Montana Tech Library
Digital Commons @ Montana Tech

Graduate Theses & Non-Theses

Student Scholarship

Fall 2016

DEPOSITION CONTROL FOR ELECTROSPUN FIBERS

Joshua Beisel
Montana Tech

Follow this and additional works at: http://digitalcommons.mtech.edu/grad_rsch

Part of the Electrical and Electronics Commons

Recommended Citation
Beisel, Joshua, "DEPOSITION CONTROL FOR ELECTROSPUN FIBERS" (2016). *Graduate Theses & Non-Theses*. 109.
http://digitalcommons.mtech.edu/grad_rsch/109

This Thesis is brought to you for free and open access by the Student Scholarship at Digital Commons @ Montana Tech. It has been accepted for inclusion in Graduate Theses & Non-Theses by an authorized administrator of Digital Commons @ Montana Tech. For more information, please contact sjuskiewicz@mtech.edu.

DEPOSITION CONTROL FOR ELECTROSPUN FIBERS

by

Joshua D. Beisel

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Electrical Engineering

Montana Tech
2016

Abstract

Electrospinning (ES) is a process for fabricating polymer fibers that have diameters that range from tens of nanons to hundreds of microns, which has been studied for over 100 years. These fibers have been studied in applications such as: the enhancement of mechanical properties including increased sensor sensitivity and increased tensile strength, filtration enhancement, drug delivery systems, and as a lithography masking material. In order to increase the effectiveness of ES, a real time feedback control mechanism to measure fiber diameters is needed. Currently only post-process methods of measuring fiber morphology, such as scanning electron microscopy or transmission electron microscopy, are used to measure ES fibers. Many parameters including: separation distance, applied voltage, polymer viscosity, polymer molecular weight, and flow rates are used to control fiber morphology. Using these parameters in combination with a feedback control mechanism, a multiple-input multiple-output control mechanism could be developed. By using laser extinction tomography, a device was built to measure fiber diameters during deposition. The laser diagnostic device (LaD) has been able to measure the laser extinction while scanning through fiber depositions with limited repeatability.

Keywords: Electrospinning (ES), Poly(vinyl alcohol) (PVA), Poly(ethylene oxide) (PEO), Feedback Control, Laser Diagnostics Device (LaD)

Dedication

To my family: there were difficult times but you motivated me to do my best.

Acknowledgements

I would like to thank Montana Tech and the Department of Electrical Engineering for the opportunity to pursue my Master's Degree with advisor Dr. Jack L. Skinner. I would like to thank the Undergraduate Research Program for the funding through the 2013-2014 school year, the ability to research a topic as an undergraduate fortified my desire to pursue my advanced degree. My sincerest appreciation to Dr. Jack Skinner for the countless hours of support during my undergraduate research, continued support throughout my Master's Degree and serving as chairman of my defense committee. My deepest gratitude to committee members: Dr. Bryce Hill and Dr. Curtis Link, for their guidance and the time that they spent reviewing and refining my thesis.

I would also like to thank the Montana Tech Nanotech Lab research group for their continued support during this research and to my research committee for their continued support. Special thanks to Gary Wyss and Ronda Coguill of the Center for Advanced Mineral and Metallurgical Processing (CAMP) at Montana Tech for his assistance.

Work presented was funded by Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-15-2-0020. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Table of Contents

List of Tables

List of Figures

List of Equations

Glossary of Terms

Term	Abbreviation	Definition
Commercial At	@	The commercial at symbol.
Electrospinning	ES	The process of applying high voltages to a polarizable polymer in order to draw small fibers.
Fiber Morphology		Fiber characteristics including: diameter, porosity, cross-sectional shape, and deposition formation, such as alignment.
Melt-Electrospinning	MES	Electrospinning using solid polymers that have been heated to lower the viscosity.
Nanon		Commonly referred to as a nanometer, equal to 1E-9 meters.
Octothorpe	#	The pound or hashtag symbol.
Solution-Electrospinning	SES	Electrospinning with solutions of polymer and polymer specific solvents.
Spinability		The ability to generate fibrous polymer structures through the process of electrospinning.
Tomography		The process of measuring sections with a penetrating wave. As used with the laser diagnostics device through ES fibers, this process uses laser light to penetrate the sample.
Onion-Peeling Deconvolution		Mathematical operation of extracting extinction coefficients of annular regions where constant concentration, particle size, and properties are assumed for each annular region.

1. Background on electrospinning

Electrospinning (ES) is a process for fabricating polymer fibers with diameters ranging from tens of nanons to hundreds of microns which has been studied for over 100 years [1]. High voltages are used to create an electric field that draws the polymer fiber from a polymer supply. Electrospun fibers have uses in: the enhancement of mechanical properties including increased sensor sensitivity [2] and increased tensile strength [3, 4], filtration enhancement [5], drug delivery systems [6, 7], and as a lithography masking material [8].

The basic setup of ES includes a polymer delivery system, a capillary, and a high voltage power supply (Figure 1). This basic setup can be used for melt-electrospinning (MES) or solution-electrospinning (SES). When used for MES, the polymer delivery system includes heaters to lower the viscosity of the polymer so that it will flow through the capillary. With SES, a polymer solution is made using polymer specific solvents to decrease viscosity. The primary focus of this research deals with SES. In the basic setup, the collection plate is a flat, conductive surface. Modifications of the collection plate include using a conductive rotating drum [9-11], conductive disk [12], and split electrodes [13, 14].

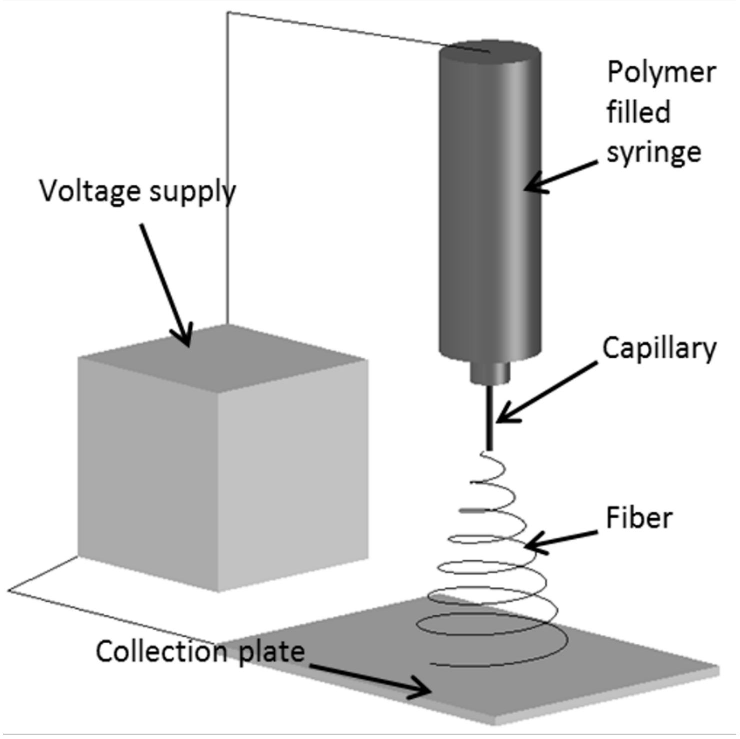

Figure 1**: Basic electrospinning setup where a polymer delivery system is connected to a capillary. A high voltage power supply applies a DC voltage potential between the capillary and the collection plate.**

Parameters of the ES process can be modified to control the fiber morphology. The fiber alignment and deposition size can be controlled through changes in the electrostatic field [14, 15]. Fiber diameter and cross-sectional shape can be modified by varying of the polymer viscosity, flow rate, and applied voltage [16, 17]. Fiber porosity can be controlled through solvent choice [18]. Research demonstrates the controllability of the ES process; however, there is currently no effective feedback mechanism for the electrospinning process. Current methods for controlling the electrospinning process include: scanning electron microscopy (SEM), transmission electron microscopy (TEM), Raman spectroscopy, and optical microscopy (for larger diameter fibers).

1.1. Controlling Deposition

Deposition control becomes critical when using electrospun fibers for device fabrication. The diameter of fibers produced is proportional to the applied voltage and the separation distance between the capillary and the collection plate (Figure 1) [15]. Research into precise spatial deposition research has included the ability to control near-field ES [19, 20]. Near-field electrospinning is defined as collecting the electrospun fibers in the stability region of the deposition (Figure 2); whereas far-field electrospinning occurs when the collection plate is in the chaotic/unstable region. An active method of controlling far-field deposition uses stacked ring electrodes to lengthen the stable region of the fiber [21]. To control far-field ES deposition the use of rotating disks [12], drum collectors [9-11], an arrangement of magnets [22], and multiple electrodes with passive and active control has been studied [23].

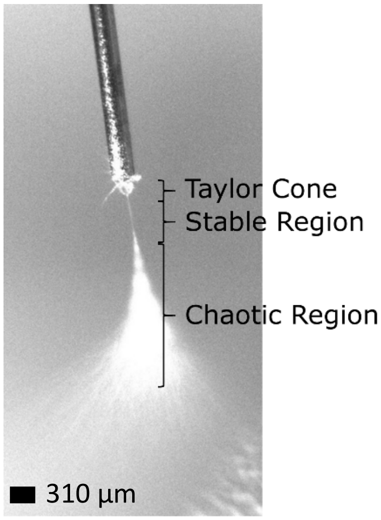

Figure 2: Regions of ES deposition. With near-field ES the collection plate is within the stability region of the fiber; whereas with far-field ES the collection plate is in the unstable/chaotic region of the fiber.

1.2. Fiber Morphology

Fiber size, shape, and porosity can be changed based on the characteristics of the polymers that are used. When performing ES with volatile solvents, fibers with a porous surface structure can be created [18]. Fiber diameter can be controlled by changing the viscosity of the polymer that is being electrospun. Higher viscosities result in larger fiber diameters whereas lower viscosities result in smaller diameter fibers [6, 17]. For poly(vinyl alcohol) (PVA) the shape of the fiber is altered by changing the molecular weight of the polymer. Lower molecular weights (9 000 – 10 000 g/mol) result in beaded fibers, whereas higher molecular weights (31 000 – 50 000 g/mol) result in flat fibers [16].

1.3. Laser Diagnostics Device

Current methods of measuring fiber diameter include the use of post process testing methods such as SEM and TEM. These methods can be time consuming, especially if the samples need to be sent to a secondary facility for imaging. The measurement techniques also require expensive, specialized equipment. In an effort to reduce the need for these post-processing techniques and develop a feedback mechanism for the ES process, a laser diagnostics device (LaD) system was built (Figure 3). The LaD utilizes the Beer-Lambert law in conjunction with the scattering of the laser on the fibers to calculate fiber diameter. Laser extinction measurements through ES fibers will be compared to polymer specific calibration curves to determine the fiber diameter. A 650 nanon laser is directed to a photodiode sensor and scans through the fibers as they are being deposited.

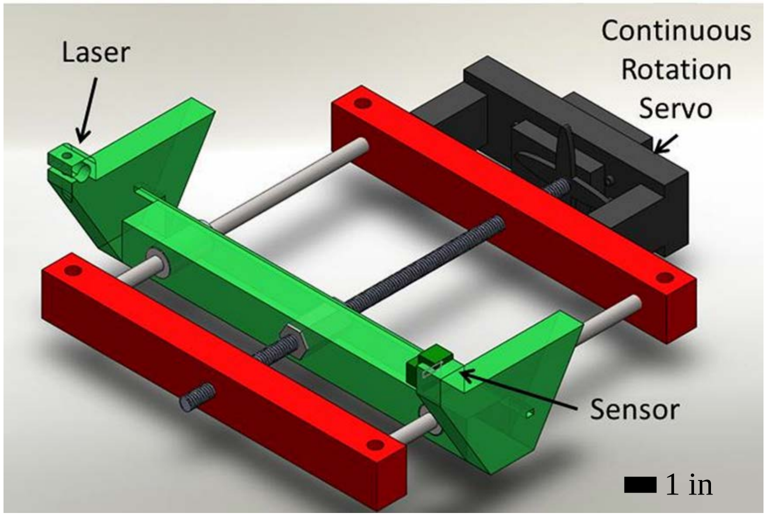

Figure 3: Original laser diagnostics tool setup. This setup utilized a continuous rotation servo motor and an additive manufactured linear stage to move the laser and sensor simultaneously.

Light interacts with matter through reflection, refraction, absorption, and transmission. Laser extinction is the combination of reflection, refraction, and absorption. Methods of calculating soot-volume fraction by measuring laser extinction through combustion reactions have been established [24-26]. After the laser extinction has been measured, tomography in combination “onion-peeling” deconvolution is used to calculate the extinction coefficient of each annular area. A vector of extinction coefficients are used to calculate soot-volume fraction [27]. The onion-peeling method slices the sample into concentric rings so that the extinction coefficient for each section can be calculated and removed from the next ring (Figure 4).

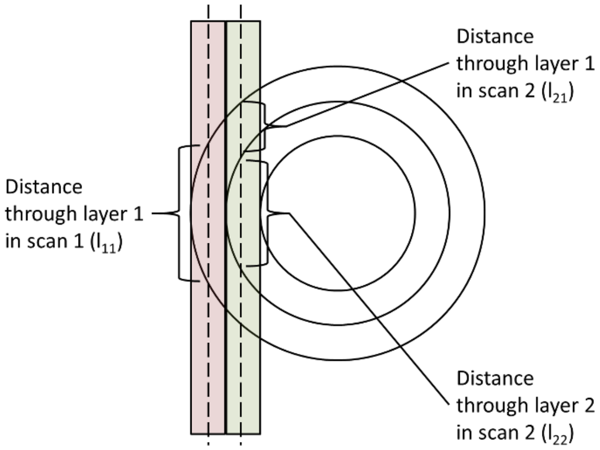

Figure 4: Laser interrogating ES fibers during deposition. The layers that are created for each scan demonstrate the onion-peel deconvolution process.

2. Controlling Deposition

Electrospinning (ES) deposition control, for the purpose of this paper, will be confined to the size of the fiber deposition and the alignment of the fibers. When measuring the size of the deposition, the critical dimension being measured is the diameter. This is assuming a random deposition with a basic ES setup (Figure 1). Alignment analysis will focus on the bandwidth (full width, half maximum) of the 2-dimensional Fast Fourier Transform (FFT) and the peak value of the radial sum of the 2-D FFT.

2.1. Size of Deposition

The following study presented in section 2.1 was previously published as an article in the Journal of Vacuum Science Technology (JVST-B) titled “Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers” [15].

2.1.1. Modeling

The electric field experienced by the polymer during ES was modeled using *COMSOL Multiphysics*®. *COMSOL Multiphysics*® is computer software package that performs finite elemental analysis to simulate physical phenomena constrained to set parameters. The model consisted of a capillary measuring 3.0 cm in length with a 1 mm diameter in combination with a 10.0 cm collection plate encompassed in a 20.0 cm diameter simulation sphere (Figure 5). These parameters were chosen to approximate the internal setup of the *SprayBase*® electrospinning tool. Simulation variables included the separation distance between the collection plate and the capillary and the voltage potential applied. To model the effect of applied voltage on the electrostatic field, separation distance was set to 4.0 cm and applied voltage was varied in the range between 8.0 kV and 15 kV in increments of 1 kV. When modeling the effect of the capillary-to-collection plate separation distance on the electrostatic field, voltage potential was set to 12.0 kV and separation distance varied between 3.0 cm and 6.0 cm in increments of 1.0 cm.

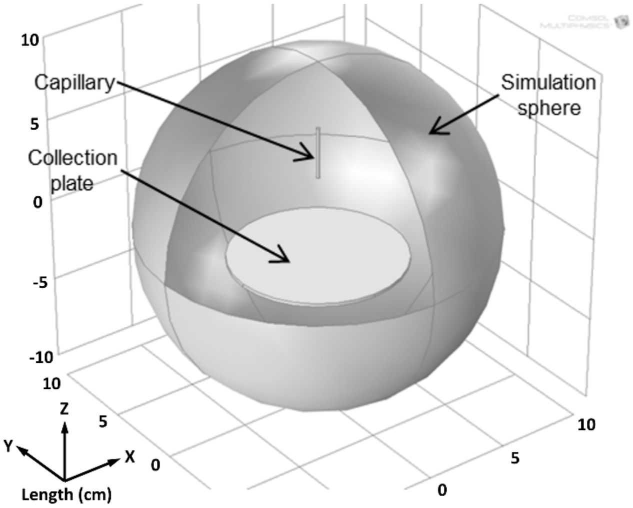

Figure 5: *COMSOL* simulation sphere containing the collection plate and capillary. The simulation assumes that the electric field is not influenced by the surroundings. Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." *Journal of Vacuum Science & Technology B* 32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.

COMSOL uses Gauss's Law to simulate the electrostatic field that is generated based on applied voltage and separation distance. Gauss's Law is applied through Maxwell's Equation to form the Neumann boundary condition:

$$\nabla \cdot E = 0 \tag{1}$$

$$E = \varepsilon \nabla V \tag{2}$$

$$\nabla \cdot (\varepsilon \nabla V) = 0 \tag{3}$$

$$\varepsilon = \varepsilon_r \varepsilon_0 \tag{4}$$

where *E* is the electric field, *V* is the applied potential, ε is the permittivity of the space between the capillary and the collection plate, ∇ is the divergence operator, ε_0 is the permittivity of free space = 8.85 E -12 F/m, and ε_r is the relative permittivity of air = 1. Through the application of the Neumann boundary condition the simulation assumes that the only field that affects the voltage is contained in the simulation sphere (i.e. the electric field lines at the simulation sphere are parallel to the sphere). Using the calculated electric field, the electrostatic force was calculated based on the electric field at a diameter of 2.5 cm. The electrostatic force experienced by the fiber was calculated from Coulomb's Law assuming a point-to-plane approximation:

$$F_{ES} = \frac{\varepsilon_r \varepsilon_0}{2d^2}\psi^2(d^2 + r^2)A \qquad [5]$$

from:

$$F_{ES} = \frac{\varepsilon_r \varepsilon_0}{2d^2}V^2 A \qquad [6]$$

where:

$$V = \psi h \qquad [7]$$

$$h = (d^2 + r^2)^{1/2} \qquad [8]$$

where *A* is the collection plate area, *d* is the separation distance, *V* is the voltage potential applied, ψ is the electric field strength, and *h* is the distance from the capillary to the collection plate (Figure 6). To calculate the changing force across the collection plate, the differential force was calculated:

$$dF_{ES} = \frac{\varepsilon_r \varepsilon_0}{2d^2}\psi^2(d^2 + r^2)dA. \qquad [9]$$

Solving for the electrostatic force with the assumption that the electric field is not a function of the radius by integrating the differential force equation yields:

$$\int dF_{ES} = \frac{\varepsilon_r \varepsilon_0}{2d^2}\int \psi^2(d^2 + r^2)\, dA \qquad [10]$$

where:

$$dA = r\, dr\, d\theta \qquad [11]$$

the electrostatic force becomes:

$$\int dF_{ES} = \frac{\varepsilon_r \varepsilon_0}{2d^2} \int_0^{2\pi} \int_0^R \psi^2 (d^2 + r^2) r dr d\theta \qquad [12]$$

$$F_{ES} = \frac{\varepsilon_r \varepsilon_0 \psi^2 R^2 \pi}{2} \left(1 + \frac{R^2}{2d^2}\right). \qquad [13]$$

However, the electric field strength is a function of the radius. By curve fitting the *COMSOL* simulation data, the field strength can be closely approximated with a quadratic equation:

$$\psi = ar^2 + br + c \qquad [14]$$

where *a*, *b*, and *c* are constants specific to the simulation. The constants change as the model parameters of applied voltage or separation distance are changed. As ψ is a function of the radius, the force can be expressed as the differential equation:

$$F_{ES} = \frac{\varepsilon_r \varepsilon_0}{2} \int_0^{2\pi} \int_0^R (ar^2 + br + c)^2 r dr d\theta + \frac{\varepsilon_r \varepsilon_0}{2d^2} \int_0^{2\pi} \int_0^R (ar^2 + br + c)^2 r^3 dr d\theta\,. \qquad [15]$$

The electrostatic force was examined at a radius of 1.25 cm. A set field strength was also examined to calculate the radius at that field strength. To calculate the radius at a specific field strength, a recursive *MATLAB* script varied the radius and solved for the field strength until the resulting field strength was equal to the desired field strength.

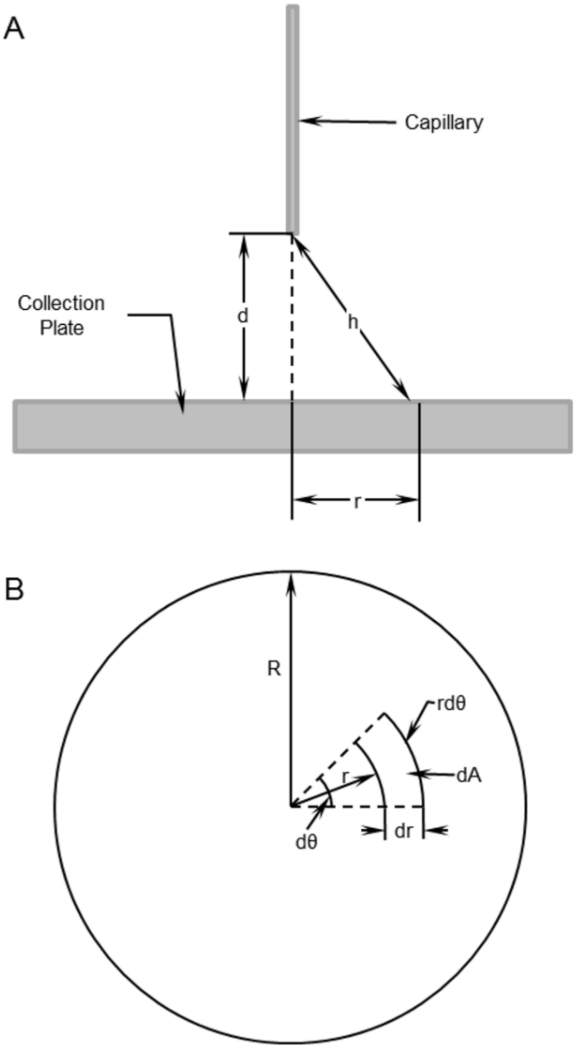

Figure 6: ES setup from the side view (A) defining measurements inside the electrospinner. A top view of the collection plate (B) defines the differential area parameters where the collection plate is centered on the capillary. Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." *Journal of Vacuum Science & Technology B* 32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.

