Rovibrational Raman thermometry for repetitive nanosecond pulsed filamentary plasma afterglows

Meekes, Wouter B.

Award date:
2022

Department of Applied Physics
Elementary Processes in Gas Discharges

Rovibrational Raman thermometry for repetitive nanosecond pulsed filamentary plasma afterglows

60 ECTS Graduation Project

W.B. Meekes

1011988

Exam Committee: dr. ir. Sander Nijdam
dr. Nico Dam
dr. Peter Zijlstra
Daily Supervisor: ir. Ravi Patel
dr. ir. Sander Nijdam

Eindhoven 7th December 2021

Abstract

Plasma-assisted combustion is a relatively novel subfield of combustion, where a plasma is used to stabilise combustion, thereby improving efficiency and reducing emissions. The advantage of plasmas for this purpose is the potential for non-thermal chemistry that plasmas offer. Rovibrational Raman spectroscopy was done on the afterglows of nano-second pulsed plasma discharges. The effect of different electrode types and discharge parameters on the temperature distribution has been investigated. Emphasis is put on preventing thermalization. The work is a proof of concept and serves to identify research areas that merit further investigation.

Three electrodes are under investigation: One electrode of exposed metal with a sharp tip, one of a metal core surrounded by a quartz tube with resin in between, and one of a metal core surrounded by a quartz tube with galinstan – a liquid metal alloy – in between. As excitation laser a pulsed Nd:YAG laser was used at a wavelength of 532 nm. The pulses were stretched using a passive pulse stretcher following the design of Gijbels [1] to allow an increase in input energy of the laser whilst remaining below the threshold for optical breakdown. A parametric study of the plasma behaviour - width, spatial spread and intensity - was conducted for comparison to the temperature measurements. This served both to investigate under what conditions the excitation laser could consistently hit the plasma and to be able to identify heating behaviour. For the thermometry measurements, a base set of parameters was determined based on ease of ignition of the plasma and future applications: Pressure 700 mbar, laser delay 13 μs, electrode distance 4 mm, burst frequency 90 Hz, flow rate 0.75 cm/s, 7 pulses per burst. For each set of measurements 1 of the parameters was varied from its base value to obtain a dependency of the temperature on that value.

The measured spectra show good agreement with the temperature fits, although some fitting parameters might be tweaked to improve the reliability of the fits. In most cases, it is found that the plasma does not thermalize. For the exposed metal electrode this is attributed to the short voltage pulses. Some unexpected thermalization is observed for this electrode, which is assumed to be a result of a spark-to-arc transition that is destabilized due to the short voltage pulses. The dielectric electrodes show no thermalization, demonstrating their use in keeping temperatures low.

Foreword

Dear reader,

A project like this is of course never the work of one person. First of all I would like to thank Sander and Nico, for their always kind and patient wisdom during our weekly meetings. You helped keep me from drifting too far in my chaotic approach to this project. I also want to thank Jaap. I thoroughly enjoyed chatting, singing, sharing memes and sometimes working, together with you in the lab and it has been super useful to have someone at the same level as me to bounce ideas off of. Furthermore, this work would not have gotten so far without the solid foundation of Timothy's work. This project would be a shadow of what it is now if you hadn't shown me the ropes. There are so many more people who helped me. The people working in the TFE labs, the lab technicians - special shoutout to Martin; with all the trouble the YAG laser gave, this project could not have been finished without his expertise - and the people of EPG, specifically the streamer group.

Apart from the people that contributed to the project directly, there have been so many people who contributed each in their own way. It has been a weird year, which made the challenge of a master's thesis both extra welcome and extra, well, challenging. My parents and my sister always provided a safe haven to come home to. A place to clear my head and get both feet solidly on the ground again. A big thank you to Priya, for always being there to talk to. I want to thank the raft crew and Frey's fiends for shaking up my sleep rhythm for the sake of my mental health on a regular basis, and so many more of my good friends.

Lastly I want to thank perhaps the most important person to this project: Ravi. Thank you for making me feel welcome in a group outside my department, for your help with prize-winning posters and for putting up with my wild ideas and chaos. Thank you for always coming to help when it was needed, for thinking along with me and showing me the practical side of physics. I have learnt so much during this project, and I couldn't have done it without you, nor would it have been nearly as enjoyable.

Since I am of course still a student - though hopefully not for too much longer - I also think it's nice to pause upon the things I've learnt during this project, as learning is of course the goal.

I've worked a lot with lasers and gotten hands-on experience with alignment and the extremely frustrating level of precision that requires. There have been countless times where I had just aligned the setup, after which the lightest of accidental touches completely misaligned the entire stupid setup. But lasers are very cool. The laser pulses bathing the lab in green light will forever be etched into my memory and there is something so exciting about a laser that is dangerous to touch. I have also learnt once more to appreciate and respect the infinite complexity of plasma physics. Through this project I have gained a much better understanding of why plasma does what it does. Countless times during the project have I bashed my head against the wall because I thought I knew what was going on, but then plasma physics decided to work differently in this regime, and my interest in it has only grown because of it. Spectroscopy was previously a field I tried to avoid because I never quite understood the plethora of diagnostics under its wing. Now that I'm more familiar with it, it scares me even more, but at least I understand it better now.

I have also learnt a lot about how (not) to tackle a project. First and foremost: Write a research proposal and make a planning on day 1! These will help guide your project and keep you focused on your goal. It helps you avoid doing useless stuff that results in a 10-page appendix chapter on something that is pretty interesting but not at all useful. Just because something is interesting, does not mean you should do it. Furthermore, on day 2 you should throw your planning out. There is no chance you'll be able to stick to it. Laser trouble, measurements going wrong, moving, lab chaos; delays are imminent. Lastly, I have learnt to just write. Bad text can be corrected later, but that will always be easier than having to produce something completely new if you sit around waiting for inspiration.

Many of these lessons I've learnt the hard way, and so the final writing phase proves a tad more stressful than I would have liked. Nevertheless I've had a blast with this project, and I genuinely hope you will enjoy my thesis as much as I've enjoyed doing the work to make it come to fruition.

- Wouter

List of symbols

Constants

Constant name	Meaning	Value
e	Elementary charge	$1.602 \cdot 10^{-19}\,\mathrm{C}$
ϵ_0	Vacuum permittivity	$8.854 \cdot 10^{-12}\,\mathrm{F/m}$
k_B	Boltzmann's constant	$1.38 \cdot 10^{-23}\,\mathrm{J/K}$
m_e	Electron mass	$9.109 \cdot 10^{-31}\,\mathrm{kg}$
m_n	Neutron mass	$1.675 \cdot 10^{-27}\,\mathrm{kg}$
m_p	Proton mass	$1.673 \cdot 10^{-27}\,\mathrm{kg}$

Table 1: List of natural constants used. All values are derived from the handbook of chemistry and physics [2].

Parameters

Parameter name	Meaning
E	Energy
E	Electric field
f	Focal lenght
f_{burst}	Burst frequency
F/O	Fuel-Oxidiser ratio
g	Degeneracy
γ_{se}	Secondary electron emission coefficient
I	Intensity
l	Streamer space charge layer thickness
λ	Wavelength
λ_D	Debye length
m	Mass
n	Particle density
N	Population
N_{pulse}	Number of pulses per burst
ω_p	Plasma frequency
P	Power
p	Pressure
ϕ	Equivalence ratio
ρ_C	Charge Density
τ	Timescale (for instance collision timescale)
θ	Divergence angle
T	Temperature
v_{01}	Vibrational transition from the ground state
v_{1v}	Vibrational transition from excited states up
V_b	Breakdown Voltage
w_0	Waist size

Table 2: List of parameters used

Abbreviations

Abbreviation	Meaning
DBD	Dielectric Barrier Discharge
DC	Direct Current
FWHM	Full Width at Half Maximum
HV	High-Voltage
IBS	Inverse Bremsstrahlung
IPCC	Intergovernmental Panel on Climate Change
KET	Knife Edge Tomography
MFC	Mass Flow Controller
OES	Optical Emission Spectroscopy
PAC	Plasma-Assisted Combustion
SNR	Signal-to-Noise Ratio
SSR	Sum Squared Residual
UV	Ultra-Violet
YAG	Yttrium Aluminium Garnet (Also used to refer to the Nd:YAG laser used throughout this thesis)

Table 3: List of used abbreviations

Contents

Chapter 1

Introduction

In october 2018, the Intergovernmental Panel on Climate Change (IPCC) published a report on the effects of climate change, with particular emphasis on the advantages of keeping the rise in global temperatures below 1.5 ^{o}C instead of the 2 ^{o}C constraint set by the Paris agreement [3]. The report estimates very drastic reductions in CO_2 emissions for models to reach this goal, requiring unprecedented transitions in most infrastructure and industrial systems. It is also explained that combating climate change will positively influence sustainable development and poverty reduction. The bottom line is that the need to reduce CO_2 emissions is high and the advantages of doing so are great [4].

The ultimate solution to reduce CO_2 emissions is to switch to renewable energy sources. This may seem simple at first glance: Just build solar panels and windmills everywhere. Sadly, the implementation of new technologies is not that simple. Both wind and solar power are still relatively new technologies, and it is expected to take several decades before either of them will cover a significant amount of the global energy usage [5]. Thus, at the same time as the global production of renewable energy is increased, the current, polluting sources of energy are gradually improved to be more efficient.

Another driver to reconsider the way current combustion engines work is air quality. The kinetics of the formation of various pollutants such as nitrogen oxides, carbon monoxide and soot have been known for quite some time now, as well as factors influencing the reactions [6]. These pollutants have various adverse effects on public health: Extended exposure to soot and other particulates (such as would be experienced by someone living in a big city) have been shown to correlate negatively with cognitive development and possibly increase the risk of Alzheimer's disease [7]. In 2015 it was discovered that Volkswagen's diesel cars had so-called 'defeat devices' that had software to reduce NO_x emissions during emission tests. This allowed the cars to pass these NO_x tests even though the standard use emissions were too high. An estimated 50% of 10000 premature deaths related to pollution in Europe in 2013 can be attributed to these defeat devices. Another 3000 of these deaths are estimated to be caused by the increased NO_x emission of diesel engines as compared to petrol engines [8].

One effort to improve current combustion engines lies in Plasma-Assisted Combustion (PAC). Here, instead of a direct heat source, a plasma is produced in the gas to be burnt, to create the radicals and excited species that are to kickstart the combustion reaction. Plasma has a unique capability in producing active species and heat. Due to the presence of charge in the plasma, the coulomb force is an important driver of transport processes in the plasma, vastly altering them with respect to a regular gas. Furthermore, the unique chemistry of the plasma opens up new chemical pathways [9].

Through this, plasma enhances combustion completeness and flame stability [10], it can extend the lower lean blowout limit and lean flammability limit [11]. As a result, the burn temperature can be lower, which improves efficiency [12]. Plasmas can very effectively reduce NO_x [13] and SO_x [14], decompose unburned hydrocarbons such as toluene [15] and control soot generation [16].

PAC relies on plasma processes taking place on sub-nanosecond timescales, whilst needing to run in

a continuous combustion mode. Combustion engines are of course macroscopic systems, but they are influenced by the behaviour of individual particles. To top it all off, it occurs in extremely non-equilibrium conditions, combining combustion kinetics with plasma physics. Evidently, current modeling technology can not fully capture all the physics on all the spatial and temporal scales and so a significant portion of our understanding must be derived from experiments.

However, inserting probes into the setup for temperature measurements inevitably perturbs the dynamics of the plasma, which means the actual behaviour of the setup is not properly measured. As such, less intrusive methods are preferentially used. The field of spectroscopy focuses on a class of measurement techniques which are generally very slightly intrusive. By measuring the spectrum of light coming off a sample, various kinds of information - from chemical composition to temperature - can be obtained. This spectrum does not have to be emitted by the sample itself, spectra of light scattered off a sample can also be analysed.

This is what happens in Raman spectroscopy. By illuminating a sample with monochromatic light, the sample can be made to scatter a small amount of light which gives information about the presence of excited species in the sample. These excited species are an indication of the temperature of the sample. Raman spectroscopy is preferentially used due to its high versatility and the ability to perform measurements on a theoretically limitless number of species simultaneously. Unfortunately, Raman spectroscopy is not all sunshine and rainbows, and there are some downsides to it. The most important downside being that Raman scattering is an extremely inefficient process. Cross sections for spontaneous Raman scattering are typically on the order of 10^{-30}cm^2 [17]. As a result, Raman spectroscopy requires very sensitive optical setups and high-powered lasers. This, in turn, means that Raman spectroscopy is usually subject to the issues associated with these setups too. Specifically for PAC, the unpredictable behaviour of plasma becomes a challenge too. Because the plasma does not always follow a straight line from one electrode to the other, the laser will sometimes miss the plasma, which can make what little signal is there inconsistent. This irreproducibility makes laser diagnostics on filamentary discharges generally difficult [18]. Furthermore, due to jitter the laser and streamer cannot be synchronised on timescales smaller than a nanosecond, which is too slow for the evolution of streamer processes. Nevertheless, Raman spectroscopy has occasionally been performed on streamer discharges to obtain rotational and vibrational temperatures in streamer afterglows [19] [20]. In the work mentioned here, the rovibrational temperatures are measured a few 10s of nanoseconds after the discharge. At that point the vibrational excitation will still be very significant, making measurements easier. In PAC the plasma serves to stabilise combustion over extended periods of time and thus it is necessary to determine the temperatures over extended periods of time. As time goes on, the vibrational states relax and the Raman signal will be reduced. Furthermore, Cléon et al. [20], use a pin-to-pin configuration with HV pulses over 3 times longer than used for this thesis, meaning they will tend towards thermal plasma, something which will be attempted to avoid in this work. The focus will instead be on creating non-thermal plasma beneficial to PAC.

All that is to say that the rovibrational Raman thermometry performed in this thesis has not been employed previously for non-thermal plasmas.

To improve the Raman signal, first of all the input laser energy will be maximised. The laser energy can not be increased indefinitely as this will induce optical breakdown, which will overshadow the Raman signal [21]. Thus, a passive pulse stretcher will be constructed following the design of Gijbels [1] and the concept of Kojima et al. [22]. In a passive pulse stretcher, a laser pulse is split into two parts, of which one will be looped around onto the beam splitter such that it attains a delay with respect to the other pulse. This way the pulse is transformed into 2 smaller pulses in quick succession, which reduces peak power and thereby decreases the probability of optical breakdown.

Furthermore, to be able to account for the unpredictability of the streamers, a parametric study is done to investigate the width, deviation and intensity of the streamers. Because of the instability of streamers, a portion of the Raman signal will be obtained from laser pulses that did not pass through the streamer. This will thereby dilute the signal from the streamer with Raman signal from the cold gas. By performing statistics on the probability that the laser hits the streamer, it will become possible to make claims about the reliability of measured temperatures and possibly correct for the dilution.

Finally, Raman measurements will be done to obtain parametric dependencies of the rovibrational temperatures of the plasma afterglow. Although coupled to some interpretation, the measurements done are a proof of concept of the methods used and serve to identify areas to investigate in future research.

Chapter 2 contains the background theory about combustion, plasmas, spectroscopy and lasers. Chapter 3 describes the used setups and methods and chapter 4 presents the results along with discussion thereof. Chapter 5 ends the thesis with some concluding remarks.

Chapter 2

Theory

This chapter explains the theory relevant to understanding the subjects tackled in this thesis. PAC is a field of study that (as the name implies) relates to both plasma and combustion physics. So before proceeding to PAC and how Raman spectroscopy comes into play it is important to consider combustion and plasma separately. After that, the theoretical principles of Raman are detailed in section 2.3. The chapter ends with the basic theory of lasers in section 2.4.

2.1 Combustion

In a combustion process a fuel and oxidising agent[1] react to release energy from their intermolecular bonds. The process requires some activation energy, and then continues due to the energy release of the reaction itself [24].

An example of a combustion reaction is that of methane and oxygen.

$$CH_4 + 2O_2 \rightarrow 2H_2O + CO_2 \tag{2.1}$$

When the reactants are available in this ratio, this reaction is called stoichiometrically balanced. The fuel-oxidizer ratio, F/O, is the mass ratio between fuel and oxidiser, and the 'equivalence ratio' ϕ is the deviation from stoichiometric fuel-oxidizer ratio:

$$\phi = \frac{(F/O)}{(F/O)_{st}} \tag{2.2}$$

At stoichiometry, $\phi = 1$, $\phi > 1$ for an excess of fuel, called 'rich' combustion, and $\phi < 1$ for a deficit of fuel, called 'lean' combustion.

A flame can burn in lean and rich conditions, but in too lean conditions there is not enough fuel for combustion, and in too rich conditions there is not enough oxygen for combustion. Too lean or too rich combustion will also decrease the flame temperature [25].

A colder flame can in some cases be beneficial. Particularly in regards to pollution hotter flames are undesirable, as high temperatures open up chemical pathways that lead to the formation of NO_x among others. For instance in a particular type of hydrogen combustion engines it has been shown that NO_x production plummets dramatically when ϕ is reduced from 0.9 to 0.55 or increased from 0.9 to 1.1 [26]. Lean mixtures are, however, more desirable than rich mixtures. In rich mixtures the exhaust contains potentially harmful unburnt compounds, such as CO [27], or a whole range of larger hydrocarbons such as aromatics [28]. Lean combustion is also more desirable from an economic perspective, as rich combustion wastes fuel - which is costly - whereas lean combustion only 'wastes' the air that's all around us.

[1] In most cases this oxidising agent is simply molecular oxygen (O_2), and for brevity we will only be referring to oxygen. However, it should be noted that there are combustion processes that do not need external oxygen, such as thermite reactions [23].

2.2 Plasma

In PAC, combustion at low equivalence ratios is stabilised with help of a plasma. Although the term is used more widely, in physics plasma refers to "an electrically conducting medium in which there are roughly equal numbers of positively and negatively charged particles" [29]. Plasmas are ionised gases, and on earth most of them are produced either during thunderstorms in the form of lightning or at the earth's poles as auroras, but in the universe over 99% of visible matter takes this form, as stars are fully ionized gases.

Since a plasma generally does not have a net charge even though it has free charges in it, it is called 'quasi-neutral'. A plasma will also try to maintain quasi-neutrality because it has free charges floating in it. If there is net charge in the plasma, this charge will produce an electric field which will tend to move the charges, eventually cancelling out the field. The length scale over which the plasma is able to shield charges is called the Debye length:

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{n_e e^2}} \tag{2.3}$$

If the Debye length is on the order of or larger than the characteristic dimension of the plasma, there will simply not be enough space to maintain quasi-neutrality. Similarly, if there are not enough particles contained within the Debye sphere (which is a volume of the order λ_D^3), the plasma will also not be able to maintain quasi-neutrality.

2.2.1 Plasma generation

Plasmas are most easily created using electric fields. By accelerating charged particles through an electric potential, they can acquire the energy to ionise neutral gas particles.

In this ionisation, another electron and an ion are produced, and the original particle also remains in the gas. All three of these particles can then be further accelerated to produce exponentially more particles in what is called a Townsend avalanche, until eventually the charge density is high enough that we speak of a plasma.

The voltage required to achieve breakdown (V_b) as function of gas pressure p and electrode gap d was done extensively by Friedrich Paschen in 1889 [30]. He found that V_b had a minimum as a function of both p and d. This makes sense since - as was previously explained - a charged particle needs some distance to obtain enough energy to ionise atoms and molecules. If there are many particles or this distance is very long, the chance of the particle colliding before getting enough velocity is high. If p or d are too small, the particle simply will not collide enough to keep the avalanche going.

Paschen's experiments led to relations for the breakdown voltage of gases:

$$V_b = \frac{Bpd}{\ln Apd - \ln\left(\ln\left(1 + 1/\gamma_{se}\right)\right)} \tag{2.4}$$

Here γ_{se} is the secondary electron emission coefficient at the cathode and A and B are coefficients specific for the type of gas. As A, B and γ_{se} are all experiment-specific parameters, we see that V_b only depends on the product $p \cdot d$. Plotting $V_b(pd)$ yields the Paschen curves, shown in figure 2.1 and which nicely show the optimum value of $p \cdot d$ for the lowest breakdown voltage.

The first particle to start the townsend avalanche is produced quite randomly, from cosmic rays to photoionization by UV light.

2.2.2 Plasma dynamics

Plasmas have many unique properties in terms of dynamics and interactions with other materials. This can be partially attributed to the fact that electrons are thousands of times lighter than ions and the neutral gas molecules. As a result, the ions respond to changes in the plasma on an entirely different timescale than the electrons.

This behaviour manifests itself in the form of the plasma frequency. The plasma frequency is an indication for the timescale on which free charged particles are able to respond and is the highest

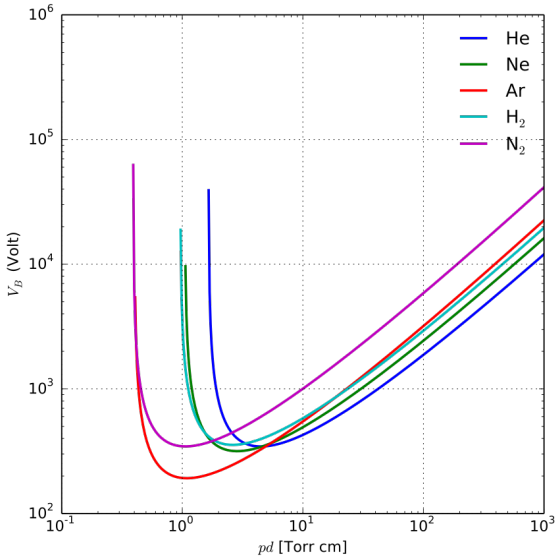


Figure 2.1: Paschen curves for various gases [31]

frequency electric field particles will be able to follow. Equation 2.5 is the equation for plasma frequency.

$$\omega_p = \sqrt{\frac{e^2 n}{\epsilon_0 m}} \tag{2.5}$$

Different particles in the same medium can have a different plasma frequency, and n and m are the density and mass of the particle for which ω_p is calculated, in cm^{-3} and kg respectively. For instance, the electron plasma frequency is around $9000\sqrt{n_e}\,\mathrm{Hz}$, and the plasma frequency of Argon ions is $33\sqrt{n_{Ar}}\,\mathrm{Hz}$. Thus at the same particle density, electrons can respond to electric fields almost 300 times as fast as Argon ions.

Furthermore, it can be derived that the ratio between the timescales of energy transfer amongst a particular particle a and between particle a and another type of particle b scales with the mass ratio between the two particles:

$$\frac{\tau_{ab}}{\tau_{aa}} = \frac{m_b}{m_a} \tag{2.6}$$

Thus the electrons will rapidly distribute the energy amongst themselves, but will take a long time to heat the ions and neutral particles. Plasmas can reach temperatures much higher than typical flames. If the applied electric field is slow enough (or DC), it will move both the ions and electrons. Although the energy transfer from electrons to the gas is low, according to equation 2.6, the energy transfer rate from the ions to the neutral particles will be as fast as the energy transfer among the ions.

However, in a plasma driven at a frequency between the electron and ion plasma frequencies[2], the ions will be largely stationary and so the applied electric field will mostly heat the electrons. Because the electrons hardly heat the gas, the electron temperature can be much higher than the temperature of the heavy particles. 2 or 3 eV (25000 K) electrons in a plasma with room temperature ions and neutrals are not uncommon in radio frequency plasmas.

This ability to produce non-thermal energy makes plasmas interesting for PAC. The plasma can supply energetic species to the gas to aid the chemical reactions without increasing the temperature of the gas which increases NO_x production.

[2]This is typically on the order of a few dozen MHz

2.2.3 Streamers

In 1939, Loeb et al. [32] observed the formation of a new type of plasma discharge, which they dubbed streamers. They illustrated that for sufficiently high values of $p \cdot d$, the simple Townsend avalanche model was insufficient to explain the breakdown behaviour of electrical sparks. Most notably the propagation of the discharge was much faster than the electron velocity, and the discharge was more filamentary than diffuse, as would be expected for a standard Townsend discharge.

Streamers benefit a lot from local field enhancement around sharp electrodes. They take 2 forms: Positive and negative streamers, which form at the positive and negative electrode, respectively. In the onset of a positive streamer, electrons are attracted towards the positive electrode and form a small Townsend discharge. The slow ions are left behind as the electrons move towards the electrode, forming a positive space charge layer further out from the electrode. If this space charge layer is sufficiently charged and sufficiently curved, it can form an ionisation front itself. It will attract electrons further out from the electrode towards it, causing ever more Townsend avalanches within an increasing range. The electrons starting the new Townsend avalanches are created through photo-ionization by UV photons emitted from the ionization region. If the background electric field is large enough, the ionisation front can propagate away from the electrode, leaving in its wake a quasi-neutral channel, surrounded by a space charge layer of positive ions. Because the streamer propagates using a sort of chain reaction of photons rather than direct electron motion, it can move much faster than the individual electrons.

The propagation mechanism of negative streamers is much more intuitive: The negative head of the streamer accelerates electrons away from it, causing the townsend avalanches in the propagation direction of the streamer. Though more intuitive, it results in instabilities in negative streamers. Because the electrons are not only accelerated forward but also sideways, the strong curvature of the streamer head is not maintained optimally. Furthermore, the space charge layer shielding the conducting channel from the surrounding air is maintained by electrons rather than heavy ions [33].

Streamers can change significantly under similar circumstances. For short voltage pulses, with rise times on the order of 15 ns, the widths of positive streamers can increase up to 15 times when the voltage is increased from 5 to 96 kV. Streamers also have a minimum width under given conditions. Significantly above this width, streamers can branch. At minimal width, the streamer can still propagate but will simply cease to branch [34].

The streamer width is limited by the requirement that the streamer tip enhances the field ahead of it. This can only take place if the space charge layer surrounding the tip is thin compared to the streamer radius. But the space charge layer must be sufficiently thick to contain enough charge to effectively shield the streamer core from the external electric field. This condition is shown in equation

$$\epsilon_0 \Delta E = \int_l \rho_C(z) dz \tag{2.7}$$

With ΔE the difference between internal and external electric field, l the charge layer thickness and $\rho_C(z)$ the charge density as function of position in the layer. ρ can be assumed to be on the order of en_i^{ch}, with n_i^{ch} the ionization density inside the channel. As a result, the minimum value of l roughly scales as

$$l \propto \frac{\Delta E}{n_i^{ch}} \tag{2.8}$$

The space charge layer must be thicker when the difference between the electric field outside and inside is larger, as well as when the ionization rate in the streamer is lower. This ionization rate, however, increases steeply with increasing electric field, and so the space charge layer can generally be thinner if the applied electric field is larger. Since the streamer diameter must be considerably larger than l, this will approximately follow the same dependence.

Streamer velocities typically range between 0.1 and 1 mm/ns, though velocities up to 5 mm/ns are reported for high applied voltages.

An advantage of streamers is that they are very transient and non-thermal plasmas [18]. However, once the streamer head has reached the other electrode, the quasi-neutral core of the streamer

allows charge to flow freely between the two electrodes. This results in a return wave of potential redistribution, constituing the transition from a streamer to a spark discharge [35]. At this point, the gas will rapidly start heating, which increases thermionic emission from the cathode and ionisation in the discharge gap. This causes a runaway process, because as the charge density increases, the plasma is able to sustain greater currents, resulting in more heating and ionisation, eventually forming an arc plasma. Arcs can easily reach temperatures on the order of 40,000 K. A spark shows more molecular emission lines, whereas an arc is characterised by broadband ionic emission [36]. Even short sparks are already very similar to arc plasma and can reach temperatures of 1-2000 K [37]. If the gas temperature is to be kept low, sparking should be avoided.

2.2.4 Temperature control of plasmas

The main reason the plasma heats up is the self-enhancing flow of continuous current. 3 things can be done to prevent this from happening:

Using short voltage pulses, the streamer can be initiated but stopped before it crosses the electrode gap, or a spark discharge can be stopped before the ions have gained significant energy. Indeed it has been shown that gas temperatures can be kept low with ns pulse spark discharges - on the order 5-600 K - even when vibrational temperatures increase to 5000 K [38].

By blocking the discharge gap with a dielectric material, the flow of charged particles is interrupted. Breakdown may take place and ions and electrons can flow for a short while, but eventually the applied electric field will result in a buildup of charge on the dielectric. This cancels out the applied electric field until the discharge stops, setting a hard limit on the duration of the discharge and thereby gas heating.

By keeping the pressure in the discharge low, the number of collisions in the plasma will be significantly decreased which reduces the heat transfer rate to the gas. For sufficiently low pressure, the discharge will enter into the glow regime. The values of $p \cdot d$ for this transition can be as low as 13 mbar mm [39]. However, in PAC this is likely too low, as high pressures are generally beneficial to combustion [12].

2.3 Spectroscopy

Spectroscopy is a generally non-intrusive measurement method that uses the light spectra emitted by samples to derive all manner of properties from. The spectroscopic method of interest here is Raman spectroscopy, specifically ro-vibrational Raman spectroscopy.

2.3.1 Energy distributions

The distribution of translational energy in a gas at fixed temperature T will equilibrate towards the Boltzmann distribution:

$$N_n = N_0 \cdot \frac{g_n}{g_0} exp\left(\frac{\Delta E_{n0}}{k_B T}\right) \tag{2.9}$$

Which shows that the population N_n of level n relates to the ground state population N_0 through the energy difference ΔE_{n0} between the two levels. The factors g indicate the degeneracies of the energy levels. Using the Boltzmann distribution a temperature can be coupled to distributions of energies other than the translational temperature [17].

Molecules can only have discrete energies. However, where the energy of atoms lies solely in their electronic and translational levels, molecules can vibrate and rotate as well. This greatly complicates their energy level structure. The spacing between rotational energies is smaller than between vibrational energies, which are in turn smaller than the electronic spacings. This results in each electronic level having several vibrational sublevels, each of which has several rotational sublevels, as shown in figure 2.2.

