

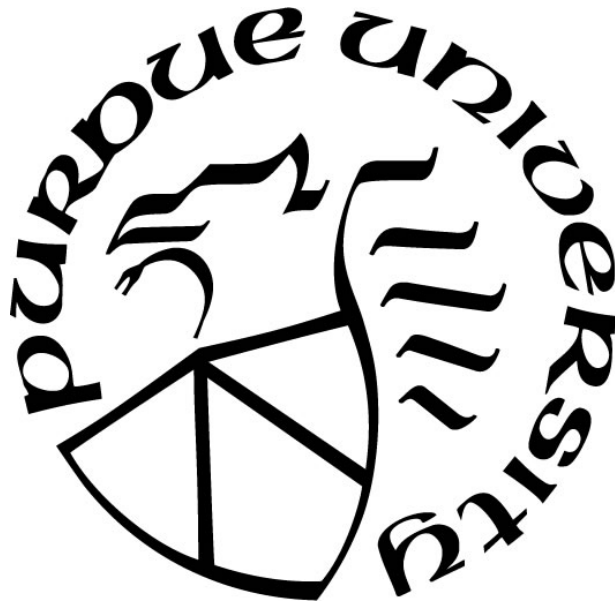
**PIEZOELECTRIC INKJET PRINTED ALUMINUM BISMUTH (III)  
OXIDE: THE EFFECTS OF PRINTING PARAMETERS ON BURNING  
RATE**

by  
**Forrest J. Son**

**A Thesis**

*Submitted to the Faculty of Purdue University  
In Partial Fulfillment of the Requirements for the degree of*

**Master of Science in Mechanical Engineering**



School of Mechanical Engineering  
West Lafayette, Indiana  
December 2018

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF COMMITTEE APPROVAL**

Dr. George T.-C Chiu, Chair

School of Mechanical Engineering

Dr. Jeffrey F. Rhoads

School of Mechanical Engineering

Dr. I. Emre Gunduz

School of Mechanical Engineering

**Approved by:**

Dr. Jay P. Gore

Head of the Graduate Program

*Dedicated to my wife, Karen, and my parents, Steve and Rosemary.*

## **ACKNOWLEDGMENTS**

This research is supported by the U.S. Department of Defense, Defense Threat Reduction Agency through grant No. HDTRA1-15-1-0010 and is managed by Dr. Allen Dalton. The content of the information does not necessarily reflect the position or the policy of the U.S. federal government, and no official endorsement should be inferred.

## TABLE OF CONTENTS

LIST OF TABLES.....	8
LIST OF FIGURES .....	9
ABSTRACT.....	11
CHAPTER 1. INTRODUCTION .....	12
1.1 Background Information .....	12
1.2 Piezoelectric Inkjet Printing Performance.....	13
1.3 Burning Rate Studies.....	14
1.4 Analytical Model of Burning Rate Trends .....	14
CHAPTER 2. PIEZOELECTRIC INKJET PRINTING PERFORMANCE .....	15
2.1 Introduction .....	15
2.2 Thermite Ink Preparation.....	16
2.3 Thermite Ink Printing .....	17
2.4 Printing Performance.....	18
2.5 Results .....	21
2.5.1 Droplet Side View Imaging.....	21
2.5.2 Printed Droplet Grid.....	22
2.5.3 Droplet-Droplet Interaction.....	22
2.6 Conclusions .....	23
CHAPTER 3. BURNING RATE STUDIES .....	25
3.1 Introduction .....	25
3.2 Experimental Methods.....	25
3.2.1 Sample Thickness Measurement.....	26
3.2.2 Burning Rate Studies.....	26
3.2.3 Statistical Analysis of Results .....	28
3.3 Results .....	29
3.3.1 Measured Sample Thickness .....	29
3.3.2 One Droplet Wide Samples Burning Rate Versus Pixel Pitch.....	35
3.3.3 One Droplet Wide Samples Burning Rate Versus Sample Thickness .....	38
3.3.4 y-Direction Pixel Pitch Versus Burning Rate.....	38
3.3.5 1.9 mm Wide Samples (y-Direction Pixel Pitch) Thickness Versus Burning Rate	40

3.3.6	Burning Rate Versus Sample Width .....	40
3.4	Conclusions .....	42
CHAPTER 4.	ANALYTICAL THERMAL MODEL .....	43
4.1	Introduction .....	43
4.2	Model Description .....	43
4.2.1	Energy Balance .....	43
4.2.2	Model Inputs .....	46
4.2.3	Solution Methods .....	48
4.3	Model Results .....	49
4.4	Conclusions .....	50
CHAPTER 5.	CONCLUSIONS .....	52
5.1	Conclusions .....	52
5.2	Future Work.....	53
APPENDIX A.	DESCRIPTION OF THE MATLAB CODE USED TO MEASURE DROPLET DIAMETER FROM SIDEVIEW IMAGES .....	55
APPENDIX B.	MAIN MATLAB CODE FOR MEASURING DROPLET DIAMETER FROM SIDEVIEW IMAGES .....	56
APPENDIX C.	DESCRIPTION OF THE MATLAB CODE USED TO MEASURE THE PRINTED DROPLET DIAMETER AND SPACING.....	58
APPENDIX D.	MAIN MATLAB CODE FOR MEASURING PRINTED DROPLET DIAMETER AND SPACING.....	59
APPENDIX E.	DESCRIPTION OF THE MATLAB CODE USED TO MEASURE THE AREA OF A PRINTED THERMITE SAMPLE .....	63
APPENDIX F.	MAIN MATLAB CODE TO MEASURE THE AREA OF A PRINTED THERMITE SAMPLE .....	65
APPENDIX G.	MATLAB FUNCTION TO BINARIZE AN IMAGE OF PRINTED THERMITE.....	68
APPENDIX H.	A DESCRIPTION OF THE MATLAB CODE USED TO MEASURE BURNING RATE .....	69
APPENDIX I.	The MAIN MATLAB CODE USED FOR MEASURING BURNING RATE FROM A SINGLE VIDEO .....	73

APPENDIX J. THE MAIN MATLAB CODE USED FOR MEASURING THE BURNING RATE FROM ALL OF THE VIDEOS IN A DIRECTORY .....	74
APPENDIX K. THE MATLAB SUBFUNCTION FOR MEASURING THE BURNING RATES .....	75
APPENDIX L. EES CODE USED FOR THE THERMAL MODEL.....	80
REFERENCES .....	81

## LIST OF TABLES

Table 3.1 Profilometer settings for measuring the sample thickness.....	26
Table 3.2 Average thickness for one-pixel wide samples.....	31
Table 3.3 Average thickness for the y-direction pixel pitch samples .....	32
Table 3.4 Average thickness for the wide samples.....	34
Table 3.5 Comparison between the measured and theoretical thicknesses .....	34
Table 3.6 Porosity values that result in matching thickness values.....	35
Table 4.1 Volume fraction, mass fraction, and thermophysical properties of the constituents of the thermite sample. ....	46
Table 4.2 Summary of model inputs.....	47

## LIST OF FIGURES

Figure 2.1 Thermite ink in a mounted syringe and Resodyn acoustic mixer. ....	17
Figure 2.2 Inkjet-printing system (adapted from [12]). ....	18
Figure 2.3 Side-view imaging and an example droplet. ....	19
Figure 2.4 Side-view imaging MATLAB flow chart.....	19
Figure 2.5 Printed droplet grid.....	20
Figure 2.6 Flow chart for calculating the printed droplet sizes in MATLAB .....	21
Figure 2.7 Droplet diameter from side-view images. ....	22
Figure 2.8 Printed lines with varied time between droplets.....	23
Figure 3.1 Top-lit image of a printed burning rate sample (0.4 mm pixel pitch). ....	27
Figure 3.2 Burning rate experimental set-up .....	27
Figure 3.3 Profilometer thickness measurement example (0.1 mm pixel pitch sample).....	29
Figure 3.4 Thickness measurements for the one-pixel-wide samples .....	30
Figure 3.5 Histogram of the distribution of thickness measurements (for a one-pixel wide sample with 0.6 mm pixel pitch).....	30
Figure 3.6 Thickness measurements for the 1.9 mm wide samples.....	31
Figure 3.7 Histogram of the distribution of thickness measurements (for a 1.9 mm wide sample with 0.2 mm pixel pitch).....	32
Figure 3.8 Thickness measurements for the samples with widths from 0.9 mm up to 3.4 mm ....	33
Figure 3.9 Histogram of the distribution of thickness measurements (3.4 mm wide sample) .....	33
Figure 3.10 Printed one-pixel wide samples with pixel pitches of: (a) 0.8 mm, (b) 0.7 mm, (c) 0.6 mm, (d) 0.5 mm, (e) 0.4 mm, (f) 0.3 mm, (g) 0.2 mm, and (h) 0.1 mm. ....	36
Figure 3.11 Burning rate versus pixel pitch for the one-pixel wide samples. There are seven samples per pixel pitch for batches 1 and 2 and fourteen samples per pixel pitch for batch 3 (224 total samples). The error bars are the corresponding standard deviations of the burning rates. ....	36
Figure 3.12 Still frames from high speed video of 0.8, 0.4, and 0.1 mm pixel pitch thermite samples taken 1 ms after ignition.....	37

Figure 3.13: One-droplet wide samples burning rate vs. sample thickness.....	38
Figure 3.14 Varied $y$ -pixel pitch 1.9 mm wide samples with pixel pitches of: (a) 0.2 mm, (b) 0.25 mm, (c) 0.33 mm, and (d) 0.5 mm. ....	39
Figure 3.15 Burning rate versus the pixel pitch in the $y$ -direction for 1.9 mm wide samples (112 total samples; 14 samples per pixel pitch per batch). ....	39
Figure 3.16 Plot of 1.9 mm wide samples burning rate vs. sample thickness .....	40
Figure 3.17 Wider burning rate samples with widths of: (a) 3.4 mm, (b) 2.9 mm, and (c) 2.4 mm. ....	41
Figure 3.18 Burning rate vs. sample width (140 total samples) .....	41
Figure 4.1 Energy balance of a burning nanothermite sample. ....	43
Figure 4.2 Thermal circuit energy balance of a burning nanothermite sample. ....	44
Figure 4.3 Experimental burning rate plotted as a function of the inverse of sample width. These data are fitted to extrapolate the expected burning rate of an infinitely-wide sample. ....	48
Figure 4.4 Burning rate vs. sample thickness model prediction and experimental values (constant sample width of 0.9 mm).....	49
Figure 4.5 Burning rate vs. sample width model prediction and experimental values. ....	50
Figure E1. Image of the printed thermite sample post-processed in MATLAB to determine the area of printed thermite.....	64
Figure H2. Still frames of a high-speed video of the thermite reaction post-processed in MATLAB by tracking the location of maximum intensity following the method of Murray <i>et al.</i> [11] The green line indicates the burning front as predicted by the tracking algorithm while the white dashed line indicates the actual location of the burning front based on visual observation. ....	69
Figure H3. Plot of burning front location versus time as measured by tracking the location of maximum intensity following the method of Murray <i>et al.</i> [11] .....	70
Figure H4. <i>Top panels</i> : Still frames of a high-speed video of the thermite reaction post-processed in MATLAB by two tracking methods. The red shaded region indicates where the pixel intensity is greater than a user-defined threshold value ( $I_{i,j} > TH$ ) and the green line indicates the burning front as predicted by each tracking algorithm. <i>Bottom panels</i> : Plot of burning front location versus time as measured by the two algorithms. ....	72

## ABSTRACT

Author: Son, Forrest, J. MSME

Institution: Purdue University

Degree Received: December 2018

Title: Piezoelectric Inkjet Printed Aluminum Bismuth (III) Oxide: The Effects of Printing Parameters on Burning Rate

Committee Chair: George T.-C Chiu

This thesis presents work on the deposition of nanothermite using a piezoelectric inkjet printer, focusing on the effects of printing parameters and sample geometry on burning rate. The ability of the printer to produce consistent droplet size and spacing was shown to have repeatable droplet size and sub-millimeter precision in droplet spacing. The droplet-droplet interaction of the nanothermite ink was examined, and a printing frequency of 10 Hz was shown to produce smooth and consistent geometry in the printed samples. The primary printing parameter varied in this study was the pixel pitch (*i.e.*, the distance between printed droplets). As pixel pitch decreased (*i.e.*, the droplets are printed closer together) in both directions (*x*- and *y*-directions), the burning rate increased, and as sample width increased the burning rate increased. A significant number of samples (476) were printed and demonstrated consistent, energetic performance; this indicated favorable high-volume production capabilities. A thermal model was developed based on an energy balance for the printed nanothermite samples. The model accurately predicted the burning rate trends observed in the experimental results. This result indicated that the increase in heat generation in both the thicker (pixel-pitch studies) and wider samples decreased the significance of heat loss to the environment. The statistically significant results presented in this work, along with a descriptive thermal model, increase the fundamental understanding of the effects of printed geometry and droplet spacing on nanothermite energetic performance.

## CHAPTER 1. INTRODUCTION

### 1.1 Background Information

There is significant interest in developing a process to precisely deposit energetic materials for integration with electronic systems. Energetic materials offer a reliable destructive mechanism. Nanonergetic materials are of particular interest to the small-scale application of electronic devices due to their low ignition energy and quick energy release. Nanoenergetic materials are characterized by a high specific surface area (*i.e.*, the surface area per unit mass) and exhibit improved ignition and destructive performance when compared with traditional energetic materials [1]. These materials have higher energy density and lower activation thresholds due to a large number of surface atoms for a given mass of material [2]. One class of materials in this category is nanoscale thermites. Nanothermites are comprised of metal and metal-oxide nanoparticles which serve as the fuel and oxidizer respectively. There are several challenges which limit the use of nanoparticles in inkjet printing systems: aggregation [3], [4], higher viscosity [5], and decreasing active aluminum content as metal particle size decreases [2]. Therefore, it is vital to develop a method for precisely depositing these materials.

Piezoelectric inkjet-printing shows promising potential as a method for integrating energetic materials with electronic systems. It is an effective process for accurately depositing precise quantities of material for a variety of applications. Over the past decade, researchers began leveraging the highly sophisticated capabilities of inkjet printers to develop novel particulate deposition techniques for a wide range of materials, from ceramics [6], metals and oxides [7], [8], and polymers [9], to proteins [10]. These studies replace the traditional ink in a printing system with a suspension of particles in solvents, and surfactants. The ability to print particles enables the fabrication and integration of nanothermite into sub-millimeter scale devices (*e.g.*, high-valued electronic systems) as demonstrated by recent works [11], [12] .

Printing particles presents a significant challenge due to the small size of inkjet nozzles (*e.g.*, less than 100  $\mu\text{m}$  in diameter); the inks must be carefully formulated to produce consistent material deposition. Common challenges with inkjet printing particles include: suspension homogeneity, agglomeration, solids loadings, and deposition uniformity.

While recent work has demonstrated the ability of inkjet printers to deposit thermite samples, the printed samples yielded inconsistent energetic performance [11], [13]. For example, Tappan *et al.* [13] reported burning rates that varied as much as 200 cm/s between printed nanothermite samples of the same thickness. Producing uniform samples with a consistent burning rate is key to proving the viability of this additive manufacturing technique for energetic materials. Thus, it is vital to understand both the accuracy and reliability of the printing system as well as the effect which printing parameters have on the energetic performance.

The work presented in this thesis focuses on the ability of a piezoelectric inkjet printer to accurately deposit nanothermite ink and the effect of process parameters on energetic performance. This work includes three main efforts: (1) piezoelectric inkjet printing performance; (2) burning rate experiments studies; and (3) an analytical thermal model for predicting energetic performance.

## 1.2 Piezoelectric Inkjet Printing Performance

A key focus of this work is to present a systematic approach for achieving consistent, energetic performance from printed nanothermite. A custom-built inkjet printing setup is used to control the deposition of the ink. The ability of the system to accurately deposit nanothermite ink is examined by imaging the droplets as they leave the printer nozzle and after they have dried on the substrate. The size and printed spacing of the droplets are measured using a custom MATLAB code. These analyses provide valuable insight into the accuracy of the printing system and are the foundation for the experimental studies presented in this work.

In addition to the reproducibility of the printed thermite, the droplet-droplet interaction of the ink is examined. As droplets are printed onto the substrate, adjacent droplets interact and form a liquid connection; once the solvent evaporates, the connection becomes solid. When the printing rate is too slow, individual droplets will dry before adjacent droplets are deposited resulting in a rippled effect. Since one of the main goals of the experimental effort of this work is to determine the effect of printed geometry on energetic performance, it is a vital step to determine what printing frequency results in smooth, consistently wide samples.

The results of the printing performance analyses provide key information for the burning rate studies. The ability of the system to deposit consistent sized droplets, precisely space them, and print smooth geometry leads to repeatable experimental results.

### 1.3 Burning Rate Studies

The goals of these studies are to determine the effect of printing parameters and sample width on burning rate. Burning rate experiments are a common technique for measuring the performance of energetic materials. Recent work shows a direct correlation between burning rate speed and the destructive capabilities of thermite [12]. There are several parameters which control the final performance of inkjet-printed thermite, some of which were investigated in previous studies (*e.g.*, solids loading [11] and equivalence ratio [12]). This work investigates the effect of a different parameter, pixel pitch. Pixel pitch is the distance between two pixels on a bitmap (*i.e.*, pixel density per unit length). The goal of the first study is to determine how the performance of printed nanothermite varies with pixel pitch in the  $x$ -direction of the sample (as defined in Figure 3.1). The second study examines how varying the pixel pitch in the  $y$ -direction (along the width of the sample) affects the burning rate. These samples are the same length as the first study (24 mm), have a constant width of 1.9 mm, and a constant  $x$ -direction pixel pitch of 0.5 mm. The final study determines the effect of sample width on burning rate (all of the samples have a fixed pixel pitch in the  $x$ - and  $y$ -direction of 0.5 mm); the sample width is varied from 0.9 to 3.4 mm.

### 1.4 Analytical Model of Burning Rate Trends

The goal of the model is to explain the physical phenomena that lead to the experimental burning rate trends. An energy balance is developed to predict the flame temperature of an inkjet-printed nanothermite sample as a function of the sample's geometry and energetic properties. The energy balance leverages the Arrhenius equation [14] as a constitutive relation to express the reaction rate of the nanothermite—which is an input to the energy balance—as a function of temperature. The resulting set of equations is solved iteratively to find the flame temperature. The burning rate can then be predicted from this flame temperature using Mallard and Le Chatelier's thermal theory for flame propagation [14]. The sample width and thickness are varied to provide predictions for the observed experimental burning rates. The results of the model are key to understanding the fundamental physical phenomena that cause the experimental burning rate trends.

## CHAPTER 2.     PIEZOELECTRIC INKJET PRINTING PERFORMANCE

### 2.1   Introduction

The final performance of a printed thermite sample depends on accurate, and uniform deposition of the thermite material. Non-uniform inks result in inconsistent printing performance and inaccurate printed geometry. The fundamental technique for preparing and printing particles is to suspend the material in a solvent and a surfactant. These suspensions are generated by mixing particles with a solvent [6]–[12]. Agglomerations are commonly broken up via ultrasonication [15] and/or the introduction of surfactants [12]. In addition to the concerns about mixture uniformity, the concentration of solids (*i.e.*, volumetric solids loading) have a significant impact on the quality of the final printed shape [16]. When formulating a new ink, it is necessary to determine the optimal concentration of solids to achieve stable jetting performance [11].

Sample uniformity (*i.e.*, surface smoothness) is also a complex function of the deposition conditions. Derby [6] studied the effect that the proximity of adjacent drops and the drying time between printing adjacent drops had on the final structure of printed ceramic samples. Ideally, a geometry is generated by printing individual droplets sufficiently close together to form liquid bridges, and eventually smooth solid samples when the solvent evaporates. As long as adjacent drops are still in liquid form, they will form a solid connection. If drops are too far apart, or if there is too long a delay between the printing of neighboring droplets resulting in the first droplet drying before the neighboring droplet is deposited, there will be gaps and ridges in the sample geometry.

