractive index of substrate
76    STRUCT Results &res
77    variable lightFac
78    if (paramIsDefault(lightFac))
79      lightFac=sqrt(.5) // standard: 45 degree configuration, surface shortenend
80    endif
81    FUNCREF one_argument_func rangeF=$rangeS, lightF=$lightS
82    if (char2num(filterList[0])>57) // not a +, - or 0..9
83      print "not a freq list: ", filterlist
84      print "do nothing, ABORT !"
85      return 1
86    endif
87    wave filterCoefs=$MakeFiltersCoefWave(filterList)
88    STRUCT Results results
89    string sname=nameofwave(spectra), iname=nameofwave(igram), stmp
90  // CURRENT (DENSITY):  transimpedance, ampl, size, /100
91    wave curr=$duplo(spectra,"curr","A m^-2 /{cm^-1}")
92    curr=spectra / (100 * (1e-6*dsize)^2 * cabs(filters(wn2nu(x,velocity),filterCoefs)))
93    // factor 100 bruker, size^2 ([um] -> [m]), [V m^-2 /{cm^-1}]
94    sprintf stmp "=%s / ( 100 * %g^2 * filters(wn2nu(x,%g),{%s})", sname, 1e-6*dSize, velocity,
          filterList
95    Note curr, stmp
96    strswitch(ampli)
97      case "EGG5113":
```

```
98      curr/=cabs( detAmpImpedance5113(wn2nu(x,velocity), 1/diffRes, dC, 1, 1) ) * gain  // [V]
            -> [A]
99      sprintf stmp "/=cabs( detAmpImpedance5113(wn2nu(x,%g), %g, %g, 1, 1) ) * %g )", velocity,
            1/diffRes, dC, gain
100     Note curr, stmp
101     break
102    case "SR570":
103     curr/=gain
104     sprintf stmp "/= %g", gain // [V] -> [A]
105     Note curr, stmp
106     break
107    default:
108     curr=NaN
109     sprintf stmp "'%s' is not a known amplifier, setting to NaN", ampli
110     note curr, stmp
111   endswitch
112 // RESPONSIVITY
113   wave Resp=$duplo(spectra,"Resp","A / W")
114   variable T_iface=4 * (1/nBulk)/(1+1/nBulk)^2
115   // lightF includes: blackbody, beamsplitter, window, FTIR-geometry, polarization
116   // but NOT! : geometry inside/at device (E_z, sqrt(2) shortend, interface-reflection
117   // and we leave it out (geometry inside), as it is already "contained" in absorption -> to
          compare!
118   Resp=curr(x) / ( T_iface * lightF(x) * lightFac )
119   sprintf stmp "= %s / (%g * %s(x) * %g)", nameofwave(curr), T_iface, lightS, lightFac
120   Note Resp, stmp
121 // ETA(OVERALL)
122   wave eta=$duplo(spectra,"eta","electrons / photon")
123   eta = Resp * (h_() * c_() / e_() * 100 * x) // *100: [cm-1] -> [m-1]
124   Note eta, "= Resp * h_ * e_ / (e_ * 100x) [wn in m-1]"
125   wave counter=$duplo(spectra,"counter","electrons")
126 // NOISE-SOURCES
127   // background in FTIR: noise due to 1/2 blackbody glowbar
128   counter = lightFac * T_iface * lightF(x) // [W m-2 /{cm-1}]
129   counter /= 2 // avg value after Beamsplitter
130   counter *= eta / (h_() * c_() * 100 * x) // electrons: [m-2 s-1 /{cm-1}]
131   counter *= isSet(rangeF(x), 1) // bit 1
132   results.NoiseEl_FTIR_2= integrateSkipNan(counter) // -> [m-2 s-1] = [( sqrt(Hz) / m ) ^2]
133   // background at 300 K:
134   counter = lightFac * T_iface * pi * K_wT(x, 300 ) / 2 // polarization, [W m-2 /{cm-1}]
135   counter *= eta / (h_() * c_() * 100 * x) // electrons: [m-2 s-1 /{cm-1}]
136   counter *= isSet(rangeF(x), 1) // bit 1
137   results.NoiseEl_BG_2= integrateSkipNan(counter) // -> [m-2 s-1] = [( sqrt(Hz) / m ) ^2]
138 // DETECTIVITIES
139   // D*_BLIP_300
140   wave DsBLIP=$duplo(spectra,"Ds_BLIP","jones")
141   DsBLIP = eta / (h_()*c_()*100*x * sqrt(2/nperiods*abs(results.NoiseEl_BG_2))) * 100
142     // -> [m W-1 sqrt(Hz)]; (*100: [m] -> [cm])
143   // D*_J and noise (one value!)
144   wave DsJ=$duplo(spectra,"DsJ","jones")
145   DsJ = Resp * sqrt(diffRes * (1e-6*dsize)^2 / (4*k_()*Temp) ) * 100 // *100 [m] -> [cm]
146   results.NoiseEl_J_2=4*k_()*Temp/(diffRes * (1e-6*dsize)^2 * e_()^2) // -> [( sqrt(Hz) / m )
          ^2]
147 // PEAK SEARCH ("abuse" wave counter)
148 // peaks in rangeF = 2: response, eta, D*_J, D*_BLIP
149   variable/C precise=0
150 // response
151   counter = isSet(rangeF(x), 2)?(Resp):(NaN); wavestats/Q counter;
152   if (0)
153     results.R_max=V_max; results.R_maxloc=V_maxloc
154   else
155     precise=fitlor(Resp,V_maxloc,50)
156     results.R_max=real(precise); results.R_maxloc=imag(precise)
157   endif
158 // quantum efficiency
```

```
159   counter = isSet(rangeF(x), 2)?(eta):(NaN); wavestats/Q counter;
160   if (0)
161     results.eta_max=V_max; results.eta_maxloc=V_maxloc
162   else
163     precise=fitlor(eta,V_maxloc,50)
164     results.eta_max=real(precise); results.eta_maxloc=imag(precise)
165   endif
166 // D* blip
167   counter = isSet(rangeF(x), 2)?(DsBLIP):(NaN); wavestats/Q counter;
168   if (1)
169     results.DsBLIP_max=V_max; results.DsBLIP_maxloc=V_maxloc
170   else
171     precise=fitlor(DsBLIP,V_maxloc,50)
172     results.DsBLIP_max=real(precise); results.DsBLIP_maxloc=imag(precise)
173   endif
174 // D* J
175   counter = isSet(rangeF(x), 2)?(DsJ):(NaN); wavestats/Q counter;
176   if (1)
177     results.DsJ_max=V_max; results.DsJ_maxloc=V_maxloc
178   else
179     precise=fitlor(DsJ,V_maxloc,50)
180     results.DsJ_max=real(precise); results.DsJ_maxloc=imag(precise)
181   endif
182 // RMS NOISE (measured)
183 // if igram not NULL: noise measured, compare with noise BLIP * J (current noise) igram * 10,
      /
184   if (waveExists(igram))
185     // "measure" noise
186     variable rms2=ig_rms(igram, 0.1, 4) * 10 // [10 V] -> [V]
187     rms2=rms2^2 * nscans // [V^2] (correct bw due to average over nscans)
188     // "sum" detector noise
189     variable noise2=results.NoiseEl_J_2 + results.NoiseEl_BG_2 + results.NoiseEl_FTIR_2
190       // [m-2 s-1] = [( sqrt(Hz) / m ) ^2]
191     noise2*=(1e-6*dsize * e_())^2      // [A^2 s] = [A^2 / Hz]
192     strswitch(ampli)
193       case "EGG5113":
194         wave Vn2=$duplo(spectra,"Vnoise2","V^2_Hz^-1")
195         Redimension/N=(freqpnts) Vn2
196         SetScale/I x 0,maxfreq,"Hz", Vn2
197         Vn2 = noise2 * cabs( detAmpImpedance5113(x, 1/diffRes, dC, 1, 1) )^2 // [V^2 / Hz]
198         Vn2 += (4e-9) ^2 // 4 nV /sqrt(Hz)
199         Vn2 *= cabs(filters(x,filterCoefs))^2
200         results.SumNoise=sqrt(integrateSkipNAN(Vn2)) // -> [V^2] -> [V]
201         // checked with qucs: this is ok if NoiseEL_BG, FTIR are correct!!
202         Vn2 *= gain^2
203         Vn2 = sqrt(Vn2)
204         results.MeasNoise=sqrt(rms2)/gain // [V] at amplifier input
205         printf "%s\t%g\t%g\t%g\tN\t%g\t%g\t%g\tM\t%g\r" nameofwave(spectra), temp-273.15,
                diffRes, gain, 1e-6*dsize*e_()*sqrt(results.NoiseEl_J_2), 1e-6*dsize*e_()*sqrt(
                results.NoiseEl_BG_2), 1e-6*dsize*e_()*sqrt(results.NoiseEl_FTIR_2), sqrt(rms2)
206         break
207       case "SR570":
208         wave An2=$duplo(spectra,"Anoise2","A^2_Hz^-1")
209         Redimension/N=(freqpnts) An2
210         SetScale/I x 0,maxfreq,"Hz", An2
211         An2=noise2 // [A^2 / Hz]
212         variable amplinoise=NaN
213         if ( fequal(gain,1/200e-12) )
214           amplinoise=60e-15 // 60fA/sqrt(Hz)
215         elseif (fequal(gain,1/100e-9) )
216           amplinoise=600e-15 // 600fA/sqrt(Hz)
217         elseif (fequal(gain,1/50e-9) )
218           amplinoise=600e-15 // 600fA/sqrt(Hz)
219         elseif (fequal(gain,1/20e-9) )
220           amplinoise=600e-15 // 600fA/sqrt(Hz)
```

```
221       endif
222       An2 += amplinoise^2
223       An2*=cabs(filters(x,filterCoefs))^2
224       results.SumNoise=sqrt(integrateSkipNAN(An2)) // -> [A^2 -> A]
225       results.MeasNoise=sqrt(rms2)/gain // [V -> A] at amplifier input
226       break
227     endswitch
228   endif
229   NVAR keep_level
230   variable keep=nvar_exists(keep_level)?(keep_level):(0)
231   if (keep < 3)
232     killwaves/Z counter, Vn2, An2
233   endif
234   if (keep < 2)
235     killwaves/Z DsBLIP, DsJ
236   endif
237   if (keep < 1)
238     killwaves/Z curr, eta
239   endif
240   res=results
241 end // doDetector
242 static function fequal(a,b) // float "equality" (in 1%)
243   variable a, b
244   return 0.01 > abs(1-a/b)
245 end // fequal
246 static function/S duplo(tmpl, pre, Dunit)
247   wave tmpl
248   string pre, Dunit
249   string name=possiblyQuoteName(pre+"_"+nameofwave(tmpl))
250   duplicate/O tmpl $name
251   wave copy=$name
252   SetScale d 0,0, Dunit, copy
253   copy=Inf
254   return name
255 end // duplo
256 static function/C fitlor(w, loc, wi) // more precise location with fit
257   wave w
258   variable loc, wi
259   variable V_fitoptions=4 // no progres window during fit
260   variable V_fiterror=0 // catch errors in fit, do not abort
261   variable fitC=x2pnt(w,loc)
262   CurveFit/N/Q/W=0 lor w[fitC-wi,fitC+wi] // +- wi points
263   if (V_fiterror) // if error in fit, return NaNs
264     Printf "␣FitError:␣%d,␣wave=%s␣@␣%g±%g\r", V_FitError, nameofwave(w), fitC, wi
265     return cmplx(NaN,NaN)
266   endif
267   wave W_coef
268   if (w_coef[3]<0)
269     Printf "␣B␣<␣0:␣wave=%s␣@␣%g±%g␣(%g,%g,%g,%g)␣->␣maxval=%g,␣loc=%g\r", nameofwave(w), fitC
          , wi, w_coef[0],w_coef[1],w_coef[2],w_coef[3], W_coef[0]+W_coef[1]/W_coef[3],W_coef[2]
270     return cmplx(NaN,NaN)
271   endif
272   fitC=W_coef[2] // new center
273   wi=sqrt(W_coef[3]*2) // new wi
274   CurveFit/N/Q/W=0 lor w(fitC-wi,fitC+wi) // +- width from 1. fit
275   if (V_fiterror) // if error in fit, return NaNs
276     Printf "␣FitError:␣%d,␣wave=%s␣@␣%g±%g\r", V_FitError, nameofwave(w), fitC, wi
277     return cmplx(NaN,NaN)
278   endif
279   return cmplx(W_coef[0]+W_coef[1]/W_coef[3],W_coef[2]) // (maxval, loc)
280 end // fitlor
281 static function ig_rms(w, range, f)
282   wave w
283   variable range, f
284   string ntmp=UniqueName("noiser", 1, 0)
```