Because the rotational energy levels lie very close together, the rotational temperature can generally be assumed to be in Boltzmann equilibrium with a temperature equal to the translational temperature.

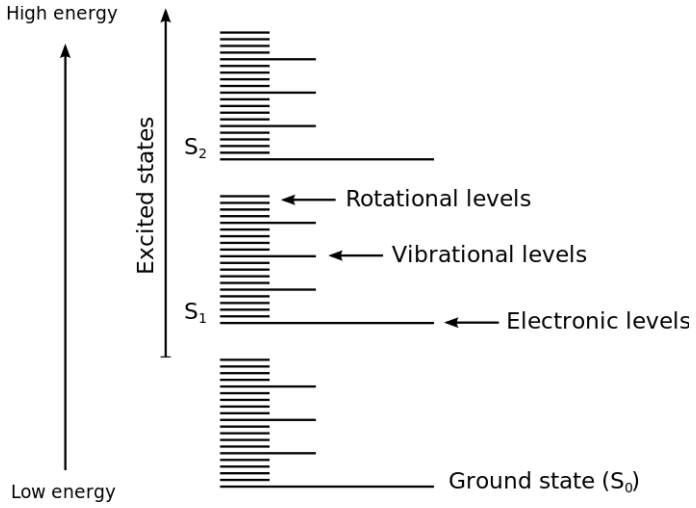


Figure 2.2: Rovibrational energy structure of a molecule [40]

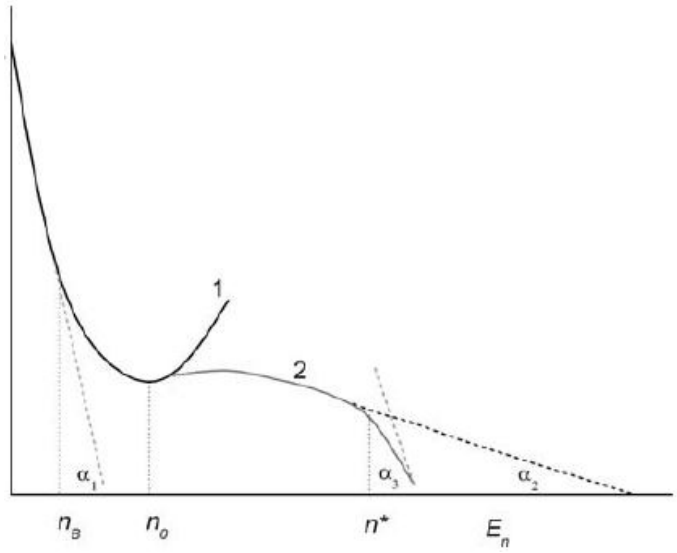


Figure 2.3: Population of vibrational states as function of vibrational quantum number for the Treanor distribution. Line 1 is the Treanor distribution without anharmonicity, line 2 accounts for the anharmonicity.

The vibrational states will also tend to relax to the Boltzmann distribution, but this can only happen if the interaction between vibrational states is much faster than the rate of creation of these states. If they are on the same order - for instance if there are many electron-neutral collisions - the vibrational energy distribution will start to deviate from Boltzmann equilibrium and tend towards the Treanor distribution. The Treanor distribution separates the temperature of the lower and higher vibrational states, making it possible to account for the extra excitation of the higher states. Furthermore, the Treanor distribution accounts for anharmonicity in the higher vibrational states, by not assuming the energy levels to be equidistant.

2.3.2 Raman shift

Raman spectroscopy relies on the Raman shift. Quantum mechanically, scattering of light on materials can be regarded as the absorption and consequent emission of photons by molecules. The absorption process will have the highest probability if the photon energy corresponds to the energy difference between two existing molecular energy levels. In this case, we speak of a resonant transition. However, it is possible for a particle to be excited to a 'virtual' energy level, after which it will almost immediately relax back to a lower level, as shown schematically in figure 2.4. In

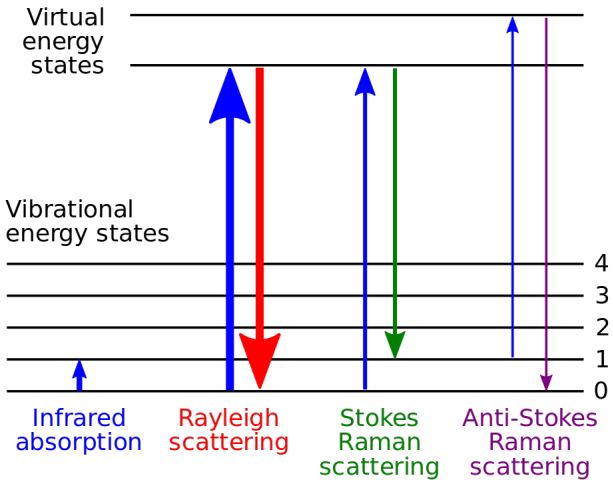


Figure 2.4: Raman scattering compared to Rayleigh scattering [41]

most cases, the emitted photon will have the same energy as the absorbed photon, in which case we speak of Rayleigh scattering. However, the particle can also relax back to a different excited state than it came from. Then the emitted photon will have a different energy than the absorbed photon. This is known as Raman scattering. If the emitted photon has a lower energy than the absorbed photon, it is called Stokes Raman scattering, if it is higher, we speak of Anti-Stokes Raman scattering.

The energy difference between the absorbed and emitted photon - the Raman shift - is equal to the energy difference between the initial and final levels. In Raman spectroscopy, the Raman shift is measured which yields a unique 'fingerprint' for the molecules under investigation.

The strength of a given Raman transition is linearly proportional to the number density of particles in the initial state. As a result, the signal strength from a given transition can be directly related to the population of a given state.

In the classical view, the electric field of the incident light with frequency ω_L induces a polarization wave in the molecules of the sample. This polarization wave - being composed of accelerating charges - emits light at ω_L. However, the polarizability of molecules changes slightly with the vibration of the molecules, which modulates the polarization wave at a frequency ω_v. This modulated wave can then be deconstructed into 3 waves with frequency ω_L, $\omega_L - \omega_v$ and $\omega_L + \omega_v$. These frequencies correspond to the Rayleigh, Stokes and Anti-Stokes frequencies, respectively [42]. Because the Rayleigh and Raman signals are emitted as accelerating dipole radiation, there is no emission in the direction of the laser polarization [43].

Symmetry also plays an important role in the strength of the Raman signal. Strongly polar bonds, such as N-O and O-H have relatively low polarizability and are thus weak Raman emitters. Apolar bonds such as C-H and the bonds in homonuclear diatomic molecules are much stronger Raman emitters [42].

2.4 Lasers

A laser is a light source capable of producing a highly coherent collimated beam of monochromatic light.

2.4.1 Light Amplification by Stimulated Emission of Radiation

The word 'LASER' is actually an acronym which stands for Light Amplification by Stimulated Emission of Radiation. Lasers are built on resonant transitions between molecular energy levels, such as were briefly mentioned in section 2.3.2. The excitation and relaxation of particles under respectively absorption and emission of a photon discussed there were spontaneous processes.

However, a particle can be made to relax from a particular resonant transition when hit by a photon with an energy equal to that transition, producing a second photon. This is called stimulated emission. The new photon will match the incident photon in energy, phase, propagation direction and polarization.

This very principle is exploited in a laser system. The stimulated emission takes place inside the gain medium. A pumping system is used to make sure the gain medium stays excited and actually has a population of particles to stimulate emission from. The pumping system can for instance be an electrical discharge or strong lamp. A set of mirrors forms a resonant cavity that reflects the laser light back and forth through the gain medium. The mirrors will be almost fully reflecting to ensure the amount of light in the gain medium remains high, but one will have a small transmission to let light escape as the usable laser.

The practical side is of course significantly more complex. For starters, gain media suitable for laser action are rare, as materials with rather specific energy level configurations are required to actually achieve lasing action. Furthermore, the construction of the resonance cavity is very critical. Deviations of microradians can cause significant drops in laser power, and the expansion or shrinking of components of the laser through temperature fluctuations can prevent laser operation altogether. Advanced laser systems thus often also require large cooling units apart from being kept in rooms with constant temperature [44].

2.4.2 Laser damage

In geometric optics, when a bundle of light is focused to a point, we consider the light to really form a single point. In practice, a light bundle has a minimum width it can shrink down to. This width is called the 'waist size', w_0, and it is inversely proportional to the divergence of the beam:

$$w_0 = \frac{\lambda}{\pi\theta} \tag{2.10}$$

Thus if a laser beam is focused with a lens with focal length f, the smaller f, the smaller the waist size. The electric field of light in a medium is related to the intensity of light as

$$\mathrm{E} = \sqrt{\frac{2I \cdot n}{c \cdot \epsilon}} \tag{2.11}$$

With n the refractive index of the medium and ϵ the dielectric constant [43]. As the intensity is power per unit area, we find the proportionalities:

$$\mathrm{E} \propto \sqrt{I} \propto \frac{1}{w_0} \propto \frac{1}{f} \tag{2.12}$$

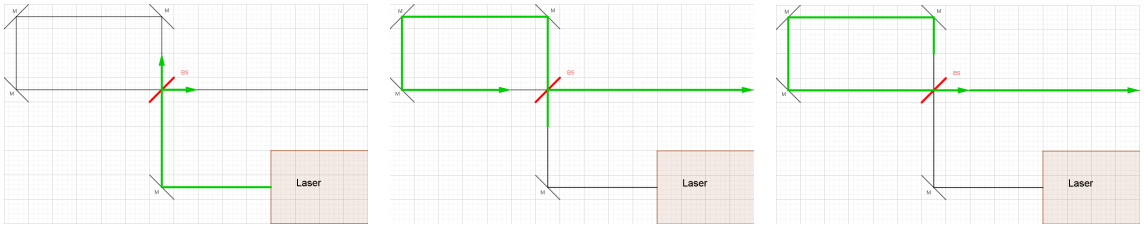
The electric field is the main culprit to causing damage for lasers with pulse lengths $< 10\,\mathrm{ns}$. For longer pulses, the main mechanism is thermal absorption which can cause fracturing or melting of the material [45].

For short pulses, the electric field causes dielectric breakdown, which follows the Townsend breakdown mechanism explained in section 2.2.1. An important requirement is that the electron-neutral collision rate during this avalanche must be much faster than the electric field of the laser. Since this rate is greater for higher particle density, optical breakdown is more likely at high pressure. Once the electron density is high enough, the material will become optically thick to the laser, resulting in a rapid increase in the absorption of laser power and thereby the temperature [46] [47].

In air, dust can greatly reduce the threshold for optical breakdown, as dust particles typically have ionization potentials $< 1\,\mathrm{eV}$ [48]. Visible light lies significantly above this energy and can thus easily release some seed electrons from these particles.

Pulsed lasers

Lasers can also be designed to output their energy in pulses rather than continuously. This is done through 3 mechanisms: Pulsed pumping, mode-locking and Q-switching. For the sake of relevance and brevity only Q-switching will be considered. A Q-switched laser employs a mechanism that can



(a) The pulse first passes through the beam splitter.

(b) The pulse is almost entirely through the beam splitter.

(c) The pulse has looped back onto itself.

Figure 2.5: Three points in time of the laser beam passing through the pulse stretcher cavity. The black diagonal lines labeled 'M' indicate mirrors, the red line 'BS' is the beam splitter, and the laser pulse is indicated in green.

vary the quality factor of the resonator cavity. When the quality factor is low, the light produced from spontaneous emission in the gain medium is lost, and the gain medium is not made to relax through stimulated emission. The population inversion of the laser is simply allowed to increase through continued pumping. Once the Q-switch is flipped, the quality factor of the resonator becomes high and lasing action starts. Because of the high population inversion, the laser will release a lot of energy in a very short time, typically on the order of several nanoseconds [49]. In Nd:YAG lasers, the quality factor can for instance be passively switched through saturable absorption. This is a property of media that increases their optical thickness when the light intensity is high. Once the amount of light from spontaneous emission reaches a certain threshold, the saturable absorber becomes opaque and the quality factor increases so lasing action starts [50] [51].

2.4.3 Pulse stretcher

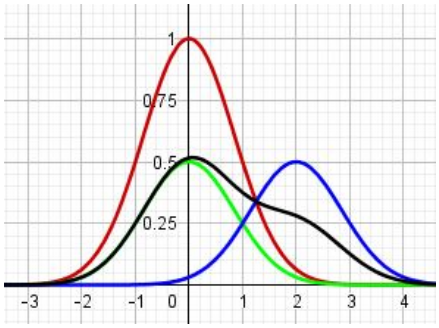
The Q-switch of the YAG laser used for this thesis in principle has a fixed pulse duration. That means its peak power can only be reduced by decreasing the total pulse energy. Reducing the energy will reduce the Raman signal. However, it is possible to build a passive pulse stretcher which can greatly reduce the peak power whilst retaining most of the pulse energy.

By using a beam splitter, a pulse's energy can be split, looped around, and turned back in on itself to effectively stretch the pulse. This process is shown in figures 2.5 a through c. First, the pulse is split into a transmitted and a reflected part. The reflected part goes on to the output of the cavity. The transmitted part is looped back to the beam splitter, but this time approaches it from the other side. The passive pulse stretcher is based on the fact that a pulse with finite duration also has a finite length. This is because the speed of light is a constant in a given medium. For instance, a 9 ns pulse in air will be 269.8 cm long. The length of the loop will be chosen such that by the time the transmitted part of the pulse is at the beam splitter, the reflected part will have just passed through completely. The loop is aligned such that the paths of the transmitted and reflected part coincide.

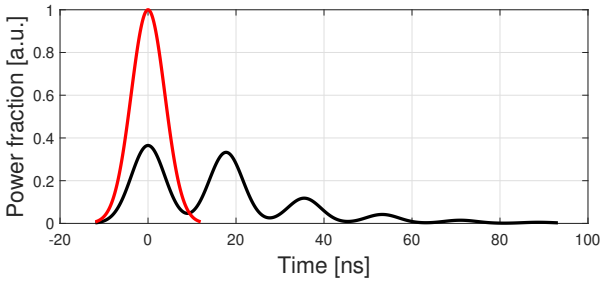
Figure 2.6a shows the power as function of time before, during, and after pulse stretching. Because the delayed pulse passes through the beam splitter twice, its energy at the output of the cavity turns out lower than the reflected pulse if a 50/50 beam splitter is used. If a beam splitter with 60 % transmission and 40 % reflectance is used, the transmitted pulse will be reduced to $0.6^2 = 0.36$ of the original pulse power and thus result in a much more even pulse profile. Furthermore, upon exiting the cavity, the delayed pulse will be split again, resulting in a train of pulses gradually decreasing in power. The resultant pulse with a 60/40 beam splitter and accounting for multiple loops through the pulse stretcher cavity is shown in figure 2.6b.

Lastly, this method of pulse stretching is not limited to stretching only once. In theory, a pulse could be stretched indefinitely, although some practical issues do arise[3]. The minimum achievable

[3]Distance becomes the biggest problem. Imagine you have managed to stretch a pulse to 1 μs and want to double the duration again. This requires a cavity of length $1\,\mu s \cdot c \approx 300$ m. Although not impossible to construct, it does become highly impractical and increasingly difficult to align properly. Furthermore, due to divergence the beam diameter becomes quite significant, requiring increasingly large mirrors.



(a) Pulse profile of 1 loop through the cavity with a 50/50 beam splitter. The time scale is arbitrary.



(b) Pulse profile of many loops through the cavity with 60% transmission, 40% reflectance.

Figure 2.6: Calculated pulse power as function of time for different beam splitter ratios. The initial pulse is shown in red. The split pulse after the beam splitter is represented by the blue and green curves, and the resulting pulse is shown in black.

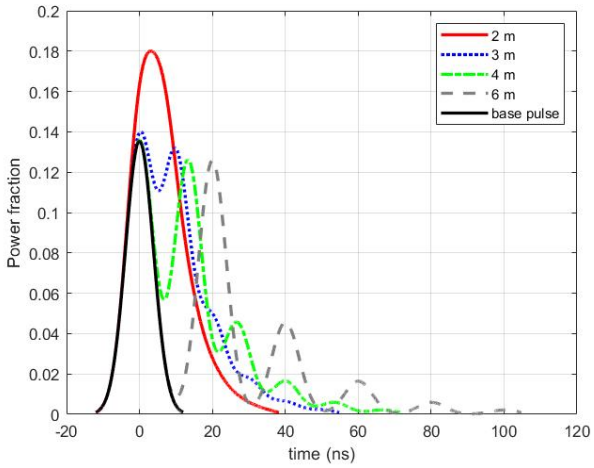


Figure 2.7: Calculated time profile for a 1-cavity pulse stretcher of lengths $2, 3, 4$ and $6\,\mathrm{m}$.

peak power is equal to the reflectivity of the beam splitters to the power of the number of beam splitters.

Gijbels [1] showed that 2 cavities provided sufficient peak power reduction to avoid breakdown in the vacuum vessel, so this was used as a starting point.

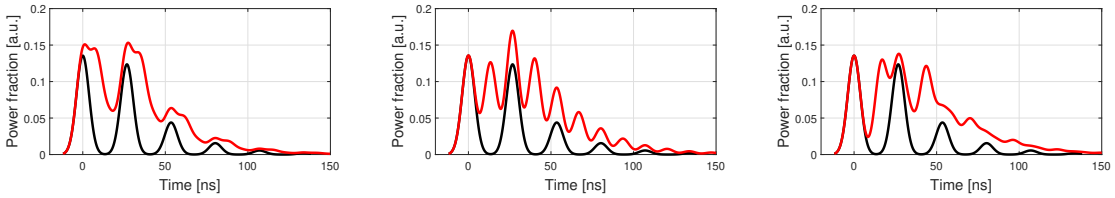
Pulse stretcher optimization

To find the optimal cavity lengths to get the minimum peak power, a script was written that made it possible to calculate and compare the peak powers for a large number of cavity combinations. The script will be described in more detail in section 3.2.1.

First off, the peak power was calculated for several single-cavity pulse stretchers with varying cavity lengths. The pulses are shown in figure 2.7.

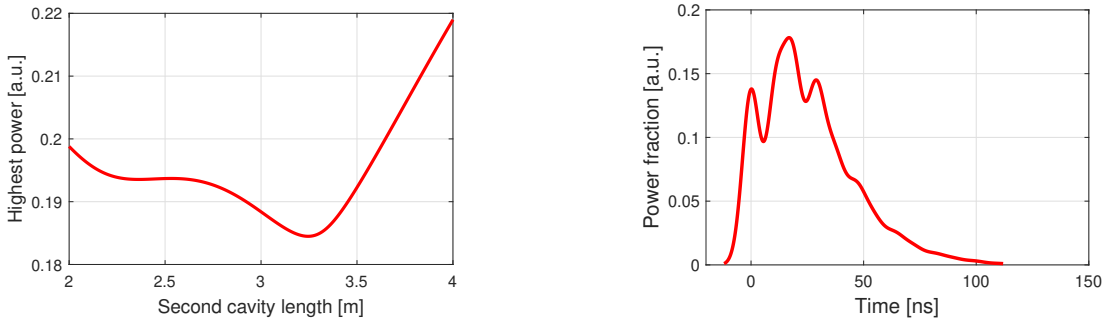
This figure shows that the peak power goes down for increasing cavity length. If the peaks are too close together in time, they will overlap, like the $2\,\mathrm{m}$ curve. From a certain delay, the pulses separate completely like for the $6\,\mathrm{m}$ cavity.

For 2 cavities the situation becomes more complicated than simply making all cavities as long as available space allows, as is illustrated in figure 2.8. The pulse stretcher designed with a $4\,\mathrm{m}$ second cavity actually has a higher peak power than the one with a $2\,\mathrm{m}$ second cavity, because the first pulse of the long cavity overlaps with the second pulse of the short cavity, resulting in the high peak at $25\,\mathrm{ns}$. Figure 2.8c shows that the solution is making the second cavity even longer.



(a) 2.5 m (b) 4 m (c) 5 m

Figure 2.8: Time profiles calculated for a 2-cavity pulse stretcher. The first cavity is 8.02 m long. The pulse with the second cavity closed is shown in black, the red curve shows the full pulse time profile for different second cavity lengths.



(a) Optimization curve for the peak power with a first cavity length of 5.32 m.

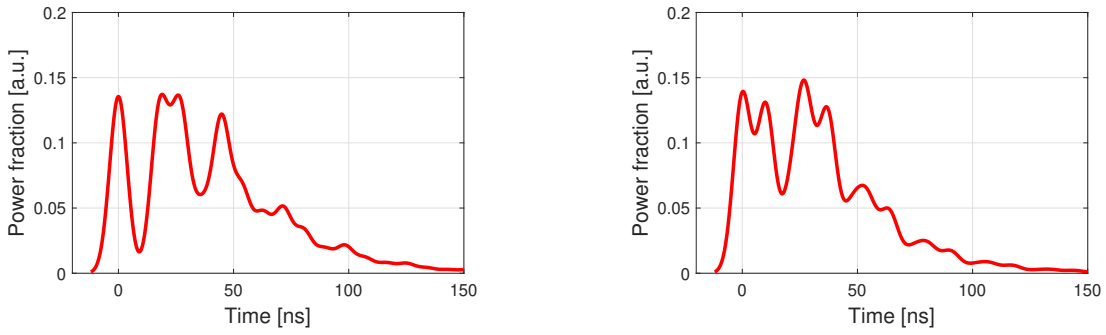
(b) Calculated time profile of the pulse with lowest peak power (5.32/3.24).

Figure 2.9

Based on available space and the pulse stretcher constructed previously by T. Gijbels [1], 2 first cavity lengths were chosen: 5.32 and 8.02 m. A 'X/Y pulse stretcher' refers to a pulse stretcher with a first cavity X m, and a second cavity Y m long.

For a first cavity length of 5.32 m, the peak power as function of second cavity length is shown in figure 2.9a. From this curve the length of the second cavity yielding the lowest peak power can be determined. The pulse corresponding to this lowest peak power is shown in figure 2.9b. The peak power has been reduced to 18.4% of its original.

For a first cavity length of 8.02 m, the optimum second cavity was found at 5.39 m. The pulse is shown in figure 2.10a. It showed a peak power reduction of 86.3%. However, there was no space available on the optical table for a second cavity this size. The optimization curve showed a second minimum in peak power at 3.06 m. This cavity still reduced the power by 85.2%. The resulting pulse is shown in figure 2.10b



(a) Second cavity 5.39 m

(b) Second cavity 3.06 m

Figure 2.10: Two best calculated time profiles for a pulse stretcher with a first cavity of 8.02 m.

Chapter 3

Setup and methods

This section describes the setups and equipment used in the project. It should be noted that not all of the setup was built from scratch. This work is a continuation of the work by T. Gijbels [1], and so section 3.1 will start by explaining which parts of the setup existed already. Section 3.2 will go into the setups revolving around the laser, t.w. the pulse stretchers and the setups to test the alignment of the pulse stretcher cavities. In section 3.3 the setups for the characterisation of the width and position of the streamer discharges are detailed. Finally, section 3.4 shows the setup used for Raman spectroscopy.

3.1 History of the setup

When the choice to do Raman spectroscopy on the PAC setup was first made, two components were present: The first was a vacuum vessel with mass flow controllers (MFCs) at the gas inlet and mounts to make different HV electrode configurations. The other component was an Nd:YAG laser. To wit: a Quanta-Ray Pro-270-10. This is a pulsed laser with pulse FWHM 8-10 ns, pulse repetition rate of 10 Hz and a maximum pulse energy of 900 mJ [52]. This yields a total average power of up to 9 W, and an instantaneous power during the pulses of up to 100 MW. One very important point of note is that although the laser had a maximum pulse energy of 900 mJ, the actual output energy fluctuated significantly per day, generally between 650 and 800 mJ. The actual power of the laser is always mentioned where relevant. Directly focusing the full laser beam inside the vacuum vessel induced breakdown. This would greatly overshadow the Raman signal, and so a passive pulse stretcher was built. The principles of this pulse stretcher were explained in section 2.4.3, and for a practical guide to building a pulse stretcher, the reader is referred to T. Gijbels [1].

Initially the laser caused damage to some of the optics of the pulse stretcher. To overcome this, new beam splitters with a slightly higher damage threshold were purchased, and a telescope was constructed to reduce the laser intensity by 43.5%. The telescope succeeded in preventing damage, but also increased losses in the pulse stretcher. An attempt was made to determine the intensity profile of the laser to calculate whether the new beam splitters would take damage from the laser. This work is presented in appendix A. Although the concept is promising, the results proved too unreliable, and so the telescope was just removed. The new beam splitters experienced no problems from direct illumination from the laser.

3.2 Laser setup

3.2.1 Pulse stretcher setups

A 5.32/3.24 and 8.02/3.06 pulse stretcher were constructed following the procedure of T. Gijbels [1]. The setup for the 5.32/3.24 pulse stretcher is shown in figure 3.1 and figures 3.2 a through c show the 8.02/3.06 pulse stretcher.

To improve and test the quality of alignment, 3 main properties were observed: The focal point

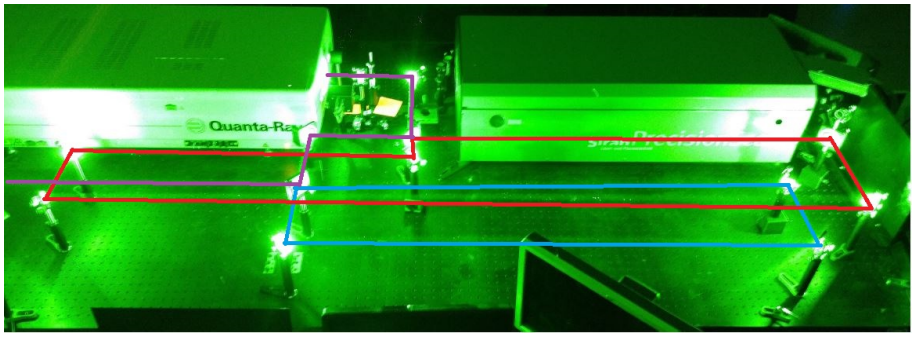
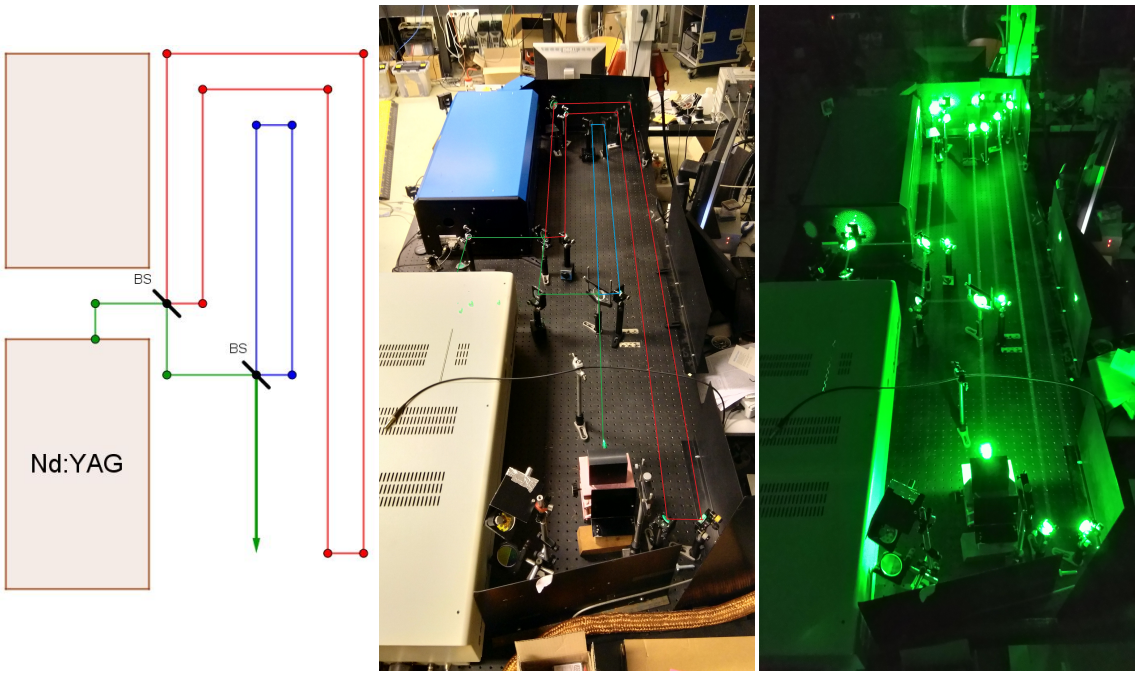


Figure 3.1: 5.32/3.24 Pulse stretcher setup, with the reflected beam in purple and the first and second cavities in red and blue, respectively.



(a) Schematic drawing (b) Laser inactive (c) With the laser active

Figure 3.2: The 8.02/3.06 pulse stretcher cavity. Shown (from left to right) schematically, without laser with the cavities marked and with the laser active. In the two left pictures, the green line shows the laser path directly to the output (without passing through the cavities), and the red and blue lines indicate the first and second cavities, respectively.

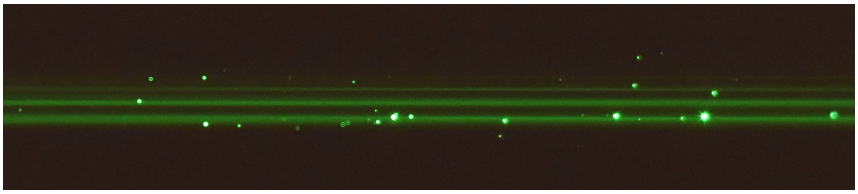


Figure 3.3: Picture made with the camera in the focal point setup. As multiple beams are visible, the alignment is not good enough here. The lowest beam is the base pulse, above it the later iterations of the pulse stretcher can be seen, slowly decreasing in power. The beam diameters are on the order of a few tenths of a millimeter.

size, the power transmission and the time profile. The transmission and time profile were also compared with calculated results, the script for which is explained first of all. Lastly, the pulse stretcher will be tested for breakdown.