In addition to the aforementioned challenges common to all particulate, inkjet printing applications, there are several additional concerns when printing energetic materials such as thermite. Safety concerns are paramount for the present application. Solution components and physical hardware are all selected to minimize handling and printing hazards. The reactive particles are mixed with sufficient solvent to reduce the system's overall potential for reacting [17]. In addition to safety concerns, there are additional complexities that come with printing thermite. Thermite is a metastable intermolecular composite (MIC) comprised of a mixture of metal and metal oxide powders; when the mixture is sufficiently heated, it undergoes an exothermic reduction-oxide

reaction. Thermite has many uses, from metallurgical applications, pyrotechnics, synthesis of composite materials and welding [18], to the application in this research, building secure electronic systems. Since thermite contains two distinct powders, the printing process is complicated. The challenge of achieving a uniform mixture of multiple components is often encountered with energetic materials. Researchers routinely mix the separate fuel- and oxidizer-components in a solvent to achieve a homogeneous mixture. The solvent's role is to improve the mixing of the particles and to desensitize the material's ignition energy threshold. The solvent mixture is ultrasonicated to break up agglomerates [15], then poured onto a thin film and allowed to dry. In addition to a solvent, a surfactant is typically added during the ink formulation process. The surfactant inhibits the particles from sticking together during the mixing and printing process, keeping the ink printable. A similar procedure is followed in the present work to create a homogeneous thermite ink mixture.

## 2.2 Thermite Ink Preparation

The mixing process is vital for formulating a homogeneous thermite ink suspension. The ratio of metal to metal oxide (*i.e.*, the equivalence ratio),

$$\phi = \frac{F/A}{(F/A)_{\text{stoic}}} \quad (2.1)$$

is an important property of thermite [12]. In Equation 2.1,  $F$  is the number of moles of fuel, and  $A$  is the number of moles of oxidizer. The stoichiometric ratio of fuel to oxidizer—the denominator of the equivalence ratio—is the ratio of fuel to oxidizer when precisely enough oxidizer is used to ideally react all of the fuel. A recent study [12] found that an equivalence ratio of 2.0 results in the fastest burning and most destructive printed samples for printed Al/Bi<sub>2</sub>O<sub>3</sub> nanothermite. Therefore, an equivalence ratio of 2.0 is used in this work. The process for preparing the thermite ink is similar to the procedure in [12]. Aluminum nanoparticles (NovaCentrix, nominal diameter of 80 nm) are mixed with bismuth oxide nanoparticles (Nanophase Technologies Corporation, nominal diameter of 38 nm) to achieve the desired equivalence ratio. The aluminum powder is stored in a vacuum seal prior to mixing to avoid excessive oxidation and contact with moisture. The powders are combined in a syringe (BD 10 mL slip tip) in preparation for mixing. A mixture of solvent (Sigma-Aldrich N, N-dimethylformamide, anhydrous, 99.8%) and surfactant (Solsperse 41000, Lubrizol) in a 19:1 ratio by mass is prepared separately and then added to the syringe containing the powders. This solvent and surfactant mixture was selected to minimize

sedimentation based on previous tests [19]. The syringe, shown in Figure 2.1, is sealed and securely mounted to a LabRAM resonant mixer (Resodyn Acoustic Mixers, Inc.), for the thorough mixing of the components. The syringe is mixed for 8 min, then flipped and mixed for an additional 8 min at 80% intensity (*i.e.*, 80 g acceleration). After mixing, the thermite ink is found to be homogeneous and is ready to be printed.

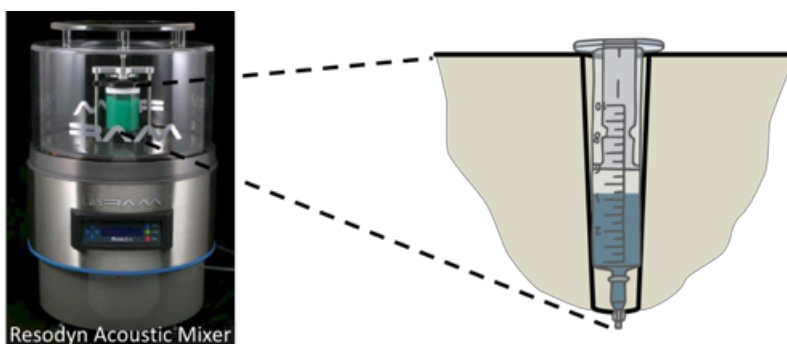


Figure 2.1 Thermite ink in a mounted syringe and Resodyn acoustic mixer.

### 2.3 Thermite Ink Printing

Based on a previous study [11], a piezoelectric inkjet printer system is used to deposit the thermite ink. To further ensure homogeneity of the thermite, the ink is sonicated (Branson, Inc. CPX ultrasonic bath) for 30 min prior to printing. After 30 min, the ink is loaded into a syringe and connected to a Pipejet® (Biofluidix) printer (500  $\mu\text{m}$  nozzle diameter). The printer is capable of depositing one drop at a time on the nanoliter scale (*e.g.*, 40 nL). The thermite ink is connected to the printer and a sub-ambient back-pressure line, to prevent the ink from free flowing through the nozzle. The Pipejet nozzle is held stationary while a dual-axis linear positioning stage (Aerotech Planar DL 200-XY, 200 mm travel, 0.5  $\mu\text{m}$  accuracy) moves to generate the desired sample geometry, see Figure 2.2. The stage and printer are controlled by a custom LabView code; the desired print parameters, as well as the desired geometry, are selected in this code prior to printing.

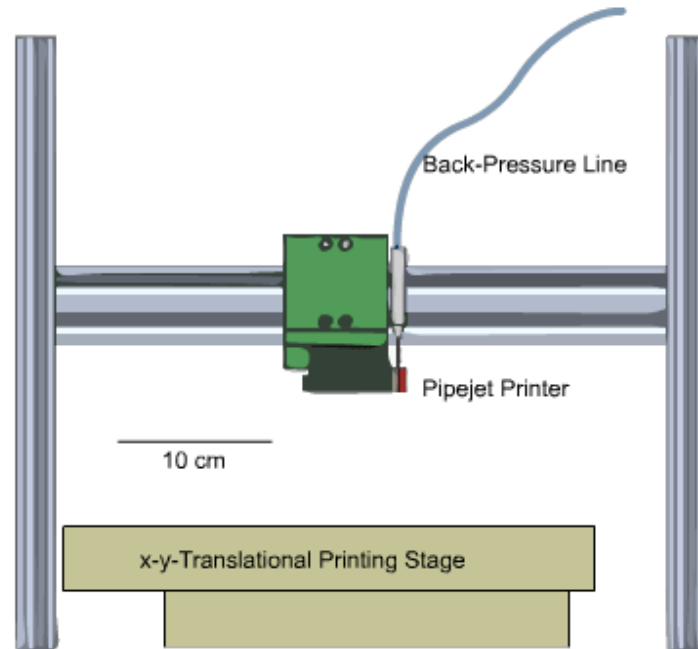


Figure 2.2 Inkjet-printing system (adapted from [12]).

## 2.4 Printing Performance

The reproducibility/accuracy of the inkjet printing system is evaluated by (1) imaging the droplets during printing and (2) analyzing the final, printed product.

First, the variability in droplet size is measured by capturing side-view images of the drops in-flight. A flashing LED is synchronized with the printed droplet's frequency to capture them in flight, as shown in Figure 2.3. The droplet diameter is extracted from a binarized version of each frame using a Hough Circle transformation in MATLAB. A flow diagram, shown in Figure 2.4, shows how the Hough Circle transformation is implemented for this calculation. A more in-depth description of this code and the MATLAB script is found in Appendix A and B, respectively. A typical frame showing a droplet in-flight is given as an inset in Figure 2.3. Droplet diameters are plotted to verify that the thermite is printing at a consistent size over time.

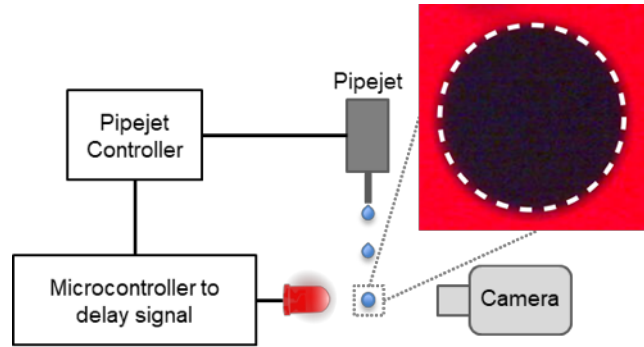


Figure 2.3 Side-view imaging and an example droplet.

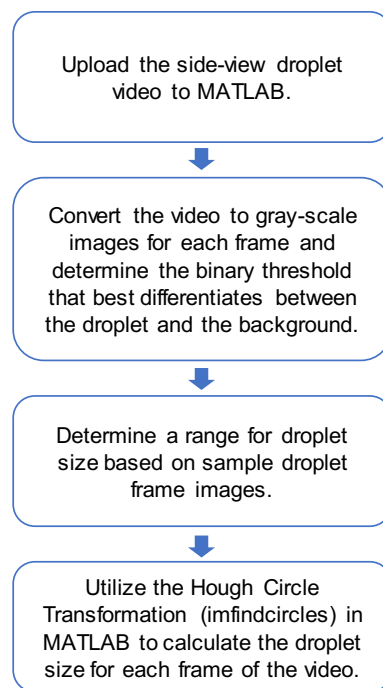


Figure 2.4 Side-view imaging MATLAB flow chart

Second, a printed 9x9 droplet grid is generated to determine the variance in the dried droplet diameters and the accuracy of the printing system's spacing in the  $x$ - and  $y$ -direction. The droplets are printed 2.5 mm apart in the  $x$ - and  $y$ -direction. After printing, the solvent and surfactant evaporate, leaving behind a circular region of solid thermite where each droplet is deposited. An image of the dried grid (shown in Figure 2.5) is then analyzed in MATLAB. A binary threshold is used to identify printed droplets and background pixels; the area of each printed droplet is then converted into an equivalent circle area diameter, and its centroid is determined. A flow diagram, shown in Figure 2.6, shows how to use MATLAB for these calculations

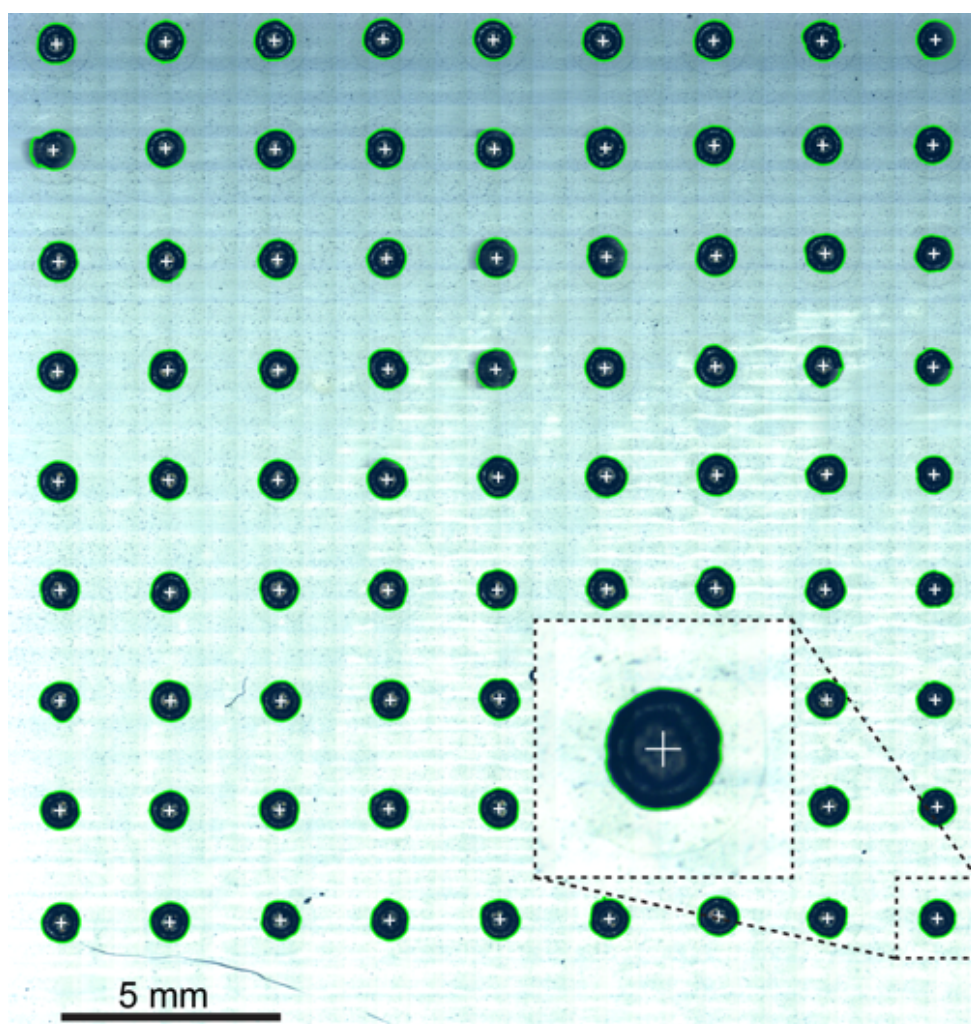


Figure 2.5 Printed droplet grid.

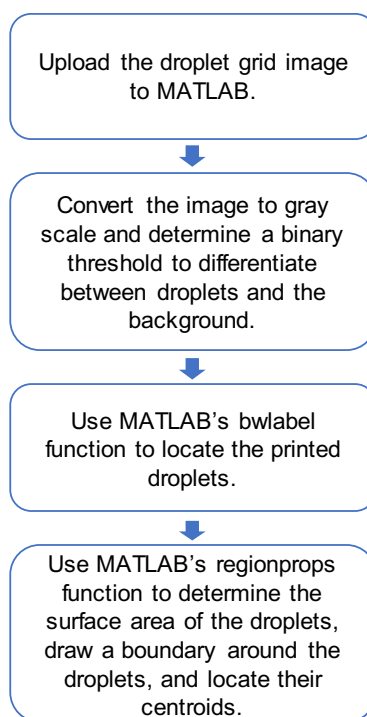


Figure 2.6 Flow chart for calculating the printed droplet sizes in MATLAB

In addition to the reproducibility of the printed thermite, the droplet-droplet interaction of the ink is examined. The objective is to understand how drying time affects droplet interaction on the substrate. Ideally, adjacent droplets interact with each other to form liquid bridges, leading to a seamless solid thermite sample. When the printing rate is too slow, individual droplets will dry before the adjacent droplets are deposited resulting in a rippled effect. Several samples are printed with deposition rates varying from 0.2 to 10 Hz while holding the pixel pitch (spacing between droplets) fixed at 0.5 mm (which allows for overlap of the droplets on the substrate). The surfaces of the resulting dried samples are then examined to determine the deposition rate which results in smooth samples.

## 2.5 Results

### 2.5.1 Droplet Side View Imaging

Aluminum-bismuth (III) oxide thermite inks are prepared at an equivalence ratio of 2.0. Side view images are captured of droplets during the printing process to determine their diameters; the results

are shown in Figure 2.7. The average droplet diameter is 0.43 mm with excellent repeatability ( $\pm 0.005$  mm), as shown in Figure 2.7.

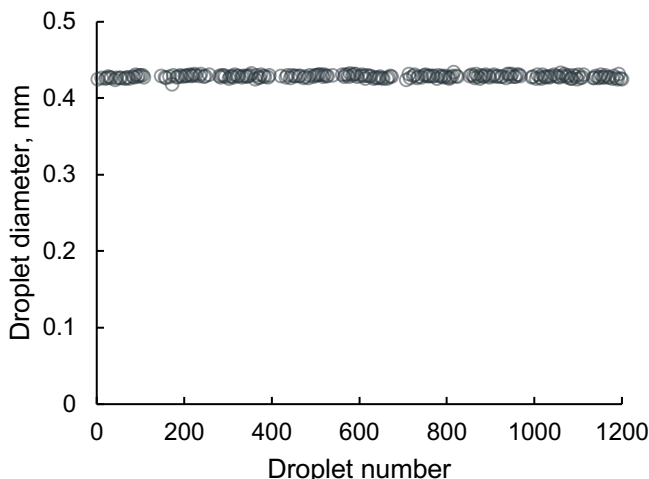


Figure 2.7 Droplet diameter from side-view images.

### 2.5.2 Printed Droplet Grid

A printed grid is then generated using the same thermite ink; it is imaged after drying for 24 h to allow the solvent and surfactant time to evaporate. The grid is shown in Figure 2.5 with green outlines around each dried droplet and a white cross at their centroids. The equivalent circle area diameter each droplet (*i.e.*, the diameter of a circle with the same area as the printed droplet) and the distance between neighboring droplets is calculated using a custom MATLAB code based on a MATLAB tutorial for segmenting an image of multiple circular objects [20]. A detailed description of this process and the MATLAB code is found in Appendix C and D. The average deposited droplet diameter is 0.90 mm with a standard deviation of 0.023 mm; this repeatability is expected based on the consistent droplet size captured with the side-view camera (Figure 2.7). The droplets are printed with a desired spacing of 2.5 mm in the  $x$ - and  $y$ -directions; the average measured droplet spacing is 2.49 mm with a standard deviation of 0.037 mm in both the  $x$ - and  $y$ -directions.

### 2.5.3 Droplet-Droplet Interaction

Another batch of aluminum bismuth-oxide ink is prepared at the same equivalence ratio ( $\phi = 2.0$ ). Lines are printed to examine how adjacent drops interacted when the drying time between droplet

deposition is varied. The time between printing adjacent drops is varied from 5 to 0.1 s (*i.e.*, deposition rates of 0.2 to 10 Hz). After drying, these printed samples are examined to determine print quality. The criteria for a successful print are a smooth, uniformly wide line with no ridges where the droplets overlap; images of these printed lines are shown in Figure 2.8. When the time between printing adjacent droplets is longer than one second, there are large, clearly visible ridges where the droplets overlap. This is likely because the first droplet has evaporated a sufficient amount of solvent to create solid bridges. This corroborates the findings of Derby [6] who noted that slow printing rates lead to partial evaporation and prevent the formation of liquid bridges between droplets; these liquid bridges are vital to forming a smooth surface upon solvent evaporation. As the time between printing adjacent droplets decreases, the ridges between droplets decrease in size and eventually disappear completely when the time between drops is 0.1 seconds (10 Hz printing frequency). Therefore, thermite samples are printed with a frequency of 10 Hz for all of the burning rate studies.

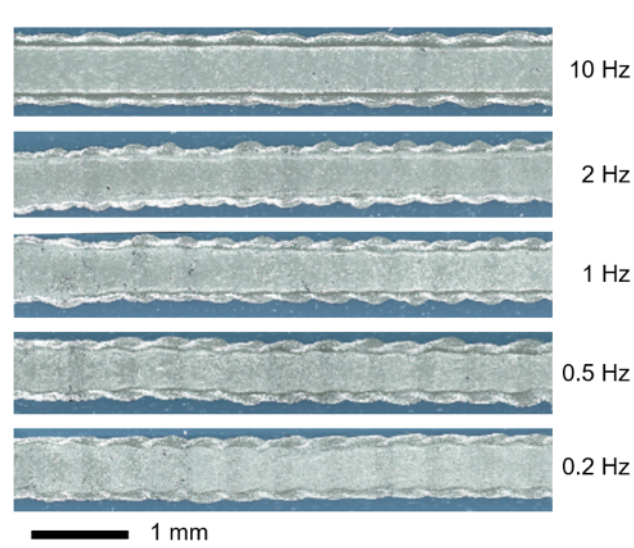


Figure 2.8 Printed lines with varied time between droplets.