```
285   duplicate w $ntmp
286   wave/D tmp=$ntmp
287   variable lastn=numpnts(tmp)
288   wavestats/Q tmp
289   tmp[x2pnt(tmp, (1-range)*V_maxloc ),x2pnt(tmp, (1+range)*V_maxloc)]=NaN // cut around
          centerburst
290   tmp[0,range*lastn]=NaN // cut ends
291   tmp[(1-range)*lastn,lastn-1]=NaN
292   wavestats/Q tmp
293   do
294     lastn=V_npnts // non-NaNs
295     tmp=abs(tmp-V_avg)> f * V_rms?(NaN):(tmp-V_avg) // cut spikes, offset linear average
296     wavestats/Q tmp
297   while (V_npnts < lastn)
298   killwaves tmp
299   return V_rms
300 end // ig_rms
301 static function integrateSkipNan(w)
302   wave/D w
303   string ntmp=UniqueName("integrator", 1, 0)
304   duplicate w $ntmp
305   wave/D tmp=$ntmp
306   tmp=(numtype(tmp)==2)?0:(tmp)
307   integrate tmp
308   variable inte=tmp[numpnts(tmp)-1]-tmp[0]
309   killwaves tmp
310   return inte*sign(deltax(w))
311 end // integrateSkipNan
312 static function isSet(val, mask)
313   variable val, mask
314   return mask == (val & mask)
315 end // isSet
316 static function nu2wn(x, vel)
317   variable x, vel
318   return HeNe * x / vel
319 end // nu2wn
320 static function wn2nu(x, vel)
321   variable x, vel
322   return vel * x / HeNe
323 end // wn2nu
324 // == amplifiers
325 static constant ampC=25e-12 // 25 pF
326 static constant ampR=10e6 // 10 Mohm
327 static constant ACmodeC=.1e-6 // 0.1 muF
328 function/C detAmpImpedance5113(nu, detG, detC, ac, ampT) // ampT = 1 or 10 sec
329   variable nu, detG, detC, ac, ampT
330   // nu = electrical frequency, detG = detector conductance (1/R), detC = detector capacity
331   // ac = 1 for AC coupling, 0 for DC coupling, ampT = 1 or 10 s (corresponds to 10 or 100 Mohm
          )
332   variable omega=2*pi*nu
333   variable/C ampZ_1=cmplx((ampT*ampR)^-1,omega*ampC)
334   variable/C ACmodeZ=ac?cmplx(0,omega*ACmodeC)^-1:0
335   variable/C detZ_1=cmplx(detG,omega*detC)
336   return (ampZ_1+detZ_1*(ACmodeZ*ampZ_1+1))^-1
337 end // detAmpImpedance5113
338 // == 6dB filters (per octave, =10dB per decade)
339 function/C highpass1Hz(nu)
340   variable nu
341   return cmplx(0, nu)/cmplx(1,nu) // omega -> nu, 3dB at 1Hz
342 end // highpass1Hz
343 function/C lowpass1Hz(nu)
344   variable nu
345   return cmplx(1, 0)/cmplx(1,nu) // omega -> nu, 3dB at 1Hz
346 end // lowpass1Hz
347 function/C filters(nu,fwave)
```

```
  variable nu
  wave fwave
  variable/C res=1
  variable i,n=numpnts(fwave)
  for (i=0;i<n;i+=1)
    if (fwave[i]<0) // negative = highpass (lower limit..)
      res*=highpass1Hz(-nu/fwave[i])
    else // positive = lowpass (higher limit ..)
      res*=lowpass1Hz(nu/fwave[i])
    endif
  endfor
  return res
end // filters
function/S MakeFiltersCoefWave(flist)
  // use as: filters(x, $filtersCoefWave("-30;+30e3;200e3;") )
  string flist
  string wname=packagesbase+":"+possiblyQuoteName(cleanupname("filterCoefs_"+flist,1))
  wave/Z coefs=$wname
  if (!waveExists(coefs))
    NewDataFolderP(packagesbase)
    variable i, n=ItemsInList(flist)
    make/O/N=(n)/D $wname
    wave coefs=$wname
    coefs=str2num(StringFromList(p,flist))
  endif
  return wname
end // MakeFiltersCoefWave
function/C filtersL (nu, flist)
  variable nu
  string flist
  return filters(nu,$MakeFiltersCoefWave(flist))
end // filtersL
static strconstant velos="16e2;22e2;3e3;4e3;5e3;75e2;1e4;2e4;4e4;6e4;8e4;1e5;13e4;14e4;15e4;16
    e4"
function BrukerVelo(i)
  variable i
  return str2num(StringFromList(i,velos))
end // BrukerVelo
function/S StringByKey_SFjdx(w, key)
  wave w
  string key
  return StringByKey(key,StringByKey_jdx(w,"SampleForm"),"=",";")
end // StringByKey_SFjdx
function/S StringByKey_jdx(w, key)
  wave w
  string key
  return ReplaceString("\r",StringByKey(key, note(w), "=", "#"),"")
end // StringByKey_jdx
```

Listing C.2: MG_Planck.ipf: blackbody emissivity (Planck)

```
#pragma rtGlobals=1 // Use modern global access method.
#include "MG_constants"
function K_nT(nu, T) // Frequency & Temperature [Hz, K] > [W s/m^2]
  variable nu, T
  return 2 * h_() * nu^3 /c_()^2 / (exp(h_()*nu/(k_()*T))-1)
end // K_nT
function K_lT(lambda, T) // Wavelength & Temperature [m, K] > [W/m^3]
  variable lambda, T
  return 2*h_()*c_()^2/lambda^5/(exp(h_()*c_()/(k_()*T*lambda))-1)
end // K_lT
function K_wT(wavenumber, T) // Wavenumber & Temperature [1/cm, K] > [W cm/m^2]
  variable wavenumber, T
  return 2*h_()*1e8*wavenumber^3*c_()^2/(exp(h_()*c_()*100*wavenumber/(k_()*T))-1)
end // K_wT
```

```
15 function partOfPi(azimutalAngle)
16   variable azimutalAngle
17   return pi/2*(1-cos(2*azimutalAngle))
18 end // partOfPi
```

C.2 Transport calculations

Igor {igor} procedure implementing the calculation of the electronic transition rates between the different states in a QCD and the simulation of the electrical resistance using these rates, as described in section 5.2

Listing C.3: MG_phonons.ipf: transition rates and more

```
#pragma rtGlobals=1 // Use modern global access method.
// algorithm parameters
static constant angle_steps=99
static constant energy_steps=57
static constant energy_inverse_offset=1
static constant fermi_decay_limit=1e-8
static constant cross_psi_decay_limit=1e-10
static constant interpol_subsample=10
static constant interpol_min=8192
static constant interpol_max=32768
function scaleForIntegral(wavefunction, [newdx, newx0])
  wave wavefunction
  variable newdx, newx0
  variable x0, dx
  newdx=ParamIsDefault(newdx) ? 1e-10 : newdx
  x0=leftx(wavefunction)
  dx=deltax(wavefunction)
  newx0=ParamIsDefault(newx0) ? newdx/dx*x0 : newx0
  SetScale/P x newx0, newdx, "m", wavefunction
  wavefunction*=sqrt(dx/newdx)
end // scaleForIntegral
static function minmax(minv, maxv)
  variable &minv, &maxv
  if (minv > maxv)
    variable tmp=minv
    minv=maxv
    maxv=tmp
  endif
end // minmax
static function/S copy_tmp(w, [prefix, npts]) // utility function to create/copy temporary
    waves
  wave/Z w
  string prefix
  variable npts
  if (paramIsDefault(prefix))
    prefix="ZZZZZZZZ"
  endif
  string scount=prefix+"_TMP"
  if (WaveExists(w))
    duplicate/O w, $scount
  else
    Make/O/D/N=(ParamIsDefault(npts)?128:(npts)) $scount
  endif
  return scount
end // copy_tmp
// plausibility check on wavefunctions scaling (approx. 1)
// due to nonparabolicity, scaling can be different from 1
function checkWFCT(psi)
  wave psi
  variable val, ret=1
  duplicate/O psi, TMP
  TMP*=TMP
  val=area(TMP)
  if (val<0.8 || val>1.25) // well, say between 0.8 and 1.25
    print nameofwave(psi), area(TMP)
    ret=0
```

```
  endif
  return ret
end // checkWFCT
function/S productWavefunctions(wf1, wf2)
  wave wf1, wf2
  variable left1, left2, right1, right2, delta1, delta2, left, right, nn
  string u1, u2, n1, n2, wn
  left1=leftx(wf1); right1=rightx(wf1); delta1=deltax(wf1)
  n1=nameofwave(wf1);u1=WaveUnits(wf1, 0)
  left2=leftx(wf2); right2=rightx(wf2); delta2=deltax(wf2)
  n2=nameofwave(wf2);u2=WaveUnits(wf2, 0)
  if (! (checkWFCT(wf1) && checkWFCT(wf2) ) )
    printf "Integrals over psi^2 for %s or %s are not close to 1\r", n1, n2
    return ""
  endif
  if (cmpstr(u1, u2)) // not equal strings
    printf "wave x units do not match: %s [%s], %s [%s]\r", n1, u1, n2, u2
    return ""
  endif
  minmax(left1, right1); minmax(left2, right2)
  left=max(left1, left2)
  right=min(right1, right2)
  nn=(right-left)/min(abs(delta1), abs(delta2))
  nn=max(0, ceil(nn))
  sprintf wn, "%s_X_%s", n1, n2
  make/D/O/N=(nn) $wn
  wave prod=$wn
  SetScale x left, right, u1, prod
  prod=wf1(x)*wf2(x)
  return wn
end // productWavefunctions
function trimTails(wavefunction, [relLimit])
  wave wavefunction
  variable relLimit
  relLimit=ParamIsDefault(relLimit) ? 0 : relLimit
  variable newx0, newdx
  wave psi_2=$copy_tmp(wavefunction, prefix="PSI_SQUARED")
  psi_2=wavefunction^2
  Integrate psi_2
  psi_2/=psi_2[dimsize(psi_2,0)-1]
  FindLevel/Q/P psi_2 1-relLimit
  if (!V_flag)
    DeletePoints V_LevelX, inf, wavefunction
  endif
  FindLevel/Q/P psi_2 relLimit
  if (!V_flag)
    newdx=deltax(wavefunction)
    newx0=pnt2x(wavefunction,V_LevelX)
    DeletePoints 0, V_LevelX, wavefunction
    SetScale/P x newx0, newdx, wavefunction
  endif
end // trimTails
function circIntegral(ki2kf2, m2kikf, deltaz)
  variable ki2kf2, m2kikf, deltaz
  wave/D/Z circ=CIRC_INTEGRAND_TMP
  if (!WaveExists(circ) || numpnts(circ)!=angle_steps)
    print "create circ:"
    wave/D circ=$copy_tmp($"", prefix="CIRC_INTEGRAND", npts=angle_steps)
  endif
  if ( abs(rightx(circ)-pi) > 1e-5)
    print "setScale on circ"
    SetScale x 0, pi, circ
  endif
  circ=sqrt(ki2kf2 + m2kikf * cos(x) ) // Q
  circ=exp(-circ*deltaz)/circ
```

