Atmospheric Pressure Weakly Ionized Plasma Reactor
Based on the Corona Discharge

By
ERIK WEMLINGER

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science
DECEMBER 2012
©Copyright by Erik C. Wemlinger, 2012
All Rights Reserved
To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of Erik Charles Wemlinger find it satisfactory and recommend that it be accepted.

Patrick Pedrow, Ph.D., Chair

Manuel García-Pérez, Ph.D

John Schneider, Ph.D.

George LaRue, Ph.D.
ACKNOWLEDGMENT

I would like to acknowledge the contributions of my committee, Patrick Pedrow who has mentored and encouraged me in this research. Without his guidance I could not have completed this project. Manuel García-Pérez and WSU Biological Systems Engineering for hosting the atmospheric pressure weakly ionized plasma reactor. Additionally I greatly appreciate the interest he has shown in my work and his help connecting me with other researchers with similar interests. John Schneider has provided a keen eye for detail that has helped me stay focused and challenged me to critically examine my own work. I would like to thank George LaRue for his dual physics and engineering background which has kept me grounded in reality.

This research would not have been possible without the assistance from the following individuals. The author has benefited from technical discussions with Shyam Sablani and John Yates. Reactor design and construction was facilitated by Garrison Riley Held, Miles Pepper, David Savage and Gary Held. Essential plasma reactor insight was obtained from William Lekobou and the other members of the WSU Cold Plasma Group. The food decontamination work would not have been possible without the collaborative effort of Daniela Bermudez-Aguirre and Abigail Elizabeth Moody which I greatly appreciate.

I would like to thank Dr. Valerie Lynch-Holm (Franceschi Microscopy and Imaging Center, WSU) for her help and expertise during electron microscopy studies.

I would like to thank AVISTA utilities for the use of the GE JWW5 bench top transformer, which made the synchronicity study possible.

To my Dad who taught me how to work with my hands.

In conclusion, I would like to thank my family for their help and encouragement. Beth, Zachary, and Joshua for the love and support in everything. I have also been blessed to be a part of Cora, Declan, and Elliott’s lives. They encourage me to have hope for the future.
Lastly, I would like to thank my mother for her support and love. Homeschooling all of us K-12, she instilled a love for God and of learning. I love to learn and am looking forward to a future of learning and discovery.

Of course all credit goes to Jesus Christ.

*In the beginning was the Word, and the Word was with God, and the Word was God. The same was in the beginning with God. All things were made by him; and without him was not any thing made that was made. In him was life; and the life was the light of men.*

*John 1:1-4*
Atmospheric Pressure Weakly Ionized Plasma Reactor

Based on the Corona Discharge

Abstract

by Erik Charles Wemlinger, Ph.D.
Washington State University
December 2012

Chair: Patrick Pedrow

Atmospheric pressure weakly ionized plasma (APWIP) is being used to treat or process goods and materials because it only activates the surface without modification of the bulk material. This work describes research into the synchronicity of corona discharges and some applications of APWIP. A reactor was built to generate corona discharges using an array of needles, the geometry of each needle being consistent with point-to-plane configuration. The interaction between corona discharges in the needle array was studied. The reactor was then used to decontaminate fresh produce inoculated with \textit{E. coli} ATCC 1177 and to deposit films via plasma polymerization of acetylene.
## Contents

1 Introduction 1
   1.1 Weakly Ionized Atmospheric Pressure Plasma 3
   1.2 Dielectric Barrier Discharge 4
   1.3 Radio Frequency Plasma 5
   1.4 Corona Discharge 6
      1.4.1 Corona Pulses 6
   1.5 APWIP Applications - Produce 9
   1.6 APWIP Applications - Acetylene Deposition 10
   1.7 Conclusion 12

2 Reactor Design 15
   2.1 Early Reactor Prototypes 16
   2.2 Final Reactor Version 18
      2.2.1 Observations 22
      2.2.2 Power Supply 23
      2.2.3 Measurement Equipment 25
      2.2.4 Needle Array and Grounded Mesh 27
      2.2.5 Electrostatics 30
      2.2.6 Equivalent Circuit 41
   2.3 Conclusion 42

3 Sympathetic Corona Current Pulses 44
   3.1 Gas Dynamics in a Non-Uniform Electric Field 45
   3.2 Electrical Breakdown in Gas 49
6.2 Summary .................................................................................. 103
6.3 FUTURE WORK ........................................................................ 105

Bibliography .................................................................................. 117

Appendices

A Additional Reactor Information .................................................. 119
   A.1 Short Gap Needle Deposition ................................................. 126
   A.2 Additional FESEM Film Deposition Images ......................... 128

B Power Calculation Illustrated ..................................................... 131

C Thin Film Thickness .................................................................... 133

D MATLAB Code ........................................................................... 135
   D.0.1 Matlab Code GUI Code: .................................................. 135
   D.0.2 Functions Called by MATLAB GUI: ............................... 159
   D.1 Synchronicity Sorting Script .............................................. 182
   D.2 Power Calculation Script .................................................. 198
List of Figures

2.1 Reactor 1 .............................................................. 16
2.2 Reactor 2 .............................................................. 17
2.3 Reactor 3 .............................................................. 19
2.4 Ground mesh .......................................................... 21
2.5 Reactor safety electrical diagram .................................. 23
2.6 Reactor circuit/operation diagram .................................. 24
2.7 Pearson example current .............................................. 27
2.8 Needle array .......................................................... 28
2.9 Needle radius of curvature ........................................... 29
2.10 Needle 1 radius of curvature ....................................... 30
2.11 Electric field point-to-plane diagram ............................... 31
2.12 Needle array and ground mesh ..................................... 33
2.13 Ground mesh .......................................................... 34
2.14 Electric potential and electric field ................................. 35
2.15 2D electric potential contours ...................................... 36
2.16 2D electric potential contours needle tip to ground ............ 37
2.17 2D electric potential contours for needle shaft ................ 38
2.18 2D electric potential contours at needle tip ..................... 39
2.19 Electric field comparison ............................................. 40
2.20 Reactor equivalent circuit .......................................... 42
3.1 Drift velocity data ..................................................... 46
3.2 Gas mobility ............................................................ 47
3.3 Ar+ crossing time ...................................................... 48
DEDICATION

To my mother for her love of Our Lord and Savior Jesus Christ and learning.

And the LORD spake unto Moses, saying, Speak unto Aaron and unto his sons, saying, On this wise ye shall bless the children of Israel, saying unto them,

The LORD bless thee, and keep thee:
The LORD make his face shine upon thee, and be gracious unto thee:
The LORD lift up his countenance upon thee, and give thee peace.
And they shall put my name upon the children of Israel; and I will bless them.

Numbers 6:22-27
Chapter 1

Introduction

Corona discharges can be observed in nature under the correct conditions at the tops of trees or other similarly sharp points, typically during thunderstorms. Some of the earliest literary references to corona discharges date back to the Greeks. Terpandros of Lesbos, fl. c. 675 BCE[46] describes the corona discharge as Castor and Pollux, more commonly referred to as St. Elmo’s fire or the fire of Mazu.

'Come hither, ye mighty sons of Zeus
And Leda; Olympus leave behind,
Your flashing home. With gladsome mind,
Castor and Pollux, appear to us.

'Ye who traverse the whole expanse
Of the earth and over the spacious seas
On your swift-footed steeds, ye save with ease
All men whom to meet chill Death did chance.

'On the tops of the well-bench'd ships ye leap,
Gleaming afar in the murky night.
As ye land on its cables ye bring a light
To the swift black ship which sails o'er the deep.'
Although experimentalists have dabbled with electricity throughout history, it was Michael Faraday who first noted a glow from a positive electrode in 1838 [32]. Led by Townsend [96, 95], great strides have been made in understanding corona discharges others who have made major contributions include Peek [71, 72, 73, 74, 75], Trichel [97], Meek [59], and Loeb [57, 58]. However, corona research is a continuing effort with new applications and understanding continually being added to the field. Some of the initial atmospheric pressure work into coronae and electrical breakdown related to high voltage (HV) power lines and ways of preventing power losses and radio interference due to this phenomenon. That motivation from the electric power engineering discipline has yielded a substantial body of work on coronae in atmospheric pressure air.

Atmospheric Pressure Weakly Ionized Plasma (APWIP) has emerged as a method to treat or process goods and materials. This research leverages previous research on corona discharges in a point-to-plane geometry and applies that established knowledge to a more complicated configuration consisting of an array of needle points opposing a grounded metal mesh. Multiple points are used in this work with industrial processing applications in mind. Introducing multiple points on which discharges can occur means that the discharge near one needle can interact with and initiate a discharge on another needle. This research elucidates these needle-to-needle interactions in an acetylene/argon environment and illustrates APWIP technology with applications in food processing and film deposition.

The rest of this chapter summarizes methods for generation of APWIP and then gives an overview of how non-thermal plasma is currently used in produce decontamination and film deposition. Chapter 2 introduces the reactor which was built to facilitate this research. Needle-to-needle interactions or sympathetic pulses are discussed in chapter 3. Chapter 4 illustrates using the reactor to decontaminate food. Chapter 5 discusses work with acetylene and film deposition.
1.1 Weakly Ionized Atmospheric Pressure Plasma

There are a variety of ways to generate weakly ionized atmospheric pressure plasma (weakly ionized plasma is also referred to as non-thermal plasma or cold plasma), which include dielectric barrier discharge (DBD), RF, and corona discharge. While other methods for generating weakly ionized plasmas exist, the DBD, RF and corona discharge methods are very promising for industrial applications. These plasma technologies will be reviewed here.

Some of the early work looking at the electrical breakdown of gases was completed by Townsend who described the gas breakdown in terms of three parameters [39, 96]. These three parameters, listed here, have come to be known as Townsend Coefficients: $\alpha$ or the Townsend ionization coefficient is equal to the ionization frequency over the electron drift velocity, while $d$ is the distance between electrodes such that a primary electron will produce $e^{\alpha d} - 1$ positive ions; $\beta$ is the second Townsend coefficient which shows electron loss due to attachment ($\beta=0$ for electropositive gases); $\gamma$ is the third Townsend coefficient representing the probability of secondary electron emission from the cathode when it experiences a flux of positive ions. Electrical discharges can be divided into these categories: an electron avalanche, streamers, glows, and sparks. For the plasma to be considered weakly ionized, only the electron avalanches, streamers, and glows are allowed. Sparks are thermal plasmas which can be used to break down materials in applications such as incineration. APWIP requires a two-temperature model involving both hot or energetic electrons and cold or non-energetic ions and neutrals. While thermal plasmas are single temperature, both the electrons and ions are hot or energetic. Streamers are transitional breakdowns between avalanches and sparks. Streamers result in ionized channels and have at their heads local net electric fields (due to local volume space charge density) intense enough to ionize the nearby gas and thereby self-propagate. If a streamer channel bridges the gap to terminate on the opposite electrode, a spark can be formed [39]. An electron avalanche occurs when a free
electron is accelerated through a field and impacts an atom, molecule, or ion with the effect of freeing other electrons. When an avalanche occurs, this cascading effect is quenched short of freeing enough electrons to yield a local electric field strong enough for self-propagation as a streamer.

It is interesting to note, prior to discussing specific discharge types, the use of argon in the reactor. Atmospheric glows are easier to achieve in argon and helium than in other gases and much easier than in electronegative gases (control of – that is, elimination of – water vapor is another reason to avoid using air directly from the atmosphere.) According to Hartnett et al. [39] there are two viable explanations as to why these noble gases work better. The first hypothesis is that the lower voltage required for breakdown keeps the temperature down and avoids thermal instabilities. The second hypothesis suggests that the resonant absorption of UV radiation causes fast ionization transfer which in turn causes the arc channel to be much larger than for electronegative gases [39]. The fast ionization is important because corona onset occurs at a lower voltage, which means plasma processing can be achieved at a lower voltage. It is for these reasons that argon was used as a carrier gas in these experiments: it is relatively inexpensive, readily available, and easily converted to the plasma state.

1.2 Dielectric Barrier Discharge

Dielectric barrier discharges (DBD) are the most common method for non-thermal plasma generation. This is because the dielectric prevents spark formation and has a number of industrial applications. DBD is operated at frequencies between 0.05 and 500 kHz with the dielectric being on one or both electrodes. These discharges from the surface have random origination and are composed of microdischarges in the gap [39]. The prevention of sparks is a major advantage for DBD; however, there is also a risk of depositing the dielectric material onto the substrate being processed. The addition of the dielectric material complicates the
reactor design and for this reason the bare metal point-to-plane geometry was used for this work.

1.3 Radio Frequency Plasma

RF plasmas can be broken down into two major configurations: capacitively coupled plasma and inductively coupled plasma. The major benefits of RF plasmas come from the fact that ion densities follow the time average of the electric field. Applications of RF discharge consist of sputtering film deposition and etching; this, however, is typically done at low pressure [39]. The drawbacks to RF plasmas include dependence on helium (helium is particularly troublesome due to dwindling supplies), expensive power supplies, matching networks, and the need for electrode cooling. Using argon instead of helium in an RF plasma was problematic in this study, as the voltage needed to be dropped after ignition and plasma operation had to be limited to less than 30 min intervals. RF plasmas need more experimental equipment compared to a corona discharge, making the operation more complex and more susceptible to maintenance-related down time. These characteristics make the use of RF plasma less appealing for industrial applications which is why 60 Hz AC excitation was used here.

There are two common RF discharge modes: \( \alpha \) and \( \gamma \) modes. The \( \alpha \) mode occurs when the plasma is weakly ionized and consists of avalanches and streamers only. The \( \gamma \) mode occurs when secondary electron emission becomes significant. Operating in the \( \gamma \) mode requires a great deal of power and can damage the electrode surface; it is for this reason that this mode is avoided. An RF plasma discharge is operated in the \( \alpha \) mode, although this term is not typically used for corona discharges used in this work it is similar [39].
1.4 Corona Discharge

According to Hartnett et al., "corona is a weakly luminous discharge which usually appears at atmospheric pressure near sharp points, edges, or thin wires where the electric field is sufficiently large [39]." Some of the most notable work on coronae began in the early 1900’s when Townsend examined the ionization of gases [96]. For gaps less than 5 cm at atmospheric pressure the Townsend breakdown mechanism is relatively homogeneous with the development of independent avalanches. For larger gaps (greater than 6 cm), avalanche electric fields can disturb the external electric field and breakdown typically occurs in argon for an E/p (kV/cm) = 2.7 [39] at atmospheric pressure, where p is the pressure. Streamers occur when the electric field of the space charge is on the order of the external electric field and both are sufficient to cause impact ionization by free electrons. There are two proposed breakdown mechanisms. The first suggests that the quasi-self-sustained streamers have a low channel conductivity and that their propagation is autonomous and independent of the anode. This model works well for high voltages, low average electric fields, and long gaps. The second proposed mechanism is that the streamer channel is an ideal conductor with the propagation velocity being that of the electron drift velocity. This model compares well with experimental results [39].

This research utilizes corona discharge because of its scalability and the ease with which it can be generated. The reactor contains an array of 12 needles with a stainless steel ground ring on which a stainless steel wire mesh is attached to provide a point-to-plane geometry.

1.4.1 Corona Pulses

Corona discharges are still being studied after more than 150 years and there is still a great deal of research needed. This section contains a brief overview of current knowledge on pulses, focusing on the positive polarity and how some of the terms have changed over time.
It is also important to realize that most of this research has been done for DC discharges. This will conclude the section with contributions this dissertation makes in this field.

By the 1970’s major progress had been made, primarily by Peek [71, 72, 73, 74, 75], Trichel [97], and Loeb [57, 58], in naming and photographing different current pulses. Giao and Jordan compiled correspondence in which Hermstein does a nice job of summarizing positive DC corona discharges listed by increasing potential:

1. “Positive onset streamers (POS)
2. For the following glow mode of stable glow the name Hermstein’s glow (HG) has been introduced.
3. Breakdown streamers, according to the negative mode, should be called positive brush streamers (PBS), especially because they do not in all cases lead to breakdown.
4. Diffused discharge; this is a stem brush streamer (SBS), sometimes reaching the cathode. [34]”

Among the key observations noted by Giao and Jordan [34] is that glows are used to describe a corona discharge in which no current was measured (due to limitations associated with their current sensors). The term streamer was used to describe a corona discharge for which the transient current could be measured with an oscilloscope. Another interesting observation was that for AC excitation only positive current pulses were observed, which was attributed to the breakdown voltage on the anode being less than that of the cathode. This is an interesting observation since, later, Czech et al. [26] makes the opposite assertion. The oscillograms provided by Giao and Jordan [34] are qualitative and do not provide magnitude information. Their work is interesting, nonetheless, because they are qualitatively consistent with observations made in this work on current pulses in acetylene/argon gas mixtures using
a 60 Hz AC voltage excitation.

With improvements in oscilloscope sensitivity, current pulses were eventually observed for glow discharges Morrow [66]. In 1997 Morrow [66] modeled the glow discharge for a wire cylinder geometry; in his simulation for DC excitation, the glow would transition into a steady pulse train after about 60 µs. The pulse frequency depended on the geometry, but was in the MHz range with current amplitudes less than a mA. A key thing to keep in mind with respect to this simulation is that it took 60 µs from the introduction of the seed electrons to the steady state pulse glow. The initial current pulse was in the mA range while the steady state glow was in the µA range. Kudu et al. [52] used a sphere to plane geometry and, using pressure-voltage diagrams, suggested the following types of positive corona.

1. Burst pulses, which are trains for electron avalanches initiated by external radiation or electrons from negative ions.
2. Positive glow, this occurs when the mean free path for the gas ionizing photons is long or the photoelectrons from the cathode can cross the gap without being attached. The glow may be oscillatory and unstable.
3. Streamers, occur when either of the above is strong enough to bring the field at the anode to about zero due to the charge in the streamer head which moves away from the needle ionizing molecules as it moves toward the cathode.
4. Glow-spike corona, which is between a glow and a streamer in terms of observation with no current pulse measured. It occurred when the oxygen percentage was less than 1% and the pressure was higher than 5 kPa.

This is claimed to be the first observation of a glow-spike corona and is still not well understood [52]. Some current oscillograms published by Kudu et al. [52] for air are similar
to those reported here for acetylene/argon gas mixtures.

Moving away from measuring current pulses to look at photographs of pulses, Czech et al. [26] makes some interesting distinctions compared with Giao and Jordan [34] about the order of pulses and suggests that the negative corona breakdown voltage is lower than the positive. Czech et al. [26] observed positive corona discharge starting with glows at the lowest potential with onset streamers, then streamers, and finally arcs at the highest potential. All of this work used DC voltage with a ballast resistor to limit the current when an arc occurred.

Thus, while there is a large body of knowledge about corona discharges, many details are missing. This dissertation is experimental in nature and contributes to the body of knowledge in terms of the nature of point-to-plane current pulses in acetylene/argon gas mixtures. In addition to observing the current pulses for a single point, correlations between needles in the needle array are reported.

1.5 APWIP Applications - Produce

Traditional microbial inactivation methods used to treat food are becoming less efficient as microbial strains become resistant to those methods [24]. Moreover, as the world’s population continues to grow, the demand for food that is free from harmful microbes will only increase. APWIP is particularly well suited for treating food products because it has minimal impact on the surface of produce as microbial inactivation is accomplished via microbial interaction with photons, electrons, ions, free radicals, and molecules in excited or non-excited states that are present in the APWIP [33]. Fernandez et al. [33] studied inoculation of *S. typhimurium* using a nitrogen plasma jet. Plasma jets typically use one of the three excitation methods described earlier. However, the plasma is formed in a constrained volume, then exits into a neutral environment and, in this case, is applied to the surface
infected with *S. Typhimurium*. Fernandez *et al.* [33] did note that effectiveness of treatment decreased when bacteria concentration increased such that microbes formed layers, allowing the deepest layer of bacteria to be protected by the dead upper layers. Bacteria at these high concentrations, however, were not typically encountered naturally.

Grzegorzekowski *et al.* [38] utilized an argon plasma jet to treat lambs lettuce and noted more discoloration of the lettuce compared to UV inoculation. This could be due to the method of treatment since excited argon interacted simultaneously with air and the lettuce surface. Ozone is very effective for sterilization, but ozone is also very harsh on fruits and vegetables making it easy to damage the produce [27]. Ozone doses to produce should therefore be limited. Preliminary research has been completed with argon APWIP and is discussed in detail in Chapter 4.

Another food application, which is closely related to the next section on films, is food packaging. There is growing interest in functionalizing inert surfaces used in food packaging and grafting proteins or enzymes onto these packaging surfaces to serve antimicrobial functions.

### 1.6 APWIP Applications - Acetylene Deposition

As mentioned above, argon has the unique property, along with helium, of having a low dielectric strength. According to Berger [22], the dielectric strength of helium is 0.15 and argon is 0.18 relative to nitrogen. These are the lowest dielectric strengths listed by Berger [22]. Berger [22] also lists the dielectric strength of acetylene as being 1.10 times that of nitrogen. This suggests that argon and acetylene make an effective gas mixture, since it is easy to generate plasma from argon and the acetylene will quench streamers before they transition to arcs. In addition, the Penning effect reduces the voltage needed for ionization in the acetylene/argon gas mixture. As Sahin *et al.* [106] notes, this effect is most significant (at
0.1% C₂H₂ in argon) in a uniform field. The higher concentration of acetylene used in this work and non-uniformity of the electric field acts to quench the streamers from developing into an arc as streamers move away from the needle tip while, near the needle tip, the intense electric field provides plenty of impact ionization.

A great deal of research, mostly low pressure, has been conducted on plasma polymerization of acetylene. Non-thermal plasma has been used to investigate the growth of nanodiamond thin films at low pressure [37]. Thin film deposition of these nanodiamond films is of great interest but thorough investigations have been limited to low pressure reactors. Plasma-polymerized acetylene has also been shown to be a good surface for bioactive proteins via research conducted at low pressures using an RF plasma [102]. Uchida et al. used RF plasma at low pressure and assumed a plug flow reactor model to deposit ultrathin, ultrasmooth film using acetylene as the precursor gas. In this reactor Uchida et al. measured film growth within milliseconds, but the acetylene did not appear to undergo chain propagation on the activated surface [98]. This work demonstrates some of the diversity of general plasma processing. Redolfi et al. investigated the efficiency of APWIP to oxidize volatile organic compounds, including acetylene in part because of its key role in soot generation [81].

Salge [85] has worked on plasma-polymerizing acetylene using DBD at atmospheric pressure and looking primarily at discharge uniformity, but Salge has also deposited some plasma-polymerized acetylene on plastic film. That setup consisted of a 60 Hz power supply attached to a Tesla transformer with a spark gap. The firing of the spark gap caused the secondary winding of the Tesla transformer to resonate, applying a large voltage across the electrodes. The gas flow was perpendicular to the DBD region. The acetylene flow rate was 1/100 that of the argon which appears to have had a flow rate of 51 sccm [85]. Salge [85] reported deposition rates of 40 m/min with an FTIR spectrum similar to altered polyacetylene.

Although Acetylene is commonly used in plasma deposition the process is poorly described which is attested to by Le Dû et al. [55]. While the process may not have been
precisely described basic models for hydrocarbon deposition have been proposed [94, 87]. Shepsis et al. outlines general polymerization model which focuses on radicals and can be broken down into 3 major steps. First the monomer is excited in the gas phase; next the radical forms chain with the monomer, in this case acetylene [87]. The last step occurs on the substrate where film deposition occurs, the film is formed when gas phase acetylene molecules and acetylene polymer chains arrive at radical sites on the substrate surface [87].

Utilizing non-thermal plasma at low pressure has been shown to work very well for thin film deposition. Our goal is to deposit film at atmospheric pressure. Deposition observed in this work is described in chapter 5.

1.7 Conclusion

Corona discharges in the acetylene/argon gas mixture and their interactions are fundamental components of this work. As illustrated in the section on corona pulses (section 1.4.1), names and the meanings of those names referring to the different kinds of discharges observed can change. Without describing these changes in detail, it seems prudent to provide an overview of the corona pulse names that will be used in this work and an intuitive meaning for the pulse. The descriptions below are discussed in terms of point-to-plane geometry, meaning the electric field is non-uniform. These terms are based on positive DC point-to-plane geometries which are true half the time for AC excitation. The AC was chosen for its stability since charge can build up on the reactor walls and cause arcing after extended DC operation. While negative corona pulses are observed, many of them have very low amplitude. This work focuses on the positive corona discharges due to the rich nature of the discharges observed.