## 2.6 Conclusions

The systematic approach presented in this work provides clear guidelines for achieving consistent and precise printed thermite samples. This was accomplished by capturing side-view images of printed droplets, printing a droplet grid and measuring dried droplet size and spacing, and varying

the drying time between printed adjacent droplets. Droplets were printed with a consistent size (0.001 mm standard deviation in diameter), dried droplets were shown to have similar size (0.023 mm standard deviation in diameter), and droplet spacing was demonstrated to have sub-millimeter precision (0.037 mm standard deviation). Additionally, a printing frequency of 10 Hz was demonstrated to result in smooth surfaced samples. These results are critical for the burning rate parametric studies; it shows that thermite can be printed at a highly consistent size and accurate spacing with sub-millimeter precision.

## CHAPTER 3. BURNING RATE STUDIES

### 3.1 Introduction

Printed thermite has promising potential for building secure electronic systems; however, there is an insufficient characterization of the energetic performance of printed thermites. The work presented in this paper builds upon recent research [11], [12] on printing nanothermite and contains new results on the functional performance of printed thermite, as well as a determination of improved printing parameters for achieving uniform printed samples. Aluminum bismuth-oxide thermite ( $\text{Al/Bi}_2\text{O}_3$ ) was selected for this study based on its favorably energetic performance, as demonstrated in previous work [12], [21]–[23]. Several batches of nano aluminum bismuth (III) oxide ( $\text{Al/Bi}_2\text{O}_3$ ) thermite are produced by carefully weighing and mixing powders with a solvent and a surfactant. An issue with previous studies was batch-to-batch repeatability. A key focus of this work is to present a systematic approach for achieving consistent, energetic performance from printed nanothermite. A custom-built inkjet printing setup is used to control the deposition of the ink. Three parametric studies are presented showing how the functional performance of the thermite varies based on droplet density (in both the  $x$ - and  $y$ -direction) and sample width. The first study determines the effect of varying the pixel pitch (*i.e.*, droplet density) in the  $x$ -direction (along the length of the sample; all of the samples had a fixed length of 24 mm and a fixed width of 0.9 mm) on burning rate. The second study looks at the effect of varying the pixel-pitch in the  $y$ -direction (along the width of the sample; all of the samples had a fixed length of 24 mm, a fixed width of 1.9 mm, and a fixed  $x$ -direction pixel pitch of 0.5 mm) on burning rate. The third study examines how sample width affected burning rate (all of the samples had a fixed pixel pitch in the  $x$ - and  $y$ -direction of 0.5 mm). A sufficient number of samples are printed for each study to produce statistically significant conclusions.

### 3.2 Experimental Methods

There are several parameters which control the final performance of inkjet-printed thermite, some of which were investigated in previous studies (*e.g.*, solids loading [11] and equivalence ratio [12]). This study investigates the effect of a different parameter, pixel pitch. Pixel pitch is the distance between two pixels on a bitmap (*i.e.*, pixel density per unit length). Each pixel on a bitmap

represents a single droplet; therefore, the pixel pitch is the distance between the centroids of adjacent droplets.

### 3.2.1 Sample Thickness Measurement

The sample thickness is measured with an optical profilometer (Zeta Instruments 3D Optical Profiler). The top and base of the sample are defined by moving the stage with the sample vertically until the sample is completely out of focus in both directions (*i.e.*, the stage is moved vertically upward to select the top and vertically downward to select the bottom). The settings for measuring sample thickness are provided in Table 3.1.

Table 3.1 Profilometer settings for measuring the sample thickness

Profilometer Settings
ZDOT Mode
Select High Dynamic Range (HDR)
400 steps (number of images taken per sample)
5X Magnification Lens

To ensure accurate measurement of sample thickness, images are taken along the entire length of the sample and thickness is measured at several locations in each respective section. The number of steps defines the resolution of the image; 400 steps are selected to provide a resolution of approximately 1  $\mu\text{m}$  per step. The 5X magnification lens is selected to provide a sufficiently large field of view (FOV) to image the sample and the substrate. Higher magnification results in improved image quality, and thus a more accurate measurement. However, it is necessary to select a magnification that provides an adequate FOV to image the entire sample and a portion of the substrate (the reference surface). After the profilometer takes the image, the sample is analyzed. Using the built-in software, the cursor is used to measure the sample height compared to the substrate height. Multiple cursors are added to aid in the analysis. This process is repeated for each section of the sample along its entire length. The multiple thickness measurements for each sample section are averaged, and the standard deviation is calculated.

### 3.2.2 Burning Rate Studies

Burning rate experiments are a common technique for measuring the performance of energetic materials. Recent work shows a direct correlation between burning rate speed and destructive capabilities of thermite [12]. The goal of the first study is to determine how the performance of

printed nanothermite varies with pixel pitch in the  $x$ -direction of the sample (see Figure 3.1). Burning rate experiments are conducted by igniting a line of thermite and measuring the propagation rate of the flame front.; an example printed thermite sample for a burning rate test is shown in Figure 3.1. The sample is composed of an ignition point (3.6 mm in diameter) connected to a 24-millimeter long line that is one pixel (*i.e.*, one droplet) wide. The samples are ignited with a capacitive discharge unit (Information Unlimited), and the reaction speed is captured with a Phantom high-speed camera (BW Phantom v7.3) as shown in Figure 3.2. An optical mirror is used to capture a top-view video of the reaction. A custom MATLAB script is used to track the flame front on the sample versus time to determine its burning rate, as described in Appendices H–K. This process is repeated for samples with varying pixel pitches from 0.1 to 0.8 mm.

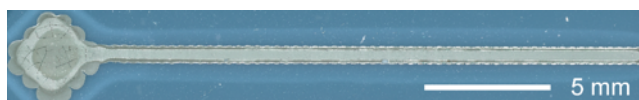


Figure 3.1 Top-lit image of a printed burning rate sample (0.4 mm pixel pitch).

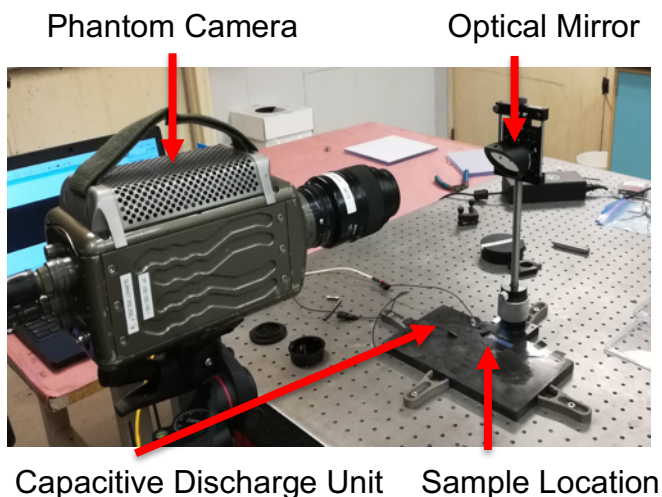


Figure 3.2 Burning rate experimental set-up

The second study examines how varying the pixel pitch in the  $y$ -direction affects the burning rate. These samples are the same length as the first study (24 mm), have a constant width of 1.9 mm, and a constant  $x$ -direction pixel pitch of 0.5 mm. The final study determines the effect of sample width on burning rate (all of the samples have a fixed pixel pitch in the  $x$ - and  $y$ -direction of 0.5

mm); the sample width is varied from 0.9 to 3.4 mm. Both of these studies use the same methods as the first study to determine the burning rate.

### 3.2.3 Statistical Analysis of Results

There are two statistical analyses performed on the experimental data for each of the three studies: (1) a single factor analysis of variation (ANOVA) test and (2) a Pearson's  $r$  correlation coefficient calculation.

The goal of the ANOVA test is to determine if there is strong evidence against the null hypothesis that all of the population means are equal. In other words, the test is used to determine if there is a statistical difference between the means of distinct sample groups. The test is comparing the variability between groups to the variability within groups. The key result of an ANOVA test is the  $p$ -value. If the  $p$ -value is less than 0.05, it means that there is strong evidence that the population means are not equal. This shows that the variability between groups (*e.g.*, different pixel pitch/sample width) is significant enough to overcome the variability within the groups (*i.e.*, there is a statistical difference between the mean burning rate for each of the burning rate studies). This does not tell us how strong the trend is; however, it does indicate that there is a statistically significant effect on burning rate based on varying pixel pitch and sample width. The ANOVA test in this work is performed by utilizing the built-in solver in Excel.

If the ANOVA test indicates that a statistical difference exists between the mean burning rate for the three studies, the next step is to determine the significance of the trends. This is accomplished by calculating the Pearson's correlation coefficient (*i.e.*, Pearson's  $r$ ). The range of values for  $r$  is between -1 and 1; a value of -1 indicates a perfect negative linear correlation and a value of 1 indicates a perfect positive linear correlation. The calculation of  $r$  includes all of the results and aims to determine the best-fit trend line for all of the data points. It is calculated by,

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{\left[ n \sum x^2 - (\sum x)^2 \right] \left[ n \sum y^2 - (\sum y)^2 \right]}} \quad (3.1)$$

where,  $n$  is the total number of samples in a burning rate study,  $x$  is the pixel pitch/sample width, and  $y$  is the corresponding burning rate value.

The goals of these two statistical analyses are to (1) determine whether or not a statistical difference exists between the mean burning rates for each study and to (2) determine how strong of a trend, if any, exist between pixel pitch/sample width and burning rate

### 3.3 Results

#### 3.3.1 Measured Sample Thickness

The profilometer (Zeta Instruments 3D Optical Profiler) is used to measure the thickness along the entire length of each sample. An example image of the profilometer software used to measure the sample thickness is shown in Figure 3.3. Figure 3.4 is a plot of each thickness measurement made for the one-pixel-wide samples. Figure 3.5 is an example histogram of the distribution of thickness measurements of one of the samples. Table 3.2 shows the average thickness for each pixel pitch and the corresponding standard deviations. As expected, as the pixel pitch decreases the sample thickness increases.

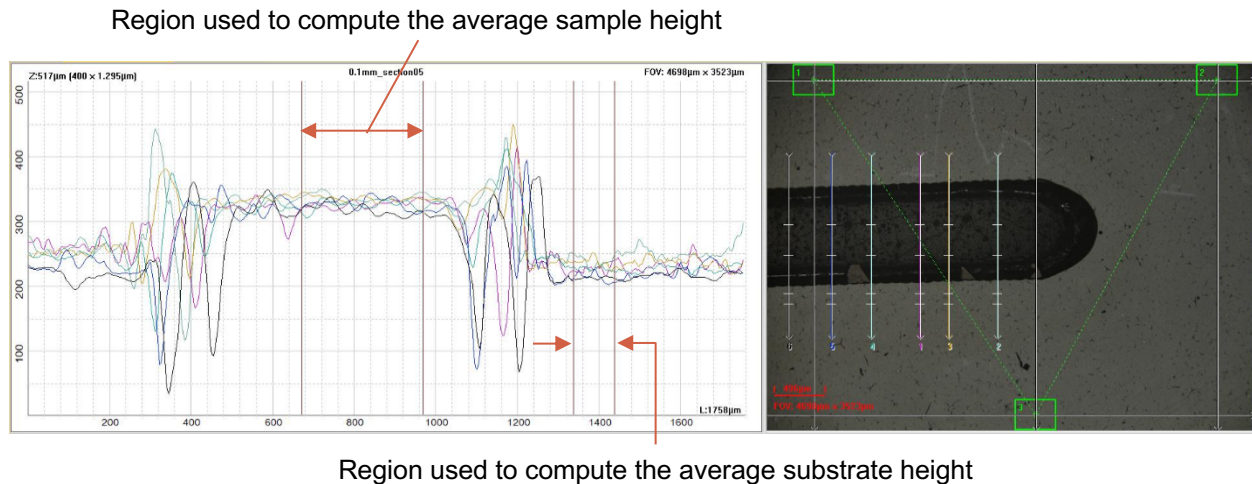


Figure 3.3 Profilometer thickness measurement example (0.1 mm pixel pitch sample)

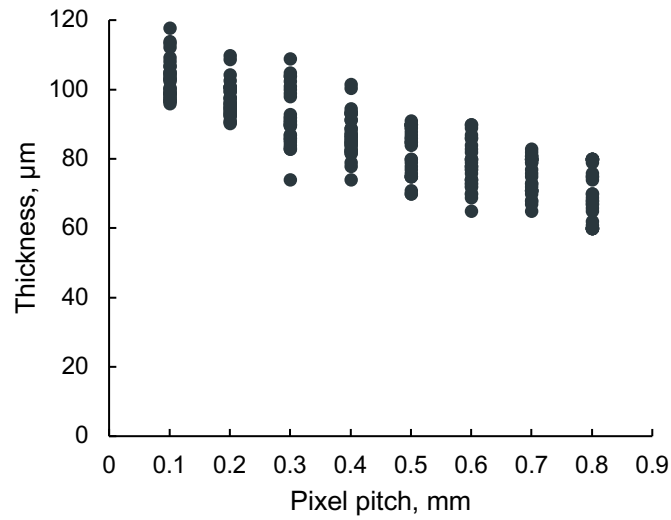


Figure 3.4 Thickness measurements for the one-pixel-wide samples

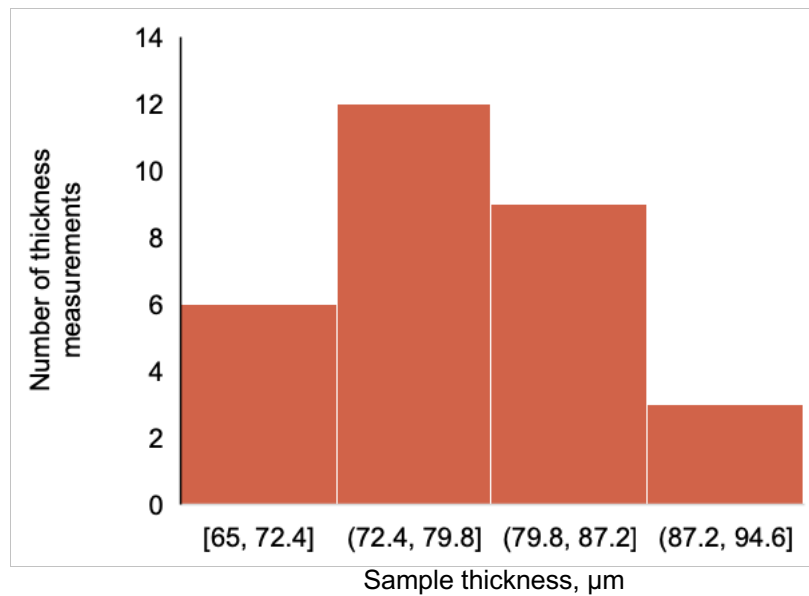


Figure 3.5 Histogram of the distribution of thickness measurements (for a one-pixel wide sample with 0.6 mm pixel pitch)

Table 3.2 Average thickness for one-pixel wide samples

Pixel pitch, mm	Average thickness, $\mu\text{m}$	Standard deviation, $\mu\text{m}$
0.1	103	6
0.2	97	5
0.3	92	8
0.4	87	6
0.5	83	7
0.6	78	6
0.7	76	5
0.8	69	8

The same procedure is followed to measure the thickness of the  $y$ -direction pixel pitch samples. Figure 3.6 is a plot of each thickness measurement made for the  $y$ -direction pixel pitch samples. Figure 3.7 shows a typical histogram of the distribution of thickness measurements of a 1.9 mm wide sample with 0.2 mm pixel pitch. Table 3.3 shows the average thickness for each pixel pitch and the corresponding standard deviations. As expected, as the pixel pitch decreases the sample thickness increases.

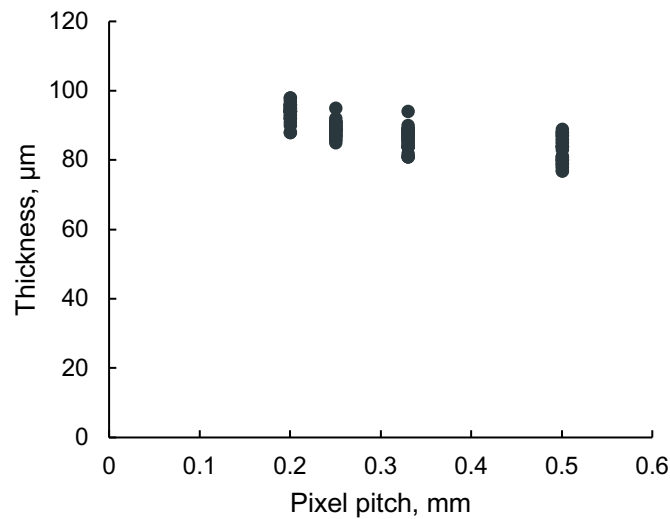


Figure 3.6 Thickness measurements for the 1.9 mm wide samples

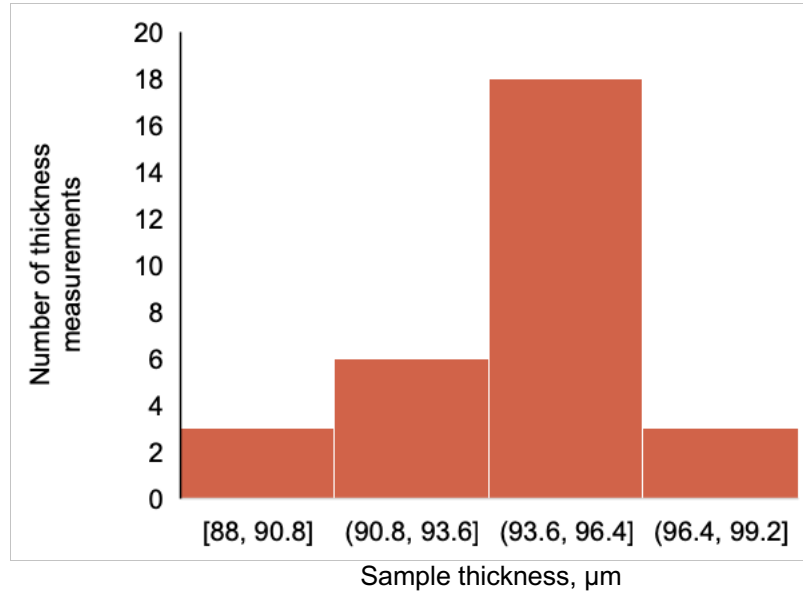


Figure 3.7 Histogram of the distribution of thickness measurements (for a 1.9 mm wide sample with 0.2 mm pixel pitch)

Table 3.3 Average thickness for the  $y$ -direction pixel pitch samples

Pixel pitch, mm	Average thickness, $\mu\text{m}$	Standard deviation, $\mu\text{m}$
0.2	94	2
0.25	89	2
0.33	86	3
0.5	83	4

The same procedure is followed to measure the thickness of the wide samples. Figure 3.8 is a plot of each thickness measurement made for the wide samples. Figure 3.9 shows the histogram of the thickness distribution of a 3.4 mm wide sample. Table 3.4 shows the average thickness for each width and the corresponding standard deviations. As expected, as the sample thickness is relatively constant as sample width increases. This is expected since the pixel pitch is constant in both the  $x$ - and  $y$ -direction for each of these samples.