```
121   return 2 * area(circ)
122 end // circIntegral
123 function Sij(Xij, E0i, E0f, E, ab) // Energies in [J]
124   wave Xij
125   variable E0i, E0f, E, ab
126   variable E_i, E_f, ki2, kf2, ki2kf2, m2kikf
127   variable res
128   E_i = E - E0i
129   ki2 = 2*meff()/h_bar()^2 * E_i
130   E_f = E - E0f + eLO() * (ab?(1):(-1))
131   kf2 = 2*meff()/h_bar()^2 * E_f
132   if (ki2 < 0 || kf2 < 0)
133     printf "//␣ki2␣or␣kf2␣<␣0,␣return␣NaN␣for␣Sij:␣ki2=%g,␣kf2=%g\r", ki2, kf2
134     return NaN
135   endif
136   ki2kf2=ki2+kf2
137   m2kikf=-2*sqrt(ki2*kf2)
138   variable p0, dx, n1
139   dx=deltax(Xij)
140   wave integ=$copy_tmp(Xij, prefix="INTEGRAL")
141   SetScale/P x 0, dx, integ // !: so that scale after correlate is centered
142   Correlate Xij, integ
143   p0=x2pnt(integ,0)
144   n1=numpnts(integ)-1
145   integ[p0,n1]*=circIntegral(ki2kf2, m2kikf, x) // do calc
146   integ[0,p0-1]=integ[n1-p] // mirror
147   res = area(integ)*dx
148   return meff() *e_()^2 *eLO() / (h_bar()*2*h_bar()^2) / eps_p() *res
149 end // Sij
150 static function meff() // in [Kg]
151   NVAR meff
152   return meff * m_e()
153 end // meff
154 static function eLO() // in [J]
155   NVAR eLO
156   return eLO * e_()
157 end // eLO
158 function eps_p()
159   // eps_inf, eps_static
160   NVAR kp0
161   return kp0 * eps_0()
162 end // eps_p
163 function nBOSE(T)
164   variable T
165   return 1/( exp(eLO()/(k_()*T)) - 1 )
166 end // nBOSE
167 function fFERMI(Ef, E,T) // [J], [K]
168   variable Ef, E, T
169   return 1/(exp( (E-Ef)/(k_()*T) ) + 1)
170 end function // fFERMI
171 function fermiINT(Ef, T, E0)
172   variable Ef, T, E0
173   return k_()*T*ln( 1 + exp( (Ef-E0)/(k_()*T) ) )
174 end // fermiINT
175 // determine chemical potential Ef (Fermi Energy)
176 function findEf(Es, n2D, T) // Es in [J], n2D in [m-2], T in [K] --> Ef in [J]
177   wave Es
178   variable n2D, T
179   variable Ef, Emax, Emin, Ediff, su
180   duplicate/O Es, densi
181   Emax=wavemax(Es)
182   Emin=wavemin(Es)
183   Ediff=Emax-Emin
184   Ef=0.5 * (Emax+Emin)
185   Emax=wavemin(Es) // +Ediff -> max ...
```

```
186   do
187     Emax+=Ediff
188     densi = D2D() * fermiINT(Emax, T, Es)
189   while (sum(densi) < n2D)
190   Emin=wavemax(Es) // -Ediff -> min
191   do
192     Emin-=Ediff
193     densi = D2D() * fermiINT(Emin, T, Es)
194   while (sum(densi) > n2D)
195   do
196     Ef= 0.5 * (Emax+Emin)
197     densi = D2D() * fermiINT(Ef, T, Es)
198     su=sum(densi)
199     if (su > n2D)
200       Emax=Ef
201     else
202       Emin=Ef
203     endif
204   while ( abs(su-n2D) > 1e-6 * n2D)
205   return Ef
206 end // findEf
207 function D2D()
208   return meff()/(pi*h_bar()^2)
209 end function // D2D
210 // calculate Gij(T): absorption/emission rate including thermal occupations
211 function Gij(Xi, Xj, E0i, E0j, Ef, T, [ab, base]) // Energies in [J], T in [K]
212   wave Xi, Xj
213   variable E0i, E0j, Ef, T, ab
214   string base
215   variable E_LO=eLO(), Emin, Emax, hwlo, nopt, val, fact
216   variable c, d, dm
217   ab=paramIsDefault(ab)?0:(ab?1:0)
218   base=SelectString(paramIsDefault(base) , base, "")
219   string sXij=productWavefunctions(Xi, Xj)
220   Wave Xij=$sXij
221   trimTails(Xij, relLimit=cross_psi_decay_limit)
222   // set sign of hwlo & phonon population for absorption/emission
223   if (ab) // absorption
224     hwlo=E_LO
225     nopt=nBOSE(T)
226   else // emission
227     hwlo= - E_LO
228     nopt=1+nBOSE(T)
229   endif
230   // set Integral limits - lower = max(E0i,E0j +- h* wLO)
231   Emin=E0j - hwlo
232   Emin=max(Emin, E0i)
233   // upper limit: ramp up until product of fermi factors is smaller than decay limit
234   Emax=Emin
235   fact=0
236   do
237     fact=max(fact,fFERMI(Ef, Emax, T) * (1-fFERMI(Ef, Emax+hwlo, T)))
238     Emax+=E_LO
239   while ( fFERMI(Ef, Emax, T) * (1-fFERMI(Ef, Emax+hwlo, T)) > fact * fermi_decay_limit)
240   // create energy & value waves, if not existing
241   string seng, sval
242   sprintf seng, "%s_%s_E%s", base, sXij, SelectString(ab , "em", "abs")
243   sprintf sval, "%s_%s_S%s", base, sXij, SelectString(ab , "em", "abs")
244   wave/Z weng=$seng, wval=$sval
245   if (WaveExists(weng) && waveexists(wval) && numpnts(weng)==numpnts(wval))
246     // exists, do not repeat calculation, just get number of points (for interpolation below)
247     dm=numpnts(weng)
248   else
249     // does not exist, create some "clever" energy value list (no linearly sampled, but denser
          for
```

```
250     // low energies, sparse for higher)
251     dm=energy_steps
252     // first create temporary waves with temporary names, in case we interupt: then the
            incomplete
253     // set would be take as given in the next run
254     wave weng=$copy_tmp($"", prefix="ENERGIES", npts=dm)
255     wave wval=$copy_tmp($"", prefix="VALUES", npts=dm)
256     weng=1/(energy_inverse_offset+dm-p) - 1/(energy_inverse_offset+dm) // "hyperbolic" energy
            list
257     weng*=wavemax(weng) // scale
258     weng=Emin + (Emax-Emin)*weng // and offset
259     wval=nan
260     // and calculate Sij (=values) for all the energies
261     printf "\t// %s ", SelectString(ab , "emission", "absorption")
262     for(d=0;d<dm;d+=1)
263       c=0
264       do
265         val=Sij(Xij, E0i, E0j, weng[d], ab) // get/calculate Sij(E)
266         if (numtype(val)==2) // nan, if result was nan, the energy is a tiny little bit to small
267           c = c ? c : 17 // (numerical), so on of the ks is not existant, just add a little bit
268           c -= 2 // (normally the first step, 1e-15 eV is enough (=double prec.),
269           printf "%X", c // but who knows - in other words, if we see more than just an F
270           weng[d]+=e_() * 10^(-c) // (hex for 15), there might be a serious problem
271         endif
272       while (numtype(val)==2 && c > 4)
273       printf "."
274       wval[d]=val
275     endfor
276     // now that everything went through, rename the wave to their destination
277     Rename weng, $seng
278     Rename wval, $sval
279     wave weng=$seng, wval=$sval
280     // and save the experiment, in case of crashes not everything will be lost
281     SaveExperiment
282     printf "\r"
283   endif
284   // to do the integral with rapidly changing fermi factors, interpolate to a large,
          equidistantly sampled wave
285   variable nint
286   nint=(weng[dm-1]-weng[0])/(weng[1]-weng[0]) // minimal resolution according to the closes
          points we have
287   nint=max(nint*interpol_subsample,interpol_min) // and minimal as set
288   nint=min(nint,interpol_max) // but limit to maximum number
289   nint=round(nint)
290   // interpolate
291   Interpolate2/T=3/N=(nint)/F=0/Y=VALUES_INTERPOLATED_TMP weng, wval
292   wave VALUES_INTERPOLATED_TMP
293   // multiply with fermifactors
294   VALUES_INTERPOLATED_TMP *= fFERMI(Ef, x, T) * (1-fFERMI(Ef, x+hwlo, T))
295   // and get the integral, times constant factors (bose, 2D density)
296   variable res= area(VALUES_INTERPOLATED_TMP, Emin, inf) * nopt * D2D()
297   return res
298 end // Gij
299 //total G, abs+em
300 function calcG(temp, ibase, obase, ebound, ebound_fermi, aa, bb)
301   variable temp // temperature [K]
302   string ibase, obase // in and out "base" names
303   wave ebound, ebound_fermi // state energies [Ev]!!, sublist for Ef determination [J]
304   string aa, bb // initial / final state "names" (as label in ebound)
305   variable ef, i, a, b, ge, ga, k, kn, sumG=0
306   NVAR dop2D
307   Ef=findEf(ebound_fermi, dop2D, temp) // get Ef, Ef in [J]
308   // get indexes of both states
309   a=FindDimLabel(ebound,0,aa)
310   b=FindDimLabel(ebound,0,bb)
```