2.1.2. Experimental Setup

Using the *COMSOL* modeling parameters, 4 wt% and 6 wt% PVA was electrospun onto aluminum collection plates. Polymer delivery was accomplished with a pressure-over-fluid delivery system. A 24-gauge capillary, outside diameter of 0.57 mm, was selected. To test the effect of the voltage potential, a separation distance of 4.0 cm was maintained while samples were electrospun at voltages ranging from 8 kV to 15 kV in increments of 1 kV. The effect of separation distance was tested by maintaining the applied voltage at 12 kV and varying the separation distance from 3.0 cm to 6.0 cm in increments of 1.0 cm. Fibers were deposited for approximately 2 min until an opaque mat became visible on the aluminum substrate. The opaque mat was measured with the assumption that minimal deposition occurs outside of the opaque area. A standard deviation for the electrospun samples was established using ten samples of 6 wt% PVA with a separation distance of 4 cm and an applied voltage of 12 kV. Polymer concentration was based on fiber morphology. When the 4 wt% PVA was ES the fiber mat contained highly beaded fibers; the 6 wt% PVA resulted in a mat of fibers with a relatively uniform diameter (Figure 7).

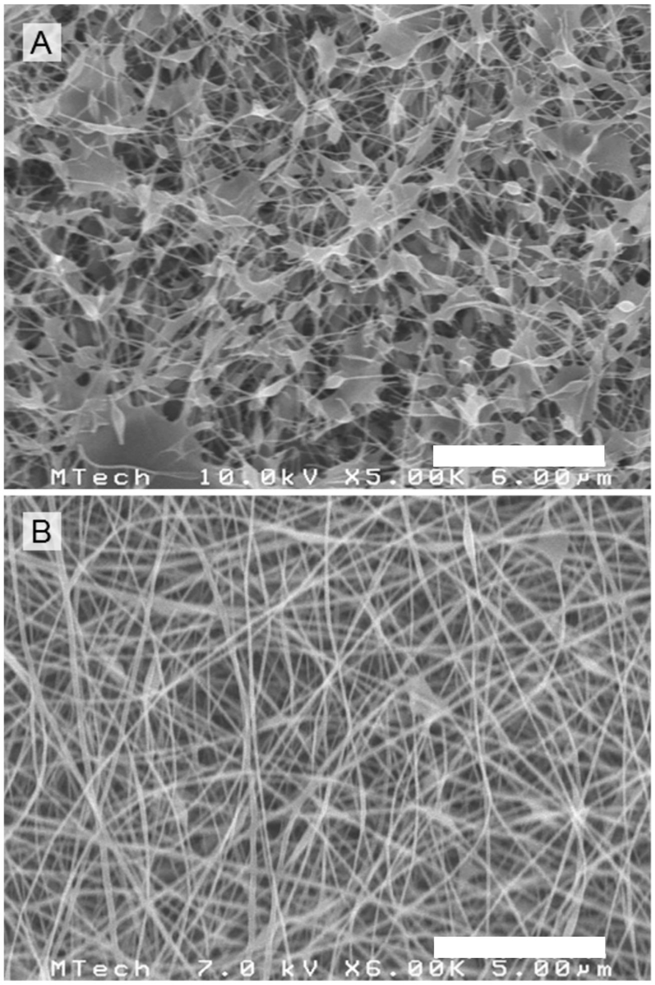

Figure 7: SEM of ES PVA fibers. The beaded fibers (A) were the result of 4 wt% PVA, whereas the uniform fibers (B) were the result of 6 wt% PVA. Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." ***Journal of Vacuum Science & Technology B*** **32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.**

2.1.3. Results

COMSOL modeling generated gradient maps that were used to evaluate the ES electric field strength (Figure 8). To eliminate the interference of the fringe effect from the recursive solver, the electric field was examined 1.0 mm above the collection plate within a 3.0 cm radius centered on the capillary. From the modeled electric field, a diameter of constant field strength was calculated. Field strength, electrostatic force, diameter of constant field strength, and deposition diameter were compared.

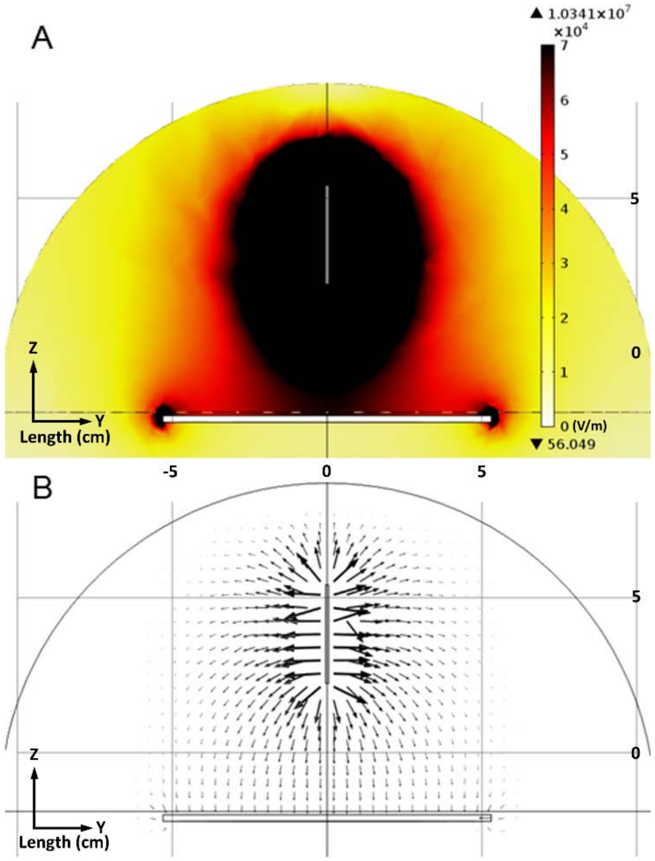

Figure 8: Heat map (A) and vector field (B) of the electric field generated from the *COMSOL* model with a separation distance of 4.0 cm and an applied voltage of 15 kV. Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." *Journal of Vacuum Science & Technology B* 32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.

When the experimental variable is the applied voltage, the deposition trend shows that the diameter of the ES deposition increases as the electrostatic force increases until a critical force is applied. After the critical force is reached the deposition diameter decreases following the diameter of constant force (Figure 9). With the separation distance set as the experimental variable, the deposition increases as the separation distance increases (Figure 10). Though similar trends are observed, there are differences between the experimental data and modeled data. By not including the opposing forces such as the capillary force and kinetics in the model, discrepancies between the modeled data and experimental data were expected.

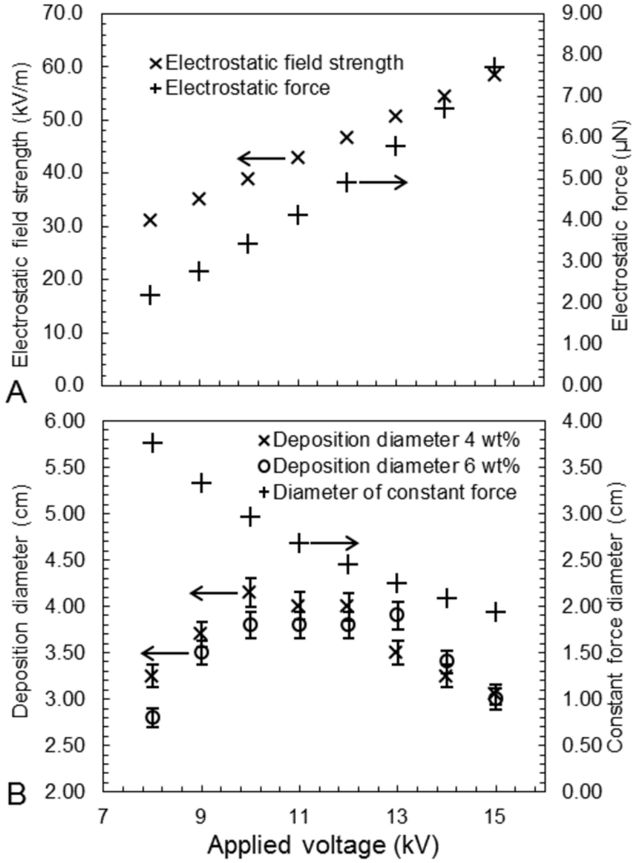

Figure 9: Electric field strength compared to electrostatic force (A) with 4.0 cm separation distance and applied voltage ranging from 8 kV to 15 kV. The experimental deposition diameter is compared to the modeled diameter of constant force (B). A trend of deposition size increasing with increasing field strength can be seen from 8 kV to 10 kV with a decreasing deposition size following the diameter of constant force from 12 kV to 15 kV. Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." ***Journal of Vacuum Science & Technology B*** **32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.**

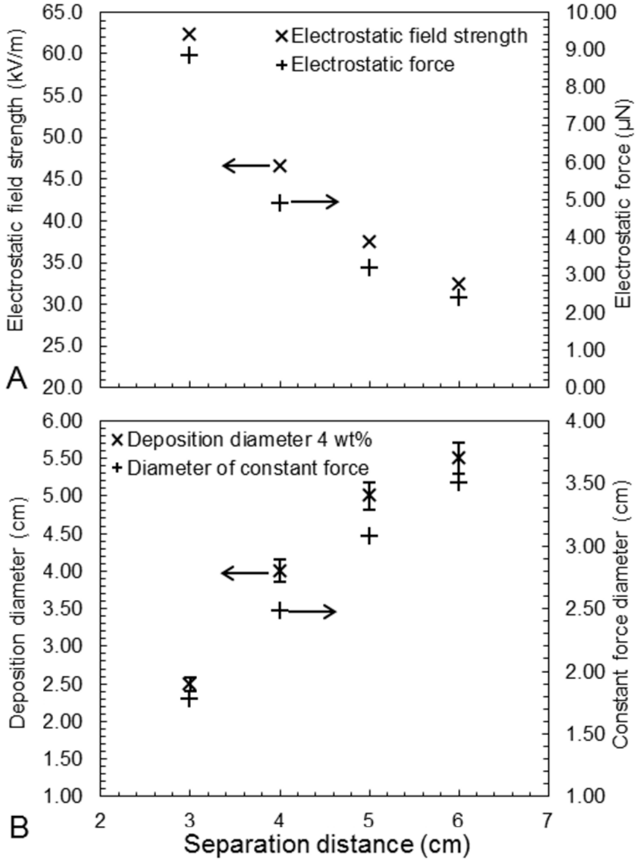

Figure 10: As the separation distance increases the electric field strength and electrostatic force decrease (A). The diameter of deposition increases with the increased separation distance following the trend of the diameter of constant force (B). Reprinted with permission from Beisel, Joshua D., et al. "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers." ***Journal of Vacuum Science & Technology B*** **32.6 (2014): 06FI03. Copyright 2014, American Vacuum Society.**

2.2. Fiber Alignment

The study contained in section 2.2 was published in Journal of Vacuum Science Technology (JVST-B) titled “Lithography via electrospun fibers with quantitative morphology analysis” [8].

Establishing active control for fiber deposition is key to being able to fabricate devices through the ES process. Aligning ES fibers with active control is possible by changing the electric field [14]. Other studies have accomplished fiber alignment through methods that do not actively control the electric field. These methods include: direct writing [20], rotating drum collectors [9-11], rotating disk collectors [12], magnets [22], stacked ring electrodes used to elongate the stable region [21], and using two equipotential electrodes with a small gap (approximately 1 mm) between the electrodes [23]. Using active control to deposit ES fibers on surfaces potentially allows for the rapid deposition of fibers on irregular surfaces and over large areas (approximately 100 cm^2). The versatility of ES in combination with fabrication methods could offer an alternative method for fabricating devices such as polarizers [28, 29], sensors [30, 31], or other electronics [14, 32, 33]. ES fibers have been fabricated with dimeters of approximately 1 nanon [34]. With a refined process of pattern transfer it could be possible to fabricate features of a similar size.

2.2.1. Experimental Setup

To align fibers with active control, two methods were studied: the function-generator modulation method and the electromechanical modulation method. Both methods of active control use two separately controlled electrodes instead of a single collection plate. A space (approximately 1.8 cm) was maintained between the electrodes for a collection plate. Modulation with the function-generator method utilizes the external controls on the high voltage power supplies to apply a sinusoidal signal with a DC offset to one electrode while the other electrode is maintained at a constant DC voltage. With the electromechanical method, the high voltage power supplies are maintained at two DC voltages that have a difference of no more than 1 kV. These two voltages are converted into two square-wave signals through a mechanical switch known as a commutator. In order to increase the fiber alignment, three electrode geometries were studied with both modulation methods. The electrode geometries are: square electrodes, right-angled (right) electrodes, and obtuse-angled (obtuse) electrodes (Figure 11). For alignment analysis, poly(caprolactone) (PCL) was electrospun for a long enough period to form a fiber mat that demonstrated the trend of the fiber deposition. PCL was dissolved in 2,2,2 trifluoroethanol (TFE) to make a 9 wt% solution for electrospinning.

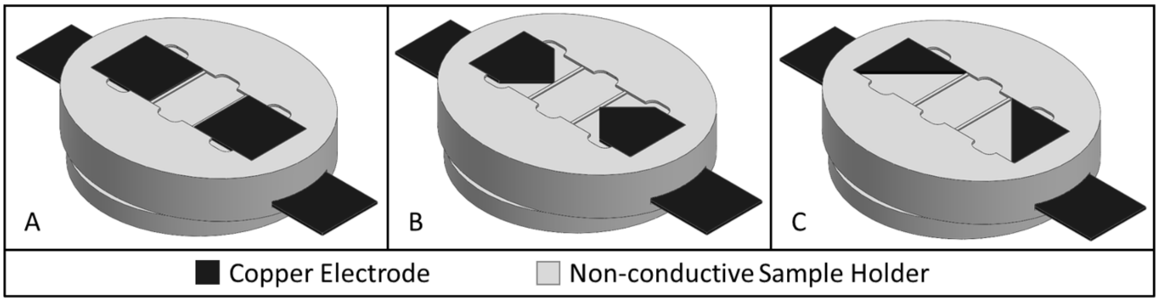

Figure 11: Independently controlled copper electrodes with square (A), obtuse (B), and right (C) geometries were placed on glass with room between the electrode pairs to place a removable collection plate. Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." ***Journal of Vacuum Science & Technology B*** **34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.**

2.2.1.1. Function-Generator Modulation of Fiber Deposition

The separation between the capillary and dual-electrode collection plate was set at 2.5 in (approximately 6 cm). The capillary was connected to the 0 V reference. DC voltage was applied to one electrode to initiate electrospinning. After a stable Taylor’s cone was established, an AC signal with a maximum peak value greater than the DC voltage and a minimum peak value below the DC voltage was applied to the other electrode. The applied voltage used to initiate electrospinning varied dependent on the electrode geometry; the square electrode was initiated at 14 kV, the obtuse at 11 kV, and the right at 9 kV. Once electrospinning had been established, a sinusoidal wave form was applied to the other electrode. The sinusoidal signal was approximately 13 kV$_{pp}$. Due to the capacitive nature of the high voltage power supplies, the signal frequency was limited to 1 Hz. High voltage power supplies use charge pumps to increase the voltage from the input to the output. Using a charge pump generates a noise signal on the rising edge of the sinusoidal as the voltage is increasing wave (Figure 12). The noise on the signal does not affect the output voltage. If the noise was carried through from the input to the output, then the noise would also have a decaying time constant similar to the 590 ms decaying time constant that is seen on the output. To control the sinusoidal wave form, a function generator is set to have an output of 2 V_{pp} with a 1 VDC offset and a frequency of 1 Hz.

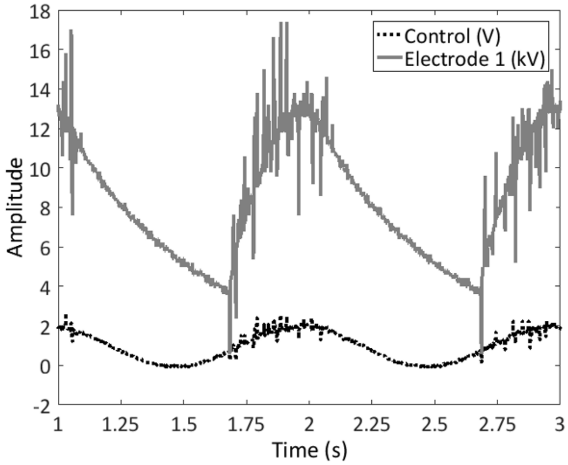

Figure 12: Sinusoidal 1 Hz control voltage and output voltage that is applied to the electrodes. The changing electric field causes the ES fiber to preferentially deposit to one electrode then the other as the voltage changes. Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." ***Journal of Vacuum Science & Technology B*** **34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.**

2.2.1.2. Electromechanical Modulation of Fiber Deposition

The electromechanical method also relies on the capillary being connected to the 0 V reference. To work around the highly capacitive nature of the power supplies, the electromechanical method modulates the signal on the output side of the power supplies. Two power supplies provide the DC voltages to the commutator which modulates the DC voltages into two square-wave signals that are 180° out-of-phase with a 50% duty cycle (Figure 13). The commutator was fabricated from poly(lactic acid) (PLA). Four carbon brushes were equally spaced around a ring gear. Strips of copper tape were applied to the outside of the ring gear to form two electrodes with a gap approximately twice the width of the carbon brush to prevent arcing between the electrodes. A maximum of 1.0 kV difference can be applied to the commutator before electric breakdown happens and the copper electrodes begin arcing to each other across the PLA. Similar to the function-generator method, the applied voltages depend on the electrode geometries that are used. The power supplies were set at 11 kV and 10 kV for the square electrodes and 7 kV and 6 kV for the right electrodes and obtuse electrodes. Rotational motion of the commutator is controlled with a 200 step-per-revolution stepper motor. A signal generator was connected to a stepper motor driver to control the speed of the stepper. When operated within the specifications of the driver and motor, the motor will step for each cycle of the control signal. The frequency of the commutator can be calculated through the equation:

$$f_{comm} = \frac{f_{ctrl}}{M_s} \qquad [16]$$

where f_{comm} is the frequency of the commutator in Hz which is equal to revolutions per second, f_{ctrl} is the frequency of the signal generator in Hz which is equal to steps-per-seconds of the stepper motor, and M_s is the step-per-revolution count of the stepper motor (200 for this design). Fiber mats were generated using 1 Hz and 2 Hz signals.

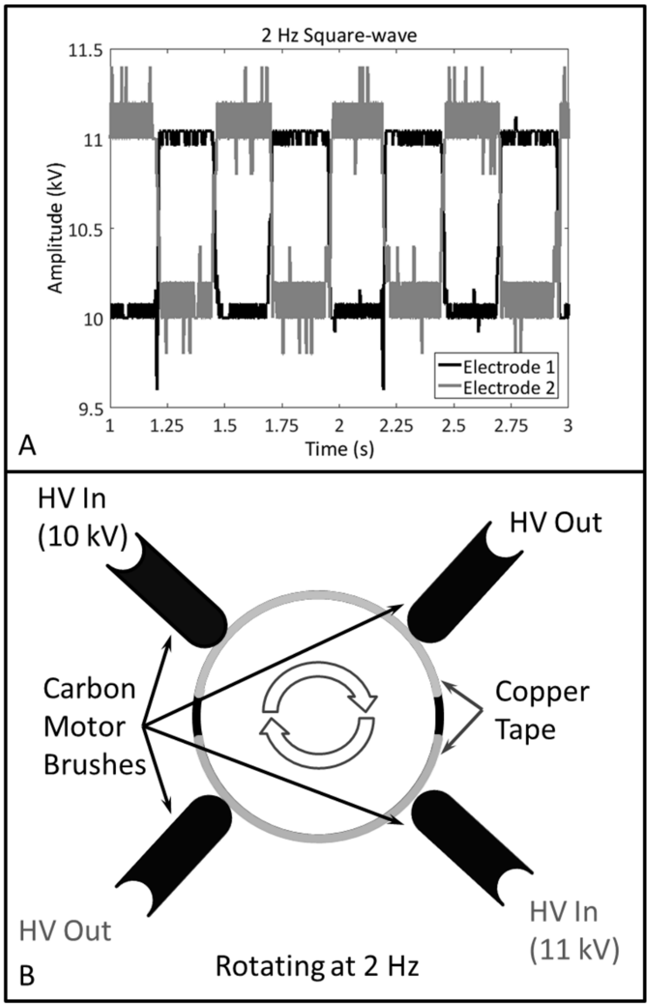

Figure 13: Square-wave signals (A) generated from the commutator (B). The input voltages are 10 kV and 11 kV which generates two 1 kVpp square-wave signals that are 180° out of phase. Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." ***Journal of Vacuum Science & Technology B*** **34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.**

2.2.1.3. Quantification of Alignment Using Fast Fourier Transform

To quantitatively define fiber alignment, the bandwidth of a two-dimensional fast Fourier transform (FFT) of the fiber mat was analyzed [35]. Fiber mats were imaged with a Hitachi S4500 SEM. Image processing began with Fiji Is Just ImageJ (FIJI), a version of ImageJ containing and assortment of extra plugins; however, FIJI discontinued support for the Oval Profile (nih.gov) java plugin which is required for the image analysis process. ImageJ (nih.gov), which still supports java plugins, was used to continue the alignment analysis. A square section of the micrograph was selected. The length of the square must have a pixel count that can be expressed as 2^n. A 512-pixel square was the largest section that could be selected while conforming to the selection criteria. This criterion eliminates distortions in the FFT. Two-dimensional FFT analysis is performed on the square section of the micrograph resulting in a spatial frequency plot. ImageJ's circle selection brush is used to select a 325-pixel diameter section of the frequency plot centered at the origin. The brush size determines the resolution of the radial sums and the amount of noise that is measured. With a 325-pixel brush, a resolution of 0.5° can be achieved with minimal pixel overlap and a low noise floor. The radial sums are calculated for the portion of the FFT that was selected with the circle selection brush using the Oval Profile plugin. The resulting radial sum is normalized to reduce the effect of uneven illumination or discrepancies in contrast and brightness between micrographs. As the fibers become more aligned, the radial sums will have a narrower bandwidth. A large quantity of fibers with the same alignment will result in a larger peak value for the radial sums plot [14]. The metrics of peak alignment factor and bandwidth do not depend on the rotation of the sample. Since the fiber sample can be rotated in the SEM the angle of fiber alignment is arbitrary, the radial sums plot was rotated to calculate the bandwidth of the peak alignment factor. *MATLAB* scripts were used to center the max alignment factor at 180° which virtually rotated the alignment angle to 180° and calculate the bandwidth of the radial sums plot (Appendix A).