- Avalanche: An avalanche occurs when an ionizing event (natural ionizing radiation, ultraviolet light, detachment of an electron from a negative ion, etc.) liberates an
electron and the electron is then is accelerated by an applied electric field toward the anode (positive electrode) or away from the cathode (negative electrode). The electron is accelerated by an electric field and is only slowed down by impacts with atoms, ions, or molecules, which will typically result in the liberation of additional electrons. This exponential process is quenched when the electric field lacks the needed magnitude to accelerate the electron to result in ionizing impacts or the electrons reach the anode.

- Primary Streamer: Primary streamers start out as an avalanches; however, in the case of the primary streamer, the number of electrons and ions is large enough to self-propagate past the ionization zone from the applied electric field. These streamers cross or nearly cross the gap, creating a channel for secondary streamers without the needed energy to transition to an arc, which is when the channel transitions to a thermal plasma.

- Secondary Streamers: Secondary streamers will be used to describe pulses which, as the name suggests, occur after a primary streamer using the channel made by the primary. These secondary streamers are significantly hampered by the presence of space charge in the gap and thus travel slower than a primary streamer. If a series of secondary pulses occur these are called burst pulses.

- Return Stroke: Return stroke occurs when the primary streamer reaches the cathode and liberates enough electrons to initiate an anode-directed streamer. As Sigmond [89] mentions that these are not typically seen in the case of short gaps (less than 5 cm), which includes the work presented in Chapter 3. Return strokes are possible for long gaps; a long gap configuration is used in Chapter 5.

- Glow: Glow will be used to describe a low-amplitude, high-frequency pulse current or a DC current. Therefore, the oscilloscope does not observe glows since the amplitude is
in the noise threshold of the oscilloscope or the frequency is below the cutoff frequency of the current monitor.

- Corona Discharge: The term corona discharge will be used to describe the general case of gas breakdown due to an applied electric field. This includes avalanches, primarily streamers, secondary streamers, and glows when the reactor is operated with a short gap. When the reactor is run in the long gap configuration (around 11 cm between the needle tips and the ground mesh) corona discharge will include avalanches, primarily streamers, secondary streamers, glows, and return strokes.

This dissertation will focus on primary streamers and secondary streamers for a point-to-plane geometry in an acetylene/argon gas mixture. Additionally, the effectiveness of an argon/oxygen APWIP to treat fresh produce will be examined. This will be done by describing the reactor that was built to facilitate this APWIP research. Next, the sympathetic pulses in the reactor will be considered. Due to the experimental nature of this research, two experimental applications are provided. The first of these applications is the use of argon/oxygen APWIP to treat fresh produce exposed to E.-coli. Plasma-polymerized acetylene is the second application with a focus on deposition rate as well as composition. Finally, observations of the reactor during operation, additional experiments, and future work will be discussed.
Chapter 2

Reactor Design

Reactor 3 was built for the APWIP research described in this dissertation. Reactors 1 and 2 were not used to collect any of the data in this paper, but they provide examples of additional applications and a frame of reference for the development that went into reactor 3. The reactors provided a platform for measurements of current pulses in an application environment as well as a means to accomplish various types of processing. The primary application of reactor 1 and 2 was to accelerate the conversion of light bio-oils into syn-gas via a steam reformer. Reactor 2 improved on electrostatic uniformity and allowed for more precise measurement of current pulses. Reactor 3 was designed to polymerize gas-phase molecules, taking advantage of gravity and leveraging the improvements made from reactor 2 in measuring current pulses. The other major changes seen in reactor 3 are the increased number of needles in the needle array and the transition from DC to AC power.

The goals of keeping the materials simple, cheap, and robust were key design principles for all 3 of the reactors that are described in this work. The transition from DC to AC 60 Hz was made to improve the reactors ability to run for long time periods. Due to the low frequency of the AC excitation no matching network was needed and the needles were connected directly to the power supply for all the reactors. Measurement techniques evolved becoming more comprehensive and insightful with each new reactor.
2.1 Early Reactor Prototypes

Reactor 1, shown in Figure 2.1 was designed as a proof of concept reactor, with the goal being to explore the use of APWIP in the production of syn-gas from the light bio-oils. The reactor was therefore expected to operate at high enough temperatures that the light bio-oils would be broken down into simpler components in the gas phase and then enter a steam reformer where they would be converted to syn-gas. For testing purposes an automotive fuel injector was used to supply the light bio-oil to the reactor. We started with methanol as surrogate for the light bio-oil. The reactor was heated to 100° C before liquid methanol was injected. Once in the reactor the methanol evaporated and the argon carrier gas pushed the vapor through the plasma activation region to break bonds. Although the application goal of the reactor has since changed due to funding, it proved to be instrumental in the design of reactors 2 and 3.

Figure 2.1: Reactor 1 used 9 needles for the production of APWIP. The array was energized from the brass bar at the bottom of the reactor. Liquid methanol and the carrier gas argon entered the reactor on the right side with the exhaust being on the left.

One important consideration of converting the bio-oil to syn-gas was the need to show that it was more efficient than conventional methods. To that end the current was measured and studied, looking at the effect the gas composition had on the current pulses. The
reactor was energized with DC voltage. Having the current pulse data provides an avenue for estimating energy consumption. The investigation of current pulses has remained central to this research. We also observed that liquid precursors at room temperature should be avoided on account of increased complexity and an inability to know the precise number density and consumption rates of the injected precursor species. The needle array also needed to be changed because there was no way to measure corona current to each needle. Future reactor designs addressed these issues.

Reactor 2 addressed difficulties encountered with reactor 1 in terms of its non-symmetric electric field for processing. The research into the current pulses motivated the separation of the electrodes so that the current through each needle could be measured. Uniformity in the electric field in terms of processing volume was addressed by utilizing 5 needles arrayed as a square with one needle comprising each corner and one in the middle. The needles entered from the downstream side of the reactor to excite the gas prior to exiting. A DC excitation was still used and the bio-oil was still being investigated. Figure 2.2 is an image of the reactor showing the needles entering the left side of the reactor which is opposite the gas inlet. The caps at both ends of the reactor are brass as are the needle holders, all of which were grounded. An acrylic cylinder was used to hold the needle inside the brass tubes.

Figure 2.2: Reactor 2 used 5 needles for the production of APWIP. The array enters the reactor from the downstream side of the reactor. The argon carrier gas and liquid methanol was injected on the right side and exited the reactor after passing through the APWIP activation zone.
Use of this reactor was limited, but some key observations were made specifically with respect to the needles and the use of materials. Using the individual needles worked very well and these were also used in reactor 3. The brass feed-through bushing significantly complicated issues, introducing a lot of capacitance between the needle holder and the needle shaft as well as splitting ground potential toward both ends of the reactor. So, in the new reactor, use of metal was limited to reduce sources that could skew the electric field.

2.2 Final Reactor Version

Key things learned from reactors 1 and 2 were incorporated into the design of reactor 3. As mentioned in the introduction, the reactor was rotated to a vertical position with the gas and electrodes entering the reactor from the top. Moving the electrodes to the same side of the reactor as the gas inlets provided a larger treatment area downstream of the plasma. Everything was made from non-conductive material, save the electrodes, to reduce capacitance. The reactor is described below followed by details about various components and electrostatic models of the reactor.
Figure 2.3: Reactor 3 after all the modifications. The HV line from the transformer attaches to the HV distributor. The high flow-rate flowmeter 1 is used for the argon while flowmeter 5 is used to measure the acetylene or oxygen. This was done to ensure that one could not accidentally use both oxygen and acetylene simultaneously. Flowmeters 2, 3, and 4 can be used to control the gas in each plenum separately.
Reactor 3 consists of an HV needle array, gas inlets and outlets, a ground, and platform for processing. The reactor wall is made from 10.16 cm schedule 80 clear PVC. The top and bottom are sealed using a rubber gasket and acrylic discs and secured via nylon threaded rods and wing-nuts. In the bottom disk 4 (1.27 cm diameter) holes act as gas outlets, routing the gas to the exhaust chimney which opens up above the reactor. The exhaust gases exit the chimney and are collected by a suction tube to ensure that the argon as well as any precursor gas is collected and dispersed safely. For commercial applications, this gas could be recycled. There are two additional holes at the bottom of the reactor. One is used to connect the ground plane to the ground and the other is used to hold the processing platform.

The top of the reactor has 12 (0.635 cm diameter) holes which allow the stainless steel (SS) needle holding rods access to the reactor chamber. Three additional holes are used for the gas inlet plenum. There are three plena: an inner, middle, and outer. This allows for rough control of the cross-section of gas flowing through the reactor.

The HV array consists of 12 hybrid needles. A nickel-plated steel needle with an average radius of curvature of $\sim 45\mu\text{m}$ is soldered on to a 0.635 cm x 2.54 cm SS rod with a hole at the end which is tapered to meet the needle. At the other end of the SS rod a 20kV HV wire is soldered to the rod with a heat shrink tube used to support the rod HV wire transition. These hybrid needles are then connected to an HV distributor which keeps all the needles at the same voltage. A nylon collar is used to adjust the needle height in the reactor. The collars and medical grade o-rings are sandwiched between two pieces of 0.635 cm thick acrylic discs to maintain a vertical position in the reactor.

Figure 2.4 shows the ground plane which consists of a ground ring made from SS that is rounded at the top and square on the bottom. The rounded top reduces the E-field strength on the top. The outside edge is concave to allow for a snap or tension connection of a metal fabric or mesh on the top. The ground plane is created by attaching a mesh to the top of the ground ring and then securing the mesh via wire to the ground ring via
the groove on the outside of the ring. The mesh has always been made from SS; while initially the mesh was a fine cloth type mesh, this was changed to a coarse square wire mesh for all the work presented here (see Figure 2.4). Initial runs trying to decontaminate produce showed a marked improvement when the coarse, more open mesh was used. The open mesh was therefore used for all the sympathetic current pulses measurements, produce decontamination, and film deposition work.

Figure 2.4: Ground mesh is made from a SS ring which supports the mesh that has been attached to the top via the groove on the side.

A window on the reactor, situated roughly in the middle, is 0.3 m tall and about 8.89 cm wide. The sides of the window are made of gray PVC while the window itself is made from the same PVC – Schedule 80 leftover from building the reactor – that makes up the reactor wall. There is a minor seam on the inside, but the inside wall generally looks continuous. The window gives easy access to the reaction zone (between the needles and the ground plane) as well as the treatment zone which is below the ground plane. The platform is supported by a nylon threaded rod which allows the stage to be adjusted. The platform itself is made
from a 0.635 cm thick 7.62 cm diameter acrylic disc.

The reactor is supported by two acrylic discs with inside measurements of 11.43 cm diameter and 15.24 cm outside diameter. The discs are secured in place with four gray 2.54 cm PVC diameter rods, which are pressure fitted at the top into a 1.27 cm thick piece acrylic disc and at the bottom to black 1.91 cm thick acrylic disc with a 0.51 m diameter. See the shop drawing in the Appendix, Figure A.4.

2.2.1 Observations

Having described the reactor this section proceeds to highlight some key features as well as things done to improve the basic operations of the reactor described and shown above (see Figure 2.3). Two of these charges were the addition of the exhaust chimney and the use of a bench-top HV transformer, which resulted in consistent operation and measurement of current pulses. The chimney, the first upgrade, was added due to difficulties in purging the reactor of air. It was hypothesized that by adding the chimney the argon would, due to its higher specific gravity, settle into the lowest gravitational potential volume and displace any lighter gas prior to exciting via the chimney exhaust. With the addition of the chimney, the operation became more consistent and the use of volatile gases such as acetylene was facilitated.

Another problem was the presence of partial discharges in the transformer being used to energize the needle array. The partial discharges were observed using the Pearson (see section 2.2.3) current monitors. The polarity of the pulses were always opposite the polarity of the voltage. This observation suggested that the measured discharge was outside the reactor and, by process of elimination, the transformer was isolated as the source for the partial discharges. The partial discharges inside the transformer frustrated the measurement of current pulses in the reactor, one of our primary goals. Avista [1] graciously loaned the department a General Electric JVW-5 100:1 transformer, which was partial discharge-free.
This new transformer has made it possible to measure the current pulses inside the reactor without interference from partial discharges in the transformer.

2.2.2 Power Supply

The reactor along with all the supporting equipment is housed in one of the Bio-Systems Engineering labs. This lab works well because of our collaborations with Bio-Systems Engineering and Food Science units on the WSU campus. This lab is a mixed-use lab supporting a variety of research. The GE JVW-5 100:1 transformer was able to supply the reactor with more power than the previous transformer. As it has an open design such that the electrodes are exposed, a high voltage safety interlock was added. Figure 2.5 shows the safety interlock (See Appendix A.1 for image of safety interlock). Note that a barrier switch was used to prevent the experimenter or others working in the lab from inadvertently getting too close to the HV and getting shocked. A red safety light was used to alert those in the area that the reactor was in operation.

![H.V. Safety Interlock Diagram](image_url)

Figure 2.5: Reactor safety electrical diagram shows the HV safety interlock that was used with the GE HV transformer. The barrier switch was used to ensure that the operator or others in the vicinity did not get too close and risk being shocked.
A series impedance was placed between the VARIAC and the transformer to limit current in the event of a spark or arc, thereby preventing damage to the needles and the reactor. It should be noted that, while it limited the current to prevent damage to the equipment, the current used while operating the APWIP was ~ 2 orders of magnitude less; thus, the series impedance did not interfere or limit plasma processing. The corona discharges measured during processing were < 0.1 A and little or no voltage regulation was observed while processing. As applied voltages were monitored, processing was stopped if voltage regulation (due to reactor corona current flow) exceeded 10%.

Figure 2.6: Reactor circuit/operation diagram highlighting key aspects of reactor 3 including the gas inlet plena.
2.2.3 Measurement Equipment

Systematic and automated current and voltage measurements were needed for the statistical approach used to investigate the current discharges in neighboring needles. The Instrument Control Toolbox for MATLAB [4] was used to automate the collection of the data from the oscilloscope. Since gas flow-rate and gas composition are key factors in how much voltage can be and was used, A GUI was written to collect the current pulses which integrated the gas flow and current pulse settings used (additional information in Appendix D).

Oscilloscopes

A Tektronix 3014B 100 MHz 4 channel oscilloscope [10] was used for the collection of the current on two needles, the transformer current, and the voltage. National Instruments VISA drivers [5] are used by MATLAB ‘s Instrument Control Toolbox to control the oscilloscope, the functions of which included applying settings, triggering the scope, and saving the data to the laptop for post processing. All communication between the laptop and the oscilloscope was done on a wired 100 MB/s LAN.

Half-wave voltage and transformer currents were occasionally collected for a total duration of about 8 ms. To do this a LeCroy 9350AL [3] was used, collecting 1,000,000 points with a resolution of 10 ns. Again, National Instruments VISA drivers were used by MATLAB ‘s Instrument Control Toolbox to collect the signals via a Prologix GPIB-ETHERNET controller [7]. The scope needed to be set by the user prior to collecting the data but, once started, the oscilloscope triggering was done by MATLAB. This scope was not used as much since only one current and the voltage could be monitored; the collection process also took much longer, restricting its ability to collect a statistically significant number of samples. The time between collections was slightly over a minute while the Tektronix scope could collect 15+ samples in a minute.
Three Pearson 6585 current monitors [6] were used to monitor the transformer current as well as the current through two of the needles. These current monitors were ideal because of their high sensitivity (1:1 ratio between the current being monitored and the voltage observed on the oscilloscope) and a large inside diameter (ID 5.08 cm) which prevented partial discharges in the insulation that protected the current monitor from the high voltage conductor. Initially a Pearson 2877 was used which had the same sensitivity, but with an inside diameter of 0.635 cm. When the voltage was above 6.25 kV RMS partial discharges were observed between the Pearson and the HV line. The Pearson 6585 has a usable rise time of 1.5 ns which is under the resolution used to collect the sympathetic current pulse data. It should be mentioned that the low frequency cut-off of the Pearson is 400 Hz which is well above the 60 Hz signal used to excite the plasma. Due to this low frequency cut-off, the Pearson observes only current pulses associated with a corona discharged, but not 60 Hz capacitive current. Figure 2.7 shows the transformer current pulse measured by the Pearson when they were placed on the same line such that each measured the same current pulse (additional images of the Pearsons can be found in Appendix A.2). The max standard deviation between the three Pearsons is 0.5 mA which is in the noise level of the Tektronix oscilloscope since it is typically operated at 5 mV/div.
Figure 2.7: Pearson example current plot showing the same current pulse measured by all three of the Pearson 6585 current monitors.

### 2.2.4 Needle Array and Grounded Mesh

The needle array, as mentioned earlier, is composed of 12 needles with four needles making up the inner array and eight in the outer array. Gas inlet plena were placed on the outside of the outer array, between the needle arrays, with a single inlet hole in the center of the needle arrays. Figure 2.8 illustrates this. It should be noted that, although gas plena can be operated independently, for this research the plena are connected outside the reactor to give uniform gas pressure and composition. A T-joint is used to connect the outer and middle plena to the feed hose, creating a void which is replicated on the opposite side. This is shown in Figure 2.6 where 14 (0.158 cm diameter) holes are used in the outer plenum, seven on one side and seven on the other. The middle plenum only has four (0.159 cm diameter)
holes, two on either side, while the central inlet is a single 0.32 cm diameter hole. Assuming similar pressures on each plenum feed hose the number of holes were chosen to create an outer sheath of gas outside the outer needle array. The middle plenum was based on the same principle, but this sheath encompasses both needle arrays while the center inlet should provide a column of gas just inside the inner needle array. This assumes minimal mixing, which would be ideal for some applications, but has not been leveraged for this work. This is a feature of the reactor that could be leveraged if needed at a later date.

Figure 2.8: Needle array with the 3 gas inlet plena highlighted.

Prior to collecting the sympathetic current pulse data each needle radius of curvature was measured. Figure 2.9 shows the measured radius of curvature sorted by the gap between the needle and needle 1.
Figure 2.9: Needle radius of curvature sorted by their distance from needle 1 (N1), making needle 14 the farthest from needle 1.

Needle 1 has been used as the primary needle for all measurements. Figure 2.10 shows the radius of curvature measurement of the needle that was made prior to collecting the sympathetic current pulse data. The needles were cleaned with isopropyl alcohol and Scotch Brite between current pulse collections to ensure consistent operation. It is well known that after extended use erosion can occur. This phenomenon has been studied by others [36, 35] and was not studied here, but was monitored for the sake of consistent operation.
2.2.5 Electrostatics

The electrostatic modeling can be done two ways: analytically and numerically. Figure 2.11 illustrates the geometry for the electric field from a needle tip on axis assuming that the tip radius is much less than the point to plane distance being given by equation 2.1.

\[ E(x) = \frac{2V}{\ln \left( \frac{4x}{r} \right)} \frac{S}{S(2x + r) - x^2} \]  

(2.1)

as derived by Eyring et al. [31] and simplified by Lama and Gallo [54]. Where V is the voltage, r is the needle radius of curvature, S is the distance from the needle tip to the plane, and l is the distance from the needle tip. The on-axis electric field equation will be used later when estimating the ion transport.
Figure 2.11: Electric field point-to-plane diagram where $r$ is the radius of curvature for the needle tip, $S$ is the distance between the needle tip and the ground plane, and $x$ is the location of interest.

Up to this point the point-to-plane geometry has been asserted as a reasonable approximation needed for modeling the charge in the reaction region. The following electrostatic simulations justify the point-to-plane geometry for the reactor with the caveat that, as the distance from the needle increases, the point-to-plane electric field from the other needles slows the decrease in the electric field. COMSOL 4.2a is a finite element code multiphysics modeling program [2] and was used for modeling the electrostatic fields in the reactor. A tetrahedral mesh is used. The surfaces that make up the needles and the ground mesh are specified as perfect electrical conductors with voltages of 9 kV and 0 V, respectively. Note that the area enclosed by the needles is a void and not meshed, since this area is of no interest because Joule heating is not an issue. Voltage boundary conditions are used with no free charge in the gap. Only the needles and ground screen boundaries are specified. The reactor volume is specified as air. The gap between the needles and the grounded mesh is 4.6 cm. The needles are modeled starting with the tip using a semicircle with a radius
of 50 \mu m. The needle is a 5.5 cm long conical frustum where it is soldered to the tapered 0.635 cm SS rod. The mesh wires have a radius of 0.035 cm with a spacing of 0.6 cm which is consistent with the SS mesh in the reactor.

Figure 2.12 shows the model of the needles and the ground plane for reactor 3. The geometry was created using CUBIT [11] and then imported into COMSOL. Figure 2.13 is a close view of the ground plane with the tetrahedral mesh resulting in \( \sim 0.5 \) million elements. The total meshed area required \( \sim 6 \) million elements. All the COMSOL modeling was done on a Windows 7 machine with 8 GB of ram using an Intel i7 with 4 physical cores, each with hyper-threading.

A 2D slice of the 3D solution can be seen in Figure 2.14 where the color gradient is based on the electric potential and the electric field is shown with white arrows whose length is proportional to the electric field strength. Figures 2.15 - 2.18 are contour plots from the center of the reactor with increasing detail of the needle area. The smooth contours seen in Figure 2.15 remain smooth until Figure 2.18 where mesh effects are observed. The COMSOL solution is compared to the analytic solution in Figure 2.19; where the COMSOL electric field magnitude values for the plot are represented by line (a) for the on-axis solution which is compared to the analytic solution given by equation 2.1 and line (b) which is in the center of the reactor in Figure 2.15.
Figure 2.12: Needle array and ground mesh used in COMSOL for the electrostatic simulations of the reactor.
Figure 2.13: Ground mesh showing the meshed geometry. The ground mesh required \( \sim 0.5 \) million elements.
Figure 2.14: Electric potential and electric field shown with the color gradient indicating the electric potential and the electric field vectors being the white arrows. This is a slice in the middle of the reactor in the yz plane. The white outlines of the outer needles are the only needles in the plane.
Figure 2.15: 2D electric potential contours are given for the slice in the middle of the reactor in the yz plane. The uniformity of the electric potential confirms the point-to-plane geometry approximation as appropriate. Lines (a) and (b) indicate the lines used to make the line plots shown on page 40.
Figure 2.16: 2D electric potential contours needle tip to ground which is 4.6 cm below the needle tip located at zero.
Figure 2.17: 2D electric potential contours for needle shaft
Figure 2.18: 2D electric potential contours at needle tip show some irregularities due to mesh imprinting, but these are minor and not significant.
Figure 2.19: Electric field comparison shows the analytic solution to the on-axis electric field from the needle tip compared to the on-axis electric field calculated by COMSOL. The COMSOL Center Line shows the electric field magnitude from above the needle to the ground plane taken at the center of the reactor.

The non-uniform electric field around the needle tips quickly gives way to a more uniform field through the gap. Near the ground mesh the electric field gets stronger and deforms slightly due to the SS wires that make up the ground mesh (see Figure 2.19). Figure 2.19 combines the analytic on-axis electric field solution given by equation 2.1 with the on-axis COMSOL solution; as well as the COMSOL electric field magnitude in the center of the reactor (where there are no needles which is why these values start above the needle tip). It should be noted that an effort was made to allow direct comparison of the analytic solution and the COMSOL solution in relation to the COMSOL geometry. As such the needle is at $x=0$ and the ground plane is at $x=-4.6$ cm. The analytic point-to-plane equation assumes $x$ starts near the needle tip and moves on-axis toward the ground plane with increasing
positive values of $x$, which is where the effort was made to allow for a direct comparison between COMSOL and the analytic on-axis solution. The analytic electric field is calculated for values closer to the needle tip than the data points used from the COMSOL solution which is why the analytic electric field is stronger. At about 1 cm below the needle the analytic solution and the COMSOL solution diverge slightly due to the needle array effect which results in the non-uniform electric field around the needle tips transitioning to a more uniform electric field towards the ground plane.

### 2.2.6 Equivalent Circuit

An equivalent circuit has been used to model the electrical response of the reactor. The model will be used in Chapter 3, but the circuit will be introduced here since it models the reactor. Starting on the left and moving to the right, the transformer has been modeled using voltage source ($V_T$) in series with an inductor ($L_T$) which represents the transformer inductance. These have been connected in series with a resistor ($R_T$) for the in series impedance between the variac and the transformer. Next a bushing capacitance ($C_B$) is placed in parallel followed by half of the loop inductance ($L_L$) which comes from the transformer-reactor loop. The voltage divider ($V_D$) is placed in parallel for the measurement of the voltage on the needles. This is followed by the rest of the loop inductance ($L_L$). The thick black lines represent the HV distributor and needles. $i_T(t)$ monitors the current from the transformer while $i_1(t)$ and $i_2(t)$ monitor the current through needles 1 and needle 2. $R_G$ is the resistance of the gas to the current. $C_N$ and $C_S$ are the needle-ground and capped needle-ground capacitance respectively. The $R_N$ and $R_S$ represent the un-capped and capped needle resistances respectively. The capped needles have a higher resistance because the conductive carbon-silicon that is used to hold the SS sphere onto the needle, making it a capped needle, is not a perfect conductor having a resistance in the 100’s of Ohms.
When the reactor is being used for processing none of the needles are capped; only when the sympathetic current pulses are being measured are all but two of the needles capped. It is important to note the locations of the Pearson current monitors. Only 2 of the needles are monitored at any time along with the current from the transformer. Therefore, if a discharge occurs on one of the monitored needles, the current observed by the transformer is going to be influenced by the charge in the unmonitored needles of the needle array. While the capped needles are nearly shut off, they are not entirely turned off and do participate occasionally.