Pulse stretcher code

To calculate the theoretical maximum transmission of the cavity, a matlab script was written. This script is shown and briefly explained in appendix B.2.

The script calculates the output of the pulse stretcher by regarding several iterations of the laser pulse through each cavity. It calculates the power each iteration has lost and the delay it has obtained. The output pulse is found by summing all iterations. The peak power of this pulse can be read out immediately, the transmission of the cavity is equal to the sum of all pulses.

The script can work with a perfect gaussian time profile with input FWHM, or it can read out a measured time profile and use that to determine the iterations of the cavities.

To be able to make an accurate estimate of the theoretical transmission of the pulse stretcher, the reflectivities of the mirrors and transmissions and reflectivities of the beam splitters must be determined.

To measure this, an Ophir L50(300)A-PF-65 thermal power meter [53] was used. For the sake of measurement accuracy, the transmission or reflectivity of an optical component was determined by measuring in front of and behind the component multiple times. This helped account for fluctuations in laser power and readout delay of the power meter.

Focal point setup

Testing the focal point size served to perfect the alignment of the pulse stretcher. The output of the pulse stretcher was focused with a 1 m focal length lens and eventually blocked with a power meter. A camera was focused on the focal point of the laser.

In the case of perfect alignment, each iteration of the beam overlaps perfectly. However, when the alignment of the pulse stretcher is not perfect, each iteration will attain a small offset. This offset is most noticeable in the focal point of the laser, where the beam itself is smallest. Using the camera, the focal point could be regarded closely to distinguish the iterations of the beams running parallel in the focal point simultaneously. What this looks like on the screen of the camera is shown in figure 3.3. By looking at the camera and tweaking the knobs on the last mirror in each cavity, these beams could be made to overlap.

If the alignment of the cavities is completely off, the alignment seems perfect on the camera because there are no extra beams. However, in that case the power drops drastically as well. By measuring the power at the output of the cavities, it could be verified that the alignment was indeed not terrible. To align the pulse stretcher in both the vertical and horizontal directions, the camera was turned to regard the focal point horizontally and vertically, respectively.

Transmission setup

The transmission of the entire pulse stretcher was measured like the transmission of the beam splitters, alternatingly placing the power meter at the in- and output of each cavity.

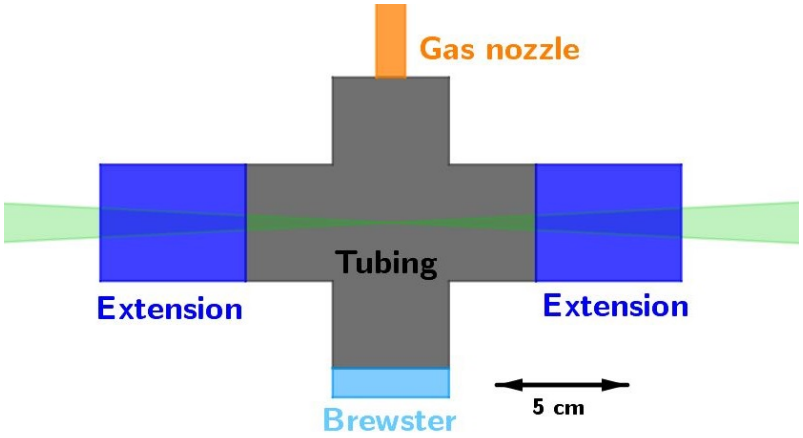


Figure 3.4: Schematic top view of the tubing used to create a dust-free setup to focus the laser in atmospheric air. The laser goes through from right to left. The gas comes in from behind and the opening on the opposite side is blocked with a Brewster window. On either side is an extra piece of metal tubing.

Time profile setup

For the time profiling measurements the power meter was placed at the output of the last cavity and the light from the laser spot collected by a Thorlabs DET025A/M 2 GHz high-speed photodiode. The signal of the photodiode was recorded with a LeCroy waverunner 44MXi-A 400 MHz oscilloscope. Both cavities were closed so only the base pulse shape would reach the photodiode. Cavities can be closed by simply interrupting the beam inside the cavity, for instance with a beam dump.

After the base pulse, the time profiles of the individual cavities and the full pulse stretcher were measured by opening either and both cavities. The photodiode was not moved in between the measurements.

Breakdown setup

To test whether the laser would induce breakdown, it would be focused to a point using lenses of various focal lengths. These tests were conducted in atmospheric air, to test a worst-case scenario with high pressure and dust.

To reduce the amount of dust around the focal point of the laser a small setup was devised that maintained a constant airflow on the focal point. The setup consists of a cross of metal piping, shown in figure 3.4. Two opposite sides of the cross were left open. Pressurised air was blown in from the back end of the cross, and the front end was blocked by a brewster window to increase the airflow speed to the sides. To increase the length of clean air, 2 extra pieces of piping were added to the open sides of the cross.

The laser was focused inside the tubing using lenses with focal lengths of 100, 65 and 30 cm. To determine the severity of breakdown for each situation, the number of breakdown events was counted over 10 minutes.

3.3 Plasma characterisation setup

To investigate the behaviour of the plasma under different circumstances, ICCD images of the plasma were taken. The pictures were then processed in a matlab script that determined the width and position of each individual streamer. From this data it could be determined what fraction of laser pulses would hit the streamer.

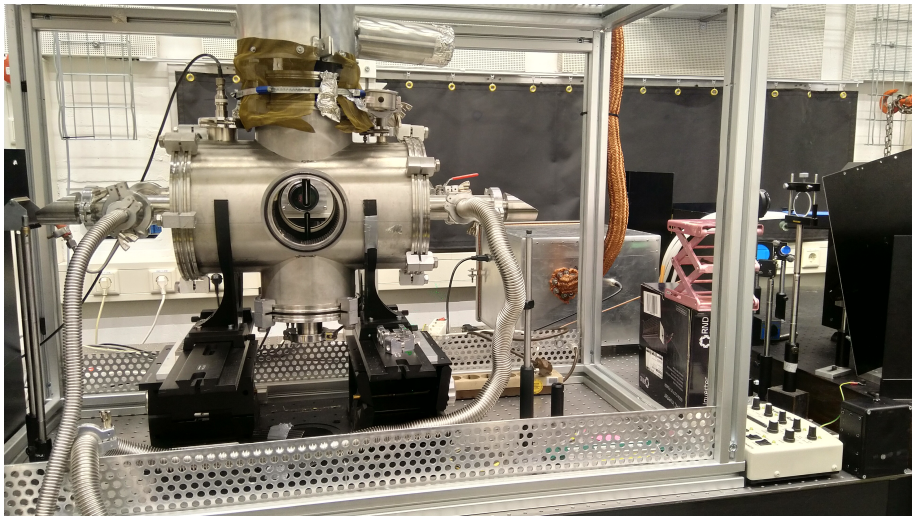


Figure 3.5: Picture of the vacuum vessel, as seen from the top right corner of figure 3.9

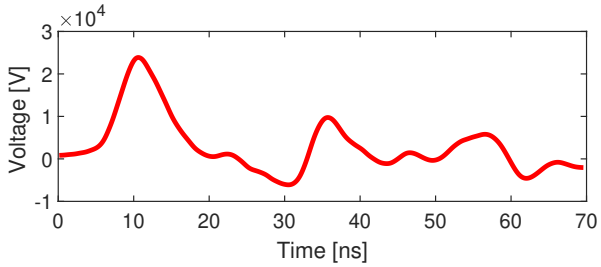


Figure 3.6: Voltage pulse shape measured over the electrodes

3.3.1 Plasma imaging

Figure 3.5 shows the vacuum vessel in which the plasma is created. In this setup the electrode distance, gas pressure, gas flow rate, gas composition and the timing of the plasma pulse w.r.t. the camera can be varied. The plasma operates in a burst mode at frequency f_{burst}, typically on the order of 90 Hz. Each burst consists of a train of N_{pulse} plasma pulses of 24 kV and 10 ns, given at 3 kHz, as shown schematically in figure 3.7. The plasma pulse is triggered using a Rigol DG1032Z pulse generator [54] and generated by a Megaimpulse NPG-18/3500 high voltage supply [55]. The shape of the voltage pulse is shown in figure 3.6 and the triggering configuration will be described in more detail in section 3.4. The plasma emission is sent through an Acton SP-500i 0.5 m spectrograph [56] in imaging mode. The emission is recorded by a PI-MAX 3 ICCD camera [57].

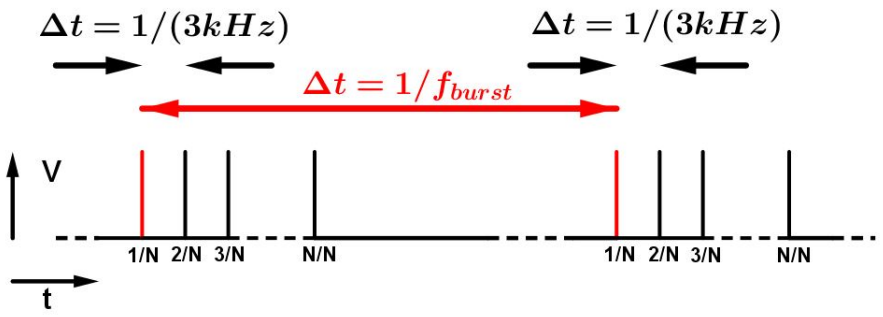
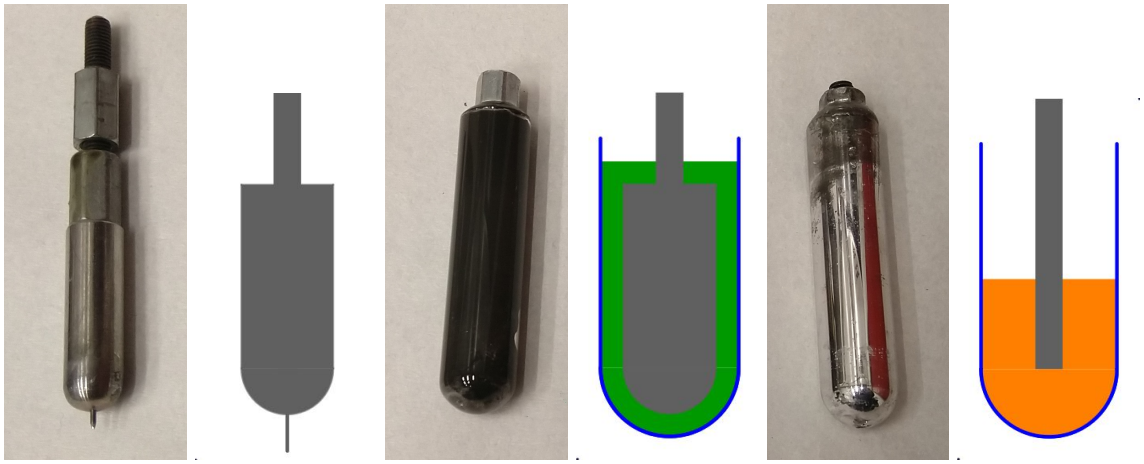


Figure 3.7: Schematic of the HV pulse pattern. The red line indicates the start of a burst of N pulses.



(a) Pin (b) Pin (c) Resin (d) Resin (e) Galinstan (f) Galinstan

Figure 3.8: Pictures and schematic drawings of the 3 electrodes used. In the schematic drawings, gray is stainless steel, blue is quartz, green is resin and orange is galinstan.

To compare the discharge and heating behaviour of metal-to-metal and dielectric barrier discharges, the top electrode can be exchanged for one of 3 different electrode types, shown in figure 3.8. The pin electrode, shown in figures 3.8a and b, is fully made of metal, with a sharp tip at the end. The first dielectric electrode consists of a stainless steel core, surrounded by quartz and kept in place with resin as shown in figures 3.8c and d. This electrode proved fragile under prolonged use - most likely as a result of heating of gas bubbles in the resin which caused stress on the quartz - and so for later experiments a new electrode was made. This electrode was filled with a liquid metal - Galinstan[1] - to increase the durability and make discharging easier w.r.t. the resin electrode. This electrode is shown in figure 3.8e and f.

The parameter under investigation were changed within a given range and for each electrode 100 pictures were taken for each combination of parameters. Pictures were taken with a gate width of 5 μs. The plasma images were processed in a matlab script where the streamer path, intensity, and width are determined automatically. From this, the number of laser-streamer hits can be calculated. The script is shown in appendix B.3.

3.4 Raman setup

A schematic floorplan of the setup for the raman measurements is shown in figure 3.9. Before entering the vacuum vessel, the laser is focused with a 1 m lens placed such that the focal point of the laser lies at the plasma position, as shown in figure 3.10. The viewing direction of the spectrograph is perpendicular to both the laser propagation direction and the electric field of the electrodes. Before entering the vessel, the laser is sent through a polarizer to ensure that the polarization of light is perpendicular to the viewing direction of the spectrograph. It enters the vacuum vessel through one of the brewster windows, and after exiting out the other brewster window is stopped by a beam dump.

For the Raman measurements several triggers have to be synchronised in time. This is achieved with two Rigol pulse generators [54]. These pulse generators have 2 outputs, each of which can take an external trigger. First, a 10 Hz signal is generated, which is sent to 2 other trigger inputs on the pulse generators. One of these creates the trigger for a burst at f_{burst}, which is then sent to the last trigger input where the 3 kHz bursts are created which trigger the HV supply. The other pulse is simply delayed in the pulse generator and consequently sent to the ICCD and laser. Through this delay, the laser and plasma pulse can be synchronised. The laser and spectrograph are synchronised by setting a delay in the WinSpec32 software used to operate the ICCD. The actual timing of the 3 components is monitored with a LeCroy WaveRunner 44MXi-A oscilloscope [58]. It

[1] Galinstan is an alloy of Gallium, Indium and Tin, and is commonly referred to as 'Bio mercury' due to its being liquid at room temperature but without the toxicity.

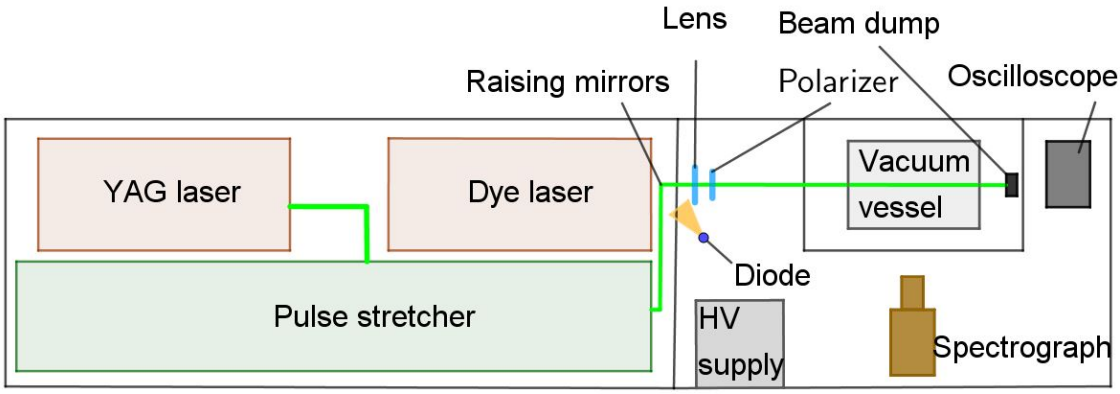


Figure 3.9: Schematic top view of the optical tables upon which the setup is built. The laser enters and exits the vacuum vessel through brewster windows. The vacuum vessel is encased in a metal cage to prevent EMC noise. The raising mirrors are a set of 2 mirrors angled at 45^o to get the laser to the height of the vacuum vessel. The orange cone in front of the diode indicates the viewing direction of the diode.

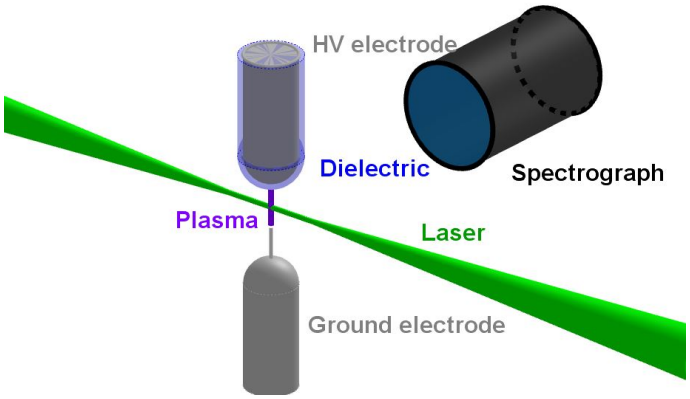


Figure 3.10: Schematic representation of the raman configuration

measures the monitor signal from the ICCD, the EMC noise from the plasma pulse, and the light emitted from one of the raising mirrors shown in figure 3.9, measured by the high-speed diode also shown there.

Measurements were done with a spectrograph slit width of 45 μm, a grating of 1200 grooves/mm and gate duration of 200 ns. Pictures were made with 600 gates per exposure at an ICCD gain of 100.

Two sets of thermometry measurements were done, one with the pin electrode and one with galinstan. To obtain dependencies of the temperatures on the parameters under investigation, each parameter was varied individually from a base set of parameters. The base parameter set is 700 mbar pressure, 0.75 cm/s flow speed, 13 μs delay, 90 Hz burst frequency and 7 pulses per burst, where the afterglow of the 4th pulse is regarded. For the pin electrode the base gap was set to 8 mm, for the galinstan electrode that gap was 4 mm.

Furthermore, as it was expected that many of the 600 accumulations of the measurements would be misses or failed ignitions, the obtained spectrum would not be entirely the plasma spectrum, but also a small portion the surrounding air. To quantify the effect of this 'dilution', the base spectrum was also obtained for both electrodes by taking 600 single measurements. These single measurements were analysed to find misses and failed ignitions, after which the temperature of the measurement was determined both with and without the misses included to find the impact of the dilution.

The script used to process the measurements is described briefly in appendix B.4. It will calculate the rovibrational temperatures in a small region around the position of the plasma. The measurements will be fitted to the Boltzmann-Treanor distribution, because it was found that the standard two-temperature Boltzmann distribution overestimated the populations of the lower vibrational states and underestimated the higher states. By accounting for the high and low vibrational states with the Treanor distribution, all states can be fit correctly.

The script also outputs a measurement accuracy which is based on a Monte Carlo model. Furthermore, the quality of the fits can be determined by taking the sum squared residual (SSR) between the measurement and the fit. This residual is calculated by summing over all measurement points i as

$$SSR = \sqrt{\sum_i \frac{(meas_i - fit_i)^2}{meas_i^2}} \tag{3.1}$$

Chapter 4

Results and discussion

4.1 Laser Results

4.1.1 Pulse stretcher measurements

Pulse stretcher code verification

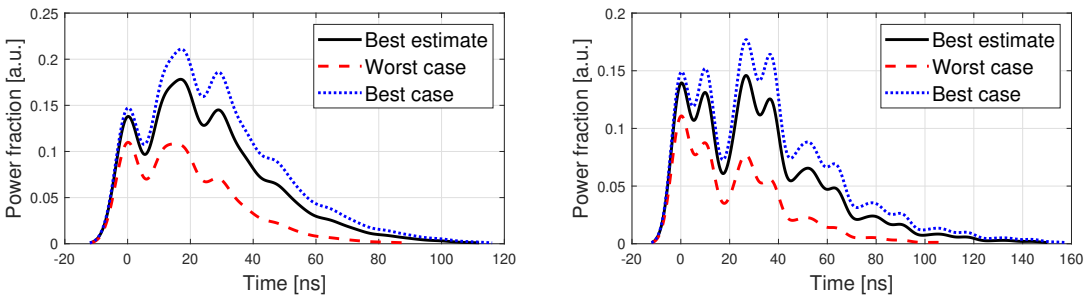
The transmissions and reflectivities of the beam splitters and mirrors are shown in table 4.1. The uncertainties are 5%, based on the uncertainty in the power measurement (4%) and the fluctuations in laser power (3%), combined in quadrature.

Object	Reflectivity	Transmission
Beam splitter	37 ± 2	60 ± 3
Mirror	99 ± 5	

Table 4.1: Measured transmissions and reflectivities of the beam splitters and mirrors in percentages

The beam splitters are rated for $40 \pm 3\%$ reflectivity [59], so the measured reflectivity comes out lower than expected though this is still within the stated error. The mirrors are rated at 99.0% reflectivity [60], so those seem to work as presented.

Figures 4.1 a and b show the calculated pulse profiles for the first and second iteration of the pulse stretcher. The best and worst estimate are the pulse profiles calculated with the reflectivity and transmission values set to their maximum or minimum estimated value. For the lower bound that means subtracting the uncertainty, for the upper bound the mirror reflectivity is set to 100% and the beam splitter is estimated at 38/62%.



(a) 5.32/3.24 m (b) 8.02/3.06 m

Figure 4.1: Calculated pulse time profiles for measured ranges of the transmissions of the optical components.

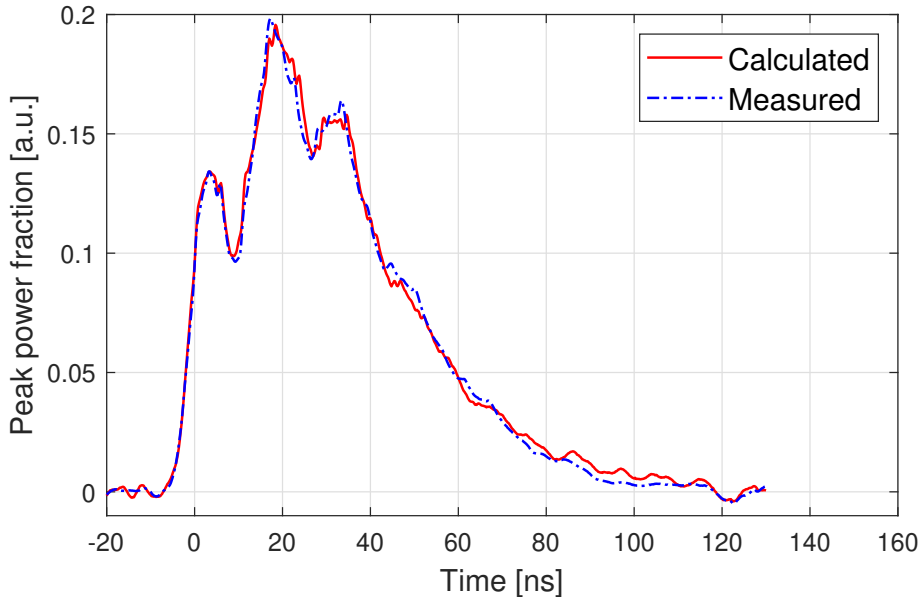


Figure 4.2: Measured and calculated time profile of the laser with the 5.32/3.24 m pulse stretcher. Input FWHM = 9 ns, output FWHM = 42.2 ns. P_{max} = 19.4%

Table 4.2 contains the calculated theoretical transmissions for each scenario. Since the mirrors and beam splitters will never have 100% efficiency, the upper bound is of course unreachable.

Estimate	5.32/3.24	8.02/3.06
Best estimate	79.7	79.4
Worst case	43.6	40.5
Best case	100	100

Table 4.2: Calculated transmissions for different estimates of the optical components in the different pulse stretchers.

Transmission measurements

Cavity	5.32/3.24	8.02/3.06
Full	87 ± 4	84 ± 4
1st	89 ± 4	91 ± 5
2nd	94 ± 5	93 ± 5

Table 4.3: Measured transmissions of both pulse stretchers and the individual cavities of each.

Table 4.3 contains the measured transmissions of the pulse stretchers. The measured transmissions of both pulse stretchers are found to be higher than their calculated best estimates. This is an indication that the mirrors and beam splitters may have higher efficiencies than measured.

Time profile measurements

Figures 4.2 and 4.3 show the measured versus calculated time profiles of the laser pulse for the two different pulse stretchers. Both show excellent agreement between the two curves, indicating that the pulse stretcher theory agrees well with experiments. Furthermore, it shows that losses resulting from bad alignment of the alignment are minimal, as those are not accounted for in the script.

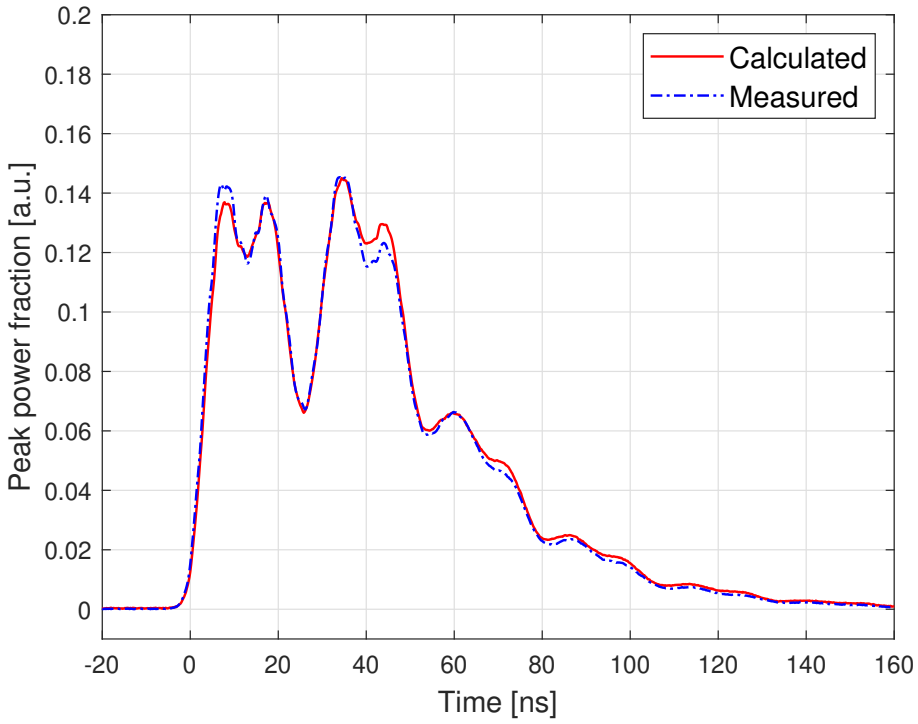


Figure 4.3: Measured and calculated time profile of the laser with the 8.02/3.06 m pulse stretcher. Input FWHM = 9.4 ns, output FWHM = 47.4 ns. $P_{max} = 14.5\%$

From these fits we find that the peak power of the laser with the 5.32/3.24 pulse stretcher has been reduced by 80.4%, and with the 8.02/3.06 pulse stretcher by 84.0%.

Breakdown measurements

The breakdown rates for both cavities are shown in table 4.4. As the frequency of the laser is 10 Hz, 6000 laser pulses occur within these 10 minutes.

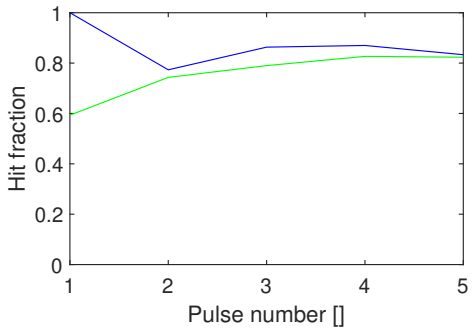
f (m)	1	0.65	0.3
5.32/3.24	20	48	6000
8.02/3.06	0	2	6000

Table 4.4: Number of breakdown events over 10 minutes for different focal lengths for the 5.32/3.24 and 8.02/3.06 cavities.

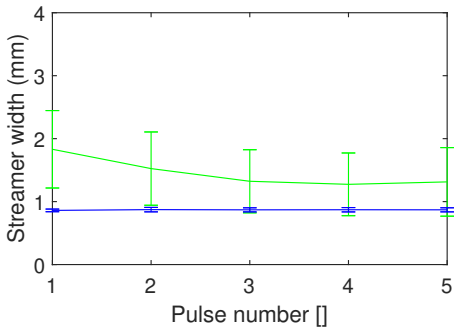
The $f = 0.3$ m lens caused breakdown at every pulse for both pulse stretchers. Unfortunately the longer focal length lenses still caused breakdown on a regular basis for the 5.32/3.24 pulse stretcher. Even the $f = 1$ m lens caused breakdown on average every 30 seconds, which is far too often to do Raman measurements.

For the 8.02/3.06 pulse stretcher, the 1 m lens to be used did not cause breakdown at all and with the $f = 0.65$ m lens breakdown was rare enough that Raman measurements would still be possible under these circumstances.

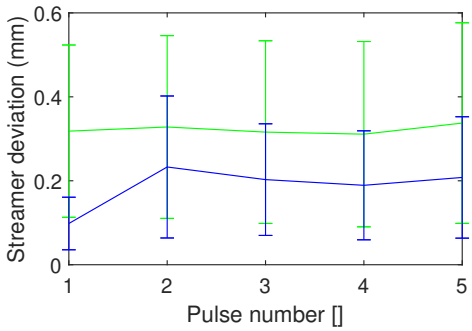
Furthermore, it should be noted that the absence of breakdown events in the work of Gijbels [1] does not conflict with the rate of breakdown found here. Their pulse stretcher suffered significantly greater energy losses, and thus had much lower pulse energies at the output. These likely contributed to a reduction in breakdown rate in their work.



(a) Hit rate



(b) Width



(c) Deviation

Figure 4.4: Plasma properties as function of pulse number at 700 mbar. Pin electrode is shown in blue, the solid resin electrode is shown in green. The error bars indicate a spread, not an uncertainty.

4.2 Plasma behaviour results

This section will show some of the results from the plasma behaviour measurements. Not all measurements will be presented, only those relevant for understanding the behaviour of the temperature presented in the next section. To reiterate, the variables under consideration are gap size, flow rate, pressure, pulse number and electrode type. The laser-streamer hit rate, and the width, deviation and intensity of the streamer have been determined.