2.2.1.4. Pattern Transfer of Electrospun Fiber Template

A process to pattern transfer ES fibers through reactive ion etching (RIE) was established (Figure 14). Poly(ethylene oxide) (PEO) and PVA fibers were deposited onto silicon substrates. Since PVA and PEO are water soluble, the need for, and exposure to, hazardous solvents is reduced. Thermal evaporation was used to deposit an aluminum thin film onto the ES fiber covered substrate. The fibers were removed through the use of a heated sonication bath. ES fibers have a circular cross-sectional area which is ideal for the lift-off (removal) process. A circular cross-section in combination with the directional deposition of the aluminum creates voids that aid in the removal of the ES fibers. An Oxford Plasmalab System 100 with inductively coupled plasma (ICP) etch with sulfur hexafluoride (SF_6) (etching) and octafluorocyclobutane (C_4F_8) (passivation) was used for RIE with gas flow rates: SF_6 held at 33 standard cubic centimeter per minute (sccm) with C_4F_8 stepped from 57 sccm to 21 sccm to 57 sccm in 5 sec intervals. Forward power of 20 W and ICP power of 1200 W were used. The aluminum thin film was removed after RIE with a phosphoric acetic nitric (PAN) etchant with a 16:2:1:1 ratio (water: phosphoric: acetic: nitric) at 40 °C.

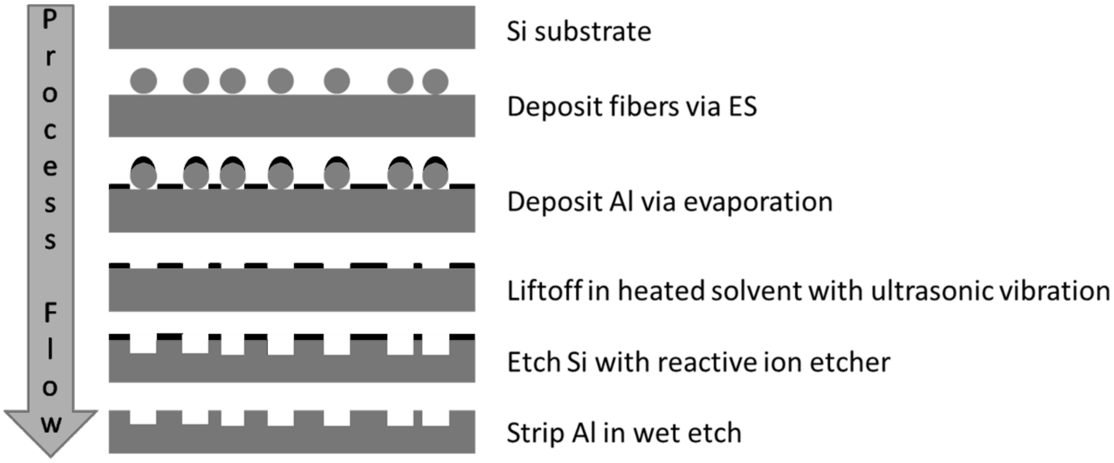
Process Flow
Si substrate
Deposit fibers via ES
Deposit Al via evaporation
Liftoff in heated solvent with ultrasonic vibration
Etch Si with reactive ion etcher
Strip Al in wet etch

Figure 14: Process to pattern transfer the ES fibers onto a silicon substrate through RIE. Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." *Journal of Vacuum Science & Technology B* 34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.

2.2.2. Results

The ability to use ES fibers for pattern transfer demonstrates a need for increased deposition control. Active control of fiber alignment has been established through two methods. Both modulation methods demonstrated the ability to increase the alignment values of ES fibers.

2.2.2.1. Function-Generator Modulation of Fiber Deposition

Each electrode pair was able to control the deposition of the ES fibers mat. The right-angled electrodes best demonstrated the ability to align fibers. Table I shows that each electrode geometry increases the alignment of the ES fibers. Square electrodes resulted in a comparable bandwidth to the random deposition of fibers; however, the increase in the peak alignment value shows that a significant amount of alignment has occurred with square electrodes. This case demonstrates the need for both metrics of alignment. When only analyzing the bandwidth there is not a significant amount of alignment between the square electrodes and the random deposition. Only analyzing the peak alignment value suggests that there is not a significant increase in pitch control between each electrode geometry. The decrease in bandwidth between the right-angled electrodes and the obtuse-angled electrodes demonstrates increased pitch control of the fibers. As the bandwidth decreases, the difference in deposition angle also decreases; these results are exemplified when comparing the right electrodes (Figure 15) to the random fiber deposition (Figure 16). When examining the results of the radial sums it is expected to observe two peaks 180° apart as an alias of the FFT; a horizontal fiber with a 0° pitch is also measured as a 180° pitch. Multiple peaks of significant magnitude are the result of multiple orientations. A secondary alignment peak can be observed at approximately 195° in Figure 15. This peak suggests that a secondary, minor, alignment orientation exists with a large enough difference in the pitch to separate it from the primary alignment angle.

Table I: Bandwidth and peak alignment values of the function-generator method compared to a random deposition of ES fibers.

Electrode Geometry	Bandwidth (degrees)	+/-	Peak Alignment	+/-
Random Deposition	56.06	3.78	0.1038	0.0119
Function-Generator Method				
Right Triangle	22.09	1.33	0.3756	0.0627
Obtuse Triangle	28.41	2.36	0.3694	0.0422
Square	50.67	9.52	0.3716	0.0279

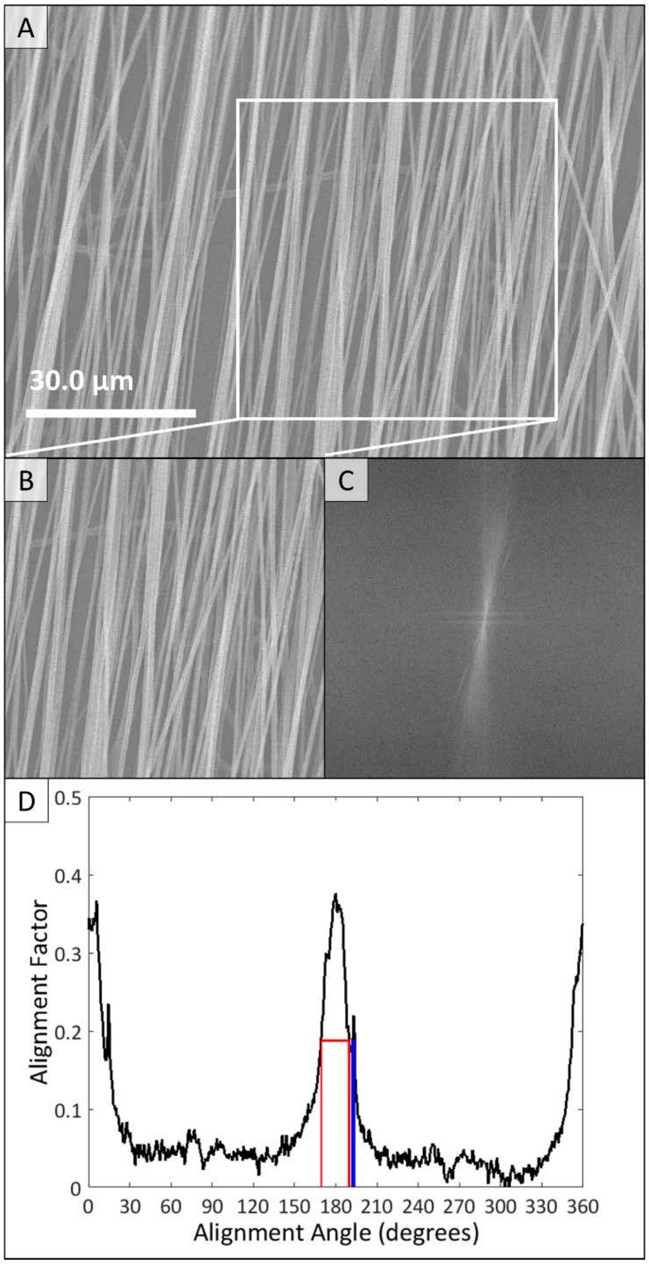

Figure 15: The micrograph of the fibers produced with the right electrodes (A) is cropped into a 512-pixel square (B). A 2-D FFT analysis of the image is shown (C). The Oval Profile plugin of FIJI calculated the radial sum of a selected number of pixels which is normalized to the minimum value to set the noise floor (D). Bandwidth and peak alignment values are calculated from the normalized data. Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." ***Journal of Vacuum Science & Technology B*** **34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.**

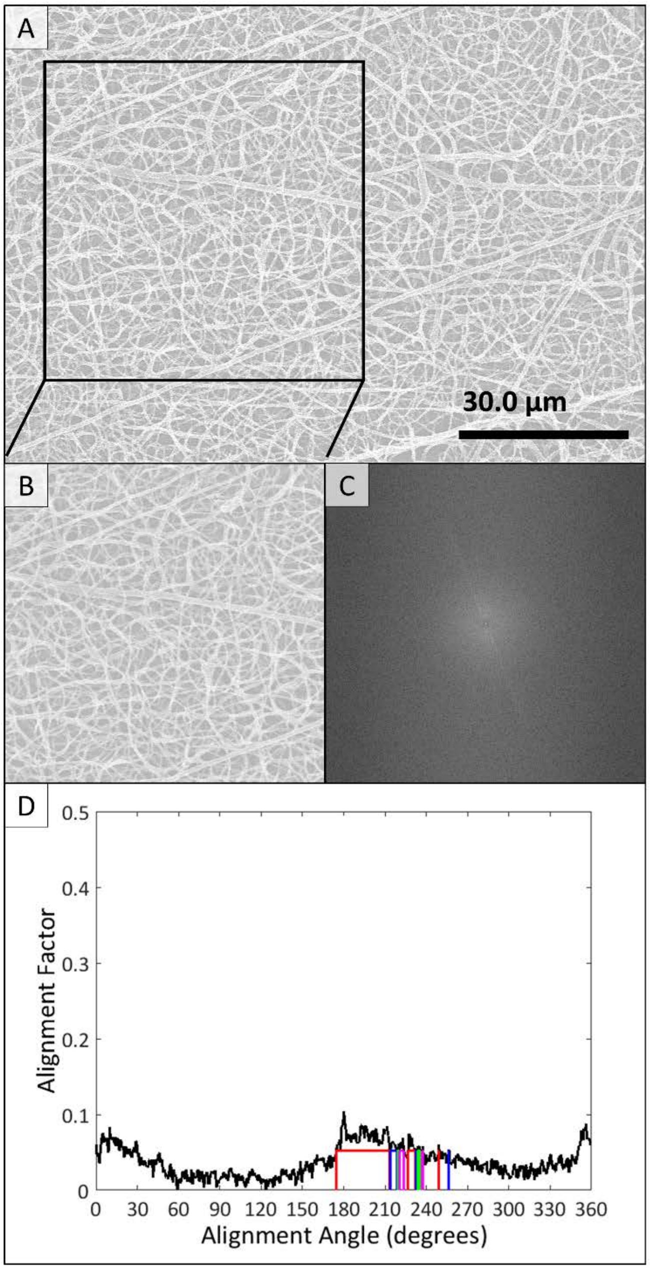

Figure 16: A random fiber sample (A) was deposited onto a single electrode for comparison. FFT analysis was completed on the random deposition follow the procedure used on aligned fibers. A 512-pixel square was cropped from the image (B) followed by 2-D FFT analysis (C). The wide bandwidth and low alignment factor provide baseline comparison values (D). Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." *Journal of Vacuum Science & Technology B* 34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.

2.2.2.2. Electromechanical Modulation of Fiber Deposition

Through the use of a commutator, the internal limitations of the power supplies were avoided. By keeping the power supplies at a constant voltage and using the commutator to modulate the output, the 590 ms time constant that is observed in the sinusoidal signal is avoided (Figure 13). Modulating the signal with the commutator also resulted in an increase of alignment (Table II) and provides a higher modulation speed than the signal-generator method. Similar to the signal-generator method the right electrodes produced the most aligned fibers. Measuring the fiber alignment with the FFT method works on dense fiber mats (Figure 15) as well as on sparse fiber mats (Figure 17). The electromechanical method is currently limited to modulation frequencies up to 2 Hz. Decreasing the amount of drag on the commutator would increase the top speed of the commutator. Increasing the signal frequency could increase the amount of pitch control of the ES fibers. With a high enough switching speed, it could be possible to deposit fibers in more complex patterns.

Table II: Electromechanical method results for fiber alignment compared to a random fiber deposition.

Electrode Geometry	Bandwidth (degrees)	+/-	Peak Alignment	+/-
Random Deposition	56.06	3.78	0.1038	0.0119
Electromechanical Method				
Right Triangle	26.28	5.51	0.4114	0.0299
Obtuse Triangle	30.66	3.37	0.4128	0.0497
Square	33.12	0.64	0.3928	0.0442

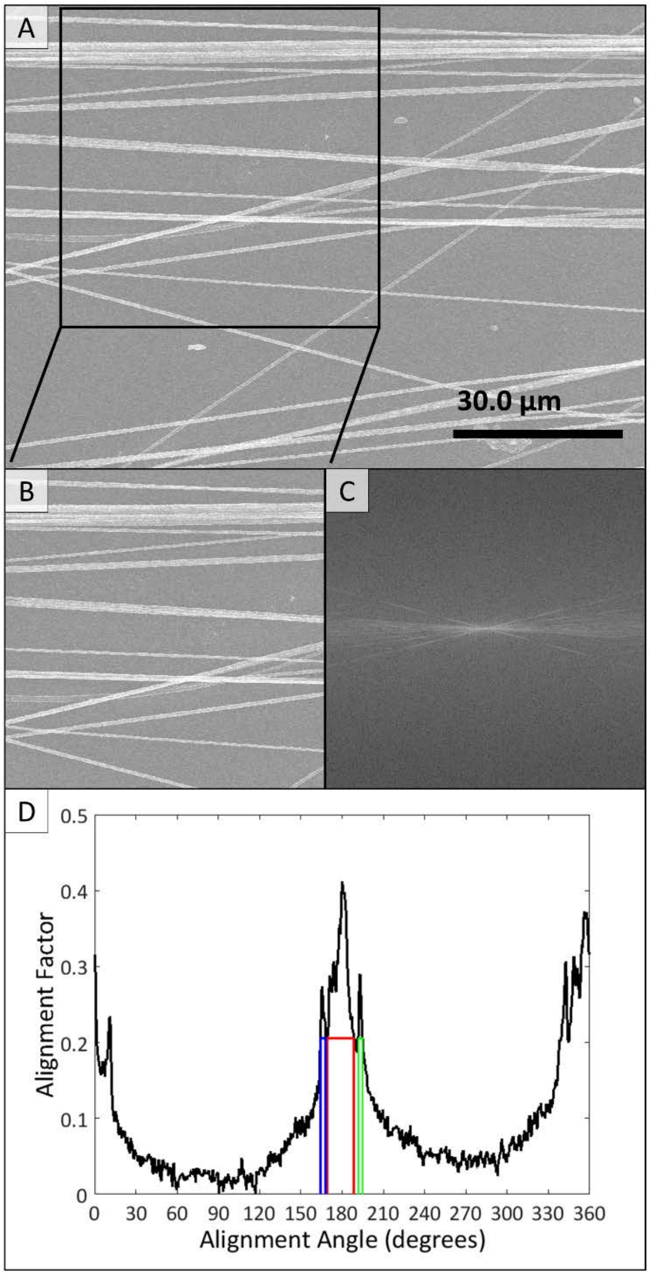

Figure 17: Fiber deposition occurred between right electrodes (A). A section of sparse fibers (B) was analyzed with a 2-D FFT (C). The sparse deposition revealed a combined bandwidth of 26.28° with a peak alignment factor of 0.4114 (D). Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." *Journal of Vacuum Science & Technology B* 34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.

2.2.2.3. Pattern Transfer of Electrospun Fiber Template

Initial aluminum thin film deposition resulted in partially conformal coverage of the ES fibers which resulted in an incomplete lift-off of the aluminum thin film (Figure 18). When incomplete lift-off occurs, defects in the pattern transferred to the silicon are formed. By adjusting the process, increasing the fiber diameter and decreasing the thickness of the aluminum film, the amount of conformal coverage was reduced which decreased the amount of defects caused by incomplete lift-off. Using the developed process for pattern transfer (Figure 14), successful RIE and pattern transfer was obtained. With ES fibers that had an average diameter of 165 nanons, features with a depth of approximately 220 nanons were transferred to the silicon substrate (Figure 19).

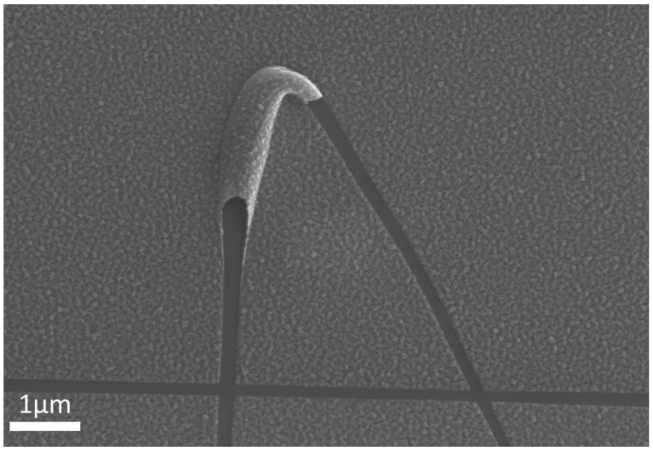

Figure 18: Incomplete lift-off of the aluminum thin film. Defects such as this are caused by a conformal coating of aluminum on the electrospun fiber. Adjusting the process parameters these defects were reduced.

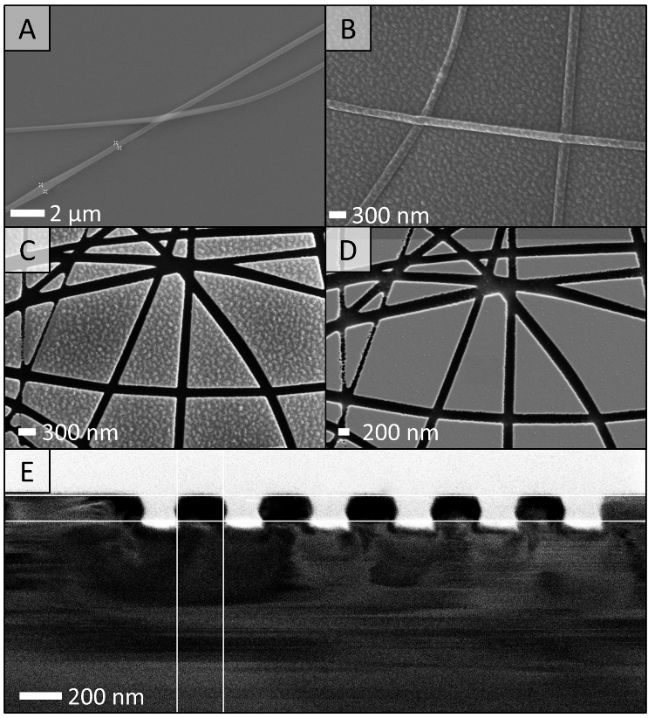

Figure 19: Following the deposition of fibers onto a silicon substrate (A), a thin film of aluminum was thermal evaporated onto the substrate (B). Where complete lift-off occurred (C), RIE successfully transferred the ES fiber pattern to the silicon. The aluminum was removed revealing an average feature size of approximately 165 nanons (D). A depth of 122 nanons into the Si substrate was achieved through the RIE process (E). Reprinted with permission from Beisel, Joshua D., et al. "Lithography via electrospun fibers with quantitative morphology analysis." *Journal of Vacuum Science & Technology B* 34.6 (2016): 06KG02. Copyright 2016, American Vacuum Society.

3. Fiber Morphology

Polymer viscosity, flow rate, applied voltage, separation distance, molecular weight, and the solvent used are parameters that influence fiber diameter, cross-sectional geometry, and porosity [6, 16-18]. These controllable parameters can be split into the categories of pre-process parameters and mid-process parameters. The solvent used to make the polymer solution and the molecular weight of the polymer are pre-process parameters. Separation distance, applied voltage, and flow rate can be changed mid-process. Polymer viscosity belongs to both categories.

3.1. Pre-Process Parameters

Complete morphology control during deposition is not practical. Some parameters need to be established before electrospinning (ES) fiber mats. For solution based ES, the solvent used to make the polymer solution can influence the porosity of the fibers (Figure 20) [18]. The use of a volatile solvent, dichloromethane, with poly(lactic acid) (PLLA) resulted in porous fibers, whereas when mixed with chloroform, which has a lower vapor pressure, the formation of pores was reduced [18]. Since the porosity of the fiber is dependent on the volatility of the solvent used, the solvent used should be considered before fiber deposition.