### 2.3 Conclusion

This chapter began by discussing the initial reactors that were built which led to the reactor used for the research presented in this work. The next chapter (Chapter 3) will discuss how gas composition and purity have a major effect on the corona discharges observed. Measurements based on specific pure gas compositions require a vacuum system. This reactor
was not designed for that purpose (pure gas studies), but rather to experiment with APWIP for industrial applications. The other major function of the reactor is the measurement of corona discharges. To that end, the only conductive material used was for electrical components; all other components are made from PVC, nylon, or acrylic. The vertical orientation was used to improve electrical symmetry with regards to the earth’s ground and gravity works with the reactor to assist in taking the excited gas to the afterglow region.
Chapter 3

Sympathetic Corona Current Pulses

The sympathetic corona pulse study has been done to provide insight into the reactor operation, specifically needle to needle interaction. This work can be used to optimize the reactor for a specific processing application. While others have studied \cite{12, 20, 45, 15, 51} corona pulses from multiple needles, those works have been limited to air, flue gas, or pure argon. The corona pulses used in these works are in an acetylene-argon gas mixture. Acetylene gas was used since it has been shown to polymerize and form films \cite{60, 85}; understanding the corona pulses in this gas mixture, therefore, supports work in film deposition (Chapter 5).

Argon and acetylene have long been a gas mixture of interest for use in gaseous detectors since this combination has one of the largest known Penning effects \cite{106, 47, 41} for low concentrations of acetylene. The Penning effect lowers the breakdown voltage needed to excite a metastable state in argon which is near the ionization potential of acetylene (11.42 eV). The lowest and third-lowest excited states in argon are metastable at 11.55 eV and 11.72 eV. Both are just above the acetylene ionization potential with lifetimes in the range of seconds \cite{106}. This long lifetime increases the range in which a metastable can impact the acetylene to result in an ionizing event. The Penning effect is most pronounced around 0.1% C$_2$H$_2$ and, if the concentration is increased, the gas quenches secondary avalanches \cite{41, 106}. For the data presented in this chapter an argon flow rate of 1242 sccm and an acetylene flow rate of 61 sccm is used, so acetylene makes up 4.6% of the whole gas mixture or the acetylene can be viewed as 4.9% (approximately 5%) of the argon flow rate.
3.1 Gas Dynamics in a Non-Uniform Electric Field

Having introduced some of the key complexities and the reasons for the gas mixture being used, it is appropriate to provide some basic background about the gas dynamics in the reactor. In chapter 1 a broad overview of the basics of corona discharges is provided. Here a more detailed discussion of basic breakdown theory in light of the specific setup used in this research. The values calculated are all for the case of atmospheric pressure and a temperature of 20°C which are in line with reactor operation. The mean free path is a key property, as it is a measure of how far a particle can travel on average before impacting another particle. Calculation of the mean free path is given by Kuffel and Zaengl [53] for a gas mixture (equation 3.1). Equation 3.1 can be used to calculate the mean free path ($\lambda_1$) of a specific gas phase species (with radius $r_1$ and mass $m_1$) in a gas mixture. It treats each gas species as hard spheres where $N_i$ is the number density (particles/m³) and $r_i$ is the radius of a specific species.

$$\lambda_1 = \frac{1}{\pi \sum_{i=1}^{n} N_i (r_1 + r_i)^2 \sqrt{1 + \frac{m_1}{m_i}}} \tag{3.1}$$

The number density of argon is calculated to be $N_{\text{argon}} = 2.39 \times 10^{25}$ while the acetylene (5% of the argon flow rate) has a number density of $N_{\text{acetylene}} = 1.19 \times 10^{24}$. Straightforward as equation 3.1 may seem, it relies on knowing the diameters of the atoms or molecules. As Sahin et al. [106] states, knowing the diameters is not straightforward with cross sections depending on energy, temperature and, of course, the gas mixture, to name the primary dependencies. Nakamura [69] has measured the electron drift velocities as a function of the applied electric field for an argon and 5% acetylene gas mixture. Using Nakamura’s [69] electron drift velocity data the electron mean path and the mobility can be extracted as well as an estimate for the time needed for the electron to cross the gap. The following
equation 3.2 was given by Holstein [43] to relate the mobility and the average cross-section.

\[ K = \frac{3}{8} \frac{\sqrt{\pi} e}{\sqrt{M k T N [Q_M]_{av}}} \]  

(3.2)

where \( k \) is the universal Boltzmann’s constant with a value of \( 1.3804 \times 10^{-23} \) Joules °K, \( T \) is the temperature which is room temperature, and \( M \) is the weighted mean particle mass. \([Q_M]_{av}\) is the average cross-section for momentum transfer, which can be used to determine the mean free path. Figure 3.1 shows the drift velocities for electrons and for argon according to Nakamura and Phelps respectively [69, 78]. While the values cover a wide range of electric field strengths, neither completely spans the electric field range seen in the reactor.

![Drift Velocities](image)

Figure 3.1: Drift velocity data from Nakamura and Phelps [69, 78] has been used to model the particles in the gas.

Mobility values outside the data shown in Figure 3.1 were extrapolated assuming that mobility does not change for increasing or decreasing values of the electric field. Figure 3.2 shows the result with the extrapolated values being the horizontal portions of the curve. The mobility is used to estimate the crossing time for the argon ions and the electrons. While the
extrapolated values may not be exact, they are similar in magnitude to the mobility values found in Kuffel and Zaengl [53], so this works as a good estimate.

Figure 3.2: Gas mobility was calculated using Equation 3.2 \( \left( K = \frac{\frac{3}{2} \sqrt{\frac{e}{M \kappa T N [Q_M]_{av}}}}{m} \right) \) from Nakamura and Phelps [69, 78] drift velocity data.

When a particle becomes charged it will experience a force due to an applied electric field. The mobility is a measure of how the particle moves in such a field. Since the mobility (K) relates the particle velocity in a gas to the applied electric field, it is straightforward to extract the mobility by dividing the velocity by the electric field. Having the mobility for both the argon ions and the electrons, the gap crossing time can be estimated. The electron crossing time is estimated to be 1.2 \( \mu \text{s} \), while an argon ion takes milliseconds to cross the needle to ground gap. It should be noted that the electron mobility, as mentioned earlier, uses Nakamura’s [69] data which is for a gas mixture that is similar to what was used in the reactor, while the argon ion crossing time is for a pure or nearly pure argon gas. Figure 3.3 shows the estimated crossing time, when the positive argon ion is released from near the needle tip for a time dependent RMS voltage 6.5 kV, phase shifted such that at \( t = 0 \) the needle voltage is 8 kV. This release voltage was chosen by looking at typical values at which
corona pulses can start to be seen in the reactor. As can be seen in Figure 3.3, when the
time dependent voltage source is used the argon ion does not make it to the ground screen.
This estimate also includes the gas drift velocity. This illustrates how long ion clean out
(the time required for an ion to exit the excitation zone) can take and that ions may pass
through the plasma more than once before exiting the plasma zone and moving into the
afterglow zone. These estimates provide a great deal of insight into the gas dynamics seen in
the reactor and will be leveraged when interpreting results. However, before presenting the
results, a more detailed analysis of the corona discharge is needed to facilitate interpretation
of the experimental results.

![Ar⁺ Ion Crossing Times](image)

Figure 3.3: Ar⁺ crossing time for the positive argon ions may stay in the gap for multiple
periods excitation.
3.2 Electrical Breakdown in Gas

As was mentioned in the introduction (Chapter 1), the theory behind gas breakdown in a non-uniform electric field has developed over time. The work of Townsend et al. [96, 95] and Loeb [57, 58] is fundamental, but with the technological improvements of the last 30 years this fundamental work has been refined. This work will be using the more recent work of Sigmond [89], Morrow et al. [66, 68, 67], Abdel-Salam et al. [15, 17, 13, 16, 19, 18, 14], and Merbahi et al. [61, 62] to describe the gas breakdown that is observed in the acetylene-argon gas mixture.

3.2.1 Corona Discharges

A corona discharge starts with electron avalanches and electron avalanches occur when a free electron gains enough energy (being accelerated by the applied electric field) to liberate additional electrons on impact with molecules or atoms. Morrow and Lowke [68] modeled a positive streamer in air and estimated a 7 nA Gaussian type pulse with an avalanche duration of less than 0.25 ns. While the exact time and amplitudes are not important, their relative values are important since this avalanche transitioned into a 7 mA primary streamer with a duration in the 10 $\mu$s range. The other key point shown by Morrow and Lowke [68] was the dependence of photoionization from the avalanche to initiate the primary streamer. Without photoionization no primary streamer developed. So, for an avalanche to transition from an avalanche to a streamer, photoionization needs to occur to liberate enough electrons for the development of the streamer.

According to Abdel-Salam and Allen [14], onset streamers occur when a series of avalanches occur one after the other, building each on the previous. This continues until the positive charges from the avalanches accumulate around the tip and effectively terminate any more avalanches. This process will be repeated after the positive ions have cleared out; the onset
streamers do not form a head and will not travel past the ionization boundary. Merbahi et al. [61] illustrates an onset streamer with an amplitude of about 1 mA and does not have sharp peak, but rolls over.

Increasing the voltage causes a glow to develop which is pulseless or, if pulses occur, the magnitude of the pulses are very small such that they are in the noise for our equipment [14, 61, 52, 34]. Sigmond [89] starts his description of the primary streamers with a positive glow around the needle and Morrow [66] shows how a primary streamer can transition to a glow for an applied DC voltage. Using the AC excitation the voltage transitions through glows, but due to the low amplitude they are not measured with the oscilloscope.

As the voltage increases still further, primary streamers (PS) are observed. The primary streamer starts with an avalanche but, due to the increased electric field strength, the avalanche will transition to a primary streamer due to the space charge which is enough to have an electric field strength on the order of the applied electric field. This is known as Meek’s Criterion [59]. This concentration of charge is called the head and self-propagates across the gap, faster than the electron gas speed due to photoionization. This occurs because the streamer head travels as a wave and leaves behind a channel of positive ions, negative ions, electrons, radicals, and meta-stables [89, 100, 101]. This is an important transition, since all the previous discharges could be described using particle motion, for the positive streamer photoionization is invoked to account for the speed \((2.0 \times 10^5 \text{ m/s} - 2.0 \times 10^6 \text{ m/s})\) at which the streamers cross cm gaps [89, 66]. Figure 3.4 illustrates a streamer, where the different aspects of the streamers have been deduced utilizing work done by Merbahi et al. [61]. Merbahi et al. [61] measured the current pulse synchronized with the streak photos which allow the identification of the streamer propagation, residual channel clean-out, and the development of a secondary streamer. One key thing noted by Merbahi et al. [61] is that the secondary streamer is only observable with sufficient voltage. Otherwise, only the primary streamer is observed. Another interesting observation by Merbahi et al. [61] is
that for voltages where only a primary streamer is observed the streamer propagation can be much slower and is very pronounced. This can be seen in Figure 3.4 and Figure 3.5 which shows a streamer propagated across the gap, dramatically peaking when it reaches the cathode.

Secondary steamers occur in the channel created by the primary streamer, travel slower than the primary steamer, and stop where the primary steamer stops [89, 61, 67]. Figure 3.5 illustrates a primary streamer followed by a secondary streamer which is the second peak after the steamer has reached the cathode. Secondary steamers are more luminous and more diffuse, where the primary is filamentary and not very luminous [89, 61, 30]. Having laid a foundation for the corona discharges and showing examples of the streamers, the data collection process is described next.

Figure 3.4: Primary streamer example current starts out at 4 mA and accelerates toward the ground screen as the steamer approaches the ground screen. The steamer connects with the cathode at \(t = 0\).
Figure 3.5: Primary and secondary streamer example current where the primary streamer crosses quickly at impacts the cathode at $t = 0$. Channel begins to clean out when a secondary streamer occurs less then 1 µs after the primary impacts the cathode.

3.3 Corona Pulse Acquisition

For this study 500 corona pulses were acquired for each needle pair. A Tektronix [10] 3014B 100MHz 4 channel oscilloscope was used to capture the data. Channel 1 of the oscilloscope records current supplied to the needle array from the transformer, channel 2 the voltage, and channels 3 and 4 the current through needle 1 and the current through the other needle, respectively. Matlab [4] was used to automate the data collection process via the instrument control toolbox and National Instruments VISA drivers [5]. Before collecting samples for a needle pair, the ground screen and the needles are cleaned with isopropyl alcohol. Once cleaned the reactor is purged for 10 minutes prior to applying the high voltage to the needles. Argon is used as the carrier gas with a flow rate of 1242.32 sccm and a flow rate of 61.8 sccm is used for the acetylene ($\text{C}_2\text{H}_2$), equal to 5% of the argon flow rate. An electrode gap of 4.6cm was used, giving a residence time in the gap of 15.45 seconds (or gas velocity of
0.298 cm/s). Corona discharges from the transformer were used to trigger data collection utilizing a 10 mA rising edge trigger. All the data was collected relative to needle 1, so all gaps are between needle 1 and the other needle that is being monitored. After acquiring the 500 samples, the next step was to post process the collected data and sort the corona pulses into different categories.

Statistically significant sample sizes were chosen by collecting two sets of 1000 corona pulses and then looking at the percentage of dual corona discharges. The variation between percentage of dual corona discharges for the first 200 and all 1000 corona pulses varied by less than 5% suggesting that a statistical sample had already been met by 200 corona pulses. Deposition on the needles was observed and will be discussed in Chapter 5. Cleaning the needles and the ground plane between data collections insured consistent and repeatable reactor operation.
Figure 3.6: Corona pulse collection number check shows how as the sample size increased the number of dual pulses does not change significantly.

### 3.4 Post Processing of the Current Pulse Data

Having acquired the data the next step was to determine the type of pulses recorded. The focus was on the possibility of sympathetic corona pulses. To that end the pulses were sorted into 4 categories,

- **Category 1**: a corona discharge observed only through needle 1 and correlated with the discharge current observed at the transformer (see Figure 3.10),

- **Category 2**: a corona discharge observed only through the other needle and correlated
• Category 3: corona discharge current pulses observed through both needles with the sum being correlated with the discharge current observed at the transformer (see Figure 3.12),

• Category 4: corona discharge current pulses observed through one or both needles, but not correlated with the discharge current observed at the transformer (see Figure 3.13).

In order to make the sorting more efficient a hybrid case was made that treated the sum of the current measured by needle 1 and the other needle as its own pseudo needle current. Each signal was DC corrected using the first 50 points of the signal; the signal charge and the maximum currents for each pulse were calculated. The transformer current was used as the standard for evaluating the type of pulse.

Before sorting the pulses the correlation between each needle and the transformer current was calculated. Using the correlations, charge, and maximum current, an if statement was used to sort the pulses. Figure 3.7 illustrates the process where the top row is the most strict and the second row is slightly less strict. A key point of the algorithm is that, if the charge measured thru the needles is more than the charge measured through the transformer, the pulse is considered well defined ($\Delta Q$). Referring back to the equivalent circuit 2.20, the capped needles store a small amount of charge so that when the corona discharge occurs part of the current through the needles is supplied by the capacitance from the capped needles. In the if statement Figure 3.7, C1 stands for the correlation between needle 1 and the transformer current, C2 refers to the the correlation between the other needle and the transformer, and C3 stands for the correlation between the current sum of needles and the transformer.
As can be seen from the if statement Figure 3.7, after the algorithm has tried each option twice the corona pulses are considered uncorrelated with the transformer corona pulse. This occurs when one of the capped needles gets involved, which occurs about 1/5th of the time. Figure 3.8 shows the corona pulses measured for needle combination sorted by needle gap. Note that the first two columns have a gap of 1.87 cm. This is the smallest gap which occurs between needle 1 and needle 20. This combination was the first to be measured and, once all the other needle combinations relative to needle 1 were measured, the needle 1 and needle 20 combination was measured again. The dual pulse percentage only varied by 3% between the two collections, which occurred a day apart. The percentages were calculated based on
the number of corona pulses which were correlated with the transformer corona pulse. Since
Figure 3.8 shows the number and not the percentage of pulses one can see the number of
pulses that were considered uncorrelated at the top. This number averaged less than 150
pulses.

Figure 3.8: Corona pulse collection counts shows all the pulses collected sorted into the 4
different categories used.
Figure 3.9: Needle correlation vs. gap with illustrates the decrease in the needle interaction after 4 cm. For a gap less than 3 cm the interaction is significant and increases as the gap decreases.

Plotting the percentage of dual pulses (Figure 3.9) out of all the correlated pulses acquired shows needle interaction for gaps less than 3 cm and little interaction for gaps larger than 4 cm. Figures 3.10 thru 3.13 illustrate the different current pulse categories used. Figures 3.10 thru 3.13 shows the current measured by each Pearson starting with the current measured by the Pearson that monitors the HV lead that connects the transformer to the needle array. Next the current from needle 1 is plotted followed by the current measured on the other needle which is, in this case, needle 20 and 1.87 cm away from needle 1. Figures 3.10 and 3.11 demonstrated the pulses that were categorized as being correlated with the transformer, but not with each other. Figure 3.12 exemplified a pulse correlated with the transformer current as well as the other needle. An un-correlated pulse (Figure 3.12) illustrates a situation in which corona discharges from more than the monitored needles occur.
Figure 3.10: Correlated needle 1 corona pulse example (Category 1). Here the current through needle 1 (red) is similar to the transformer current (black). Since the max current through the other needle (blue) is less than 2 mA makes the pulse a category 1 pulse. Notice the slight dip on the other needle, this is consistent with stored charge being delivered to needle 1 via the other needle.
Figure 3.11: Correlated other needle corona pulse example (Category 2). The current pulse through the other needle (blue) suggests a primary then secondary streamers. The sharper edge of the transfer current is illustrates the smearing out of charge that occurs in the needle array.
Figure 3.12: Correlated dual needle corona pulse example (Category 3). Only a primary steamer is seen on needle 1, while the other needle records a primary and secondary steamer.
Figure 3.13: Uncorrelated needle corona pulse example (Category 4). This illustrates current being supplied by the transformer, but not observed going through needle 1 or the other needle.

3.5 Discussion

The uncorrelated category, category 4, is interesting since it catches pulses which don’t occur from the 2 monitored needles. There is a whole array of needles in the reactor and all needles are maintained at the same voltage, but all save two are capped. As mentioned earlier, occasionally the transformer supplies more current to the needle array than what is seen by the sum of needle 1 and the other needle. While there is a complex variety of species in the excitation region, the irregularity of the uncorrelated pulses suggests a random seed. It is hypothesize that metastable argon may play role in enhancing steamers from the capped
needles. Ultraviolet light could also induce streamers from the capped needles, but the random nature of the discharges eliminates this as a viable hypotheses since the discharges would in that case be more regular.

If all the needles in the needle array are capped no discharges are observed but, as Figure 3.13 illustrates, uncorrelated pulses are roughly constant between needle combinations. Photos for gas mixtures of 10% and 0.95% acetylene capture light from the capped needles (see Figure 6.1.1). The lack of discharges when all the needles are capped, contrasted with occasional discharges when two needles are uncapped, suggests that one of the plasma products is responsible for the uncorrelated discharge.

UV photons play an important role in synchronizing the needles. All the needles in the needle array except needle 1 and needle 20 (1.87 cm gap) were capped. Then a filter were placed such that a photon from the tip area of needle 1 would have to pass through the filter to reach the tip area of needle 20. The reactor was run with same parameters used for synchronization study described in Section 3.3. Three filters were used, an opaque filter, a glass filter which only allows wavelengths greater than 300 nm [9], and a quartz slide which has a strong transmittance range of 270 nm to 2500 nm [8]. Figure 3.14 shows the result of this study which illustrates the effect of UV photons between 210 nm and 310 nm. The Quartz filter has a transmittance of 50% for 210 nm which increases to 90% for wavelengths greater than 270 nm and less than 2500 nm. The percentage of dual pulses out of 500 samples drops from 75% to about 50% when the quartz filter is used. This points to a strong dependence on UV photons with a wavelength less than 210 nm. When glass is used as a filter the number of dual pulses drops to about 41% suggesting that the UV photons between 210 nm and 310 nm effect needle synchronization by 10%. The opaque filter does not differ significantly from the glass in terms of the probability of dual pulses suggesting that the visible spectrum does not contribute to needle synchronization.
3.6 Conclusion

In this chapter the focus has been on the measured corona pulses and how they correlated with corona pulses measured by others. 500 corona pulses were collected for each needle pair and analyzed to determine if a corona discharge from one needle could initiate a corona discharge in an adjacent needle. The data shown here suggests that for gaps less than 3 cm the interaction between needles is greater than 40% and that this interaction increases as the gap decreases. This dependence on gap spacing should be considered when designing a reactor and can be used to facilitate a more random processing volume with less demand on the power supply or a more synchronized processing volume with a higher demand on the power supply by adjusting the gap between needles.

This study in synchronization of the needle in the needle array also elucidated steamer behavior in the reactor. The information gleaned while studying needle synchronization was
used in reactor applications which illustrate the reactors potential, but not its limits.
Chapter 4

Decontamination of Fresh Produce

4.1 Introduction

In the introduction (Chapter 1) some basic information was given explaining why the decontamination of produce is important and why new decontamination methods are of such interest. Having already established the general need for more focused research using APWIP to enhance food security, First a brief overview of the current status of food decontamination is given. Next popular plasma sources used today are examined. Finally, the collaborative produce decontamination work that was done with Daniela Bermudez-Aguirre [23] and Abigail Moody [64] will be discussed. The collaborative work with Bermudez-Aguirre focused on using argon alone to decontaminate tomatoes, lettuce, and carrots inoculated with E. coli ATCC 11775. The collaborative work with Moody has focused on the use of argon as the carrier gas and oxygen as a precursor gas for the decontamination of carrots inoculated with E. coli ATCC 11775. It should be kept in mind that this research is still very preliminary. No one is commercially utilizing plasma decontamination, though the results are encouraging and, with optimization, hold the potential to revolutionize food processing. The need to pursue this research is clear, as no other viable solution has yet to be found to replace the chemical washes currently used that may need to be replaced as bacteria develop resistances to such solutions. The produce decontamination work presented in this chapter utilizes tomatoes, lettuce, and carrots. These three different types of produce comprise a wide variety of typical characteristics in terms of texture and treatment surface area. This variety in texture and surface area is used to explore the potential of APWIP in the decontamination
of fresh produce that has been inoculated with *E. coli* ATCC 11775.

Bermudez-Aguirre and Moody prepared the samples for treatment and post analysis to determine the effectiveness of the APWIP treatment. The focus is on the operation of the reactor, optimizing the gas flow, voltage, and electrode gap. The collaboration between Bermudez-Aguirre and Moody was highly integrated. After running the samples and getting the results back we would discuss the results and use our varied backgrounds to improve the next run. Following preliminary sample runs, experimental design methods were used to run the series of experiments that are shown in this chapter.

### 4.1.1 Plasma vs. Other Disinfection Techniques

APWIP is very attractive as a method for food decontamination due to its minimal impact on the produce and lack of harsh and harmful chemicals involved in currently used processes. Today produce is typically disinfected with a chemical wash such as a chlorine solution. The issues with this method are that the chemicals are not completely removed after disinfection and that microbes are becoming resistant to these solutions [70, 86], prompting the need for new disinfection methods. Some of the popular new methods being explored are ozone, ultraviolet light, and irradiation in addition to the APWIP [84]. Each method has several hurdles that must be overcome before widespread implementation is possible. Two major factors include the public’s demand for minimally processed food with minimal change in the color and texture and assessment of how safe APWIP treated food is for consumption. Commercial concerns include extending shelf life and the effectiveness of sterilization.

The treatment method with the least impact on color and texture while maximizing shelf life and sterilizing the produce will have the most potential for commercial use. Irradiation is very effective at reducing bacteria and extending shelf life, however it has a major effect on the texture of the produce [84]. Ultraviolet exposure works very well by damaging the DNA of the microbes, but high doses can cause significant damage to the produce [84]. Ozone is
highly reactive and works well as an antimicrobial; however the corrosive nature of ozone is a limiting factor [84, 50]. All these methods have potential. There has, however, been limited testing and the testing that has been done suggested that each method has a strength for specific types of produce and for specific types of treatment. It would be ideal to have a technology that could be more general.