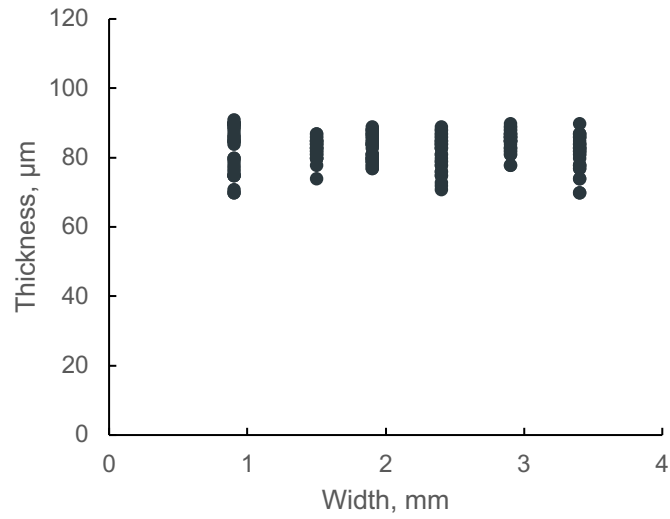


Figure 3.8 Thickness measurements for the samples with widths from 0.9 mm up to 3.4 mm

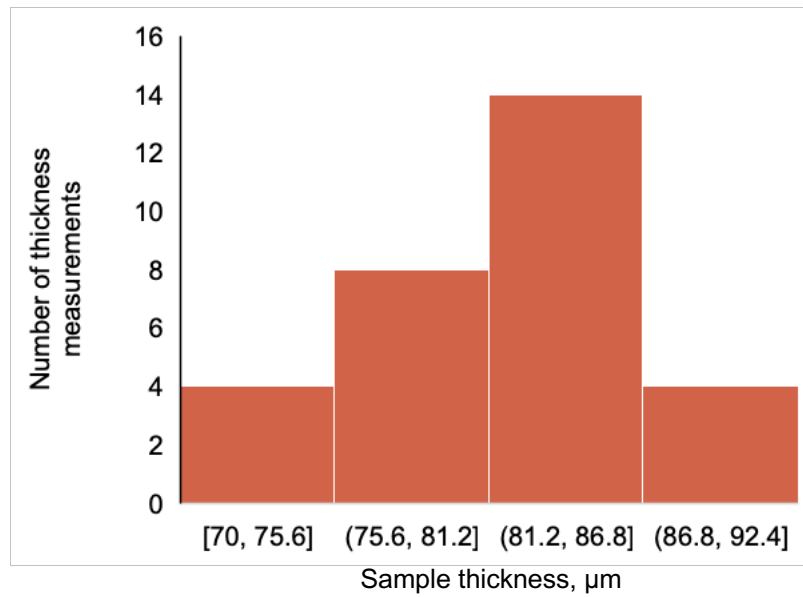


Figure 3.9 Histogram of the distribution of thickness measurements (3.4 mm wide sample)

Table 3.4 Average thickness for the wide samples

Width, mm	Average thickness, $\mu\text{m}$	Standard deviation, $\mu\text{m}$
0.9	83	7
1.5	83	3
1.9	83	4
2.4	81	5
2.9	85	3
3.4	81	5

The values obtained from these measurements are used to determine if there is a trend of burning rate versus sample thickness.

To further examine the accuracy of the profilometer thickness measurements, a theoretical calculation is performed for comparison. Based on the side-view image droplet measurements (Section 2.5.1), the droplet diameter of printed Al/Bi<sub>2</sub>O<sub>3</sub> ink is 0.43 mm. Assuming the droplets are approximately spherical, the resulting volume is 0.042 mm<sup>3</sup>. Since the volumetric solids loading of the thermite ink is 8%, the volume of thermite per droplet is approximately 0.0034 mm<sup>3</sup>. This volume combined with the printed geometry of the one-droplet wide samples is used to calculate a theoretical thickness. The width of the one-droplet wide lines is measured in MATLAB using the image processing toolbox to segment an image of the printed sample as described in Appendices E–G. The thickness is calculated from this equivalent area width by multiplying the number of droplets per sample length (droplets/mm length) by the volume of thermite per droplet and dividing by the equivalent area width, and the assumed solid volume fraction (45%). Results are shown in Table 3.5.

Table 3.5 Comparison between the measured and theoretical thicknesses

Pixel pitch, mm	Equivalent width, mm	Theoretical thickness, $\mu\text{m}$	Measured thickness, $\mu\text{m}$
0.1	0.95	79	103
0.2	0.82	45	97
0.3	0.79	31	92
0.4	0.78	24	87
0.5	0.75	20	83
0.6	0.75	17	78
0.7	0.73	15	76
0.8	0.71	13	69

There is a significant difference between the theoretical thickness and the profilometer measured thickness. A possible explanation for this discrepancy could be the assumed packing efficiency of the printed thermite. Table 3.6 contains a column that lists the porosity values that would result in a match between the theoretical thickness calculations and the measured thicknesses. It is also possible that the porosity of the printed samples varies with pixel pitch. This would account for some of the discrepancies between the thickness values. The profilometer could also have errors in its thickness measurements. For the purposes of this thesis, the measured thicknesses of the profilometer are assumed accurate.

Table 3.6 Porosity values that result in matching thickness values

Pixel Pitch	Equivalent Width, mm	Measured thickness, $\mu\text{m}$	Porosity
0.1	0.95	103	66%
0.2	0.82	97	79%
0.3	0.79	92	85%
0.4	0.78	87	88%
0.5	0.75	83	89%
0.6	0.75	78	90%
0.7	0.73	76	91%
0.8	0.71	69	91%

### 3.3.2 One Droplet Wide Samples Burning Rate Versus Pixel Pitch

Three sets of one-pixel wide (one droplet wide) burning rate samples are printed from three separate batches with the pixel pitch in the  $x$ -direction varied from 0.1–0.8 mm in 0.1 mm increments. A batch is defined as thermite ink that is prepared independently (*i.e.*, prepared from a separate batch of aluminum and bismuth-oxide powders on a different day) and used to print one set of samples. All of the samples are printed with 0.1 seconds between droplets (10 Hz printing frequency), and the ignition point is the same size for all of the samples (3.6 mm diameter circle). An example image of each pixel pitch of the printed samples is shown in Figure 3.10. The samples are ignited using a capacitive discharge unit, and the reaction is captured using a BW Phantom camera. The burning rate is determined using a custom MATLAB code which identifies and tracks the bright region where reactions are occurring (see Appendices H–K). Figure 3.11 is a plot of the burning rate versus pixel pitch in the  $x$ -direction for each of the three batches; the error bars in the plot are the corresponding standard deviations. Still frame images for three of the samples are shown in Figure 3.12 to illustrate the differences in sample performance.

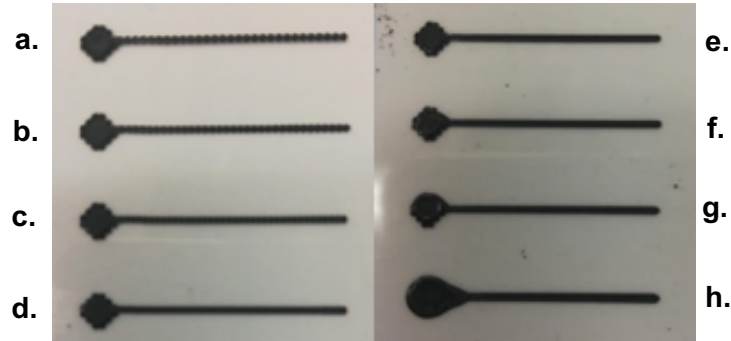


Figure 3.10 Printed one-pixel wide samples with pixel pitches of: (a) 0.8 mm, (b) 0.7 mm, (c) 0.6 mm, (d) 0.5 mm, (e) 0.4 mm, (f) 0.3 mm, (g) 0.2 mm, and (h) 0.1 mm.

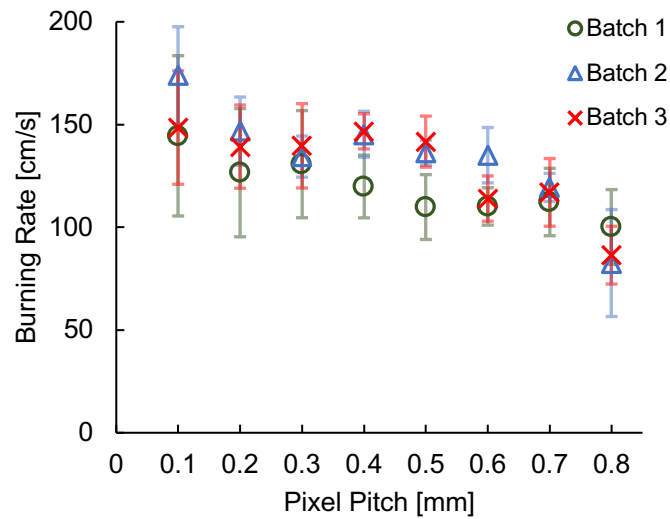


Figure 3.11 Burning rate versus pixel pitch for the one-pixel wide samples. There are seven samples per pixel pitch for batches 1 and 2 and fourteen samples per pixel pitch for batch 3 (224 total samples). The error bars are the corresponding standard deviations of the burning rates.

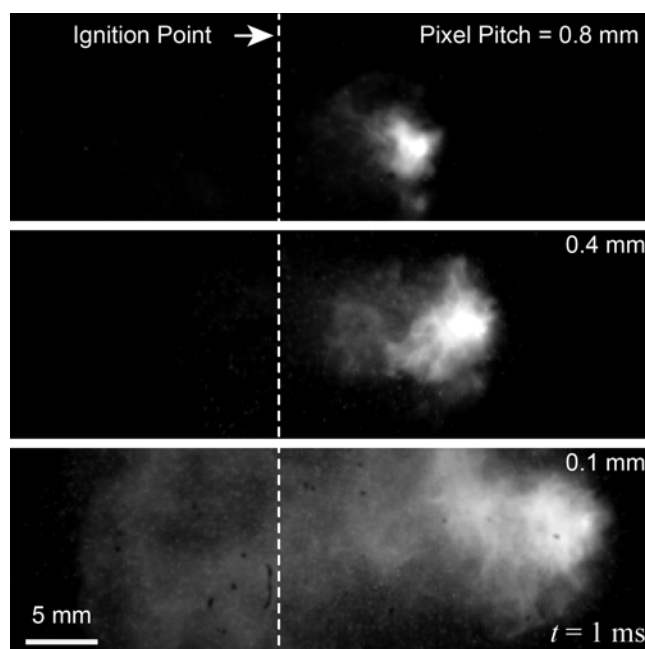


Figure 3.12 Still frames from high speed video of 0.8, 0.4, and 0.1 mm pixel pitch thermite samples taken 1 ms after ignition.

In Figure 3.11, relatively consistent results are observed within and between batches. The burning rate is also a similar speed to recent work [12], which investigated the effect of equivalence ratio on inkjet printed aluminum-bismuth (III) oxide. As the pixel pitch in the  $x$ -direction (burning rate direction) increases, the burning rate decreases. This result is consistent with a previous study [24] that showed that the burning rate of Al/MoO<sub>3</sub> increases as tube diameter increases. A single factor ANOVA test (performed using all of the data points from each of the three batches) confirms that there is a statistical difference between the mean burning rate of each pixel pitch ( $p$ -value  $< 0.001$ ). The significance of this trend is determined by calculating the effect size (the correlation coefficient Pearson's  $r$ ) for all of the samples in this study. This correlation coefficient has a possible value between -1 and 1; -1 indicating a perfect negative linear correlation and 1 indicating a perfect positive linear correlation. The result of this analysis is a correlation coefficient of -0.54, indicating that burning rate has a moderately negative trend with respect to the pixel pitch. This means that while pixel pitch is shown to affect the burning rate, it is not a dramatic effect. The standard deviations (the error bars) indicate some variability but overall consistent results.

### 3.3.3 One Droplet Wide Samples Burning Rate Versus Sample Thickness

Based on the thickness measurements from Section 3.3.1, the burning rate is plotted versus sample thickness in Figure 3.14.

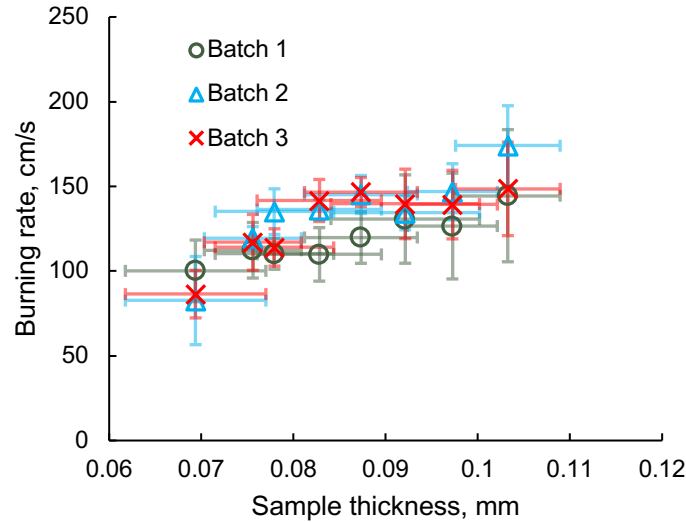


Figure 3.13: One-droplet wide samples burning rate vs. sample thickness

The significance of the trend is determined by calculating the Pearson's correlation coefficient value for all the samples in this study. The result of this analysis is a correlation coefficient of 0.66, indicating that burning rate has a moderately positive trend with respect to the sample thickness. This agrees with the pixel pitch versus burning rate result (see Section 3.3.2) which resulted in a similar trend.

### 3.3.4 y-Direction Pixel Pitch Versus Burning Rate

Samples with a fixed width of 1.9 mm are printed with a varied number of lines to increase the thermite density per sample area. These 1.9 mm wide samples are printed by varying the pixel pitch in the y-direction to fit 3–6 lines of thermite within the fixed width. An example of these printed samples is shown in Figure 3.14. Figure 3.15 is a plot of the burning rate versus pixel pitch in the y-direction for the two batches; the error bars in the plot are the corresponding standard deviations.



Figure 3.14 Varied  $y$ -pixel pitch 1.9 mm wide samples with pixel pitches of: (a) 0.2 mm, (b) 0.25 mm, (c) 0.33 mm, and (d) 0.5 mm.

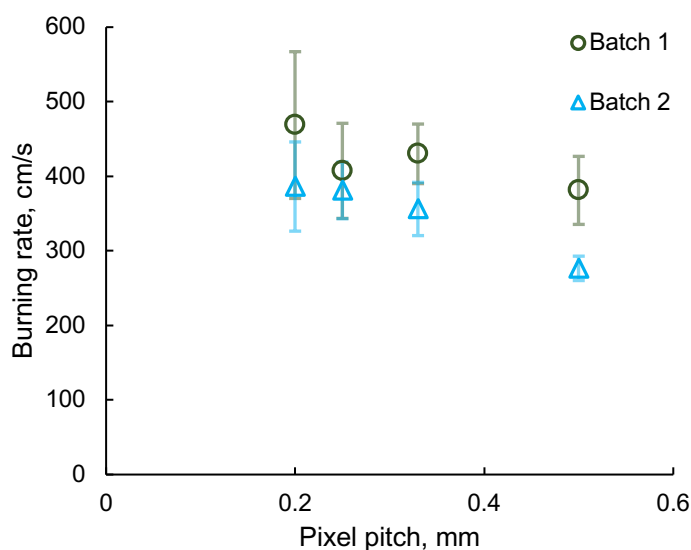


Figure 3.15 Burning rate versus the pixel pitch in the  $y$ -direction for 1.9 mm wide samples (112 total samples; 14 samples per pixel pitch per batch).

There is some batch-to-batch variability, but the values are similar, and the trend is consistent between the batches. The burning rate increases with decreasing pixel pitch in the  $y$ -direction. This trend is expected based on the results of the one-droplet wide sample study (see Section 3.3.2) since a decreasing pixel pitch in the  $y$ -direction, with a fixed width, increases thermite density per sample length and area. A single factor ANOVA (performed using all of the data points from both batches) test confirms that there is a statistical difference between the mean burning rates of each pixel pitch ( $p$ -value  $< 0.001$ ). The significance of this trend is determined by calculating the effect size (the Pearson's correlation coefficient value) for all of the samples in this study. The result of this analysis is a correlation coefficient of -0.42, indicating that burning rate has a moderate

negative trend with respect to the pixel pitch. The burning rate of these samples is significantly faster than the one-droplet wide samples (compare Figure 3.11 and Figure 3.15). This indicates that in addition to droplet density increasing burning rate, sample width also has an effect.

### 3.3.5 1.9 mm Wide Samples ( $y$ -Direction Pixel Pitch) Thickness Versus Burning Rate

Again, based on the thickness measurements from Section 3.3.1, the burning rate is plotted versus sample thickness in Figure 3.16.

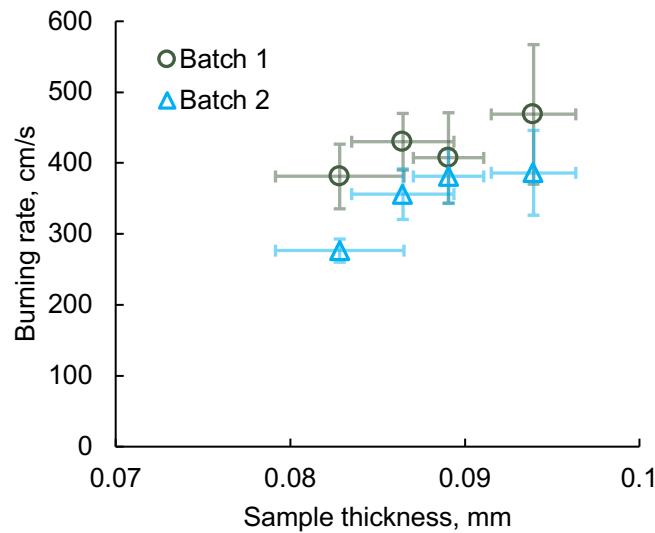


Figure 3.16 Plot of 1.9 mm wide samples burning rate vs. sample thickness

The significance of the trend is determined by calculating the Pearson's correlation coefficient for all of the samples in this study. The result of this analysis is a correlation coefficient of 0.40, indicating that burning rate has a moderately positive trend with respect to the sample thickness. This agrees with the pixel pitch versus burning rate result which resulted in a similar trend.

### 3.3.6 Burning Rate Versus Sample Width

Samples with varying widths from 0.9–3.4 mm are printed, and burning rate experiments are conducted (an example of these wider samples is shown in Figure 3.17). The pixel pitches in the  $x$ - and  $y$ -direction are held constant at 0.5 mm. The sample width is increased by printing adjacent lines (0.5 mm spacing in the  $y$ -direction) from one to six lines. Figure 3.18 is a plot of the average burning rate, from two batches (14 samples for the first two widths and 28 for the final four samples; 140 total samples) versus sample width, error bars show the corresponding standard deviations.

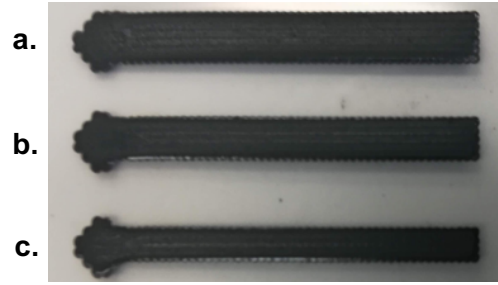


Figure 3.17 Wider burning rate samples with widths of: (a) 3.4 mm, (b) 2.9 mm, and (c) 2.4 mm.

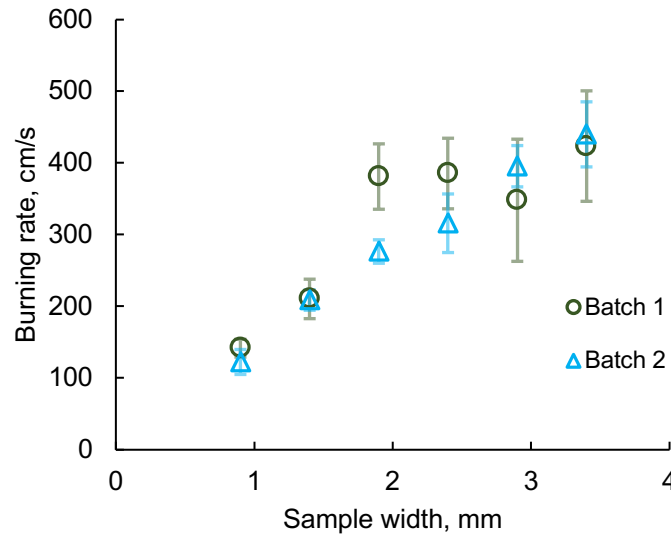


Figure 3.18 Burning rate vs. sample width (140 total samples)

There is some batch to batch variability, however, the trend is consistent between the batches. As predicted by the difference in burning rate speeds shown in the  $x$ - and  $y$ -direction pixel pitch studies (see Sections 3.3.2 and 3.3.4), as sample width increases burning rate also increases. This result is consistent with previous studies that observed similar trends of increasing burning rates with increasing sample width of printed aluminum copper (II) oxide [11] and increasing microchannel width in burning Al/MoO<sub>3</sub> [24]. A single factor ANOVA test (performed using all of the data from both of the batches) confirms that there is a statistical difference between the mean burning rate of each sample width ( $p$ -value < 0.001). The significance of this trend is determined by calculating the effect size (the correlation coefficient Pearson's  $r$ ) for all of the samples in this study. The result of this analysis is a correlation coefficient of 0.78, indicating that burning rate has a strong positive trend with respect to sample width. This result also indicates that sample width is a more significant factor for controlling burning rate than the pixel pitch.