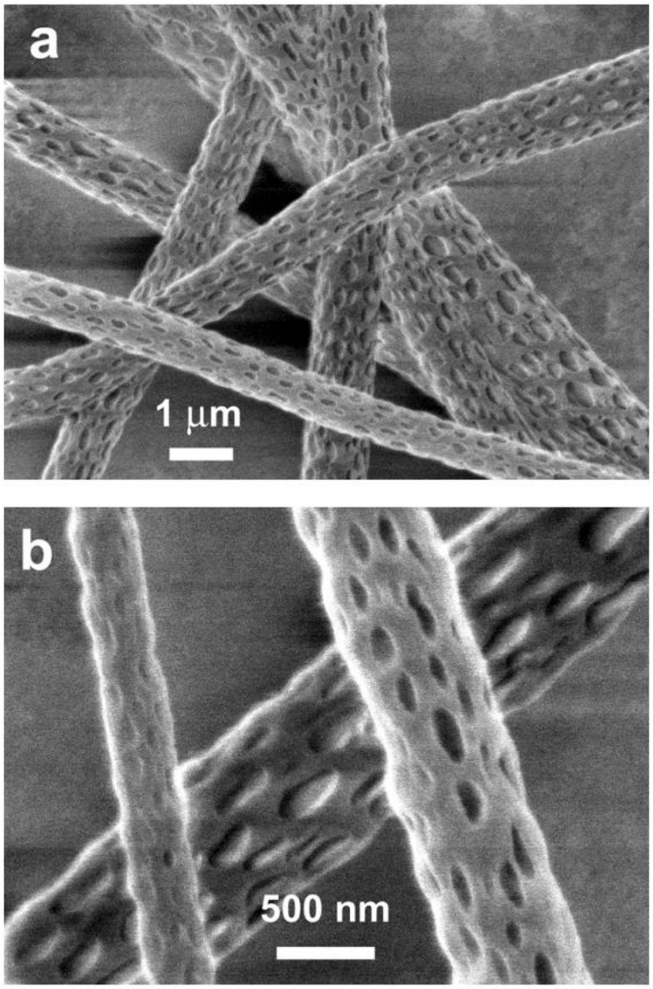

Figure 20: Micrograph of porous fibers fabricated from a PLLA dichloromethane solution (a) with magnification of the porous structures (b). Reprinted with permission from Bognitzki, Michael, et al. "Nanostructured fibers via electrospinning." Advanced Materials 13.1 (2001): 70-72. Copyright 2001, Advanced Materials.

Another pre-process control factor is molecular weight. The molecular weight of the polymer relates to the spin-ability of the polymer solution and can also influence the shape of the fibers [16]. With PVA it was shown that increasing the molecular weight of the polymer changed the fibers' morphology from beaded fibers, to round fibers, to flat fibers (Figure 21). Changing the molecular weight of the polymer also changes the viscosity of the polymer following the Mark-Houwink relationship:

$$[\eta] = KM^a \qquad [17]$$

where [η] is the intrinsic viscosity, K and a are polymer-solvent system dependent constants, and M is the molecular weight (M_W) of the polymer. The relationship factors for PVA and select solvents has been calculated. For PVA in water K = 6.51E-4 with a = 0.628; in DMSO K = 1.51e-4 with a = 0.804; and in ethylene glycol K = 3.54e-4 with a = 0.692 [36].

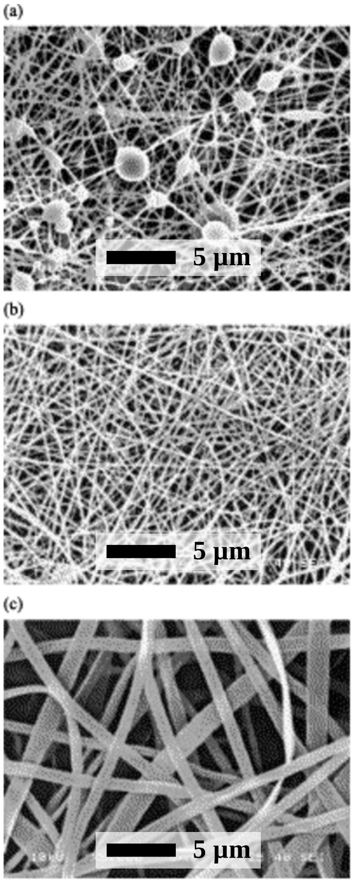

Figure 21: Various fiber morphologies of ES PVA at various molecular weights. Each solution is at 25 wt%. The 9 000-10 000 g/mol MW resulted in beaded fibers (a); 13 000-23 000 g/mol MW resulted in fiber with a circular morphology (b); and 31 000-50 000 g/mol MW PVA resulted in flat fibers (c). Reprinted from Materials Letters, Vol 58, A. Koski, K. Yim, S. Shivkumar, Effect of molecular weight on fibrous PVA produced by electrospinning, 493-497, Copyright 2004, with permission from Elsevier.

Polymer concentration also has an impact on the formation of fibers through changes in viscosity. Solutions with higher viscosities generate larger fibers [6]. This trend holds true with flat fibers. Larger flat fibers were produced by increasing the polymer concentration of PVA with higher molecular weights (Figure 22). Viscosity can also change the process from electrospinning to electrospraying. Electrospraying occurs when the solution viscosity is too low (< 800 centipoises) [17]. It is also important to note that viscosities that are too high (> 4 000 centipoises) can impede fluid flow and are too thick for electrospinning [17]. For PVA in solution with water, the spinability was examined through the Mark-Houwink relationship. Electrospinning was not stable when $[\eta]C < 4$, where C is the concentration, but became stable for $5 < [\eta]C < 12$ [16].

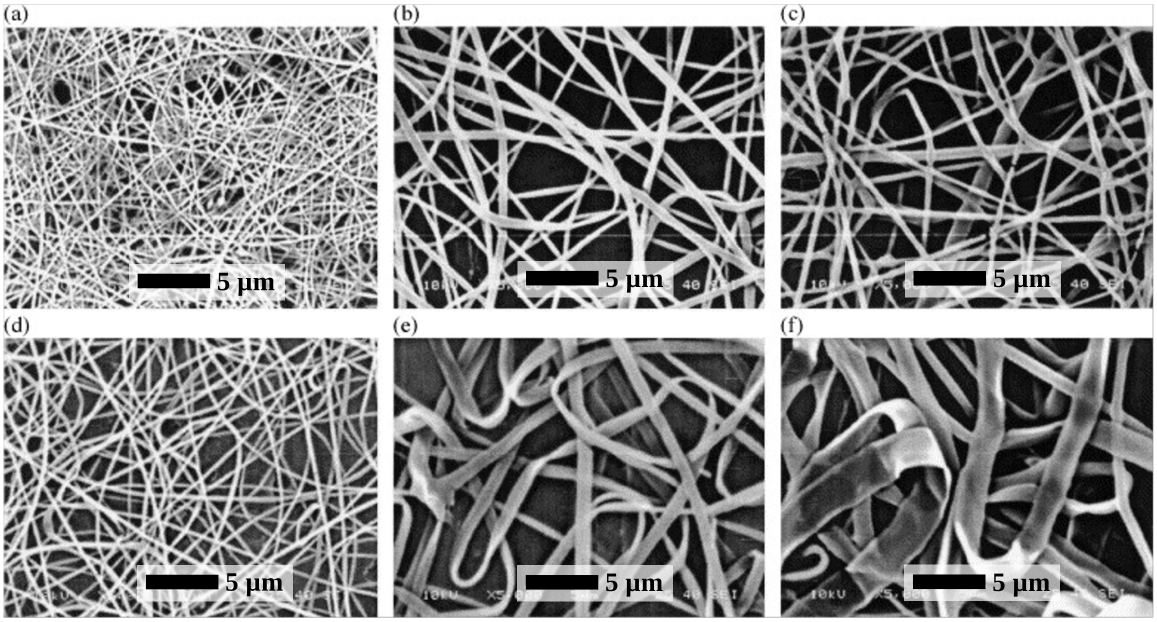

Figure 22: Micrographs of ES PVA fibers where the solvent is water. Two molecular weights were used to generate fiber mats. A molecular weight of 13 000-23 000 g/mol at 21 wt% (a), 27 wt% (b), and 31 wt% (c) resulted in fibers with a cylindrical shape. Whereas a molecular weight of 50 000-89 000 g/mol at 9 wt% (d), 13 wt% (e), and 17 wt% (f) resulted in flat fibers. Reprinted from Materials Letters, Vol 58, A. Koski, K. Yim, S. Shivkumar, Effect of molecular weight on fibrous PVA produced by electrospinning, 493-497, Copyright 2004, with permission from Elsevier.

3.2. Mid-Process Parameters

During the electrospinning process, some operation parameters can be easily varied while other parameters can be varied with additional equipment. Flow rate and applied voltage can be varied without extra equipment. Separation distance and polymer viscosity need modified setups to change these parameters during ES. In the SprayBase electrospinner, the addition of a stepper motor with added controls could be used to control the separation distance. The addition of

Peltier cells to the fluid delivery system could be used to heat or cool the polymer which in turn changes the viscosity.

Polymer flow rate affects the diameter of ES fiber. In general, slower flow rates generate smaller fibers and quicker flow rates generate larger fibers [6, 37]. Flow rate also has an effect on porosity and shape. Fiber beading is observed at higher flow rate as well as increased pore size [38]. Since flow rate correlates to the charge density of the polymer jet, flow rate is not an independent control factor [39]. As a part of the correlation, applied voltage also affects the diameter of ES fibers. The relation of fiber diameter and applied voltage is inversely correlated; as the voltage is increased, the diameter decreases [6, 37]. Applied voltage also influences the deposition area through changes in the electrostatic force used to draw the fibers. As the applied voltage increases the deposition diameter initially will increase; however, after a critical electrostatic force is achieved, the deposition diameter will decrease [15].

The electrostatic force that is applied to the ES fiber is also dependent on the separation distance between the capillary and the collection plate (Figure 1). This shows that separation distance, like applied voltage and flow rate, is not an independent parameter. By changing the separation distance the fiber diameter is changed [17]. The separation distance also influences fiber bonding [15]. If the capillary is too close to the collection plate during solution ES, then fibers will bond together through solvent bonding. With large separation distances, electrospraying is more likely to occur with solution ES whereas sponge like depositions occur with melt ES. Separation distance is a contributing factor that distinguishes the difference between near-field ES and far-field ES. As the separation distance increases a bending instability occurs which causes a whipping motion in the ES fiber [37]. The addition of ring electrodes around the ES fiber can reduce this whipping [21]. Separation distance is included as a mid-process parameter since it is able to be changed while electrospinning; although, for safety reasons it is not ideal to manually control the separation distance. The addition of servo motors or stepper motors could allow the separation distance to be safely adjusted mid-process.

Polymer viscosity, the effects of which have already been discussed, is also adjustable during ES. For polymers, viscosity can be defined as a function of temperature through the Williams-Landel-Ferry equation as:

$$\mu(T) = \mu_0 e^{\left(\frac{-C_1(T-T_r)}{C_2+T-T_r}\right)} \qquad [18]$$

where μ is the viscosity of the polymer, T is the temperature in Kelvin, T_r, C_1, C_2, and μ_0 are polymer specific variables which can be expressed in relation to the glass transition temperature of the polymer, but are best defined experimentally. For example, melt ES uses a heated chamber to decrease the viscosity of a polymer so that it can flow through the polymer delivery system to the capillary where the voltage is applied and the fiber is drawn. To fabricate smaller fibers a heated chamber around the capillary can be used to keep the melt fibers at a lower viscosity allowing the formation of smaller fibers [40, 41]. The viscosity of polymer solutions is also temperature dependent, though the use of a heated chamber around the capillary is usually used to drive off solvents [41].

4. The Laser Diagnostics Device

Currently fiber measurements are accomplished through post process imaging methods such as scanning electron microscopy (SEM) or transmission electron microscopy (TEM). This inhibits the ability to control fiber diameters during the ES process with deterministic outcomes. Because of the limitations that are imposed by post process measurements, a new method for measuring fiber diameters in real time using laser extinction and tomography is proposed. The Laser Diagnostics Device (LaD) is designed to provide a fiber diameter to the user. Based on the feedback from the LaD, the user can alter mid-process parameters or restart the process after altering pre-process parameters. However, with further study this device could be installed and interfaced as a feedback control mechanism that controls the electrospinning parameters based on the user input.

4.1. Design

Quantitative measurements of soot particles have been measured through the use of laser extinction in combination with tomography [24-27]. For this process, the position of the laser relative to the center of the deposition, the diameter of deposition, and the amount of attenuation in the intensity of the laser need to be measured. The attenuation of the laser intensity is called extinction and is the combination of refraction, reflection, and absorption. These factors are used to calculate the extinction coefficient from the Beer-Lambert law:

$$I = I_0 e^{-K*l} \qquad [19]$$

where: I is the intensity of the laser beam through the fiber deposition in watts, I_0 is the initial intensity of the laser which is measured while the laser is not in the field of deposition, K is the per meter extinction coefficient, and l is the distance through the fiber deposition calculated based on radius of deposition and the distance away from the center of deposition. The length of the laser through the sample is measured as:

$$l = 2\sqrt{r^2 - d^2} \qquad [20]$$

where: *r* is the radius of deposition and *d* is the distance from the laser to the center of deposition (Figure 23).

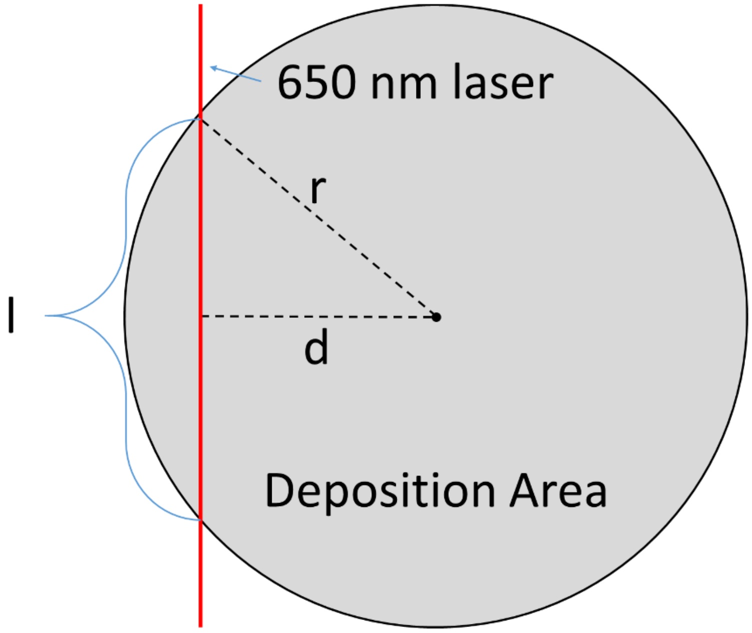

Figure 23: The length that the laser travels through the area of deposition is used in the onion-peeling deconvolution method to calculate the extinction coefficient of an annular region of the fibers that are being deposited. Through calibration curves, the extinction coefficient will be used to determine the diameter of the fibers being deposited.

As the laser passes through the deposition area, the extinction coefficient is calculated for each segment of the deposition that the laser passes through. This method of tomography is referred to as “onion-peeling” (Figure 4) [27]. For each layer of the “onion-peel” the coefficient of extinction is calculated using:

$$K_i = \frac{-ln\left(\frac{I_i}{I_0}\right) + 2 * \sum_{n=1}^{i-1}\left(K_n * l_{i,n}\right)}{l_{i,i}} \qquad [21]$$

where: K_i is the extinction coefficient of the i^{th} layer and $l_{i,n}$ is the distance through the sampled region of deposition (Figure 4).

For construction of the LaD device the processing environment needed to be considered. Due to the presence of high voltages that are shielded with a mechanical interlocking system, the LaD needs to fit inside the electrospinner without interfering with the safety interlock. The laser needs to scan through the area of deposition without stopping the ES process (i.e. the door must remain closed through the operation of the LaD during ES). Enough separation from the capillary or collection plate, whichever is at a higher potential, is needed to prevent arcing and reduce the possibility of deposition onto the device.

4.1.1. Electrical Design

Measuring the power of the laser along with the length that the laser travels through the sample are critical for calculating the extinction coefficient. Since power and voltage are proportional through the equation:

$$P = V * I \qquad [22]$$

where: P is power in watts, V is voltage, and I is current in amperes, the power of the laser can be measured as a voltage through a trans-impedance amplifier. The distance that the laser travels through the sample can be measured through motor rotation.

The initial construction of the system required the use of: a 3.3 V source to power the eZ430-RF2500 TI microcontroller, a 5.0 V source to power the laser and the continuous-rotation servo motor, and a ±15 V power supply to power the 741 operational amplifiers. This power supply was constructed with a low voltage transformer, full-wave rectifier, filtering capacitors, and voltage limiters. Initial tool design used a pulse-width-modulated signal from the microcontroller to rotate the servo motor and measure the voltage signal from a photo-diode with the analog to digital converter (ADC) contained on the microcontroller. The microcontroller programming relied on timing to determine the position of the laser and sensor and assumed that a constant rotational speed was maintained during operation. A calibration factor was included to determine the distance that the arm moved during a complete rotation of the motor. The microcontroller would move the arm a specific, user defined, distance and report a user defined quantity of measurements at each location. Data was recorded with a *MATLAB* program through a universal asynchronous receiver/transmitter (UART) virtual serial port connection. With each data measurement, a location was also sent to *MATLAB*.

After collecting all of the data from the LaD tool *MATLAB* prompted the user to define the center and edges of the deposition in relation to the collected data. From the user inputs *MATLAB* calculated the extinction coefficients through each onion-peel. The device was tested on a candle, measuring the soot-volume fraction, to show a proof of concept with a 2 mm step size defined so that the laser would not intersect other layers of the onion peel at the edge (Figure 24). Extinction coefficients were used to calculate the soot-volume fraction using:

$$\phi = \frac{\lambda * \bar{K}}{6\pi * E} \qquad [23]$$

where: $\bar{K}$ is the vector of extinction coefficients, λ is the wavelength of the laser in meters and:

$$E = -Imag\left(\frac{v^2 - 1}{v^2 + 2}\right) \qquad [24]$$

where: *v* is the index of refraction (assumed to be 1.56-j0.57 for carbon). While scanning through the candle several issues were discovered. The candle flame was not steady, the measured signal showed large errors due to noise, and the linear stage was observed to have binding issues. Because of these issues several attempts at measuring the soot-volume fraction of the candle were made before valid results were recorded. After the extinction coefficients were calculated, a soot-volume fraction was calculated for the candle. When this initial setup was installed in the electrospinner, the signal-to-noise ratio was too small for the collection of meaningful data.

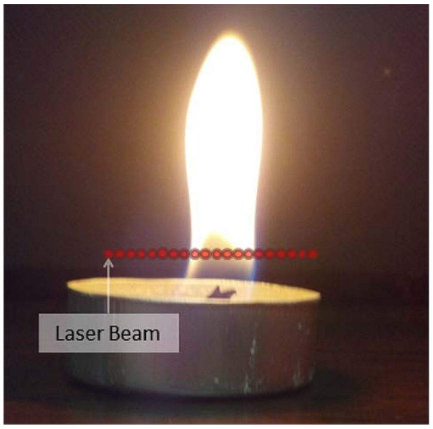

Figure 24: Synthetic image of the LaD system scanning through a candle flame to calculate the soot-volume fraction of the concentric layers of the candle. The amount of transmitted light from the laser is measured through the flame is measured as a voltage and compared to the initial intensity of the laser to calculate the amount of extinction that has occurred due to the soot particles that were generated.

The initial LaD device was not able to measure the fibers due to the low signal-to-noise ratio. Several errors that were made in the initial system were discovered which included: not using a trans-impedance amplifier, poor data filtering, and a very noisy power supply. Originally the microcontroller only reported a select number of data points when the set distance was achieved which caused issues with evaluating the noise of the signal. To analyze the signal several locations on the power supply and device were measured with an oscilloscope. A fast Fourier transform of each location showed that 60 Hz and odd harmonics were present on the power supply, the laser, and the sensor (Figure 25). To reduce the amount of noise on the circuitry a 300 W, HIPRO computer power supply with 3.3 V, 5 V, and ±12 V, replaced the original power supply. FFT analysis of the HIPRO power supply was used to verify noise reduction (Figure 26).

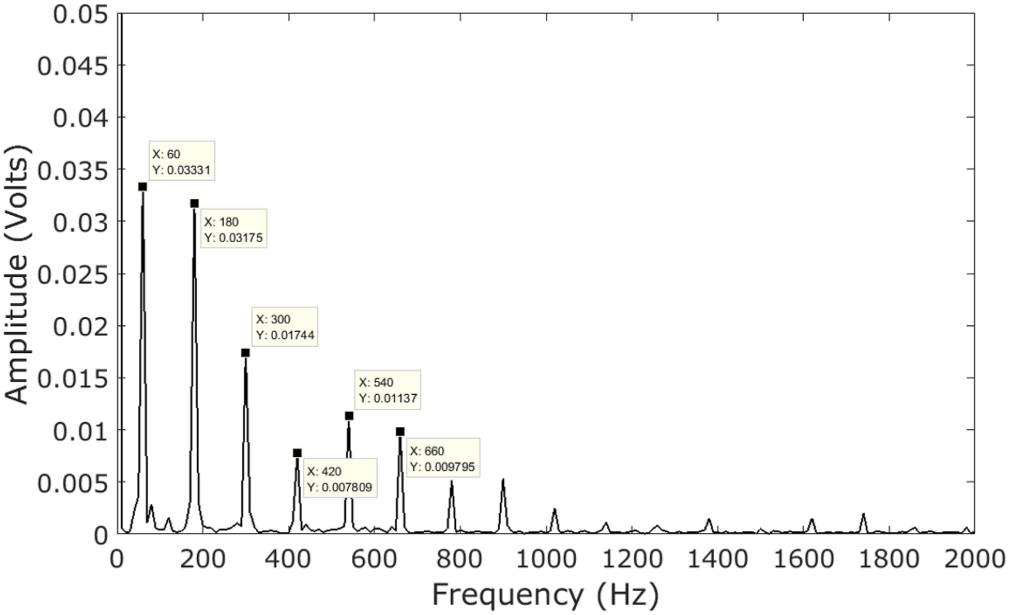

Figure 25: Frequency analysis of original fabricated power supply. Noise levels ranging from 5 mV to 33 mV contributed to the low signal to noise ratio of the laser extinction measurements.