Each of the methods mentioned above show potential for effectively treating fresh produce. APWIP, with the exception of irradiation, can produce ozone, UV, as well as a variety of other reactive species for disinfection of produce in one step. This potential of APWIP is one of the reasons for the increased interest in APWIP with the possibility of meeting the public’s demand for minimal processing of produce. The gases used in the APWIP are a major factor in the production of reactive species as well as UV and ozone. Argon is used as the carrier gas due to its inert nature which should limit health related concerns. We do not produce any ozone using only argon, although we do expect to produce some UV near the needles. Starting with just argon will allow for a baseline which we expect to improve on with the addition of other precursor gases which can be selected to improve decontamination of the produce. To that end, some preliminary oxygen precursor gas studies have also been done to estimate the effectiveness of ozone and other oxygen species with the argon plasma. A brief overview of additional plasma generation methods are discussed below, followed by the current results using the argon and oxygen/argon corona plasma.

4.1.2 Popular Plasmas Used in Disinfection Research

A more comprehensive list of plasma sources was given in the introduction (Chapter 1). This section will focus on dielectric barrier discharge (DBD) and plasma jet. The DBD and the plasma jet are the two most common APWIP used in decontamination research [63, 86, 93, 77, 76, 104, 80, 33, 38, 65]. The reason for their popularity is ease of operation since the dielectric barrier inhibits the transition of weakly ionized plasma to a thermal plasma.
**DBD**

Dielectric Barrier Discharge (DBD) is a very popular method for generating a plasma both at low pressure and at atmospheric pressure. Figure 4.1 shows a typical setup for the DBD. The plasma is generated between the electrodes. Treatment of an item will typically be done in one of two ways. Either the sample will be placed (1) between the electrodes inside the discharge zone or (2) at one end such that the effluent passes over the sample. This second method is called a plasma jet when a cylindrical discharge is used instead of the block geometry of a typical DBD setup.

![Diagram of DBD setup](image)

Figure 4.1: Typical DBD. Here the dielectric material is on the high voltage electrode. DBD are typically energized with kHz or MHz voltage supply.

Ragni [80] used a standard DBD setup (here glass was used as the dielectric) to decontaminate egg shells inoculated with *Salmonella*. The benefit of DBD discharge is that it provides a somewhat diffuse discharge (larger processing volume) with little chance of sparking since the dielectric will typically prevent a transition of a streamer to a spark by the buildup of charge on the dielectric, which reduces the electric field. One of the major
drawbacks to using DBD is the possibility of the dielectric material getting on or interacting with the sample and thus contaminating it. One popular material used in plasma jets is quartz since it does not typically interact with the sample. The limiting factors to using quartz are cost and geometry. Atoms from the metal electrodes can also be liberated, but this is a negligible concern at best. Stainless steel is typically used in the reactor to mitigate health concerns if ingested.

**Plasma Jet**

Plasma jets are a subset of DBD in which one or both electrodes are insulated; the main difference between plasma jets and other DBD is the directionality of the plasma. As can be seen in Figure 4.1, for the typical DBD setup the discharge area has a large volume where the object to be processed can be placed in the gap or next to the gap. The plasma jet however, is collimated and directed by pushing the plasma out in a specific localized direction. Figure 4.2 illustrates a plasma jet which has been used on living tissue by Hoentsch et al. [42]. Hoentsch et al. [42] worked with living tissue, which is tangentially related to the work presented here in that this plasma jet is used to decontaminate a wound and promote healing. This particular plasma jet was used as an example because it uses argon as its carrier gas and illustrates how this technology is being applied in different fields.
Figure 4.2: Example of a plasma jet used by Hoentsch [42]. This jet uses argon as a carrier gas. The plasma jet affords a great deal of control and localization for processing.

For this plasma jet (Figure 4.2) the excitation is done using a high frequency electrode operated at 1.1 MHz [42]. This, however, is not the only way to excite a plasma jet and any excitation for DBD will of course work for a plasma jet. One key difference between DBD and plasma jets compared to APWIP corona discharge presented in this work is the ability to collect all or some of the charge prior to the surface treatment. The benefit of collecting the charge species is a reduction in the chemical reactions on the surface. Another drawback to the typical DBD and plasma jets is the use of pulsed or high-frequency excitation. The equipment is more expensive and adds layers of failure points to the system.
4.2 Experiments

A variety of experiments were done exploring the effects of the APWIP at decontaminating the produce that had been inoculated with *E. coli*. Figure 4.3 illustrates how the produce was exposed to the plasma. Between the needles and the stainless steel (SS) mesh the gas or gas mixture gets excited. The gas flow is from top to bottom. So, the activated or reactive species move out of the plasma zone with some of the charged particles being collected by the mesh such that the majority or the reactive species have a neutral charge. The produce is not moved so only the top area exposed to the gas interacts with the plasma effluent.

![Image of reactor with food](image)

Figure 4.3: Example of Reactor with food. The petri dish just fits through the window. Zip-ties have been replaced with a wire to secure the mesh to the ground ring.

Each sample is inoculated with *E. coli* prior to treatment in accordance with standard
techniques used in food science; see Bermudez [25] for more details. Inoculation consisted of incubating the produce in a solution containing E. coli at a concentration of $10^5$ cfu/mL for 30 min, where cfu stands for colony forming unit [25]. This pathogen load is consistent with a real-world contamination level. After the 30 min exposure the produce was dried before the plasma treatment [25] was applied.

The produce was placed on a petri dish inside the reactor on the stage below the grounded SS mesh. The reactor was sealed and then purged before energizing the needle array. After the plasma treatment the control or untreated sample as well as the plasma-treated samples were mashed, incubated for 48 hours at 35$^o$ C and then counted [25]. The control sample is always shown as the zero time sample. Treated samples were exposed for times varying from 30 sec to 10 min, indicating the time for which the needle array was energized.

The sample preparation described above was used for each run as well as for the basic reactor operation. In addition to varying the treatment time the voltage, gas flow rate, and gas composition were varied. The variable being optimized was the effectiveness of the plasma to decontaminate the samples. The results below are quite impressive considering that only the top of the produce is exposed to the reactive species from the plasma.

4.2.1 Argon-Only Experiments

Tomatoes, lettuce, and carrots were used in these experiments. However, since the carrots are sliced no major decontamination was observed. The argon-only results focus on lettuce and tomatoes. The oxygen/argon experiments explore the decontamination of carrots in more depth. Figure 4.4 shows results as a factor of treatment time. Here you can see that, as the treatment time increases, so does the reduction in cfu/mL. A voltage of 8.1 kVRMS is used and an argon flow rate of 460 sccm. For commercial use, a treatment method needs to use the minimum possible amount of time. It is for this reason that the max treatment time was limited to 10 minutes. Treatment times greater than 10 min are expected to be
commercially prohibitive when applied to high processing volumes.

Figure 4.4: Argon APWIP *E. coli* decontamination vs. treatment time showing time dependence. The goal is to keep the treatment under 10 minutes.

Using the data from Figure 4.4 a treatment time of 10 min is fixed and the voltage is varied. Figure 4.5 shows a strong voltage dependence with the 9.07 kV RMS showing a CFU $\log_{10}$ reduction of 1.5. Tomatoes show the highest reduction, most likely due to their smooth surface and the ability for reactive species to interact with a higher percentage of the surface area.
Figure 4.5: Argon APWIP *E. coli* decontamination vs. voltage is quite strong with most of the decontamination occurring at the 9.3kVRMS voltage.

Color change in the produce was also measured; however no change was observed so these results will not be presented here. More information on color change can be found in Bermudez-Aguirre [25].

**Electron Microscopy**

Electron microscopy was performed on a lettuce sample after treatment with APWIP argon plasma for a total of 15 min. The sample was placed 5.2 cm from the top of the grounded mesh and a stepped voltage was used: 4 min at 9.3 kVRMS, then 2 min at 8.4 kVRMS, then 9 min at 7.5 kVRMS. The voltage had to be lowered because corona discharge transitioned to a thermal discharge at higher voltages. Figure 4.6 shows a healthy *E. coli* cell that has not been exposed to the argon APWIP. The cell has clean, strong, healthy cell walls. Figure 4.7 shows an *E. coli* cell after exposure to the argon plasma. The argon APWIP treated *E. coli* cell shows strong degradation of the cell wall consistent with electroporation.
Bermudez-Aguirre [25] elaborates on these observations with additional Field Emission Electron Scanning Microscopy (FESEM) images showing that the APWIP did not damage the lettuce surface but did effectively damage the *E. coli* cells. Some of these images show even more significant cell wall damage than is shown in Figure 4.7.

Figure 4.6: Untreated *E. coli* cell showing a healthy control *E. coli* cell on lettuce.
Figure 4.7: Treated *E. coli* cell. Here you can see the effect of argon plasma treatment. The damage to the cell wall in the form of electroporation is quite encouraging.

Figures 4.6 and 4.7 illustrate the effect of a pure argon APWIP plasma on the *E. coli* cells which is very encouraging in combination with seeing minimal impact on the surface of the lettuce. More work is needed to improve the decontamination rate, but it is expected that geometric factors will play a significant effect, as will improving exposure to the surface area. The next step was to examine the effect of oxygen on carrots since the plasma was not as effective on the carrot samples as it was on the lettuce and tomato. Ozone is also known to reduce other types of contaminants such as pesticides [84].
4.2.2 Argon-Oxygen Experiments

A series of preliminary experiments were run (Figure 4.8) suggesting that for a flow rate of 3 L/min of argon and low concentrations of oxygen there is an enhancement in \textit{E. coli} reproduction. Oxygen concentrations greater than 3% do begin to kill \textit{E. coli} compared to the control. The nature of carrots is much different since carrot slices are used which expose the inside of the carrot to \textit{E. coli} during the inoculation process. This might explain why pure argon and low concentrations of oxygen (less than 3%) actually enhance bacterial growth, possibly by killing off the weaker cells and increasing the food source for the healthier cells. The reduced effectiveness of the the APWIP at decontaminating cut produce is consistent with Shama [86] who also observed difficulty in the decontamination of cut melons.

![Oxygen Precursor Concentration Scan](image)

Figure 4.8: \textit{E. coli} decontamination vs. O\textsubscript{2} concentration suggests that the \textit{E. coli} on carrots actually has a higher growth rate than the control until more than 3% oxygen is used.
4.3 Conclusion

APWIP has shown the potential to decontaminate produce inoculated with *E. coli*. While results are not up to the levels seen by some of the DBD and plasma jet APWIP results – those that use helium specifically – the use of APWIP for food safety is very encouraging. The use of helium [77, 76, 93] is currently very common, but use patterns are shifting toward argon even though argon is not as effective at decontamination when compared to helium. The reason for the shift is the rising cost of the helium, which is becoming prohibitive and which will only continue to increase due to dwindling supplies. Helium is commonly found in natural gas and currently collected from underground reserves that have built up from the decay of radioactive material such as uranium [49, 28]. The use of argon is therefore not only fiscally promising, but it is also good stewardship of natural resources.

Washes seem to work only moderately well, with typical log\(_{10}\) contamination reductions of 1 or 2 [90, 91, 44], while a number of DBD and plasma jet decontamination methods have a least log contamination reduction of 3 [86, 80]. With more optimization the corona discharge APWIP can compete with DBD and plasma jets in terms of decontamination effectiveness while being cheaper to build and operate. Looking at industrial food safety requirements, the corona plasma APWIP is more scalable to be used for large scale treatment. Chapter 5 will continue looking at applications for APWIP in the promising field of polymerization.
Chapter 5

Acetylene Deposition

5.1 Introduction

Hydrocarbons have rich history of being used in plasma discharges for surface modification, typically at low pressure [83, 29, 102, 88, 105, 40]. Atmospheric pressure plasma deposition is a relatively new field attracting a growing amount of interest due to its reduced operation complexity and operation costs compared with low pressure plasma processing [88, 48]. DBD and plasma jets have both been used to deposit films [85, 60, 21, 82, 48, 99, 79, 103]; however, deposition using point-to-plane geometry is not typically performed for reasons of irregular deposition and a tendency of streamers to transition into sparks. The irregularity of deposition has been minimized with the use of a needle array with a long gap between the needle array and ground. Acetylene concentrations were always above 1% such that acetylene quenched the streamers’ potential to transition into sparks [41, 106]. In this chapter I examine deposition on the needles both for the short (4.6 cm) and long (11.2 cm) gap setups. The acetylene film deposition rate is estimated using Field Emission Electron Scanning Microscopy (FESEM).

5.2 Needle Deposition – Short Gap

Deposition on electrodes is typically avoided; there are, however, cases in which such deposition is desired. One example is in carbon fiber growth for a point to plane configuration; in this case, the carbon fibers grow on the needle tips [92]. Sobczyk et al. [92] used cy-
clohexane as the precursor gas in an argon-dominant environment at atmospheric pressure with a positive DC discharge to grow carbon fibers from needle tips [92]. Sobczyk et al. [92] showed that, for low current discharges, less than 1.6 mA of the carbon deposited on the needle was simply spherical. For discharges with a current between 1.6 mA and 2.2 mA, smooth fibers in the mm range were grown [92]. Interestingly, when the discharge current is more than 3 mA, Sobczyk et al. [92] describes the fibers as irregular: “chain-like structures of soot and larger spheroid grains forming chains were also deposited on the needles.” Unfortunately, Sobczyk et al. [92] did not include images of these chains. However, from the description by Sobczyk et al. [92] the carbon growth described is similar to the deposition observed on the needles for the reactor used in this dissertation. A sample of the carbon fiber growth from the needle tip in our experiments, viewed through an optical microscope, can be seen in Figure 5.1. Figure 5.1 is consistent with carbon fiber growth seen for high-current discharge (greater than 3 mA) by Sobczyk et al. [92]. Interestingly, when a long gap is used, the irregular deposition becomes stretched and appears more linear, but still remains somewhat irregular. Additional images of the short gap needle deposition can be found in Appendix A.1.
Operating in the short gap regime minimized deposition on the needles as well as on the inside of the reactor. While this scenario was optimal for studying the needle, the reactor needed to be adjusted for deposition on a substrate. Benedikt et al. [21] showed a decrease in deposition thickness from 7 nm/s to 1.3 nm/s when the gap between the acetylene/argon plasma jet was reduced from 1 mm to 2 mm. As only these two values are given by Benedikt et al. [21] there is not enough data to allow for an estimation of decay rate as it becomes negative at 3 mm. These data do, however, point to the dependence of deposition thickness on the gap for any atmospheric plasma deposition. To address this apparent rapid decrease in deposition thickness, a long needle to ground mesh gap was used for film deposition.
5.3 Needle Deposition – Long Gap

In order to compensate for rapid decreases in deposition observed by Benedikt et al. [21], an extend gap and a higher flow rate was used relative to the sympathetic corona current pulse measurement settings. The rationale behind this change in the reactor setup was increasing the power that could be applied and to enable the use of return strokes to activate the acetylene gas molecules as well as acetylene which had been polymerized in the gap. Using primary and secondary streamers as well as return strokes means that molecules can be activated by the streamers primarily at the needle tip and, secondarily, near the ground screen with the return strokes. The distance between the needle tips and the top of the grounded mesh is 11.2 cm with an applied voltage of 8.4 kV RMS. A 1.7% concentration of acetylene is used with an argon flow rate of 6818 sccm. While the deposition on the substrate will be discussed in Section 5.5 the deposition on the needles is significantly longer when compared to the deposition on the needles for the short gap configuration.
Figure 5.2: Deposition on needles in the long-gap scenario is about four times longer than the deposition seen for the short-gap case. The deposition is much more linear when compared to the deposition seen on the short-gap configuration. When viewed under the optical microscope the deposition is mostly yellow.

In addition to the gap change the voltage and flow rate were also increased from the short- to the long-gap runs. The voltage increased from 6.5 kV RMS to 8.4 kV RMS and the argon flow rate increased from 1242 sccm to 6718 sccm. The operation did not change significantly in terms of peak current. The average peak current pulse is 30 mA; however, the number of pulses did increase. Figure 5.2 illustrates the difference in the deposition on the needle between the two cases, with length changing from 500 µm to 2 mm with the transition from a short to a long gap. Additionally, the long-gap case yields a deposition much more yellow in color with few if any black portions compared to the short gap case.

Figure 5.3 is an FESEM image of the deposition under 1000x magnification showing the
tip of the deposition collected from the reactor shown in Figure 5.4. The radius of curvature was measured to be 13.08 \( \mu \text{m} \), much smaller than the nominal 45 \( \mu \text{m} \) radius of curvature of the needles.

Figure 5.3: Long-gap FESEM deposition looks very much like the tip of a needle. The 13 \( \mu \text{m} \) radius of curvature is much smaller than the 45 \( \mu \text{m} \) mean needle radius of curvature.

5.4 Needle Deposition Discussion

As mentioned in Chapter 3.3 on sympathetic corona current pulses, the deposition was removed between runs to ensure consistent starting conditions for each run. When starting reactor with clean needles, it typically took several minutes for the corona discharges to
achieve a magnitude greater than 10 mV. I hypothesize that the typical 10-20 mA peak current pulses are due to the deposition on the needles, which acts as an extension to the metal needle that is continually being resharpened. The deposition collected off the needles after running with a short gap is shown in Figure 5.4(a). When viewed under the microscope the deposition is black and yellow. The black is expected to be a more carbon-rich portion while the yellow results from carbon-hydrogen bonds. The deposition from the needle has more black portions when run in the short-gap configuration than it did when reactor was operated in the long-gap configuration.

The long gap needle deposition is significantly more noticeable. Figure 5.4 shows the reactor after (a) short gap and (b) a long gap run. Deposition on the needles can be easily seen for the long-gap run while the short gap is noticeable, but not significant. Our hypothesis is that higher carbon concentration can be found in the short gap needle deposition and we expect that this will lead to a more brittle deposition which limits the maximum length by breaking.
Figure 5.4: Comparison between short- and long-gap configuration of the reactor needle. The short-gap deposition is very small compared to the long-gap configuration in which it is easy to achieve deposition on all the needles.

The reactor has been run continuously for two hours in the short-gap configuration and for one hour in the long-gap case. Running the reactor for longer durations has not been explored and it is unclear if the deposition will continue until the gap is reduce enough for a streamer to transition to an arc or if, as in the short-gap case, the deposition will break off at some point prior to this transition. Ideally, the deposition seen when operating for time longer than one hour with a long gap will break off, resulting in consistent reactor operation.

5.5 Substrate Deposition

All film deposition was performed using a long gap as described above (Section 5.3). The gap between the needle tips and the grounded mesh is 11.2 cm. The gap between the bottom of the grounded mesh and the top of the substrate is less than 5 mm. To estimate the acetylene
film deposition rate potassium bromide (KBr) was used to form pellets on which the film was deposited. KBr is a salt that dissolves quickly in water and has a crystalline geometry similar to sodium chloride (table salt). The KBr pellets were placed in the reactor for one hour before the film thickness was measured.

Two methods were used to calculate the film thickness. The first involved dissolving the KBr pellet and placing the film onto a mica slide. Mica was used because it has a very uniform flat surface. Figure 5.5 shows the FESESM image of the mica slide with the plasma-polymerized acetylene film. In order to see both the edge and the top of the film the mica slide was placed at a 45° angle from the horizontal. The annotation on Figure 5.5 indicates the mica slide and the acetylene film. The actual film thickness, corrected for the angle, was 8µm. A film thickness of 8µm for a one hour treatment equals a deposition rate of 2.2 nm/s which is similar to the deposition rate evidenced by Benedikt et al. [21].
Figure 5.5: FESEM mica (29x Magnification), which was attached at a 45° angle so the top of the slide is on the right and the edge of the slide is on the left.

The second method used to measure the deposition rate was to cool the KBr pellet with liquid nitrogen (-196 °C) and then break the KBr pellet in order to look at the height of the film along one of the broken edges. Figure 5.6 shows the film at low magnification and Figures 5.7 to 5.9 show the film at successively higher magnification. The first image in the series (Figure 5.6) shows one of the broken edges of the KBr pellet with the various components labeled. Note that the outer edge of the pellet is at the bottom of Figure 5.6, where the KBr pellet flairs out. The area shown in Figure 5.7 and 5.8 is highlighted in
Figure 5.6 with the square box. The deposition measurements are taken approximately 4 mm from the edge. Figure 5.7 is focused on broken edge of the KBr pellet, the acetylene (C$_2$H$_2$) film can be seen in the background looking like a mountain range. Using the working distance (WD) shown in the FESEM images the distance from the edge of the KBr to the film is estimated to be 5.3 mm. Figure 5.8 is focused on the KBr film. Using the lowest point of the ridge line the film’s height is measured to be 83 µm thick. This gives a deposition rate of 23 nm/s, higher than Benedikt’s et al. [21] 7 nm/s, but much lower than Salge’s [85] 666.6 nm/s. The last FESEM image of the KBr pellet (Figure 5.9) is a closer view of the acetylene film, focusing on the crystal, which is most likely KBr that became embedded in the film when the pellet was broken.
Figure 5.6: FESEM KBr (29x Magnification) with a view of the sample holder, adhesive tape, and the KBr pellet. A box has been drawn around the area used to measure the deposition rate.
Figure 5.7: FESEM KBr (200x Magnification) showing the edge of the KBr film on top of the adhesive which holds the film to the sample holder. The out-of-focus ridge is the acetylene film, which has receded from edge of the KBr.
Figure 5.8: FESEM KBr (250x Magnification) showing the edge of the KBr focusing down the top of the slide on the ridge, which is the acetylene film.
The average lower bound of free electron power for the positive voltage is $3.4 \pm 0.9$ W. To calculate power, a LeCroy 9350AL 500 MHz 2 channel oscilloscope was used to collect the positive period of the applied AC voltage. Approximately 100 oscillograms of the current and voltage were collected to use in this calculation. The instantaneous power calculation was made by multiplying the voltage and current after current values less than 2 mA were zeroed. This instantaneous power is integrated to yield the positive period energy, which is
divided by the positive voltage time to estimate the average power. Only the positive portion of the applied voltage is calculated since the negative corona discharges have a much lower magnitude approaching the noise level (See Appendix B for more details). This only gives a lower bound on the free electron energy since the ions are not moving quickly enough to be picked up on the Pearson [6]. This free electron power is however, important because the APIWP is all about cold ions and hot electrons. The electrons are, therefore, expected to be doing most of the work.

To explore the use of different substrates KBr was replaced with filter paper. The same settings that were used to deposit the acetylene film on the KBr substrate were used to deposit on the filter paper. Figure 5.10 and Figure 5.11 compare the untreated and treated filter paper. The filter paper treated with the acetylene film for one hour clearly shows nodules, possibly on top of a film that is smoothing out the fibers of the filter paper. The complex nature of the filter paper makes the analysis of the film deposited on the film difficult; nonetheless, the deposition is consistent with that seen by colleagues [56]. See Appendix A.2 for additional images of the deposited film.
Figure 5.10: FESEM filter paper that was not treated.
Figure 5.11: FESEM filter paper treated with plasma polymerized acetylene. Compare with the untreated filter paper and note that nodules are observed only for the treated samples. The measured deposition rate would suggest that this filter is coated with an acetylene film with nodules on top.

5.6 Conclusion

In this chapter APWIP has been used to deposit film with a deposition rate between 2.2 nm/s and 23 nm/s using an average free electron power of 3.4 ± 0.9 W. The key to film deposition was recognizing that the activated gas phase acetylene molecules were quickly quenched. To compensate, the gap was increased to 11.2 cm which moved into a scenario that allowed return strokes in addition to the use of higher voltages which translate into higher power.
Chapter 6

Discussion

6.1 Considerations

In this chapter I will discuss some of the observations made while conducting the experiments described in this dissertation. I will begin by discussing the reactor operation and a qualitative corona mode map, then give a summary of the work described in this dissertation and finally future work.

6.1.1 Reactor Operation Observations

In Chapter 2 the reactor was described along with improvements made to ensure consistent operation of the reactor. After designing the basic reactor, several general operating studies were conducted to explore the basic response of the reactor to different flow rates and different concentrations of acetylene. The results of these studies confirm observations made by others that, as the flow rate increases or the acetylene concentration decreases, the needed voltage decreases \[106\]. Gas breakdown can be difficult to quantify and is strongly dependent on the sensitivity of the equipment and environment.