First off, the measurements showed no variation of any of the properties with variable flow rate. As the flow was less than 1 cm/s, the surrounding air can probably be considered stationary for the duration of most plasma processes.

Figures 4.4 a through c show some plasma properties as function of pulse number for the pin and solid resin electrodes. Let us first regard the hit rate for both electrodes, shown in figure 4.4a. Starting from the second pulse, the hit rate is similar for both electrodes. However, the way these hit rates come to be is very different. The ignition rate for the pin electrode was 100% across all pulse numbers, but for the solid resin electrode, this followed a curve almost identical to the hit rate. The deviation of the streamers shown in figure 4.4c shows hardly no difference for solid resin, but the pin electrode shows a trend similar to the hit fraction upside-down. This implies that the hit rate of the solid resin electrode is a result of failed ignitions, whereas the hit rate of the pin electrode is a result of the spread in lateral position of the plasma. Figure 4.5 also illustrates this. For the solid resin electrode, the plasma emission in the first pulse is almost undetectable, whereas the second pulse shows a clear bright spot around the lower electrode. The pin electrode, however, shows a broadening from the first to the second pulse. The width of the individual streamers does not increase, and thus this must be a result of the increased deviation. Regardless, both show around 20% misses at the 4th pulse.

Furthermore, looking at the plasma intensity in figure 4.6b, a jump between the intensity of the first and second plasma pulse can be seen for the pin electrode. The intensity then decreases slightly from the 2nd to the 5th pulse and seems to tend towards an equilibrium intensity. The jump is

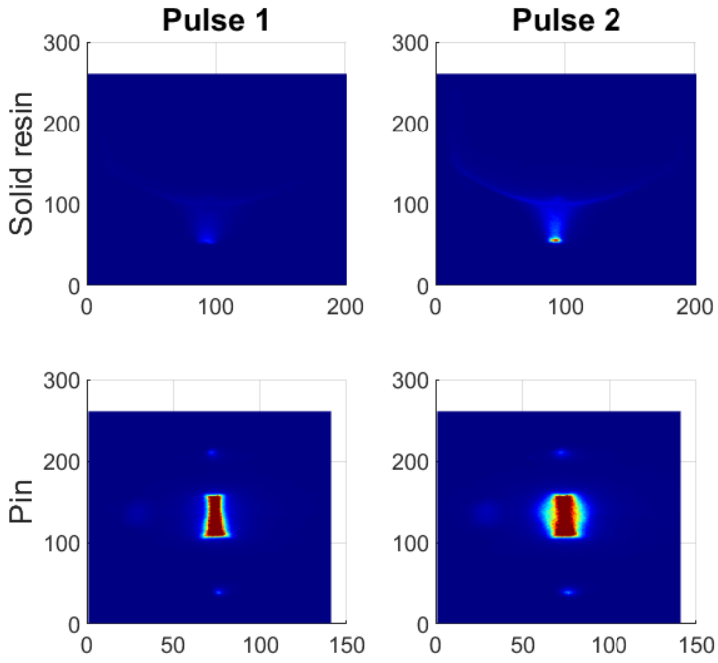
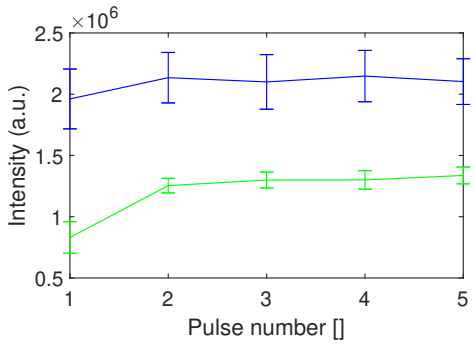


Figure 4.5: Comparison of the plasma emission between the first and second pulse of the pin and solid resin electrodes at 700 mbar. The images are averaged over 300 plasma pulses. The color scales have been set equal for all 4 pictures.

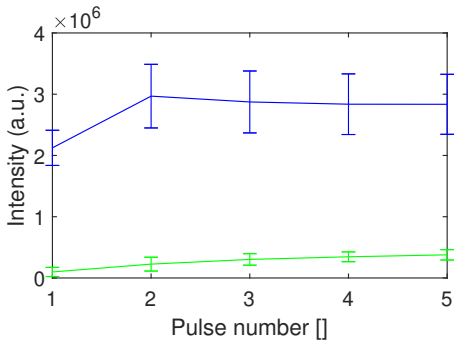
much less pronounced for lower pressure, as shown in figure 4.6a. The intensity is assumed to correlate with the temperature. With more energy in the plasma, more can be emitted in the form of radiation. At lower pressure, collision rates are much lower, which means that recombination and relaxation processes are also slowed down. As a result, the remaining energy (both in terms of heat and reactive species) will be larger at the beginning of the burst for lower pressure. Furthermore, because collision rates are increased, the equilibrium temperature of the plasma will increase for higher pressure, adding to the larger jump.

With the resin electrode, the increase in plasma emission is more likely related to ignition difficulty, where the discharges need to rely on the afterglow of previous pulses to ignite fully.

More thermalization behaviour is observed when regarding the streamer intensity as function of gap size. A plot of this is presented in figure 4.7. It shows that particularly for the first pulse, the plasma emission increases well over an order of magnitude when the gap size is decreased.



(a) 300 mbar



(b) 700 mbar

Figure 4.6: Plasma intensity as function of pulse number for different pressures. Pin electrode is shown in blue, the solid resin electrode is shown in green. The error bars indicate a spread, not an uncertainty.

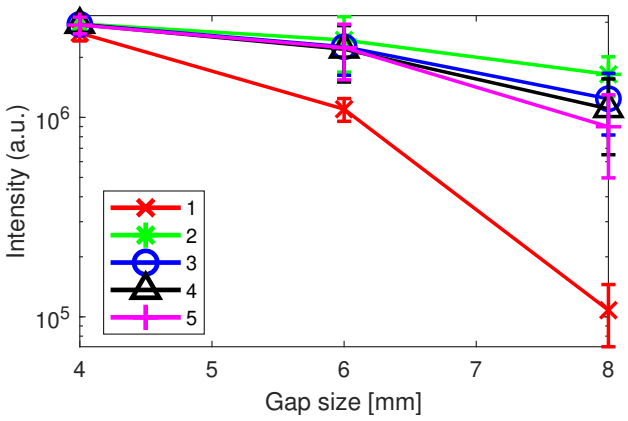
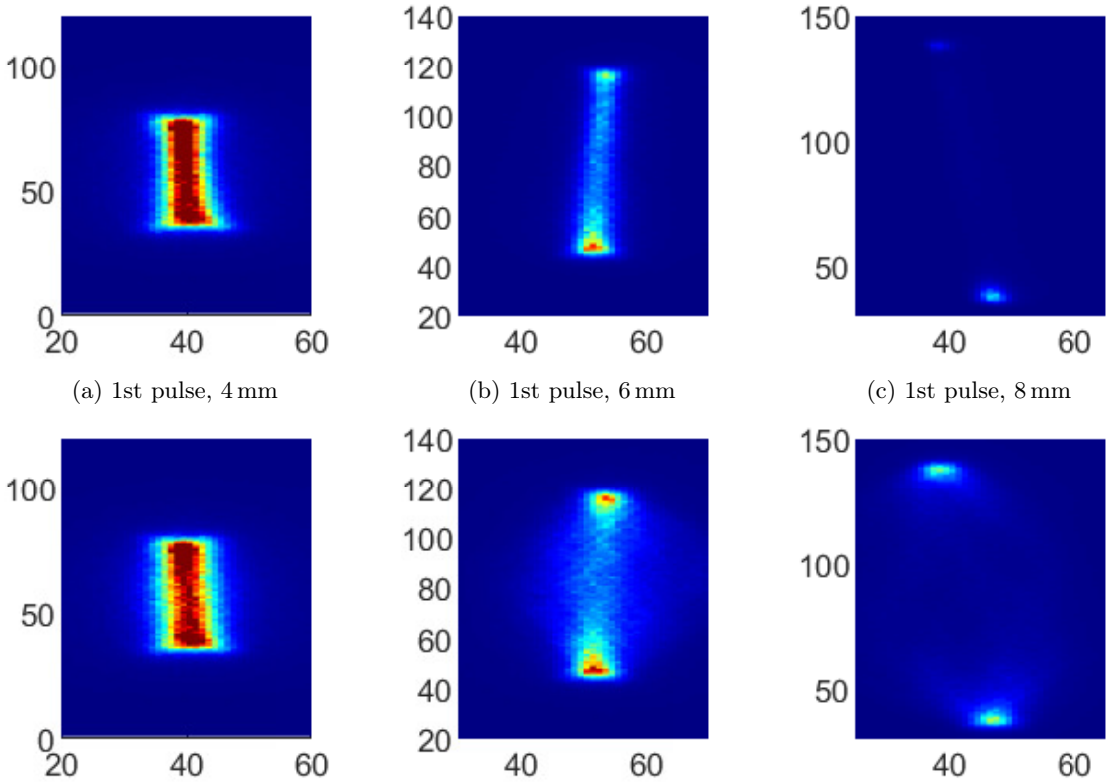


Figure 4.7: Total pin-to-pin streamer intensity as function of gap size for different pulses in a 5-pulse burst. Pressure = 700 mbar.

To be able to cross the discharge gap within the 10 ns of the HV pulse, a streamer would need to propagate at 0.4 and 0.8 mm/ns for the 4 and 8 mm electrode gaps, respectively. Since this is right in between the 0.1 to 1 mm/ns typical velocity of streamers briefly discussed in section 2.2, it stands to reason that the streamers sometimes will and sometimes won't be able to cross the discharge gap entirely. At 4 mm gap, the streamers will cross more often and thus spark more often than for longer gaps, and there will be more time left for plasma heating after sparking.

Figures 4.8 a through f show how the plasma changes for different gap sizes. At 8 mm gap, plasma emission is almost exclusively around the electrodes, similar to the emission of streamers. At 6 mm a plasma channel is clearly visible, although the bulk of the emission is still around the electrodes. This resembles the emission of a spark discharge or early arc discharge [36]. A full plasma channel has formed at 4 mm, akin to an arc discharge.



(a) 1st pulse, 4 mm (b) 1st pulse, 6 mm (c) 1st pulse, 8 mm

(d) 4th pulse, 4 mm (e) 4th pulse, 6 mm (f) 4th pulse, 8 mm

Figure 4.8: Pin electrode plasma emission for different gaps and pulse numbers at 700 mbar. The color scaling is equal for all 6 pictures.

4.3 Raman results

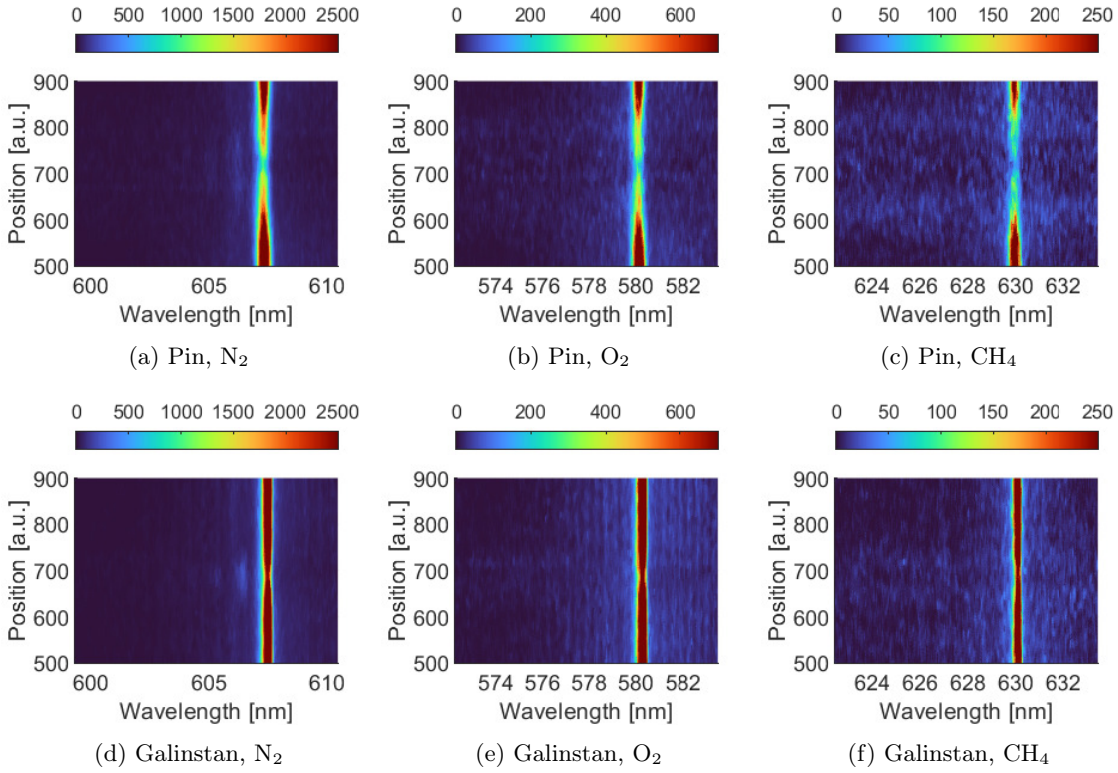
In this section, the measured and analysed Raman spectra will be discussed. Subsection 4.3.1 discusses a selection of the spectra in detail together with a qualitative correction to the measurements. Subsection 4.3.2 will demonstrate how the temperature changes as function of the different parameters. To reiterate, the parameters under investigation are pressure, delay, gap, burst frequency, flow and burst number. Specifically for burst number, a burst number noted as '4/7' regards the 4th pulse in a 7-pulse burst. The parameters were varied one by one from a base set of parameters, which is Pressure = 700 mbar, Delay = 13 μs, Burst frequency = 90 Hz, Flow = 0.75 cm/s, Pulse number =4/7. The base gaps are 4 and 8 mm for the galinstan and pin electrodes, respectively. In 4.3.3 the accuracy of the measurements is discussed, as well as some critical points that could influence the reliability.

4.3.1 Raman analysis

Figures 4.9 a through f show the base spectra of the pin and galinstan electrodes for nitrogen, oxygen and methane. Most peculiar about these figures is that in both nitrogen spectra the Raman emission from the higher vibrational levels is clearly visible around pixel 700, but for both O_2 and CH_4 this emission is not visible. Do note that the extra vertical lines around the v01 line in figure 4.9e are not vibrational bands, but most likely a consequence of incorrect flatfield correction.

It is not known whether the vibrational bands are only visible for nitrogen because the plasma vibrationally excites only nitrogen, or whether the bands are obscured for other species due to a decrease in the signal-to-noise ratio. More research is needed to determine this.

Figures 4.10 show the temperature distribution as function of position for both electrodes. This indicates that the v_{1v} temperatures may be inaccurate, as it is unlikely that these temperatures would be so constant far away from the plasma. We would expect the temperatures to decrease



(a) Pin, N_2 (b) Pin, O_2 (c) Pin, CH_4

(d) Galinstan, N_2 (e) Galinstan, O_2 (f) Galinstan, CH_4

Figure 4.9: Plasma spectrum in different wavelength ranges to measure the rovibrational temperatures of different gases for the pin and galinstan electrodes. Pressure = 700 mbar, Delay = 13 μs, Pulse 4/7, Flow = 0.75 cm/s

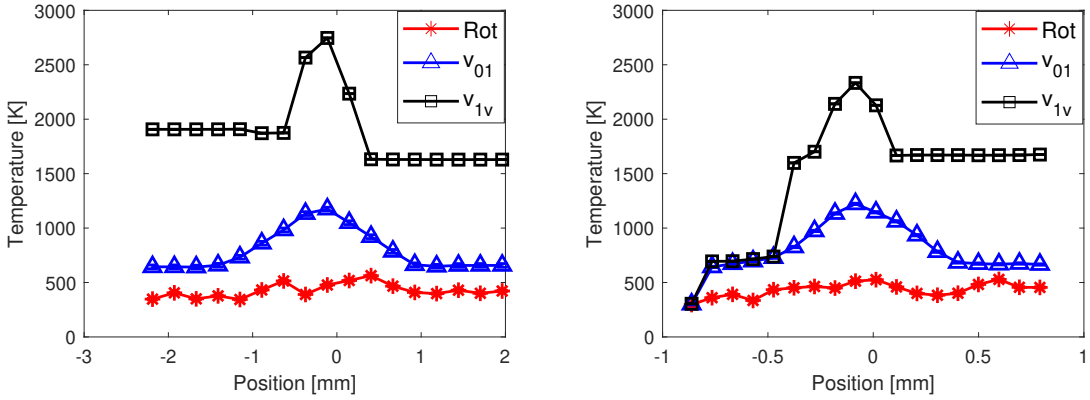
further out from the plasma. Furthermore, the difference between the temperatures to the left and right of the plasma in figure 4.10a is unlikely. Especially because the flow is so little, there would not be such a clear jump between the sides.

It is more likely that this is a result of the flatfield of the ICCD. Although they're much more significant in figure 4.9e, vertical lines resulting from the flatfield are visible in the nitrogen spectra shown in figure 4.9. The fitting script may perceive these lines as vibrational bands and fit accordingly, and only fit correctly when the actual high vibrational temperature is greater than the perceived temperature.

The v_{01} temperatures show the kind of trend that would be expected: A nice, smooth curve peaking around the plasma position. The rotational temperatures are to be questioned. One indication for this is the fluctuations observed, and another is the actual magnitude of the temperature. Both temperatures are on the order of 400 K all the way to the edge of the measured spectrum. This temperature would have been noticeable when removing the electrodes from the vacuum vessel. The electrodes themselves were warm to the touch, but the air around them was not. Since the fit of the rotational temperatures relies on the instrumental function of the spectrograph, it is possible that this instrumental function was simply incorrect. This can be corrected in the future by obtaining a spectrum of the rayleigh emission and determining the instrumental function from that.

Influence of misses

We will now briefly discuss the influence of misses or failed ignitions on the measured plasma. Table 4.5 shows a comparison between the different measurement methods. It shows that the vibrational temperatures are much lower for the single-shot measurements for both electrodes. The reason for that is shown in figure 4.11. The spectra derived from single images drop below 0 counts. This is most likely a result of incorrect removal of the background and flatfield. Due to this negative part, small peaks of the higher vibrational states are not detected and the temperature



(a) Pin electrode

(b) Galinstan electrode

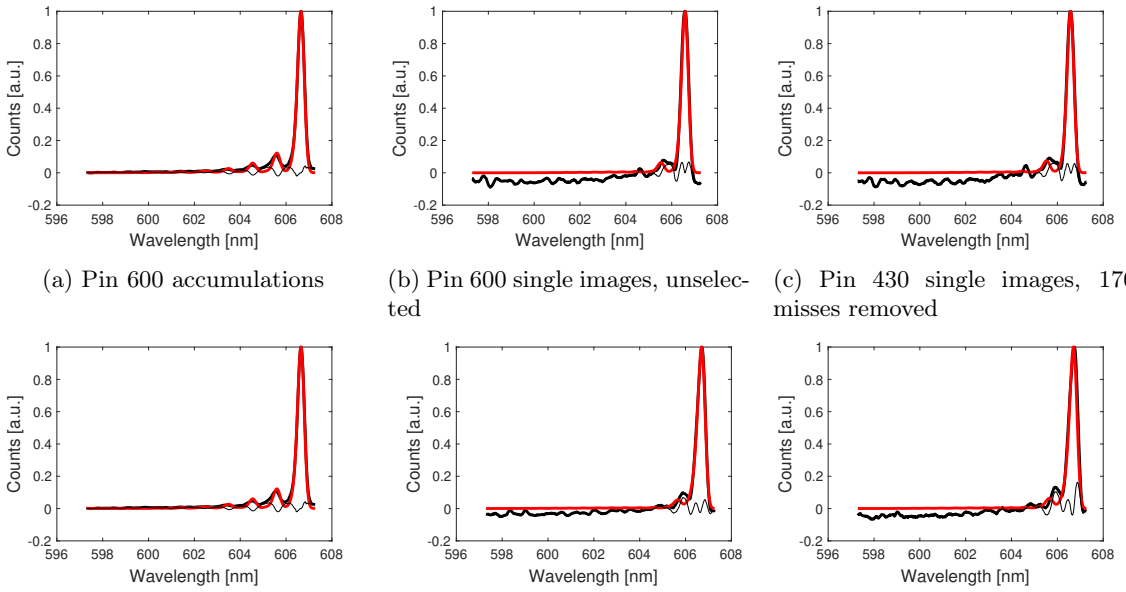
Figure 4.10: Temperature distribution along the laser (vertical direction in figures 4.9) for the base spectra of the pin and galinstan electrodes. The gas flow goes from right to left in both figures. The plasma is at position -1.

is perceived lower. The rotational temperature is not affected in the same way, as this is simply fitted to the shape of the rightmost peak. The rotational temperatures are overestimated for the Galinstan electrode, which is most likely a result of the small peak visible between the first and second peaks from the right. It is uncertain where this peak comes from, it might be a result of plasma emission.

Temperature	Electrode	600 accumulations	600 single shots	
			Misses removed	Full
T_{rot}	Pin	477 ± 7	460 ± 30	453 ± 5
	GaInSn	512 ± 7	700 ± 20	730 ± 10
T_{v01}	Pin	1170 ± 20	1000 ± 10	814 ± 3
	GaInSn	1222 ± 6	942 ± 3	891 ± 7
T_{v1v}	Pin	2746 ± 0.3	1000 ± 10	814 ± 3
	GaInSn	2334 ± 1	940 ± 30	891 ± 3

Table 4.5: Rovibrational temperatures for the base parameters of the pin and galinstan electrodes. The temperatures are calculated from 1 picture of 600 accumulations, 600 single-shot pictures, and 600 single-shot pictures from which misses have been removed. The errors indicate a variance, calculated using a MC model.

Despite the discrepancy between the two methods of measuring, we can see the influence that removing the misses has. In both cases, about 25% of the measurements are neglected - which is slightly more than the 20% determined in section 4.2 - and the vibrational temperatures are determined to be 23 and 6% greater for respectively the pin and galinstan electrodes, when misses are removed. This difference between the corrections for the two electrodes is unexpected. Since both the pin and resin electrode had approximately 20% misses, it was presumed that the correction to the temperatures would be on the same order. However, the misses on the resin electrode were mostly a result of failed ignitions. Since the Raman measurements were done with the galinstan electrode, there was probably a significantly lower number of failed ignitions, resulting in a much smaller temperature correction. The high number of removed measurements could simply be a consequence of the script being too strict about what constitutes a hit or a miss. Figures 4.11 b, c, e and f show that the temperature correction is not just an artefact of the measurement. The intensity of emission from the first 2 vibrationally excited states is actually increased for both electrodes. This goes to show that the temperature is strongly underestimated when not accounting for misses. Nonetheless, more work is needed to be able to obtain reliable single-shot measurements and determine corrections to the temperatures more accurately.



(a) Pin 600 accumulations

(b) Pin 600 single images, unselected

(c) Pin 430 single images, 170 misses removed

(d) Galinstan 600 accumulations

(e) Galinstan 600 single images, unselected

(f) Galinstan 444 single images, 156 misses removed

Figure 4.11: Measured (black) and fitted (red) central spectral line for the base spectra for the pin and galinstan electrodes for different ways of obtaining the spectra. The thin black line is the difference between the two curves.

4.3.2 Parameter dependencies

Due to the relatively low flow velocities investigated, it was expected that flow rate would not influence the temperature significantly. This was indeed observed and hence the data is not presented here for the sake of brevity.

For PAC, the most relevant information is how the heat from the plasma lingers between plasma pulses. Figure 4.12 shows how the different temperatures decay directly after the plasma pulse. It is clearly visible how the temperatures decay on different timescales. For both electrodes, the higher vibrational states decay within several microseconds after the plasma pulse, during which the lower vibrational and rotational states are populated. Furthermore, it can be seen that the rotational temperature steadily increases after the plasma pulse with the pin electrode, but not with the Galinstan electrode.

Across the plasma burst, the temperature can be observed to rise steadily for the galinstan electrode in figure 4.13a. For the pin electrode, shown in figure 4.13b, the situation is much more complex.

Pulse 1/7 is significantly hotter than pulse 1/4, but 4/4 shows much higher temperatures than 4/7. It was expected that more pulses in a burst would equal higher temperatures that lingered longer, but this shows the opposite. Pulse 1/7, 1/4 and 4/4 also showed significantly more plasma emission than the other 3 pulses. Particularly figure 4.14d does not tend towards 0 for low wavelengths. It shows the random broadband emission characteristic for an arc plasma.

The question is why plasma heating occurs for more pulses when the total number of pulses per burst is lower. It seems more intuitive that more pulses per burst equals more heat. And this is in part true. Figure 4.13b shows that the first pulse transfers much more energy to the plasma - both vibrational and rotational - in the 7-pulse burst than the 4-pulse burst. In particular, the rotational and vibrational temperatures are identical in the first of the 7-pulse burst, implying that the plasma is becoming thermalized and arcing, whereas the 4-pulse burst stays in the spark regime. When run with a continuous current, an arc sustains itself through a coupling between heating and ionisation. However, when the arc is repeatedly turned on and off, the stable channel is interrupted and allowed to diffuse, which will happen quickly due to its high temperature. That means that for a second pulse, there will be no clear plasma channel or a high density of particles

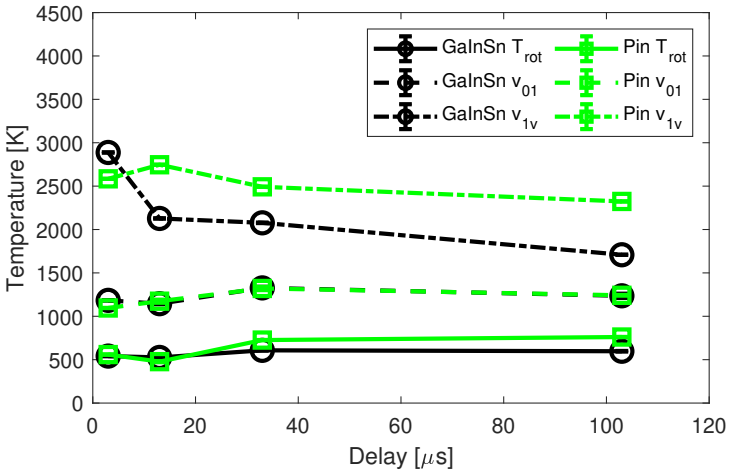
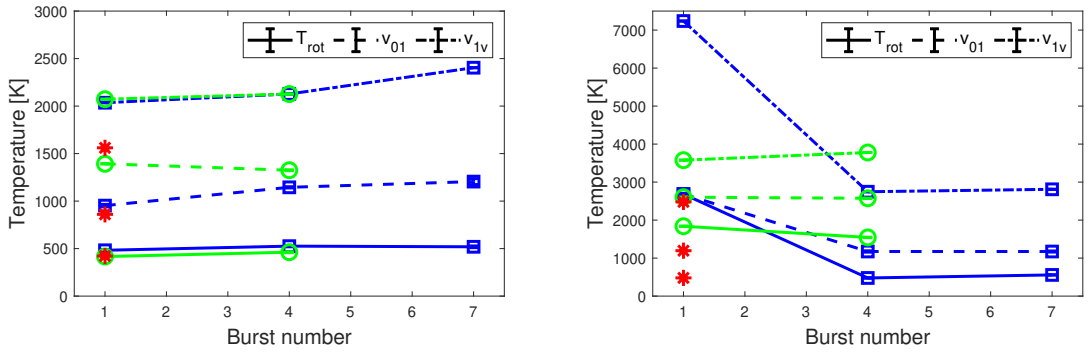
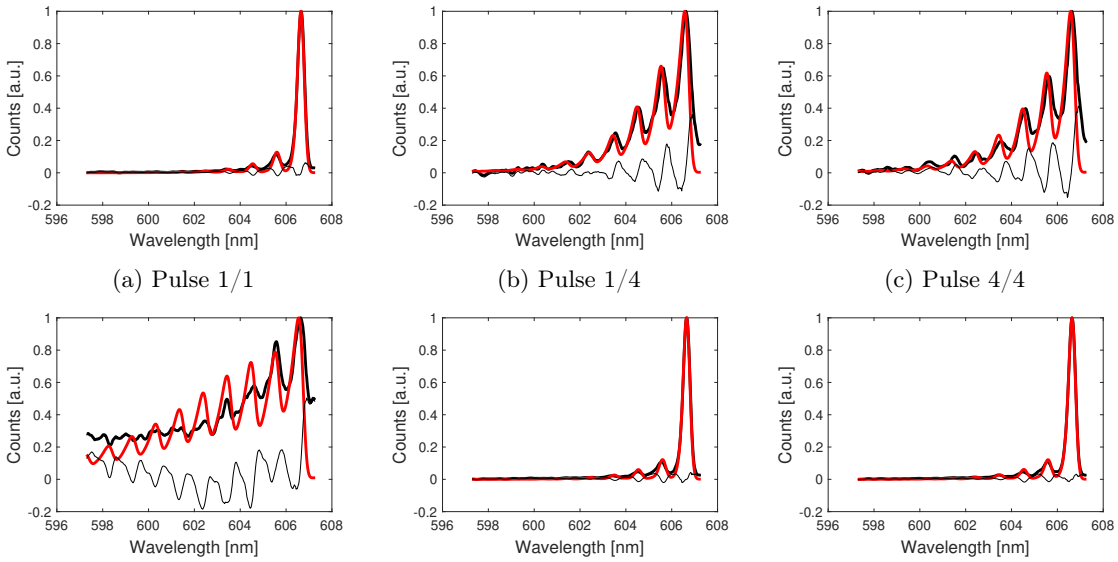


Figure 4.12: Rovibrational temperatures as function of delay between plasma pulse and laser for galinstan and pin electrode. Pressure = 700 mbar, Burst frequency = 90 Hz, Flow = 0.75 cm/s, Pulse number =4/7.