### 3.4 Conclusions

The results of the burning rate parametric studies were shown to have inter-batch and batch-to-batch repeatability, an issue previous studies couldn't overcome. The magnitude of the burning rate is consistent with prior inkjet printed nanothermite results [11], [12]. A significant number of samples (476) were printed and demonstrated consistent, energetic performance; this indicates favorable high-volume production capabilities. The burning rate was demonstrated to increase with increasing thermite droplet density in both the  $x$ - and  $y$ -directions (moderate correlation). These correlations were consistent with the burning rate trends versus sample thickness. An increase in burning rate also occurred with increasing sample width (strong correlation). The physical phenomena causing these trends are examined in Chapter 4 through the development of an analytical thermal model. The results presented in this work further demonstrate the ability to precisely deposit and print energetic material and provide a more fundamental understanding of how burning rate can be controlled by varying deposited thermite droplet density and sample width.

## CHAPTER 4. ANALYTICAL THERMAL MODEL

### 4.1 Introduction

The work presented in this chapter focuses on the development of an analytical model to approximate the physical phenomena that led to the observed burning rate trends in the experiments. An energy balance coupled with a simple Arrhenius rate [14] expression is developed to predict the flame temperature and burning rate of an inkjet-printed nanothermite sample as a function of the sample's geometry and energetic properties. The governing equations, associated model inputs, and solution method are described in Sections 4.2.1, 4.2.2, and 4.2.3 respectively.

### 4.2 Model Description

#### 4.2.1 Energy Balance

The present analysis assumes that the propagation speed of a reaction is a function of the rate that heat is transferred from the hot, reaction zone of burning material to the adjacent section of the same material. This idea of heat transfer driving the reaction forward has been used to explain trends in a variety of combustion applications [25]–[31]. An energy balance is performed based on the burning rate studies presented in the previous chapter. A steady burning flame is considered. The printed thermite samples were ignited with a capacitive discharge device. Energy is released as the material reacts. A portion of this energy is conducted through the sample, heating the adjacent region until it reaches the ignition temperature and reacts (see Figure 4.1). The rest of the generated energy is lost to the surroundings. The energy balance on the reaction zone is given by,

$$\dot{q}_{\text{gen}} = \dot{q}_{\text{cond}} + \dot{q}_{\text{loss}} . \quad (4.1)$$

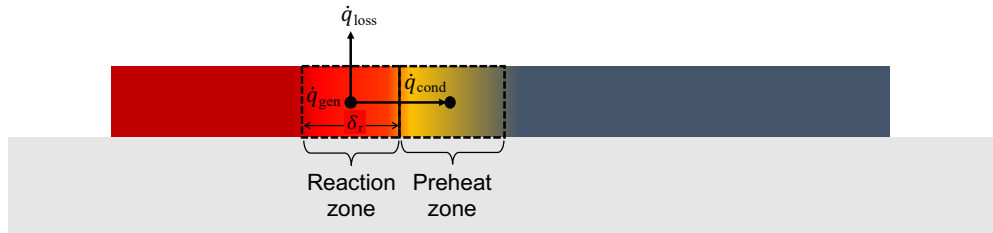


Figure 4.1 Energy balance of a burning nanothermite sample.

Here  $\dot{q}_{\text{gen}}$  is the heat generated by the reacting thermite,

$$\dot{q}_{\text{gen}} = \dot{m} q_{\text{ad}} A_c, \quad (4.2)$$

where  $q_{\text{ad}}$  is the adiabatic heat of reaction,  $\dot{m}$  is the rate of mass consumption per unit area (*i.e.*, the mass flux of burning thermite), and  $A_c$  is the cross-sectional area of the thermite sample. The mass flux is modeled assuming a zeroth order global condensed phase reaction as [32]

$$\dot{m} = ce^{\left[\frac{-E_a}{2R_u T_f}\right]}, \quad (4.3)$$

where  $c$  is the preexponential factor,  $E_a$  is the activation energy,  $R_u$  is the universal gas constant, and  $T_f$  is the flame temperature. Note that the propagation speed (*i.e.*, the burning rate),  $S_L$ , is a function of the burning mass flux  $\dot{m}$ ,

$$S_L = \frac{\dot{m}}{\rho}, \quad (4.4)$$

where  $\rho$  is the density of the thermite.

Both the conduction and energy loss term in Eq. 4.1, can be modeled using a thermal resistance network to simplify the heat transfer mechanisms while still elucidating the phenomena that control the burning rate. Figure 4.2 shows the thermal resistance network that accounts for heat generation due to the reaction, conduction through the sample, and heat loss through the top and sides of the sample. The heat transfer through the bottom of the sample is assumed to be negligible because the substrate has a low thermal diffusivity ( $\alpha_{\text{substrate}} = 9 \times 10^{-8} \text{ m}^2/\text{s}$ ) in comparison to the thermal diffusivity of thermite ( $\alpha_{\text{thermite}} = 2 \times 10^{-7} \text{ m}^2/\text{s}$ , see Section 4.2.2); this results in a significantly slower heat transfer to the substrate compared with the timescale of conduction through the sample and convection to the ambient.

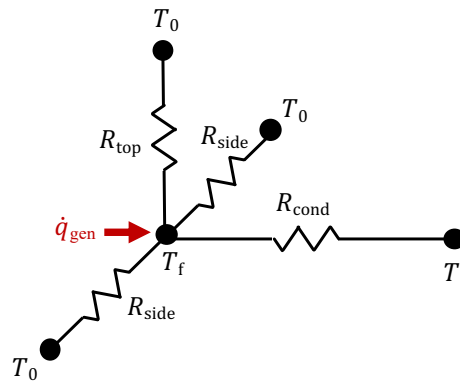


Figure 4.2 Thermal circuit energy balance of a burning nanothermite sample.

The conductive resistance between the reacting zone and the preheat zone (see Figures 4.1 and 4.2) is calculated by

$$R_{cond} = \frac{\delta_r}{kA_c}, \quad (4.5)$$

where  $k$  is the thermal conductivity of the thermite,  $A_c$  is the cross-sectional area of the thermite sample, and  $\delta_r$  is the reaction zone thickness [32] computed by

$$\delta_r = \frac{\alpha}{S_L} \frac{R_u T_f}{E_a}. \quad (4.6)$$

This resistance is used to calculate the conductive heat transfer as

$$\dot{q}_{cond} = \frac{1}{R_{cond}} (T_f - T_i) = \frac{k}{\delta_r} A_c (T_f - T_i). \quad (4.7)$$

By assuming that the heat loss from the top and sides can be modeled as a convective resistance, the thermal resistance can be written as [33],

$$R_{top} = \frac{1}{h_{top} A_{top}} \quad (4.8)$$

and

$$R_{side} = \frac{1}{h_{side} A_{side}} \quad (4.9)$$

The total resistance between the reaction zone and the ambient is given by

$$R_{total} = \left( \frac{1}{R_{side}} + \frac{1}{R_{top}} + \frac{1}{R_{side}} \right)^{-1} = \frac{1}{h_{top} A_{top} + 2h_{side} A_{side}} \quad (4.10)$$

Thus, the heat loss rate to ambient in Equation 4.3 can be written as

$$\dot{q}_{loss} = \frac{1}{R_{total}} (T_f - T_0) = (h_{top} A_{top} + 2h_{side} A_{side}) (T_f - T_0). \quad (4.11)$$

The  $h$  terms in the above equations are the convective heat transfer coefficients at the top and side respectively. These are calibrated parameters by fitting the model to the experimentally measured burning rate values. With the fitted heat transfer coefficients, the change in burning rate is predicted. The terms  $A_{top}$  and  $A_{side}$  are top and side surface areas of the sample respectively. They are calculated from the sample width and thickness as  $w \times \delta_r$  and  $t \times \delta_r$ , respectively.

### 4.2.2 Model Inputs

The nanothermite used in this thesis is composed of 77% Bi<sub>2</sub>O<sub>3</sub> and 33% Al by volume (91.8% Bi<sub>2</sub>O<sub>3</sub> and 8.2% Al by mass). The Bi<sub>2</sub>O<sub>3</sub> particles have a mean diameter of 38 nm while the Al nanoparticles have a mean diameter of 80 nm. The theoretical porosity for a random-close packing of bidisperse spherical particles with these characteristic—a large-to-small diameter ratio of 2 and 77% volume fraction of small particles—is approximately 44% [34]. Based on Mezger *et al.* [35] all sample thermophysical properties are calculated assuming that the thermite reaches 68% of the theoretical maximum packing density; this yields a printed porosity of 55%. The resulting volume and mass fractions of the three components of the thermite samples—Al, Bi<sub>2</sub>O<sub>3</sub>, and air—are listed in Table 4.1 along with the thermophysical properties of each individual constituent.

Table 4.1 Volume fraction, mass fraction, and thermophysical properties of the constituents of the thermite sample.

	volume fraction $\phi$ [-]	mass fraction $\omega$ [-]	density $\rho$ [g/cm <sup>3</sup> ]	specific heat capacity $c_p$ [J/kg-K]	thermal conductivity $k$ [W/m-K]
Bi <sub>2</sub> O <sub>3</sub>	0.347	0.9181	8900	258	2.2
Al	0.102	0.0817	2700	900	250
Air	0.551	0.0002	1.23	1020	0.015
Sample			3364	310	$0.027 < k < 26$

The density,  $\rho$ , of the thermite sample is computed as [34]

$$\rho = \sum_{i=1}^3 \phi_i \rho_i \cdot \quad (4.12)$$

where  $\phi_i$  is the solid volume fraction of the  $i^{\text{th}}$  constituent (*i.e.*, Bi<sub>2</sub>O<sub>3</sub>, Al, or air). The specific heat capacity,  $c_p$ , of the thermite sample is computed as [34]

$$c_p = \sum_{i=1}^3 \omega_i c_{p,i} \cdot \quad (4.13)$$

where  $\omega_i$  is the mass fraction of the  $i^{\text{th}}$  constituent. While the thermal conductivity of the sample is difficult to determine without experimental measurements, the theoretical upper and lower bounds for the effective thermal conductivity of a multi-component system can be computed using the weighted arithmetic mean of the constituent's conductivities [36],

$$k_{\text{upper-bound}} = \sum_{i=1}^3 \phi_i k_i \quad (4.14)$$

and weighted harmonic mean [36],

$$k_{\text{lower-bound}} = \frac{1}{\sum_{i=1}^3 \frac{\phi_i}{k_i}} \quad (4.15)$$

respectively. The resulting sample thermophysical properties are listed in Table 4.1. The thermal conductivity is assumed to be 0.2 W/m-K (near the lower theoretical limit of  $k$ ) for the purpose of modeling this reaction. Based on this assumed  $k$ , the sample's thermal diffusivity is  $1.9 \times 10^{-7} \text{ m}^2/\text{s}$ .

Several additional model inputs are required to solve the system of governing equations described in Section 4.2.2 above including temperatures, reaction rate constants, and heat transfer coefficients. These inputs and their sources are summarized in Table 4.2 and described in more detail below.

Table 4.2 Summary of model inputs.

Model Input	Value	Source
Ambient temperature, $T_0$	298 K	measured
Ignition temperature, $T_i$	773 K	Piekiel <i>et al.</i> [37]
Adiabatic flame temperature, $T_i$	3319 K	Puszynski <i>et al.</i> [38]
Adiabatic heat of reaction, $q_{\text{ad}}$	506.1 cal/g	Fischer and Grubelich [39]
Pre-exponential constant, $c$	65232 kg/m <sup>2</sup> s	fitted
Activation energy, $E_a$	74037 J/mol	fitted
Heat transfer coefficient for the sample side, $h_{\text{side}}$	$2 \times 10^{10} \text{ W/m}^2\text{K}$	fitted
Heat transfer coefficient for sample top, $h_{\text{top}}$	$2 \times 10^9 \text{ W/m}^2\text{K}$	fitted

The ambient temperature was maintained at 298 K for all the burning rate experiments. Piekiel *et al.* [37] measured the ignition temperature of Al/Bi<sub>2</sub>O<sub>3</sub> to be between 749 and 797 K; the present analysis used the average of these two values (*i.e.*,  $T_i = 773 \text{ K}$ ). The adiabatic heat of reaction for Al/Bi<sub>2</sub>O<sub>3</sub> was determined by Fischer and Grubelich [39] to be  $q_{\text{ad}} = 506.1 \text{ cal/g}$ . The value of the fitted activation energy,  $E_a$ , is similar to the experimental value determined by Puszynski *et al.* [40],  $E_a = 60,800 \text{ kJ/mol}$ . The activation energy,  $E_a$ , and the pre-exponential factor,  $c$ , were selected so that (1) the heat generation rate exactly equals the conduction losses in an adiabatic sample, *i.e.*

$$\dot{q}''_{\text{gen}} = ce^{\frac{-E_a}{2R_u T_i}} q_{\text{ad}} = \frac{k}{\delta_r} (T_{\text{f,ad}} - T_i) = \dot{q}''_{\text{cond}}, \quad (4.16)$$

and (2) that the propagation speed of an adiabatic sample is 507 cm/s,

$$S_{L,\text{ad}} = \rho c e^{\frac{-E_a}{2R_u T_{\text{f,ad}}}} = 507 \text{ cm/s}, \quad (4.17)$$

where the adiabatic flame temperature nano-Al/Bi<sub>2</sub>O<sub>3</sub> was measured as  $T_{f,ad} = 3319$  K by Puszynski *et al.* [38]. The propagation speed of an adiabatic sample is approximated based on the burning rate of an infinitely wide sample, which was found to be 507 cm/s by extrapolating from experimental measurements (see Figure 4.2). The heat transfer coefficients ( $h_{side}$  and  $h_{top}$ ) were selected to match the experimental trend of increasing burning rate with increasing sample thickness and width. For this simplified model to predict the experimental trends, the convective heat transfer coefficients were raised to very large values.

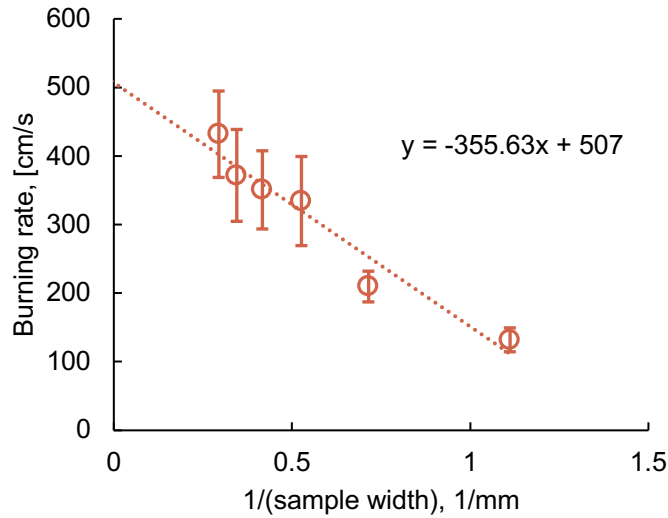


Figure 4.3 Experimental burning rate plotted as a function of the inverse of sample width. These data are fitted to extrapolate the expected burning rate of an infinitely-wide sample.

#### 4.2.3 Solution Methods

The governing equations and model inputs as described in Sections 4.2.1 and 4.2.2 respectively were implemented in Engineering Equation Solver (EES, by F-Chart Software). The equations were solved iteratively until the relative residual was less than  $1 \times 10^{-4}$ . Two parametric tables were created to determine how varying the sample width and thickness affected burning rate. The EES script used for these calculations is found in Appendix L.

### 4.3 Model Results

The sample thickness in the model is varied from 0.069 mm to 0.12 mm while the width is held constant at 0.9 mm. This is to replicate the samples used in the  $x$ -direction pixel-pitch study (see Section 3.3.2). Figure 4.4 is a plot of burning rate versus sample thickness.

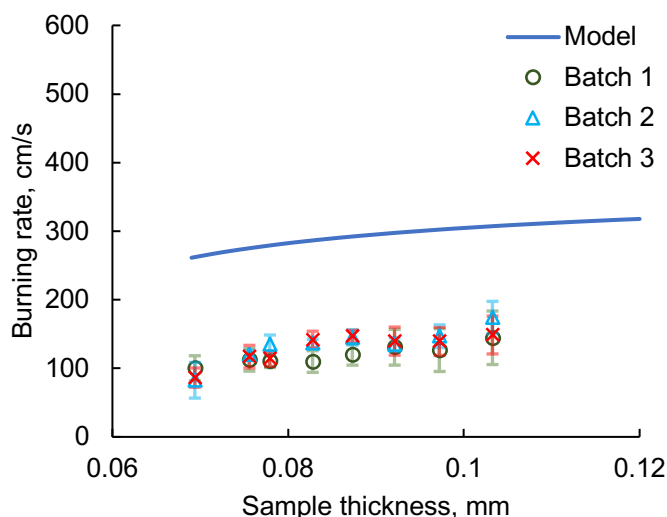


Figure 4.4 Burning rate vs. sample thickness model prediction and experimental values (constant sample width of 0.9 mm).

The model predicts a trend of increasing flame temperature and burning rate with increasing sample thickness. This matches the observed experimental trend of increasing burning rate with decreasing pixel-pitch, corresponding to a thicker sample. This qualitative agreement between the analytical model and experimental results indicates that heat losses decrease with increasing sample thickness (decreasing pixel pitch) resulting in higher flame temperatures and thus faster burning rate. Note that the magnitude of the model predictions differs from experimental measurements; this may be due to many different factors. The model seems to predict the general trend that burning rate increases with sample thickness. However, the somewhat consistent offset in the actual burning rate suggests that there are more substantive factors that need to be considered in the model. One possible factor is the percentage of active material in the printed sample. As can be deduced from the actual printed geometry, there are voids in addition to the reactive material. The ratio between void and material was not accounted for in the model, which can, in part, account for the offset in measured burn rate. Further research is required to determine which effects control the reaction rate in addition to the heat losses.

The sample width is varied from 0.9 mm to 4 mm while the thickness is held constant at 0.1 mm. This is to replicate the samples used in the width study (see Section 3.3.6). Figure 4.5 is a plot of burning rate versus sample width.

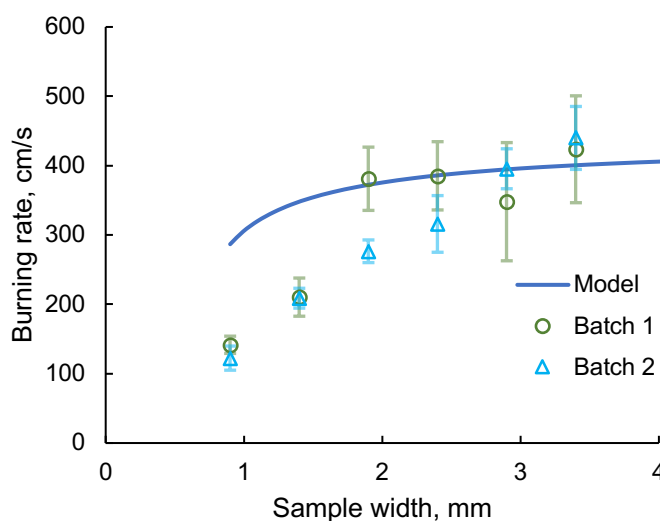


Figure 4.5 Burning rate vs. sample width model prediction and experimental values.