The dependence on the flow rate and acetylene concentration of the voltage has thus been explored in a general manner by looking at the voltage needed to produce current pulses in the 10 mA range. A 10 mA current pulse magnitude was chosen because it is well above the noise level of the oscilloscope which is in the 1 mV range. Figure 6.1 shows the 10% \( \text{C}_2\text{H}_2 \) and 0.95% \( \text{C}_2\text{H}_2 \) scans as well as a 1% \( \text{C}_2\text{H}_2 \) and 5% \( \text{C}_2\text{H}_2 \) case. The 10% \( \text{C}_2\text{H}_2 \) and 0.95% \( \text{C}_2\text{H}_2 \) scans looked at the extreme values of the voltage as well as argon flow rate.
Photographs (Figure 6.1.1) of the plasma have been annotated to indicate the voltage and the exposure time in addition to the argon flow rate. The needle array has been overlaid on the top portion of photos 6.3(g) and 6.3(b) to confirm the discharge region.

Figure 6.1: Voltage as a function of the flow rate for a typical 10 mA current peak current pulse.
Figure 6.2: Photographs of the plasma at different voltages and argon flow rates for a 10% $\text{C}_2\text{H}_2$
Figure 6.3: Photographs of the plasma at different voltages and argon flow rates for 0.95% \( \text{C}_2\text{H}_2 \)
6.1.2 Corona Mode Map

Having already explored the relationship between the voltage and gas flow rate on corona discharges it seems prudent to mention some qualitative guidelines for corona discharges and the needle-to-plane gap distance. Because Kuffel and Zaengl [53] have already started such a guide, I will start with their figure and then add information about return strokes as outlined by Sigmond [89]. I will also add the observations made in this work that illustrate when deposition is seen on a substrate. Figure 6.4 illustrates the effect of increasing the gap. As the gap increases the voltage also increases until a threshold is reached. After the gap threshold is reached onset streamers are seen; however, the onset streamers typically won’t reach the ground [53]. As the voltage increases the first onset streamers are seen, followed by glows and streamers, followed by breakdown streamers [53]. There is nothing particularly special about breakdown streamers; the voltage is such that one starts to balance on a proverbial knife’s edge. Depending on how the discharge falls it can remain cool, in which case it is called a breakdown streamer, or it can fall the other way and transition to a thermal discharge or spark.

Return strokes occur when a primary streamer connects with the ground, freeing enough electrons to create an anode-directed streamer [89]. To observe return strokes a gap greater than 5 cm is needed, according to Sigmond [89]. Figure 6.4 shows the return stroke region being above the onset and, of course, below the breakdown streamers; remember that only streamers that cross the gap can result in return strokes. Deposition (labeled in Figure 6.4) on a substrate was only observed in the case of long gaps, which also corresponds to the region in which return strokes occur. This qualitative diagram (Figure 6.4) illustrates the corona regimes and the regime which was used for deposition.
Figure 6.4: Corona mode map illustrating where various corona discharges are expected. The gap spacing is the distance between the needle tip and the ground plane. Overlayed on this diagram is the deposition seen in the reactor.

6.2 Summary

An APWIP reactor was built to test needle to needle streamer synchronicity as well as being versatile enough to process produce and deposit film on small substrate samples. The materials used to build the reactor are all readily available and relatively inexpensive. The addition of the window was conceptually straight-forward, but required a machinist. Excitation was done using a HV AC signal realized with an off-the-shelf VARIAC and a low-noise GE transformer.

Streamer synchronicity was found to be dependent on the gap between the needles being considered. After collecting samples for each needle pair using needle one as the pivot needle...
it was found that, for gaps less than 3 cm, the interaction between needles is significant. For gaps greater than 3 cm, on the other hand, there is no appreciable synchronicity. This dependence on gap spacing is a key design component to be considered when designing a reactor.

The APWIP has shown potential in decontamination of produce inoculated with \textit{E. coli}. Probably the most exciting aspect of these results is how well the argon-only plasma works at killing the \textit{E. coli} due to the inert nature of the argon and the likelihood of this treatment being readily accepted as safe.

Plasma-polymerized acetylene film deposition was conducted on both KBr pellets and filter paper to explore deposition on different substrates. A long gap was needed to deposit the film. Due the high quenching rate, the long gap allowed for the return strokes which activated the acetylene near the substrate. Deposition rate was between 2.2 nm/s and 23 nm/s, consistent with deposition rates seen by others [21, 85].

It should be noted that the film deposited and measured on the KBr pellet had receded from the edge. While this could be due to a number of reasons it is possible that the film rolled up, if this did happen then it is estimated that the un-rolled film thickness would be on the order of microns rather than tens of microns. More information can be seen in Appendix C, but the film rolling up has been observed by colleagues [56]. Having a thinner film near the center of the KBr pellet compared to near its outer radius would be consistent with expected deposition patterns dominated by convection and diffusion of radicals around and to the substrate.

While the reactor provided a platform to study both the steamer interactions and some applications steamers are at the center of the research. Steamer synchronicity study proved to be critical in being able to confidently use the reactor for both the food decontamination and film deposition. The reactor has proven to be very versatile two applications have been demonstrated, but the possible applications are many.
6.3 FUTURE WORK

There are a myriad options for the direction future work could take. Polymerized film deposition is very exciting because it might be possible to decontaminate fresh produce and then apply an edible polymer film that extends shelf life and slows nutrition losses. In addition to applying a simple protective film to the produce one can also imagine the film being used to fortify produce for increased health benefits.

With those types of goals in mind the deposition range needs to be explored and parameterized. The range of deposition will be a critical factor when designing future reactors. Power and its dependence on deposition is another key factor. This dependence suggests a response surface approach to parameterize the deposition rate based on the independent variables of power and deposition range.

The need to use return strokes is an interesting factor which needs additional research along with the power injection into the plasma relative to the power injected at the needles. The use of return strokes should also be investigated for the decontamination of produce, as it might be more effective than the short gap approach.
Bibliography


[14] M. Abdel-Salam and N. L. Allen. Onset voltage of positive glow corona in rod-plane
2005.

tive corona pulse characteristics from two interacting needles in Air. *IEEE Transactions

[16] M. Abdel-Salam and A. Hashem. Positive corona inception in point-plane gaps as

[17] M. Abdel-Salam, M. Nakano, and A. Mizuno. Corona-induced pressures, potentials,
fields and currents in electrostatic precipitator configurations. *Journal of Physics D:

[18] M. Abdel-Salam and E. K. Stanek. Mathematical-physical model of corona from surges
on high-voltage lines. *IEEE Transactions on Industrial Applications*, IA-23(3):481–489,
1987.

[19] M. Abdel-Salam, A. A. Turky, and A. A. Hashem. The onset voltage of coronas on
1998.


A. V. Phelps. *Cross sections and swarm coefficients for nitrogen ions and neutrals in N2 and argon ions and neutrals in Ar for energies from 0.1 eV to 10 keV*, volume 20. 1991.


Appendices
Appendix A

Additional Reactor Information

Figure A.1: Safety interlock box with barrier shown below the orange/black power cable.
Figure A.2: HV to needle array block with the transformer Pearson on the right and needle 1 and the other needle Pearsons on the left. Voltage divider connected to the needle array block in the middle.
Figure A.3: Needle 1 and Other Needle Pearsons with needle 1 going through the Pearson labeled Ch3. Channel 4 Pearson used to monitor the current through the other needle.
Figure A.4: Reactor 3 with Dimensions
Figure A.5: KBr deposition run
Figure A.6: Reactor showing deposition inside reactor when run using a long gap configuration.
Figure A.7: VARIAC to HV (kV RMS) data fit
A.1 Short Gap Needle Deposition

Figure A.8: Short gap needle deposition 10x shows the irregular pattern with multiple paths moving away from the needle tip.
Figure A.9: Short gap needle deposition 20x on an optical microscope. The lighter colored portions are yellow while the darker colored segments are black.
A.2 Additional FESEM Film Deposition Images

Figure A.10: Untreated KBr pellet surface showing the surface that the film is deposited on.
Figure A.11: The polymerized acetylene film deposited on the KBr with the film showing similar pattern as the one from the untreated KBr pellet.
Figure A.12: The polymerized acetylene film deposited on mica slide after dissolving the KBr away. A similar pattern is that seen on both treated and untreated KBr pellet. The texture has changed due to the addition of the polymerized film on the treated KBr and the film raft, which was placed on the mica slide.
Appendix B

Power Calculation Illustrated

As mentioned in Chapter 5 the power was calculated by integrating the product of the voltage and current of the positive voltage period. This integral is of course the energy and the average power was calculated by dividing energy by $\Delta t$ (see Figure B.2).

Figure B.1: LeCroy deposition oscillogram
Figure B.2: LeCroy signals used for power calculation
Appendix C

Thin Film Thickness

As mentioned in Chapter 6 the reason for the increased thickness of the film on the KBr pellet could be due to the film rolling up. Figure C.1 illustrates this idea. If the film did roll up on the KBr its thickness (1 \( \mu \text{m} \)) is similar to the thickness measured on the mica slide (8.3 \( \mu \text{m} \)).

Conservation of mass considerations allow one to write:

\[
I = \pi N^2 h \tag{C.1}
\]

\[
H = 2Nh \tag{C.2}
\]

From Figure 5.8 we can estimate \( I = 5,300 \ \mu \text{m} \) and \( H = 83 \ \mu \text{m} \) which when inserted into...
the two equations shown above yield $N=41$ and $h=1 \, \mu m$. This model was not studied in detail but should be considered by researchers using similar techniques (KBr substrates) to study these and similar APWIP-polymerized films.
Appendix D

MATLAB Code

A GUI interface was built for MATLAB to centralize the gas flow rate information as well as the collection of current and voltage signals. Figure D.1 shows the GUI interface built.

![MATLAB Code GUI](image)

Figure D.1: MATLAB data collection GUI is used to set and log the gas flow rates and notes in addition so specify the collection parameters.

D.0.1 Matlab Code GUI Code:

```matlab
function varargout = get_data_gui(varargin)
% GET_DATA_GUI MATLAB code for get_data_gui.fig
% GET_DATA_GUI, by itself, creates a new GET_DATA_GUI or raises the existing singleton.
% H = GET_DATA_GUI returns the handle to a new GET_DATA_GUI or the handle to the existing singleton.
```

135
GET_DATA_GUI('CALLBACK', hObject, eventData, handles, ...) calls the local
function named CALLBACK in GET_DATA_GUI.M with the given input arguments.

GET_DATA_GUI('Property', 'Value', ...) creates a new GET_DATA_GUI or raises the
existing singleton*. Starting from the left, property value pairs are
applied to the GUI before get_data_gui_OpeningFcn gets called. An
unrecognized property name or invalid value makes property application
stop. All inputs are passed to get_data_gui_OpeningFcn via varargin.

*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
instance to run (singleton)."

See also: GUIDE, GUIDATA, GUIHANDLES

Edit the above text to modify the response to help get_data_gui

Last Modified by GUIDE v2.5 18-Feb-2012 11:52:14

Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @get_data_gui_OpeningFcn, ...
    'gui_OutputFcn', @get_data_gui_OutputFcn, ...
    'gui_Layou Fn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

--- Executes just before get_data_gui is made visible.
function get_data_gui_OpeningFcn(hObject, ~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to get_data_gui (see VARARGIN)

% Choose default command line output for get_data_gui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes get_data_gui wait for user response (see UIRESUME)
% uwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = get_data_gui_OutputFcn(~, ~, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function H2C2_percent_CreateFcn(hObject, ~, handles)
% hObject handle to H2C2_percent (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
function H2C2_reading_Callback(~, ~, ~)

    % hObject handle to H2C2_reading (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of H2C2_reading as text
    % str2double(get(hObject,'String')) returns contents of H2C2_reading as a double

% --- Executes during object creation, after setting all properties.
function H2C2_reading_CreateFcn(hObject, ~, ~)

    % hObject handle to H2C2_reading (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    % See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function O2_percent_Callback(~, ~, ~)

    % hObject handle to O2_percent (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of O2_percent as text
    % str2double(get(hObject,'String')) returns contents of O2_percent as a double

% --- Executes during object creation, after setting all properties.
function O2_percent_CreateFcn(hObject, ~, ~)
% hObject handle to O2_percent (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end

function O2_reading_Callback(~, ~, ~)
% hObject handle to O2_reading (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of O2_reading as text
% str2double(get(hObject,'String')) returns contents of O2_reading as a double

% --- Executes during object creation, after setting all properties.
function O2_reading_CreateFcn(hObject, ~, ~)
% hObject handle to O2_reading (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end

function Ar_reading_Callback(~, ~, ~)
% hObject handle to Ar_reading (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

139
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ar_reading as text
% str2double(get(hObject,'String')) returns contents of Ar_reading as a double

% --- Executes during object creation, after setting all properties.
function Ar_reading_CreateFcn(hObject, ~, ~)
    % hObject handle to Ar_reading (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    % See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function Ar_psig_Callback(hObject, eventdata, handles)
    % hObject handle to Ar_psig (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of Ar_psig as text
    % str2double(get(hObject,'String')) returns contents of Ar_psig as a double

    % --- Executes during object creation, after setting all properties.
    function Ar_psig_CreateFcn(hObject, eventdata, handles)
        % hObject handle to Ar_psig (see GCBO)
        % eventdata reserved - to be defined in a future version of MATLAB
        % handles empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        % See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

140
function H2C2_psig_Callback(~, ~, ~)
% hObject handle to H2C2_psig (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of H2C2_psig as text
% str2double(get(hObject, 'String')) returns contents of H2C2_psig as a double

% --- Executes during object creation, after setting all properties.
function H2C2_psig_CreateFcn(hObject, ~, ~)
% hObject handle to H2C2_psig (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Untitled_1_Callback(~, ~, ~)
% hObject handle to Untitled_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in gas_flow_check.
function gas_flow_check_Callback(~, ~, handles)
% hObject handle to gas_flow_check (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

temp_data(1) = str2double(get(handles.Ar_reading, 'String'));
temp_data(2) = str2double(get(handles.O2_reading,'String'));

temp_data(3) = str2double(get(handles.O2_percent,'String'));

temp_data(4) = str2double(get(handles.H2C2_reading,'String'));

global gas_data;
gas_data = gas_flow_FUN(temp_data);

set(handles.suggested_O2_setting,'String',num2str(gas_data(1)));
set(handles.suggested_H2C2_setting,'String',num2str(gas_data(2)));
set(handles.Ar_flowrate,'String',num2str(gas_data(3)));
set(handles.O2_flowrate,'String',num2str(gas_data(4)));
set(handles.H2C2_flowrate,'String',num2str(gas_data(5)));
set(handles.total_gas_flow,'String',['Total Gas flow = ',num2str(sum(gas_data(3:5))),' (mL/min)']);

% --- Executes on button press in tek_start_current.
function tek_start_current_Callback(~, ~, handles)
    % hObject handle to tek_start_current (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    IP_ADDRESS = get(handles.tek_current_ip,'String');
    folder_header = get(handles.tek_current_folder_name,'String');
    file_header = get(handles.tek_current_file_name,'String');
    file_header = get(handles.tek_current_file_name,'String');
    temp_data(1) = str2double(get(handles.tek_current_vloop,'String'));
    temp_data(2) = str2double(get(handles.tek_current_data_sample,'String'));
    temp_data(3) = str2double(get(handles.tek_current_timebase,'String'));
    temp_data(4) = str2double(get(handles.tek_current_trigger_level,'String'));
    temp_data(5) = str2double(get(handles.tek_current_div,'String'));

tic
[done_note,device_info] = TekTronix_TDS3014B_Get_I_pulses_trigger_on_I_FUN(IP_ADDRESS,...
    temp_data,folder_header,file_header);
toc_time = toc;

% Write out log
[Y, M, D, -, -, S] = datevec ( now );
folder_name = strcat(folder_header,int2str(M),'_',int2str(D),'_',int2str(Y));
filename = strcat(folder_name,'/log','_',int2str(M),'_',int2str(D),'_',int2str(Y),'.dat');
dataLOG = fopen(filename, 'w');
% Write log
fprintf(dataLOG, '-------- Tektronix Current Pulse DATA Triggered on Current --------
');
fprintf(dataLOG, 'Erik Wemlinger
');
fprintf(dataLOG, 'Created at: %s
', timestring());
fprintf(dataLOG, 'Device Information: %s
', device_info);
fprintf(dataLOG, 'Elapsed time to collect data: %d (s)
', toc_time);
fprintf(dataLOG, '-------- Flow rate information --------
');
get(handles.Ar_flowrate,'String');
global Ar_flowrate
get(handles.O2_flowrate,'String');
global O2_flowrate
get(handles.H2C2_flowrate,'String');
global H2C2_flowrate
fprintf(dataLOG, 'Argon flow rate (%s mm): %s (mL/min)
', get(handles.Ar_reading,'String'), ...
get(handles.Ar_flowrate,'String'));
fprintf(dataLOG, 'Oxygen flow rate (%s mm): %s (mL/min)
', get(handles.O2_reading,'String'), ...
get(handles.O2_flowrate,'String'));
fprintf(dataLOG, 'Acetylene flow rate (%s div): %s (mL/min)
', get(handles.H2C2_reading,'String'), ...
get(handles.H2C2_flowrate,'String'));
fprintf(dataLOG, 'Acetylene flow rate percentage %s %
', get(handles.H2C2_percent,'String'));
fprintf(dataLOG, '-------- GUI Settings --------
');
fprintf(dataLOG, 'Folder Header: %s 
', folder_name);
fprintf(dataLOG, 'File Header: %s 
', file_header);
fprintf(dataLOG, 'Number of voltage samples to take: %s 
', get(handles.tek_current_vloop, ...
'String'));
fprintf(dataLOG, 'Number of data samples to take: %s 
', get(handles.tek_current_data_sample, ...
'String'));
fprintf(dataLOG, 'Scope timebase: %s 
', get(handles.tek_current_timebase,'String'));
fprintf(dataLOG, 'Scope trigger level (CH 1, total current): %s 
', ...
get(handles.tek_current_trigger_level,'String'));
fprintf(dataLOG, '-------- Notes --------
');
temp_notes = get(handles.notes,'String');
[nrows,ncols] = size(temp_notes);
for row=1:nrows
fprintf(dataLOG, '%s
', temp_notes(row,:));
end
fprintf(dataLOG, '-------- END OF NOTES --------
');
fprintf(dataLOG, 'Thats All Folks\n');

% Close files
fclose(dataLOG);

% Let folks know your done!!!

143
msgbox(done_note);

function tek_current_ip_Callback(~, ~, ~)
  hObject handle to tek_current_ip (see GCBO)
  eventdata reserved - to be defined in a future version of MATLAB
  handles structure with handles and user data (see GUIDATA)

  % Hints: get(hObject,'String') returns contents of tek_current_ip as text
  %       str2double(get(hObject,'String')) returns contents of tek_current_ip as a double

  % --- Executes during object creation, after setting all properties.
  function tek_current_ip_CreateFcn(hObject, ~, ~)
  hObject handle to tek_current_ip (see GCBO)
  eventdata reserved - to be defined in a future version of MATLAB
  handles empty - handles not created until after all CreateFcns called

  % Hint: edit controls usually have a white background on Windows.
  %       See ISPC and COMPUTER.
  if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
  end

function tek_current_folder_name_Callback(~, ~, ~)
  hObject handle to tek_current_folder_name (see GCBO)
  eventdata reserved - to be defined in a future version of MATLAB
  handles structure with handles and user data (see GUIDATA)

  % Hints: get(hObject,'String') returns contents of tek_current_folder_name as text
  %       str2double(get(hObject,'String')) returns contents of tek_current_folder_name as a double

  % --- Executes during object creation, after setting all properties.
  function tek_current_folder_name_CreateFcn(hObject, ~, ~)
  hObject handle to tek_current_folder_name (see GCBO)
  eventdata reserved - to be defined in a future version of MATLAB
  handles empty - handles not created until after all CreateFcns called
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in tek_start_volt.
function tek_start_volt_Callback(~, ~, handles)
    % hObject handle to tek_start_volt (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)
    IP_ADDRESS = get(handles.tek_volt_ip,'String');
    folder_header = get(handles.tek_volt_folder_name,'String');
    file_header = get(handles.tek_volt_file_name,'String');
    temp_data(1) = str2double(get(handles.tek_volt_vloops,'String'));
    temp_data(2) = str2double(get(handles.tek_volt_sine_cycles,'String'));
    temp_data(3) = str2double(get(handles.tek_volt_timebase,'String'));

    tic
    [done_note,device_info] = TekTronix_TDS3014B_GetSympatheticPulses_FUN(IP_ADDRESS,temp_data, ...          
                        folder_header,file_header);
    toc_time = toc;

    % Write out log
    [Y, M, D, H, MN, S] = datevec ( now );
    folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
    filename = strcat(folder_name,'/log','_',int2str(M),'-',int2str(D),'-',int2str(Y),'.dat');
    dataLOG = fopen(filename, 'w');

    % Write log
    fprintf(dataLOG, '-------- Tektronix Current Pulse DATA Triggered on Voltage --------
');
    fprintf(dataLOG, 'Erik Wemlinger
');
    fprintf(dataLOG, 'Created at: %s
',timestring());
    fprintf(dataLOG, 'Device IP Address: %s
',IP_ADDRESS);
    fprintf(dataLOG, 'Device Information: %s
',device_info);
    fprintf(dataLOG, 'Elapsed time to collect data: %d (s)
',toc_time);
fprintf(dataLOG, '-------- Flow rate information --------
');
fprintf(dataLOG, 'Argon flow rate (%s mm): %s (mL/min)
',get(handles.Ar_reading,'String'), ...
    get(handles.Ar_flowrate,'String'));
fprintf(dataLOG, 'Oxygen flow rate (%s mm): %s (mL/min)
',get(handles.O2_reading,'String'), ...
    get(handles.O2_flowrate,'String'));
fprintf(dataLOG, 'Oxygen flow rate percentage %s %%
',get(handles.O2_percent,'String'));
fprintf(dataLOG, 'Acetylene flow rate (%s div): %s (mL/min)
', ...
    get(handles.H2C2_reading,'String'), get(handles.H2C2_flowrate,'String'));
fprintf(dataLOG, 'Acetylene flow rate percentage %s %%
',get(handles.H2C2_percent,'String'));
fprintf(dataLOG, '-------- GUI Settings --------
');
fprintf(dataLOG, 'Folder Header: %s 
',folder_header);
fprintf(dataLOG, 'File Header: %s 
',file_header);
fprintf(dataLOG, 'Number of voltage samples to take: %s \n',get(handles.tek_volt_vloops,'String'));
fprintf(dataLOG, 'Number of times to sample a complete sine period: %s \n', ...
    get(handles.tek_volt_sine_cycles,'String'));
fprintf(dataLOG, 'Scope timebase: %s 
',get(handles.tek_volt_timebase,'String'));
fprintf(dataLOG, '-------- Notes --------
');
temp_notes = get(handles.notes,'String');
[nrows,ncols]= size(temp_notes);
for row=1:nrows
    fprintf(dataLOG, '%s
', temp_notes(row,:));
end
fprintf(dataLOG, '-------- END OF NOTES --------
');
fprintf(dataLOG, 'Thats All Folks\n');

% Close files
fclose(dataLOG);
% Let folks know your done!!!
msgbox(done_note);

function tek_volt_ip_Callback(~, ~, ~)
% hObject handle to tek_volt_ip (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of tek_volt_ip as text
% str2double(get(hObject,'String')) returns contents of tek_volt_ip as a double
function tek_volt_ip_CreateFcn(hObject, ~, ~)
% hObject handle to tek_volt_ip (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tek_volt_folder_name_Callback(~, ~, ~)
% hObject handle to tek_volt_folder_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_volt_folder_name as text
% str2double(get(hObject,'String')) returns contents of tek_volt_folder_name as a double

function tek_volt_folder_name_CreateFcn(hObject, ~, ~)
% hObject handle to tek_volt_folder_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function lecroy_start_Callback(~, ~, handles)
% --- Executes on button press in lecroy_start.
function lecroy_start_CreateFcn(hObject, ~, ~, handles)
IP_ADDRESS = get(handles.lecroy_ip,'String');
folder_header = get(handles.lecroy_folder_name,'String');
file_header = get(handles.lecroy_file_name,'String');
temp_data(1) = str2double(get(handles.lecroy_gpib_address,'String'));
temp_data(2) = str2double(get(handles.lecroy_loop_num,'String'));
tic
[done_note,prologix_vers,device_info] = LeCroy_Prologix_IVGetData_FUN(IP_ADDRESS, ...
temp_data,folder_header,file_header);
toc_time = toc;

% Write out log
[Y, M, D, H, MN, S] = datevec ( now );;
folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
filename = strcat(folder_name,'/log','_',int2str(M),'-',int2str(D),'-',int2str(Y),'.dat');
dataLOG = fopen(filename, 'w');