```
311   // construct wave names, e.g "wavefct4"
312   wave wa=$(ibase+num2str(a))
313   wave wb=$(ibase+num2str(b))
314   // and calculate Gij emission and Gij absorbtion
315   ge= Gij(wa, wb, Ebound[a]*e_(), Ebound[b]*e_(), Ef, temp, ab=0, base=obase) // Gij needs all
          energies in [J]: * e_()
316   ga= Gij(wa, wb, Ebound[a]*e_(), Ebound[b]*e_(), Ef, temp, ab=1, base=obase)
317   //return summed Gij
318   return ge+ga
319 end // calcG
320 function generateR0A(invT, ibase, baseT, ebound, ebound_fermi, list, Nperiods)
321   wave invT // list of inverse temperatures
322   string ibase // base name
323   variable baseT // base temparature (for max energy determination) in Sij calculation
324   wave ebound, ebound_fermi // list of all energies, and sublist of those used in Ef
          determination
325   string list // list of transitions to include, e.g. "A7 B3;A7 B2;..." where A7, B3, ... are
          labels in ebound
326   variable Nperiods // number of periods in structure
327   string aa, bb, obase
328   variable k, kn
329   string pair, gn, rn, formu
330   sprintf obase, "S%03g", baseT
331   kn=ItemsInList(list)
332   sprintf rn, "R0A_%s", obase
333   NVAR dop2D, meff, eL0, kp0
334   printf "generateR0A: dop2D=%g m-2, eL0 = %g eV, meff = %g, kp0=%g, obase=%s\r", dop2D, eL0,
          meff, kp0, obase
335   duplicate/O invT $rn
336   wave rw=$rn
337   rw=0
338   SetScale d 0,0,"ohm cm2", rw
339   // setup graph
340   display/R rw vs invT
341   string win=StringFromList(0,WinList("*", ";", "WIN:1" ))
342   print win
343   ModifyGraph rgb($ nameofwave(rw))=(0,65280,0)
344   // for all pairs
345   for(k=0;k<kn;k+=1)
346     pair=StringFromList(k,list)
347     sscanf pair, "%s_%s", aa, bb
348     sprintf gn, "G%s%s_%s", aa, bb, obase
349     if (waveExists($gn)) // already calculated, just set wave reference
350       wave gw=$gn
351     else // do the calculation of all Gij
352       duplicate/O invT $gn
353       wave gw=$gn
354       printf "// %s %s ", aa, bb
355       calcG(baseT, ibase, obase, ebound, ebound_fermi, aa, bb) // calculate G at base
            temperature, this calculates all Sij waves
356       gw=calcG(1/invT, ibase, obase, ebound, ebound_fermi, aa, bb) // and now at all the
            temperatures (gw is a wave, implicit for loop)
357       SetScale d 0,0,"m-2 s-1", gw
358     endif
359     // add this G to graph
360     Dowindow/F $win
361     appendtograph gw vs invT
362     ModifyGraph log(left)=1,tickUnit(left)=1,log(right)=1,tickUnit(right)=1
363     Label left "\\U"
364     Label right "\\U"
365     // and add to sum of Gs
366     if (k==0)
367       formu=gn
368       rw=gw
369     else
```

```
370       sprintf formu, "%s+%s", formu, gn
371       rw+=gw
372     endif
373     doupdate
374   endfor
375   // calculcate R0A from sum of Gij
376   sprintf formu, "k_()/%s/(e_()^2*(%s))*%g*1e4", nameofwave(invT), formu, Nperiods // 1e4: [m
          ^2] -> [cm^2]
377   rw=k_()/invT/(e_()^2*rw)*Nperiods*1e4
378   SetFormula rw, formu
379   Note rw, formu
380   printf "\r%s:=%s\r", nameofwave(rw),formu
381 end // generateR0A
382 // load sewself result files
383 function loadSEWSELF(path)
384   string path
385   string wn, psin
386   variable i, in, xL, xE, val
387   LoadWave/G/O/D/A path+"Ebound.dat"
388   duplicate/O $ StringFromList(0,S_wavenames) Ebound, Ebound_fermi
389   Ebound_fermi*=e_() // convert Ebound_fermi to [J] (ebound is in [eV]
390   killwaves $ StringFromList(0,S_wavenames)
391   LoadWave/G/O/D/A path+"pse.dat"
392   in=ItemsInList(S_wavenames)
393   for(i=0;i<in;i+=1)
394     wn=StringFromList(i,S_wavenames)
395     sprintf psin, "wavefct%d", i
396     duplicate/O $wn, $psin
397     killwaves $wn
398     wave psi=$psin
399     scaleForIntegral(psi)
400     SetDimLabel 0,i,$psin,Ebound
401     checkWFCT(psi)
402   endfor
403   edit Ebound.ld, Ebound_fermi
404 end // loadSEWSELF
```

C.3 Package sewcalc

The *Objective Caml* {ocaml} sourcecode of the package sewcalc, a wrapper - controller around buildmatself and sewself which calculates and outputs the calculations described in equations 5.6-5.8. Although routines were mainly used for the structure N538 (and to some extent N516), the code is kept fairly general and it will be rather simple to add some structure descriptions to the module `Files` C.3.5 for other devices. The package can either be used as standalone program, described in C.3.9, or the different modules can be loaded into the toplevel OCaml interpreter, allowing interactive access to all the data structures and functions listed in the module describtion sections below.

C.3.1 Module Type : data types and structure definitions

```
type transLower = {
  lower : int ;
  z : float ;
  energy : float ;
  rate_em : float ;
  rate_abs : float ;
  integral : float ;
  rate_0 : float ;
  rate_T : float ;
}
```

Data structure storing the values per upper / lower state pair: the number of the lower state: `lower`, the `z` integral, the transition `energy` between the upper and lower state, the emission and absorption rates: `rate_em`, `rate_abs` and the overall rate from upper → lower state at $T = 0\,\mathrm{K}$: `rate_0` and the simulation temperature: `rate_T`.

```
type transUpper = {
  upper : int ;
  lowers : transLower list ;
  rate_total : float ;
}
```

Data structure per `upper` state, contains list of all possible lower states `lowers` and the total rate `rate_total` out of the upper state.

```
type transTemp = {
  temperature : float ;
  uppers : transUpper list ;
  energies : float array ;
  densities : float array ;
  wavefunctions : float array array ;
}
```

Data structure for one complete simulation at one `temperature`, with a list of all upper states `uppers`, the `energies` of all states as well as their occupation `densities` and `wavefunctions`.

```
type transTempList = transTemp list
```

List of transTemps, used to store a whole temperature series simulations

```
val marshout : ?append:bool -> string -> transTempList -> unit
```

`marshout dataList fileName` saves (marshals "out") the `dataList` to `fileName` (binary).

```
val marshin : string -> transTempList
```

`marshin fileName` reads (marshals "in") from `fileName` (binary), reading all chunks up to EOF.

```
val jsonout : string -> transTempList -> unit
```

`jsonout dataList fileName` saves the `dataList` to `fileName` (JSON text format).

```
val jsonin : string -> transTempList
```

`jsonin fileName` reads from `fileName` (JSON).

```
val varout : string -> transTempList -> unit
```

`varout dataList fileName` saves the `dataList` to `fileName` (JSON for .json & .txt, binary otherwise).

```
val varin : string -> transTempList
```

`varin fileName` reads from `fileName` (JSON for .json & .txt, binary otherwise).

Listing C.4: type.ml

```
let currentVersion : string = "transTemp 1.0.0"
(* has Marshal.data_size = 16*)
(* use the json-static syntax extension to get automatic
textual representation of data structures "for free" *)
type json transLower =
  { lower : int;
    z : float;
    energy : float;
    rate_em : float;
    rate_abs : float;
    integral : float;
    rate_0 : float;
    rate_T : float }
and transUpper = { upper : int; lowers : transLower list; rate_total : float }
and transTemp =
  { temperature : float;
    uppers : transUpper list;
    energies : float array;
    densities : float array;
    wavefunctions : float array array }
and transTempList = transTemp list
let first_data_size fd =
  let buf = "01234567890123456789012345678901234567890123456789"
  and _ = seek_in fd 0 in
  let _ = input fd buf 0 40 in seek_in fd 0; (*set pointer back to beginning of file*)
  Marshal.data_size buf 0
let marshout ?(append = false) fn ll =
  let perms = 0o666
  and flags =
```

```
   [if append then Open_append else Open_trunc; Open_wronly; Open_creat;
    Open_binary]
  in
  let fd = open_out_gen flags perms fn in
  if not append then Marshal.to_channel fd currentVersion [];
  Marshal.to_channel fd (ll : transTempList) [];
  close_out fd
and marshin fn =
  let rec mi fd =
    try let (t : transTempList) = Marshal.from_channel fd in t @ mi fd with
      End_of_file -> close_in fd; []
  in
  let fd = open_in_bin fn in
  let vers =
    if first_data_size fd = 16 then (Marshal.from_channel fd : string) else ""
  in
  print_endline vers;
  match vers with
    "" -> mi fd
  | v when v = currentVersion -> mi fd
  | _ -> close_in fd; raise (Failure "unknown file format")
(* use of the automagically created json_of_X and X_of_json functions*)
let jsonout fn ll =
  Json_io.save_json ~compact:true fn (json_of_transTempList ll)
and jsonin fn = transTempList_of_json (Json_io.load_json fn)
(* in/out functions with format choice based on file ending*)
let jsonsuffix fn =
  Str.string_match (Str.regexp ".*\\.\\(json\\|txt\\)$") fn 0
let varout fn = if jsonsuffix fn then jsonout fn else marshout fn
and varin fn = if jsonsuffix fn then jsonin fn else marshin fn
(* one-line multi-format converter:
varout "outfile.bin" (varin "infile.json") *)
```

C.3.2 Module Paths : generalised path handling

The functions in this module allow the creation of all possible paths through a given list of nodes - or omitting some of these. In the same manner a list of node-hops for such a given path can be created, and finally a "pair-function" can be applied on all these hops, multiplied (generalised) along the path and summed (generalised) over all possible path for the given nodelist.

```
val explodePath : 'a list -> 'a list list
```

`Paths.explodePath nodeList` takes list of nodes `nodeList` and returns all possible "forward" paths as list of node lists.

```
val allPairs : 'a list -> ('a * 'a) list
```

`Paths.allPairs nodeList` takes list of nodes `nodeList` and returns all "forward" pairings as list of pairs.

```
val applyPair : ('a -> 'a -> 'b) -> 'a list list -> 'b list list
```

For each Transitions step in the inner node Lists, `Paths.applyPair pairFunction nodeListList` applies pairFunction, composing a list of the results. This is done for all elements of the outer list, finally returning in a list of step results list.