(a) Galinstan (b) Pin

Figure 4.13: Rovibrational temperatures as function of pulse number for the pin and galinstan electrodes. The red, green and blue curves indicate $N_{burst} = 1, 4, 7$, respectively. Pressure = 700 mbar, Delay = 13 μs, Burst frequency = 90 Hz, Flow = 0.75 cm/s



(a) Pulse 1/1 (b) Pulse 1/4 (c) Pulse 4/4

(d) Pulse 1/7 (e) Pulse 4/7 (f) Pulse 7/7

Figure 4.14: Measured (black) and fitted (red) spectra in the centre of the pin electrode plasma for different pulse numbers. The thin black line is the difference between the two curves. Pressure = 700 mbar, Delay = 13 μs, Burst frequency = 90 Hz, Flow = 0.75 cm/s, Gap = 8 mm

to ionise and heat, such that heating is impeded.

Though more investigation is needed, the mechanism is proposed as follows:

In a 7-pulse burst, the later pulses produce a non-thermal plasma. This leaves a lot of excited species combined with high density, allowing the first pulse of the next burst to arc fully. This arc, however, impedes arcing on the second pulse, which will consequently be non-thermal again. In the 4-pulse burst, there will not be enough energy at the beginning of the burst for the first pulse to arc, leaving the plasma somewhere between non-thermal and arcing. The plasma does not have enough energy to arc fully, but too much energy to relax to a non-thermal state.

To verify this hypothesis, a more detailed map of the temperature as function of time is needed. Besides measurements after each pulse in the burst, not just a selection of pulses, it can be interesting to measure the temperatures right before the plasma pulse to show the gas conditions the plasma pulse discharges into.

Measuring the temperature right before the first pulse in a burst could also be interesting in regards to burst-to-burst heating. The influence of heating between bursts can be gauged by changing the burst frequency. The measured temperatures for both electrodes at 30 and 90 Hz burst frequency are shown in table 4.6. For both electrodes we see the difference between 30 and 90 Hz decrease for the higher vibrational temperatures. However, for the Galinstan electrode the temperatures are higher at 90 Hz than at 30 Hz, whereas for the pin electrode they are consistently lower. This implies that the Galinstan electrode benefits from remainder heat for its heating. The pin electrode seems to impede its own heating, which fits with the observations made previously about pulse number concerning figures 4.13 and 4.14.

Figure 4.15 shows that the temperatures as function of gap remain largely of the same order, except for one point. At 4 mm gap size, the pin electrode produces a much hotter plasma. Most notably, the lower vibrational temperatures are equal to the rotational temperature. This means that the plasma has started thermalizing and has begun to form an arc. That is not unexpected. A shorter gap distance means less time needed for the streamer to cross and thus more time left for gas heating. This measurement showed a lot of plasma emission, which is also the reason why the measurements with the pin electrode were done with an 8 mm gap rather than the 4 mm gap used for the galinstan electrode. Figure 4.16 shows the difference between the full spectrum at 4 and 8 mm. Figure 4.16a shows that even after background subtraction - which should also account for plasma emission - the plasma emission is still prevalent. Figure 4.17 shows the spectrum of

Temperature	Electrode	30 Hz	90 Hz	% increase
T_{rot}	Pin	543	477	-12.2
	GaInSn	421	526	24.9
T_{v01}	Pin	1196	1174	-1.8
	GaInSn	1052	1145	8.8
T_{v1v}	Pin	2747	2746	0.0
	GaInSn	2211	2128	3.6

Table 4.6: Measured rovibrational temperatures for the pin and galinstan electrodes, as well as the increase in temperature from 30 to 90 Hz.

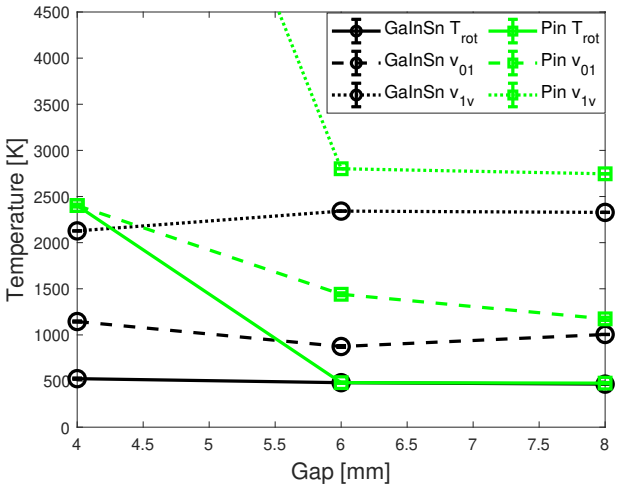


Figure 4.15: Rovibrational temperatures as function of gap for different electrode types. Pressure = 700 mbar, Delay = 13 μs, Burst frequency = 90 Hz, Flow = 0.75 cm/s, Burst number = 4/7.

the plasma emission for 2 different pressures. Note that this image accounts for the background and flatfield of the spectrograph. An excess of plasma emission will thus result in a significant amount of noise and due to the curved shape of the emission can obscure raman emission from higher vibrational levels. The rovibrational temperatures of these kinds of measurements are thus unreliable, although high plasma emission is still an indication of high temperature.

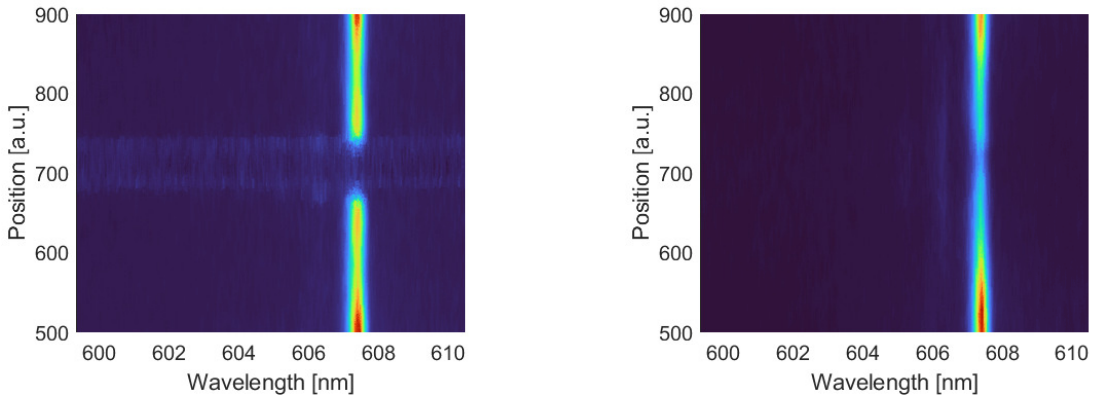
Lastly, as function of pressure the higher vibrational states show the most significant variations. The lower vibrational and the rotational temperature appear to increase slightly. The different temperatures are expected to come closer together for higher pressures, as higher pressure means higher number density, thus more collisions and thus faster thermalization.

4.3.3 Measurement accuracy

It was already mentioned that misses and failed ignitions influence the measured temperature. This section will not discuss such influences, but only regards the inaccuracy and fluctuations in the measurements.

The Raman fitting script contained a Monte Carlo model which calculated a variance in the fitted temperature. The average uncertainties in the measurements are shown in table 4.7. The mean variances are minimal with respect to the measured temperatures which were no lower than 300 K. No variances greater than 40 K were found. It should be noted that the variances for the rotational temperature of Galinstan are on average a factor 2 higher than for the pin electrode. An important contributor to this could be the overall decrease in signal, thus resulting in relatively more noise and thus more sensitivity to fluctuations.

The SSRs of the measurements are also all very low. Apart from 4 extremities which are all measurements with excessive plasma emission, the highest SSR found is 5.0%, the average SSR $2.1 \pm 0.9\%$. The SSR of the measurements with high plasma emission was found to be $17 \pm 2\%$. The



(a) Pin 4 mm

(b) Pin 8 mm

Figure 4.16: Measured spectra for the pin electrode with different gap sizes.

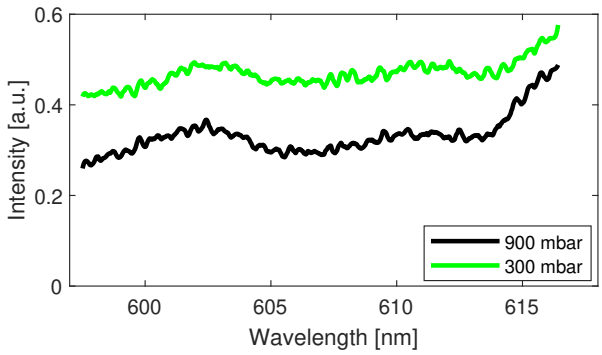


Figure 4.17: Emission spectrum of the plasma at base parameters, at 300 and 900 mbar with the pin electrode.

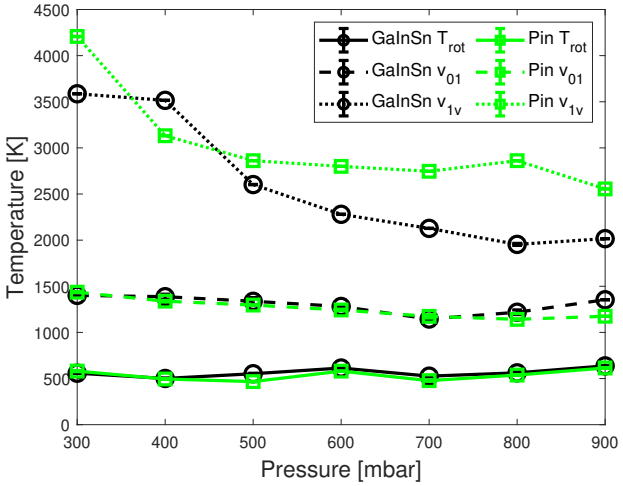
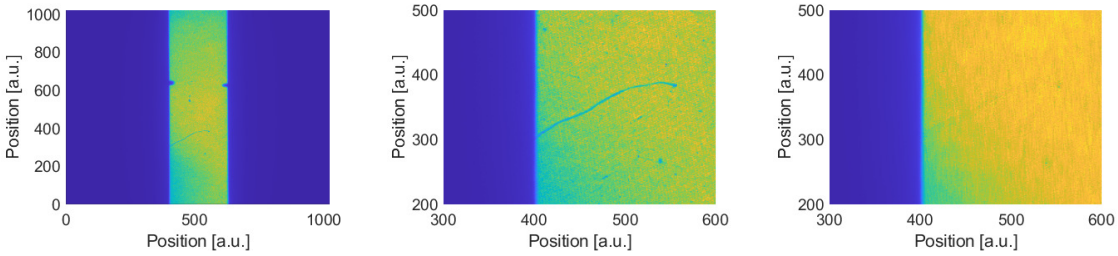


Figure 4.18: Rovibrational temperatures as function of pressure for different electrode types. Delay = 13 μs, Burst frequency = 90 Hz, Flow = 0.75 cm/s, Burst number = 4/7.

Temperature	σ (Galinstan)	σ (Pin)
Rotational	6.8	3.3
v01	4.8	5.1
v1v	8.4	8.4

Table 4.7: Average variances (in K) in the fitted temperatures for each electrode



(a) Image containing the hair (b) Zoomed in on the hair (c) Hampel filtered

Figure 4.19: ICCD image showing a hair on the ICCD. The position indicates the pixel number on the ICCD.

4 measurements with high plasma emission are the 4 mm pin electrode and the 3 aforementioned burst number measurements.

The Raman measurements were performed in a different lab than all the previous measurements. Due to temperature fluctuations in this new lab, the input laser power of the YAG fluctuated quite significantly. Furthermore, the maximum power achievable was roughly 12% lower in the new lab: Between 6.5 and 7 W, such that the output pulse energy of the pulse stretcher is between 550 and 590 mJ. If the temperature fluctuations caused the laser power to drop more than 25%, this would be visible on the spectrograph and so it is known for certain that the output pulse energy of the pulse stretcher is greater than 400 mJ for all measurements. However, fluctuations between 400 and 600 mJ will not be distinguished.

That being said, a decrease in pulse energy will not change the Raman spectrum. It will only decrease the SNR, which means the variance and SSR might be slightly raised for those particular measurements. The measured temperatures should then be largely unaffected.

During the measurements it was found that there was a hair on the ICCD. The hair is shown in figures 4.19 a and b. Due to time constraints it was not possible to remove the hair and redo the measurements. However, the hair most likely did not influence the measurements. As shown in figure 4.19b, the hair lies below pixel 400 in the vertical direction. The electrodes have at all times been above pixel 600 in the vertical direction, and thus the spectrum of the plasma will have remained largely unperturbed. Furthermore, all images were hampel filtered before fitting. Figure 4.19c demonstrates that the hampel filter removes the hair. It is thus deemed highly unlikely that the hair had a noticeable influence on the temperature measurements.

A note should also be made about the influence of the laser on the plasma and vice versa. Although Raman spectroscopy is certainly a less intrusive method of thermometry than inserting a probe into the plasma, it proved still somewhat intrusive. Whilst the breakdown rate had become negligible in atmospheric air, it was found that breakdown did occur more regularly inside the vacuum vessel when the plasma was active with metal electrodes. This indicates that the laser does transfer a significant amount of energy to the plasma afterglow. Conversely, it was found that the plasma ignited more easily for the galinstan electrode when the laser was active, suggesting that the laser does produce active species and seed electrons that improve plasma ignition.

Lastly, it is important to consider whether the streamers were positive or negative. The bias on the powered electrode is positive. When both electrodes have a sharp tip, the easiest electrode to form the streamer from is the powered electrode, as this would produce a positive streamer. However, with the DBDs installed the lower (negative) electrode is much sharper and thus produces much more local field enhancement. This would make streamer inception on the lower electrode much

easier, and thus cause a tendency to produce more negative streamers. In future research it can be interesting to make the bias on the upper electrode negative, such that most streamers will start from the lower electrode as positive streamers.

Chapter 5

Conclusion

In this thesis, rovibrational Raman thermometry was performed on nanosecond pulsed filamentary discharges.

To increase the Raman signal whilst preventing optical breakdown, two pulse stretchers were constructed. One 5.32/3.24 m and one 8.02/3.06 m, which showed total transmissions of 87 ± 4 and $84 \pm 4\%$, respectively. The 5.32/3.24 pulse stretcher showed a breakdown rate too high to do Raman spectroscopy with, but the 8.02/3.06 pulse stretcher had a sufficiently low breakdown rate when focused with an $f = 1$ m lens.

Using ICCD imaging, the arcing behaviour and instability of the streamer channel were investigated. As expected, it was found that the plasma had much more trouble forming with the dielectric barrier configuration than with exposed electrodes. This results in the plasma benefiting noticeably from the presence of active species and becoming brighter with pulse number as a result with the DBD electrode. However, the exposed electrodes demonstrated much more complex behaviour, likely due to interaction between the pulses that promotes or inhibits heating and breakdown.

The Galinstan electrode designed for this thesis was promising too. Plasma ignition is much better than with resin electrodes and the electrode does not break upon heating. The construction of Galinstan electrodes should be done with caution as the strong surface tension and adhesion of Galinstan can cause air cavities that break the symmetry of electrodes and change discharge properties.

Most of the Raman measurements agreed well to the fitted model, apart from an almost definite overestimation of the rotational temperatures. This can be overcome with Rayleigh measurements. The influence and importance of correcting for misses and failed ignitions has been shown. It will require more work to make more accurate corrections to the measured temperatures. Another observation that bears further investigation is the difference in Raman spectra between N_2, O_2 and CH_4. No Raman signal from higher vibrational levels was measured for oxygen and methane under circumstances where nitrogen did show this signal. Although the possibility that this is a measurement error remains, it can be interesting to investigate because if this is not a measurement error it teaches us about how different species are heated differently under the same conditions.

Exemplary parameter dependence measurements were done, illustrating the relaxation of high-energy particles to lower energies, increased non-thermal behaviour for lower pressure and unexpected thermalization behaviour for different burst numbers.

For the bulk of the measurements, the plasma produced with the pin electrodes was non-thermal, showing the effect of short pulses on gas heating. However, there were still some measurements that started transitioning to a thermal arc plasma. The Galinstan electrode showed similar non-thermal behaviour, except no plasma thermalized for this electrode. Rotational temperatures with the Galinstan electrode stayed mostly on the same order below 500 K, whereas vibrational temperatures up to 2000 K could be reached. This displays the strength of using a DBD to produce non-thermal plasmas.

Bibliography

[1] Timothy Gareth Gijbels. Laser pulse stretching for spontaneous raman spectroscopy of plasma assisted combustion. Master's thesis, Eindhoven University of Technology, 2020.

[2] Taylor & Francis Group. Handbook of chemistry and physics online, 2020. accessed: 25-01-2021.

[3] UNFCCC. Paris agreement. In *United Nations Framework Convention on Climate Change*, November 2015.

[4] V. Masson-Delmotte, P. Zhai, H. O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J. B. R. Matthews, Y. Chen, X. Zhou, M. I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield. Global warming of 1.5°c. an ipcc special report on the impacts of global warming of 1.5°c above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty. *In Press*, Oct 2018.

[5] Gert Jan Kramer and Martin Haigh. No quick switch to low-carbon energy. *Nature*, 462(7273):568–569, 2009.

[6] Craig T Bowman. Kinetics of pollutant formation and destruction in combustion. *Progress in energy and combustion science*, 1(1):33–45, 1975.

[7] Gabriele Cipriani, Sabrina Danti, Cecilia Carlesi, and Gemma Borin. Danger in the air: air pollution and cognitive dysfunction. *American Journal of Alzheimer's Disease & Other Dementias®*, 33(6):333–341, 2018.

[8] Jan E Jonson, Jens Borken-Kleefeld, Deighton Simpson, Agnes Nyíri, Maximilian Posch, and Chris Heyes. Impact of excess nox emissions from diesel cars on air quality, public health and eutrophication in europe. *Environmental Research Letters*, 12(9):094017, 2017.

[9] Yiguang Ju and Wenting Sun. Plasma assisted combustion: Dynamics and chemistry. *Progress in Energy and Combustion Science*, 48:21–83, 2015.

[10] Anatoly Klimov, Valentin Bityurin, A Kuznetsov, B Tolkunov, N Vystavkin, and M Vasiliev. External and internal plasma-assisted combustion. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*, page 1014, 2004.

[11] Guillaume Pilla, David Galley, Deanna A Lacoste, Franois Lacas, Denis Veynante, and Christophe O Laux. Stabilization of a turbulent premixed flame using a nanosecond repetitively pulsed plasma. *IEEE Transactions on Plasma Science*, 34(6):2471–2477, 2006.

[12] Wenting Sun and Yiguang Ju. Nonequilibrium plasma-assisted combustion: a review of recent progress. *J. Plasma Fusion Res*, 89(4):208–219, 2013.

[13] Ahmed Khacef, Jean Marie Cormier, and Jean Michel Pouvesle. Nox remediation in oxygen-rich exhaust gas using atmospheric pressure non-thermal plasma generated by a pulsed nanosecond dielectric barrier discharge. *Journal of Physics D: Applied Physics*, 35(13):1491, 2002.

[14] Manabu Higashi, Satoshi Uchida, Nagatoshi Suzuki, and Kanichi Fujii. Soot elimination and no/sub x/and so/sub x/reduction in diesel-engine exhaust by a combination of discharge plasma and oil dynamics. *IEEE transactions on plasma science*, 20(1):1–12, 1992.

[15] Chang Ming Du, Jian Hua Yan, and Bruno Cheron. Decomposition of toluene in a gliding arc discharge plasma reactor. *Plasma Sources Science and Technology*, 16(4):791, 2007.

[16] Min Suk Cha, SM Lee, KT Kim, and Suk Ho Chung. Soot suppression by nonthermal plasma in coflow jet diffusion flames using a dielectric barrier discharge. *Combustion and flame*, 141(4):438–447, 2005.

[17] Vladimir N Ochkin. *Spectroscopy of low temperature plasma*. John Wiley & Sons, 2009.

[18] Sander Nijdam, Jannis Teunissen, and Ute Ebert. The physics of streamer discharge phenomena. *arXiv preprint arXiv:2005.14588*, 2020.

[19] M Šimek. Optical diagnostics of streamer discharges in atmospheric gases. *Journal of Physics D: Applied Physics*, 47(46):463001, 2014.

[20] A Lo, G Cléon, P Vervisch, and A Cessou. Spontaneous raman scattering: a useful tool for investigating the afterglow of nanosecond scale discharges in air. *Applied Physics B*, 107(1):229–242, 2012.

[21] Magesh Thiyagarajan and Shane Thompson. Optical breakdown threshold investigation of 1064 nm laser induced air plasmas. *Journal of Applied Physics*, 111(7):073302, 2012.

[22] Jun Kojima and Quang-Viet Nguyen. Laser pulse-stretching with multiple optical ring cavities. *Applied optics*, 41(30):6360–6370, 2002.

[23] L_L Wang, ZA Munir, and Yu M Maximov. Thermite reactions: their utilization in the synthesis and processing of materials. *Journal of Materials Science*, 28(14):3693–3708, 1993.

[24] James G Quintiere. *Principles of fire behavior*. CRC Press, 2016.

[25] Chung K Law. *Combustion Physics*. Cambridge University Press, 2016.

[26] Dong-Shuo Ma and ZY Sun. Progress on the studies about nox emission in pfi-h2ice. *International Journal of Hydrogen Energy*, 45(17):10580–10591, 2020.

[27] N_K Rizk and HC Mongia. Semianalytical correlations for nox, co, and uhc emissions. *Engineering for gas turbines and power*, 1993.

[28] E Zervas, X Montagne, and J Lahaye. Influence of fuel and air/fuel equivalence ratio on the emission of hydrocarbons from a si engine. 1. experimental findings. *Fuel*, 83(17-18):2301–2311, 2004.

[29] Bruce Sween Liley, Simeon Potter, and Michael C. Kelley. Plasma. Encyclopedia Britannica, march 2021.

[30] Friedrich Paschen. Ueber die zum funkenübergang in luft, wasserstoff und kohlensäure bei verschiedenen drucken erforderliche potentialdifferenz. *Annalen der Physik*, 273(5):69–96, 1889.

[31] Krishnavedala. Paschen curves, 9 2014.

[32] Leonard B Loeb and Arthur F Kip. Electrical discharges in air at atmospheric pressure the nature of the positive and negative point-to-plane coronas and the mechanism of spark propagation. *Journal of Applied Physics*, 10(3):142–160, 1939.

[33] Alejandro Luque, Valeria Ratushnaya, and Ute Ebert. Positive and negative streamers in ambient air: modelling evolution and velocities. *Journal of Physics D: Applied Physics*, 41(23):234005, 2008.

[34] TMP Briels, J Kos, GJJ Winands, EM Van Veldhuizen, and Ute Ebert. Positive and negative streamers in ambient air: measuring diameter, velocity and dissipated energy. *Journal of Physics D: Applied Physics*, 41(23):234004, 2008.

[35] David Z Pai, Deanna A Lacoste, and Christophe O Laux. Transitions between corona, glow, and spark regimes of nanosecond repetitively pulsed discharges in air at atmospheric pressure. *Journal of Applied Physics*, 107(9):093303, 2010.

[36] Nicolas Minesi, Sergey A Stepanyan, Pierre B Mariotto, Gabi-Daniel Stancu, and Christophe O Laux. On the arc transition mechanism in nanosecond air discharges. In *AIAA Scitech 2019 Forum*, page 0463, 2019.

[37] Eduard Meerovič Bazelyan and Yu P Raizer. *Spark discharge*. Routledge, 2017.

[38] Ryo Ono, Masaharu Nifuku, Shuzo Fujiwara, Sadashige Horiguchi, and Tetsuji Oda. Gas temperature of capacitance spark discharge in air. *Journal of applied physics*, 97(12):123307, 2005.

[39] Jai Hyuk Choi, Tae Il Lee, Inho Han, Hong Koo Baik, Kie Moon Song, Yong Sik Lim, and Eung Suok Lee. Investigation of the transition between glow and streamer discharges in atmospheric air. *Plasma Sources Science and Technology*, 15(3):416, 2006.

[40] Lvzon. Molecular energy levels, 9 2009.

[41] Moxfyre. Raman energy levels, 9 2009.

[42] Peter Larkin. *Infrared and Raman spectroscopy: principles and spectral interpretation*. Elsevier, 2017.

[43] David J Griffiths. Introduction to electrodynamics, 2005.

[44] V.K Jain. *Laser Systems and Applications*. Alpha Science Internation Limited, 2013.

[45] Detlev Ristau. *Laser-induced damage in optical materials*. CRC Press, 2014.

[46] Arlee V Smith and Binh T Do. Bulk and surface laser damage of silica by picosecond and nanosecond pulses at 1064 nm. *Applied optics*, 47(26):4812–4832, 2008.

[47] C Grey Morgan. Laser-induced breakdown of gases. *Reports on Progress in Physics*, 38(5):621, 1975.

[48] E Schwarz, S Gross, B Fischer, I Muri, J Tauer, H Kofler, and E Wintner. Laser-induced optical breakdown applied for laser spark ignition. *Laser and Particle Beams*, 28(1):109, 2010.

[49] Hans-Jochen Foth. Principles of lasers. *Lackner M. Lasers in chemistry: probing matter. Weinheim: Wiley-VCH*, 2008.

[50] Y Shimony, Z Burshtein, and Y Kalisky. Cr/sup 4+: Yag as passive q-switch and brewster plate in a pulsed nd: Yag laser. *IEEE journal of quantum electronics*, 31(10):1738–1741, 1995.

[51] Nicolaie Pavel, Jiro Saikawa, Sunao Kurimura, and Takunori Taira. High average power diode end-pumped composite nd: Yag laser passively q-switched by cr4+: Yag saturable absorber. *Japanese Journal of Applied Physics*, 40(3R):1253, 2001.

[52] Spectra-Physics. Quanta-ray nd-yag laser family, 2010.

[53] Ophir. *1.1.2.5 Medium Power Large Aperture Thermal Sensors - Apertures to 65mm*, 1 2021.

[54] Rigol. *DG1000Z Series Function/Arbitrary Waveform Generator*, 2016.

[55] Megaimpulse Ltd. *NANOSECOND PULSE GENERATOR NPG-18/3500(N) USER MANUAL*, 2014.

[56] Acton Research Corporation. *Operating Instructions Acton Research Corporation SpectraPro-500i*.

[57] Princeton Instruments. *Princeton Instruments PI-MAX3 1024i ICCD Camera Datasheet*.

[58] LeCroy. Lecroy waverunner xi-a series specifications, 2010.

[59] High-energy beam splitters, specifications.

[60] Nd:yag, nd:ylf laser mirrors (1064nm, 532nm, 355nm, 266nm), specifications.

[61] Mario González-Cardel, Pedro Arguijo, and Rufino Díaz-Uribe. Gaussian beam radius measurement with a knife-edge: a polynomial approximation to the inverse error function. *Applied optics*, 52(16):3849–3855, 2013.

[62] Rodney A Brooks and Giovanni Di Chiro. Principles of computer assisted tomography (cat) in radiographic and radioisotopic imaging. *Physics in Medicine & Biology*, 21(5):689, 1976.

[63] Conrad Hessels. Quantitative analysis of non-premixed flames using raman spectroscopy. Master's thesis, Eindhoven University of Technology, 2017.

Appendix A

Knife edge tomography

A cheap method of beam profiling dubbed 'knife edge tomography' (KET) was devised to measure the intensity distribution in the laser. Please be aware that this portion of the report is not finetuned like the rest of the thesis because of the limited relevance to the project. That being said, the method as a concept is interesting to say the least and can definitely prove useful for qualitative beam profiling when specific profiling equipment is not available. As such, this low-quality 'mini-thesis' on KET is presented here nonetheless.

A.1 KET theory

A.1.1 Knife edge measurements

A very commonly used technique for beam profiling is the knife edge method. A schematic drawing of the technique is shown in figure A.1. Here a razor blade or otherwise straight, thin-edged object is mounted on a translation stage and used to block an increasing part of the laser beam. The power throughput of the laser beam is measured as function of the knife position and plotted [61]. Please note that in our case a power meter was used, but for weaker lasers a photodiode can also be used.

As an example, let's consider a Gaussian beam. A plot of the power throughput for this beam is shown in figure A.2a. It can be seen that as the knife moves further and further into the path of the beam, the power throughput goes down. Now regard the highlighted part in figure A.2a. When the knife is moved from 8 to 9 mm, the power decreases by roughly 1.5 W. This means that the power contained within that 1 mm slice of the beam is 1.5 W. Plotting the change in power throughput as function of knife position yields figure A.2b. The curve plotted here indeed closely matches a Gaussian, as expected.