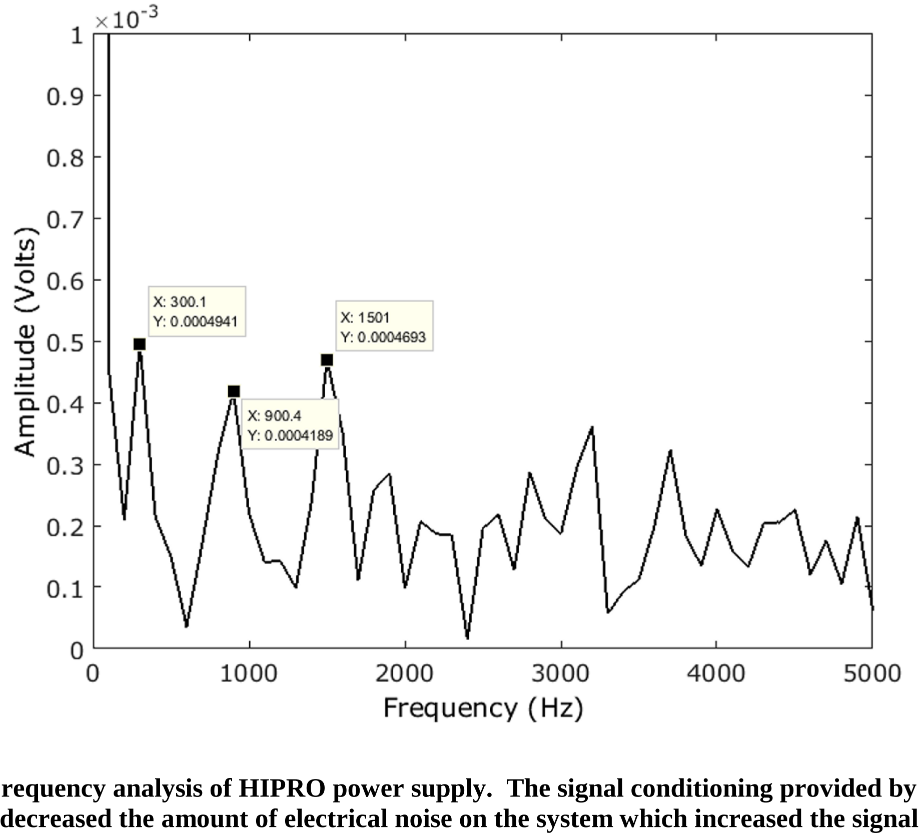

Figure 26: Frequency analysis of HIPRO power supply. The signal conditioning provided by the HIPRO power supply decreased the amount of electrical noise on the system which increased the signal to noise ratio of the extinction measurements.

Corrections to the circuitry included adding a trans-impedance amplifier to the photodiode before the low pass filter. Additional filtering was added in the *MATLAB* code to filter data outliers known as "salt-and-pepper noise" and discretizing noise from the ADC. Preliminary results were obtained through this setup and will be discussed in greater detail in section 4.2. From the preliminary results an offset-gain amplifier was added to the circuit and calibrated to increase the signal and reduce the effect of the ADC errors. The circuit diagram shown in Figure 27 models the photodiode as a current source and uses a 3.3 V Zener diode as a voltage limiter to protect the ADC input of the microcontroller.

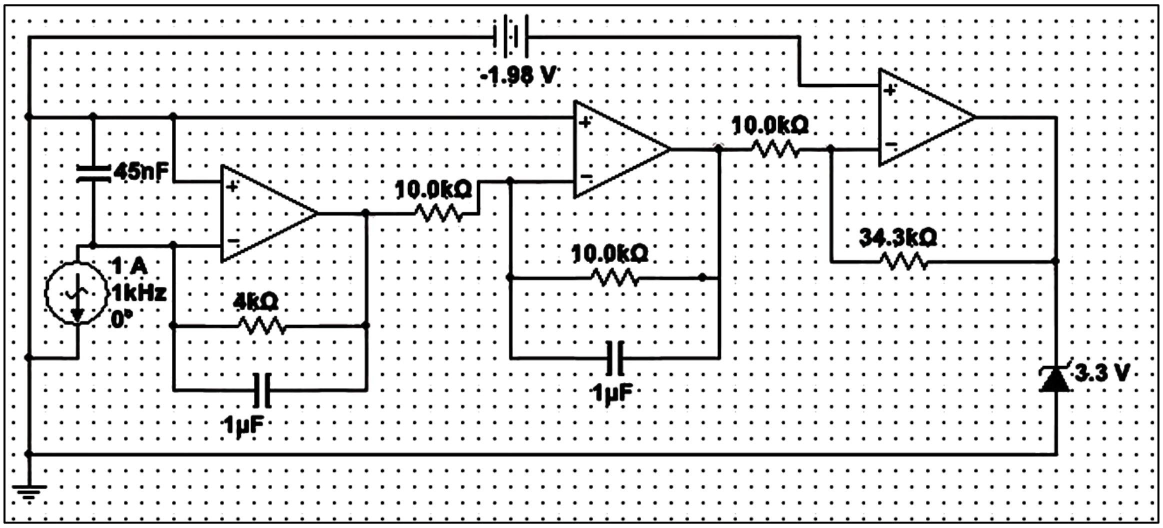

Figure 27: Circuit schematic of LaD circuitry. The node output from op-amp that is connected to the Zener is also connected to the microcontroller. An alternating current source was used in place of a photodiode to simulate the response characteristics and analyze the circuit stability.

4.1.2. Mechanical Design

Initially the LaD device was fabricated with additive manufactured parts with stainless steel hardware (Figure 3). The linear stage that moved the laser arm was constructed with two guide rails for stability. A section of ¼-20 all-thread, rotated by a continuous rotation servo, was used to move the laser arm through the electrospun sample. The system was designed to fit inside the electrospinner underneath the collection plate (Figure 28). This design was compact enough that it did not interfere with the operation of the interlock system of the electrospinner. Using a mechanical means of moving the laser and sensor meant that the ES process did not need to be stopped to advance the stage. By using additive manufactured parts, the system became highly customizable. A large separation distance (between 1.5 and 3 inches depending on capillary location) was used to reduce the possibility of arcing to the device. At the minimum distance between the capillary and LaD a voltage of approximately 114 kV is needed to achieve electrical breakdown. The electrospinner is capable of outputting 20 kV, giving the mechanical design a margin of safety of 470%.

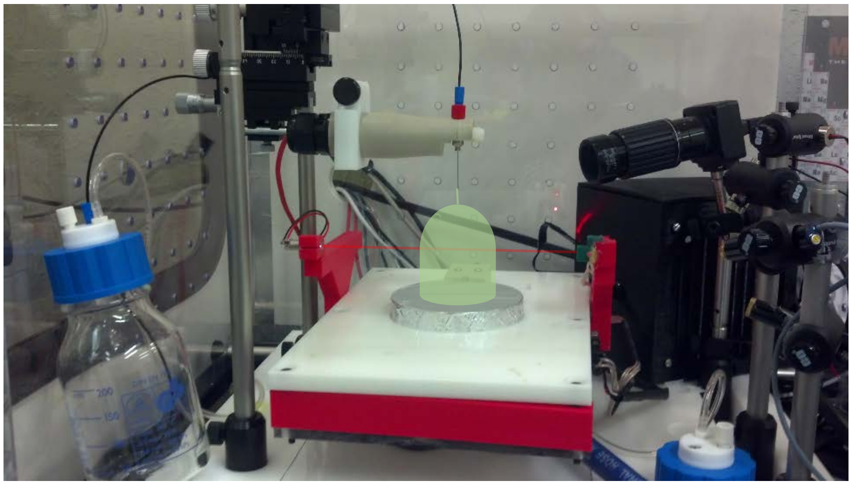

Figure 28: Initial construction of the LaD system installed in a SprayBase® electrospinner with added graphics to depict the interaction between the laser and the fibers during deposition. The laser and sensor simultaneously move to measure the amount of laser extinction due to the fibers being deposited.

Issues with the fabrication of the design became apparent during test runs of the system. The parts that were fabricated had enough discrepancies that mechanical binding occurred. When mechanical binding would occur, the distance that the microcontroller would report to *MATLAB* was longer than actual distance travelled by the laser. The difference was not enough to cause problems when measuring the soot-volume fraction of a candle with a 7 mm diameter flame. However, the difference became an issue when compounded over a 90 mm diameter collection plate. Replacing the continuous rotation servo motor with a higher torque stepper motor resolved several issues: the amount of binding that occurred was reduced and the location of the arm could be calculated with more certainty by counting the number of steps made by the stepper motor. To couple the stepper motor to the linear stage a coupling was fabricated from a flexible material used in additive manufacturing known as NinjaFlex® (Appendix B for LulzBot® TAZ 5 settings).

After removing a significant amount of noise (approximately 30 mV) from the signal the system was tested again. A slight misalignment between the coupling and the linear stage induced mechanical noise into the system (Figure 29). The offset in the coupling caused the LaD arm to bend which altered the laser alignment as the LaD was scanning through the ES fibers. Increasing oscillation amplitudes as the laser arm moved closer to the motor are most likely the result of increased deflection in the arm. The period of the sinusoidal measurements correlated to the rotation of the stepper motor. To eliminate the mechanical noise, the linear stage should be replaced by a 100 mm travel-distance linear stage.

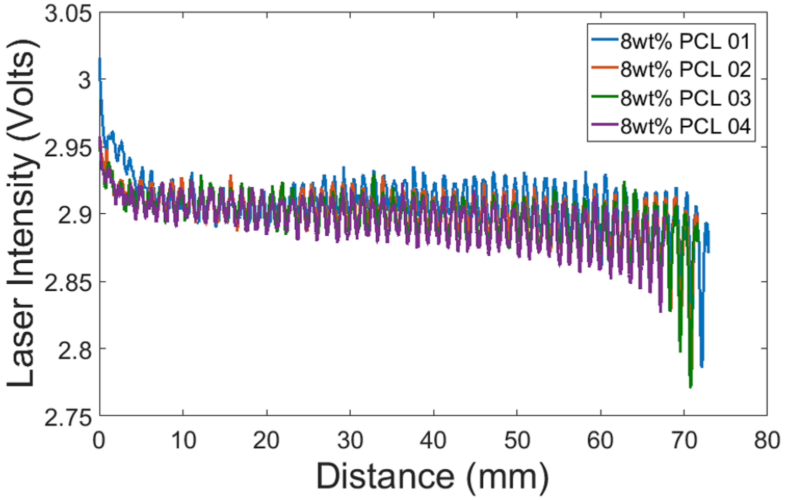

Figure 29: Sinusoidal noise induced from the misalignment between the stepper motor and the linear stage drive shaft. The period of the induced signal is equal to one revolution of the stepper motor.

4.1.3. Programming Design

A *MATLAB* script was used to create a human machine interface (HMI) for the LaD system. The initial program used the *MATLAB* workspace as the HMI. Prompts for baud rate and com port number were used to initialize UART communications to the microcontroller; whereas prompts for step size, counts per step, and scanning distance were used to set the parameters on the microcontroller. Position of the laser arm was set to a home position before the user was prompted for scanning parameters. After each scan, the collected data was saved to a spreadsheet in an xlsx format. While measuring the soot-volume fraction of a candle, modifications were noted that would increase the usability of the HMI. Initially, each scan that was performed required the user to change the filename of the previously collected data so that it would not be overwritten. Each scan also required the user to call the function from the workspace. Issues with the UART connection would present after calling the first call of the function. To decrease the likelihood of the error occurring the user was required to close the com port between each scan. Adding a while loop with an incrementing index increased the usability of the function. The user was able to run several scans without having to close the com port or call the function. An incrementing index was used to change which row the data would be saved to in the xlsx file. One series of scans would be saved to one file with multiple rows of data.

After collecting the data, the user would be prompted for more inputs including: center of deflection as a scan index, and the width of the deflection as a measurement of the scanning index. From the scanning index the distance travelled by the arm was calculated, the length that the laser travelled through the sample was also calculated. The points collected before the signal deflection were averaged to estimate the initial intensity of the laser. Prompting the user for this information increased the difficulty of the program.

The microcontroller program was designed to communicate with *MATLAB* through a UART communication protocol. A coded system was developed so that design parameters and data could be transferred between *MATLAB* and the microcontroller with little overhead. Each code corresponds to a case in a switch/case statement and are specifically formatted. When sending a code to the microcontroller the code format begins with an octothorpe followed by a two letter code and a period to mark the end of transmission (EOT). Some codes require a numeric input along with the two letter code, in these cases a colon followed by the numeric input is inserted between the two letter code and the EOT. Table III contains the codes, case number, and description for each valid code sent to the microcontroller. The data sent from the microcontroller to *MATLAB* are also formatted. Each datum is ten digits long and represents a voltage measurement and the location of the measurement. Datum formatting consists of a four digit, decimal representation of the 10-bit ADC voltage measurement, followed by an octothorpe, a four-digit measurement of the distance travelled with a commercial at to mark the EOT.

Table III: Two character codes used to reduce data overhead sent between the microcontroller and *MATLAB*.

Code:	Case:	Description:
#		Begin code.
id	0	Identify device.
sa	1	Start sending ADC values from µC (format: 1234#5678@ where: 1234 is the ADC value and 5678 is the current step count*).
st	2	Stop sending ADC values from µC.
fh	3	Find home position; this code will move the platform to the limit switch that is furthest from the stepper motor.
ss	4	Set step size; this code will be followed a signed integer.
ml	5	Start moving linear stage in set increments (need to set step size and set the number of ADC values).
cn	6	Set the number of ADC values that are sent at each increment.
sl	7	Stop moving linear stage.
:		Separate code from quantity (i.e. #ss:2.).
.		End code.

The initial LaD program interface was not user friendly, errors in the location of the laser arm were prevalent, and the rate of data transmission was seriously flawed. Changing the continuous-rotation servo motor to a 400 step-per-revolution stepper motor with a higher torque corrected the errors with reporting the location of the laser arm. Slight modifications to the microcontroller code were made to accommodate the stepper motor. Instead of a pulse-width-modulated signal controlling the rotation of the servo directly, a clock signal is sent to a stepper-motor driver along with a logic bit that controls the direction of rotation. Distance in steps is reported to *MATLAB* and converted into a distance from the origin measured in mm. The conversion factor, in mm-per-step, was calculated by measuring the distance travelled for one rotation of the stepper motor. This distance was measured for ten rotations in each direction of the stepper motor and averaged which resulted in a conversion factor of 0.0032 mm/step.

Improving the rate of data transmission required more significant changes. A timer was added to the microcontroller coding so that the data could be measured at a specific frequency.

By increasing the baud rate from 9 600 bits/sec to 115 200 bits/sec, a data rate of 100 samples/sec can be transferred from the microcontroller to *MATLAB*. Appendix C contains the microcontroller code with the modifications to the data rate and motor type. Only sending data at the required locations inhibited the ability to measure the noise; however, with the data reported at 100 samples/sec, FFT analysis on the signal can be used for targeted noise filtering. As previously discussed, changing the power supply reduced the amount of 60 Hz noise present in the system. Reducing the 60 Hz noise and increasing the data rate revealed that salt-and-pepper noise was affecting the system. A 25th order median filter was added to the *MATLAB* code to remove the outliers. The median filter was combined with a 5th order moving-average filter to remove the discretized noise from the system. Discretized noise occurs when measuring an analog signal with a digital sensor. Voltage measurements from the photodiode were obtained with a 10-bit ADC in discrete increments of 3.22 mV. Moving average filters distribute the data between the discrete points based on the surrounding data points.

LaD usability was increased through the use of a graphical user interface (GUI) which contains parameter inputs and a data plot (Figure 30). The GUI incorporates multiple *MATLAB* functions (Appendix D) that the user can call by clicking on a button. Buttons become available as the user inputs parameters, this helps to stabilize the program since the user cannot input parameters out of order. By selecting the button ‘Find LaD COM’ the GUI calls the AutoCOM function. At a set baud rate of 115 200 bits/sec, the AutoCOM function finds all available COM ports and sends the identification code to each COM port. When *MATLAB* receives the code ‘LaD_Tool’ the COM port is set for communication and the port number is displayed so the user can verify that the correct port is in use. If the code is not received from the LaD, then an error message appears letting the user know that communication to the LaD has not been established. The COM port can be opened after the correct port has been identified.

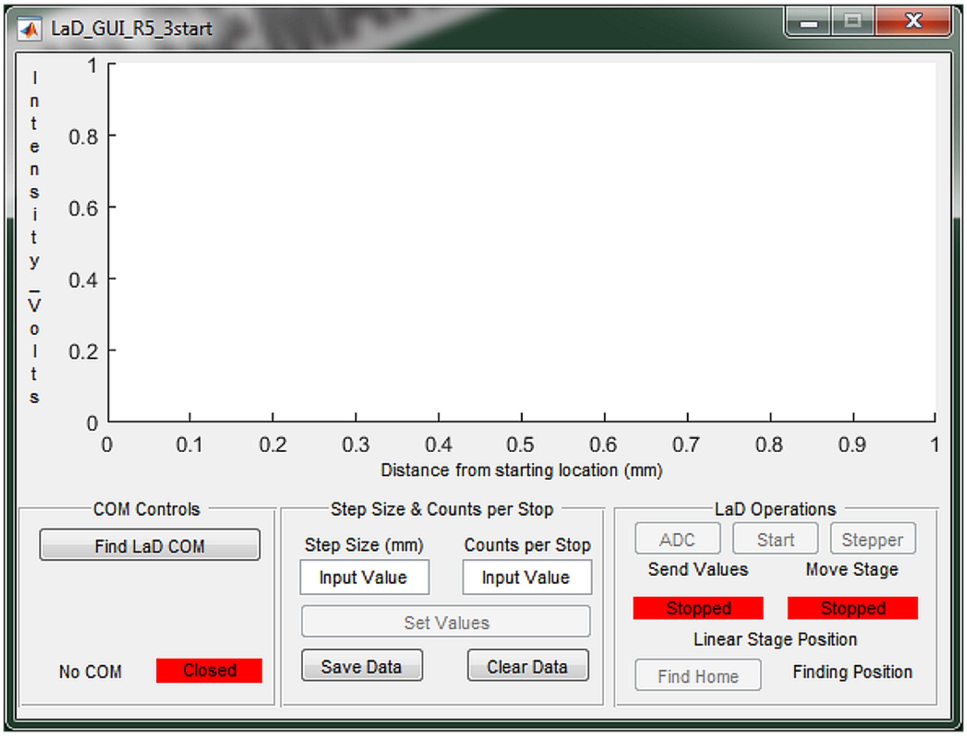

Figure 30: UI for the LaD system. The interface uses push buttons and user defined inputs to scan through the ES deposition. Buttons are available sequentially to guide the user through the operation of the LaD tool.

With an open port the user is able to set the step size and counts per step of the LaD. The step size determines the distance that the laser arm will travel before it stops to collect multiple data points. Counts per step defines the number of data points that are to be collected. Setting the parameters includes a verification between *MATLAB* and the microcontroller. If the verification step does not pass, the user will not be able to start scanning. Three scanning options are available to the user. The ‘ADC’ button starts collecting ADC measurements only, laser arm movement requires the user to manually move the arm. Changing the x-axis label from ‘Distance from starting location (mm)’ to ‘Data Point’ signifies that the location of the arm is not being tracked. ‘Stepper’ is used to start moving the arm through the use of the stepper motor. ‘Start’ is used to start the laser arm movement and the data collection.

Data storage options and a button to send the laser arm to the home position are used to collect multiple data sets. Clear data is used to remove the data points from the GUI plot. If this button is pressed before the data is saved, a warning appears letting the user know that the data has not been saved which allows the user to either save the data or clear the data without saving. When the user saves the data, a dialog box appears allowing the user to input the filename. A warning dialog appears if the filename already exists, allowing the user to either overwrite the file or change the filename. The data is saved in an xlsx format. Using sequentially available buttons and functions to automatically find the port increases the usability. An independent executable file from the GUI can be created if the required *MATLAB* packages are installed.

4.2. Laser Diagnostics Data

As a proof of concept, the initial system was used to measure the soot-volume fraction of a tea candle. The LaD scanned through the flame of the candle recording the voltage measured at the ADC. Figure 31 shows the voltage measured through the candle flame. Low signal-to-noise ratios made it difficult to obtain a usable signal. In the case of the soot-volume fraction measurement, most of the noise is attributed to the power supply that is used for the LaD. Noise levels were measured to be approximately 33 mV at 60 Hz and 180 Hz with lower noise levels on the 60 Hz odd harmonics. Discretizing noise and salt-and-pepper noise are also a contributing factor to the amount of noise that is seen on the collected data. The data shows a deflection approximately equal to the amount of 60 Hz noise seen in Figure 25, giving the LaD an estimated signal-to-noise ratio of 1:1.

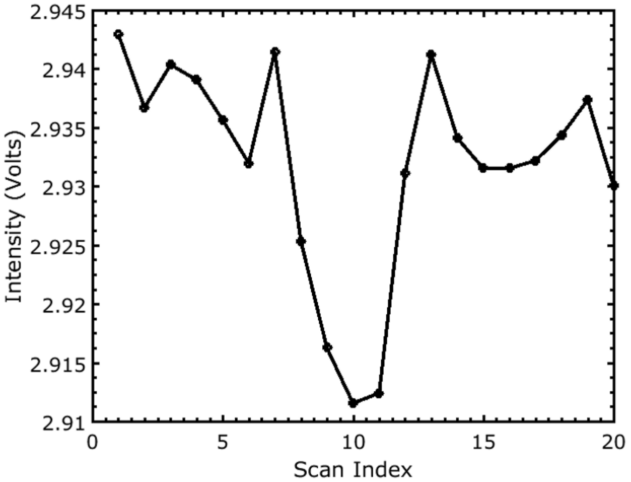

Figure 31: Reduction in laser intensity as the LaD system scans through the candle flame. The reduction is caused by the interaction between the laser and the soot particles such as reflection, refraction, and absorption which are grouped together as a measurement of laser extinction.

Despite the low signal-to-noise ratio, the collected data was able to give a reasonable quantity for the soot-volume fraction. Using Equations 23 and 24, with an assumed extinction coefficient of 1.56-j0.57 for carbon, the soot-volume fraction was calculated for each layer of the candle (Figure 32). The collected data showed that the LaD system could operate on the principles that it was designed for; however, because of the signal-to-noise ratio the original design of the LaD was not suited for ES measurements.