The model predicts a trend of increasing flame temperature and burning rate with increasing sample width. This is expected based on the experimental trend of increasing burning rate with increasing sample width. The model shows that as samples increase in width, the heat loss is less significant than the increase in heat generation from the nanothermite.

#### 4.4 Conclusions

A thermal model was developed based on an energy balance equating the rate of heat generation by the thermite reaction with the heat loss through conduction within the thermite and convection to the ambient air. The thickness and width of the samples were varied to compare with the experimental results. The model predicted the similar trends to those observed experimentally (see Chapter 3). The model indicates that the increase in heat generation in both the thicker and wider samples offsets the increased heat loss to the environment (caused by higher surface area). This results in higher flame temperature and subsequently faster burning rate. For this simplified model to predict the experimental trends, the convective heat transfer coefficients were raised to very large values; this indicates that the reaction observed in the experiments presented in this thesis is

more complex than a simplified model can predict. A more complicated model would be required to fully capture the physical phenomena that caused the experimental burning rate trends and accurately predict the burning rates.

## CHAPTER 5. CONCLUSIONS

The main objective of this thesis was to determine the ability to control the burning rate of printed Al/Bi<sub>2</sub>O<sub>3</sub> by varying printing parameters and sample geometry. This work included a systematic approach to quantifying printer performance and detailed experimental investigations with statistically significant results. In addition to highly repeatable experimental results, this work presented an analytical thermal model that provided insight into the physical phenomena causing the observed burning rate trends. The main conclusions from this work are highlighted in this chapter along with an outline of recommendations for future work.

### 5.1 Conclusions

A systematic approach to quantify the performance of a piezoelectric inkjet printer was presented in Chapter 2. A custom-built inkjet printing setup was used to control the deposition of the nanothermite ink. The ability of the system to accurately deposit nanothermite ink is examined by imaging droplets as they leave the printer nozzle and after they have dried on the substrate. Droplets were shown to print with consistent size, dried droplets were also shown to have similar size (0.023 mm standard deviation in diameter), and droplet spacing was demonstrated to have sub-millimeter precision (0.037 mm standard deviation). Additionally, a printing frequency of 10 Hz was demonstrated to result in smooth surfaced samples.

Based on the results of the printing performance study, three burning rate studies were conducted and discussed in Chapter 3. The burning rate was demonstrated to increase with increasing thermite droplet density in both the *x*- and *y*-directions (moderate correlation). An increase in burning rate also occurred with increasing sample width (strong correlation). An increase in burning rate was also observed with increased sample thickness. The increase in burning rate is likely due to less heat loss in thicker/wider samples. A significant number of samples (476) were printed and demonstrated consistent, energetic performance; this indicates favorable high-volume production capabilities.

An analytical thermal model was developed in Chapter 4 to explain the physical phenomena that led to the burning rate trends presented in Chapter 3. The thickness and width of the samples were

varied to compare with the experimental results. The model predicted similar trends observed in the experiments (see Chapter 3). The model indicates that the increase in heat generation in both the thicker and wider samples offsets the increased heat loss to the environment (caused by higher surface area). This causes an increase in flame temperature and subsequently burning rate.

## 5.2 Future Work

The following is recommended future work for this research topic. The first recommendation is to determine the ability of printed nanothermite to propagate across gaps and around tight turns (*e.g.*, 90-degree angle) and to examine the feasibility of using printed nanothermite as an energetic delay. A potential issue with integrating printed thermite into electronic systems is that due to the highly exothermic reaction of the thermite, it is possible that reacting thermite could ignite neighboring thermite regions. If an electronic system has multiple regions of printed thermite that are intended to ignite at different times, it is necessary to determine what gap distance is necessary to prevent unwanted reactions. Two experiments are proposed to determine this gap distance. The first experiment consists of printing a line of thermite droplets, varying the droplet spacing, and igniting the line. This will provide insight into the ability of a small amount of thermite to propagate across gaps. The second experiment consists of printing adjacent vertical lines and varying the gap between the lines. This will show at what gap distance adjacent lines need to be spaced to prevent cross ignition. Another application for this work is in energetic delays. This thesis presented the ability to accurately control the burning rate of printed thermite through printing parameters and sample geometry. Another method for controlling the time it takes for the reaction to get from one point to another is to print geometries with turns and curves. The goals of this effort will be to determine the ability of printed thermite to continue reacting around tight turns (*e.g.*, 90-degree angle) and to determine the feasibility of using printed thermite as a delay (*i.e.*, determine how long of a delay printed thermite can provide).

Another area to further explore involves developing techniques for printing thicker samples and examining additional methods for accurately measuring the thickness of printed samples. Decreasing the pixel pitch does result in thicker samples, however, as the droplets are printed too close together the sample tends to bulge (for the 0.9 mm wide samples in this thesis, the samples began to bulge at a pixel pitch in the *x*-direction of less than 0.1 mm). In order to print thicker samples, it is necessary to print multiple layers. To reduce sample bulging, it is recommended that

a wider layer of thermite be printed as a base and that subsequent layers be thinner (this process is similar to a pyramid shape).

The final recommendation is to develop a more comprehensive combustion model to predict the burning rate of printed thermite accurately. The simplified model showed that in order to predict the experimental trends, the convective heat transfer coefficients had to be raised to non-physical values. This indicated that the experimental burning rate trend was not driven purely by simple convective heat losses. A higher fidelity model is necessary to accurately predict these trends.

## APPENDIX A. DESCRIPTION OF THE MATLAB CODE USED TO MEASURE DROPLET DIAMETER FROM SIDEVIEW IMAGES

Droplet diameter is extracted from a binarized version of each frame using a Hough Circle transformation in MATLAB. This method is implemented in MATLAB through the following script:

- ‘MAIN\_Droplet\_Sideview\_Size.m’ – the main code reads in a side-view image video, parses it into individual frames and measures the droplet sizes. This code additionally takes a pixel-size calibration number as an input for equating pixels in the video with length. Note that the calibration value is determined from a calibration image taken separately before each side-view imaging experiment. This code is provided in Appendix B.

The video is read into MATLAB using the built-in function ‘VideoReader’. Each individual frame is then saved as a .png image in the local directory ‘individual\_images/’ named based on the frame number as ‘img0001.png’, ‘img0002.png’, ‘img0003.png’ *etc.* The code then loops through the each of the image files in the directory ‘individual\_images/’ and executes the following commands:

1. The image is converted from red-green-blue (RGB) to gray-scale using the built-in MATLAB function ‘rgb2gray’.
2. The next step checks to see if the frame was taken out of sync with the LED light (*i.e.*, the image is black). First, the average luminescence of the image is calculated by taking the mean value of all the gray-scale image’s pixels. If this average luminescence is less than 65, steps 3–4 are skipped for this image and the code proceeds to analyze the next image starting at step 1.
3. A black-and-white image matrix, BW, is generated based on a user-defined threshold value where all of the pixels with intensities less than the threshold are set to a value of zero (black) and all of the pixels with intensities greater than the threshold are set to a value of one (white).
4. The built-in MATLAB function ‘imfindcircles’ is used to identify circles in the black-and-white image via the circular Hough transform [41]. This function outputs the diameter of the droplet in this specific image.

Steps 1–4 are repeated for each frame of the video storing the droplet diameter for each frame. The code then plots the diameter as a function of droplet number and computes the mean and standard deviation of the measurement.

## APPENDIX B. MAIN MATLAB CODE FOR MEASURING DROPLET DIAMETER FROM SIDEVIEW IMAGES

Filename: MAIN\_Droplet\_Sideview\_Size.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Created: 03/21/18
%Author: Forrest Son
%Purpose: measure the droplet diameter from side-view images of printer
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear all; close all;
% read in video file and save as a series of tiff images
v = VideoReader('Transient_thermitetest05_2000drops_20Hz.avi')
Nframes = v.Duration*v.FrameRate;
i = 1;
while hasFrame(v)
    img = readFrame(v);
    filename = [sprintf('%04d',i) '.png'];
    fullname = fullfile('individual_images/',filename);
    imwrite(img,fullname)    % Write out to a png file (img0001.png, img0002.png, etc.)
    i = i+1;
end

% calibration
cal = 382; %382 px = 1 mm from calibration image

% post-process images to find the droplet size
images = dir('individual_images/*.png');
binaryThreshold = 75;
sensitivity = 0.93;
radii = NaN(length(images),1);
centers = NaN(length(images),2);
number_of_images = num2str(length(images));
for i=1:length(images)
    str = strcat('processing image :',num2str(i),' / ',number_of_images);
    disp(str)
    r=0;
    img = imread(['individual_images/',images(i).name]);
    % check to see if the image is black, if so, skip and move to next
    % image
    lum(i) = mean2(rgb2gray(img));
    if lum(i) < 65
        continue;
    end;
    [BW,~] = createMask_Lab(img,binaryThreshold);
    [c, r] =
imfindcircles(BW,[75,100],'ObjectPolarity','dark','Sensitivity',sensitivity);
    if r>0
        radii(i) = r;
        centers(i,:) = c;
    else;
        radii(i) = NaN;
        centers(i,:) = [NaN NaN];
    end
    clear c r img BW maskedRGBImage
end

```

```

%% Plot volume
% scatter(1:length(radII),4/3*pi*(radII/cal).^3*1000);
dropNum=(1:length(radII))-1552;
dropNumPlot = dropNum(radII>10);
volumePlot = 4/3*pi*(radIIPlot/cal).^3*1000;
figure; plot(dropNumPlot,volumePlot);
xlim([0,1200]);
ylim([0,50]);
ylabel('Volume, nL');
xlabel('Droplet Number');
fig_size = [0,0,240,180]*1.5;
set(gcf, 'Position', fig_size)

%% mean & standard deviation of droplet
avg_volume = mean(volumePlot(dropNumPlot<1200))
stdev_volume = std(volumePlot(dropNumPlot<1200))

%% analysis
% count the number of circles for which we successfully measured the radius
idx=sum(~cellfun(@isempty,radII),2);
disp(i);

%% check if the circle looks good (single image)
% i = 113 % the image number you want to read in
img = imread(['individual_images/',images(i).name]);
figure; imshow(img)
h=viscircles(centers(i,:),radII(i),'EnhanceVisibility',0,'LineWidth',0.1,'Color','w');

```

## APPENDIX C. DESCRIPTION OF THE MATLAB CODE USED TO MEASURE THE PRINTED DROPLET DIAMETER AND SPACING

The equivalent circle area diameter and centroid of each droplet are extracted from an image of a grid of printed droplets. This code is based on the tutorial of Brett Shoelson for segmenting an image of multiple circular objects [20]. This method is implemented in MATLAB through the following script:

- ‘MAIN\_GridAnalyzer.m’ – the main code reads in a droplet grid image and measures the droplet size and spacing. This code additionally takes a pixel-size calibration number as an input for equating pixels in the video with length. Note that the calibration value is determined from a calibration image. This code is provided in Appendix D.

The image of a printed droplet grid is read into MATLAB using the built-in function ‘imread’. The original red-green-blue (RGB) image is converted to grayscale using the built-in function ‘rgb2gray’; then the grayscale image is converted to a black-and-white image based on a user-defined threshold. This thresholding effectively segments the image into regions with pure-substrate and regions with printed thermite. Each individual printed droplet is given a numeric label using the built-in MATLAB function ‘bwlabel’; the size and centroid (*i.e.*, location) of each droplet are measured by ‘regionprops’, another built-in MATLAB function. The edges of each droplet are overlaid on the original RGB image of the droplet grid so the user can verify that the image was processed correctly.

After segmenting the image to identify each droplet, the average distance between droplets is computed. This is done by first calculating the distances between all the possible droplet pairs by comparing centroids. This leaves a list of 80 distances for each individual droplet; the nearest neighbors are then identified as droplets horizontal and vertical neighbors (*i.e.*, droplets with centroids less than 3.2 mm away). The mean and standard deviation between neighboring droplets is then calculated.

## APPENDIX D. MAIN MATLAB CODE FOR MEASURING PRINTED DROPLET DIAMETER AND SPACING

Filename: MAIN\_GridAnalyzer.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Created: 04/13/18
%Author: Forrest Son
%Purpose: measure the droplet diameter from side-view images of printer
%Requires: the Image Processing Toolbox (IPT)
% based on a tutorial found at
% https://blogs.mathworks.com/pick/2009/11/06/segmenting-coinsa-tutorial-on-blob-
analysis/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; close all; clearvars;
imtool close all;
format long g;
format compact;
captionFontSize = 14;

baseFileName = '2018-04-13 image of grid (0.5mmpp).png'; % images taken on Monday April
16
thresholdValue = 100;
IMG = imread(baseFileName);
%% Convert image from RGB to grayscale
[rows, columns, numberOfColorChannels] = size(IMG);
if numberOfColorChannels > 1
    originalImageRGB = IMG;
    IMG = rgb2gray(IMG);
end

%% Display the grayscale image.
subplot(2, 2, 1);
imshow(IMG);
set(gcf, 'units','normalized','outerposition',[0 0 1 1]); % Maximize the figure
window.
drawnow;
caption = sprintf('Original image \nshowing centroids');
title(caption, 'FontSize', captionFontSize);
axis image;

% Histogram of image levels
[pixelCount, grayLevels] = imhist(IMG);
subplot(2, 2, 2);
bar(pixelCount);
title('Histogram', 'FontSize', captionFontSize);
xlim([0 grayLevels(end)]);
grid on;

%% Threshold
BW = IMG < thresholdValue;
se = strel('disk',2);
C = imdilate(BW,se);
D = bwareaopen(C,10000);
BW = imfill(D, 'holes'); % fill in any internal holes

```

```

% Show the threshold as a vertical red bar on the histogram.
hold on;
maxYValue = ylim;
line([thresholdValue, thresholdValue], maxYValue, 'Color', 'r');
% Place a text label on the bar chart showing the threshold.
annotationText = sprintf('Thresholded at %d gray levels', thresholdValue);
text(double(thresholdValue + 5), double(0.5 * maxYValue(2)), annotationText,
'FontSize', 10, 'Color', [0 .5 0]);
text(double(thresholdValue - 70), double(0.94 * maxYValue(2)), 'Background',
'FontSize', 10, 'Color', [0 0 .5]);
text(double(thresholdValue + 50), double(0.94 * maxYValue(2)), 'Foreground',
'FontSize', 10, 'Color', [0 0 .5]);

%% Identify individual droplets
labeledImage = bwlabel(BW, 8); % Label each droplet for later analysis
dropMeasurements = regionprops(labeledImage, IMG, 'all');
numberOfBlobs = size(dropMeasurements, 1);

% bwboundaries() returns a cell array, where each cell contains the
% row/column coordinates for an object in the image.
% Plot the borders of all the droplets on the original grayscale image
% using the coordinates returned by bwboundaries.
subplot(2, 2, 4);
imshow(originalImageRGB);
title('Outlines', 'FontSize', captionFontSize);
axis image; hold on;
boundaries = bwboundaries(BW);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;

% Get centroids.
% We can get the centroids of ALL the droplets into 2 arrays,
% one for the centroid x values and one for the centroid y values.
allBlobCentroids = [dropMeasurements.Centroid];
centroidsX = allBlobCentroids(1:2:end-1);
centroidsY = allBlobCentroids(2:2:end);
allBlobAreas = [dropMeasurements.Area];

%% rainbow plot
% Assign each droplet a different color.
coloredLabels = label2rgb (labeledImage, 'hsv', 'k', 'shuffle'); % pseudo random color
labels
% coloredLabels is an RGB image. We could have applied a colormap instead (but only
with R2014b and later)
subplot(2, 2, 3);
imshow(coloredLabels);
axis image; % Make sure image is not artificially stretched because of screen's aspect
ratio.
caption = sprintf('Pseudo colored labels, from label2rgb().\nBlobs are numbered from
top to bottom, then from left to right.');
```

```

title(caption, 'FontSize', captionFontSize);

%% print droplet properties to command window
textFontSize = 9; % Used to control size of "droplet number" labels put atop the
image.
labelShiftX = -300; % Used to align the labels in the centers of the coins.
labelShiftY = 750; % Used to align the labels in the centers of the coins.

```

```

blobECD = zeros(1, numberOfBlobs);
% Print header line in the command window.
fprintf(1, 'Blob #      Mean Intensity  Area  Perimeter  Centroid
Diameter\n');
% Loop over all droplets printing their measurements to the command window.
for k = 1 : numberOfBlobs % Loop through all droplets.
    % Find the mean of each droplet.
    thisBlobsPixels = dropMeasurements(k).PixelIdxList; % Get list of pixels in
current droplet.
    meanGL = mean(IMG(thisBlobsPixels)); % Find mean intensity
    meanGL2008a = dropMeasurements(k).MeanIntensity;
    blobArea = dropMeasurements(k).Area; % Measure the droplet area.
    blobPerimeter = dropMeasurements(k).Perimeter; % Measure droplet perimeter.
    blobCentroid = dropMeasurements(k).Centroid; % Locate each droplets
centroid
    blobECD(k) = sqrt(4 * blobArea / pi); % Compute ECD - Equivalent
Circular Diameter for each droplet.
    fprintf(1, '%2d %17.1f %11.1f %8.1f %8.1f % 8.1f\n', k, meanGL, blobArea,
blobPerimeter, blobCentroid, blobECD(k));
    % Put the "droplet number" labels on the "boundaries" grayscale image.
    text(blobCentroid(1) + labelShiftX, blobCentroid(2)+ labelShiftY, num2str(k),
'FontSize', textFontSize, 'Color', 'k', 'FontWeight', 'Bold');
end

%% Put droplet number labels on the rgb labeled image
subplot(2, 2, 3);
for k = 1 : numberOfBlobs % Loop through all blobs.
    text(centroidsX(k)+ labelShiftX , centroidsY(k)+ labelShiftY, num2str(k),
'FontSize', textFontSize, 'Color', 'w', 'FontWeight', 'Bold');
end

%% Plot the centroids in the original image in the upper left.
subplot(2, 2, 1);
hold on;
for k = 1 : numberOfBlobs
    % Identify if blob #k is a droplet or noise.
    isDrop = allBlobAreas(k) > 200000; % big enough to be a droplet?
    if isDrop
        % Plot drops with a red +.
        plot(centroidsX(k), centroidsY(k), 'r+', 'MarkerSize', 5, 'LineWidth', 1);
    else
        % Plot noise with a blue x.
        plot(centroidsX(k), centroidsY(k), 'bx', 'MarkerSize', 5, 'LineWidth', 1);
    end
end
end
elapsedTime = toc;

%% generate large image with centroid and edges marked
% Original image with outlines and centroids;
% Display the grayscale image.
figure;
imshow(originalImageRGB); hold on;
drawnow;
for k = 1 : numberOfBlobs
    isDrop = allBlobAreas(k) > 200000; % Small blobs, too small to be a droplet.
    if isDrop
        plot(centroidsX(k), centroidsY(k), 'r+', 'MarkerSize', 5, 'LineWidth', 1);
        %
        text(centroidsX(k)+ labelShiftX , centroidsY(k)+ labelShiftY, num2str(k),
'FontSize', textFontSize, 'Color', 'k', 'FontWeight', 'Bold');
    end
end
end
boundaries = bwboundaries(BW);

```

```

numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;

%% find distance from each droplet to its nearest neighbors
isDrop = allBlobAreas > 200000; % Small blobs, too small to be a droplet.
distX = zeros(numberOfBlobs);
distY = zeros(numberOfBlobs);
dist = zeros(numberOfBlobs);
neighbor = zeros(numberOfBlobs);

for k = 1 : numberOfBlobs
    for j = 1 : numberOfBlobs
        if isDrop(k) && isDrop(j)
            distX(k,j) = centroidsX(k) - centroidsX(j);
            distY(k,j) = centroidsY(k) - centroidsY(j);
        else
            distX(k,j) = 4000; % (centroidsX(k) - centroidsX(j))*100
            distY(k,j) = 4000; % (centroidsY(k) - centroidsY(j))*100
        end
        dist(k,j) = sqrt(distX(k,j)^2 + distY(k,j)^2);
    end
end
neighbor = dist < 2450;

for k = 1 : numberOfBlobs
    numNeighbor(k) = sum(neighbor(k,:));
end;

dist_neighbors_only = neighbor.*dist;
dist_neighbors_only = triu(dist_neighbors_only);
dist_neighbors_only(dist_neighbors_only < 1) = NaN;

inc = 1;
for k = 1 : numberOfBlobs
    for j = 1 : numberOfBlobs
        if ~isnan(dist_neighbors_only(k,j))
            dist_array(inc) = dist_neighbors_only(k,j);
            inc = inc + 1;
        end
    end
end
mean_dist = mean(dist_array)/761
stdv_dist = std(dist_array)/761

dropArea = allBlobAreas(allBlobAreas > 200000);
dropDiam = (dropArea./761./761./pi).^(1/2)*2; %calculating the droplet diameter from
the calibrated pixel to mm value
mean_area = mean(dropArea)/761/761
stdv_area = std(dropArea)/761/761
mean_diam = (mean_area/pi)^(1/2)*2
stdv_diam = (stdv_area/pi)^(1/2)*2
figure; hist(dropDiam,5); xlabel("Droplet diameter, mm"); ylabel("Count");

```

## **APPENDIX E. DESCRIPTION OF THE MATLAB CODE USED TO MEASURE THE AREA OF A PRINTED THERMITE SAMPLE**

The average width of the printed samples was extracted from images of the single-line samples taken on an optical profilometer (Zeta Instruments 3D Optical Profiler) using the image analysis toolbox in MATLAB. This code—like the code of Appendix C—is based on the tutorial of Brett Shoelson for image segmentation [20]. This method is implemented in MATLAB through the following scripts:

- ‘MAIN\_area.m’ – the main code reads in a microscope image and measures the printed sample area. This script uses a hard-coded calibration value to convert a number of pixels to physical distance. This code is provided in Appendix F.
- ‘createMask.m’ – a function which converts the original red-green-black image into a binary image for later processing. This code is provided in Appendix G.