% Write log
fprintf(dataLOG, '-------- LeCroy Total Current and Voltage Data --------
');
fprintf(dataLOG, 'Erik Wemlinger\n');
fprintf(dataLOG, 'Created at: %s\n',timestring());
fprintf(dataLOG, 'Device IP Address: %s\n',IP_ADDRESS);
fprintf(dataLOG, 'Device Information: %s\n',device_info);
fprintf(dataLOG, 'GPIB (Prologix) INFO: %s\n',prologix_vers);
fprintf(dataLOG, 'GPIB ADDRESS: %s 
',get(handles.lecroy_gpib_address,'String'));
fprintf(dataLOG, 'Elapsed time to collect data: %d (s)
',toc_time);
fprintf(dataLOG, '-------- Flow rate information --------
');
fprintf(dataLOG, 'Argon flow rate (%s mm): %s (mL/min)
',get(handles.Ar_reading,'String'), ...
get(handles.Ar_flowrate,'String'));
fprintf(dataLOG, 'Oxygen flow rate (%s mm): %s (mL/min)
',get(handles.O2_reading,'String'), ...
get(handles.O2_flowrate,'String'));
fprintf(dataLOG, 'Oxygen flow rate percentage %s %%
',get(handles.O2_percent,'String'));
fprintf(dataLOG, 'Acetylene flow rate (%s div): %s (mL/min)
', ... 
get(handles.H2C2_reading,'String'),get(handles.H2C2_flowrate,'String'));
fprintf(dataLOG, 'Acetylene flow rate percentage %s %%
',get(handles.H2C2_percent,'String'));
fprintf(dataLOG, '-------- GUI Settings --------
');
fprintf(dataLOG, 'Folder Header: %s \n',folder_header);
fprintf(dataLOG, 'File Header: %s \n',file_header);
fprintf(dataLOG, 'Number of Complete periods to acquire: %s \n', ...
    get(handles.lecroy_loop_num,'String'));
fprintf(dataLOG, '-------- Notes --------
');
temp_notes = get(handles.notes,'String');
[nrows,ncols]= size(temp_notes);
for row=1:nrows
    fprintf(dataLOG, '%s
', temp_notes(row,:));
end
fprintf(dataLOG, '-------- END OF NOTES --------
');
fprintf(dataLOG, 'Thats All Folks\n');

% Close files
fclose(dataLOG);
% Let folks know your done!!!
msgbox(done_note);

function lecroy_ip_Callback(~, ~, ~)
% hObject handle to lecroy_ip (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lecroy_ip as text
% str2double(get(hObject,'String')) returns contents of lecroy_ip as a double

% --- Executes during object creation, after setting all properties.
function lecroy_ip_CreateFcn(hObject, ~, ~)
% hObject handle to lecroy_ip (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function lecroy_folder_name_Callback(hObject, eventdata, handles)

% hObject handle to lecroy_folder_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lecroy_folder_name as text
% str2double(get(hObject,'String')) returns contents of lecroy_folder_name as a double

% --- Executes during object creation, after setting all properties.
function lecroy_folder_name_CreateFcn(hObject, eventdata, handles)

% hObject handle to lecroy_folder_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tek_current_file_name_Callback(hObject, eventdata, handles)

% hObject handle to tek_current_file_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_current_file_name as text
% str2double(get(hObject,'String')) returns contents of tek_current_file_name as a double

% --- Executes during object creation, after setting all properties.
function tek_current_file_name_CreateFcn(hObject, eventdata, handles)

% hObject handle to tek_current_file_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
function tek_volt_file_name_Callback(~, ~, ~)
    % hObject handle to tek_volt_file_name (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of tek_volt_file_name as text
    %       str2double(get(hObject,'String')) returns contents of tek_volt_file_name as a double

function lecroy_file_name_Callback(~, ~, ~)
    % hObject handle to lecroy_file_name (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of lecroy_file_name as text
    %       str2double(get(hObject,'String')) returns contents of lecroy_file_name as a double
function lecroy_file_name_CreateFcn(hObject, ~, ~)
% hObject handle to lecroy_file_name (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function lecroy_gpib_address_Callback(~, ~, ~)
% hObject handle to lecroy_gpib_address (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lecroy_gpib_address as text
%        str2double(get(hObject,'String')) returns contents of lecroy_gpib_address as a double

% --- Executes during object creation, after setting all properties.
function lecroy_gpib_address_CreateFcn(hObject, ~, ~)
% hObject handle to lecroy_gpib_address (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

152
function lecroy_loop_num_Callback(~, ~, ~)
    hObject handle to lecroy_loop_num (see GCBO)
    eventdata reserved - to be defined in a future version of MATLAB
    handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of lecroy_loop_num as text
    % str2double(get(hObject,'String')) returns contents of lecroy_loop_num as a double

    % --- Executes during object creation, after setting all properties.
    function lecroy_loop_num_CreateFcn(hObject, ~, ~)
        hObject handle to lecroy_loop_num (see GCBO)
        eventdata reserved - to be defined in a future version of MATLAB
        handles empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

function tek_current_vloop_Callback(~, ~, ~)
    hObject handle to tek_current_vloop (see GCBO)
    eventdata reserved - to be defined in a future version of MATLAB
    handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of tek_current_vloop as text
    % str2double(get(hObject,'String')) returns contents of tek_current_vloop as a double

    % --- Executes during object creation, after setting all properties.
    function tek_current_vloop_CreateFcn(hObject, ~, ~)
        hObject handle to tek_current_vloop (see GCBO)
        eventdata reserved - to be defined in a future version of MATLAB
        handles empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
function tek_current_data_sample_Callback(~, ~, ~)
    % hObject handle to tek_current_data_sample (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of tek_current_data_sample as text
    % str2double(get(hObject,'String')) returns contents of tek_current_data_sample as a double

% --- Executes during object creation, after setting all properties.
function tek_current_data_sample_CreateFcn(hObject, ~, ~)
    % hObject handle to tek_current_data_sample (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    % See ISPC and COMPUTER.
    if ispc & isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function tek_current_timebase_Callback(~, ~, ~)
    % hObject handle to tek_current_timebase (see GCBO)
    % eventdata reserved - to be defined in a future version of MATLAB
    % handles structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of tek_current_timebase as text
    % str2double(get(hObject,'String')) returns contents of tek_current_timebase as a double
% --- Executes during object creation, after setting all properties.
function tek_current_timebase_CreateFcn(hObject, -, -)
% hObject handle to tek_current_timebase (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tek_current_trigger_level_Callback(~, ~, ~)
% hObject handle to tek_current_trigger_level (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_current_trigger_level as text
%        str2double(get(hObject,'String')) returns contents of tek_current_trigger_level as a double

% --- Executes during object creation, after setting all properties.
function tek_current_trigger_level_CreateFcn(hObject, -, -)
% hObject handle to tek_current_trigger_level (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tek_volt_vloops_Callback(~, ~, ~)
% hObject handle to tek_volt_vloops (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_volt_vloops as text
% str2double(get(hObject,'String')) returns contents of tek_volt_vloops as a double

% --- Executes during object creation, after setting all properties.
function tek_volt_vloops_CreateFcn(hObject, ~, ~)
% hObject handle to tek_volt_vloops (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tek_volt_sine_cycles_Callback(~, ~, ~)
% hObject handle to tek_volt_sine_cycles (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_volt_sine_cycles as text
% str2double(get(hObject,'String')) returns contents of tek_volt_sine_cycles as a double

% --- Executes during object creation, after setting all properties.
function tek_volt_sine_cycles_CreateFcn(hObject, ~, ~)
% hObject handle to tek_volt_sine_cycles (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function tek_volt_timebase_Callback(~, ~, ~)
% hObject handle to tek_volt_timebase (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_volt_timebase as text
% str2double(get(hObject,'String')) returns contents of tek_volt_timebase as a double

% --- Executes during object creation, after setting all properties.
function tek_volt_timebase_CreateFcn(hObject, ~, ~)
% hObject handle to tek_volt_timebase (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --------------------------------------------------------------------
function Untitled_2_Callback(~, ~, ~)
% hObject handle to Untitled_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --------------------------------------------------------------------
function Untitled_3_Callback(~, ~, ~)
% hObject handle to Untitled_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
function tek_current_div_Callback(~, ~, ~)
% hObject handle to tek_current_div (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tek_current_div as text
% str2double(get(hObject,'String')) returns contents of tek_current_div as a double

% --- Executes during object creation, after setting all properties.
function tek_current_div_CreateFcn(hObject, ~, ~)
% hObject handle to tek_current_div (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function H2C2_percent_Callback(hObject, eventdata, handles)
% hObject handle to H2C2_percent (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of H2C2_percent as text
% str2double(get(hObject,'String')) returns contents of H2C2_percent as a double

function notes_Callback(hObject, eventdata, handles)
% hObject handle to notes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of notes as text
% str2double(get(hObject,'String')) returns contents of notes as a double

% --- Executes during object creation, after setting all properties.
function notes_CreateFcn(hObject, eventdata, handles)
% hObject handle to notes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

D.0.2 Functions Called by MATLAB GUI:

This MATLAB function calculates the gas flow rates:

function [return_data] = gas_flow_FUN(input_data)
    folder_name = 'Flowmeter_Data';
    lowflow_file_name = 'Matheson_610A_Air-STP.dat';
    highflow_file_name = 'Matheson_E606_NUM6-Tube_Ar_Glass.dat';
    highflow_SS_file_name = 'Matheson_E606_NUM6-Tube_Ar_SS.dat';

    % get the flowmeter curves
    matheson_lowflow_data = importfile(strcat(folder_name,'/lowflow_file_name));
    matheson_highflow_data = importfile(strcat(folder_name,'/highflow_file_name));
    matheson_highflow_SS_data = importfile(strcat(folder_name,'/highflow_SS_file_name));

% Since the flowmeters are calibrated for certain gases and at certain
% pressures correction factors need to be applied to measure the actual
% flow rate.
%
% Currently you need to specify the correction factors, see page 7 & 8 of
% the "Summer 2012 - " labbook for more details and the correction factors.
% Since the addition of the High Rate Flow meter, that Ibrahim used to use
% I will just include the pdf's in the dissertation.

% Matheson (Ibrahim) -> Argon
% Specific Gravity Correction Factor:
ar_sgcf = 1.0; % from Argon to Argon
% Pressure Correction Factor:
ar_pc = 1.536; % for 20 psig at tank

% Matheson -> Acetylene (H2C2) or Oxygen
% Specific Gravity Correction Factor:
h2c2_sgcf = 0.95; % from Air to Acetylene
o2_sgcf = 1.05; % from Air to Oxygen
% Pressure Correction Factor:
h2c2_pc = 1.1575; % for 5 psig at tank
o2_pc = 1.296; % for 10 psig at tank

% Want the percentage of O2 or H2C2 to be
% based on the Argon flow rate. So 10% O2
% would be mean 10mL/min for an Ar flowrate
% of 100mL/min and the total gas flow will be 110mL/min.

Ar_measurement = input_data(1); % (mm) for Ar
o2_measurement = input_data(2); % (div)
o2_percent = input_data(3); % specify value as percentage
h2c2_measurement = input_data(4); % (div)
h2c2_percent = input_data(5); % specify value as percentage

% The high flowrate data is in SLPM will use SCCM for all records -> must
% convert
SLPM_to_SCCM = 1000;
\begin{verbatim}
Ar_Flow = spline(matheson_highflow_SS_data(:,1), ...
        matheson_highflow_SS_data(:,2)*ar_sgcf*ar_pc*SLPM_to_SCCM);
O2_Flow = spline(matheson_lowflow_data(:,1),matheson_lowflow_data(:,2)*o2_sgcf*o2_pc);
H2C2_Flow = spline(matheson_lowflow_data(:,1),matheson_lowflow_data(:,2)*h2c2_sgcf*h2c2_pc);

% Warn users if both O2 and H2C2 are specified
% this is dangerous or an error
if h2c2_percent > 0 & o2_percent > 0
    msgbox('It appears that both Oxygen and Acetylene are being used this is EXPLOSIVE.','',' ...
            Flow Rate Function','warn')
end

% Acetylene
if h2c2_percent > 0
    h2c2_ideal_flowrate = (h2c2_percent/100)*ppval(Ar_measurement,Ar_Flow);
    h2c2_measurment_new = fzero(@(x) ppval(x,H2C2_Flow)-h2c2_ideal_flowrate,50);
    h2c2_flow = ppval(h2c2_measurment,H2C2_Flow);
else
    h2c2_ideal_flowrate = 0;
    h2c2_measurment_new = 0;
    h2c2_flow = 0;
end

% Oxygen
if o2_percent > 0
    o2_ideal_flowrate = (o2_percent/100)*ppval(Ar_measurement,Ar_Flow);
    o2_measurment_new = fzero(@(x) ppval(x,O2_Flow)-o2_ideal_flowrate,50);
    o2_flow = ppval(o2_measurment,O2_Flow);
else
    o2_ideal_flowrate = 0;
    o2_measurment_new = 0;
    o2_flow = 0;
end

return_data(1) = o2_measurment_new;
return_data(2) = h2c2_measurment_new;
return_data(3) = ppval(Ar_measurement,Ar_Flow);
\end{verbatim}
This MATLAB function collects the oscillograms from the LeCroy. This collection method was used to calculate the data for the long gap power calculations:

function [done_note,prologix_vers,scope_info] = LeCroy_Prologix_IV_getData_FUN(IP_ADDRESS, num_data,folder_header,file_header)

% XXXXXXXXX USER INPUT XXXXXXXXXX
% IP_ADDRESS = '192.168.1.11'; % typical value
gpib_Address = num_data(1); % typically 4;
Loop_Times = num_data(2); % typically 10;

% Write the data to a file
% folder_header = 'LeCroy_Test_Needle1-Only_';
% file_header = 'Needle-1_Test_Run-';

% XXXXXXXXX END INPUT XXXXXXXXXX

% Make Directory to Store Files
[Y, M, D, H, MN, S] = datevec ( now );
folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
[s,mess,messid]=mkdir(folder_name);
mkdir_try = 1;
while strcmp(messid,'MATLAB:MKDIR:DirectoryExists')
    folder_header = strcat(folder_header,num2str(mkdir_try),'_');
    mkdir_try = mkdir_try +1;
end

```matlab
folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
[s,mess,messid]=mkdir(folder_name);
end

%%%%%% Make a connection %%%%%%%

% Create TCP/IP object 't'. Specify server machine and port number.
t = tcpip(IP_ADDRESS, 1234);

%%%%%% SETUP The GPIB CONTROLLER %%%%%%%

% Set size of receiving buffer, if needed.
set(t, 'InputBufferSize', 1000376);
set(t, 'Timeout',30.0);

% Open connection to the server.
fopen(t);

% Request Prologix version.
fprintf(t, '++ver\n');

% Pause for the communication delay, if needed.
pause(1)

fprintf(t, '++read eoi\n');
pause(1)
% Receive lines of data from gpib controller
while (get(t, 'BytesAvailable') > 0)
t.BytesAvailable;
prologix_vers = fscanf(t);
end

% Tell Prologix to operate as a controller.
fprintf(t, '++mode 1\n');
pause(1)

% Turns off read-after-write and address the instrumentation to listen
fprintf(t, '++auto 0\n');
pause(1)

% Specified GPIB termination characters (0=CR+LF, 1=CR, 2=LF, 3=none)

163
```
fprintf(t, '++eos 0\n');
pause(1)

% Enable EDI assertion with last character
fprintf(t, '++eoi 1\n');
% Disable EDI assertion with last character
% fprintf(t, '++eoi 0\n');
pause(1)

%%%%%%%%%%%%%%%% SETUP SCOPE %%%%%%%%%%%%%%%%%

log.write("------- Setup Scope 1 -------" + "\n")
% Talk to device with GPIB address 1
fprintf(t, '+addr %d\n',gpib_Address);
pause(1)

% Inquire the scope information
fprintf(t, '*IDN?\n');
pause(1)

fprintf(t, '+read eoi\n');
pause(1)
% Receive lines of data from scope
while (get(t, 'BytesAvailable') > 0)
t.BytesAvailable;
scope_info = fscanf(t);
end

fprintf(t, 'TRIG_MODE SINGLE\n');
pause(1)

fprintf(t, 'TRIG_SELECT EDGE,SR,C2\n');
pause(1)

% Might need to use this command?
% fprintf(t, 'TRIG_DELAY \n');
% pause(1)

% Scope Settings:
fprintf(t, 'COMM_HEADER LONG\n');
pause(1)

fprintf(t, 'COMM_FORMAT DEF9,BYTE,BIN\n');
pause(1)

fprintf(t, 'SEQ OFF\n');
pause(1)

% Size that worked 50k, 100k, 250k, 500k, 1M
% 2M, 4M don't work.
fprintf(t, 'MEMORY_SIZE 1M\n');
pause(1)

fprintf(t, 'BUZZ:BEEP\n');
pause(1)
fprintf(t, 'BUZZ:BEEP\n');
pause(1)

%%%%%%%% Start getting data %%%%%%%%
for i=1:Loop_Times
    fprintf(t, 'CLEAR_SWEEPS\n');
pause(1)

    % Set the trigger to positive rising edge slope
    fprintf(t, 'C2:TRIG_SLOPE POS\n');
pause(1)
    fprintf(t, 'C2:TRIG_LEVEL 0.2\n');
pause(1)

    % Trigger the scope
    fprintf(t, '++trg %d\n',gpib_Address);
pause(1)

    % GET The Current XXX
    fprintf(t, 'C1:WAVEFORM?\n');
pause(1)

    fprintf(t, '++read eoi\n');
pause(1)
end
filename = strcat(folder_name,'/',file_header,int2str(i),'_Current-Rising','_',...
    int2str(M),'-',int2str(D),'-',int2str(Y),'.trc');
data_I = fopen(filename, 'w+');

waveform = fread(t);
count = fwrite(data_I,waveform);

%%% GET The Voltage
fprintf(t, 'C2:WAVEFORM?\n');
pause(1)

fprintf(t, '++read eoi\n');
pause(1)

filename = strcat(folder_name,'\',file_header,int2str(i),'_Voltage-Rising','_',...
    int2str(M),'-',int2str(D),'-',int2str(Y),'.trc');
data_V = fopen(filename, 'w+');

waveform = fread(t);
count = fwrite(data_V,waveform);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Set the trigger to Neg falling edge slope
% fprintf(t, 'C2:TRIG_SLOPE NEG\n');
% pause(1)
% fprintf(t, 'C2:TRIG_LEVEL -0.2\n');
% pause(1)
% % Trigger the scope
% fprintf(t, '++trg %d\n',gpib_Address);
% pause(1)
%  
% XXX GET The Current  
%
% fprintf(t, 'C1:WAVEFORM?\n');
% pause(1)
fprintf(t, '++read eoi
');
pause(1)
filename = strcat(folder_name, '\', file_header, int2str(i), '_Current-Falling', '_', ...
int2str(M), '_', int2str(D), '_', int2str(Y), '.trc');
data_I = fopen(filename, 'w+');
waveform = fread(t);
count = fwrite(data_I, waveform);

%%% GET The Voltage
fprintf(t, 'C2:WAVEFORM?
');
pause(1)
fprintf(t, '++read eoi
');
pause(1)
filename = strcat(folder_name, '\', file_header, int2str(i), '_Voltage-Falling', '_', ...
int2str(M), '_', int2str(D), '_', int2str(Y), '.trc');
data_V = fopen(filename, 'w+');
waveform = fread(t);
count = fwrite(data_V, waveform);
end

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXX Done Getting Data XXXXX

fprintf(t, 'BUZZ:BEEP
');
pause(1)
fprintf(t, 'BUZZ:BEEP
');
pause(1)
fprintf(t, 'BUZZ:BEEP
');
pause(1)
fclose(t);
delete(t);
clear t
This MATLAB function collects the oscillograms from the Tektronix 3014B scope, triggering on voltage. Did not use this much, only a couple times:

```matlab
% TekTronix_TDS3014B_GetSympatheticPulses Code for communicating with an instrument.
% This is the machine generated representation of an instrument control session using a device object. The instrument control session comprises all the steps you are likely to take when communicating with your instrument. These steps are:
% 1. Create a device object
% 2. Connect to the instrument
% 3. Configure properties
% 4. Invoke functions
% 5. Disconnect from the instrument

% To run the instrument control session, type the name of the file, untitled, at the MATLAB command prompt.
% The file, TekTronix_TDS3014B_GetSympatheticPulses.M must be on your MATLAB PATH.
% For additional information on setting your MATLAB PATH, type 'help addpath' at the MATLAB command prompt.

% Basic Use:
% This code is meant to capture 4 signals from a Tektronix TDS3014B oscilloscope. The first channel should be the total current, the second should be the voltage, the third and forth signals are the needle currents. The second channel is used for triggering, so the voltage is the trigger and the script will cycle through the sine wave a specified number of times.

% Prior to taking the current pulses the script will capture 10 profiles of the voltage for specifying the location of each current pulse on a sine wave.

% See also ICDEVICE.

% Creation time: 15-Aug-2011 20:57:37
```
function [done_note, device_info] = TekTronix_TDS3014B_GetSympatheticPulses_FUN(IP_ADDRESS, ... 
    num_data, folder_header, file_header)

%%%%%% USER INPUT %%%%%%%

% IP_ADDRESS = '192.168.1.11';
voop_num = num_data(1) % 10; % the number of full period voltage signals to take
sine_cycles = num_data(2) % 1; % the number of times to sample a sine wave

current_pulse_timebase = num_data(3) % 2e-6;

%%%%%% END INPUT %%%%%%%

% Make Directory to Store Files
[Y, M, D, H, MN, S] = datevec(now)
folder_name = strcat(folder_header, int2str(M), '-', int2str(D), '-', int2str(Y));
[s, mess, messid] = mkdir(folder_name);
mdir_try = 1;
while strcmp(messid, 'MATLAB:MKDIR:DirectoryExists')
    folder_header = strcat(folder_header, '-', num2str(mdir_try), '_');
    mdir_try = mdir_try + 1;
    folder_name = strcat(folder_header, int2str(M), '-', int2str(D), '-', int2str(Y));
    [s, mess, messid] = mkdir(folder_name);
end

% Create a VISA-TCPIP object.
interfaceObj = instrfind('Type', 'visa-tcip', 'RsrcName', strcat('TCPIP0::', ... 
    IP_ADDRESS, '::inst0::INSTR'), 'Tag', '');

% Create the VISA-TCPIP object if it does not exist
% otherwise use the object that was found.
if isempty(interfaceObj)
    interfaceObj = visa('NI', strcat('TCPIP0::', IP_ADDRESS, '::inst0::INSTR'));
else
    fclose(interfaceObj);
    interfaceObj = interfaceObj(1);
end
% Create a device object.
deviceObj = icdevice('tektronix_tds3014B.mdd', interfaceObj);

% Connect device object to hardware.
connect(deviceObj);

% Query property value(s).
device_info = get(deviceObj, 'InstrumentModel');

XXXXXXXXX GET SINE WAVE XXXXXXXXXXXXXXXX

% Configure property values for sine wave.
set(deviceObj.Channel(1), 'State', 'on');
set(deviceObj.Channel(2), 'State', 'on');
set(deviceObj.Channel(3), 'State', 'on');
set(deviceObj.Channel(4), 'State', 'on');
set(deviceObj.Channel, 'Coupling', 'dc'); %sets all channels to DC coupling
set(deviceObj.Channel, 'Scale', 2);
set(deviceObj.Acquisition(1), 'Control', 'single');
set(deviceObj.Acquisition(1), 'Timebase', 0.002);
set(deviceObj.Trigger(1), 'Source', 'channel2');
set(deviceObj.Trigger(1), 'TriggerType', 'edge');
set(deviceObj.Trigger(1), 'Slope', 'rising');
set(deviceObj.Trigger(1), 'Level', 0.5);

% Configure property values to measure RMS.
set(deviceObj.Measurement(1), 'Source', 'channel2');
set(deviceObj.Measurement(1), 'MeasurementType', 'crms');