```
val contractGeneral :
  ('a -> 'b -> 'a) -> 'a -> ('c -> 'a -> 'c) -> 'c -> 'b list list -> 'c
```

Nested double "fold_left" on internal and external Operator & Constant, respectively: `contractGeneral iOper iCons eOper eCons aListList`

```
val contract : float list list -> float
```

Specialisation thereof, internal product (1.0), external sum (0.0)

```
val sumPaths : ('a -> 'a -> float) -> 'a list list -> float
```

Multiply `pairF` along all possible pathes along `nodeListList`, sum it up.

Listing C.5: paths.ml

```
let explodePath nodeL =
  let res = ref [] in
  let rec slider le ri =
    match le, ri with
      left, [] -> res := left :: !res
    | left, [e] -> res := (left @ [e]) :: !res
    | left, hd :: tl ->
       let rest = List.map (fun e -> List.filter (fun x -> x <= e) tl) tl in
       List.iter (fun li -> slider (left @ [hd]) li) rest
  in
  slider [] nodeL; !res (* nodeLL *)
let allPairs nodeList =
  let nodeArray = Array.of_list nodeList in
  let res = ref [] in
  Array.iteri
    (fun upi up ->
      Array.iteri (fun lowi low -> if lowi < upi then res := (up, low) :: !res)
```

```
      nodeArray)
   nodeArray;
  !res
let applyPair pairFunc nodeLL =
  let rec product nodeL =
    match nodeL with
      [] -> []
    | [a] -> []
    | a :: b :: tl -> pairFunc a b :: product (b :: tl)
  in
  List.map product nodeLL (* transLL *)
let contractGeneral iOP iC eOP eC transLL =
  let iList = List.map (fun path -> List.fold_left iOP iC path) transLL in
  List.fold_left eOP eC iList
let contract transLL = contractGeneral ( *. ) 1. ( +. ) 0. transLL
let contractList transLL =
  contractGeneral (fun a b -> a @ [b]) [] (fun a b -> a @ [b]) [] transLL
let sumPaths pairF nodeLL = contract (applyPair pairF nodeLL)
```

C.3.3 Module `Creator` : create input files for sewself

```
val buildmatself : ?exe:string -> string -> unit
```

`buildmatself ~exe:"/path/to/buildmatself" file` just runs buildmatself on the structure definition `file`, optionally with the path to buildmatself in the labeled argument `~exe`

```
val create :
  ?efield:float ->
  ?coef:float -> string -> (string * float * float) list -> unit
```

`create ~efield:field ~coef:stretch_coef filename layer_list` write an input file for buildmatself, with optional bias voltage (defined by `field`) and a stretching coefficient to the file specified by `filename`

```
val create_mat_par : unit -> unit
```

create standard "mat.par"

```
val create_sewself_par : int -> unit
```

create standard "sewself.par", with given state number marked as upper

Listing C.6: creator.ml

```
open UnixLabels
let basepath = "/home/marcel/local/bin/"
let buildmatself ?(exe = basepath ^ "buildmatself") filename =
  match system (exe ^ " " ^ filename) with
    WEXITED i -> Printf.printf "EXITED: %u\n" i
  | WSIGNALED i -> Printf.printf "SIGNALED: %u\n" i
  | WSTOPPED i -> Printf.printf "STOPPED: %u\n" i
let barr ?(dot = 0.) (thick : float) = "AlInAs", thick, dot
let well ?(dot = 0.) (thick : float) = "GaInAs", thick, dot
let create ?(efield = 0.) ?(coef = 1.) filename layerL =
  let fd = open_out filename in
  try
    Printf.fprintf fd "%8.6f  efield
0.00  diffusion length
%u  numbers of layers
7 nbswellinjector
1 nbswellactive
%.2f  coeff (longer/shorter structure)
switches for structure length
0 injector only
0 active only
1 one period
material parameters
3 matchoice
GaInAs    0    0
AlInAs   0.52   1
end discretisation of the potential
10  inc0
20  incw
Structure, starting with a (phony) barrier (mat, thick (A), doping (1e18 cm-3)"
     efield (List.length layerL) coef;
   List.iter (fun (m, t, d) -> Printf.fprintf fd "\n%s %G  %G" m t d)
     (barr 50. :: layerL);
   close_out fd
  with
```

```
   exc ->
    begin try close_out fd with
      _ -> ()
    end;
    raise exc
let rec rotate n l =
 if n < 0 then List.rev (rotate (- n) (List.rev l))
 else
  let len = List.length l in
  let off = n mod len in
  let a = Array.of_list l in
  Array.to_list (Array.sub a (len - off) off) @
   Array.to_list (Array.sub a 0 (len - off))
let n538 =
 [well 59. ~dot:0.4; barr 60.; well 17.; barr 44.; well 20.; barr 39.; well 23.;
  barr 37.; well 27.; barr 35.; well 32.; barr 31.; well 39.; barr 28.; well 47.;
  barr 26.]
let create_N538 filename efield coef = create ~efield:efield ~coef:coef filename n538
let n516 =
 [well 100. ~dot:0.5; barr 47.; well 39.; barr 25.; well 43.; barr 19.; well 54.;
  barr 16.; well 66.; barr 17.]
let create_N516 filename efield coef = create ~efield:efield ~coef:coef filename n516
let create_mat_par () =
 let fd = open_out "mat.par" in
 try
  Printf.fprintf fd "mwell, mbarrier, Gamma, hlo, kp0
0.067  0.15  4.9e-19  0.036  71.3 GaAs/AlGaAs
0.0427 0.08  1.13e-18 0.032   64   GaInAsP system
0.0427 0.076 1.13e-18 0.032   64   AlInAs/GaInAs
0.035  0.082 1.53e-18 0.034   64   AlInAs/GaInAs 1%% strain
0.039  0.081 1.31e-18 0.034   64   AlInAs/GaInAs 0.5%% strain
0.035  0.0896 1.53e-18 0.034  64   AlInAs/GaInAs 1.2/0.8%% strain
";
  close_out fd
 with
  exc ->
   begin try close_out fd with
     _ -> ()
   end;
   raise exc
let create_sewself_par upstate =
 let fd = open_out "sewself.par" in
 try
  Printf.fprintf fd "0.05 emin = .00205; /* .108 */
0.40   emax = .41;
1600   n = 700;
3 stepwf = 3;
30   iniz0 = 30; /* 100, 95 */
25   fin0 = 25;
1 writecoef = 1; /*1 = writes the coeffs in \"coef.dat\"*/
1 gnuplotgenere
1e-8   eerr0_double = .00000001;
1e-21 trespetit_double = 1e-21;
1e-20 incmine_double = .0000000000001; /*Minimum increment FOR energy*/
150 clockmax = 100;
9 clockbin = 7;
1 indexx[1] = 1;
8 indexx[2] = 2;
%u   indexx[3] = 3;
-0.15 mumin;
0.15   mumax;
0.002 stepmu;
11   imax: number of relevant states for the self-consistency;
0.3 ni = ni exp(-alpha) + ni2 exp(-2alpha) ++nijmax exp(-jmax*alpha)
1 itarget: state on which the convergence is made
```

```
6 idepth : number of states taken for the convergence
2e-4  convergence target for the self-consistant programm, in eV
0.54  Energy spanned during the self-consistant computation
20  Maximum number of iterations (needed for version 2.13)
0.001 Halfwidth half max for the absorption (needed for version 2.15)
3.6 Refractive index                  (needed for version 2.15)
0.014 Target wavelength (needed by 2.15 to compute identify absorption)"
    upstate;
  close_out fd
 with
  exc ->
   begin try close_out fd with
    _ -> ()
   end;
   raise exc
```

C.3.4 Module `Tool` : utility routines

```
val read_file : string -> string
```

`read_file filename` reads the file specified by `filename` and returns its content as one (big) string. Used by the modules `Files` and `Sewer`.

Listing C.7: tool.ml

```
let read_file name =
 let file = open_in name in
 let size = in_channel_length file in
 try
  let buf = String.create size in really_input file buf 0 size; close_in file; buf
 with
  exc ->
   begin try close_in file with
    _ -> ()
   end;
   raise exc
```

C.3.5 Module `Files` : sewself .dat file reader

Reads sewselfs output files: 'densities.dat', 'pse.dat', 'Ebound.dat' (electron densities, wavefunction data, energies of bound states) and get the numerical values as array of floats

```
val get_dens : string -> float array
```

`get_dens filename` reads the occupation densities written by sewself from `filename`

```
val get_ebound : string -> float array
```

`get_ebound filename` reads the energies of the bound states written by sewself from `filename`

```
val get_pse : string -> float array array
```

`get_pse filename` reads the state wavefunctions written by sewself from `filename`

Listing C.8: files.ml

```
let transposeArray (m : float array array) =
 let mx = Array.make_matrix (Array.length m.(0)) (Array.length m) m.(0).(0) in
 for i = 0 to Array.length m - 1 do
  let row_i = m.(i) in
  for j = 0 to Array.length row_i - 1 do mx.(j).(i) <- row_i.(j) done
 done;
 mx
let readpse txt =
 let rec readlines ibt =
  let rec readnums ibl =
   try Scanf.bscanf ibl " %E " (fun x -> x :: readnums ibl) with
     End_of_file -> []
  in
  try
   Scanf.bscanf ibt "%[^\n]\n"
     (fun x ->
       Array.of_list (readnums (Scanf.Scanning.from_string x)) :: readlines ibt)
  with
   End_of_file -> []
 in
 transposeArray (Array.of_list (readlines (Scanf.Scanning.from_string txt)))
let readdens txt =
 let s = String.index txt '\n'
 and m = String.length txt in
 (readpse (String.sub txt (s + 2) (m - s - 2))).(0)
let readebound = readdens
(* public functions *)
let get_dens fn = if Sys.file_exists fn then readdens (Tool.read_file fn) else [| |]
let get_ebound = get_dens
let get_pse fn = if Sys.file_exists fn then readpse (Tool.read_file fn) else [| |]
```

C.3.6 Module `Igor` : *Igor TEXT* file writer

Routines to create & write *Igor TEXT* (.itx) files.

```
val header : string -> Pervasives.out_channel
```

Opens (creates) file and writes Igor Textfile header.

```
val wave :
  Pervasives.out_channel ->
  ?x0:float ->
  ?dx:float ->
  ?xU:string ->
  ?dataU:string -> ?notes:string list -> ?name:string -> float array -> unit
```

Writes data (float array) to open file descriptor.

```
val itxFile :
  ?x0:float ->
  ?dx:float ->
  ?xU:string -> ?dataU:string -> ?name:string -> float array -> string -> unit
```

Writes just one wave file (created, writes, closes).

Listing C.9: igor.ml

```
let header filename = let file = open_out filename in Printf.fprintf file "IGOR\n"; file
let wave
  file ?(x0 = 0.) ?(dx = 1.) ?(xU = "") ?(dataU = "") ?(notes : string list = [])
    ?(name = "wave0") (dat : float array) =
  let name = "'" ^ name ^ "'" in
  Printf.fprintf file "WAVES/D %s\n" name;
  Printf.fprintf file "BEGIN\n";
  Array.iter (fun x -> Printf.fprintf file "\t%E\n" x) dat;
  Printf.fprintf file "END\n";
  Printf.fprintf file "X SetScale/P x %E,%E,\"%s\", %s;\n" x0 dx xU name;
  Printf.fprintf file "X SetScale y 0,0,\"%s\", %s;\n" dataU name;
  List.iter (fun p -> Printf.fprintf file "X Note %s \"%s\";\n" name p) notes;
  Printf.fprintf file "\n"
let itxFile
  ?(x0 = 0.) ?(dx = 1.) ?(xU = "") ?(dataU = "") ?(name = "wave0") (dat : float array)
    filename =
  let file = header filename in
  try wave file ~x0:x0 ~dx:dx ~xU:xU ~dataU:dataU ~name:name dat; close_out file with
    exc ->
      begin try close_out file with
        _ -> ()
      end;
      raise exc
```

C.3.7 Module Sewer : sewself controller & parser

Wrapper around sewself, tries to control (by faking keys) and parse output from sewself.