The microscope image of a printed thermite sample is read into MATLAB using the built-in function ‘imread’. The original red-green-blue (RGB) image is binarized using a threshold for V in the HSV (hue, saturation, value). This thresholding segments the image into regions with pure-substrate and regions printed thermite. Holes within connected regions of the segmented image are filled using the built-in MATLAB function ‘imfill’. The size of the printed droplet region is then measured using the built-in MATLAB function ‘regionprops’. The edges of each droplet are overlaid on the original RGB image of the droplet grid so that the user can verify that the image was processed correctly. A post-processed image showing the region of printed thermite highlighted in green is shown in Figure E1.

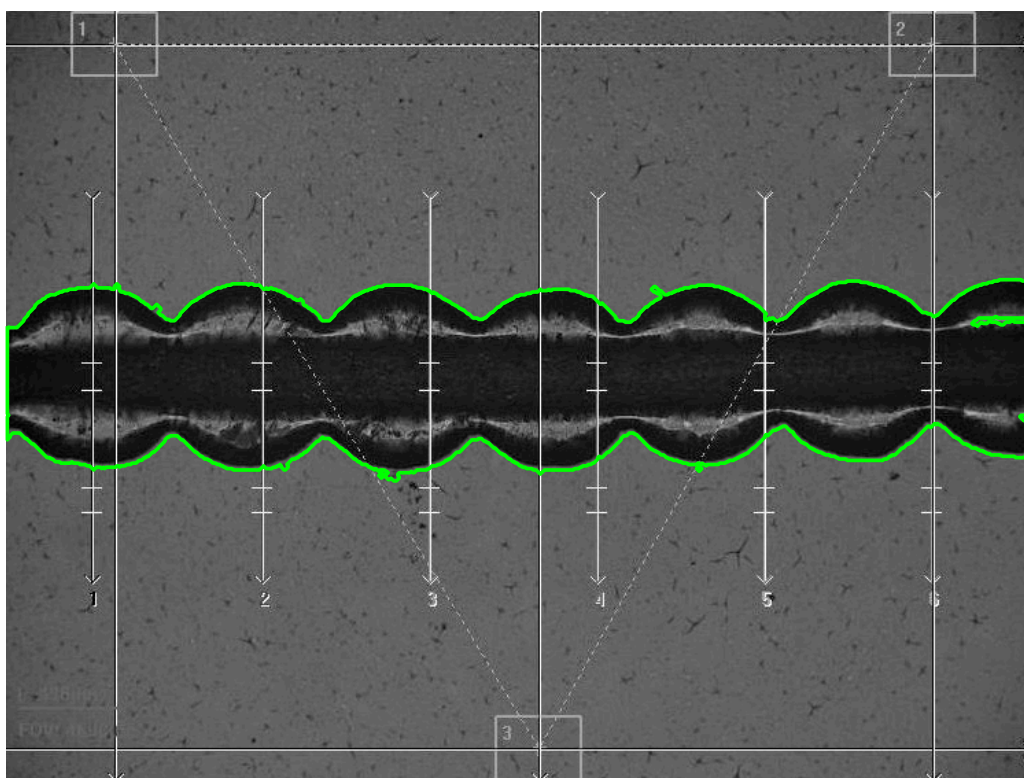


Figure E1. Image of the printed thermite sample post-processed in MATLAB to determine the area of printed thermite.

## APPENDIX F. MAIN MATLAB CODE TO MEASURE THE AREA OF A PRINTED THERMITE SAMPLE

Filename: MAIN\_area.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Created: 11/24/18
%Author: Forrest Son
%Purpose: measure the area of a printed thermite sample
%Requires: the Image Processing Toolbox (IPT)
% based on a tutorial found at
% https://blogs.mathworks.com/pick/2009/11/06/segmenting-coinsa-tutorial-on-blob-
analysis/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Startup code.
tic; % Start timer.
close all; clearvars;
imtool close all; % Close all imtool figures.
format long g;
format compact;
captionFontSize = 14;

% Read in image flie
[file,path] = uigetfile('E:\Profilometer - Cropped Images 2018-11-24\*.jpg','Select an
Image File');
fullFileName = fullfile(path, file);
originalImage = imread(fullFileName);

% convert to grayscale image
[rows, columns, numberOfColorChannels] = size(originalImage);
if numberOfColorChannels > 1
    originalImageRGB = originalImage;
    originalImage = rgb2gray(originalImage);
end

% Display the image.
subplot(2, 2, 1);
imshow(originalImageRGB);
% Maximize the figure window.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
drawnow;
caption = sprintf('Original image \nshowing centroids');
title(caption, 'FontSize', captionFontSize);
axis image;
% display image histogram.
[pixelCount, grayLevels] = imhist(originalImage);
subplot(2, 2, 2);
bar(pixelCount);
title('Histogram of original image', 'FontSize', captionFontSize);
xlim([0 grayLevels(end)]);
grid on;

%% Threshold the image to get a binary image
[binaryImage,maskedRGBImage] = createMask(originalImageRGB);
se = strel('disk',2);

```

```

    C = imdilate(binaryImage,se);
    D = bwareaopen(C,10000);
% Do a "hole fill" to get rid of any background pixels or "holes" inside the blobs.
    binaryImage = imfill(D, 'holes');

%% Identify individual blobs by seeing which pixels are connected to each other.
labeledImage = bwlabel(binaryImage, 8);    % Label each blob so we can make
measurements of it

CAL = 78/0.496;

%% Get all the blob properties.
% Can only pass in originalImage in version R2008a and later.
blobMeasurements = regionprops(labeledImage, originalImage, 'all');
numberOfBlobs = size(blobMeasurements, 1);

% bwboundaries() returns a cell array, where each cell contains the row/column
coordinates for an object in the image.
% Plot the borders of all the blobs on the original grayscale image using the
coordinates returned by bwboundaries.
subplot(2, 2, 4);
imshow(originalImageRGB);
title('Outlines', 'FontSize', captionFontSize);
axis image; % Make sure image is not artificially stretched because of screen's aspect
ratio.
hold on;
boundaries = bwboundaries(binaryImage);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;

% Get centroids.
% We can get the centroids of ALL the blobs into 2 arrays,
% one for the centroid x values and one for the centroid y values.
allBlobCentroids = [blobMeasurements.Centroid];
centroidsX = allBlobCentroids(1:2:end-1);
centroidsY = allBlobCentroids(2:2:end);
allBlobAreas = [blobMeasurements.Area];

%% rainbow plot
% Let's assign each blob a different color to visually show the user the distinct
blobs.
coloredLabels = label2rgb (labeledImage, 'hsv', 'k', 'shuffle'); % pseudo random color
labels
% coloredLabels is an RGB image. We could have applied a colormap instead (but only
with R2014b and later)
subplot(2, 2, 3);
imshow(coloredLabels);
axis image; % Make sure image is not artificially stretched because of screen's aspect
ratio.
caption = sprintf('Pseudo colored labels, from label2rgb().\nBlobs are numbered from
top to bottom, then from left to right. ');
title(caption, 'FontSize', captionFontSize);

%% print blob properties to command window
textFontSize = 9;    % Used to control size of "blob number" labels put atop the image.
labelShiftX = -300;  % Used to align the labels in the centers of the coins.
labelShiftY = 750;  % Used to align the labels in the centers of the coins.

```

```

blobECD = zeros(1, numberOfBlobs);
% Print header line in the command window.
fprintf(1, 'Blob #      Area    Perimeter    Length    Area/Length\n');
% Loop over all blobs printing their measurements to the command window.
for k = 1 : numberOfBlobs           % Loop through all blobs.
    % Find the mean of each blob. (R2008a has a better way where you can pass the
    original image
    % directly into regionprops. The way below works for all versions including
    earlier versions.)
    thisBlobsPixels = blobMeasurements(k).PixelIdxList; % Get list of pixels in
    current blob.
    meanGL = mean(originalImage(thisBlobsPixels)); % Find mean intensity (in original
    image!)
    meanGL2008a = blobMeasurements(k).MeanIntensity; % Mean again, but only for
    version >= R2008a

    blobArea = blobMeasurements(k).Area/CAL^2;           % Get area.
    blobPerimeter = blobMeasurements(k).Perimeter/CAL;   % Get perimeter.
    blobCentroid = blobMeasurements(k).Centroid/CAL;     % Get centroid one at a
time
    blobECD(k) = sqrt(4 * blobArea / pi)/CAL^2;           % Compute ECD -
Equivalent Circular Diameter.
    fprintf(1, '%2d %11.3f %11.2f %8.3f %10.6f %8.1f % 8.1f\n', k, blobArea,
    blobPerimeter, size(binaryImage,2)/CAL, blobArea/(size(binaryImage,2)/CAL));
    % Put the "blob number" labels on the "boundaries" grayscale image.
    text(blobCentroid(1) + labelShiftX, blobCentroid(2)+ labelShiftY, num2str(k),
    'FontSize', textFontSize, 'Color', 'k', 'FontWeight', 'Bold');
end

%% generate large image with edges marked
figure;
imshow(originalImage); hold on;
drawnow;
boundaries = bwboundaries(binaryImage);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;

```

## APPENDIX G. MATLAB FUNCTION TO BINARIZE AN IMAGE OF PRINTED THERMITE

Filename: createMask.m

```
function [BW,maskedRGBImage] = createMask(RGB)
%createMask Threshold RGB image using auto-generated code from colorThresholder app.
% [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
% auto-generated code from the colorThresholder App. The colorspace and
% minimum/maximum values for each channel of the colorspace were set in the
% App and result in a binary mask BW and a composite image maskedRGBImage,
% which shows the original RGB image values under the mask BW.

% Auto-generated by colorThresholder app on 24-Nov-2018
%-----

% Convert RGB image to chosen color space
I = rgb2hsv(RGB);

% Define thresholds for channel 1 based on histogram settings
channel1Min = 0.000;
channel1Max = 1.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 0.000;
channel2Max = 0.687;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 0.000;
channel3Max = 0.231;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
BW = sliderBW;

% Initialize masked output image based on the input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end
```

## APPENDIX H. A DESCRIPTION OF THE MATLAB CODE USED TO MEASURE BURNING RATE

There are two methods for tracking the flame propagation of the printed thermite samples. The first method involves tracking the highest intensity point over time; as described by Murray *et al.* [11] The second method was developed as part of this thesis and is described at the end of this appendix.

The algorithm used in the first method was tried as part of this thesis. It failed to capture the burning front location of samples tested in this thesis as illustrated in Figure H2 and Figure H3. Notice how this method fails even to capture the direction of propagation. This failure is most likely due to the different video collection techniques used in this method as compared to the method ultimately used in this thesis. Sample high-speed videos recorded as part of this thesis were captured with a top-down view in contrast to the side-view imaging technique used by Murray *et al.* [11] Therefore, a new algorithm was developed as part of this thesis work to measure the burning rate. This algorithm tracks the burning front by use of a segmentation tool to identify the region(s) where reactions are occurring and then tracking the movement of these regions.

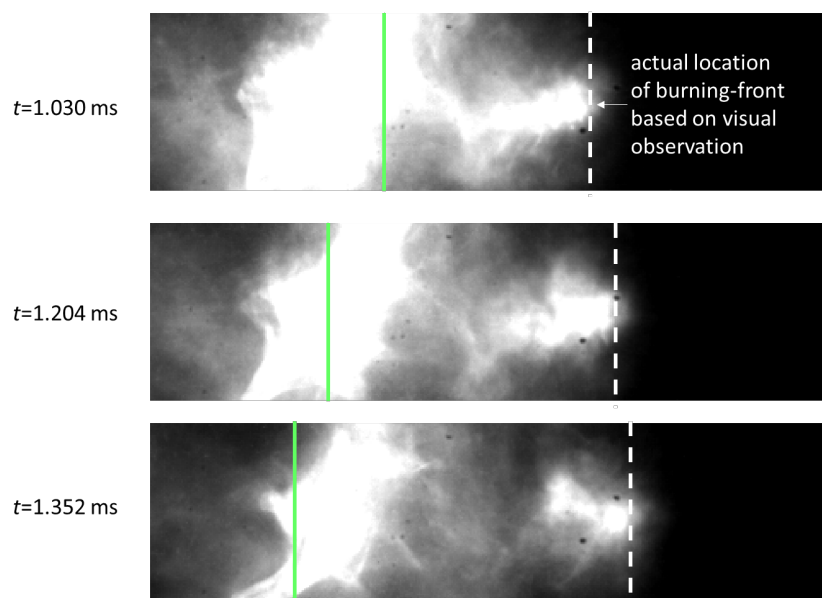


Figure H2. Still frames of a high-speed video of the thermite reaction post-processed in MATLAB by tracking the location of maximum intensity following the method of Murray *et al.* [11] The green line indicates the burning front as predicted by the tracking algorithm while the white dashed line indicates the actual location of the burning front based on visual observation.

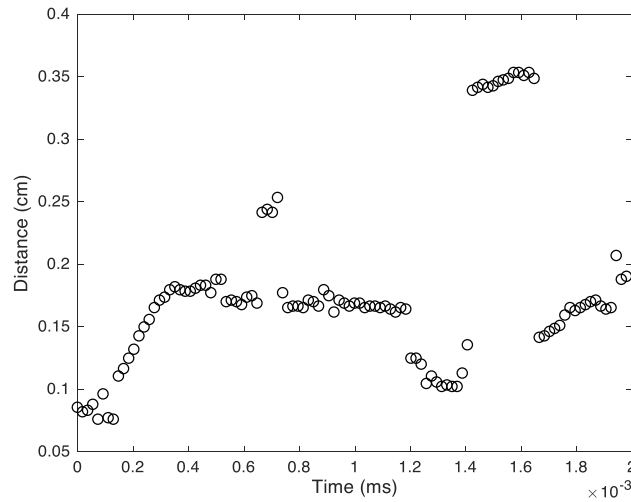


Figure H3. Plot of burning front location versus time as measured by tracking the location of maximum intensity following the method of Murray *et al.* [11]

This method is implemented in MATLAB in the following scripts

- ‘MAIN\_BurnRateCalc\_SINGLE\_File.m’ & MAIN\_BurnRateCalc\_LoopThroughFiles.m – The main code opens a GUI for the user to select a video file to analyze then calls subfunctions to extract the burning rate. This code additionally takes a pixel-size calibration number as an input for equating pixels in the video with distance. Note that the calibration value is determined from a calibration image taken separately before each set of burning rate experiments. The code for this script is provided in Appendix F and Appendix G.
- ‘f\_burnRateFromCine.m’ – a function that examines the video to identify, track, and analyze the burning front as described below. The code for this script is provided in Appendix H.

First, the frame rate and the number of frames are extracted from the video file. The function then loops through the video analyzing each frame individually. MATLAB reads each frame in as a matrix,  $\mathbf{I}$ , where each matrix value represents the light intensity of a single pixel,  $\mathbf{I}_{i,j}$  for the pixel at location  $(i,j)$ . A user-specified threshold is used to identify regions of ‘bright’ pixels. A black-and-white image matrix,  $\mathbf{BW}$ , is generated based on the threshold value where all pixels with intensities less than the threshold are set to a value of zero and all of the pixels with intensities greater than the threshold are set to a value of one.

Connected components (*i.e.*, clusters of bright pixels) in the black-and-white image are identified using the built-in MATLAB function ‘bwconncomp’. The size and location of each cluster are then measured via the built-in MATLAB function ‘regionprops’. Clusters that are more than 10% smaller than the largest cluster are ignored to avoid accidentally tracking ejected particles or other spurious artifacts. The burning front is defined as the leading edge of the largest cluster (*i.e.*, the largest bright region), unless there is another cluster within 10% of the size of the largest region that is closer to the burn front of the previous time step, in which case that closer cluster is assumed to be the burning zone, and its leading edge is taken as the burning front. Figure H4 compares this burning front algorithm with a simpler algorithm that tracks only the leading edge of the largest cluster to illustrate the need for tracking both the sizes of clusters and their proximity to the burning front from the previous time step. The algorithm used for the left-most images assumes that the burning front always coincides with the leading edge of the largest cluster of hot gas. In contrast, the algorithm used for the right images (*i.e.*, the algorithm used for all analysis in this thesis) compares each cluster’s location with the burning front location from the previous time-step and then tracks the cluster that is closest to the prior burning front.

After the burning front is identified for all of the time steps, the initial and final steps are excluded from analysis to prevent biasing the results with data from the non-linear portion of the burn (*i.e.*, the ignition point burning). This can be done either manually or automatically. The ‘manual’ setting allows a user to manually select the first and last data point to be considered in the analysis. The ‘automatic’ setting first computes the derivative of position vs. time ( $dx/dt$ ) at each time step. The statistical mode of  $dx/dt$  is then used to determine the start and end time of the linear portion of the flame propagation. The start and end times of linear propagation are defined as the first and last time where the instantaneous derivative of position  $dx/dt$  exactly equals the mode.

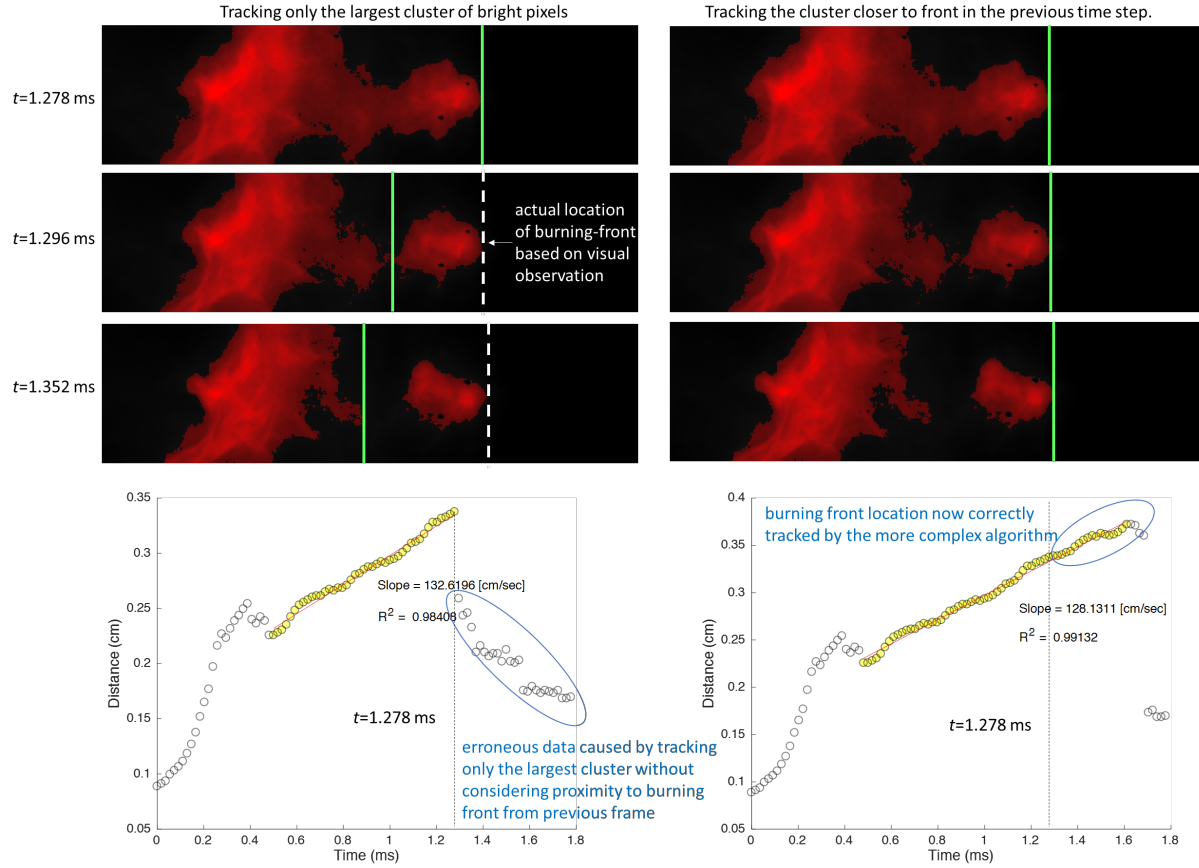


Figure H4. *Top panels*: Still frames of a high-speed video of the thermite reaction post-processed in MATLAB by two tracking methods. The red shaded region indicates where the pixel intensity is greater than a user-defined threshold value ( $I_{ij} > TH$ ) and the green line indicates the burning front as predicted by each tracking algorithm. *Bottom panels*: Plot of burning front location versus time as measured by the two algorithms.

## APPENDIX I. THE MAIN MATLAB CODE USED FOR MEASURING BURNING RATE FROM A SINGLE VIDEO

Filename: MAIN\_BurnRateCalc\_SINGLE\_File.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Created: 05/25/18
%Author: Forrest Son
%Purpose: Calculate the burning rate from a SINGLE high-speed video
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; clear; close all;

% ask user to select the cine files:
originalPath = cd;
[cineFileName,cineFilePath] = uigetfile('*..*','Select cine file.');
```

```

if(cineFileName~=0)
    cd(cineFilePath);
end
cd(originalPath);

% user-defined parameters for video analysis
skip = 1;          % number of frames to skip between data points
gamma = 1;         % value of gamma for the image display

BWthreshold = 4500;
% pixel-size calibration (This is currently a hard coded value)
r = 0.011904762;

baseFileName = erase(cineFileName, '.cine')
fullFileName = fullfile(cineFilePath, cineFileName);
fprintf(1, 'Now reading %s\n', fullFileName);
[burnRate, R2, skip] = f_burnRateFromCine_maxI( fullFileName, r,...

'BWthreshold',BWthreshold,'skip',1,'SelectionMode','manual'); %,'SelectionMode','manua
l'
fprintf('Burn Rate: %1.3f cm/sec \n',burnRate);
fprintf('R^2: %1.4f \n',R2);