% Query property value(s).
sine_units = get(deviceObj.Measurement(1), 'Units'); % should be volts
if sine_units ~= 'volts'
    error('getData:argChkV', 'RMS should be in volts')
end

for i=1:vloop_num
    set(deviceObj.Acquisition(1), 'State', 'run');
sine_rms = get(deviceObj.Measurement(1), 'Value');
end
groupObj = get(deviceObj, 'Waveform');
groupObj = groupObj(1);
[y,x] = invoke(groupObj, 'readwaveform', 'channel2');

if i==1,
    voltage = y;
    avg_rms = sine_rms;
else
    voltage = [voltage; y];
    avg_rms = avg_rms + sine_rms;
end

filename = strcat(folder_name,'\voltage_waveform-','_','_',int2str(M),'-',...
                  int2str(D),'-',int2str(Y),'.dat');
data = [x;y];
[nrows,ncols] = size(data);

fid = fopen(filename, 'w');
fprintf(fid, 'Time (s), Channel 2 (V) (%5.8 VRMS)\n',sine_rms);
for row=1:nrows
    fprintf(fid, '%5.10f, %5.10f\n', data(row,:));
end
fclose(fid);

time = x';
voltage = voltage';
avg_rms = avg_rms/vloop_num;

for i=1:length(voltage)
    signal(i,1) = mean(voltage(i,:));
end

f = ezfit(time,signal,'beta(rho) = beta_0 + delta * sin(rho * omega_0+phase); ...
          delta = 1.19; omega_0 = 376; beta_0 = 0.01');
clf
temp_fig = figure;
plot(time,signal,'r*');
showfig(f);
title('Average Voltage Sine Wave');
ylabel('Scope Voltage (V)');
xlabel('Time (s)');
set(temp_fig,'PaperUnits','inches','PaperPosition',[0,0,8,5]);
print(temp_fig,'-dpng','-zbuffer','-r600',strcat(folder_name,'/Avg_Voltage_1-Period.png'));

makevarfit(f)
VoltageFit = @(t) f.m(2)*sin(f.m(3)*t)+f.m(1);

% write the time and average voltage to a file
filename = strcat(folder_name,'\voltage_waveform_AVG_Fitted\',int2str(M),'\-', ...
int2str(D),'\-',int2str(Y),'.dat');
[nrows,ncols]= size(signal);
fid = fopen(filename, 'w');
fprintf(fid, 'Time (s), AVG Measured Voltage (V), Fitted Voltage
');
fprintf(fid, 'Fit Parameters: Amplitude = %5.5f, Angular Freq = %5.5f, ...
Phase Shift = %5.5f, DC Offset = %5.5f\n\n', f.m(2),f.m(3),f.m(4),f.m(1));
for row=1:nrows
fprintf(fid, '%5.10f, %5.10f, %5.10f\n', time(row),signal(row),VoltageFit(time(row)));
end
fclose(fid);

% configure property values for acquiring.
% expect there will be about 10,000 points and will use 7,000

% Set the channel voltage scale
set(deviceObj.Channel(1), 'Scale', 0.005);
set(deviceObj.Channel(2), 'Scale', 0.5);
set(deviceObj.Channel(3), 'Scale', 0.005);
set(deviceObj.Channel(4), 'Scale', 0.005);

% set the timebase and get data sample to build
% trigger array
set(deviceObj.Acquisition(1), 'Timebase', current_pulse_timebase);
set(deviceObj.Trigger(1), 'Source', 'channel2');
set(deviceObj.Trigger(1), 'TriggerType', 'edge');
set(deviceObj.Trigger(1), 'Slope', 'rising');
set(deviceObj.Trigger(1), 'Level', 1)

set(deviceObj.Acquisition(1), 'State', 'run');
groupObj = get(deviceObj, 'Waveform');
groupObj = groupObj(1);
[y,x] = invoke(groupObj, 'readwaveform', 'channel2');

period = (2*pi)/f.m(3);
temp_time = 0.0;
trigger_value = 0.01; % V
i=1;
tstep = 10e-9;
while ((temp_time < period) | (trigger_value < 0.005))
    if (((VoltageFit(temp_time)<trigger_value) & (VoltageFit(temp_time+tstep)>trigger_value))
        trigger_array(i,:)=[temp_time, trigger_value];
        trigger_value = trigger_value + 0.01;
        slope = 'rising ';
        slope_array(i,:)= slope;
        i = i+1;
    elseif (((VoltageFit(temp_time)>trigger_value) & (VoltageFit(temp_time+tstep)<trigger_value))
        trigger_array(i,:)=[temp_time, trigger_value];
        trigger_value = trigger_value - 0.01;
        slope = 'falling';
        slope_array(i,:)= slope;
        i = i+1;
    end

    if trigger_value > (f.m(2)+f.m(1))
        trigger_value = trigger_value -0.01;
    elseif trigger_value < -(f.m(2)+f.m(1))
        trigger_value = trigger_value + 0.01;
    end

    temp_time= temp_time+tstep;
end

filename = strcat(folder_name,'\Trigger_Data_','int2str(M),'-','int2str(D),'-','int2str(Y)','.dat');
[nrows, ncols] = size(trigger_array);

fid = fopen(filename, 'w');

fprintf(fid, 'Trigger Time (s), Trigger Voltage (V), Slope
');

for row = 1:nrows
    fprintf(fid, '%5.10f, %5.10f, %s
', trigger_array(row,:), slope_array(row,:));
end

fclose(fid);

%%%%%%%%%% GET Current Pulses %%%%%%%%%%%%%%%

for i = 1:sine_cycles
    for j = 1:length(trigger_array)
        if slope_array(j,:) == 'rising '
            set(deviceObj.Trigger(1), 'Slope', 'rising');
        else
            set(deviceObj.Trigger(1), 'Slope', 'falling');
        end

        set(deviceObj.Trigger(1), 'Level', trigger_array(j,2))
        set(deviceObj.Acquisition(1), 'State', 'run');

        groupObj = get(deviceObj, 'Waveform');
        groupObj = groupObj(1);
        [y1,x1] = invoke(groupObj, 'readwaveform', 'channel1');
        % should be the total current

        groupObj = get(deviceObj, 'Waveform');
        groupObj = groupObj(1);
        [y2,x2] = invoke(groupObj, 'readwaveform', 'channel2');
        % should be the voltage

        groupObj = get(deviceObj, 'Waveform');
        groupObj = groupObj(1);
        [y3,x3] = invoke(groupObj, 'readwaveform', 'channel3');
        % should be the current in a single needle

        groupObj = get(deviceObj, 'Waveform');
groupObj = groupObj(1);
[y4,x4] = invoke(groupObj, 'readwaveform', 'channel4');
% should be a current in a single needle

filename = strcat(folder_name,'\',file_header,int2str(j),'_', ... 
          int2str(M),'-',int2str(D),'-',int2str(Y),'.dat');

data = [x1;x1+trigger_array(j,1);y1;y2;y3;y4]';
[nrows,ncols]= size(data);

fid = fopen(filename, 'w');
fprintf(fid, 'Scope Time (s), Sine Time (s), Channel 1 (Amps), ... 
        Channel 2 (V), Channel 3 (Amps), Channel 4 (Amps)\n');

for row=1:nrows
    fprintf(fid, '%5.10f, %5.10f, %5.10f, %5.10f, %5.10f, %5.10f
', data(row,:));
end

close(fid);
end

done_note = 'Data Collection Script Done'

% Disconnect device object from hardware.
disconnect(deviceObj);

% Delete objects.
delete([deviceObj interfaceObj]);

done_note = 'Data Collection Script Done'

This MATLAB function collects the oscillograms from the Tektronix 3014B scope, triggering on current. This method was used to collect all the corona pulses used in the synchronicity study:

% TekTronix_TDS3014B_GetSympatheticPulses Code for communicating with an instrument.
% 
% This is the machine generated representation of an instrument control
% session using a device object. The instrument control session comprises
% all the steps you are likely to take when communicating with your
% instrument. These steps are:
%
% 1. Create a device object
% 2. Connect to the instrument
% 3. Configure properties
% 4. Invoke functions
% 5. Disconnect from the instrument
%
% To run the instrument control session, type the name of the file,
% untitled, at the MATLAB command prompt.
%
% The file, TekTronix_TDS3014B_GetSympatheticPulses.m must be on your MATLAB PATH.
% For additional information on setting your MATLAB PATH, type 'help addpath' at the MATLAB
% command
% prompt.
%
% Basic Use:
% This code is meant to capture 4 signals from a TekTronix TDS3014B oscilloscope.
% The first channel should be the total current, the second should be the voltage,
% the third and forth signals are the needle currents. The second channel is used
% for triggering, so the voltage is the trigger and the script will cycle through
% the sine wave a specified number of times.
%
% Prior to taking the current pulses the script will capture 10 profiles of the voltage
% for specifying the location of each current pulse on a sine wave.
%
% See also ICDEVICE.
%
% Creation time: 15-Aug-2011 20:57:37
function [done_note,device_info] = TekTronix_TDS3014B_Get_I_pulses_trigger_on_I_FUN(IP_ADDRESS, ...
    num_data,folder_header,file_header_main)

% USER INPUT

IP_ADDRESS = '192.168.1.11';

vloop_num = num_data(1); %10;  % number of voltage profiles to average
sample_num = num_data(2); % 100;

% This number of positive/negative triggered samples (total # samples 2*sample_num)
current_pulse_timebase = num_data(3); \% 1e-6;
current_pulse_trigger = num_data(4); \% 0.01;
current_pules_div = num_data(5); \% 10mV

% Write the data to a file
folder_header = 'Sympathetic_Data_Current_Test-5_';
file_header_main = 'Needle_Test_';

%% END INPUT

% Make Directory to Store Files
[Y, M, D, H, MN, S] = datevec(now);
folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
[s,mess,messid]=mkdir(folder_name);
mdir_try = 1;
while strcmp(messid,'MATLAB:MKDIR:DirectoryExists')
    folder_header = strcat(folder_header,'-',num2str(mdir_try),'_');
    mdir_try = mdir_try +1;
    folder_name = strcat(folder_header,int2str(M),'-',int2str(D),'-',int2str(Y));
    [s,mess,messid]=mkdir(folder_name);
end

% Create a VISA-TCPIP object.
interfaceObj = instrfind('Type', 'visa-tcpip', 'RsrcName', strcat('TCPIP0::',...
    IP_ADDRESS,'::inst0::INSTR'), 'Tag', '');

% Create the VISA-TCPIP object if it does not exist
% otherwise use the object that was found.
if isempty(interfaceObj)
    interfaceObj = visa('NI', strcat('TCPIPO::',IP_ADDRESS,'::inst0::INSTR'));
else
    fclose(interfaceObj);
    interfaceObj = interfaceObj(1);
end

% Create a device object.
deviceObj = icdevice('tektronix_tds3014B.mdd', interfaceObj);

% Connect device object to hardware.
connect(deviceObj);
% Query property value(s).
device_info = get(deviceObj, 'InstrumentModel');

% Configure property values for sine wave.
set(deviceObj.Channel(1), 'State', 'on');
set(deviceObj.Channel(2), 'State', 'on');
set(deviceObj.Channel(3), 'State', 'on');
set(deviceObj.Channel(4), 'State', 'on');
set(deviceObj.Acquisition, 'Delay', 0.5e-6);
set(deviceObj.Channel, 'BandwidthLimit', 20.0); % Limit the Bandwidth 20MHz
set(deviceObj.Channel, 'Coupling', 'dc'); % sets all channels to DC coupling
set(deviceObj.Channel, 'Scale', 2);
set(deviceObj.Acquisition(1), 'Control', 'single');
set(deviceObj.Acquisition(1), 'Timebase', 0.002);
set(deviceObj.Trigger(1), 'Source', 'channel2');
set(deviceObj.Trigger(1), 'TriggerType', 'edge');
set(deviceObj.Trigger(1), 'Slope', 'rising');
set(deviceObj.Trigger(1), 'Level', 0.5);

% Configure property values to measure RMS.
set(deviceObj.Measurement(1), 'Source', 'channel2');
set(deviceObj.Measurement(1), 'MeasurementType', 'crms');

% Query property value(s).
sine_units = get(deviceObj.Measurement(1), 'Units'); % should be volts
if sine_units ~= 'volts'
    error('getData:argChkV', 'RMS should be in volts')
end

i=1;
while i < vloop_num+1
    set(deviceObj.Acquisition(1), 'State', 'run');
sine_rms = get(deviceObj.Measurement(1), 'Value');

    groupObj = get(deviceObj, 'Waveform');
    groupObj = groupObj(1);
    [y,x] = invoke(groupObj, 'readwaveform', 'channel2');
if ((min(y)==max(y)) | (min(x)==max(x)))
    fprintf(1,'Bad data capture trying again (run = %d)',i)
else
    if i==1,
        voltage = y;
        avg_rms = sine_rms;
    else
        voltage = [voltage ; y];
        avg_rms = avg_rms + sine_rms;
    end

filename = strcat(folder_name,'/voltage_waveform-',int2str(i),'_',int2str(M),'-',...
    int2str(D),'_',int2str(Y),'.dat');
data = [x;y]';
[nrows,ncols]= size(data);

fid = fopen(filename, 'w');
fprintf(fid, 'Time (s), Channel 2 (V), (%5.8 VRMS)
',sine_rms);
for row=1:nrows
    fprintf(fid, '%5.10f, %5.10f
', data(row,:));
end
fclose(fid);
i = i+1;
end
time = x';
voltage = voltage';
avg_rms = avg_rms/vloop_num;

for i=1:length(voltage)
    signal(i,1) = mean(voltage(i,:));
end

f = ezfit(time,signal,'beta(rho) = beta_0 + delta * sin(rho * omega_0+phase); delta = 1.19; ...
\[ \omega_0 = 376; \beta_0 = 0.01 \];

cf

tmp_fig = figure;
plot(time,signal,'r*');
showfit(f);
title('Average Voltage Sine Wave');
ylabel('Scope Voltage (V)');
xlabel('Time (s)');
set(tmp_fig,'PaperUnits','inches','PaperPosition',[0,0,8,6]);
print(tmp_fig,'-dpng','-zbuffer','-r600',strcat(folder_name,'/Avg_Voltage_1-Period.png'));

makevarfit(f)
VoltageFit = @(t) f.m(2)*sin(f.m(3)*t)+f.m(1);

% write the time and average voltage to a file
filename = strcat(folder_name,'\voltage_waveform_AVG_Fitted_',int2str(M),'-',int2str(D),'-',...
int2str(Y),'.dat');
[nrows,ncols]= size(signal);
fid = fopen(filename, 'w');
fprintf(fid, 'Time (s), AVG Measured Voltage (V), Fitted Voltage
');
fprintf(fid, 'Fit Parameters: Amplitude = %5.5f, Angular Freq = %5.5f, Phase Shift = %5.5f, ...
DC Offset = %5.5f
', f.m(2),f.m(3),f.m(4),f.m(1));
for row=1:nrows
fprintf(fid, '%5.10f, %5.10f, %5.10f
', time(row),signal(row),VoltageFit(time(row)));
end
fclose(fid);

% GET TRIGGER ARRAY
% Configure property values for acquiring.
% expect there will be about 10,000 points and will use 7,000

% Set the channel voltage scale
set(deviceObj.Channel(1), 'Scale', current_pules_div);
set(deviceObj.Channel(2), 'Scale', 2);
set(deviceObj.Channel(3), 'Scale', current_pules_div);
set(deviceObj.Channel(4), 'Scale', current_pules_div);

% set the timebase and get data sample to build
% trigger array
set(deviceObj.Acquisition(1), 'Timebase', current_pulse_timebase);
set(deviceObj.Trigger(1), 'Source', 'channel1');
set(deviceObj.Trigger(1), 'TriggerType', 'edge');
set(deviceObj.Trigger(1), 'Level', current_pulse_trigger)

if current_pulse_trigger > 0
    file_header = strcat(file_header_main,'Pos-trig_Run-');
    set(deviceObj.Trigger(1), 'Slope', 'rising');
else
    file_header = strcat(file_header_main,'Neg-trig_Run-');
    set(deviceObj.Trigger(1), 'Slope', 'falling');
end

%---------------- GET Current Pulses -----------------

j = 1;
while j<sample_num+1
    set(deviceObj.Acquisition(1), 'State', 'run');

    groupObj = get(deviceObj, 'Waveform');
    groupObj = groupObj(1);
    [y1,x1] = invoke(groupObj, 'readwaveform', 'channel1'); % should be the total current

    groupObj = get(deviceObj, 'Waveform');
    groupObj = groupObj(1);
    [y2,x2] = invoke(groupObj, 'readwaveform', 'channel2'); % should be the voltage

    groupObj = get(deviceObj, 'Waveform');
    groupObj = groupObj(1);
    [y3,x3] = invoke(groupObj, 'readwaveform', 'channel3');
    % should be the current in a single needle

    groupObj = get(deviceObj, 'Waveform');
    groupObj = groupObj(1);
    [y4,x4] = invoke(groupObj, 'readwaveform', 'channel4');
    % should be a current in a single needle

    if ((min(x1)==max(x1)) | (min(y1)==max(y1)) | (min(y2)==max(y2)) | ...
        (min(y3)==max(y3)) | (min(y4)==max(y4)))
        fprintf(1,'Bad data caputre trying again (run = %d'),j)
    end

else
    filename = strcat(folder_name,'/',file_header,int2str(j),'_',int2str(M),'-',...
    int2str(D),'-',int2str(Y),'.dat');

    data = [x1;y1;y2;y3;y4]';
    [nrows,ncols]= size(data);

    fid = fopen(filename, 'w');
    fprintf(fid, '%s
',timestring());
    fprintf(fid, 'Scope Time (s), Channel 1 (Amps), Channel 2 (V), ... 
    Channel 3 (Amps), Channel 4 (Amps)\n');

    for row=1:nrows
        fprintf(fid, '%5.10f, %5.10f, %5.10f, %5.10f, %5.10f
', data(row,:));
    end

    fclose(fid);
    j=j+1;
end

% Disconnect device object from hardware.
disconnect(deviceObj);

% Delete objects.
delete([deviceObj interfaceObj]);

done_note = 'Data Collection Script Done';

D.1 Synchronicity Sorting Script

The following MATLAB script was used to sort and count the corona pulses:

% Erik Wemlinger
% Created: 8/15/2011
% Use: This script was written to plot and look at the possibility of
% sympathetic current pulses. The script expects a total current, voltage,
% current through needle 1 and current through some other needle
folder_name = 'Sym_N1-N20_T2_20MHz_10-25-2011';

folder_name = char(import_folder_list('folder_names.txt'));

folder_name = char(import_folder_list('folder_names.txt'));

names = {'Sym_Pulses_N1-N3_3-3-2012'; ... 
'Sym_Pulses_N1-N6_3-3-2012'; ... 
'Sym_Pulses_N1-N11_3-3-2012'; ... 
'Sym_Pulses_N1-N12_3-3-2012'; ... 
'Sym_Pulses_N1-N14_3-3-2012'; ... 
'Sym_Pulses_N1-N16_3-3-2012'; ... 
'Sym_Pulses_N1-N20_3-2-2012'; ... 
'Sym_Pulses_N1-N20-END_3-3-2012'; ... 
'Sym_Pulses_N1-N21_3-3-2012'; ... 
'Sym_Pulses_N1-N2_3-3-2012'; ... 
'Sym_Pulses_N1-N5_3-3-2012'; ... 
'Sym_Pulses_N1-N7_3-2-2012'; ... 
'Sym_Pulses_N1-Only_3-2-2012'; ... 
'Needle-1_Needle-20_Set-1A_2-18-2012'; ... 
'Needle-1_Needle-20_Set-1B_2-18-2012'; ... 
'Needle-1_Needle-20_Set-1C_2-18-2012'; ... 
'Needle-1_Needle-20_Set-1D_2-18-2012'; ... 
'Needle-1_Needle-20_Set-1E_2-18-2012'; ... 
'Needle-1_Needle-20_Set-2A_2-18-2012'; ... 
'Needle-1_Needle-20_Set-2B_2-18-2012'; ... 
'Needle-1_Needle-20_Set-2C_2-18-2012'; ... 
'Needle-1_Needle-20_Set-2D_2-18-2012'; ... 
'Needle-1_Needle-20_Set-2E_2-18-2012'; ...};

'LOS_N1-N20_4-14-2012'; ...

'LOS_N1-N20_Glass_4-14-2012'; ...

'LOS_N1-N20_Plastic-Opaque_4-14-2012'; ...

'../Acetylene_Data/Hybrid_Run2-4_Vertical_ab_4-3-2012'; ...

'../Acetylene_Data/Hybrid_Run2-4_Vertical_cd_4-3-2012'; ...

'../Acetylene_Data/Hybrid_Run2_4_3-31-2012';

distance = [5.987,4.6,4.65,5.95,6.466,3.0,1.866,1.866,4.128,4.79,2.465,2.47,0,1.866, ... 
1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866,1.866, ... 
1.866,1.866,1.866]; % (cm)
% For the radius of curvature the radii are for the second needle, N3, N6,
% ... the last is needle 1 N1
rc = [49.6,47.4,43.4.1,41.5,51.1,51.1,42.5,42.5,50.3,55.35.5,42.2,41.5,42.5,42.5,42.5,42.5, ... 
42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5,42.5]; % (um)

% Write a result summary
filename = 'Summary_LOG.dat';
runlog = fopen(filename,'w');

fprintf(runlog, '------- Log Summary -------
');
fprintf(runlog, 'Erik Wemlinger
');
fprintf(runlog, 'Created at: %s
', timestring());
fprintf(runlog, '------- >>> ||||||||||||||| <<< -------
');

% Want to write out the percentage as a function of distance
% for the html page
percent_distance = [];
name_pair = [];
for i=1:length(folder_name(:,1))

% get the name of the files in the folder
files = dir(strcat(folder_name(i,:),'/needle_*Pos*.dat'));

% Want to place the results in a subfolder
mkdir(strcat(folder_name(i,:),'/Images'));
mkdir(strcat(folder_name(i,:),'/Images/Needle1'));
mkdir(strcat(folder_name(i,:),'/Images/Needle2'));
mkdir(strcat(folder_name(i,:),'/Images/Both_Needles'));
mkdir(strcat(folder_name(i,:),'/Images/Intermediate'));
mkdir(strcat(folder_name(i,:),'/Images/More_Pulses'));

delete(strcat(folder_name(i,:),'/Images/*.png'));
delete(strcat(folder_name(i,:),'/Images/Needle1/*.png'));
delete(strcat(folder_name(i,:),'/Images/Needle2/*.png'));
delete(strcat(folder_name(i,:),'/Images/Both_Needles/*.png'));
delete(strcat(folder_name(i,:),'/Images/Intermediate/*.png'));
delete(strcat(folder_name(i,:),'/Images/More_Pulses/*.png'));

184
glitch_data = 0;
not_counted = 0;
ch3_and_ch4_pulse = 0;
ch3_pulse = 0;
ch4_pulse = 0;
ch3_ch4_and_others_pulse = 0;
data_index = 1;
total_num_pulses = length(files);
pulseNUM = [];
comp = [];
V_maxI = [];
Run_NAME = [];
charge_data = [];
delta_Q = [];

% Write the DATA file
filename = strcat(folder_name(i,:),'/Names_of_Not_Counted_Pulses.dat');
bad_fid = fopen(filename, 'w');
fprintf(bad_fid, 'Created: %s\n', timestring());

for j=1:length(files)
    % Note that for new files there are 2 header lines while for the
    % old data (before the middle of April there is only one. The extra
    % header line is a born on timestamp
    M = dlmread(strcat(folder_name(i,:),'/',files(j).name),',',1,0);
    M = dlmread(strcat(folder_name(i,:),'/',files(j).name),',',2,0);

    %%% Define some array sizes %%%
    [row,col] = size(M);

    [C,I] = max(abs(M(:,2)));
    % Voltage AVG
    voltageAVG = mean(M(:,3));

    % Checking to make sure the signals are good, this should
if ((min(M(:,1))==max(M(:,1))) || (min(M(:,2))==max(M(:,2))) || ...
    (min(M(:,3))==max(M(:,3))) || (min(M(:,4))==max(M(:,4))) || (min(M(:,5))==max(M(:,5)))
    glitch_data = glitch_data+1;
    fprintf(bad_fid,'Glitch - Ch %d - %s
', strcat(files(j).name(1:end-15)));
else if ((voltageAVG < 0) && (M(I,2) > 0))
    display('Positive current, negative voltage');
else if ((voltageAVG > 0) && (M(I,2) > 0))

    time = M(:,1); % Tek Scope time
    totI = M(:,2); % Channel 1 signal (total current)
    voltage = M(:,3); % Channel 2 signal (Voltage)
    needle1 = M(:,4); % Channel 3 (typically current through needle 1)
    needle2 = M(:,5); % Channel 4 (typically current through the other needle)

    % Going to do a bit of signal processing before comparing
    % signals. (1) going to Savitzky-Golay filtering, then (2)
    % going to apply a DC offset using the first 50 elements.