```
val send_keys : Pervasives.out_channel -> string list -> unit
```

`send_keys outchannel keylist` sequentialy sends all keys (characters) `keylist` to the process listening on the other end of `outchannel` (sewself)

```
val get_out : Pervasives.in_channel -> string list
```

`get_out inchannel` returns a list of strings (one per line) of the output from the process writing into the other end of `inchannel` (sewself), up to the line " q :quit " (sewself's command prompt)

```
val all_out : Pervasives.in_channel -> unit
```

discard all output in `inchannel` up to End-Of-File

```
val parse : int -> Pervasives.in_channel -> Type.transUpper
```

`parse upstate inchannel` parses the output for one upper state, returning a `Type.transUpper structure`

```
val scanT :
  ?sewselfpath:string ->
  ?selfcons:bool ->
  ?tmin:float -> ?tmax:float -> ?tstep:float -> int list -> string -> unit
```

`scanT ~sewselfpath:"/path/to/sewself/executable" ~selfcons:TrueOrFalse ~tmin:startTemperature ~tmax:stopTemperature ~tstep:temperatureStep upperStatesList "/path/to/binfile"` runs sewself for states in `upperStatesList` and the specified temperatures, appending the structured results (marshall out) to the binary file

Listing C.10: sewer.ml

```
let psedat = "pse.dat"
and dendat = "densities.dat"
and enedat = "Ebound.dat"
open UnixLabels
open Type
let rec send_keys ch vals =
  match vals with
    [] -> ()
  | h :: t -> output_string ch (h ^ "\n"); flush ch; send_keys ch t
let rec get_out ch =
  match input_line ch with
    " q :quit " as q -> [q]
  | o -> o :: get_out ch
let rec all_out ch =
  try ignore (input_line ch); all_out ch with
    End_of_file -> ()
let rec print_list l =
  match l with
    [] -> ()
  | h :: t -> print_endline h; print_list t
let rec parse up ch =
  let ib = Scanf.Scanning.from_channel ch in
```

```
  let rec scanner down =
    if down <= 0 then []
    else
      Scanf.bscanf ib "%_[^\n]\n%_[^\n]\n_%s_=_%E_meV,_z_=_%E_tauem_\
=_%E_tauabs_=_%E_The_integral_is_=_%E_The_\
lifetime_is_=_%E_ps_Lifetime_(at_T_=_%E_K)_=_%E_"
        (fun st e zz tem tabs int l0 temp lt ->
          if st <> Printf.sprintf "E%u%u" up down then
            raise (Failure "wrong_order_of_transitions");
          {lower = down; energy = e; z = zz; rate_em = 1. /. tem;
           rate_abs = 1. /. tabs; integral = int; rate_0 = 1. /. l0;
           rate_T = 1. /. lt} ::
           scanner (down - 1))
  in
  let lws = scanner (up - 1) in
  Scanf.bscanf ib "%_[^\n]\n_approximate_upper_state_\
lifetime_tau_(at_T_=_%E_K)_=_%E_"
    (fun t tot -> {upper = up; lowers = lws; rate_total = 1. /. tot})
let rec iSeries ?(vstep = 1) vmin vmax =
  if vmin > vmax then [] else vmin :: iSeries ~vstep:vstep (vmin + vstep) vmax
let rec fSeries ?(vstep = 1.) vmin vmax =
  if vmin > vmax then [] else vmin :: fSeries ~vstep:vstep (vmin +. vstep) vmax
let rec msew sewselfpath sc temps ups logfile =
  if sc && List.length temps <> 1 then
    List.map (fun t -> List.nth (msew sewselfpath sc [t] ups logfile) 0) temps
  else
    begin
      List.iter (fun f -> if Sys.file_exists f then Sys.remove f)
        [psedat; dendat; enedat];
      let ene = ref [| |]
      and den = ref [| |]
      and pse = ref [| |] in
      let (cin, cout) = open_process sewselfpath in
      Printf.printf "Started_sewself";
      let sk = send_keys cout
      and igo () = ignore (get_out cin) in
      let readfiles () =
        sk ["w"];
        igo ();
        ene := Files.get_ebound enedat;
        pse := Files.get_pse psedat;
        if sc && Sys.file_exists dendat then den := Files.get_dens dendat
      in
      let parseIn up temp =
        if sc then sk ["b"; string_of_int up]
        else
          begin
            sk ["b"; string_of_float temp; string_of_int up]; ignore (input_line cin)
          end;
        let tu = parse up cin in Printf.printf "%u." up; igo (); tu
      in
      let parseEx temp =
        if sc then
          begin
            Printf.printf "_(selfconsistent)_for_%g_K" temp;
            sk ["e"; string_of_float temp];
            igo ();
            readfiles ()
          end
        else Printf.printf "\n__for_%g_K:_%!" temp;
        let uppers = List.map (fun u -> parseIn u temp) ups in
        assert (Array.length !ene = Array.length !pse);
        let d =
          {temperature = temp; uppers = uppers; energies = !ene; densities = !den;
           wavefunctions = !pse}
```

```
      in
      begin match logfile with
        Some tofile ->
          marshout ~append:true tofile [d]; Printf.printf "␣appended␣to␣%s" tofile
      | None -> ()
      end;
      d
    in
    igo ();
    readfiles ();
    let tuls = List.map parseEx temps in
    sk ["q"];
    ignore (all_out cin);
    begin match close_process (cin, cout) with
      WEXITED i -> Printf.printf "EXIT:␣%u\n%!" i
    | WSIGNALED i -> Printf.printf "SIGNAL:␣%u\n%!" i
    | WSTOPPED i -> Printf.printf "STOP:␣%u\n%!" i
    end;
    tuls
  end
let scanT
  ?(sewselfpath = "msewself") ?(selfcons = false) ?(tmin = 1.) ?(tmax = 350.)
   ?(tstep = 1.) upl tofile =
  let _ =
    msew sewselfpath selfcons (fSeries ~vstep:tstep tmin tmax) upl (Some tofile)
  in
  ()
```

To allow for the output parsing, a slightly modified sewself (msewself) has to be used. This is obtained by patching sewself.c (version 2.18, cvs revision 1.12) with the modifications as in listing C.11 below with the command

```
$ patch -o msewself.c sewself.c msewself.patch
```

and compiling it with

```
$ gcc -O3 -lm -o msewself msewself.c
```

Listing C.11: msewself.patch

```
--- sewself.c
+++ msewself.c
@@ -537,4 +537,5 @@
   printf("␣w␣:write␣the␣wavefunction␣in␣a␣file␣\n");
   printf("␣q␣:quit␣\n");
+  fflush(stdout);
   ch2 = getchar();
   getchar();
@@ -558,4 +559,5 @@
        fprintf(stderr,"Can't␣open␣file␣%s\n",fn5__S); exit(1);
      }
+     double Tbkup=T; //local modification to not keep T if no selfcons
     if (T<0) {
       printf("␣The␣temperature␣");scanf("%lf",&T);printf("␣K␣\n");
@@ -564,4 +566,5 @@
     printmorestates(1);
     fclose(fp_6);
+    T=Tbkup; //local modification - restore T in all cases
     break;
   case 'c':
@@ -619,4 +622,5 @@
   case 'w':
     writewfct();
```

```
+    writedensities(); //write densities also in 'w'
     break;

@@ -1021,8 +1025,8 @@
 { double temp;
  if (k) {
-  printf("␣up:␣E%2d␣=␣%1.4f␣␣m␣=␣%1.3f␣location␣<x>␣=␣%4.0f␣<psi|1+T|psi>␣=␣%1.3f\n",
     i,Ebound2[i],msn[i],x[i],unplust[i]);
-  printf("␣up:␣E%2d␣=␣%1.4f␣␣m␣=␣%1.3f␣location␣<x>␣=␣%4.0f␣<psi|1+T|psi>␣=␣%1.3f\n",
     j,Ebound2[j],msn[j],x[j],unplust[j]);
+  printf("␣up:␣E%2d␣=␣%g␣␣m␣=␣%g␣location␣<x>␣=␣%g␣<psi|1+T|psi>␣=␣%g\n",i,Ebound2[i
     ],msn[i],x[i],unplust[i]);
+  printf("␣up:␣E%2d␣=␣%g␣␣m␣=␣%g␣location␣<x>␣=␣%g␣<psi|1+T|psi>␣=␣%g\n",j,Ebound2[j
     ],msn[j],x[j],unplust[j]);
 }
-printf("␣E%d%d␣=␣%3.2f␣meV,␣z␣=␣%4.2f␣tauem␣=␣%4.2f␣tauabs␣=␣%4.2f\n",i,j,(float)
     1000*(Ebound2[i]-Ebound2[j]),
+printf("␣E%d%d␣=␣%g␣meV,␣z␣=␣%g␣tauem␣=␣%g␣tauabs␣=␣%g\n",i,j,(float)1000*(Ebound2[i
     ]-Ebound2[j]),
 (float)dipole(i,j),(float)phononem(i,j),(float)phononabs(i,j));
 }
@@ -1046,8 +1050,14 @@
    kT = 0.087e-3*T;
    bosefact = 1/(exp(hlo/kT)-1); temp1 = 1e-6; /* small enough, no? */
+   /* ask indexx[3] each time, keep a backup in case it is used elsewhere ... */
+   int i3bkup, i3local;
+   printf("␣upper␣state␣");
+   scanf("%i",&i3local ); getchar();
+   indexx[3]=i3local;
+   /* --- */
    for(j = indexx[3]-1;j>0;j--){
     printstates(indexx[3],j,0);
     temp2 =(Ebound2[indexx[3]] - Ebound2[j]);
-    fprintf(fp_6,"␣%f␣␣␣␣␣%f␣\n",(float)(1000*temp2),(float)temp2*pow(dip,2));
+    fprintf(fp_6,"␣%g␣␣␣␣␣%g␣\n",(float)(1000*temp2),(float)temp2*pow(dip,2));
     if (temp2 >= hlo){
 energemiss = temp2 + kT;} /* Transition energy + kT */
@@ -1062,9 +1072,12 @@
 ratetempem += (1 + bosefact)*exp(-(hlo-temp2)/kT)/tauarb;
     }
-    printf("␣Lifetime␣(at␣T␣=␣%2.3f␣K)␣=␣%2.3f␣\n␣",T,1/(ratetrans));
+    printf("␣Lifetime␣(at␣T␣=␣%g␣K)␣=␣%g\n␣",T,1/(ratetrans));
     ratetempabs += bosefact/tauabs;
     ratetot = ratetempabs + ratetempem;}
-  printf("␣approximate␣upper␣state␣lifetime␣tau␣at␣(T␣=␣0)␣=␣%2.3f␣\n␣",1/temp1);
-  printf("␣approximate␣upper␣state␣lifetime␣tau␣(at␣T␣=␣%2.3f␣K)␣=␣%2.3f␣\n␣",T,1/
     ratetot);
+  printf("␣approximate␣upper␣state␣lifetime␣tau␣at␣(T␣=␣0)␣=␣%g␣\n␣",1/temp1);
+  printf("␣approximate␣upper␣state␣lifetime␣tau␣(at␣T␣=␣%g␣K)␣=␣%g\n␣",T,1/ratetot);
+  /* restore indexx[3] */
+  indexx[3]=i3bkup;
+  /* --- */
   break;
   case 2: /* QWIP absorption from upper state */
@@ -1421,10 +1434,10 @@
     };
     sum = 2*deltheta*sum;
-    printf("␣The␣integral␣is␣=␣%f␣\n",(sum));
+    printf("␣The␣integral␣is␣=␣%g␣\n",(sum));
     rate = mwi*m0*pow((q0/hbarre),3)*hlo/(8*pi*eps0*kp0);
     rate = 1e-10*rate*sum;
     rate = rate*unplust[iinit]*unplust[ifin];
     tauarb = 1e12/rate;
-    printf("␣The␣lifetime␣is␣␣=␣%e␣ps␣\n",(float)tauarb);
+    printf("␣The␣lifetime␣is␣␣=␣%g␣ps␣\n",(float)tauarb);
```

```
82      return tauarb;
83      };
```

C.3.8 Module `Calculator` : calculations from sewself output

Calculates and sums up probabilities, densities and more

```
type func = {
 waveName : string ;
 dataUnit : string ;
 waveNotes : string list ;
 func : Type.transTemp -> float ;
}
```

"calculator functions" prototype: Function type allowing to "pack" all different kind of evaluation in one common type of structure. Might contain the metadata for the to-be-written Igor Wave and a function `func` extraction the value from one data set (per temperature)