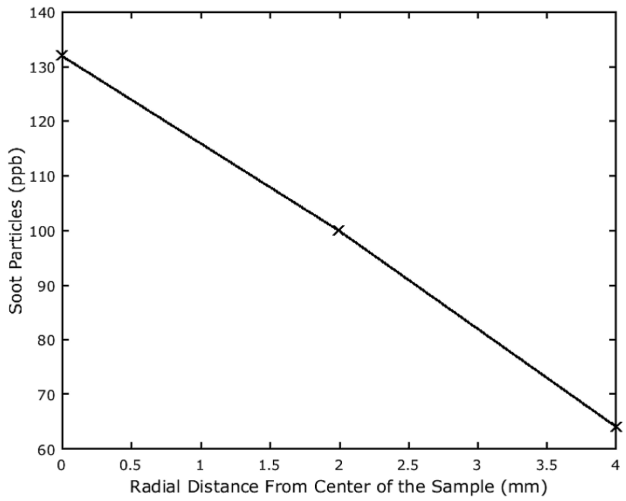

Figure 32: Soot-volume fraction through a tea candle as calculated from LaD measurements. The size of the candle flame in combination with the spot size of the laser allowed for the soot concentration to be calculated in three annular regions of the candle.

Replacing the power supply greatly reduced the noise level induced on the circuitry. Unfortunately, the noise from the power supply was not the only issue with the LaD system and other problems existed with the design of the LaD. After correcting several of these problems, the LaD was tested in the electrospinner resulting in the sinusoidal data shown in Figure 29. Switching to a manually driven system showed that the LaD tool was capable of measuring extinction due to electrospinning (Figure 33). The arm was moved by pulling the arm with a string through the ES deposition while using the original HMI to collect the data. For comparison, a noise signal was also measured while the LaD was in the electrospinner.

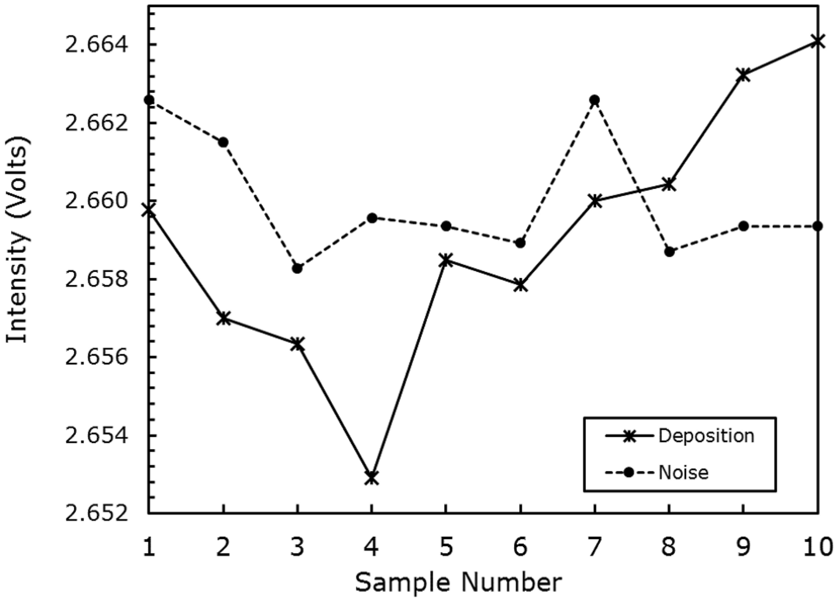

Figure 33: Extinction caused by ES fibers compared to noise floor of the sample. Interactions between the laser and the fibers were visually observed for sample numbers 2-4 with corresponding laser extinction measurements.

After demonstrating that the LaD could measure the extinction caused by ES fibers, a *MATLAB* GUI was built to control the system. The microcontroller programming was also changed to increase the baud rate and modify the rate of data acquisition. As a part of the GUI, median filtering and moving-average filtering was also added to the data acquisition. Figure 34 shows the raw measurements and the filtered measurements of the laser extinction through ES PEO fibers. The LaD was manually moved through the sample with a brief pause outside of the area of deposition then moved back through the sample. As a result of moving the stage manually, the vibration of the arm added noise into the system. Salt-and-pepper noise was filtered with the 25th order median filter and discretizing noise was reduced with the 5th order moving average filter while still conforming to the rapid changes in the collected data (Figure 34).

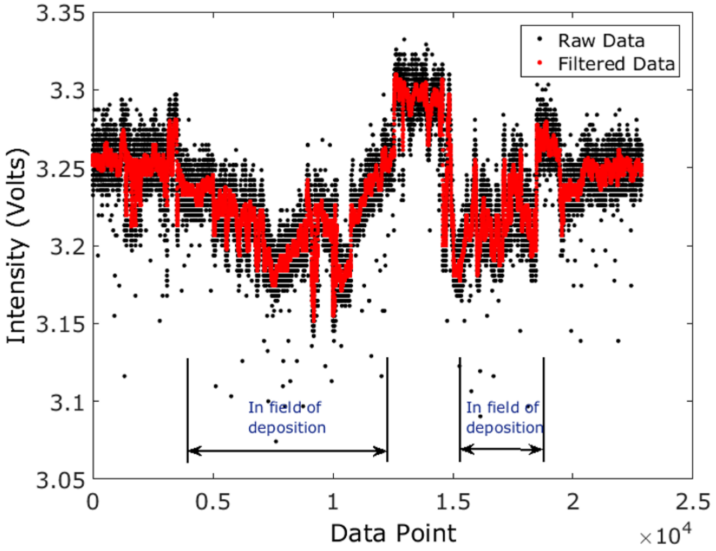

Figure 34: Laser extinction through PEO fibers at a 100 sps data rate. The fiber deposition was scanned twice, the data points designated as "in field of deposition" correspond to points where the interactions between the laser and the fibers were visually confirmed.

Based on the magnitude of the data collected, an offset-gain amplifier was calibrated to operate between 0.5 V and 2.9 V when measuring a signal of comparable magnitude. The data shown in Figure 34 only uses six of the ten bits available for data acquisition. Increasing the span of the data allows the full 10-bit ADC to be utilized. By using a larger portion of the range of the ADC, the effect of the discretizing noise would be minimized in comparison to the range of the data collected. The circuitry used to measure the laser intensity, shown in Figure 27, is a stable circuit with a gain margin of 118 dB with a phase margin of 78° (Figure 35).

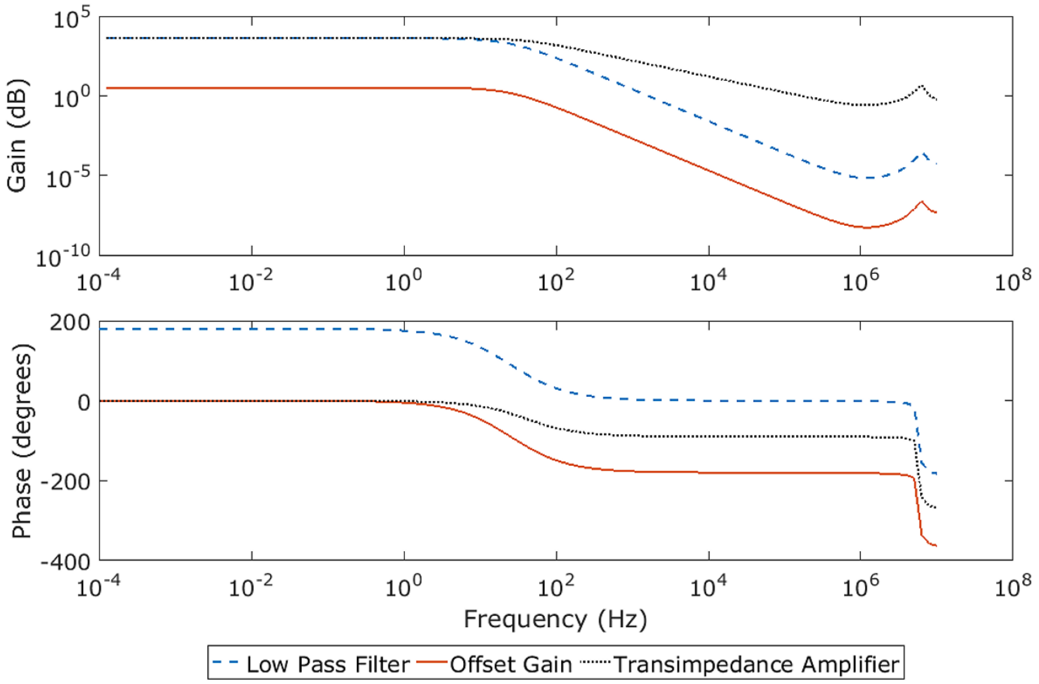

Figure 35: Frequency response of the LaD circuitry. The gain margin and phase margin of the response confirm circuit stability.

4.3. Summary

Deposition and morphology control of electrospinning has been demonstrated through multiple studies. A majority of morphology parameters can be controlled during electrospinning. With a real-time feedback mechanism, deposition control and fiber morphology can be controlled with deterministic outcomes. Based on previous studies in controlling the fiber morphology and deposition, a multiple-input multiple-output (MIMO) control mechanism could be designed to control the deposition of ES fibers.

The LaD system has the potential to be a feedback mechanism for ES. Preliminary results demonstrate the LaD's ability to measure fibers; however, the inconsistency of the results requires further development of the tool. Since the preliminary results were generated through manual movement of the stage, tomography calculations could not be completed. Movement of the laser arm is not measured when the arm is moved manually, this prohibits calculating the path length of the laser through the electrospun sample.

To troubleshoot the problems causing the inconsistent ability to record data with the LaD: the polarization of the laser, the focus of the laser, and size of the fibers have been tested. Since the laser is inherently linearly polarized, it was theorized that the angle of polarization could affect the amplitude of the extinction as the laser interacted with the fiber. If the polarization of the laser was perpendicular to the deposition of the fibers, the interaction between the fiber and the laser would be maximized and would maximize the extinction caused by the fibers. While measuring the laser intensity in the field of deposition and outside of the field of deposition, the laser was rotated to change the angle of polarization. The rotation of the laser had little to no effect on the measured intensity.

Changing the focus of the laser allows for more or less interaction between the laser and the fibers. It is theorized that a single fiber will have a greater effect on the intensity of a smaller, columnated laser. Alternatively, it is theorized that a larger, columnated laser will interact with more fibers which will also greatly affect the intensity of the measured intensity. Focusing the laser in combination with installing pinhole apertures reduced the spot size of the laser. The LaD circuitry was recalibrated to accommodate the decreased signal. Decreasing the laser spot size did not result in a measurable change in the intensity. By decreasing the focus of the laser, the spot size increases. The larger spot size did not affect the measured signal.

PEO, PVA, and PCL were used to electrospin fibers. These polymers have similar extinction coefficients. The PVA was made to electrospin small beaded fibers, PEO was used electrospin larger fibers, and PCL was used to electrospin large smooth fibers. While electrospinning with each of these polymers, the LaD circuitry was tested at specific nodes with the rest of the circuit disconnected. The circuitry was tested at the output of the trans-impedance amplifier with the low pass filter and the offset-gain filter removed from the circuit, at the output of the low pass filter with the offset-gain filter removed, and at the output of the offset-gain filter. Varying the polymer did not significantly affect the output of the LaD at any node.

5. Continuing Work

The LaD system has demonstrated the potential to be used as a feedback mechanism. Collected data would be more relevant to the process if the location of the laser arm was recorded. Manually moving the laser arm is useful to demonstrate that the device can measure the extinction caused by the fibers; however, incomplete data is useless in calibrating and measuring the diameter of the electrospun fiber. The previously mentioned linear stage needs to be incorporated into further data collection.

Changing the corner frequency of the LaD circuitry could change the way that the data is collected. A higher corner frequency of the trans-impedance amplifier and low pass filter might allow the LaD to track the extinction of the fibers as it chaotically moves through the laser, but this could mean that higher frequency noise would also appear on the measured signal. Alternatively, a lower corner frequency could integrate the signal over a longer period of time and measure the average effect of the extinction caused by the fibers.

The LaD was designed to measure fibers, based on laser scattering theory that is used to measure soot particles. Most polymers have a very low absorption coefficient (approximately 0 m^{-1}). Soot particles have a much higher absorption coefficient of 0.57 m^{-1}. With a low absorption coefficient reflection and refraction become the major interactions measured in fiber extinction. This means that the volume fraction of polymer fibers cannot be calculated. Equations 23 and 24 use the absorption coefficient to calculate a volume fraction. To increase the absorption coefficient of the fibers, the polymers could be doped with carbon. If the doped fibers yield consistent results, then the doping concentration could slowly be decreased while monitoring the effect on the measured signal. While decreasing the concentration of the dopant if the measured extinction decreases, then better signal amplification is needed to consistently measure the fibers.

The theory of laser extinction is also based on Rayleigh scattering, which occurs when light interacts with particles that are much smaller than the wavelength of the interrogating light. Mei scattering occurs as the particle size becomes approximately-equal-to or larger than the wavelength of light. This might mean that the laser wavelength needs to be increased from 650 nm. Before changing the laser, this theory could be tested by electrospinning fibers of increasing sizes beginning with electrospraying polymers. The concentrations and molecular weight of the PVA were not conducive to electrospinning smooth fibers with small diameters during the previous study of fiber sizes.

Each test with the LaD should be collected by mechanically moving the laser arm, in this way the collected data could be used to measure the extinction coefficients of the deposition. When the LaD is consistently measuring extinction, the calibration curves for individual polymers should be calculated. After the device is fully calibrated, decreasing the amount of time required to measure fibers should be investigated and the LaD should be incorporated into a MIMO feedback control loop.

6. References

[1] J. Zeleny, "The electrical discharge from liquid points, and a hydrostatic method of measuring the electric intensity at their surfaces," *Physical Review,* vol. 3, no. 2, p. 69, 1914.

[2] W. Sigmund *et al.*, "Processing and structure relationships in electrospinning of ceramic fiber systems," *Journal of the American Ceramic Society,* vol. 89, no. 2, pp. 395-407, 2006.

[3] J. s. Kim and D. H. Reneker, "Mechanical properties of composites using ultrafine electrospun fibers," *Polymer composites,* vol. 20, no. 1, pp. 124-131, 1999.

[4] M. M. Bergshoef and G. J. Vancso, "Transparent nanocomposites with ultrathin, electrospun nylon-4, 6 fiber reinforcement," *Advanced Materials,* vol. 11, no. 16, pp. 1362-1365, 1999.

[5] R. Gopal *et al.,* "Electrospun nanofibrous polysulfone membranes as pre-filters: Particulate removal," *Journal of Membrane Science,* vol. 289, no. 1, pp. 210-219, 2007.

[6] T. J. Sill and H. A. von Recum, "Electrospinning: applications in drug delivery and tissue engineering," *Biomaterials*, vol. 29, no. 13, pp. 1989-2006, 2008.

[7] J. Zeng *et al.*, "Biodegradable electrospun fibers for drug delivery," *Journal of Controlled Release,* vol. 92, no. 3, pp. 227-231, 2003.

[8] J. D. Beisel *et al.*, "Lithography via electrospun fibers with quantitative morphology analysis.," *Journal of Vacuum Science & Technology B,* vol. 34, no. 6, 2016.

[9] T. Subbiah, G. S. Bhat, R. W. Tock, S. Parameswaran, and S. S. Ramkumar, "Electrospinning of nanofibers," *Journal of Applied Polymer Science,* vol. 96, no. 2, pp. 557-569, 2005.

[10] H. Pan, L. Li, L. Hu, and X. Cui, "Continuous aligned polymer fibers produced by a modified electrospinning method," *Polymer*, vol. 47, no. 14, pp. 4901-4904, 2006.

[11] S. F. Fennessey and R. J. Farris, "Fabrication of aligned and molecularly oriented electrospun polyacrylonitrile nanofibers and the mechanical behavior of their twisted yarns," *Polymer*, vol. 45, no. 12, pp. 4217-4225, 2004.

[12] E. Zussman, A. Theron, and A. L. Yarin, "Formation of nanofiber crossbars in electrospinning," *Applied Physics Letters,* vol. 82, no. 6, pp. 973-975, 2003.

[13] P. D. Dalton, D. Klee, and M. Möller, "Electrospinning with dual collection rings," *Polymer,* vol. 46, no. 3, pp. 611-614, 2005.

[14] J. L. Skinner *et al.*, "Using electric field manipulation to fabricate nanoscale fibers on large areas: a path to electronic and photonic devices," *SPIE Nanoscience Engineering,* pp. 955302-955302-15, 2015.

[15] J. D. Beisel, J. Kyeremateng, L. Purkett, J. M. Andriolo, and J. L. Skinner, "Analytical parametric model used to study the influence of electrostatic force on surface coverage during electrospinning of polymer fibers," *Journal of Vacuum Science & Technology B,* vol. 32, no. 6, p. 06FI03, 2014.

[16] A. Koski, K. Yim, and S. Shivkumar, "Effect of molecular weight on fibrous PVA produced by electrospinning," *Materials Letters,* vol. 58, no. 3, pp. 493-497, 2004.

[17] J. Doshi and D. H. Reneker, "Electrospinning process and applications of electrospun fibers," in *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, pp. 1698-1703: IEEE.

[18] M. Bognitzki *et al.*, "Nanostructured fibers via electrospinning," *Advanced Materials,* vol. 13, no. 1, pp. 70-72, 2001.

[19] C. Hellmann, J. Belardi, R. Dersch, A. Greiner, J. H. Wendorff, and S. Bahnmueller, "High precision deposition electrospinning of nanofibers and nanofiber nonwovens," *Polymer,* vol. 50, no. 5, pp. 1197-1205, 2009.

[20] T. D. Brown, P. D. Dalton, and D. W. Hutmacher, "Direct writing by way of melt electrospinning," *Advanced Materials,* vol. 23, no. 47, pp. 5651-5657, 2011.

[21] J. M. Deitzel, J. D. Kleinmeyer, J. K. Hirvonen, and N. C. B. Tan, "Controlled deposition of electrospun poly(ethylene oxide) fibers," *Polymer,* vol. 42, no. 19, pp. 8163-8170, 2001.

[22] D. Yang, B. Lu, Y. Zhao, and X. Jiang, "Fabrication of aligned fibrous arrays by magnetic electrospinning," *Advanced Materials,* vol. 19, no. 21, pp. 3702-3706, 2007.

[23] D. Li, Y. Wang, and Y. Xia, "Electrospinning nanofibers as uniaxially aligned arrays and layer-by-layer stacked films," *Advanced Materials,* vol. 16, no. 4, pp. 361-366, 2004.

[24] K. Lee, Y. Han, W. Lee, J. Chung, and C. Lee, "Quantitative measurements of soot particles in a laminar diffusion flame using a LII/LIS technique," *Measurement Science and Technology,* vol. 16, no. 2, p. 519, 2005.

[25] G. W. Mulholland and M. Y. Choi, "Measurement of the mass specific extinction coefficient for acetylene and ethene smoke using the large agglomerate optics facility," in *Symposium (international) on combustion*, vol. 27, pp. 1515-1522: Elsevier.

[26] T. Neill and I. M. Kennedy, "Soot formation in ducted turbulent diffusion flames," *AIAA Journal,* vol. 29, no. 6, pp. 932-935, 1991.

[27] A. R. Jones, "Light scattering for particle characterization," *Progress in Energy and Combustion Science,* vol. 25, no. 1, pp. 1-53, 1999.

[28] S. Pagliara, A. Camposeo, E. Mele, L. Persano, R. Cingolani, and D. Pisignano, "Enhancement of light polarization from electrospun polymer fibers by room temperature nanoimprint lithography," *Nanotechnology,* vol. 21, no. 21, p. 215304, 2010.

[29] H. Xin, O. G. Reid, G. Ren, F. S. Kim, D. S. Ginger, and S. A. Jenekhe, "Polymer nanowire/fullerene bulk heterojunction solar cells: how nanostructure determines photovoltaic properties," *Acs Nano,* vol. 4, no. 4, pp. 1861-1872, 2010.

[30] H. N. Tsao and K. Müllen, "Improving polymer transistor performance via morphology control," *Chemical Society Reviews,* vol. 39, no. 7, pp. 2372-2386, 2010.

[31] Z. Zheng *et al.*, "Uniaxial alignment of liquid-crystalline conjugated polymers by nanoconfinement," *Nano letters,* vol. 7, no. 4, pp. 987-992, 2007.

[32] H. Xin, F. S. Kim, and S. A. Jenekhe, "Highly efficient solar cells based on poly(3-butylthiophene) nanowires," *Journal of the American Chemical Society,* vol. 130, no. 16, pp. 5424-5425, 2008.

[33] T. Kongkhlang, K. Tashiro, M. Kotaki, and S. Chirachanchai, "Electrospinning as a new technique to control the crystal morphology and molecular orientation of polyoxymethylene nanofibers," *Journal of the American Chemical Society,* vol. 130, no. 46, pp. 15460-15466, 2008.

[34] C. Huang *et al.*, "Electrospun polymer nanofibres with small diameters," *Nanotechnology,* vol. 17, no. 6, p. 1558, 2006.

[35] C. E. Ayres *et al.*, "Measuring fiber alignment in electrospun scaffolds: a user's guide to the 2D fast Fourier transform approach," *Journal of Biomaterials Science, Polymer Edition,* vol. 19, no. 5, pp. 603-621, 2008.

[36] J. Tacx, H. M. Schoffeleers, A. G. M. Brands, and L. Teuwen, "Dissolution behavior and solution properties of polyvinylalcohol as determined by viscometry and light scattering in DMSO, ethyleneglycol and water," *Polymer,* vol. 41, no. 3, pp. 947-957, 2000.

[37] S. V. Fridrikh, H. Y. Jian, M. P. Brenner, and G. C. Rutledge, "Controlling the fiber diameter during electrospinning," *Physical Review Letters,* vol. 90, no. 14, p. 144502, 2003.

[38] S. Megelski, J. S. Stephens, D. B. Chase, and J. F. Rabolt, "Micro-and nanostructured surface morphology on electrospun polymer fibers," *Macromolecules,* vol. 35, no. 22, pp. 8456-8466, 2002.