A big disadvantage of the knife edge method is that it is not possible to resolve the full 2D beam

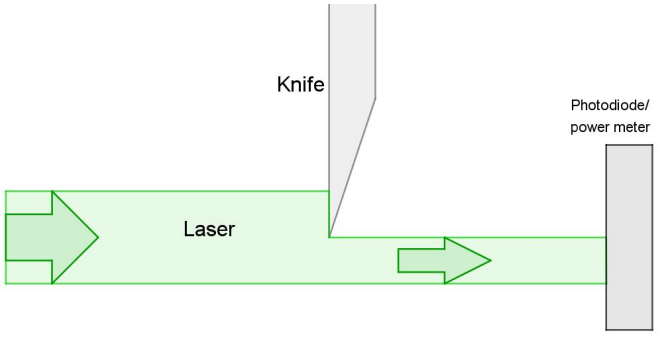
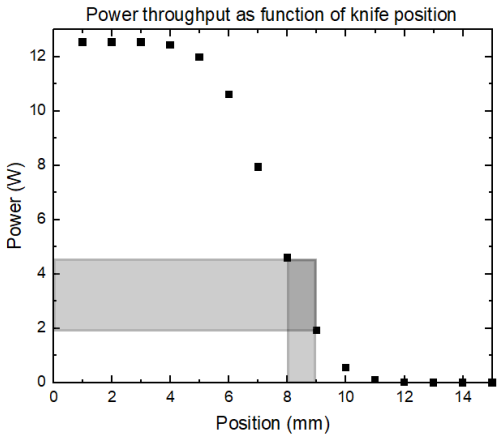
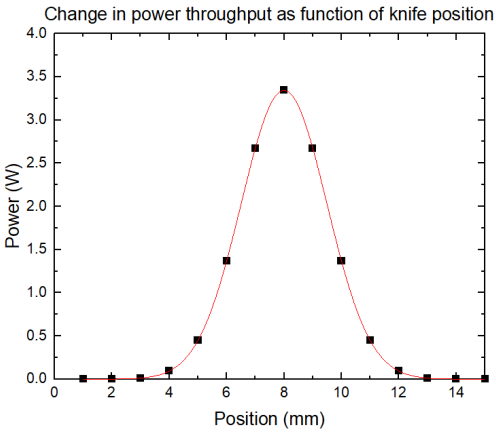


Figure A.1: Schematic illustration of how the knife edge is used to block part of the laser. The power throughput is measured.



(a) Measured power throughput.



(b) Calculated power density fitted against a Gaussian curve ($y = A \cdot exp(-0.5 \cdot (\frac{x-x_0}{w})^2)$, with $A = 3.34$, $x_0 = 8$, $w = 1.50$).

Figure A.2: Two hypothetical data plots as would be obtained using the knife edge method.

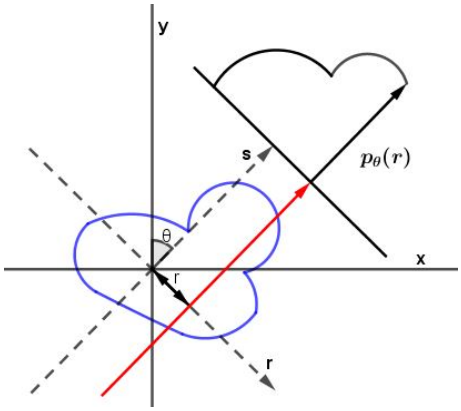


Figure A.3: In a CT scanner, a ray of light (red) is shone through a sample (blue) under an angle θ at a position r. The absorption of this ray $p_\theta(r)$ is recorded.

profile. As a result, it lacks the ability to accurately quantify the intensity distribution, particularly for less smooth or symmetric beam profiles. Therefore it requires some tweaking to be useful for this thesis.

A.1.2 Knife edge tomography

The knife edge method can not measure points of intensity, but only slices of the intensity. By measuring over different angles, more and more spatial information can be resolved. Reconstructing an image by measuring slices over a large number of angles is known as tomography. Tomography is most notably used in medical CT scanners, where the difference in absorption coefficients of different tissues is used to reconstruct internal structures inside the human body. The principle is shown in figure A.3. Rays of light are shone through a sample under a given angle θ. The intensity of light absorbed here is the equivalent of the change in power throughput in the knife edge method. By measuring the absorption under different angles and given the absorption coefficients of different tissues doctors can locate these tissues, most notably tumours [62].

Given an intensity profile or tissue sample, the making of the series of projection images across various angles (the measurement) is mathematically called the radon transform. In nigh on all cases the projection images are available from the measurement and we're interested in the intensity profile, such that the inverse radon transform must be used. However, the radon transform is useful to see the effect of the dataset on the final reconstruction. Figure A.4 illustrates this. The left image is the original image. It is a 2D sinc function and may for instance represent an intensity

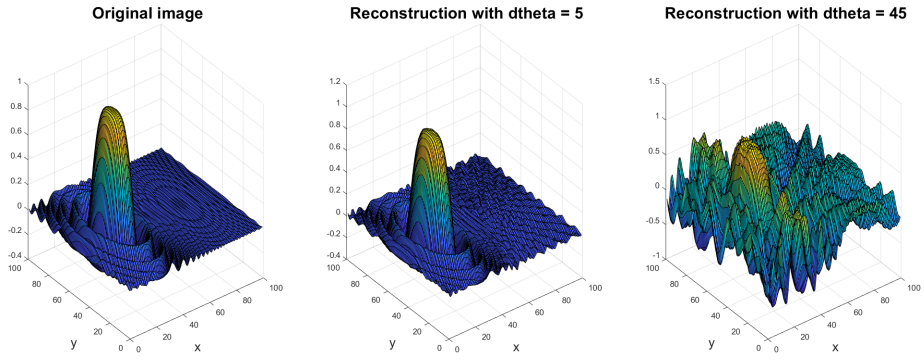


Figure A.4: Tomographic reconstruction of a 2D sinc function (left) using angle step sizes of 5 (middle) and 45 (right) degrees.

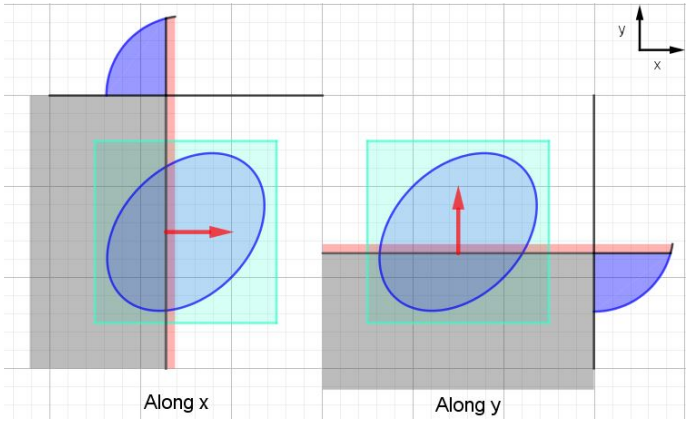


Figure A.5: A knife edge measurement in 2 different directions. The measurement domain is the cyan rectangle, the dark blue ellipse is the intensity profile to be measured. In a measurement along the x direction, the knife (grey) is moved along the x direction. However, that means that each data point is the integration over a line in the y direction (red line).

profile. This original image was radon transformed and again reconstructed twice, once for 36 angles with 5 degrees between them, and once for 4 angles with 45 degrees between them[1]. For both the radon transform and the inverse radon transform, the pre-defined `radon()` and `iradon()` functions from the Matlab image processing toolbox are used.

As expected, the reconstruction with angle step size 5 resembles the original image more closely. What is interesting is how much the right image deviates from the original. The lack of resolution in the data introduces some sort of ridges that seem to emanate from the peak of the sinc function at precisely the angles used for the radon transform. These ridges are quite worrisome, as some of them are on the order of the peak of the sinc and much larger than the second and third oscillations of the function. This hides these features completely.

A.1.3 Sinograms and measurements

Before going into how the data for measurements is obtained and stored, let us define the coordinate system. A measurement 'along the y direction' means that the knife was moved in the y direction. However, as the knife is moved perpendicularly to the blade, that means that every data point on this curve corresponds to the sum of all intensities along a line in the x direction. This is also shown in figure A.5. For future reference, the measurement along the x direction (where thus the knife lies along the y direction) is considered the 0 degrees measurement.

The full dataset obtained using the radon transform is called a 'sinogram'. As an example, figure

[1]The observant reader may have noticed that both these radon transforms only cover 180 degrees. No more angles are needed because a measurement directly opposite another will yield the same data except mirrored.

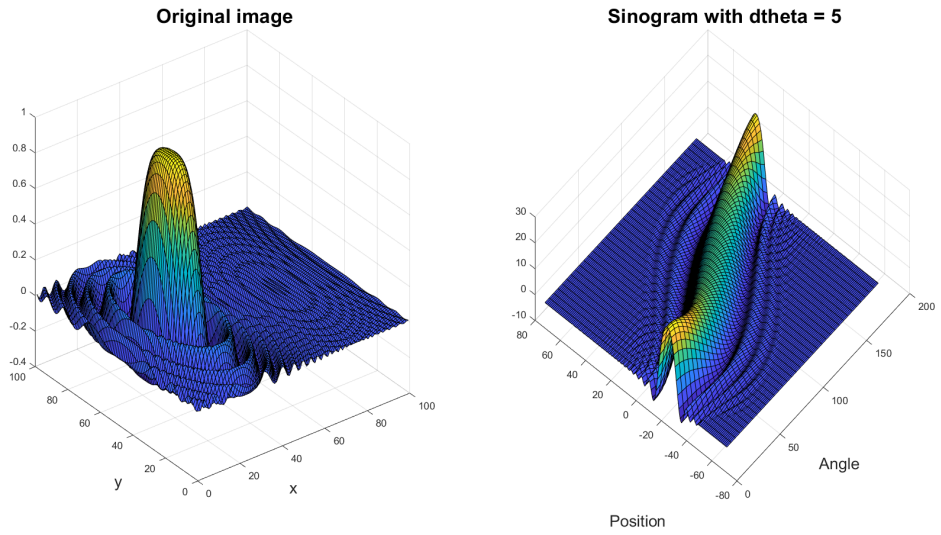


Figure A.6: 2D sinc function with associated sinogram for an angle step size of 5 degrees. The original image is repeated for clarity.

A.6 shows the sinogram accompanying the $d\theta = 5$ reconstruction of the 2D sinc function that was shown in figure A.4.

This figure shows for each angle the power difference at each step along the path of the knife. For instance, at $\theta = 90$, the measurement is along the y direction. That means the knife will cover the peak about halfway through its path, and thus the sinogram also shows a peak in the center of the $\theta = 90$ line. However, as the peak of the sinc function lies off-center in the x-direction, the knife will cover the peak sooner in the $\theta = 0$ direction, and thus the peak at $\theta = 0$ in the sinogram also lies off-center.

When doing tomographic measurements, the data also forms a sinogram, but the real life case is a lot less trivial than simply performing a radon transform.

For starters, when doing tomographic measurements it is important that the blade step size or distance between light rays is constant. That means that more measurements are needed along the diagonal than along the x and y directions. Specifically, for a c-by-c domain (with c either in pixels or a physical length), it can be derived using simply trigonometry that the length of the sinogram d as function of angle is

$$d(\theta) = c \cdot \left(1 - \frac{1}{\sqrt{2}}\right) \cdot \sin(2\theta) + \frac{c}{\sqrt{2}} \tag{A.1}$$

With θ in degrees. For $\theta = 45$ we of course find $d = \sqrt{2} \cdot c$. When reconstructing an image, the values outside this range must be disregarded.

The requirement for constant step size also influences the way a measurement setup must be made or the way the measurements must be made. One option is to make it possible to change the angle of the translation stage the knife is mounted to. However, this is expected to come at the expense of the stability of the setup, which introduces noise into the measurements. If the translation stage is placed either horizontally or vertically, the angle of the blade with respect to the movement direction must be taken into account as well. This is illustrated in figure A.7. Here, the blade is inclined at an angle of 45 degrees w.r.t. the direction of motion. If the translation stage moves by dx, the movement in the direction of the blade is only $dx \cdot \cos(45^o)$. Thus, the step size on the translation stage must be increased by a factor $1/\cos(45^o) = 1.41$ to compensate.

Another important difference between matlab and real measurements is that matlab 'knows' the location of the measurement area very precisely and thus knows where to start each stroke for the sinogram very precisely. Much like the cyan square in figure A.5, there is a well-defined domain to start from to produce a sinogram. In practice, the boundaries of the measurement domain are much less well-defined. A specific domain is difficult to pin down, and the edges of the laser can

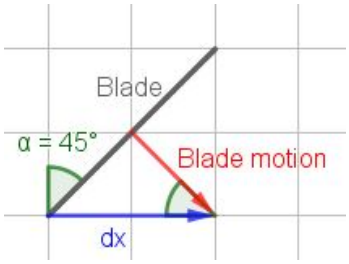
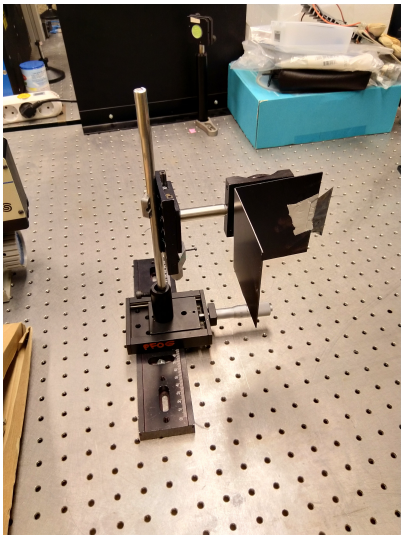
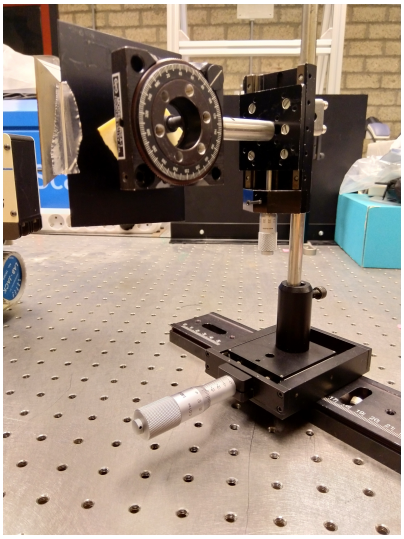


Figure A.7: Schematic representation of how the inclination of the blade changes the movement in the direction of the blade. Shown are the blade (grey), the motion of the translation stage (blue), the motion in the direction of the blade (red) and the inclination angle (green).



(a) Front side



(b) Backside

Figure A.8: The setup to mount and move the knife for knife edge tomography.

not be used accurately, as the laser has a gaussian beam profile rather than sharp edges. Thus, for each individual stroke a guess must be made as to where it fits in the sinogram.

The last point of note is noise and measurement error. There are a few sources of noise in the system. The most notable one is the laser itself, which by sheer virtue of being a Nd:YAG laser has fluctuations in power. In our case, the short term energy stability was quoted as $\pm 3\%$ [52]. Furthermore, an unstable setup can result in the blade changing position slightly. Naturally, errors can influence the data and make conclusions impossible or nonsensical. The smaller the SNR, the larger this problem. Now remember that in the case of knife edge tomography the sinogram represents the amount of power contained within a sliver of the laser beam. Obviously, this power will go down if this sliver is made thinner, i.e. if the knife step size is reduced. The noise and error, on the other hand, don't go down when dx is reduced. As a result, the relative measurement error goes up when the step size goes down. This limits the minimum step size that can practically be used, and thus also limits the minimum feature size that can be resolved.

A.2 KET setup

The knife edge tomography setup consisted of 2 main components: The knife mount and the power meter. The knife mount is shown in figure A.8. The setup consists of a horizontal translation stage with a pole placed on it. Another translation stage is mounted vertically to this pole. A half-wave plate (HWP) is placed on this vertical translation stage, although any object with angle indicators (such as a polariser) could have been used. To the HWP a black metal beam blocker is attached, and the knife is taped to this beam blocker such that the knife edge is horizontal and facing down when the HWP is at 0 degrees.

A.2.1 Method

Once a location for a tomographic measurement was chosen, the power meter was placed in the beam path approximately 20 cm behind it. The knife mount was positioned such that the knife would intersect the beam at that location. It should be noted that the mount had to be moved between measurements for each angle, as rotating the HWP changed the distance from the vertical pole to the knife edge. This is bound to have increased the uncertainty in the measurements.

For the measurements, preferentially the lower translation stage was used, as this was longer than the vertical stage and was thus in most cases sufficient to cover the entirety of the beam. When the lower stage was found to be too short, the upper stage would be moved as well. However, the only situation where this method was not possible was for the 0 degree measurement. This measurement could only be done with the vertical stage, which was too short to cover the whole beam. To make it possible to move the entire translation stage during a measurement, the translation stage was supported by various unused optical components present in the lab of which the dimensions were measured accurately. By adding or removing an extra component, the translation stage could be moved by a fixed amount with less influence of measurement inaccuracy.

The power throughput as function of the knife position for each angle was saved in a .csv file which could be processed by matlab.

Two sets of measurements were performed. The first set were the accuracy measurements. In these, the number of angles required in a measurement to get a sufficient degree of accuracy was determined. A tomographic measurement was done at 1 m distance from the beam output, using 24 different angles. The angles were chosen such that the analysis could be conducted with $d\theta = 10$ and $d\theta = 15$ degrees, or multiples of those, i.e. $\theta = 0, 10, 15, 20, 30, ..., 170$. Tomographic reconstructions were made with various subsets of these angles and compared to decide what the optimal angular resolution would be for the second set of measurements.

The second set of measurements, the profiling measurements, served to quantitatively look at the evolution of the beam profile as a function of propagation distance of the beam. Tomographic measurements were done at 1, 7 and 14 m from the beam output, using the number of angles determined to be sufficient from the accuracy measurements, and from these measurements the beam profile was reconstructed.

The final version of the matlab script for tomography processing is shown in appendix B.1.

A.3 KET measurements

A.3.1 Accuracy measurements

The reconstructions of the accuracy measurements are shown in figure A.9. Figure A.10 shows an image of the beam profile made with laser profiling paper.

It should be noted that the method of producing the sinogram in the script (matching the power curves to overlap as much as possible) essentially forces the beam profile into a circular shape. It helps to produce any beam profile at all, but it may change the found beam profile quite significantly. This will be elaborated on later.

The agreement between the flash image and the top row of A.9 is quite strong, with a bright spot in the lower right corner and an overall sort of horseshoe-shape. The pictures for $d\theta = 45, 60$ show significantly less resemblance. Instead, these have more of a v-shape and show significantly more streaks through the profile. The conclusion seems obvious: Make the tomographic images by measuring over 24 angles. However, the total measurement took a full day. That is, although inconvenient, in and of itself not such a big problem, but if the intensity profile is to be measured along the beam path, several such measurements have to be made. These would thus have to be done over several days, meaning that the laser would have to restart between each measurement, with possible fluctuations in the beam intensity and profile as a result. Consequentially, any comparison made between the profiles at different positions would be increasingly unreliable. Furthermore, the larger the beam profile, the longer the measurements take, and thus measurements further from the laser output might take too long to even complete in one day.

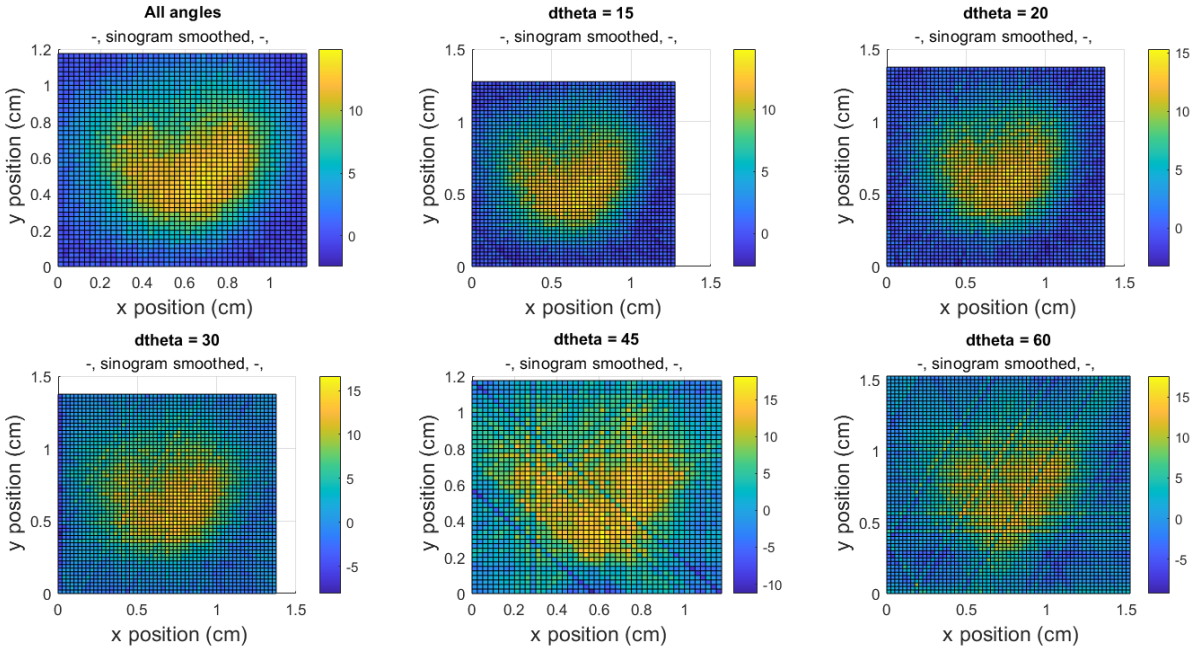
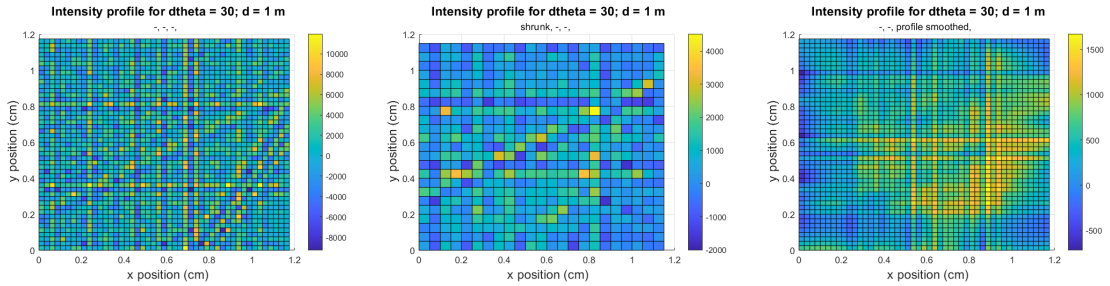


Figure A.9: Reconstructions of the beam intensity profile for different angle step sizes. The number of angles used for each picture is from left to right: 24, 12, 9, 6, 4, 3.

Figure A.10: Flash image of the laser profile to compare to the accuracy measurements.



(a) No processing (b) Changing resolution (c) Profile smoothing

Figure A.11: Tomographic reconstructions for the beam profile at 1 m from the laser using various processing options.

Thus, an optimum has to be chosen that takes the time for a measurement into account. The $d\theta = 30$ measurement would take about an hour. Taking into account that the knife edge has to be set up and the beam diverges, each measurement takes about 1.5 to 2 hours, and so 3 or 4 measurements can be made in one day. This measurement also still roughly shows the horseshoe shape, (although it is a bit more noisy) and is a lot less streaky than the 45 and 60 degree measurements. Therefore $d\theta = 30$ was chosen as the optimal step size for the profiling measurements.

A.3.2 Profiling measurements

Tomographic measurements were performed at 3 different locations, 1, 7 and 14 m from the laser output, with the telescope present in the beam path. The beam profile was determined qualitatively using beam profiling paper. Images of this are shown in figure A.12 a, b and c

After conducting the measurements at the given 3 positions with $d\theta = 30$, a problem arose. In the script, there were 3 processing options, and it was unknown which would yield the most accurate result. The options were to shrink the data (halve the measurement resolution), smoothen the sinogram (apply a savitsky-golay filter to the raw data), or smoothen the profile (apply a savitsky-golay filter once in each direction over the intensity profile). These could be applied in any combination, but the results varied greatly. First off, the result of not processing, or applying only shrinking or profile smoothing to the 1 m measurements is shown in figure A.11.

These figures should look like figure A.12a. Figures A.11a and A.11b are completely indistinguishable from random noise. Figure A.11c does show some resemblance to figure A.12a, but it shows a lot of streaks. It should be noted that these streaks are not a result of the inverse radon transform, but rather of the savitsky-golay filter. As this filter can only work in one direction, the filtering happens in lines rather than surfaces. As a result, these lines appear in the final profile. The pictures made further from the beam output were equally useless with these processing options.

Figure A.12 shows tomographic reconstructions of the data at 1, 7 and 14 m for the remaining processing options. The left column contains images of the 1 m data, the middle column 7 m, and the right column 14 m. The top row contains a picture made with beam profiling paper for comparison to the reconstructions. The colorbars next to the reconstructions are in units of $\mathrm{mJ/cm^2/pulse}$.

It is important to point out that it is no mistake that the reconstructions have different sizes for different processing options. Due to smoothing and shrinking, some non-zero values at the edge of the sinogram may appear or disappear. As a result, MATLAB may perceive the edge of the sinogram to change position slightly. If the sinogram becomes larger, so too will the domain become larger.

The first notable observation is in the pictures d, e and f. These use only sinogram smoothing. The pictures made at 7 and 14 m look like pure noise, whereas the leftmost picture does show a similar general shape to the beam profiling picture. This seems to suggest that the 1m pictures are somehow easier to reconstruct. This would make sense, as the profile is quite a bit larger at 1m, and thus presumably contains less sharp peaks, making it less susceptible to noise. Figure A.12d

is still quite noisy though, but we see that that is resolved quite well when shrinking the data as well, as shown in figure A.12g. Figures h and i to the right of it show that with these settings it is even possible to resolve the images further from the output.

Going further down in the columns, the images become increasingly less noisy. Most notable is figure A.12m, which shows a strong resemblance to the flash profile including a lot of detail. However, the 7 and 14 m reconstructions with the same options show a lot more streaks inherent to the smoothing, and appear as just a few spots of intensity. If we purely look at how well the 7 m reconstructions match the flash image, figure h would appear to be best as figures k and q seem to have lost all detail. For 14 m on the other hand, l and r match the best as these are the only two that show the single bright spot on the left.

This leaves us with a problem. For all 3 distances, different processing options yield the best matching reconstructions. Thus, there is no conclusively best set of processing options. All of this is exacerbated by a fact mentioned previously: due to the way the sinogram is created, the overall beam shape might be warped with respect to its original shape. As a result, it is difficult to say to what extent even the best reconstructions actually match the beam profile.

Particularly for the 14 m measurement (which is actually the most interesting) the maximum intensity varies significantly depending on the processing options, by up to a factor 2. As the maximum intensity is precisely the quantity that interests us, this means that this method is not really suitable for our purposes.

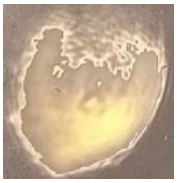
A.4 Conclusion on beam profiling

To conclude, although KET seemed a good and safe method to determine whether the beam splitter could be illuminated directly with the laser, it proved too unreliable for our purposes. Thus, it was decided to simply try the laser on the beam splitter.

The laser was aimed at the edge of the beam splitter (so as to avoid rendering the BS completely useless if the laser proved too strong), and the power was gradually increased. Fortunately, even at maximum pulse energy (750 mJ), no effect of the laser on the BS was observed.

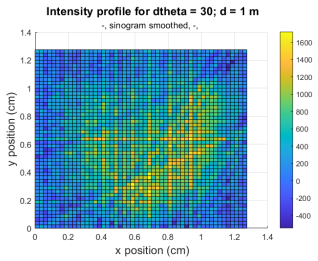
This means that the telescope can be safely removed from the laser output, giving the advantage of less beam profile distortion due to spherical aberrations and smaller truncation losses.

(a) 1 m

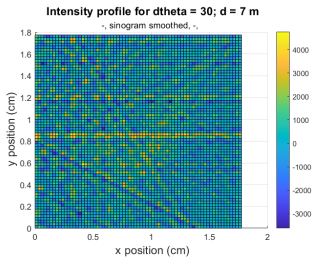


(b) 7 m

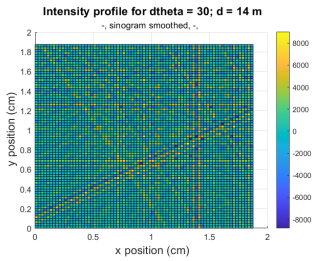
(c) 14 m



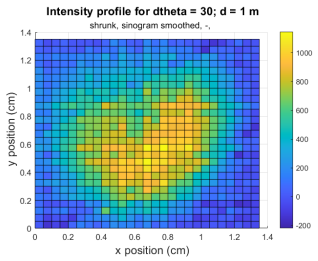
(d)



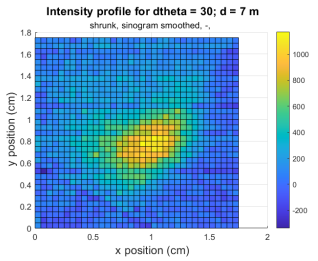
(e)



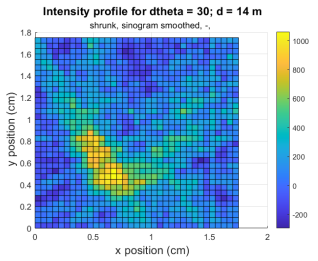
(f)



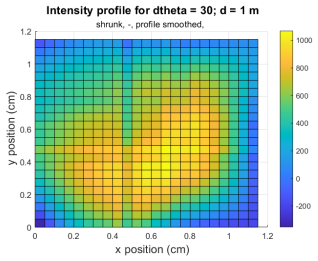
(g)



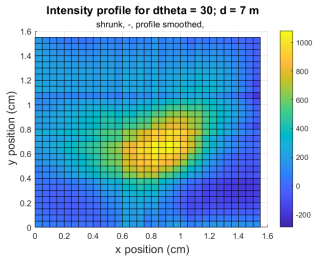
(h)



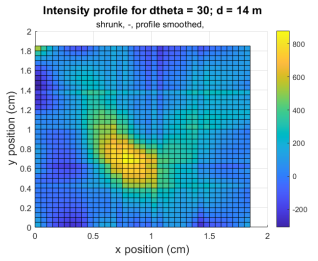
(i)



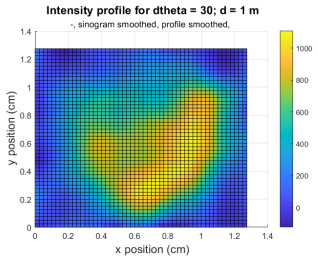
(j)



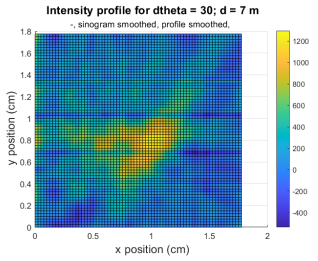
(k)



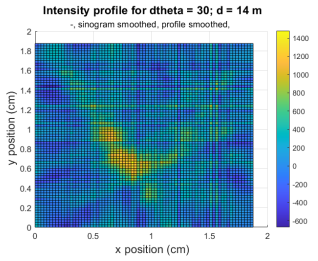
(l)



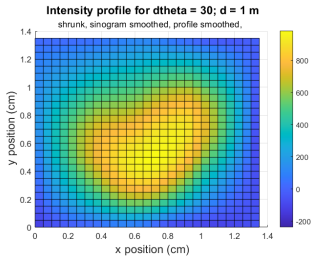
(m)



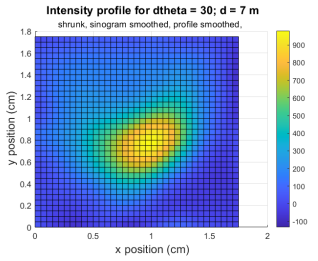
(n)



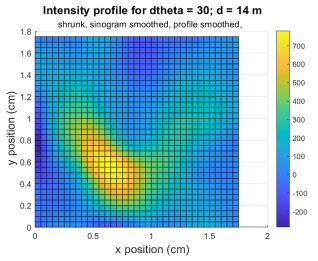
(o)



(p)



(q)



(r)

Figure A.12: a-c: Laser profiling paper images made at 1, 7 and 14 m from the laser output. d-r: Tomographic images of the beam intensity profile using various processing options. Intensity is in mJ/cm^2/pulse. The telescope was in the beam path at 20 cm from the laser output.