```
val density : int -> func
```

`density state` creates a `func` structure extracting the occupation density of the given `state`

```
val energy : int -> func
```

`energy state` creates a `func` structure extracting the energy of the given `state`

```
val absorption : int -> int -> func
```

`absorption upstate downstate` creates a `func` structure extracting the absorption rate from `upstate` to `downstate`

```
val summedPaths : int list -> func
```

`summedPaths nodeList` creates a `func` structure extracting the complete, summed up probability along all possible paths through the path nodes given in `nodeList`

```
val summedTimes : int list -> func
```

as `summedPaths`, but finally returns the total time

```
val evaluator :
 Type.transTemp -> (float -> float -> float -> float) -> int -> int -> float
```

```
val xLoc : int -> func
```

`xLoc state` creates a `func` structure calculating <s|x|s> ("center of mass")

```
val export :
 ?statel:int list ->
 ?templ:float list -> string -> string -> func list -> unit
```

`export ~statel:[0;1;2;3] ~templ:[10.;12.;14.;300.]  inbin outitx ~genL:func_list` calculates and exports all calculations in the `func list` for the given temperatures (list) and states (list), reading from the binary file `inbin` and writing to the Igor Text File `outitx`

Listing C.12: calculator.ml

```
open Type
open ExtLib
(* accessor functions *)
let a_upper (a : transTemp) up = List.find (fun x -> x.upper = up) a.uppers
let a_lower (a : transTemp) up down =
 List.find (fun x -> x.lower = down) (a_upper a up).lowers
(*some test functions*)
let plotter (a : transTemp) up down =
 let lower = a_lower a up down
 and counter s =
  let count = ref 0 in
  String.iter (fun a -> if a = '\n' then incr count) s; count.contents
 in
 Printf.printf "%E\t%E\t%u\n" a.temperature lower.energy (Array.length a.densities)
let tester ?(err = 0.05) up (a : transTemp) =
 let upper = a_upper a up in
 let rat =
  List.fold_left (fun c e -> c +. e.rate_T) 0. upper.lowers /. upper.rate_total
 in
 abs_float (rat -. 1.) < err
let testPair (al : transTemp list) (u, l) =
 List.fold_left (fun c a -> c +. (a_lower a u l).rate_T) 0. al
let listPair (al : transTemp list) (u, l) =
 List.map (fun a -> (a_lower a u l).rate_T) al
let testorder file i up =
 let lows = (a_upper (List.nth (marshin file) i) up).lowers in
 List.iter (fun l -> Printf.printf "%u>%u %g\n" up l.lower l.energy) lows
type func =
 { waveName : string;
  dataUnit : string;
  waveNotes : string list;
  func : Type.transTemp -> float }
(* creates func, which calculates the occupied densityaa
of state s of any transTemp*)
let density s =
 {waveName = Printf.sprintf "dens%u" s; dataUnit = "cm^-2";
  waveNotes = [Printf.sprintf "density of state %u" s];
  func = fun a -> a.densities.(s - 1)}
(* creates func, which calculates the energy of state
s of any transTemp*)
let energy s =
 {waveName = Printf.sprintf "en%u" s; dataUnit = "eV";
  waveNotes = [Printf.sprintf "energy of state %u" s];
  func = fun a -> a.energies.(s - 1)}
(* creates func, which calculates the absorption rate
from d -> u, on any transTemp*)
let absorption (d : int) (u : int) =
 {waveName = Printf.sprintf "abs%ux%u" d u; dataUnit = "ps^-1";
  waveNotes = [Printf.sprintf "absorption from state %u to state %u" d u];
  func = fun a -> (a_lower a u d).rate_abs}
(* creates func, which calculates the going from state
u to state d, on any transTemp*)
let probability (u : int) (d : int) a =
 (a_lower a u d).rate_T /. (a_upper a u).rate_total
(* creates func, which sums up all probabilities of all
possible paths along the state-list nL, on any transTemp*)
let summedPaths (nL : int list) =
 let nLL = Paths.explodePath nL
 and n = String.join "x" (List.map string_of_int nL) in
 let concatMax ma sl =
  let ff r e =
   match r with
    rh :: rt when String.length e + String.length rh < ma -> (rh ^ ", " ^ e) :: rt
   | r -> e :: r
```

```
   in
   List.fold_left ff [] sl
  in
  let aps =
   List.map (fun nl -> ExtString.String.join ">" (List.map string_of_int nl)) nLL
  in
  {waveName = "p" ^ n; dataUnit = "";
   waveNotes =
    "probabilities, summed along these paths:" ::
     List.rev_map (fun x -> x ^ ",") (concatMax 300 aps);
   func = fun a -> Paths.contract (Paths.applyPair (fun u d -> probability u d a) nLL)}
(* creates func, which calculates the time used going from
state u to state d, on any transTemp*)
let onetime (u : int) (d : int) a =
  if a.energies.(u - 1) -. a.energies.(d - 1) > 0.032 then 1. /. (a_lower a u d).rate_T
  else 1.
(* fun a -> 1. /. ((a_lower a u d).rate_T );;*)
(* creates func, which sums up all probabilities of all
possible paths along the state-list nL, on any transTemp*)
let summedTimes (nL : int list) =
  let nLL = Paths.explodePath nL
  and n = String.join "x" (List.map string_of_int nL) in
  let concatMax ma sl =
   let ff r e =
    match r with
     rh :: rt when String.length e + String.length rh < ma -> (rh ^ ", " ^ e) :: rt
    | r -> e :: r
   in
   List.fold_left ff [] sl
  in
  let aps =
   List.map (fun nl -> ExtString.String.join ">" (List.map string_of_int nl)) nLL
  in
  {waveName = "t" ^ n; dataUnit = "ps";
   waveNotes =
    "times, summed along these paths:" ::
     List.rev_map (fun x -> x ^ ",") (concatMax 300 aps);
   func =
    fun a ->
     1. /.
      Paths.contractGeneral ( +. ) 0. (fun a b -> a +. 1. /. b) 0.
       (Paths.applyPair (fun u d -> onetime u d a) nLL)}
let evaluator a op l r =
  let psiL = a.wavefunctions.(l)
  and psiR = a.wavefunctions.(r) in
  let va =
   Array.fold_left ( +. ) 0.
    (Array.mapi (fun i x -> op (float_of_int i) x psiR.(i)) psiL)
  in
  let noL = Array.fold_left ( +. ) 0. (Array.map (fun x -> x *. x) psiL)
  and noR = Array.fold_left ( +. ) 0. (Array.map (fun x -> x *. x) psiR) in
  va /. sqrt (noL *. noR)
let locator a n = evaluator a (fun x l r -> l *. x *. r) n n
(* creates func, which calculates the x - location of a
state, on any transTemp*)
let xLoc (s : int) =
  {waveName = Printf.sprintf "loc%u" s; dataUnit = "A";
   waveNotes = [Printf.sprintf "<x-location> of state %u" s];
   func = fun a -> locator a (s - 1)}
(* write wavefunctions to igorfile,
todo: add interface in main.ml*)
let write_states (a : transTemp) sl fd =
  let ws s =
   let na = Printf.sprintf "wf%03u_%05g" s a.temperature
   and no =
```

```
    [Printf.sprintf "state=%u" s; Printf.sprintf "temperature=%.6g" a.temperature]
   in
   Igor.wave fd ~xU:"A" ~name:na ~notes:no a.wavefunctions.(s - 1)
  in
  List.iter ws sl
let find_temps ?(prec = 0.001) al tl =
  let res = ref [] in
  let find_temp t =
   res := !res @ List.find_all (fun a -> abs_float (a.temperature -. t) < t *. prec) al
  in
  List.iter find_temp tl; !res
(* main export function/loop*)
let export
  ?(statel : int list = []) ?(templ : float list = []) inbin outitx
   (genL : func list) =
  let dat = varin inbin in
  let fd = Igor.header outitx in
  let x0 = if List.length dat >= 1 then (List.nth dat 0).temperature else 0.
  and x1 = if List.length dat >= 2 then (List.nth dat 1).temperature else 1.
  and xUnit = "K" in
  let iW = Igor.wave fd ~x0:x0 ~dx:(x1 -. x0) ~xU:xUnit in
  let genCalc f =
   Printf.printf "start: %s " f.waveName;
   iW ~name:(f.waveName) ~dataU:(f.dataUnit) ~notes:(f.waveNotes)
     (Array.map f.func (Array.of_list dat));
   Printf.printf " done\n"
  in
  try
   List.iter genCalc genL;
   List.iter (fun a -> write_states a statel fd) (find_temps dat templ);
   close_out fd
  with
   exc ->
    begin try close_out fd with
     _ -> ()
    end;
    raise exc
```

C.3.9 Module `Main` : sewcalc - the main program

The main program, command line parsing stuff and structure setup for the other modules. The available command line options as of `sewcalc -h`

```
usage: sewcalc [options]
options:
  general options
  --version             show program's version and exit
  -h, --help            show this help message and exit
  -bFILE.bin, --binaryfile=FILE.bin
                        binary file to be used as output for the sew step /
                        input for the calc step (on input, also JSON text
                        format if FILE.json or FILE.txt)
  sewing:
    options relevant for "sewing"
    -sSTRUC.txt, --structurefile=STRUC.txt
                        do sew(self) with this file (toggles sewing on)
    --self, --sewself   switch self-consistent calculation on
    -n10, --number-of-states=10, --highest-state=10
                        the number of states to be stored
    --min=A             lowest temperature
    --max=B             highest temperature
    --step=C            temperature step
  calculating:
    options relevant for calculating
    -p9.8.7.3.2,9.7.2.1,..., --paths=9.8.7.3.2,9.7.2.1,...
                        list of paths to sum up
    -t9.8.7.3.2,9.7.2.1,..., --timepaths=9.8.7.3.2,9.7.2.1,...
                        list of paths to sum up transition times
    -a2.6,1.8,..., --abs=2.6,1.8,..., --absorption-pairs=2.6,1.8,...
                        list of absorption pairs to calculate
    -d1,2,3,..., --dens=1,2,3,..., --densities=1,2,3,...
                        list of states for which to include densities
    -l1,2,3,..., --loc=1,2,3,..., --locations=1,2,3,...
                        list of states for which to include x-locations
    -e1,2,3,..., --en=1,2,3,..., --energies=1,2,3,...
                        list of states for which to include energies
    -iWAVES.itx, --igorfile=WAVES.itx
                        filename to save the calculated results to (toggles the
                        calculation on)
    --prefix=aaa_       each wave in the output igorfile gets prefixed by this
                        string
  configuration:
    general configuration, rareley needed
    --sewselfpath=/path/to/msewself
                        name of / full path to the msewself binary
    --buildmatselfpath=/path/to/buildmatself
                        name of / full path to the buildmatself binary
    --workdir=/path/to/working_directory
                        full path to the desired working directory
```

Listing C.13: main.ml

```
open ExtLib
open OptParse
let theparser =
  OptParser.make ~usage:"%prog [options]" ~description:"general options" ~version:"0.2"
    ()
let gsew =
  OptParser.add_group theparser ~description:"options relevant for \"sewing\"" "sewing"
let strucf = StdOpt.str_option ~metavar:"STRUC.txt" ()
let _ =
  OptParser.add theparser ~group:gsew ~short_name:'s' ~long_name:"structurefile"
```