[39] S. A. Theron, E. Zussman, and A. L. Yarin, "Experimental investigation of the governing parameters in the electrospinning of polymer solutions," *Polymer,* vol. 45, no. 6, pp. 2017-2030, 2004.

[40] H. Zhou, T. B. Green, and Y. L. Joo, "The thermal effects on electrospinning of polylactic acid melts," *Polymer,* vol. 47, no. 21, pp. 7497-7505, 2006.

[41] Y. L. Joo and H. Zhou, *Apparatus and method for elevated temperature electrospinning,* 2008.

7. Appendix A: Full-Width Half-Maximum

7.1. Full-Width Half-Max *MATLAB* 2013:

```
function [ wdth, HM, pks ] = FWHM2( deg,amp,varargin )
%This function is designed to calculate the full width half max value of 2D
%FFT data obtained from FIJI using the Oval Profile script to obtain the
%amplitude and the degree.  This function is designed to be a metric of
%fiber alignment.
%   The input values are degrees and amplitude. If the maximum value of the
%   input is greater than 180 degrees, then the data will be rotated until
%   the center of the peak value is at 180 degrees.  After the rotation the
%   FWHM is calculated and the data is plotted to show the measurement. The
%   function has the option to normalize the data to (point/min)-1 as an
%   alignment factor.
clrs = {'r','b','g','m'};
cx = 2;
nopkl = 0;
nopkh = 0;
[pkval,inx] = max(amp);
mn = min(amp);
HM = (pkval-mn)/2+mn;
if isempty(varargin)
    Norm = 'n';
    yax = 'Amplitude of Radial Sums';
elseif strcmpi(varargin,'y')
    Norm = 'y';
    yax = 'Alignment Factor';
end
if strcmpi(Norm,'y');
    amp = (amp./mn)-1;
    HM = (max(amp)-min(amp))/2+min(amp);
end
if deg(inx)>180
    dgsh = deg(inx)-180;
    rt = 540-deg(inx+1);
    nxs = find(deg==rt);
    pad = zeros(nxs,1);
    dega = [pad;deg];
    ampa = [pad;amp];
    for nx = length(deg)+1:length(dega)
        dega(nx-length(deg)) = dega(nx);
        ampa(nx-length(deg)) = ampa(nx);
    end
    amps = ampa(1:length(amp));
    degs = dega(1:length(deg))-dgsh;
    for nx = 1:length(degs)
        if degs(nx)<0
            degs(nx) = degs(nx)+360;
        end
    end
elseif deg(inx)<180
    dgsh = 180-deg(inx);
    rt = 180-deg(inx+1);
    nxs = find(deg==rt);
    pad = zeros(nxs,1);
    dega = [pad;deg];
    ampa = [pad;amp];
    for nx = length(deg)+1:length(dega)
        dega(nx-length(deg)) = dega(nx);
        ampa(nx-length(deg)) = ampa(nx);
    end
    amps = ampa(1:length(amp));
    degs = dega(1:length(deg))+dgsh;
    for nx = 1:length(degs)
        if degs(nx)>=360
            degs(nx) = degs(nx)-360;
        end
    end
end
for nx = ceil(length(degs)/2):-1:0
    if amps(nx) < HM
        anx(1) = nx;
        break;
    end
end
for nx = floor(length(degs)/2):1:length(degs)
    if amps(nx) < HM
        anx(2) = nx;
        break;
    end
end
wdmn = interp1([amps(anx(1)),amps(anx(1)+1)],[degs(anx(1)),degs(anx(1)+1)],HM);
wdmx = interp1([amps(anx(2)),amps(anx(2)-1)],[degs(anx(2)),degs(anx(2)-1)],HM);
wdth = wdmx-wdmn;
wdthx = [wdmn,wdmn,wdmx,wdmx];
wdthy = [min(amps),HM,HM,min(amps)];

pks = degs(find(amps == max(amps(anx(1):anx(2)))));
swd = find(degs == ceil(wdth));

figure(1);
plot(degs,amps,'k',deg,amp,'r');
hold on;
axis([0 360 min(amps) max(amps)+.025*max(amps)]);
ax = gca;
set(ax,'XTick',0:30:360);
title('Data shift to place max amplitude at 180^o');
legend('Shifted Data','Original Data');
xlabel('Alighnment Angle (degrees)');
ylabel(yax);
hold off;

ttlstr = sprintf('FWHM = %3.2f^o, Maximum = %.4f, Half Max = %.4f.',wdth,max(amps),HM);
figure(2);
plot(degs,amps,'k',wdthx,wdthy,'r');
hold on;
axis([0 360 min(amps) max(amps)+.025*max(amps)]);
ax = gca;
set(ax,'XTick', 0:30:360);
title(ttlstr);
xlabel('Alighnment Angle (degrees)');
ylabel(yax);
hold off;


anxm = anx;
wx = 2;
while nopkl == 0;
    [expkl,~] = pkcheck(anx,amps,swd,HM,'l');
    if expkl == 0
        nopkl = 1;
    else
```

```
        for nx = anx(1):-1:1
            if amps(nx) > HM
                anx(2) = nx;
                break;
            end
        end
        for nx = anx(2):-1:1
            if amps(nx) < HM
                anx(1) = nx;
                break;
            end
        end
        wdmn = interp1([amps(anx(1)),amps(anx(1)+1)],[degs(anx(1)),degs(anx(1)+1)],HM);
        wdmx = interp1([amps(anx(2)),amps(anx(2)+1)],[degs(anx(2)),degs(anx(2)+1)],HM);
        wdth(wx) = wdmx-wdmn;
        pks(wx) = degs(find(amps == max(amps(anx(1):anx(2)))));
        wdthx = [wdmn,wdmn,wdmx,wdmx];
        wdthy = [min(amps),HM,HM,min(amps)];
        figure(2);
        hold on;
        if cx > length(clrs)
            cx = cx-length(clrs)
        end
        plot(wdthx,wdthy,clrs{cx});
        hold off;
        wx = wx+1;
        cx = cx+1;
    end
end
anx = anxm;
while nopkh == 0;
    [~,expkh] = pkcheck(anx,amps,swd,HM,'h');
    if expkh == 0
        nopkh = 1;
    else
        for nx = anx(2):length(degs)
            if amps(nx) > HM
                anx(1) = nx;
                break;
            end
        end
        for nx = anx(1):length(degs)
            if amps(nx) < HM
                anx(2) = nx;
                break;
            end
        end
        wdmn = interp1([amps(anx(1)),amps(anx(1)-1)],[degs(anx(1)),degs(anx(1)-1)],HM);
        wdmx = interp1([amps(anx(2)),amps(anx(2)-1)],[degs(anx(2)),degs(anx(2)-1)],HM);
        pks(wx) = degs(find(amps == max(amps(anx(1):anx(2)))));
        wdth(wx) = wdmx-wdmn;
        wdthx = [wdmn,wdmn,wdmx,wdmx];
        wdthy = [min(amps),HM,HM,min(amps)];
        figure(2);
        hold on;
        if cx > length(clrs)
            cx = cx-length(clrs);
        end
        plot(wdthx,wdthy,clrs{cx});
        hold off;
        wx = wx+1;
        cx = cx+1;
    end
end
ttlstr = sprintf('FWHM = %3.2f^o, Maximum = %.4f, Half Max = %.4f.',sum(wdth),max(amps),HM);
figure(2);
hold on;
title(ttlstr);
hold off;
end
```

7.2. Peak Check

```
function [ expkl, expkh ] = pkcheck( anx, amps, swd, HM, hl )
% This function checks for peaks that are close to the main alignment factor peak.
expkl = 0;
expkh = 0;
if strcmpi(hl,'l')
    for nx = anx(1):-1:anx(1)-swd
        if amps(nx) > HM
            expkl = 1;
        end
    end
end
if strcmpi(hl,'h')
    for nx = anx(2):anx(2)+swd
        if amps(nx) > HM
            expkh = 1;
        end
    end
end

end
```

8. Appendix B: Lulzbot TAZ 5 Settings

Table IV: Cura expert mode settings for printing Ninjaflex®.

Basic Tab	Parameter	Value
Quality		
	Layer height (mm)	0.2
	Shell thickness (mm)	0.7
	Enable retraction	Yes
Fill		
	Bottom/Top thickness (mm)	0.6
	Fill density (%)	25
Speed and Temperature		
	Print speed (mm/s)	10
	Printing temperature (C)	265
	Bed temperature (C)	45
Support		
	Support type	None
	Platform adhesion type	None
Filament		
	Diameter (mm)	3
	Flot(%)	125

Advanced Tab	Parameter	Value
Machine		
	Nozzle size (mm)	0.35
Retraction		
	Speed (mm/s)	25
	Distance (mm)	2
Quality		
	Initial layer thickness (mm)	0
	Initial layer line width (%)	125
	Cut off object bottom (mm)	0
	Dual extrusion overlap (mm)	0.15
Speed		
	Travel speed (mm/s)	125
	Bottom layer speed (mm/s)	5
	Infill speed (mm/s)	0
	Outer shell speed (mm/s)	0
	Inner shell speed (mm/s)	0
Cool		
	Minimal layer time (sec)	10
	Enable cooling fan	No

For printing with NinjaFlex® materials the feeding mechanism on the LulzBot® TAZ5 needs to be replaced with continuous support feeding mechanism (Figure 36) with a 0.5 mm nozzle.

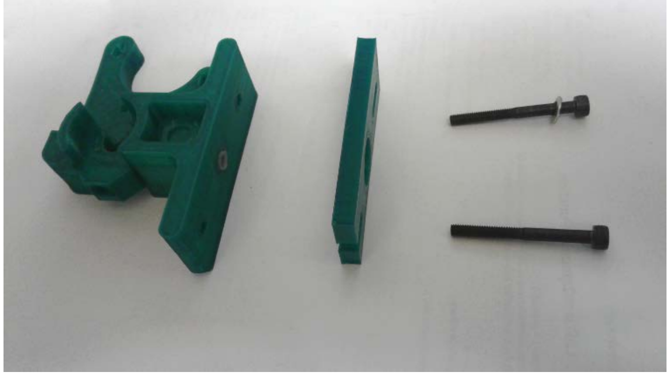

Figure 36: LulzBot(R) Flexystruder print head. This print head offers continual filament support which is required when printing with NinjaFlex®.

9. Appendix C: Microcontroller Code

```
/*
 * main.c
 */
#include  "msp430x22x4.h"
//          Global Variables
int ACounts, AdcCnts = 0, ADCflg = 0, ADCval, adcpar, avgnx = 0, Cdex = 20, chkFlg, ChkStepFlg = 0, Cndx = 0, csFlg = 0,
DCflag = 0, EOT = 0, limFlg = 0, ndx;
int nmbr, nmbrsize, Quant = 0, Rxn = 0, StepCnt = 0, Steps, stppar, TXd[64], txflag = 0, TXndx;
unsigned long int AvgDta = 0;
char input[] = {1,2,3,4,'#',0,0,0,0,'@'}, Code[2], str1[12], str2[12], RXBUF[32];
char id[] = "LaD_Tool", *idcmp[] = {"id","sa","st","fh","ss","ml","cn","sl","vs","vc"}, ermsg[] = "I'm sorry Dave, I can't do
that", home[] = "Home", num[] = "0000", stop[] = "End";

//          Functions
void adc2str (void);
void checkstep (void);
void dccase (void);
void decode (void);
void inputs (void);
void num2str(int nmbr);
void poleLS (void);
void sendTX (char TXd[], int TXndx);
int strcmp (char str1[], char str2[]);
void tnavg (void);

//          Main
void main(void)
{
  WDTCTL = WDTPW + WDTHOLD;                             // Stop WDT
  ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE;                   // ADC10ON, interrupt not enabled
  ADC10AE0 |= 0x01;                                                       // P2.0 ADC option select
  BCSCTL1 = CALBC1_1MHZ;                          // Set DCO
  DCOCTL = CALDCO_1MHZ;
  P1DIR  = 0x01;
  P1OUT  = 0x00;
  P2DIR |= 0x0C;                                                          // P2.3 and P2.2 Outputs step
and direction
  P3SEL  = 0x30;                              // P3.4,5 = USCI_A0 TXD/RXD
  P4DIR  = 0x00;                                                          // P4 set to inputs
  P4OUT  = 0x18;                                                          // P4.3 and P4.4 pullup
  P4REN |= 0x18;                                                          // P4.3 and P4.4 pullup
  TACCTL0 &= ~CCIE;                                                             // TACCR0
interrupt disabled
  TACCR0 = 50000;                                                               // Timing set for
stepper motor control
  TACTL = TASSEL_2 + MC_2;                                                // SMCLK, Contmode
  TBCCTL0 &= ~CCIE;                             // TBCCR0 interrupt disabled
  TBCCR0 = 1000;                                                          // Timing set to 1000Hz for
timer operation
  TBCTL = TBSSEL_2 + MC_1;                        // SMCLK, contmode
  UCA0CTL1 |= UCSSEL_2;                           // SMCLK
  UCA0BR0 = 8;                              // 1MHz 115200
  UCA0BR1 = 0;                              // 1MHz 115200
  UCA0MCTL = UCBRS2 + UCBRS0;                     // Modulation UCBRSx = 5
  UCA0CTL1 &= ~UCSWRST;                           // **Initialize USCI state machine**
  IE2 |= UCA0RXIE;                          // Enable USCI_A0 RX interrupt
  __bis_SR_register(GIE);                                       // Interrupts enabled
  while(1){                                                                           // Infinite
while loop
```

```
        poleLS();                                                                       // Check
Limit swtich values
        if(txflag == 1){                                                   // If ADC value is ready to
send
                adc2str();                                                         // Convert ADC
value to a string for sending
                sendTX(input, 10);                                         // Send the ADC value in
1234# format
        }
        if(EOT == 1){                                                              // If Incoming code
has finished transmission
                Cndx = 0;
                inputs();                                                          // Parse inputs into
#code:number. where code is a two letter code and number applies to the given code
        }
        if(DCflag == 1){                                                   // If the input is ready to
decode
                decode();                                                                  // decode
the two letter code and determine which switch case applies to the given code if exists
                DCflag = 0;
        }
        if(csFlg == 1){
                dccase();                                                                  // execute
the specified switch case
                csFlg = 0;
        }
        if(ChkStepFlg == 1){
                checkstep();
        }
  }
}


// INTERRUPTS

// Read coded values for uC operations
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
        RXBUF[Rxn] = UCA0RXBUF;                                                    // Place incoming
characters into a buffer
        Rxn++;                                                                             //
Increment buffer index
        if(RXBUF[Rxn-1] == '.'){                                   // If a . is sent
                EOT = 1;                                                                   // Flag end of
transmission
        }
}

// Timer A0 interrupt service routine for moving stepper motor
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
                P2OUT ^= 0x08;                  // Toggle P2.3 to step motor
                if ((P2OUT & 0x08) == 0x08){
                        TACCR0 += 500;          // Add offset to TACCR0
                        StepCnt++;                                                         //
Increment step count
                }
                else{
                        TACCR0 += 2500;
                }
```

```
}

// Timer B0 interrupt service routine for sending ADC values at 100 Hz
#pragma vector=TIMERB0_VECTOR
__interrupt void Timer_B (void)
{
        ADC10CTL0 |= ENC + ADC10SC;                     // Sampling and conversion start
        TBCCR0 = 1000;
}

// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
        tnavg();
}


// FUNCTIONS
void adc2str (void){
        volatile unsigned int numndx;
        input[0] = (ADCval/1000);
        input[1] = (ADCval/100)-input[0]*10;
        input[2] = (ADCval/10)-input[0]*100-input[1]*10;
        input[3] = (ADCval/1)-input[0]*1000-input[1]*100-input[2]*10;
        input[5] = (StepCnt/1000);
        input[6] = (StepCnt/100)-input[5]*10;
        input[7] = (StepCnt/10)-input[5]*100-input[6]*10;
        input[8] = (StepCnt/1)-input[5]*1000-input[6]*100-input[7]*10;
        for (numndx = 0; numndx <= 3; numndx++){
                input[numndx] = input[numndx]+48;
                input[numndx+5] = input[numndx+5]+48;
        }
}

void checkstep (void){
        if ((StepCnt >= Steps) && (adcpar == 0)){
                TACCTL0 &= ~CCIE;
                ADCflg = 1;
                if(AdcCnts >= ACounts){
                        ADCflg = 0;
                        AdcCnts = 0;
                        StepCnt = 0;
                        chkFlg = 1;
                }
        }
        else if ((chkFlg == 1) || (adcpar == 1)){
                TACCTL0 = CCIE;
                chkFlg = 0;
        }
}

void dccase (void){
        // int txdflag = 0;
        switch (Cdex){
        case 0:  // Identify
                sendTX(id,8);                                                   // Send device
identification
                break;
        case 1:  // Start ADC
                TBCCTL0 |= CCIE;                                                // Start Sending
values Timer B interrupt enabled
```

```
                adcpar = 0;                                                              // ADC is
sampling
                break;
        case 2:  // Stop ADC
                TBCCTL0 &= ~CCIE;                                                // Stop Sending
values Timer B interrupt disabled
                adcpar = 1;                                                              // ADC is
stopped
                break;
        case 3:  // Find Home
                ChkStepFlg = 0;
                P2OUT &= ~0x04;                                                  // Set Stepper
direction to move stage away from stepper motor
                TACCTL0 |= CCIE;                                                 // Timer A interrupt
enabled start stepping motor
                break;
        case 4:  // Set Step Size
                Steps = Quant;                                                   // Set number of
steps equal to sent value
                Quant = 0;                                                               // Clear
Quant
                break;
        case 5:  // Move Stage
                P2OUT |= 0x04;                                                   // Direction bit
                ChkStepFlg = 1;                                                  // Count steps for
precise movement
                stppar = 0;                                                              // Stepper
is running
                TACCTL0 |= CCIE;                                                 // Start motor
timing
                break;
        case 6:  // Set Increments
                ACounts = Quant;
                Quant = 0;
                break;
        case 7:  // Stop Stage
                TACCTL0 &= ~CCIE;                                                // Pause motor
timing
                ChkStepFlg = 0;                                                  // Stop counting
steps
                stppar = 1;                                                              // Stepper
is stopped
                break;
        case 8:     // Verify Step Size
                num2str(Steps);
                sendTX(num,4);
                break;
        case 9:  // Verify Increments
                num2str(ACounts);
                sendTX(num,4);
                break;
        case 13:  // When the end position limit switch is activated
                if ((P2OUT & 0x04) == 0x04){
                        TACCTL0 &= ~CCIE;
                }
                if(limFlg == 0){
                        sendTX(home,3);
                        limFlg = 1;
                }
                break;
        case 14:  // When the home limit switch is activated
                if ((P2OUT & 0x04) == 0x00){
```

```
            TACCTL0 &= ~CCIE;
        }
        if(limFlg == 0){
            sendTX(stop,4);
            limFlg = 1;
            StepCnt = 0;
        }
        break;
    case 19:  // Bad Code error
        sendTX(ermsg,31);                                       // Send error message if bad
code is received
        sendTX(Code,2);
        break;
    }
    DCflag = 0;
    Cdex = 20;
}

void decode (void){
    int cndx = 0, dc = 0;
    while (dc == 0){
        dc = strcmp(Code,idcmp[cndx]);
        cndx++;
        Cdex = cndx-1;
        if (cndx > 10){
            Cdex = 19;
            csFlg = 1;
            return;
        }
    }
    csFlg = 1;
}

void inputs (void){
    volatile unsigned int CBgn, QBgn, inx;
    for (inx = 0; inx < Rxn; inx++){
        if (RXBUF[inx] == '.'){
            EOT = 0;
            Rxn = 0;
            CBgn = 0;
            QBgn = 0;
            DCflag = 1;
//          limFlg = 0;
            break;
        }
        else if (RXBUF[inx] == '#'){
            CBgn = 1;
        }
        else if (RXBUF[inx] == ':'){
            CBgn = 0;
            QBgn = 1;
        }
        else if (CBgn == 1){
            Code[Cndx] = RXBUF[inx];
            Cndx++;
        }
        else if (QBgn == 1){
            Quant = (Quant*10)+(RXBUF[inx]-48);
        }
    }
}
```

```
void num2str(int nmbr){
        int nsndx;
        num[0] = (nmbr/1000);
        num[1] = (nmbr/100)-num[0]*10;
        num[2] = (nmbr/10)-num[0]*100-num[1]*10;
        num[3] = (nmbr/1)-num[0]*1000-num[1]*100-num[2]*10;
        for (nsndx = 0; nsndx <=3; nsndx++){
                num[nsndx] = num[nsndx]+48;
        }
}

void poleLS(void){
        if((P4IN & 0x10) != 0x10){
                Cdex = 13;
                csFlg = 1;
        }
        else if((P4IN & 0x08) != 0x08){
                Cdex = 14;
                csFlg = 1;
        }
        else{
                limFlg = 0;
        }
}

void sendTX (char TXd[], int TXndx){
        int tndx;
        txflag = 0;
        for (tndx = 0; tndx<TXndx; tndx++){
                UCA0TXBUF = TXd[tndx];
                __delay_cycles(100);
        }
        if(ADCflg == 1){
                AdcCnts++;
        }
}

int strcmp(char str1[], char str2[]){
        volatile unsigned int stndx;
        for (stndx = 0; stndx < 2; stndx++){
                if (str1[stndx] != str2[stndx]){
                        return 0;
                }
        }
        return 1;
}

void tnavg(void){
        AvgDta = AvgDta + ADC10MEM;
        avgnx++;
        if (avgnx == 10){
                ADCval = AvgDta/(avgnx-1);
                avgnx = 0;
                AvgDta = 0;
          txflag = 1;
        }
}
```

10. Appendix D: *MATLAB* 2013 Code

10.1. Laser Diagnostics (LaD) User Interface (UI):

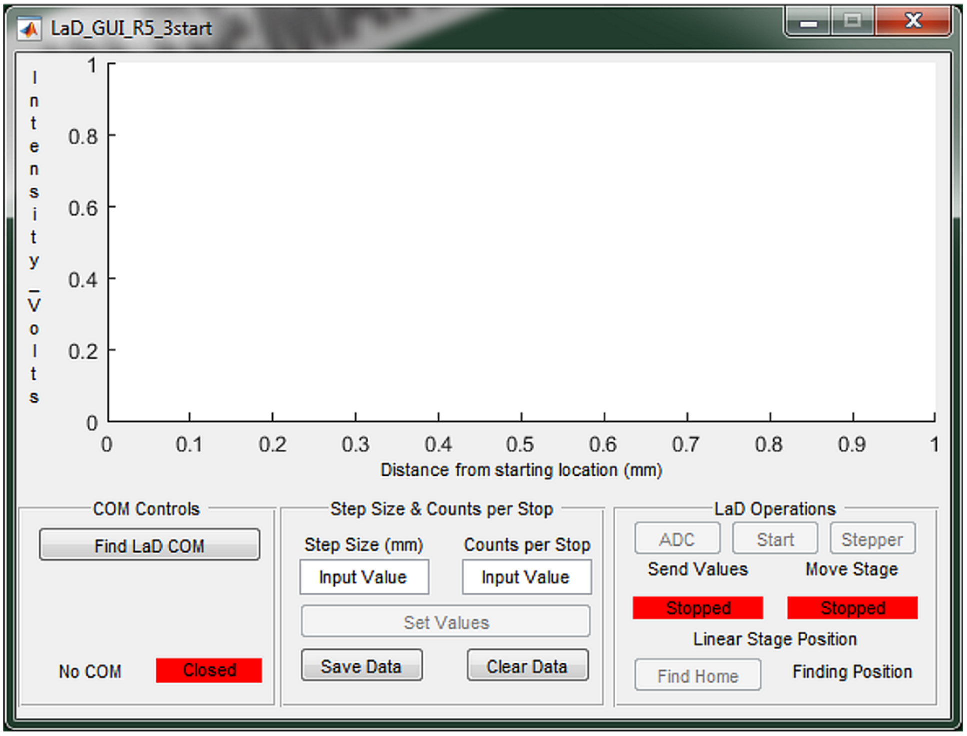

Figure 30: UI for the LaD system. The interface uses push buttons and user defined inputs to scan through the ES deposition. Buttons are available sequentially to guide the user through the operation of the LaD tool.