    % consider the dual as the sum of the two signals
    dual_needle = needle1+needle2;

    %
    % totI = sgolayfilt(totI, 3,51);
    % voltage = sgolayfilt(voltage, 3,51);
    % needle1 = sgolayfilt(needle1, 3,51);
    % needle2 = sgolayfilt(needle2, 3,51);
    % dual_needle = sgolayfilt(dual_needle, 3,51);

    % DC Correct
    totI = totI - mean(totI(1:50));
    dual_needle = dual_needle - mean(dual_needle(1:50));
    needle1 = needle1 - mean(needle1(1:50));
    needle2 = needle2 - mean(needle2(1:50));

    ymin = min([totI;needle1;needle2;dual_needle]);
    if ymin < 0
        ymin = ymin + ymin*0.2;

186
else
    ymin = ymin - ymin*0.2;
end
ymax = max([totI;needle1;needle2;dual_needle]);
ymax = ymax + ymax*0.2;

% convert to mA
ymax = ymax*1e3;
ymin = ymin*1e3;

% Calculate the correlation
rD = corr2(totI,dual_needle);

% Calculate the correlation Total and Needle 1
rN1 = corr2(totI,needle1);

% Calculate the correlation Total and Needle 2
rN2 = corr2(totI,needle2);

% Calculate the max current on each needle
[C,I] = max(abs(totI));
[C1,I1] = max(abs(needle1));
[C2,I2] = max(abs(needle2));
[CD,ID] = max(abs(dual_needle));

% Voltage AVG
voltageAVG = mean(voltage);

v_maxI(data_index,:) = [voltageAVG,totI(I)];
Run_NAME(data_index) = j;

% Calculate the charge measured by each Pearson
charge_data(data_index,:) = [trapz(time,totI),trapz(time,needle1),...
trapz(time,needle2)]; % (C)
delta_Q(data_index) = charge_data(data_index,1)-trapz(time,dual_needle);
% Record down the correlation or RMSE
comp(data_index,:) = [rN1, rN2, rD, rD, rD];

charge_std = 0.5538e-9;
pulse_floor = 2.0e-3;
pulse_trigger = 10.0e-3;

% Clearly a Needle 1 Pulse
if (rN1 >= rD) && (rN2 < rD) && (delta_Q(data_index) <= 0) && (C2 < pulse_floor)
    ch3_pulse = ch3_pulse +1;
pulseNUM(data_index) = 1;
end

% Clearly a Needle 2 Pulse
elseif (rN1 < rD) && (rN2 >= rD) && (delta_Q(data_index) <= 0) && (C1 < pulse_floor)
    ch4_pulse = ch4_pulse +1;
pulseNUM(data_index) = 2;
end

% Clearly a Needle 1 and Needle 2 Pulse
elseif (rN1 <= rD) && (rN2 <= rD) && (C1 > pulse_floor) && (C2 > pulse_floor) ... && (delta_Q(data_index) <= 0)
    ch3_and_ch4_pulse = ch3_and_ch4_pulse +1;
pulseNUM(data_index) = 3;
end

% Most likely Needle 1 Pulse
else
    % Initial conditions for Needle 1 Pulse
    ch3_pulse = 1;
pulseNUM(data_index) = 1;
end

% Clearly a Needle 2 Pulse
elseif (rN2 < rD) && (delta_Q(data_index) <= 0) && (C1 > 0.65*pulse_trigger) ... && (C2 < pulse_floor)
    ch4_pulse = 1;
pulseNUM(data_index) = 2;
end
ch3_pulse = ch3_pulse +1;
pulseNUM(data_index) = 1;
needle_pulse(data_index,:) = [1,0,0];
plot_name = strcat(folder_name(i,:),'/Images/Needle1/', ...
  files(j).name(1:end-3),'png');
New_title = ['Needle 1 Must be!, (dQ ',num2str(delta_Q(data_index)*1e9),' ...
  (nC)) -> Corr = ',num2str(rD)];

% Most likely Needle 2 Pulse
elseif (rN1 < rD) && (delta_Q(data_index) <= 0) && (C2 > 0.65*pulse_trigger) ...
  && (C1 < pulse_floor)
    ch4_pulse = ch4_pulse +1;
pulseNUM(data_index) = 2;
needle_pulse(data_index,:) = [0,1,0];
plot_name = strcat(folder_name(i,:),'/Images/Needle2/', ...
  files(j).name(1:end-3),'png');
New_title = ['Needle 2 Must be!, (dQ ',num2str(delta_Q(data_index)*1e9),' ...
  (nC)) -> Corr = ',num2str(rD)];

% Negative Charge Difference so it must be a dual
elseif (rD >= 0.9 ) && (C1 > pulse_floor) && (C2 > pulse_floor) ...
  && (delta_Q(data_index)*1e9 <= charge_std*5*1e9)
    ch3_and_ch4_pulse = ch3_and_ch4_pulse +1;
pulseNUM(data_index) = 3;
needle_pulse(data_index,:) = [0,0,1];
plot_name = strcat(folder_name(i,:),'/Images/Both_Needles/', ...
  files(j).name(1:end-3),'png');
New_title = ['Dual Must be!, (dQ ',num2str(delta_Q(data_index)*1e9),' ...
  (nC)) -> Corr = ',num2str(rD)];
else
    pulseNUM(data_index) = 4;
    needle_pulse(data_index,:) = [0,0,0];
    ch3_ch4_and_others_pulse = ch3_ch4_and_others_pulse + 1;
    plot_name = strcat(folder_name(i,:),'/Images/Intermediate/', ...
      files(j).name(1:end-3),'png');
    New_title = ['Intermediate Pulses, (dQ ',num2str(delta_Q(data_index)*1e9),' ...
      (nC)) -> Corr = ',num2str(rD)];

% More pulses
else
    needle_pulse(data_index,:) = [0,0,0];
pulseNUM(data_index) = 5;
ch3_ch4_and_others_pulse = ch3_ch4_and_others_pulse + 1;
plot_name = strcat(folder_name(i,:),'/Images/More_Pulses/', ... files(j).name(1:end-3),'png');
New_title = [ 'More Pulses, (dQ ',num2str(delta_Q(data_index)*1e9),' ... (nC)) -> Corr = ',num2str(rD)];
end

% Save the figure
plot_FFT_name = strcat(folder_name(i,:),'/Images/', ... files(j).name(1:end-4),'_FFT','.png');
plot_pulse_compare(time, totI, voltage, needle1, needle2, plot_name);
plot_pulse_compare_filtering(time, totI, voltage, needle1, needle2, plot_name);

% Start plotting
clf
subplot(411)
c1 = totI;
temp = find(abs(c1) < 0.002);
c1(temp) = c1(temp)*0.0;
% charge = cumtrapz(time,c1);
% temp2 = find(abs(c1) > 0);
% c1(temp2) = 1e9;
% resetCHARGE = charge.*c1;
sum = c1*0.0;
rollingSUM = 0.0;
for k=2:length(c1)
if c1(k) > 0
    rollingSUM = rollingSUM + 0.5*(time(k)-time(k-1))*(c1(k-1)+c1(k));
    sum(k) = rollingSUM;
elseif (c1(k) == 0) && (c1(k-1) ==0)
    rollingSUM = 0;
end
end
% plot(time*1e3,sum,'r')
[AX,H1,H2] = plotyy(time*1e6,voltage*4.105,time*1e6,sum*1e9,'plot');
set(get(AX(2),'Ylabel'),'String','Charge (nC)')
set(get(AX(1),'Ylabel'),'String','Volts (kV)')
set(get(AX(1),'Xlabel'),'String','Time (us)')
title('Total Pulse Charge and Voltage')
% Compare total current to the current through needle 1

New_title = ['Correlation = ',num2str(rN1)];

subplot(413)
plot(time*1e6,totI*1e3,'r.')
hold on
plot(time*1e6,needle1*1e3,'g.')
hold off
ylabel('Current (mA)');
xlabel('Time (us)');
ylim([ymin ymax])
title(New_title);
% legend('Total Current','N1+N1','TotI Filtered','N1+N1 Filtered')
legend('TotI Filtered','N1 Filtered')

% Compare total current to the current through needle 2

New_title = ['Correlation = ',num2str(rN2)];

subplot(414)
plot(time*1e6,totI*1e3,'r.')
hold on
plot(time*1e6,needle2*1e3,'b.')
hold off
ylabel('Current (mA)');
xlabel('Time (us)');
ylim([ymin ymax])
title(New_title);

% legend('Total Current','N1+Ni','TotI Filtered','N1+Ni Filtered')
legend('TotI Filtered','N2 Filtered')

% Save the figure
print('-dpng','-zbuffer',plot_name);

% Done Plotting

data_index = data_index+1;

else
not_counted = not_counted+1;
fprintf(bad_fid,'NOT Counted - %s
', strcat(files(j).name(1:end-15)));
end
end
fclose(bad_fid); % close the list of files which are not counted

% Write the DATA file
filename = strcat(folder_name(i,:),'/Current_Pulse_DATA','.csv');

fid = fopen(filename, 'w');
fprintf(fid, 'Created: %s
', timestring());
fprintf(fid, 'Summary of the current pulses in this folder
');
fprintf(fid, ' -Note: That the avg voltage as been scaled to actual value ...
 based on the voltage divider\n');
fprintf(fid, 'Total Number of files = %d\n',total_num_pulses);
fprintf(fid, 'Total Number of files with data = %d\n',total_num_pulses-glitch_data);
fprintf(fid, 'Current Pulses not counted = %d, Glitch DATA (not counted) = %d', ... not_counted,glitch_data);
fprintf(fid, 'Number of dual pulses = %d\n',ch3_and_ch4_pulse);
fprintf(fid, 'Number of single pulses (Ch3) = %d\n',ch3_pulse);
fprintf(fid, 'Number of single pulses (Ch4) = %d\n',ch4_pulse);
fprintf(fid, 'Number of pulses not measured by ch3 and ch4 = %d\n', ch3_ch4_and_others_pulse);
fprintf(fid, 'Run #, Needle 1 Pulse, Needle 2 Pulse, Needle 1 and 2 Pulse, ...
Total Pearson Charge (C), Needle 1 Charge (C), Needle 2 Charge (C), AVG Voltage (V), ...
Peak Total Current (A), Correlation N1+N2, Correlation N1, Correlation N2 \n');

for row=1:data_index-2
  Run_NAME_temp = strcat(files(Run_NAME(row)).name(1:end-3));
  fprintf(fid, '%s, %i, %i, %i, %8.8e, %8.8e, %8.8e, %8.8e, %8.8e, %8.8e\n', ...
    Run_NAME_temp, needle_pulse(row,:), charge_data(row,:), v_maxI(row,1)*4105, ...
    v_maxI(row,2),comp(row,:));
end
fclose(fid);

% Write the DATA file to be included in html file
% code to look at the current pulses from the max V vs max I and RMSE vs Delta Q
filename = strcat(folder_name(i,:),'/For_HTML','.data');

fid = fopen(filename, 'w');
%fprintf(fid, 'Run #\n');
fprintf(fid, 'var names=[');
for row=1:data_index-2
  switch pulseNUM(row)
    case 1
      Run_NAME_temp = strcat('../',folder_name(i,:),'/Images/Needle1/', ...
        files(Run_NAME(row)).name(1:end-3),'png');
    case 2
      Run_NAME_temp = strcat('../',folder_name(i,:),'/Images/Needle2/', ...
        files(Run_NAME(row)).name(1:end-3),'png');
    case 3
      Run_NAME_temp = strcat('../',folder_name(i,:),'/Images/Both_Needles/', ...
        files(Run_NAME(row)).name(1:end-3),'png');
    case 4
      Run_NAME_temp = strcat('../',folder_name(i,:),'/Images/Intermediate/', ...
        files(Run_NAME(row)).name(1:end-3),'png');
    case 5
      Run_NAME_temp = strcat('../',folder_name(i,:),'/Images/More_Pulses/', ...
        files(Run_NAME(row)).name(1:end-3),'png');
  end
end
%fprintf(fid, ']');
fclose(fid);
otherwise
    Run_NAME_temp = strcat('../', folder_name(i,:), '/Images/', ...
    files(Run_NAME(row)).name(1:end-3), 'png');
end

fprintf(fid, "%.s\n", Run_NAME_temp);
end
if data_index == 1
    row = 1;
    fprintf(fid, '];
');
else
    row = row+1;
    Run_NAME_temp = strcat('../', folder_name(i,:), '/Images/', ...
    files(Run_NAME(row)).name(1:end-3), 'png');
    fprintf(fid, "%.s\n", Run_NAME_temp);
end

fprintf(fid, 'AVG Voltage (V), Peak Total Current (A)\n');
fprintf(fid, 'var voltI=[');
for row=1:data_index-2
    fprintf(fid, '[%.8e, %.8e],
', v_maxI(row,1)*4.105, v_maxI(row,2)*1e3);
end
if data_index == 1
    row = 1;
    fprintf(fid, '];
');
else
    row = row+1;
    fprintf(fid, '[%.8e, %.8e]
', v_maxI(row,1)*4.105, v_maxI(row,2)*1e3);
end
 fprintf(fid, 'var Correlation=[');
for row=1:data_index-2
    fprintf(fid, '[%.8e, %.8e],
', pulseNUM(row), comp(row,pulseNUM(row)));
end
if data_index == 1
    row = 1;
    fprintf(fid, '];
');
else
    row = row+1;
end
fprintf(fid, ' [%8.8e, %8.8e]);
n', pulseNUM(row), comp(row, pulseNUM(row)));
end
fclose(fid);

if isempty(v_maxI)
    display('Skip Plotting, no vmax_I');
else
    
    subplot(111)
    plot(v_maxI(:,1)*4105/1e3, v_maxI(:,2)*1e3,'r. ')
    title('Voltage vs Max Current');
    ylabel('Current (mA)');
    xlabel('Voltage (kV) ');

    print('-dpng','-zbuffer','-r600',strcat(folder_name(i,:), ... 
    '/Current_Pulse_Polarity_Check.png'));
    print('-dpng','-zbuffer',strcat(folder_name(i,:), ... 
    '/Images/Current_Pulse_Polarity_Check.png'));
    clf

    subplot(111)
    plot(pulseNUM, comp,'r. ')
    title('PulseNum vs Correlation');
    ylabel('RSME');
    xlabel('Number of Pulses (1-n1, 2-n?, 3-n1 and n2, 4-n_more');

    print('-dpng','-zbuffer',strcat(folder_name(i,:), '/RSME.png'));
end

% Write to log
% get the name of the files in the folder
filename = dir(strcat(folder_name(i,:),'/log_*.dat'));
[Y, MO, D, H, MN, S] = datevec ( now );

if isempty(filename)
    filename = strcat(folder_name(i,:),'/log', '_', int2str(MO), '-', int2str(D), '-', ... 
    int2str(Y),'.dat');
    fid = fopen(filename,'w');
else
    fid = fopen(strcat(directory{i}, '/', filename.name), 'a');
end

fprintf(fid, '-------- Pulse Polarity Data Appended --------
');
fprintf(fid, 'Erik Wemlinger
');
fprintf(fid, 'Created at: %s', strftime(0));
fprintf(fid, 'Not counted pulses or files (%d), length(%d)
', not_counted);
fprintf(fid, 'Total number of files (%d)
', length(files));
if isempty(charge_data)
    fprintf(fid, 'No Charge data
');
else
    fprintf(fid, 'Needle 1 Average Charge = %d +/- %d (nC)
', mean(charge_data(:,2))*1e9, std(charge_data(:,2))*1e9);
    fprintf(fid, 'Needle 2 Average Charge = %d +/- %d (nC)
', mean(charge_data(:,3))*1e9, std(charge_data(:,3))*1e9);
    fprintf(fid, 'Average Total Charge = %d +/- %d (nC)
', mean(charge_data(:,1))*1e9, std(charge_data(:,1))*1e9);
    fprintf(fid, '------ >>> Percentages <<< ------
');
    fprintf(fid, '------ >>> of three cases N1 and N?, N1 only, or N? only <<< ------
');
    fprintf(fid, 'Number of dual pulses = %d (out of %d), -> %4d %%
', length(find(needle_pulse(:,3) == 1)), length(needle_pulse(:,3)), (length(find(needle_pulse(:,3) == 1)) / length(needle_pulse(:,3))) * 100);
    fprintf(fid, 'Number of single pulses (ch3) = %d (out of %d), -> %4d %%
', length(find(needle_pulse(:,1) == 1)), length(needle_pulse(:,1)), (length(find(needle_pulse(:,1) == 1)) / length(needle_pulse(:,1))) * 100);
    fprintf(fid, 'Number of single pulses (ch4) = %d (out of %d), -> %4d %%
', length(find(needle_pulse(:,2) == 1)), length(needle_pulse(:,2)), (length(find(needle_pulse(:,2) == 1)) / length(needle_pulse(:,2))) * 100);
end
fclose(fid);

for k=1:length(names)
    n = min([length(folder_name{i,:}), length(char(names(k)))]);
    if strncmp(folder_name{i,:}, char(names(k)), n)
        fprintf(runlog, '------ >>> Folder: %s <<< ------
', char(names(k)));
        fprintf(runlog, 'Needle to Needle Gap = %d (cm)
', distance(k));
        total_pulses_counted = ch3_and_ch4_pulse + ch3_pulse + ch4_pulse;
end

196
rc_norm = rc(k)/rc(13);
dual_percent = ((ch3_and_ch4_pulse)/(total_pulses_counted))*100;
ch3_percent = ((ch3_pulse)/(total_pulses_counted))*100;
ch4_percent = ((ch4_pulse)/(total_pulses_counted))*100;
single_percent = ((ch3_pulse+ch4_pulse)/(total_pulses_counted))*100;
name_pair(end+1,:) = k;
percent_distance(end+1,:) = [distance(k),dual_percent,single_percent, ...
ch3_pulse,ch4_pulse,total_num_pulses-total_pulses_counted,total_num_pulses, ...
total_pulses_counted,rc_norm,ch4_percent/ch3_percent];
fprintf(runlog, 'Number of dual pulses = %d (out of %d), -> %4d %%
', ... 
ch3_and_ch4_pulse,(ch3_and_ch4_pulse+ch3_pulse+ch4_pulse),dual_percent);
fprintf(runlog, 'Number of single pulses = %d (out of %d), -> %4d %%
', ... 
(ch3_pulse+ch4_pulse),((ch3_pulse+ch4_pulse)/total_pulses_counted)*100,single_percent);
fprintf(runlog, '------ >>> Some Details <<< ------
');
fprintf(runlog, 'Number of single pulses (ch3) = %d (out of %d), -> %4d %%
', ... 
ch3_pulse,total_pulses_counted,(ch3_pulse/total_pulses_counted)*100);
fprintf(runlog, 'Number of single pulses (ch4) = %d (out of %d), -> %4d %%
', ... 
ch4_pulse,total_pulses_counted,(ch4_pulse/total_pulses_counted)*100);
fprintf(runlog, '------ >>> Charge Information <<< ------
');
if isempty(charge_data)
    fprintf(runlog, 'No Charge Data\n');
else
    fprintf(runlog, 'Needle 1 Average Charge = %d +/- %d (nC)\n', ... 
    mean(charge_data(:,2))*1e9,std(charge_data(:,2))*1e9);
    fprintf(runlog, 'Needle 2 Average Charge = %d +/- %d (nC)\n', ... 
    mean(charge_data(:,3))*1e9,std(charge_data(:,3))*1e9);
    fprintf(runlog, 'Average Total Charge = %d +/- %d (nC)\n', ... 
    mean(charge_data(:,1))*1e9,std(charge_data(:,1))*1e9);
    fprintf(runlog, '----------------------------------------------------
');
end
end
end
fclose(runlog);

% Write Percentages for HTML page
writeHTML = fopen('./Data_Check/pulses.dat','w');
fprintf(writeHTML, 'var dualPulses=[');

D.2 Power Calculation Script

The following MATLAB script was used to calculate the power using the oscillograms collected using the LeCroy:
% Erik Wemlinger
% Created: 11/2/2011
% Use: This script was written to plot and look data collected from the
% LeCroy scope taking long data sets of total current and voltage.

[Y, MO, D, H, MN, S] = datevec ( now );
current_threshold = 0.002; % amps

% folder_name = 'LeCroy_LONG-GAP_90variac8-1-2012/';
% folder_name = 'LeCroy_Long-Gap_RunC_8-1-2012/';
% folder_name = 'LeCroy_LONG-GAP_RunD_Collection2_8-1-2012/';
% folder_name = 'LeCroy_90percent_KBr_1hr_Run-2_8-10-2012/';
% folder_name = 'LeCroy_90percent_KBr_1hr_Run_8-10-2012/';
folder_name = {'LeCroy_90percent_KBr_1hr_Run_8-10-2012/', ...
               'LeCroy_90percent_KBr_1hr_Run-2_8-10-2012/'};

filename = strcat('Power_Data','_',int2str(MO),'-',int2str(D),'-',int2str(Y),'.dat');
fileData = fopen(filename,'w');
fprintf(fileData, '-------- LeCroy Positive Half-Wave Power Data --------
');
fprintf(fileData, '-------- Current less than %d set so zero --------
',current_threshold);
fprintf(fileData, 'Erik Wemlinger
');
fprintf(fileData, 'Created at: %s
',timestring());
fprintf(fileData, 'File, Peak Voltage (V), Peak Current (A), Positive Period (s), ...
Mean Energy (J), Mean Power (W)
');

for j=1:length(folder_name)
  current_files_rise = dir(strcat(char(folder_name(j)),'needle_*Current-Rising*.trc.txt'));
  voltage_files_rise = dir(strcat(char(folder_name(j)),'needle_*Voltage-Rising*.trc.txt'));
  for i=1:length(current_files_rise)
    s1 = dir(strcat(char(folder_name(j)),current_files_rise(i).name));
    s2 = dir(strcat(char(folder_name(j)),voltage_files_rise(i).name));
    if (s1.bytes == 0) || (s2.bytes == 0)
      display('skipping this file');
      display(strcat(current_files_rise(i).name));
    else
      Iup = dlmread(strcat(char(folder_name(j)),current_files_rise(i).name),',',5,0);
      Vup = dlmread(strcat(char(folder_name(j)),voltage_files_rise(i).name),',',5,0);
current = Iup(:,2);
voltage = Vup(:,2)*4105;

time = Vup(:,1);

% want the power and average power
% pick just the positive side of the voltage
temp = find(voltage > 0.0);

% Going to zero all negative values as well as values less than
% 0.002 amps.
% temp2 = find(abs(current) < current_threshold);
temp2 = find(current < current_threshold);
current(temp2) = current(temp2)*0.0;

power = voltage(temp).*current(temp);

% Calculate the charge
energy(i) = trapz(time(temp),power);  \ (J)
avgPower(i) = energy(i)/(max(time(temp))-min(time(temp)));

clf
subplot(211)
[AX,H1,H2] = plotyy(time*1e3,current*1e3,time*1e3,voltage*1e-3,'plot');
set(get(AX(1),'Ylabel'),'String','Current (mA)')
set(get(AX(2),'Ylabel'),'String','Volts (kV)')
set(get(AX(1),'Xlabel'),'String','Time (ms)')
legend('Total Current','Voltage')
title([strcat(current_files_rise(i).name)])

subplot(212)
plot(time(temp)*1e3,power,'r')
ylabel('Charge (W/s)')
xlabel('Time (ms)')
title('Instantaneous Power')
title(['Instantaneous Power (AVG Power = ',num2str(avgPower(i)),' (W)']);
% Save the figure
filename = strcat(char(folder_name(j)),'Half_Period_Data_Signal_PURE_', ... 
    int2str(i),'.png');
print('-dpng','-zbuffer',filename);

fprintf(fileData, '%s, %8.8e, %8.8e, %8.8e, %8.8e, %8.8e
', ... 
    strcat(char(folder_name(j)),current_files_rise(i).name),max(voltage), ... 
    max(current),(max(time(temp))-min(time(temp))),mean(energy), mean(avgPower) );
end

end % file for loop
d % folder for loop

fclose(fileData);

% Write to log
% get the name of the files in the folder
filename = dir(strcat('log_*.dat'));

if isempty(filename)
    filename = strcat('log','_',int2str(MO),'-',int2str(D),'-',int2str(Y),'.dat');
    fid = fopen(filename,'w');
else
    fid = fopen(strcat(filename.name), 'a');
end

fprintf(fid, '-------- Pulse Polarity Data Appended --------
');
fprintf(fid, '-------- Current less than %d set so zero --------
',current_threashold);
fprintf(fid, 'Erik Wemlinger
');
fprintf(fid, 'Created at: %s
',timestring());
fprintf(fid, 'Folder Name: %s
',char(folder_name));
fprintf(fid, 'Period = %d (s)
',max(time));
fprintf(fid, 'Energy = %d (J)
',mean(energy));
fprintf(fid, 'Avg Power = %d (W)
',mean(avgPower));
fclose(fid);