Appendix B

Matlab scripts

The following sections will display the most important matlab scripts used for this thesis, particularly for data processing. These serve only as a rough outline, the scripts need some finetuning and additional commenting to be useful by people other than me. If you would like to use the scripts, do contact me so that I can send them to you once the finetuning is done.

B.1 Tomography data processing script

This script processes data from knife edge tomography measurements and attempts to reconstruct the original image.

First the script will ask for a file to process. This file should be a .csv file with the power difference data in the rows, and the first column containing the angle corresponding to each row. It will then fix some preset options in the script.

Then the processing starts. The script first loads the specified file and determines which angles from that file are to be used and what data from the file accompanies each of these angles. Depending on what initial options were chosen, the script may then alter some of the data (change the resolution or smoothen it).

The selected dataset is then turned into a sinogram. As previously mentioned, the start and end points of each stroke are not very well defined. If this is not taken into account, every stroke might be shifted significantly with respect to the previous one. This causes the sinogram to become very jagged as opposed to a smooth ridge that would be expected, like shown in figure A.6. To overcome this, the script shifts each stroke such that it overlaps as much as possible with the previous stroke, to make the sinogram as smooth as possible.

Lastly, either the intensity profile or the sinogram will be plotted.

```
%--------------------------------------------------------------------------
%% Determine input file
file_specified = exist('filename', 'var'); % Find if a filename has already been input, to
    ↪ avoid typing it 100 times
% If you want a new filename, just set 'filename = [your file name]' in the
% command window or clear the variable filename and specify it during the script

if file_specified == 0
    filename = input('Specify filename: ','s');
end

%--------------------------------------------------------------------------
%% Options

lower_res = 0; % Change the data resolution
sin_smooth = 1; % Smooth the sinogram data (before the inverse radon transform)
```

```
change_data_orientation = 0; % Rotate the dataset, this allows you to correct for
    ↪ processing errors that rotate the data
start_threshold = 0.15; % Threshold for data edge (sinogram)

% c is a parameter that truncates the sinogram to the shape it should have
% With the following you can set the up and down values of c to some value
% 1 = set to largest, 2 = set to smallest, 3 = set to average, 0 = do nothing
equalise_c = 1;

normalise = 0; % Normalise data such that the total power (not maximum intensity) is 1
prof_smooth = 1; % Smooth the profile (after the inverse radon transform)

% Plot the intensity profile or the sinogram
% 1 = plot the profile, 2 = plot the sinogram, anything else is do nothing
plot_profile = 1;
measure_res = 0.025; % Set the step size (in m) that was used for the tomography

%------------------------------------------------------------------------
%% Determine the dataset to be used, find the angles and the data accompanying it

dataset = load(filename);
% Choose which part of the data you want to use. You can specify a subset,
% use all the data in a set, or process only the angles for which data is present
selectbool = input('Process all data or a chosen subset? Only nonzero(2), All(1), Sub(0):
    ↪ ');

if selectbool == 0 % Chosen subset
    angle_set = dataset(:,1); % All the available angles
    % You can set here 2 options: Use equidistant angles from 0-180, or
    dtheta = input('Set dtheta: ');
    used_angles = 0:dtheta:180-dtheta;
    % Use a self-chosen specified set
% used_angles = input('Vector of chosen angles: ');
    angle_index = zeros(length(used_angles),1);
    for i = 1:length(used_angles) % Find each chosen angle and save its index
        current_angle = used_angles(i);
        angle_index(i) = find(angle_set == current_angle);
    end
    used_intensities = dataset(angle_index,2:end); % Use the indices to get the intensity
        ↪ curves
    disp('Used angles: ')
    disp(transpose(used_angles));
elseif selectbool == 1 % Use all data
    used_angles = dataset(:,1); % Get angles
    used_intensities = dataset(:,2:end); % Get intensity curves
    disp('Used angles: ')
    disp(transpose(used_angles));
elseif selectbool == 2 % Only non-zero
    base_angles = dataset(:,1);
    used_angles = zeros(length(base_angles),1); % Vector with only the non-zero angles
    useindex = zeros(length(base_angles),1); % Vector with indices of those angles
    used_intensities = dataset(:,2:end);
    j = 1;
    for i = 1:length(base_angles)
        emptydatabool = find(used_intensities(i,:)); % Find any non-zero values
        if emptydatabool ~= 0 % If there aren't no nonzero (if there are only nonzero)
            ↪ values
            used_angles(j) = base_angles(i); % Save this angle
            useindex(j) = i; % And its index
            j = j + 1;
        end
    end
    used_angles = used_angles(1:j-1); % Omit zero angles
```

```
    useindex = useindex(1:j-1);
    disp('Used angles: ')
    disp(transpose(used_angles));
    used_intensities = dataset(useindex,2:end); % Select non-zero intensity curves
else % If none of the options was chosen, do the same as for only non-zero
    disp('Wrong input, processing all nonzero data...')
    selectbool = 2;
    base_angles = dataset(:,1);
    used_angles = zeros(length(base_angles),1);
    useindex = zeros(length(base_angles),1);
    used_intensities = dataset(:,2:end);
    j = 1;
    for i = 1:length(base_angles)
        emptydatabool = find(used_intensities(i,:));
        if emptydatabool ~= 0
            used_angles(j) = base_angles(i);
            useindex(j) = i;
            j = j + 1;
        end
    end
    used_angles = used_angles(1:j-1);
    useindex = useindex(1:j-1);
    disp('Used angles: ')
    disp(transpose(used_angles));
    used_intensities = dataset(useindex,2:end);
end

%--------------------------------------------------------------------------
%% Processing options 1
% Here some of the earlier chosen options are applied

% Change the data resolution
shrunk = '-, ';
if lower_res == 1
    shrink_data % This is a separate script that merges the data in the sinogram to lower
        ↪ resolution
    shrunk = 'shrunk, ';
end

% Smooth the sinogram data (before the inverse radon transform)
sin_sm_prompt = '-, ';
if sin_smooth == 1
    used_intensities = smoothdata(used_intensities,2,'sgolay');
    sin_sm_prompt = 'sinogram smoothed, ';
end

% Rotate the dataset, this allows you to correct for processing errors that
% rotate the data. You will have to set this correction by hand.
if change_data_orientation == 1
    used_angles = flip(used_angles);
    used_angles = used_angles -60;
end

[N_theta,N_data] = size(used_intensities);

%--------------------------------------------------------------------------
%% Turn selected data into sinogram

% Each knife stroke is positioned such in the sinogram that it overlaps the
% most with the previous stroke, to get as smooth a curve as possible

best_fits = zeros(N_theta-1,1); % Stores how much each stroke must be shifted
% The first stroke will be centred in an array that is 3 times as long.
```

```
data_0 = zeros(1,N_data*3);
data_0(1,N_data+1:2*N_data) = used_intensities(1,:);

% The second stroke will be shifted in that same array from beginning to
% end, where at each position it is compared to the previous stroke.
% The position where the difference from the first stroke is the smallest,
% will be set as the final position.
% This will be repeated for all strokes
for i = 1:N_theta - 1
    % Create 3 times as long array
    res_array = zeros(N_data * 2 , N_data * 3);
    for j = 1:2*N_data % Test 2*N_data shifts of the data w.r.t. the middle
        res_array(j,j:N_data+j-1) = used_intensities(i+1,:);
        res_array(j,:) = res_array(j,:) - data_0; % Determine difference of each shift to
            ↪ previous stroke
    end
    residuals = sum(res_array.^2,2); % Calculate sum of the square of the differences
    best_fits(i) = find(residuals == min(residuals),1); % And find the minimum from that
    data_0 = zeros(1,N_data*3); % Reset the array
    data_0(1,best_fits(i):N_data - 1 +best_fits(i)) = used_intensities(i+1,:); % Set the
        ↪ newly found stroke as the previous stroke for the next measurement
end

% Fill the sinogram
sino_initial = zeros(N_theta,N_data*3);
sino_initial(1,N_data+1:2*N_data) = used_intensities(1,:);
for i = 2:N_theta
    sino_initial(i,best_fits(i-1):N_data-1+best_fits(i-1))...
        = used_intensities(i,:);
end

% Find center of the sinogram (This might be possible in a more reliable
% way, but for now just N_data*1.5 is used)
midpoint = round(N_data*1.5);
low_side = 0;
high_side = 0;

% The parameters c1 and c2 indicate the position of the edges of the sinogram
% Since both c's are as large as the midpoint, c is taken as large as
% possible initially
c1 = midpoint;
c2 = 3*N_data - midpoint;
amp_factor = 1 - 1/sqrt(2);
borderdown = zeros(1,N_theta);
downfound = zeros(1,N_theta);
borderup = zeros(1,N_theta);
upfound = zeros(1,N_theta);

% Here we calculate how far the sinogram can stretch. At every iteration,
% the number c is decreased by 1, and then matlab checks if every angle
% actually contains only data points.
while low_side == 0
    c1 = c1 - 1; % Decrease c
    % Fit the shape of the borders to indices in the sinogram
    borderdown = round(abs(sin(2*used_angles*pi/180)) .* c1 .* amp_factor + c1/sqrt(2));
    for i = 1:N_theta % For every angle check if the border value is zero
        if sino_initial(i,midpoint-borderdown(i)) ~= 0
            downfound(i) = 1;
        end
    end
    border_zeros = find(downfound == 0);
    if isempty(border_zeros) % If no angle is zero any more stop iteration
        low_side = 1;
```

```
    end
end

% Same thing here
while high_side == 0
    c2 = c2 - 1;
    borderup = round(abs(sin(2*used_angles*pi/180)) .* c2 .* amp_factor + c2/sqrt(2));
    for i = 1:N_theta
        if sino_initial(i,midpoint+borderup(i)) ~= 0
            upfound(i) = 1;
        end
    end
    border_zeros = find(upfound == 0);
    if isempty(border_zeros)
        high_side = 1;
    end
end

%--------------------------------------------------------------------------
%% Choose sinogram borders
% with this you can set the up and down values of c to some value
% 1 = set to largest, 2 = set to smallest, 3 = set to average, 0 = do nothing
% c1 and c2 must be equal to make the sinogram symmetrical
if equalise_c == 1 % Set to largest
    if c1 > c2
        c2 = c1;
        borderup = round(abs(sin(2*used_angles*pi/180)) .* c2 .* amp_factor + c2/sqrt(2));
    else
        c1 = c2;
        borderdown = round(abs(sin(2*used_angles*pi/180)) .* c1 .* amp_factor + c1/sqrt(2))
            ↪ ;
    end
elseif equalise_c == 2
    if c1 > c2
        c1 = c2;
        borderdown = round(abs(sin(2*used_angles*pi/180)) .* c1 .* amp_factor + c1/sqrt(2))
            ↪ ;
    else
        c2 = c1;
        borderup = round(abs(sin(2*used_angles*pi/180)) .* c2 .* amp_factor + c2/sqrt(2));
    end
elseif equalise_c == 3
    c1 = round((c1+c2)/2);
    c2 = c1;
    borderdown = round(abs(sin(2*used_angles*pi/180)) .* c1 .* amp_factor + c1/sqrt(2));
    borderup = round(abs(sin(2*used_angles*pi/180)) .* c2 .* amp_factor + c2/sqrt(2));
end

% Truncate the sinogram to its smallest possible size
sinogram = zeros(N_theta,c2+c1);
for i = 1:N_theta % Plug in the data for each angle
    sinogram(i,c1-borderdown(i)+1:c1+borderup(i)) = sino_initial(i,midpoint-borderdown(i)
        ↪ +1:midpoint+borderup(i));
end

sinogram = transpose(sinogram);

%--------------------------------------------------------------------------
%% Plot profile

% Transform the sinogram into the intensity profile
profile = iradon(sinogram,used_angles);
```

```
% Normalise data such that the total power (not maximum intensity) is 1
if normalise == 1
    P_tot = sum(sum(profile));
    profile = profile/P_tot;
end

% Smooth the profile (after the inverse radon transform)
prof_sm_prompt = '-, ';
if prof_smooth == 1
    profile = smoothdata(smoothdata(profile,1,'sgolay',15),2,'sgolay',15);
    prof_sm_prompt = 'profile smoothed, ';
end

% Plot the intensity profile or the sinogram
% 1 = plot the profile, 2 = plot the sinogram, anything else is do nothing
if plot_profile == 1
    [size_x,size_y] = size(profile);
    % Set the axes
    x_ax = 0 : measure_res : (size_x - 1) * measure_res;
    y_ax = 0 : measure_res : (size_y - 1) * measure_res;
    if normalise ~= 1 % If you want the actual intensities
        profile = profile/measure_res^2; % Scale intensity to W/cm^2
    end
    surf(x_ax,y_ax,profile);
    % Change title depending on chosen dataset
    if selectbool == 1
        title('\fontsize{16} Intensity profile for all angles')
    elseif selectbool == 2
        title('\fontsize{16} Intensity profile for non-zero angles')
    else
        title(['\fontsize{14} Intensity profile for dtheta = ',num2str(dtheta)])
    end
    xlabel('\fontsize{14} x position (cm)')
    ylabel('\fontsize{14} y position (cm)')
    zlabel('\fontsize{14} Intensity (W/cm^2)')
    colorbar
    view(-45,50)
elseif plot_profile == 2
    x = (0:c1+c2-1)*measure_res; % Knife position values
    surf(used_angles,x,sinogram/measure_res)
    % Change title depending on chosen dataset
    if selectbool == 1
        title('\fontsize{14} Sinogram for all angles')
    elseif selectbool == 2
        title('\fontsize{14} Sinogram for non-zero angles')
    else
        title(['\fontsize{14} Sinogram for dtheta = ',num2str(dtheta)])
    end
    xlabel('\fontsize{14} theta (degrees)')
    ylabel('\fontsize{14} position (cm)')
    zlabel('\fontsize{14} Power difference (W/cm)')
    colorbar
    view(-45,50)
end

% Indicate the processing options in the subtitle of the figure
subtitle([shrunk, sin_sm_prompt, prof_sm_prompt])
```

B.2 Time profile calculation script

This script first loads the time profile of the laser measured with the oscilloscope and high-speed diode. Depending on what the user wants to look at (which cavities need to be closed), the script

will set the `measure_data` variable to one of the loaded measurements. The script will also extract the shape of the original (unstretched) pulse, which can be obtained by closing both cavities [1]. This pulse is used for the theoretical pulse. Rather than by assuming a gaussian profile, the theoretical pulse will be created by stretching and duplicating this base pulse to get better agreement. It is, however, also possible to let the script use Gaussian time profiles as base pulse. The script will then extract some basic values from the measurements to determine the start and the FWHM.

In the next part of the script, some parameters can be specified that are necessary to calculate the theoretical time profile of the pulse. Most notable are the cavity length and the transmission and reflectivity of the mirrors and beam splitters.

The script then calculates the position and power of each pulse separately. A distinction must be made between pulses that are immediately reflected at the front of the beam splitter, and pulses that pass through a cavity at least once. A pulse that is immediately reflected does not encounter the transmission of the beam splitter, only the reflectivity. This pulse is simply reduced in height by a factor R_{bs}, the reflectivity of the beam splitter. Pulses that are transmitted encounter this transmission twice: once when entering the cavity, and once when exiting, thus these are reduced by a factor T_{bs}^2. Furthermore, they lose additional power for each mirror in the cavity: $R_m^{N_{mir}}$, where R_m is the reflectivity of the mirror and N_{mir} is the number of mirrors in that cavity. Lastly, if a pulse makes not 1 but n round trips through a cavity, it will reflect off the back of the beam splitter $n-1$ times, and see all mirrors $n-1$ more times as well. Thus, a pulse that makes n round trips through the cavity will transmit a fraction of the power of the unstretched pulse of[2]

$$T_{tot} = T_{bs}^2 \cdot R_m^{N_{mir}} \cdot (R_{bs} \cdot R_m^{N_{mir}})^{n-1} \tag{B.1}$$

This way the total peak height of each iteration of the pulse can be determined, and each of them is then shifted by its time delay.

Finally, the total transmission of the system can be calculated by summing the peak heights of all pulses. This will be considered the theoretical maximum achievable transmission. It thus takes into account the reflectivities of the mirrors and beam splitters, the transmission of the beam splitters and the number of mirrors each pulse encounters, but not the truncation losses.

The final pulse is created by summing all individual pulses, accounting for their power reduction and delay. The maximum power and pulse duration are then calculated, and both the theoretical pulse and the pulse for comparison are plotted.

```
% Calculate 1 time profile for given time delays and shit

%% Load data

time_profile_extractor % Script that loads time profile data
% Output data is 4 strings named 'close_1', 'close_2', 'close_both', 'full_cav'
% These strings are the filenames of different combinations of closed cavities

% In principle, the script will not ask for new input every time. If you
% want to select a new set of measurements to process, type 'measure_date = [your file
    ↪ name]'
% in the command window, or clear the variable measure_date and specify it when prompted.

% Process the data for either or both cavities blocked. 1 = block, 0 = open
block1 = 0;
block2 = 0;

% Select the appropriate measurement with the closed cavities
if block1 == 1
    if block2 == 1
        measurement = close_both;
```

[1] A cavity can be closed by placing a beam dump in the laser path inside the cavity.

[2] Please be mindful that in the Matlab script the power is not $n-1$ but $i,j-2$. This is because the recursion values i and j are defined such that $i,j=1$ is the reflected pulse, and $i,j=2$ is the pulse that makes 1 round trip through the cavity. I.e. $i,j=n+1$.

```
    else
        measurement = close_1;
    end
else
    if block2 == 1
        measurement = close_2;
    else
        measurement = full_cav;
    end
end

% Load the data
measure_data = load(measurement); % Not to be confused with measure_date
close_measure = load(close_both); % Find the base (unstretched) pulse shape
pulse_shape = close_measure(:,2);
t = measure_data(:,1)*10^9; % Time in ns

%% Process curve_shape

% Determine pulse start, peak height and fwhm
peak_00 = max(pulse_shape); % Peak height
i0 = find(pulse_shape == peak_00,1);
t0 = close_measure(i0,1); % Peak position in ns
dt = (close_measure(2,1) - close_measure(1,1))*10^9; % Timestep
pulse_start = find(pulse_shape > peak_00/2,1); % Determine pulse duration
pulse_end = find(pulse_shape > peak_00/2,1,'last');
base_fwhm = (close_measure(pulse_end,1) - close_measure(pulse_start,1))*10^9;

% Normalize
pulse_shape = pulse_shape/peak_00;


%% Parameters

% Pulse and cavity parameters
L1 = 8.02; % First cavity length in m
Nmir1 = 7; % Number of mirrors in the first cavity
L2 = 2.90; % Second cavity length in m
Nmir2 = 3; % Number of mirrors in the second cavity
c = 0.299792458; % Speed of light in m/ns
t1 = L1/c; % Time delay of cavity 1
t2 = L2/c; % Time delay of cavity 2

Nmax1 = 5; % Fixed number of trips through the cavity
Nmax2 = 5;

% Automatically set the subtitle in the plot
cav1open = [num2str(L1), 'm'];
cav2open = [num2str(L2), 'm'];

if block1 == 1
    Nmax1 = 0;
    cav1open = 'closed';
end
if block2 == 1
    Nmax2 = 0;
    cav2open = 'closed';
end


% Reflectivity parameters
Rbs = 0.38; % Beam splitter reflectivity
Tbs = 0.6; % And transmission
```

```
Rm = 0.99; % Mirror reflectivity

% For comparing the two plotted curves, you can scale and shift the data to
% fit them
shift = 0.2;
fac = 2.9;

% Arrays that save the transmissions and shapes, respectively, of each
% iteration of the stretched pulse
Tfacs = zeros(Nmax1+1,Nmax2+1);
pulses = zeros(Nmax2+1,Nmax1+1,length(t));


%% Calculate pulse
for j = 1:Nmax2+1
    for i = 1:Nmax1+1
        T = 1; % Reset R

        T = T * Rm; % Mirror from laser to BS1

        % Cavity 1:
        if i == 1 % Reflected part
            T = T * Rbs;
        else % Transmitted part and i-2 repetitions
            T = T * Tbs^2 * Rm^Nmir1 * (Rbs * Rm^Nmir1)^(i-2);
        end

        T = T * Rm; % Mirror between first and second BS

        % Cavity 2
        if j == 1 % Reflected part
            T = T * Rbs;
        else % Transmitted part and j-2 repetitions
            T = T * Tbs^2 * Rm^Nmir2 * (Rbs * Rm^Nmir2)^(j-2);
        end

        delay = round((shift + t1*(i-1) + t2*(j-1))/dt);

        pulses(i,j,delay:end) = T*pulse_shape(1:end-delay+1);

        % Uncomment the next line to use a gaussian profile for fitting
% pulses(i,j,:) = R*exp(-4*log(2)*(t-shift-(t1*(i-1) + t2*(j-1))).^2/base_fwhm^2);

        % Save transmission
        Tfacs(i,j) = T;
    end
end

% The total transmission is the sum of all transmitted peaks
T_tot = sum(sum(Tfacs));

% The total, final pulse can be found by summing each individual pulse
fin_pulse = sum(sum(pulses,1),2);
fin_pulse = squeeze(squeeze(fin_pulse));
P_max = max(fin_pulse); % Calculate the peak power reduction
% Calculate the fwhm of the final pulse
P_half = P_max/2;
pulse_start = find(fin_pulse > P_half,1);
pulse_end = find(fin_pulse > P_half,1,'last');
pulse_duration = t(pulse_end) - t(pulse_start);


% Determine the range to plot (only the parts where the pulse is above a
```

```
% particular threshold)
pulse_pos = find(fin_pulse > 0.001);
last = max(pulse_pos);
% Start 0.5 fwhm from the beginning of the pulse
first = round(pulse_start - (pulse_end - pulse_start)/2);

% Plot
plot(t(first:last),fin_pulse(first:last));
xlabel('Time (ns)');
ylabel('Peak power fraction');
title('Laser pulse time profile');
subtitle(['Input FWHM = ', num2str(base_fwhm), 'ns ; L_1 = ', cav1open, '; L_2 = ',
    ↪ cav2open]);
hold on
plot(t(first:last),measure_data(first:last,2)*fac);
grid on
hold off

% Output results
disp(['T_tot: ', num2str(T_tot), ', P_max = ', num2str(P_max), ', FWHM = ', num2str(
    ↪ pulse_duration), ' ns'])
```

B.3 Plasma image processing script

This script processes the plasma ICCD images to determine amongst others their width and the laser-streamer hit rate.

Firstly, the excel file that contains the relevant picture indices is loaded. The file is analyzed to determine the parameters investigated during this set of measurements, as well as the possible values these parameters have taken. This allows the script to both determine the parameter combination corresponding to a particular measurement, as well as to find a measurement with a particular combination of parameters.

Next, the script will run through all electrode references in the file. The user must then set the positions of the electrodes in the file and set the borders of the figure to crop the pictures and save data. It will then also go through all indices in the excel file and crop the images accordingly. The cropped images and configuration data will be saved, so a user may choose to load the data from an old file and skip this step.

The script will then prompt the user to indicate which parameter must be set on the x-axis and for which parameter(s) the values will be compared in the final plot. It then determines which parameters are left and prompts the user to either select a value for them or to average over all values for this parameter.

The last step before the plasma properties are calculated is to determine the scale of the picture. Because all pictures are made with the same lens at the same distance, the number of pixels per mm should be constant for all pictures. The physical size of the electrode gap was measured for each configuration as it was one of the parameters to be set. The number of pixels corresponding to the gap length can be determined from the electrode references. The conversion from number of pixels to mm is calculated by dividing the number of pixels by the gap size for each electrode reference, and averaging over all the results.

Based on all the above, the script will be able to go through all the selected indices and calculate the plasma properties for those. The center position of the streamer is found by taking horizontal slices of the ICCD images and finding the maximum in each, the edges of the streamer are considered located at half the maximum value in their slice. As a result, the width becomes the full width at half maximum of the streamer intensity.

The saved plasma properties are the mean and standard deviation of the total intensity, the width, the deviation and the movement of the streamer. The difference between the deviation and movement is that the deviation only takes the absolute distance between the streamer and

the plasma axis into account, whereas the movement is the actual position w.r.t. the axis so that consistent deviations can be detected.

Lastly, the ignition and hit fractions are determined and saved. If the total streamer intensity lies below a particular threshold, the picture is considered a failed ignition. If the center of the plasma axis does not lie between the left and right edges of the streamer, it is considered a miss. Naturally, every failed ignition is also a miss.