```
   ~help:"do␣sew(self)␣with␣this␣file␣(toggles␣sewing␣on)" strucf
let self = StdOpt.store_true ()
let _ =
 OptParser.add theparser ~group:gsew ~long_names:(["self"; "sewself"])
   ~help:"switch␣self-consistent␣calculation␣on" self
let nstates = StdOpt.int_option ~default:10 ~metavar:"10" ()
let _ =
 OptParser.add theparser ~group:gsew ~short_name:'n'
   ~long_names:(["number-of-states"; "highest-state"])
   ~help:"the␣number␣of␣states␣to␣be␣stored" nstates
let tmin = StdOpt.float_option ~default:1. ~metavar:"A" ()
let _ =
 OptParser.add theparser ~group:gsew ~long_name:"min" ~help:"lowest␣temperature" tmin
let tmax = StdOpt.float_option ~default:350. ~metavar:"B" ()
let _ =
 OptParser.add theparser ~group:gsew ~long_name:"max" ~help:"highest␣temperature" tmax
let tstep = StdOpt.float_option ~default:1. ~metavar:"C" ()
let _ =
 OptParser.add theparser ~group:gsew ~long_name:"step" ~help:"temperature␣step" tstep
let binf = StdOpt.str_option ~default:"dummy.bin" ~metavar:"FILE.bin" ()
let _ =
 OptParser.add theparser ~short_name:'b' ~long_name:"binaryfile"
   ~help:"binary␣file␣to␣be␣used␣as␣output␣for␣the␣sew␣step␣/␣\
input␣for␣the␣calc␣step␣(on␣input,␣also␣JSON␣text␣format␣\
if␣FILE.json␣or␣FILE.txt)"
   binf
let gcalc =
 OptParser.add_group theparser ~description:"options␣relevant␣for␣calculating"
   "calculating"
let paths = StdOpt.str_option ~metavar:"9.8.7.3.2,9.7.2.1,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'p' ~long_name:"paths"
   ~help:"list␣of␣paths␣to␣sum␣up" paths
let times = StdOpt.str_option ~metavar:"9.8.7.3.2,9.7.2.1,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'t' ~long_name:"timepaths"
   ~help:"list␣of␣paths␣to␣sum␣up␣transition␣times" times
let abs = StdOpt.str_option ~metavar:"2.6,1.8,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'a'
   ~long_names:(["abs"; "absorption-pairs"])
   ~help:"list␣of␣absorption␣pairs␣to␣calculate" abs
let dens = StdOpt.str_option ~metavar:"1,2,3,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'d'
   ~long_names:(["dens"; "densities"])
   ~help:"list␣of␣states␣for␣which␣to␣include␣densities" dens
let locs = StdOpt.str_option ~metavar:"1,2,3,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'l'
   ~long_names:(["loc"; "locations"])
   ~help:"list␣of␣states␣for␣which␣to␣include␣x-locations" locs
let ens = StdOpt.str_option ~metavar:"1,2,3,..." ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'e' ~long_names:(["en"; "energies"])
   ~help:"list␣of␣states␣for␣which␣to␣include␣energies" ens
let igorf = StdOpt.str_option ~metavar:"WAVES.itx" ()
let _ =
 OptParser.add theparser ~group:gcalc ~short_name:'i' ~long_name:"igorfile"
   ~help:"filename␣to␣save␣the␣calculated␣results␣to␣(toggles␣the␣calculation␣on)"
   igorf
let waveprefix = StdOpt.str_option ~metavar:"aaa_" ()
let _ =
 OptParser.add theparser ~group:gcalc ~long_name:"prefix"
   ~help:"each␣wave␣in␣the␣output␣igorfile␣gets␣prefixed␣by␣this␣string" waveprefix
```

```
let gconf =
  OptParser.add_group theparser ~description:"general configuration, rareley needed"
    "configuration"
let sewselfpath = StdOpt.str_option ~default:"msewself" ~metavar:"/path/to/msewself" ()
let _ =
  OptParser.add theparser ~group:gconf ~long_name:"sewselfpath"
    ~help:"name of / full path to the msewself binary" sewselfpath
let buildmatselfpath =
  StdOpt.str_option ~default:"buildmatself" ~metavar:"/path/to/buildmatself" ()
let _ =
  OptParser.add theparser ~group:gconf ~long_name:"buildmatselfpath"
    ~help:"name of / full path to the buildmatself binary" buildmatselfpath
let workdir = StdOpt.str_option ~metavar:"/path/to/working_directory" ()
let _ =
  OptParser.add theparser ~group:gconf ~long_name:"workdir"
    ~help:"full path to the desired working directory" workdir
let calcList = ref []
  let _ = OptParser.parse_argv theparser
let _ =
  if !(Sys.interactive) || Opt.is_set strucf || Opt.is_set igorf then ()
  else OptParser.usage theparser ();
  if Opt.is_set workdir then Sys.chdir (Opt.get workdir)
let _ =
  if Opt.is_set strucf then
    begin
      Printf.printf "do sewing %s -> %s\n" (Opt.get strucf) (Opt.get binf);
      if Opt.get self then Printf.printf "(self consistant)\n";
      Printf.printf "T = %g K .. %g K each %g K\n" (Opt.get tmin) (Opt.get tmax)
        (Opt.get tstep);
      let usl = List.rev (List.init (Opt.get nstates - 1) (fun x -> x + 2)) in
      List.iter (fun u -> Printf.printf "%u; " u) usl;
      Printf.printf "\n%!";
      Creator.buildmatself ~exe:(Opt.get buildmatselfpath) (Opt.get strucf);
      let fn = Opt.get binf in
      Type.marshout ~append:false fn [];
      Sewer.scanT ~sewselfpath:(Opt.get sewselfpath) ~selfcons:(Opt.get self)
        ~tmin:(Opt.get tmin) ~tmax:(Opt.get tmax) ~tstep:(Opt.get tstep) usl fn
    end;
  if Opt.is_set igorf then
    begin
      Printf.printf "do calculations %s -> %s\n" (Opt.get binf) (Opt.get igorf);
      let chopP s = String.nsplit s "."
      and chopK s = String.nsplit s "," in
      let preparePaths s =
        let sl = chopP s in
        let il = List.map int_of_string sl in
        calcList := Calculator.summedPaths il :: !calcList
      and prepareTimes s =
        let sl = chopP s in
        let il = List.map int_of_string sl in
        calcList := Calculator.summedTimes il :: !calcList
      and prepareAbs s =
        match chopP s with
          [d; u] ->
            let di = int_of_string d
            and ui = int_of_string u in
            calcList := Calculator.absorption di ui :: !calcList
        | _ -> OptParser.error theparser "absorbtion not a pair"
      and prepareDens s =
        let si = int_of_string s in calcList := Calculator.density si :: !calcList
      and prepareLocs s =
        let si = int_of_string s in calcList := Calculator.xLoc si :: !calcList
      and prepareEns s =
        let si = int_of_string s in calcList := Calculator.energy si :: !calcList
      and prefixer cL =
```

```
    if Opt.is_set waveprefix then
     let adder (f : Calculator.func) =
       {f with Calculator.waveName = Opt.get waveprefix ^ f.Calculator.waveName}
     in
     List.map adder cL
    else cL
  in
  if Opt.is_set abs then List.iter prepareAbs (chopK (Opt.get abs));
  if Opt.is_set paths then List.iter preparePaths (chopK (Opt.get paths));
  if Opt.is_set times then List.iter prepareTimes (chopK (Opt.get times));
  if Opt.is_set dens then List.iter prepareDens (chopK (Opt.get dens));
  if Opt.is_set locs then List.iter prepareLocs (chopK (Opt.get locs));
  if Opt.is_set ens then List.iter prepareEns (chopK (Opt.get ens));
  calcList := prefixer !calcList;
  Calculator.export (Opt.get binf) (Opt.get igorf) (List.rev !calcList)
end
```

Index

Published work

- ⋆ Marcel Graf, Giacomo Scalari, Daniel Hofstetter, Jérôme Faist, Harvey Beere, Giles Davies, Edmund Linfield, and David Ritchie. THz range quantum well infrared photodetectors. In *Infrarot Kolloquium 2003*. 2003. Oral presentation.

- ⋆ Marcel Graf, Giacomo Scalari, Daniel Hofstetter, Jérôme Faist, Harvey Beere, Giles Davies, Edmund Linfield, and Dennis Ritchie. THz range quantum well infrared photodetector. In *ITQW 2003*. 2003. Poster.

- ⋆ Marcel Graf, Giacomo Scalari, Daniel Hofstetter, Jérôme Faist, Harvey Beere, Edmund Linfield, David Ritchie, and Giles Davies. Terahertz range quantum well infrared photodetector. *Applied Physics Letters*, 84(4):475–477, 2004. doi:10.1063/1.1641165.

- ⋆ Marcel Graf, Nicolas Hoyler, Marcella Giovannini, and Daniel Hofstetter. InP-basierende Quantenkaskadendetektoren. In *Infrarot Kolloquium 2005*. 2005. Oral presentation.

- ⋆ Marcel Graf, Nicolas Hoyler, Marcella Giovannini, Jérôme Faist, and Daniel Hofstetter. InP-based quantum cascade detectors. In *ICMAT 2005*. 2005. Oral presentation.

- ⋆ Giacomo Scalari, Stéphane Blaser, Lorenzo Sirigu, Marcel Graf, Lassaad Ajili, Jérôme Faist, Harvey Beere, Edmund Linfield, David Ritchie, and Giles Davies. Terahertz quantum cascade laser emitting at 160 μm in strong magnetic field. In *PHYSICS OF SEMICONDUCTORS: 27th International Conference on the Physics of Semiconductors - ICPS-27*, volume 772, pages 1573–1574. AIP, 2005. doi:10.1063/1.1994718.

- ⋆ Daniel Hofstetter, Esther Baumann, Fabrizio Giorgetta, Marcel Graf, Manfred Maier, Fabien Guillot, Edith Bellet-Amalric, and Eva Monroy. High-quality

AlN/GaN-superlattice structures for the fabrication of narrow-band 1.4 μm photovoltaic intersubband detectors. *Applied Physics Letters*, 88:121112, 2006. doi:10.1063/1.2185613.

⋆ Marcel Graf, Nicolas Hoyler, Marcella Giovannini, Jérôme Faist, and Daniel Hofstetter. InP-based quantum cascade detectors in the mid-infrared. *Applied Physics Letters*, 88(24):241118, 2006. doi:10.1063/1.2210088.

⋆ Daniel Hofstetter, Marcel Graf, Thierry Aellen, Jérôme Faist, Lubos Hvozdara, and Stéphane Blaser. 23 GHz operation of a room temperature photovoltaic quantum cascade detector at 5.35 μm. *Applied Physics Letters*, 89(6):061119, 2006. doi:10.1063/1.2269408.

⋆ Marcel Graf, Nicolas Hoyler, Marcella Giovannini, Jérôme Faist, and Daniel Hofstetter. Mid-infrared quantum cascade detectors on InP. In *SPIE Optics & Photonics*. 2006. Oral presentation.

⋆ Giacomo Scalari, Marcel Graf, Daniel Hofstetter, Jérôme Faist, Harvey Beere, and David Ritchie. A THz quantum cascade detector in a strong perpendicular magnetic field. *Semiconductor Science and Technology*, 21(12):1743–1746, 2006. doi:10.1088/0268-1242/21/12/042.