10.2. LaD UI Program:

```
function varargout = LaD_GUI_R5_3start(varargin)
% LAD_GUI_R5_3START MATLAB code for LaD_GUI_R5_3start.fig
%      LAD_GUI_R5_3START, by itself, creates a new LAD_GUI_R5_3START or raises the existing
%      singleton*.
%
%      H = LAD_GUI_R5_3START returns the handle to a new LAD_GUI_R5_3START or the handle to
%      the existing singleton*.
%
%      LAD_GUI_R5_3START('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in LAD_GUI_R5_3START.M with the given input arguments.
%
%      LAD_GUI_R5_3START('Property','Value',...) creates a new LAD_GUI_R5_3START or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before LaD_GUI_R5_3start_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to LaD_GUI_R5_3start_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 LaD_GUI_R5_3start
% Last Modified by GUIDE v2.5 13-Apr-2016 12:42:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @LaD_GUI_R5_3start_OpeningFcn, ...
                   'gui_OutputFcn',  @LaD_GUI_R5_3start_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   '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 LaD_GUI_R5_3start is made visible.
function LaD_GUI_R5_3start_OpeningFcn(hObject, eventdata, 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 LaD_GUI_R5_3start (see VARARGIN)

% Choose default command line output for LaD_GUI_R5_3start
handles.output = hObject;




%------------------------------ Timer setup ------------------------------


global LaDCTimer;
LaDCTimer = timer('Period',0.75);                    % Set Period to .5 sec
LaDCTimer.ExecutionMode = 'fixedRate';               % When timer is started it automatically runs
LaDCTimer.TimerFcn = @(~,~)UpdatePlot(hObject, eventdata, handles);             % Read ADC values

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes LaD_GUI_R5_3start wait for user response (see UIRESUME)
% uiwait(handles.LaDGUI);


% --- Outputs from this function are returned to the command line.
function varargout = LaD_GUI_R5_3start_OutputFcn(hObject, eventdata, 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;
```

```
    %%%%%%%%%%%%%%%%%%%%%%%% BUTTON CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% --- Executes on button press in AutoCOM.
function AutoCOM_Callback(hObject, eventdata, handles)
% hObject    handle to AutoCOM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud;
[com, baud] = AutoCOM();
if strcmp(com,'LaD is not detected. Check your connections and try again.')
    set(handles.COMStr,'String','No LaD');
    set(handles.openCOM,'Visible','off');
    set(handles.closeCOM,'Visible','off');
else
    set(handles.COMStr,'String',com);
    set(handles.openCOM,'Visible','on');
    set(handles.closeCOM,'Visible','on');
end
s = serial(com,'BaudRate',baud,'InputBufferSize',10000);
guidata(hObject, handles);



% --- Executes on button press in openCOM.
function openCOM_Callback(hObject, eventdata, handles)
% hObject    handle to openCOM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud;
if strcmp(com(1:3),'COM')
    fopen(s);
    if strcmp(s.Status,'open')
        set(handles.OpenCloseID,'String','Open');
        set(handles.OpenCloseID,'BackgroundColor','g');
        set(handles.SSCpSSet,'Enable','on');
    end
end


% --- Executes on button press in closeCOM.
function closeCOM_Callback(hObject, eventdata, handles)
% hObject    handle to closeCOM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud;
if strcmp(com(1:3),'COM')
    fclose(s);
    if strcmp(s.Status,'closed')
        set(handles.OpenCloseID,'String','Closed');
        set(handles.OpenCloseID,'BackgroundColor','r');
        set(handles.SSCpSSet,'Enable','off');
        set(handles.LStageBtn,'Enable','off');
        set(handles.StartADC,'Enable','off');
        set(handles.StartStage,'Enable','off');
        set(handles.FindHomeBtn,'Enable','off');
    end
end

% --- Executes on button press in SSCpSSet.
function SSCpSSet_Callback(hObject, eventdata, handles)
% hObject    handle to SSCpSSet (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud ss;
ssmm = get(handles.SSInput,'String');
cps = get(handles.CpSInput,'String');
if strcmp(ssmm,'Input Value')||isempty(ssmm)
    toerr = errordlg('A value needs to be assigned to the field: "Step Size"');
    uiwait(toerr);
elseif strcmp(cps,'Input Value')||isempty(cps)
    toerr = errordlg('A value needs to be assigned to the field: "Counts per Stop"');
    uiwait(toerr);
else
    ssmm = str2double(ssmm);
    cps = str2double(cps);
    ss = mm2steps(ssmm,'f');
    fprintf(s,'#ss:%d.',ss);
    pause(.05);
    fprintf(s,'#cn:%d.',cps);
    pause(.05);
    [vss, vcps] = VrfySSCpS( ss, cps, s );
    if vss == 1 && vcps == 1
        set(handles.LStageBtn,'Enable','on');
        set(handles.StartADC,'Enable','on');
        set(handles.StartStage,'Enable','on');
        set(handles.FindHomeBtn,'Enable','on');
    end
end


% --- Executes on button press in LStageBtn.
function LStageBtn_Callback(hObject, eventdata, handles)
% hObject    handle to LStageBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud LaDCTimer;
stsp = get(handles.LStageBtn,'String');
if strcmp(stsp,'Start')
    fprintf(s,'#ml.');
    fprintf(s,'#sa.');
    start(LaDCTimer);
    set(handles.StartADC,'Enable','off');
    set(handles.StartStage,'Enable','off');
    set(handles.LStageBtn,'String','Stop');
    set(handles.LStageState,'String','Moving');
    set(handles.LStageState,'BackgroundColor','g');
    set(handles.SendValState,'String','Reading LaD');
    set(handles.SendValState,'BackgroundColor','g');
elseif strcmp(stsp,'Stop');
    stop(LaDCTimer);
    fprintf(s,'#sl.');
    pause(0.01);
    fprintf(s,'#st.');
    set(handles.StartADC,'Enable','on');
    set(handles.StartStage,'Enable','on');
    set(handles.LStageBtn,'String','Start');
    set(handles.LStageState,'String','Stopped');
    set(handles.LStageState,'BackgroundColor','r');
    set(handles.SendValState,'String','Stopped');
    set(handles.SendValState,'BackgroundColor','r');
end
UpdatePlot(hObject, eventdata, handles);



% --- Executes on button press in StartADC.
function StartADC_Callback(hObject, eventdata, handles)
% hObject    handle to StartADC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud LaDCTimer;
stsp = get(handles.StartADC,'String');
if strcmpi(stsp,'adc')
    fprintf(s,'#sa.');
    start(LaDCTimer);
    set(handles.StartADC,'String','Stop');
    set(handles.SendValState,'String','Reading LaD');
    set(handles.SendValState,'BackgroundColor','g');
elseif strcmpi(stsp,'Stop');
    stop(LaDCTimer);
    fprintf(s,'#st.');
    set(handles.StartADC,'String','ADC');
    set(handles.SendValState,'String','Stopped');
    set(handles.SendValState,'BackgroundColor','r');
end
UpdatePlot(hObject, eventdata, handles);

% --- Executes on button press in StartStage.
function StartStage_Callback(hObject, eventdata, handles)
% hObject    handle to StartStage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud LaDCTimer;
stsp = get(handles.StartStage,'String');
if strcmpi(stsp,'stepper')
    fprintf(s,'#ml.');
    set(handles.StartStage,'String','Stop');
    set(handles.LStageState,'String','Moving');
    set(handles.LStageState,'BackgroundColor','g');
elseif strcmpi(stsp,'Stop');
    stop(LaDCTimer);
    fprintf(s,'#sl.');
    set(handles.StartStage,'String','Stepper');
    set(handles.LStageState,'String','Stopped');
    set(handles.LStageState,'BackgroundColor','r');
end
UpdatePlot(hObject, eventdata, handles);

% --- Executes on button press in FindHomeBtn.
function FindHomeBtn_Callback(hObject, eventdata, handles)
% hObject    handle to FindHomeBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s com baud;
flushinput(s);
fprintf(s,'#fh.');
home = isHome();
```

```
set(handles.PositionVal,'String',home);




% --- Executes on button press in SaveBttn.
function SaveBttn_Callback(hObject, eventdata, handles)
% hObject    handle to SaveBttn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global vtstotal lctntotal filtvts SaveFlg;
if isempty(lctntotal)
   errordlg('This session currently does not contain any data.  Collect some data then try again.','What are you
thinking?');
else
   saved = 0;
   cellhead = {'MM','Raw','Filtered'};
   xdist = lctntotal';
   vtsraw = vtstotal';
   fvts = filtvts';
   xrng = sprintf('A2:A%d',length(xdist)+1);
   rrng = sprintf('B2:B%d',length(vtsraw)+1);
   frng = sprintf('C2:C%d',length(fvts)+1);
   while saved == 0
      flnm = inputdlg('Filepath and name','Save As:');
      flnm = sprintf('%s.xls',flnm{1});
      if exist(flnm,'file') ~= 0
         ovwrt = questdlg('This file already exists. Would you like to overwrite it?','File Exists','Yes','No','No');
         %uiwait(ovwrt);
      end
      if exist(flnm,'file') == 0 || strcmpi(ovwrt,'yes')
         xlswrite(flnm,cellhead);
         xlswrite(flnm,xdist,xrng);
         xlswrite(flnm,vtsraw,rrng);
         xlswrite(flnm,fvts,frng);
         saved = 1;
      end
   end
end
SaveFlg = 1;



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

global SaveFlg vtstotal lctntotal filtvts mfiltvts pltflg preloc rem dloc;
if isempty(SaveFlg)
   SaveFlg = 0;
end
if SaveFlg == 0 && isempty(lctntotal)==0
   sve = questdlg('Your data has not been saved, do you still want to erase the current data?','Data Not
Saved','Erase','Cancel','Cancel');
elseif SaveFlg == 1
   sve = questdlg('This will clear all of the current data, do you wish to continue?','Erase Data','Erase','Cancel','Cancel');
elseif isempty(lctntotal)==1
   errordlg('There is no data to be erased.','No Data Found');
   sve = 'no data';
end
```

```
if strcmpi(sve,'erase');
    dloc = [];
    pltflg = [];
    preloc = [];
    vtstotal = [];
    lctntotal = [];
    filtvts = [];
    mfiltvts = [];
    SaveFlg = 0;
    rem = [];
    axes(handles.VoltPlot);
    cla;
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TEXT INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%




function SSInput_Callback(hObject, eventdata, handles)
% hObject    handle to SSInput (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 SSInput as text
%        str2double(get(hObject,'String')) returns contents of SSInput as a double


% --- Executes during object creation, after setting all properties.
function SSInput_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SSInput (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 CpSInput_Callback(hObject, eventdata, handles)
% hObject    handle to CpSInput (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 CpSInput as text
%        str2double(get(hObject,'String')) returns contents of CpSInput as a double


% --- Executes during object creation, after setting all properties.
function CpSInput_CreateFcn(hObject, eventdata, handles)
% hObject    handle to CpSInput (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


% --- Executes when user attempts to close LaDGUI.
function LaDGUI_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to LaDGUI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
global LaDCTimer s
if strcmpi(LaDCTimer.running,'on')
    stop(LaDCTimer);
end
delete(LaDCTimer);
delete(s);
clearvars -global;
disp('Thank you for using the LaD');
pause(0.1);
clc;
delete(hObject);
clear all;


% --- Executes during object deletion, before destroying properties.
function LaDGUI_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to LaDGUI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function [  ] = UpdatePlot( hObject, eventdata, handles )
%This function reads the ADC values from the LaD microcontroller,
%interprets the data into voltage and location.  The newest measurements
%are concatenated with previous values in the global variables vtstotal (a
%vector containing all of the currently measured voltages), and lctntotal
%(a vector that contains the travelled distance of the LaD linear stage.

global vtstotal filtvts lctntotal rem s pltflg ss preloc dloc;
stgmv = get(handles.LStageState,'String');
if isempty(pltflg)
    pltflg = 3;
end
if isempty(preloc)
    preloc = 0;
end
bits = s.BytesAvailable;
while bits == 0
    bits = s.BytesAvailable;
end
adc = fread(s,bits);
adcraw = [rem,adc'];
vlt = strfind(adcraw,'#');
loc = strfind(adcraw,'@');
vts = zeros(1,length(vlt));
lctn = zeros(1,length(loc));
if isempty(vlt)==0 && isempty(loc)==0
    for vndx = 1:length(vlt);
        vts(vndx) = str2double(char(adcraw(vlt(vndx)-4),adcraw(vlt(vndx)-3),adcraw(vlt(vndx)-2),adcraw(vlt(vndx)-1)));
    end
    for lndx = 1:length(loc);
        lctn(lndx) = str2double(char(adcraw(loc(lndx)-4),adcraw(loc(lndx)-3),adcraw(loc(lndx)-2),adcraw(loc(lndx)-1)));
    end
    if loc(end)>vlt(end)
        rem = [adcraw(loc(end)+1:length(adcraw))];
    elseif loc(end)<vlt(end)
        rem = [adcraw(vlt(end)+1:length(adcraw))];
    elseif loc(end)==length(adcraw) || vlt(end)==length(adcraw)
        rem = [];
    end
    vts = vts.*(3.3/1023);
    % lctn = mm2steps(lctn,'r');
    vtstotal = [vtstotal, vts];
    % lctntotal = [lctntotal, lctn];
    tlend = length(lctntotal);
    if lctn(1) >= preloc
        dloc = lctn(1)-preloc;
    elseif preloc > lctn(1)
        if ss >= preloc
            dloc = ss-preloc+lctn(1);
        elseif ss < lctn(1)
            dloc = preloc-ss+lctn(1);
        end
    end
    if isempty(lctntotal)
        lctntotal = mm2steps(dloc,'r');
    else
        lctntotal(tlend+1) = lctntotal(tlend)+mm2steps(dloc,'r');
    end
    preloc = lctn(end);
    for tlndx = 2:length(loc)
        if lctn(tlndx) >= lctn(tlndx-1)
            dloc = lctn(tlndx) - lctn(tlndx-1);
        elseif lctn(tlndx)<lctn(tlndx-1)
            if ss >= lctn(tlndx-1)
                dloc = ss-lctn(tlndx-1)+lctn(tlndx);
            elseif ss < lctn(tlndx-1)
                dloc = lctn(tlndx-1)-ss+lctn(tlndx);
            end
        end
        lctntotal(tlend+tlndx) = lctntotal(tlend+tlndx-1)+mm2steps(dloc,'r');
    end
    LaDMedfilt();
    pltflg = pltflg-1;
    if pltflg == 0
        if strcmpi(stgmv,'stopped') == 0
            xlbl = get(handles.text12,'String');
            if strcmpi(xlbl,'Data Point');
                set(handles.text12,'String','Distance from starting location (mm)');
            end
            hold(handles.VoltPlot,'on');
            plot(handles.VoltPlot,lctntotal,vtstotal,'k.','MarkerSize',10);
            plot(handles.VoltPlot,lctntotal(1:length(filtvts)),filtvts,'r.','MarkerSize',5);
            hold(handles.VoltPlot,'off');
            pltflg = 3;
        elseif strcmpi(stgmv,'stopped') == 1
            xlbl = get(handles.text12,'String');
            if strcmpi(xlbl,'Distance from starting location (mm)');
                set(handles.text12,'String','Data Point');
            end
            hold(handles.VoltPlot,'on');
            plot(handles.VoltPlot,vtstotal,'k.','MarkerSize',10);
            plot(handles.VoltPlot,filtvts,'r.','MarkerSize',5);
```

```
            hold(handles.VoltPlot,'off');
            pltflg = 3;
        end
    end
    set(handles.PositionVal,'String',lctntotal(end));
end

function [  ] = LaDMedfilt( )
%median filter specifically designed to filter the incoming data from the
%LaD tool

global vtstotal filtvts mfiltvts;
medord = 25;                    % 25th order median filter
hord = floor(medord/2);
mnord = 5;                      % 5th order mean filter
hnord = floor(mnord/2);
if length(vtstotal)>medord
    if isempty(filtvts)
        for inx = hord+1:medord
            mfiltvts(inx-hord) = median(vtstotal(1:inx));
        end
        for nnx = hnord+1:mnord
            filtvts(nnx-hnord) = mean(mfiltvts(1:nnx));
        end
    end
    sln = length(mfiltvts)+1;
    for ndx = sln+hord:length(vtstotal)
        mfiltvts(ndx-hord) = median(vtstotal(ndx-medord+1:ndx));
    end
    mln = length(filtvts)+1;
    for mndx = mln+hnord:length(mfiltvts)
        filtvts(mndx-hnord) = mean(mfiltvts(mndx-mnord+1:mndx));
    end
end
```

10.3. LaD AutoCOM.m Function:

```
function [ com, baud ] = AutoCOM(  )
%This function automatically identifies the correct COM port for the LaD
%device.
%   The function retrieves the active COM ports then sends the code #id. to
%   each COM port at a baud rate of 115200, the correct COM port will return a string for
%   comparison. The output is the COM port (in form 'COM#') that is
%   connected to the LaD device.
    inx = 1;
    baud = 115200;
    [stat, cmd] = dos('mode');
    comc = strfind(cmd,'COM');
    endcom = strfind(cmd,':');
    CPRT = cell(1,length(comc));
    comf = ones(1:length(comc));
    for ndx = 1:length(endcom)
        if endcom(ndx)-comc(inx)<=10 && endcom(ndx)-comc(inx)>0
            comf(inx) = endcom(ndx);
            inx = inx+1;
            if inx>length(comc)
                break;
            end
        end
    end
    for ndx = 1:length(comc)
        CPRT{ndx} = cmd(comc(ndx):comf(ndx)-1);
    end
    for ndx = 1:length(CPRT)
        s = serial(CPRT{ndx},'BaudRate',baud);
        fopen(s);
        flushinput(s);
        fprintf(s,'#id.');
        pause(.1);
        if (s.BytesAvailable ~= 0)
            comtest = fread(s,s.BytesAvailable);
        else
            comtest = 'Not the COM your looking for';
        end
        fclose(s);
        comtest = char(comtest)';
        if strcmp(comtest,'LaD_Tool');
            com = CPRT{ndx};
            return;
        else
com = 'LaD is not detected. Check your connections and try again.';
        end
    end
end
```

10.4. LaD VrfySSCpC.m Function:

```
function [ stf, ctf ] = VrfySSCpS( ss, cps, s )
%This function will verify that the step size and the counts per step are
%set correctly with the LaD.
%   Before the user is allowed to start or stop the LaD device a
%   verification of step size and counts being set must be obtained. s is
%   the serial port, ss is the step size in steps, cps
vsbit = 0;
vcbit = 0;
fprintf(s,'#vs.');
while vsbit ==  0
    vsbit = s.BytesAvailable;
end
vss = fread(s,vsbit);
fprintf(s,'#vc.');
while vcbit == 0
    vcbit = s.BytesAvailable;
end
vcps = fread(s,vcbit);
vss = str2double(char(vss'));
vcps = str2double(char(vcps'));
stf = ss==vss;
ctf = cps==vcps;
end
```

10.5. LaD isHome.m Function:

```
function [ hstate ] = isHome( )
%This function checks the state of the linear stage to verify that the
%stage is in the home position before continuing the program.
global s
bits = 0;
while bits == 0
    bits = s.BytesAvailable;
end
hstate = char(fread(s,bits)');
end
```

10.6. LaD mm2steps.m Function

```
function [ hstate ] = isHome( )
%This function checks the state of the linear stage to verify that the
%stage is in the home position before continuing the program.
global s
bits = 0;
while bits == 0
    bits = s.BytesAvailable;
end
hstate = char(fread(s,bits)');
end
```

SIGNATURE PAGE

This is to certify that the thesis prepared by Joshua D. Beisel entitled "Deposition control of electrospun fibers" has been examined and approved for acceptance by the Department of Electrical Engineering, Montana Tech of The University of Montana, on this 14th day of December, 2016.

Jack L. Skinner, PhD, Associate Professor
Department of Electrical Engineering
Chair, Examination Committee

Bryce Hill, PhD, Assistant Professor
Department of Electrical Engineering
Member, Examination Committee

Curtis Link

Curtis Link, PhD, Director
Department of Freshman Engineering Program
Member, Examination Committee