The script can then plot the calculated values.

```
% Plot data from one large cell file which contains everything
% First run roi_selector at least once

%% Set parameters
not_first_run = exist('Npres','var');
if not_first_run == 0
    roi_selector
end

% Change whether you will be prompted about the sets of parameters to plot
setonce = 1;
if setonce == 0
    % Input stuff
    compare1 = input('Compare 1 or 2 parameters? ');
    disp('Gap = 1, Flow rate = 2, Pressure = 3, Pulse number = 4, Electrode type = 5. ');
    xparam = input('Set the parameter on the x-axis (do not use 5): ');
    param1 = input('Set first parameter for comparison: ');
    if compare1 == 2
        param2 = input('Set second parameter for comparison: ');
    end
else
    % Set stuff here
    compare1 = 1;
    % Gap = 1, Flow rate = 2, Pressure = 3, Pulse number = 4, Electrode type = 5
    xparam = 1;
    param1 = 4;
    param2 = 2;
end

Nparams = 5; % Number of parameters. MUST CORRESPOND TO ACTUAL NUMBER USED

% Cell array with the names for each parameter
paramsused = cell(Nparams,1);
paramsused{1} = 'Gap size';
paramsused{2} = 'Flow speed';
paramsused{3} = 'Pressure';
paramsused{4} = 'Pulse number';
paramsused{5} = 'Electrode type';
% paramsused{6} = 'Equivalence ratio';

% Cell array with the units for each parameter
unitcell = cell(Nparams,1);
unitcell{1} = 'mm';
unitcell{2} = 'cm/s';
unitcell{3} = 'mbar';
unitcell{4} = '';
unitcell{5} = '';
% unitcell{6} = '';

% Array with the possible values for each array
paramsets = zeros(Nparams,max([Npres,Npulse,Nflow,Ngap,Nelec]));
paramsets(1,1:Ngap) = gaps2;
paramsets(2,1:Nflow) = flows2;
```

```
paramsets(3,1:Npres) = pressures2;
paramsets(4,1:Npulse) = pulse_nos2;
paramsets(5,1:Nelec) = 1:Nelec; % Done different as it contains strings

% Set the parameter chosen as x value as x
xlist = paramsets(xparam,:);
xlist(xlist == 0) = [];
Nx = length(xlist);
% Same for first parameter for comparison
param1list = paramsets(param1,:);
param1list(param1list == 0) = [];
N1 = length(param1list);

% Mark parameters not an x or comparison
remainder1 = intersect(find(1:Nparams ~= xparam),find(1:Nparams ~= param1));

% If relevant, add second parameter for comparison and update remainder
if compare1 == 2
    param2list = paramsets(param2,:);
    param2list(param2list == 0) = [];
    N2 = length(param2list);
    remainder = intersect(remainder1, find(1:Nparams ~= param2));
else
    remainder = remainder1; % List of parameter indices still to be filled in
end

Nremain = length(remainder);

% Fill in the values of the remainder parameters
remainset = zeros(1,Nremain); % Parameter value index list
for i = 1:Nremain
    leftparams = paramsets(remainder(i),paramsets(remainder(i),:) ~= 0);
    if length(find(paramsets(remainder(i),:)))==1
        if remainder(i) == 5
            disp(string([paramsused{remainder(i)},' set to ',elec2]))
            remainset(i) = 1;
        else
            disp(string([paramsused{remainder(i)},' set to ',num2str(leftparams)]))
            remainset(i) = 1;
        end
    elseif remainder(i) == 5 % Do a bit differently for electrodes, as these are strings
        remainset(i) = input(['Set ',paramsused{remainder(i)},' index, average (0), ',elec2
            ↪ {:},': ']);
    else
        remainset(i) = input(['Set ',paramsused{remainder(i)},' index, average (0), ',
            ↪ num2str(leftparams),': ']);
    end
end

selectedparam = zeros(1,Nparams);

%% Determine statistics

% Set average pixel ratio based on input gap distances
gapstemp = indexfile{4:end,2};
pixratio = mean((configset(:,4)-configset(:,2))./gapstemp(~isnan(gapstemp)));

% Initialise result arrays
% 10 indicates 10 outputs of subplotSPE4
if compare1 == 1
    resultset = zeros(Nx,N1,10);
elseif compare1 == 2
    resultset = zeros(Nx,N1,N2,10);
```

```
end

if compare1 == 1
    for i1 = 1:N1
        for k1 = 1:Nx
            % Determine the index for each parameter required
            selectedparam(xparam) = find(xlist == xlist(k1));
            selectedparam(param1) = find(param1list == param1list(i1));
            avgi = 0;
            % Same for all remainder parameters
            for l1 = 1:Nremain
                if remainset(l1) == 0 % If something is to be averaged
                    Navgsub = length(find(paramsets(l1,:)~=0));
                    avgthing = zeros(Navgsub,10);
                    avgi = l1;
                else
                    selectedparam(remainder(l1)) = remainset(l1);
                end
            end

            if avgi == 0 % If nothing is to be averaged
                if Nparams == 3 % Different indexarray for different Nparams
                    frontindex = indexarray(selectedparam(1),selectedparam(2),selectedparam
                        ↪ (3));
                    bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam(3));
                elseif Nparams == 4
                    frontindex = indexarray(selectedparam(1),selectedparam(2),selectedparam
                        ↪ (3),selectedparam(4));
                    bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam(3),
                        ↪ selectedparam(4));
                elseif Nparams == 5
                    frontindex = indexarray(selectedparam(1),selectedparam(2),selectedparam
                        ↪ (3),selectedparam(4),selectedparam(5));
                    bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam(3),
                        ↪ selectedparam(4),selectedparam(5));
                elseif Nparams == 6
                    frontindex = indexarray(selectedparam(1),selectedparam(2),selectedparam
                        ↪ (3),selectedparam(4),selectedparam(5),selectedparam(6));
                    bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam(3),
                        ↪ selectedparam(4),selectedparam(5),selectedparam(6));
                end

                % Set correct value for configset
                Nconfig = selectedarray{2,frontindex};

                % Set electrode distance from value for configset
                eldist = gaps2(selectedparam(1));

                % Load ICCD pictures
                SPE_file = selectedarray{1,frontindex};
                bgr_file = zeros(size(SPE_file));
                [~,~,Ntemp] = size(bgr_file);
                for i = 1:Ntemp
                    bgr_file(:,:,i) = mean(selectedarray{1,bgrindex},3);
                end

                % Calculate parameters
                subplotSPE4
            else % If something is to be averaged
                for q = 1:Navgsub % For each of the averagings

                    selectedparam(remainder(avgi)) = q;
                    if Nparams == 3 % Same as before
```

```
            frontindex = indexarray(selectedparam(1),selectedparam(2),
                ↪ selectedparam(3));
            bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                ↪ (3));
        elseif Nparams == 4
            frontindex = indexarray(selectedparam(1),selectedparam(2),
                ↪ selectedparam(3),selectedparam(4));
            bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                ↪ (3),selectedparam(4));
        elseif Nparams == 5
            frontindex = indexarray(selectedparam(1),selectedparam(2),
                ↪ selectedparam(3),selectedparam(4),selectedparam(5));
            bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                ↪ (3),selectedparam(4),selectedparam(5));
        elseif Nparams == 6
            frontindex = indexarray(selectedparam(1),selectedparam(2),
                ↪ selectedparam(3),selectedparam(4),selectedparam(5),
                ↪ selectedparam(6));
            bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                ↪ (3),selectedparam(4),selectedparam(5),selectedparam(6));
        end

        % Set correct value for configset
        Nconfig = selectedarray{2,frontindex};

        % Set electrode distance from value for configset
        eldist = gaps2(selectedparam(1));

        % Load ICCD pictures
        SPE_file = selectedarray{1,frontindex};
        bgr_file = zeros(size(SPE_file));
        [~,~,Ntemp] = size(bgr_file);
        for i = 1:Ntemp
            bgr_file(:,:,i) = mean(selectedarray{1,bgrindex},3);
        end

        % Calculate parameters
        subplotSPE4
        % Save parameters in an array that averages them
        avgthing(q,1) = Nfrac;
        avgthing(q,2) = avgwidth;
        avgthing(q,3) = stdwidth;
        avgthing(q,4) = meandev; % Something is wrong with data 4+
        avgthing(q,5) = stddev;
        avgthing(q,6) = meanmov;
        avgthing(q,7) = stdmov;
        avgthing(q,8) = ignfrac;
        avgthing(q,9) = mean_int;
        avgthing(q,10) = std_int;
    end

    % Then average to get mean value
    avgd = mean(avgthing);

    Nfrac = avgd(1);
    avgwidth = avgd(2);
    stdwidth = avgd(3);
    meandev = avgd(4);
    stddev = avgd(5);
    meanmov = avgd(6);
    stdmov = avgd(7);
    ignfrac = avgd(8);
    mean_int = avgd(9);
```

```
            std_int = avgd(10);
        end

        % Set parameters in resultset

        resultset(k1,i1,1) = Nfrac;
        resultset(k1,i1,2) = avgwidth;
        resultset(k1,i1,3) = stdwidth;
        resultset(k1,i1,4) = meandev; % Something is wrong with data 4+
        resultset(k1,i1,5) = stddev;
        resultset(k1,i1,6) = meanmov;
        resultset(k1,i1,7) = stdmov;
        resultset(k1,i1,8) = ignfrac;
        resultset(k1,i1,9) = mean_int;
        resultset(k1,i1,10) = std_int;
    end
end
elseif compare1 == 2
    for i1 = 1:N1
        for j1 = 1:N2
            for k1 = 1:Nx
                % Determine current index for varied parameters
                selectedparam(xparam) = find(xlist == xlist(k1));
                selectedparam(param1) = find(param1list == param1list(i1));
                selectedparam(param2) = find(param2list == param2list(j1));
                % Determine index for fixed or averaged parameters
                for l1 = 1:Nremain
                    if remainset(l1) == 0
                        Navgsub = length(find(paramsets(l1,:)~=0));
                        avgthing = zeros(Navgsub,10);
                        avgi = l1;
                    else
                        selectedparam(remainder(l1)) = remainset(l1);
                        avgi = 0;
                    end
                end

                if avgi == 0 % If we don't average
                    if Nparams == 3
                        frontindex = indexarray(selectedparam(1),selectedparam(2),
                            ↪ selectedparam(3));
                        bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                            ↪ (3));
                    elseif Nparams == 4
                        frontindex = indexarray(selectedparam(1),selectedparam(2),
                            ↪ selectedparam(3),selectedparam(4));
                        bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                            ↪ (3),selectedparam(4));
                    elseif Nparams == 5
                        frontindex = indexarray(selectedparam(1),selectedparam(2),
                            ↪ selectedparam(3),selectedparam(4),selectedparam(5));
                        bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                            ↪ (3),selectedparam(4),selectedparam(5));
                    elseif Nparams == 6
                        frontindex = indexarray(selectedparam(1),selectedparam(2),
                            ↪ selectedparam(3),selectedparam(4),selectedparam(5),
                            ↪ selectedparam(6));
                        bgrindex = bgrarray(selectedparam(1),selectedparam(2),selectedparam
                            ↪ (3),selectedparam(4),selectedparam(5),selectedparam(6));
                    end

                    % Set correct value for configset
                    Nconfig = selectedarray{2,frontindex};
```

```
        % Set electrode distance from value for configset
        eldist = gaps2(selectedparam(1));

        % Load ICCD pictures
        SPE_file = selectedarray{1,frontindex};
        bgr_file = zeros(size(SPE_file));
        [~,~,Ntemp] = size(bgr_file);
        for i = 1:Ntemp
            bgr_file(:,:,i) = mean(selectedarray{1,bgrindex},3);
        end

        % Calculate stats
        subplotSPE4

    else % If we average
        for q = 1:Navgsub % Over each option for the averaging parameter
            selectedparam(remainder(avgi)) = q;
            if Nparams == 3
                frontindex = indexarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3));
                bgrindex = bgrarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3));
            elseif Nparams == 4
                frontindex = indexarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4));
                bgrindex = bgrarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4));
            elseif Nparams == 5
                frontindex = indexarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4),selectedparam(5));
                bgrindex = bgrarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4),selectedparam(5));
            elseif Nparams == 6
                frontindex = indexarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4),selectedparam(5),
                    ↪ selectedparam(6));
                bgrindex = bgrarray(selectedparam(1),selectedparam(2),
                    ↪ selectedparam(3),selectedparam(4),selectedparam(5),
                    ↪ selectedparam(6));
            end

            % Set correct value for configset
            Nconfig = selectedarray{2,frontindex};

            % Set electrode distance from value for configset
            eldist = gaps2(selectedparam(1));

            % Load ICCD pictures
            SPE_file = selectedarray{1,frontindex};
            bgr_file = zeros(size(SPE_file));
            [~,~,Ntemp] = size(bgr_file);
            for i = 1:Ntemp
                bgr_file(:,:,i) = mean(selectedarray{1,bgrindex},3);
            end

            % Calculate stats
            subplotSPE4

            % Save stats in array for averaging
            avgthing(q,1) = Nfrac;
            avgthing(q,2) = avgwidth;
            avgthing(q,3) = stdwidth;
```

```
                    avgthing(q,4) = meandev; % Something is wrong with data 4+
                    avgthing(q,5) = stddev;
                    avgthing(q,6) = meanmov;
                    avgthing(q,7) = stdmov;
                    avgthing(q,8) = ignfrac;
                    avgthing(q,9) = mean_int;
                    avgthing(q,10) = std_int;
                end

                % Average to obtain output values
                avgd = mean(avgthing);

                Nfrac = avgd(1);
                avgwidth = avgd(2);
                stdwidth = avgd(3);
                meandev = avgd(4);
                stddev = avgd(5);
                meanmov = avgd(6);
                stdmov = avgd(7);
                ignfrac = avgd(8);
                mean_int = avgd(9);
                std_int = avgd(10);

            end
            % Save stats
            resultset(k1,i1,j1,1) = Nfrac;
            resultset(k1,i1,j1,2) = avgwidth;
            resultset(k1,i1,j1,3) = stdwidth;
            resultset(k1,i1,j1,4) = meandev;
            resultset(k1,i1,j1,5) = stddev;
            resultset(k1,i1,j1,6) = meanmov;
            resultset(k1,i1,j1,7) = stdmov;
            resultset(k1,i1,j1,8) = ignfrac;
            resultset(k1,i1,j1,9) = mean_int;
            resultset(k1,i1,j1,10) = std_int;
        end
    end
end
end

%% Plot section
plot_on = 1;
if plot_on == 1
    subplasmaplots3
end
```

B.4 Raman processing script

This script is designed to run overnight for a fixed amount of time to process (part of) a sequence of measurements. For each measurement it filters the data, and determines the position of the plasma and the measurement range. Then it runs a Raman fitting script originally written by Conrad Hessels [63] to calculate the rovibrational temperatures of the gas. The script will be expanded to also calculate the gas densities.

```
% RDC (raman data collector) is designed to run overnight, where it loads
% PAC raman spectra and analyses their temperature and density distribution

% The following does not work; I saw myself forced to add the path manually
% If you figure out how to do it, by all means go ahead
save_dir = [fileparts(which('rdc.m')),'\']; % Find the folder rdc is in
% addpath(genpath(save_dir)) % Add this folder and all subfolders to matlab
```

```
% rmpath([save_dir,'\Raman filedump']) % Remove the filedump from it
% addpath('D:\RamanFinal') % Folder of the location of your files

load('rdc_values.mat')

% The file rdc_values.mat holds the variable 'rdc_cell', which saves in its
% columns the following:
% 1: The measurement number (digit taken from indexfile to find the file)
% 2: The full picture after noise and miss removal
% 3: The number of images rejected
% 4: The combination of parameters
% 5: The parameter that was set variable for these measurements
% 6: The electrode type
% 7: The fitted temperatures
% 8: The calculated densities
% 9: Edge temperatures
% 10: Position data
% 11: The spectrum of the central plasma line
% 12: Wavelength data
% 13: Temperature uncertainties
% 14: v01 line?

% Load array that couples measurement indices to parameters
tempparamselect3
if ~exist('SPEstring','var')
    SPEstring = input('Specify SPE filename (without filetype): ','s');
end
stop_hour = 14;
overnight = 0;

% Debug: Set empty rdc_cell
% rdc_cell = cell(1,13);
% Edit rdc_cell to fit the number of measurements
inspected = [rdc_cell{:,1}];
prev_Nmeas = length(inspected); % Measurements already done
measoverlap = intersect(inspected,to_inspect);
Noverlap = length(measoverlap);
if Noverlap < prev_Nmeas % Add any missing measurements to total set
    Nmeas = Nmeas + prev_Nmeas - Noverlap;
end

% If rdc_cell needs to be made longer, paste the old one in the new, larger one
if prev_Nmeas < Nmeas
    new_cell = cell(Nmeas,13);
    new_cell(1:prev_Nmeas,:) = rdc_cell(1:prev_Nmeas,:);
    rdc_cell = new_cell;
end

% Remove processed measurements from to_inspect
for i = 1:Noverlap
    to_inspect = to_inspect(to_inspect ~= measoverlap(i));
end

t = prev_Nmeas + 1; % Line in rdc_cell to start in
finished = 0; % Boolean to check if processing is done (for now)
if isempty(to_inspect)
    finished = 1;
end
c=clock;
start_day = c(3);

while finished == 0
    %% Determine indices
```

```
    frontindex = to_inspect(1); % Index of the image
    [x, y, z] = ind2sub(size(frontarray),find(frontarray == frontindex,1));
    % Notify the user if a measurement is used in multiple parameter sets
    if length(find(frontarray == frontindex)) > 1
        handle_i = 100;
        while ishandle(handle_i)
            handle_i = handle_i + 1;
        end
        figure(handle_i)
        text(0.2,0.5,['Measurement ',num2str(frontindex),' fits in multiple locations'])
        text(0.2,0.4,'Only the first position is accounted')
        text(0.2,0.3,'Add others manually for automatic picture processing')
    end
    backindex = backarray(x(1),y(1),z(1));
rdc_cell{t,1} = frontindex;
    parameter_set = base_params;
    parameter_set(y(1)) = param_vals(x(1),y(1));
rdc_cell{t,4} = parameter_set;
rdc_cell{t,5} = param_row{y};
rdc_cell{t,6} = elec_col2{z+2};

    %% Load SPE file

    % Determine filenames and locations
    spe_file = [SPEstring,num2str(frontindex),'.SPE'];
    bgr_file = [SPEstring,num2str(backindex),'.SPE'];
    file_dir = [fileparts(which(spe_file)),'\'];

    % Load the background and filter it
    [background,wave,ngrooves] = readSPE2(file_dir,bgr_file);
    background = double(background);
    for i = 1:size(background,3)
        background(:,:,i) = applyfilter(background(:,:,i),'hampel',[10 1 10 1]);
    end

    % Load the image and filter it
    [perper,wave2,ngrooves2] = readSPE2(file_dir,spe_file);
    perper = double(perper);
    for i = 1:size(perper,3)
        perper(:,:,i) = applyfilter(perper(:,:,i),'hampel',[10 1 10 1]);
    end

rdc_cell{t,12} = wave2;

    %% Find breakdown events

    % During a breakdown event, broadband radiation is emitted. This is shown
    % in the spectrum as a horizontal peak. In a normal spectrum, the
    % distribution of maxima in the spectrum will be somewhat random, and will
    % thus have a large standard deviation. However, due to the broadband
    % radiation, the maxima will have a very small spread in position during a
    % breakdown event. Spectra will be omitted if the standard deviation of
    % their vertical maxima is deviates 10% from the mean standard deviation.
    % This also catches excessive plasma emission

    breakdown_tol = 0.95; % Deviation required to count as breakdown
    if size(perper,3) > 1
        faults = zeros(1,size(perper,3));
        for i = 1:size(perper,3)
            imtemp = perper(:,:,i);
            [breakx,breaky] = find(imtemp == max(imtemp,[],1));
            faults(i) = std(breakx);
        end
```

```
    nobreak = find(faults >= mean(faults)*breakdown_tol); % Remaining spectra
else
    nobreak = 1;
end

% Find breakdowns in background
faults = zeros(1,size(background,3));
for i = 1:size(background,3)
    imtemp = perper(:,:,i);
    [breakx,breaky] = find(imtemp == max(imtemp,[],1));
    faults(i) = std(breakx);
end
nobreak2 = find(faults >= mean(faults)*breakdown_tol);

% Create the average background
back0 = mean(background(:,:,nobreak2),3);

% Subtract background from single images
spec_tot = mean(perper(:,:,nobreak),3)-back0;

%% Find misses

% Finding the location and width of the v01 line might be a bit redundant,
% but we need a simple one over here to find meashits, which in turn is
% needed for the more extensive search for the v01 line later
plotrange = 10;
plotcenter = find(mean(spec_tot,1) == max(mean(spec_tot,1)));


signal_strength = mean(mean(spec_tot([1:400 800:1024],plotcenter-plotrange:plotcenter+
    ↪ plotrange)));

filter1 = 'sgolay';
smoothstrength1 = 60;
filter2 = 'movmedian';
smoothstrength2 = 20;

sets = squeeze(mean(perper(:,plotcenter-plotrange:plotcenter+plotrange,:)-back0(:,
    ↪ plotcenter-plotrange:plotcenter+plotrange),2))/signal_strength;
filsets = smoothdata(smoothdata(sets,filter1,smoothstrength1),filter2,smoothstrength2)
    ↪ ;

% We need an initial guess to investigate the position of the plasma
if ~exist('plasma_center','var')
    % Display the v01 line so that user can input position
    plot(mean(filsets,2));
    drawnow
    plasma_init = input('Indicate plasma position: ');
    % Find the actual plasma position based
else
    plasma_init = plasma_center;
end

pits = find(islocalmin(mean(filsets,3)));
pitssq = (pits-plasma_init).^2;
close_val = find(pitssq == min(pitssq));
plasma_center = pits(close_val);

filmins = min(filsets(plasma_center-200:plasma_center+200,:),[],1);
filmeans = mean(filsets,1);
filnormmins = filmins./filmeans;

measthres = mean(filnormmins)+0.5*std(filnormmins);
```

```
    meashits = find(filnormmins <= measthres);
    meashits = intersect(meashits,nobreak);

rdc_cell{t,3} = size(perper,3)-length(meashits);

    %% Determine measurement range

    % For most positions in the spectrum, the v01 ridge is expected to be the
    % maximum. We start by finding the positions of the maxima along the laser path
    im0 = mean(perper(:,:,meashits),3)-back0;
rdc_cell{t,2} = im0;
    im0 = smoothdata(im0,2,'sgolay',3);
% [posx,posy] = find(im0 == max(im0,[],2));
    posy = zeros(1,size(im0,2));
    for i = 1:size(im0,2)
        posy(i) = find(im0(i,:) == max(im0(i,:),[],2),1);
    end

    % Since most maxima will probably lie around the ridge, we make an initial
    % guess that the ridge is the average of these positions.
    % This seems a bit circumstantial, but is necessary to avoid using the
    % plasma emission at the edges as position of the v01 line.
    % To remove outliers, we only use the data within a certain range of
    % this average, and find a new average using only this data
    findrange = 15;
    mposy = mean(posy);
    c = posy<mean(posy)+findrange & posy>mean(posy)-findrange;
    b=find(c);
    while isempty(b) % If no data lies within these ranges, extend range
        findrange = findrange + 5;
        b=find(posy<mean(posy)+findrange & posy>mean(posy)-findrange);
    end
    findrange = 15;
    posy2 = posy(b); % We then exclude outliers by looking only at the points +- 15 within
        ↪  this mean
    mposy2 = mean(posy2); % And compute the mean again
    del = abs(mposy2-mposy);
    while del > 1 % Until sufficiently converged.
        mposy = mposy2;
        b=find(posy<mposy2+findrange & posy>mposy2-findrange);
        posy2 = posy(b);
        mposy2 = mean(posy2);
        del = abs(mposy2-mposy);
    end
    mposy = round(mposy2);
    delposy = zeros(size(posy));
    delposy(c) = (posy2-mposy);

    im0temp = zeros(size(im0));
    perpertot = mean(perper(:,:,meashits),3);
    back0temp = zeros(size(back0));
    perpertemp = zeros(size(perpertot));

    startshift = zeros(1,1024);
    endshift = startshift;

    for i = 1:size(im0,1)
        if delposy(i) < 0
            startshift(i) = delposy(i);
            endshift(i) = 0;
        else
            startshift(i) = 0;
            endshift(i) = delposy(i);
```

```
        end
        im0temp(i, 1-startshift(i) :  end-endshift(i) ) = im0(i, 1+endshift(i) :  end+
            ↪ startshift(i));
        perpertemp(i, 1-startshift(i) :  end-endshift(i) ) = perpertot(i, 1+endshift(i)
            ↪ :  end+startshift(i));
        back0temp(i, 1-startshift(i) :  end-endshift(i) ) = back0(i, 1+endshift(i) :  end+
            ↪ startshift(i));
    end
    im0 = im0temp;
    back0 = back0temp;
    back0 = smoothdata(back0,2,'sgolay',3);
    perper = perpertemp;
    perper = smoothdata(perper,2,'sgolay',3);
    clear im0temp
    clear back0temp
    clear perpertemp

    v01line = mean(im0(:,mposy-10:mposy+10),2);
    v01smooth = smoothdata(v01line,'movmean',50);
    gapmin = min(v01smooth(plasma_center-200:plasma_center+200));
    gapminx = find(v01smooth == gapmin);
    % Maximum on the left and right of the gap
    lmax = max(v01smooth(1:gapminx));
    lmaxx = find(v01smooth == lmax);
    rmax = max(v01smooth(gapminx:end));
    rmaxx = find(v01smooth == rmax);
    % Gap edges are the points where the v01line has gone up halfway from
    % the gap bottom to the maximum value
    gapl = find(v01smooth(1:gapminx)>(gapmin+lmax)/2,1,'last');
    gapr = find(v01smooth(gapminx:end)>(gapmin+rmax)/2,1,'first')+gapminx;
    plasma_r = (gapr-gapl)/2;
    gapmid = round((gapr+gapl)/2);

    % Measure temperature through the entire streamer
    MeasOptions.ipix = round(gapmid-3*plasma_r);
    % Prevent crossing the ends of the domain
    if MeasOptions.ipix < 1
        MeasOptions.ipix = 1;
    end
% MeasOptions.ipix = 635;

    MeasOptions.fpix = round(gapmid+3*plasma_r);
    if MeasOptions.fpix > 1024
        MeasOptions.fpix = 1024;
    end
% MeasOptions.fpix = 685;

    % Make sure the ranges extend equally to the left or the right of the
    % plasma center
    if abs(MeasOptions.fpix - gapmid) > abs(MeasOptions.ipix - gapmid)
        MeasOptions.fpix = 2*gapmid - MeasOptions.ipix;
    elseif abs(MeasOptions.fpix - gapmid) < abs(MeasOptions.ipix - gapmid)
        MeasOptions.ipix = 2*gapmid - MeasOptions.fpix;
    end

    % Step size in multiples of 5 such that at most 20 steps are taken
    N_spec_step = 20;
    measlength = MeasOptions.fpix - MeasOptions.ipix;
    MeasOptions.rowbin = 5*ceil(measlength/5/N_spec_step);
    MeasOptions.plasma_center = gapmid;
    MeasOptions.plasma_r = plasma_r;
    MeasOptions.pixsize = 12.4375; % Number of pixels per mm
    MeasOptions.burnerwidth = 10;
```

```
    MeasOptions.burnerleft = gapmid + round(MeasOptions.burnerwidth*MeasOptions.pixsize)
        ↪ /2;
    MeasOptions.burnerright = gapmid - round(MeasOptions.burnerwidth*MeasOptions.pixsize)
        ↪ /2;


    preloads.background = back0;
    preloads.wave = wave;
    preloads.ngrooves = ngrooves;
    preloads.perper = perper;
    preloads.wave2 = wave2;
    preloads.ngrooves2 = ngrooves2;

    clear perper
    clear background

    % Determine laser input wavelength needed to adjust for spectrograph
    % calibration
    v01_raman_shift = 2327.22; %cm-1
    lv01 = wave(mposy);
    kv01 = 10^7/lv01;
    klas = kv01 + v01_raman_shift;
    llas = 10^7/klas;
    FitOptions.laser = llas;

    plotoutput = 0;

    % Run the thermometry script
    Fitting_new_v6

    plotoutput = 0;

    % Plot the most central measurement together with the fit
    i = round((size(Im,1)+1)/2);

    Imnorm(i,:) = Im(i,:)./max(Im(i,:));
    Inorm(i,:) = I(i,:)./max(I(i,:));
    Resscale(i,:) = Imnorm(i,:)-Inorm(i,:);
    SSR = sum(Resscale(i,:).^2);

    spec_plot = [lambdam; Imnorm(i,:); Inorm(i,:); Resscale(i,:)];
    % To actually plot it, input the following line in the command window:
% plot(spec_plot(1,:),spec_plot(2:4,:))
rdc_cell{t,11} = spec_plot;

    N_temps = length(width);
rdc_cell{t,10} = width;

rdc_cell{t,7} = var;
rdc_cell{t,13} = varacc;

    % var(:,1) gives rotational temperature
    % var(:,6) gives v01 temp
    % var(:,7) v1v temp?


    %% Measure edge temperatures

    dim1 = size(preloads.perper,1);
    T_edge = [0 0];

    % Make sure the two measurements are equidistant
    if gapmid < dim1/2
```

```
        MeasOptions.ipix = 1;
    else
        MeasOptions.ipix = 2*gapmid - dim1;
    end
    MeasOptions.fpix = MeasOptions.ipix + MeasOptions.rowbin;

    Fitting_new_v6
    T_edge(1) = var(1,1);

    if gapmid > dim1/2
        MeasOptions.fpix = dim1;
    else
        MeasOptions.fpix = 2*gapmid;
    end
    MeasOptions.ipix = MeasOptions.fpix - MeasOptions.rowbin;

    Fitting_new_v6
    T_edge(2) = var(1,1);

% T_edge = [350 350];

rdc_cell{t,9} = T_edge;

    %% Calculate density

    % Using the ideal gas law, we calculate the number density of the gas
    % along the plasma

    p = rdc_cell{t,4}(1)*100;
    kb = 1.38064852*10^-23;
    n0 = p/kb./T_edge;

    signal0 = [0 0];

    i1 = find(wavelengthdata > 598,1,'first');
    i2 = find(wavelengthdata < 608,1,'last');

    signal0(1) = sum(sum(im0(1:MeasOptions.rowbin,i1:i2)));
    signal0(2) = sum(sum(im0(end-MeasOptions.rowbin+1:end,i1:i2)));
    % We assume the density is proportional to the total signal within the
    % measurement range
    signal_to_n0 = n0./signal0;

    first_plasma = round(round(gapmid-3*plasma_r) + 0.5*MeasOptions.rowbin); % Leftmost
        ↪ measurement point
    space_offset = mod(first_plasma,MeasOptions.rowbin); % Set the center of this point as
        ↪  the center for all points
    bin_pos = space_offset:MeasOptions.rowbin:dim1; % Extend the position of the bins
        ↪ across the whole domain
    signal_to_n = linspace(signal_to_n0(1),signal_to_n0(2),length(bin_pos)); % Determine
        ↪ signal_to_n0 across the whole domain; account for laser focus
    first_index = find(bin_pos == first_plasma);
    signal_array = zeros(1,N_temps);
    for i = 1:N_temps
        signal_array(i) = sum(sum(im0(round(gapmid-3*plasma_r) + MeasOptions.rowbin*(i-1):
            ↪ round(gapmid-3*plasma_r) + MeasOptions.rowbin*i-1,:)));
    end
    n = signal_array.*signal_to_n(first_index:first_index+N_temps-1);

rdc_cell{t,8} = n;


    %% Finish up
```

```
    % If this measurement number corresponds to a measurement appropriate
    % in multiple places, copy those places as well


    N_complete = length(find([rdc_cell{:,1}]));
    save_dir = [fileparts(which('rdc.m')),'\']; % Find the folder rdc is in
    save([save_dir,'rdc_values'],'rdc_cell','SPEstring','indexstring','N_complete')

    t = t + 1;

    c=clock;
    if overnight == 0
        c(3) = 0;
    end
    if c(3) ~= start_day && c(4) >= stop_hour % If the next morning has begun
        finished = 1; % End run
    end
    if length(to_inspect) == 1 % If this was the last spectrum
        finished = 1; % End run
        to_inspect = [];
    else
        data_left = 0;
        to_inspect = to_inspect(2:end); % Otherwise remove this spectrum from to do list
        for i = 1:length(to_inspect)
            frontindex = to_inspect(i);
            spe_file = [SPEstring,num2str(frontindex),'.SPE'];
            file_dir = [fileparts(which(spe_file)),'\'];
            if isfile([file_dir spe_file])
                data_left = 1;
            end
        end
        if data_left == 0
            finished = 1;
        end
    end
% finished = 1;
end

disp(['Files completed: ',num2str(prev_Nmeas),' + ',num2str(N_complete-prev_Nmeas)])
disp(['Files left: ',num2str(length(to_inspect))])

% Backup of the final values
save([save_dir,'rdc_backup'],'rdc_cell','SPEstring','indexstring')
save('D:RamanFinal/rdc_values','rdc_cell','SPEstring','indexstring','N_complete')
```