```

## APPENDIX J. THE MAIN MATLAB CODE USED FOR MEASURING THE BURNING RATE FROM ALL OF THE VIDEOS IN A DIRECTORY

Filename: MAIN\_BurnRateCalc\_LoopThroughFiles.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Created: 05/25/18
%Author: Forrest Son
%Purpose: Calculate the burning rate from ALL high-speed videos in a
%         directory
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; clear;

% ask user to select directory with cine files:
cineFilePath = uigetdir('*.','Select directory with cine files.');
```

% directory where pictures will be saved:

```
saveFilePath = strcat(cineFilePath,'\Burning Rate\');
```

% automatically extract list of all cine files in the directory:

```
cineFiles = dir(fullfile(cineFilePath,'*.cine'));
```

% user-defined parameters for video analysis

```
skip = 1;           % number of frames to skip between data points
gamma = 1;          % value of gamma for the image display
```

% pixel-size calibration (This is currently a hard coded value)

```
r = 0.011627907;
BWthreshold = 8000;
% now loop through files and analyze
for k = 1:length(cineFiles)
    cineFileName = cineFiles(k).name;
    pixelPitch(k) = str2num(cineFileName(1:3));
    nSample(k) = str2num(cineFileName(end-6:end-5));
    fullFileName = fullfile(cineFilePath, cineFileName);
    baseFileName = erase(cineFileName, '.cine');
    fprintf(1, '_____ \n');
    fprintf(1, 'Now reading %s\n', fullFileName);
    [burnRate(k), R2(k), skip(k)] = f_burnRateFromCine( fullFileName, r,...
'BWthreshold',BWthreshold,'SelectionMode','auto'); %, 'SelectionMode','manual'
    fprintf('Burn Rate: %1.3f cm/sec \n',burnRate(k));
    fprintf('R^2: %1.4f \n',R2(k));
end

if length(pixelPitch)>length(burnRate)
    pixelPitch(end) = [];
    nSample(end) = [];
end

Results = table([pixelPitch.', nSample.', burnRate.', R2.',skip.]);

close all;
figure;
scatter (pixelPitch, burnRate)
```

## APPENDIX K. THE MATLAB SUBFUNCTION FOR MEASURING THE BURNING RATES

Filename: f\_burnRateFromCine.m

```
function [ burnRate R2 skip ] = f_burnRateFromCine( cineFile, calibration, varargin)
%EXTRACT THE BURN RATE FROM A CINE VIDEO AND SAVE A PNG OF THE BURN FRONT
%PROPOGATION. FORREST SON MAY 2018
% This function uses a thresholding value followed by a segmenting tool
% to find the largest bright spot(s). The largest blob is taken as the
% burn front unless there is a spot within 10% of the size of the largest
% blob that is closer to the burn front in the previous time step.
%
% After the burn front is identified for all time steps, the initial and
% final steps are excluded from analysis to prevent biasing the data
% during the non-linear portion of the burn. This is done in two steps.
% I use the mode of the derivative of position vs. time (dp/dt) to
% eliminate the start and end data. Then I remove all data that falls
% outside of a 10% error band from a linear fit of the remaining data.
close all;

%% extract details on the video from .cine file
info = cineHeader(cineFile);
firstImage = info.CINEFILEHEADER.FirstImageNo;
nFrames = info.CINEFILEHEADER.ImageCount;
frameRate = info.SETUP.FrameRate;
I = cineRead(strcat(cineFile),firstImage+floor(nFrames/2));

%% parse user inputs
% set default values for video analysis parameters & overwrite those
% defaults where the user passed in preferred values through varargin:
p = inputParser;

defaultSaveMode = 'auto';
validSaveMode = {'auto','manual','no'};
checkSaveMode = @(x) any(validatestring(x,validSaveMode));

defaultSelectionMode = 'auto';
validSelectionMode = {'auto','manual'};
checkSelectionMode = @(x) any(validatestring(x,validSelectionMode));

s = split(cineFile,'\');
defaultBaseFileName = erase(char(s(end)),'.cine');
defaultSaveFilePath = erase(cineFile,strcat(defaultBaseFileName,'.cine'));
defaultSaveFilePath = strcat(defaultSaveFilePath,'Burning Rate\');

defaultSkip = max(floor(nFrames / 500),1);
defaultGamma = 1;
defaultMaxIntensity = 163830;
defaultBWthreshold = 8000;
defaultLinearExclusionThreshold = 0.05;

% these are all of the optional parameters you can pass this function
addRequired(p,'cineFile',@ischar);
```

```

addRequired(p, 'calibration', @isnumeric);
addOptional(p, 'SaveMode', defaultSaveMode, checkSaveMode)
addOptional(p, 'SelectionMode', defaultSelectionMode, checkSelectionMode)
addParameter(p, 'baseFileName', defaultBaseFileName, @ischar)
addParameter(p, 'saveFilePath', defaultSaveFilePath, @ischar)
addParameter(p, 'skip', defaultSkip, @isnumeric)
addParameter(p, 'gamma', defaultGamma, @isnumeric)
addParameter(p, 'maxIntensity', defaultMaxIntensity, @isnumeric)
addParameter(p, 'BWthreshold', defaultBWthreshold, @isnumeric)
addParameter(p, 'LinearExclusionThreshold', defaultLinearExclusionThreshold, @isnumeric)

p.KeepUnmatched = true;
parse(p, cineFile, calibration, varargin{:})

r      = p.Results.calibration;
skip   = p.Results.skip;

% debugging code:
% if ~isempty(fieldnames(p.Unmatched))
%     disp('Extra inputs:')
%     disp(p.Unmatched)
% end
% if ~isempty(p.UsingDefaults)
%     disp('Using defaults: ')
%     disp(p.UsingDefaults)
% end

%% ~~~~~
% ANALYZE VIDEO
pos=struct;
counter=0;

fig_movie_frame = figure(1); % figure window for movie frame
movegui('east') % move figure to specific location
title('Click on Surface');
fig_dist_vs_time = figure(2); % plot of distance vs time
xlabel('Time (s)'); ylabel('Distance (cm)');
movegui('northwest');
set(gcf, 'color', 'w');
fig_rate_vs_time = figure(3); % plot of burn rate vs time
xlabel('Time (s)'); ylabel('Instantaneous Burn Rate (cm/s)');
movegui([0, 50]);
set(gcf, 'color', 'w');

for i = firstImage:skip:firstImage+nFrames-1
    I = cineRead(strcat(cineFile), i);
    rgb = I;
    max_I = max(max(I));
    %     avg_I = mean(mean(I));
    %     std_I = mean(std(I));
    TH      = zeros(size(I)); % only include the hot-stuff matrix
    TH(I>p.Results.BWthreshold) = I(I>p.Results.BWthreshold);
    BW      = zeros(size(I)); % only include the hot-stuff matrix
    BW(TH>0) = 1; % binary matrix
    cc      = bwconncomp(BW); % find blobs
    cc_props = regionprops(cc); % blob sizes and locations
    if cc.NumObjects > 0
        %     labeled      = labelmatrix(cc);
        sizes          = vertcat(cc_props.Area);
        SizBigBlob     = max(sizes);

```

```

if SizBigBlob < 25
%   fprintf('size : %1.3f \n',SizBigBlob)
    continue;
end
NumBigBlob = sizes == SizBigBlob;
BoxBigBlob = cc_props(NumBigBlob).BoundingBox;
LocBigBlob_x = BoxBigBlob(1)+BoxBigBlob(3);
LocBigBlob_y = BoxBigBlob(2)+BoxBigBlob(4)/2;
if counter < 1
    x = LocBigBlob_x;
    y = LocBigBlob_y;
end
dx = LocBigBlob_x - x;
dy = LocBigBlob_y - y;
distBigBlob = sqrt(dx^2+dy^2);
LocBestBlob_x = LocBigBlob_x;
LocBestBlob_y = LocBigBlob_y;
% see if there is a large blob that is closer to the burn front in
% the previous timestep
for n_blob = 1:cc.NumObjects
    if sizes(n_blob)/SizBigBlob < 0.1
        if sizes(n_blob) < 100; continue; end
    end
    BoxBlob = cc_props(n_blob).BoundingBox;
    LocBlob_x = BoxBlob(1)+BoxBlob(3);
    LocBlob_y = BoxBlob(2)+BoxBlob(4)/2;
    dx = LocBlob_x - x;
    dy = LocBlob_y - y;
    dist = sqrt(dx^2+dy^2);
    if dist < distBigBlob
%       disp('BETTER BLOB FOUND')
        NumBestBlob = n_blob;
        LocBestBlob_x = LocBlob_x;
        LocBestBlob_y = LocBlob_y;
    end
end
x = LocBestBlob_x;
y = LocBestBlob_y;
pause(0.001);
counter=counter+1;
blobSize(counter) = sizes(n_blob);
pos.x(counter)=x;
pos.y(counter)=y; %pixels
pos.t(counter)=(i-firstImage)/frameRate; %time in seconds
end
% ~~~~~~ Show the user what is going on ~~~~~~
if counter>=1
    % ~~~~~~ SHOW IMAGE / BURN FRONT ~~~~~~
    figure(1); % set current figure handle
    imshow(rgb,[0 p.Results.maxIntensity]*0.5,'InitialMagnification',250)
    red = cat(3, ones(size(I)), zeros(size(I)), zeros(size(I)));
    hold on
    h = imshow(red); % red filter
    set(h, 'AlphaData', TH/max_I) % set filter alpha based on threshold
    figure(1); line([x,x],[0,104],'Color','g') % green line @ the burn front
    hold off
    % images for debugging purposes only
    figure(4); % set current figure handle
    imshow(rgb,[0 p.Results.maxIntensity]*0.5,'InitialMagnification',250)
    figure(4); % set current figure handle
    RGB_label = label2rgb(labeled, @copper, 'c', 'shuffle');
    imshow(RGB_label,'InitialMagnification','fit')
    % ~~~~~~ PLOT POSITION ~~~~~~

```

```

        figure(2);
        plot(pos.t,pos.x.*r./10,'ok');
    end
    if counter>=2
        % ~~~~~~ CALCULATE BURN RATE ~~~~~~
        pos.avg_t(counter-1)=mean([pos.t(counter),pos.t(counter-1)]);
        pos.br(counter-1)=(pos.x(counter)-pos.x(counter-1))*r/10/...
            (pos.t(counter)-pos.t(counter-1));
        % ~~~~~~ PLOT BURN RATE ~~~~~~
        figure(3);
        plot(pos.avg_t,pos.br,'ok');
    end
end
close(figure(1));
%% ~~~~~~ POST PROCESS DATA ~~~~~~
x      = pos.t;
y      = pos.x*r./10;

% SELECT THE STARTING AND ENDING DATA POINTS
switch p.Results.SelectionMode
    case 'manual'
        get(fig_dist_vs_time, 'Renderer');
        disp('Click the first point you want to consider, press ENTER when done')
        [s_x, s_y] = getpts(fig_dist_vs_time,1);
        [c_index_s] = min(abs(pos.t-s_x));
        disp('Click the last point you want to consider, press ENTER when done')
        [e_x, e_y] = getpts(fig_dist_vs_time,1);0
        [c_index_e] = min(abs(pos.t-e_x));
    case 'auto'
        clear diff
        dydx = diff([eps; y(:)])./diff([eps; x(:)]);
        % figure;
        % scatter(x(dydx>0),dydx(dydx>0));
        % set(gca,'YScale','log')
        cutOff = mode(dydx(dydx>0));
        cutOffLoc = find(dydx==cutOff);
        index_s = min(cutOffLoc);
        index_e = max(cutOffLoc);
end

% EXCLUDE BAD DATA @ START AND END
time      = pos.t(index_s:index_e);
position  = pos.x(index_s:index_e).*r./10;
x         = time;
y         = position;

% REMOVE OUTLIERS BASED ON PROXIMITY TO INITIAL LINEAR FIT
c         = polyfit(x,y,1);
y_pred   = x*c(1) + c(2);
DIFF      = abs((y-y_pred))./y_pred;
time_clean = x(DIFF<p.Results.LinearExclusionThreshold);
pos_clean  = y(DIFF<p.Results.LinearExclusionThreshold);

% figure; hold on;
% scatter(time, position);
% scatter(time_clean, pos_clean, 'r');

% CALCULATE THE BURN RATE BASED ON SLOPE
[c,~] = polyfit(time_clean,pos_clean,1);
[~,~,~] = polyfit(time_clean,pos_clean,1);
burnRate = c(1); %burn rate in [cm/s]

```

```

fitPosition = polyval(c,time_clean);
mid = floor(numel(pos_clean)/2);
xshift = (max(time_clean)-min(time_clean))/10;
yshift = (max(pos_clean)-min(pos_clean))/10;

%% PLOT DATA ALONG WITH FIT
fig_fitting_data = figure(4);
hold on
%
scatter(time_clean(1:2:end)*1000,pos_clean(1:2:end),'ok','MarkerEdgeAlpha',.5,'LineWid
th',0.25);
scatter(time_clean*1000,pos_clean,'oy','filled','MarkerFaceAlpha',1/2);
scatter(pos.t*1000,pos.x*r./10,'ok','MarkerEdgeAlpha',1/2,'LineWidth',0.25);
plot(time_clean*1000,fitPosition,'r');
text((time(mid)+xshift)*1000,position(mid)-yshift,['Slope = ',num2str(burnRate), '
[cm/sec]']);
xlabel('Time (ms)');
ylabel('Distance (cm)');
% fprintf('\nBurn Rate: %1.3f cm/sec \n',burnRate);

%R^2 value
y_mean = mean(pos_clean);
SS_res = sum((pos_clean-fitPosition).^2);
SS_tot = sum((pos_clean-y_mean).^2);
R2 = 1-(SS_res/SS_tot);
% fprintf('R^2: %1.4f \n',R2);
text((time_clean(mid)+xshift)*1000,0.9*(pos_clean(mid)-yshift),['R^2 =
',num2str(R2)]);

% ylim([0 0.25]);
switch p.Results.SaveMode
case 'auto'
    filename = strcat(p.Results.saveFilePath,p.Results.baseFileName,'.png');
    saveas(fig_fitting_data,filename,'png')
case 'manual'
    doSave = questdlg('Would you like to save the data?');
    if strcmp(doSave,'Yes')
        saveName = char(inputdlg('Save file as:'));
        saveLocation = uigetdir('*.','Select save location (a new folder will be
created in the selected location).');
        saveFolder = strcat(saveName,' Data');
        cd(saveLocation); mkdir(saveFolder);
    cd(fullfile(saveLocation,saveFolder));
        save(strcat(saveName,' Data'),'pos','r_b','Rsqu','skip','r');
        saveas(figure(2),strcat(saveName,' Selected Positions'),'pdf');
        saveas(figure(3),strcat(saveName,' Instantaneous Burn Rates'),'pdf');
        saveas(figure(4),strcat(saveName,' Final Burn Rate Plot'),'pdf');
        saveas(figure(4),strcat(saveName,' Final Burn Rate Plot'),'fig');
    end
    cd(originalPath); % Goes back to original folder
case 'no'
end
end
end

```

## APPENDIX L. EES CODE USED FOR THE THERMAL MODEL

```

k      = 0.2 [W/m*K]
c_p    = 310 [ J/kg*K]
rho    = 3262 [kg/m^3]
alpha  = k/(rho*c_p)

q      = 2117522.4 [J/kg]
width  = 0.0009 [m]
thickness = 0.000083 [m]

A_c = thickness * width

R_u = R#
T_i = 749.4 [K]
T_0 = 298.15 [K]

c = 65232 [kg/s*m^2]
E = 74037 [J/mol]

h_side = 20000000000 [ W/m^2*K]
h_top  = 20000000000 [ W/m^2*K]

m      = c*exp(-E/(2*R#*T_f))
S_L    = m / rho

delta  = alpha / S_L * (R#*T_f/E)

q_gen  = m*q*A_c

q_cond = k*A_c *(T_f-T_i)/delta

q_loss_top = h_top*(width*delta)*(T_f-T_0)

q_loss_side = h_side*(2*thickness*delta)*(T_f-T_0)

q_gen = q_cond+ q_loss_side + q_loss_top

```

## REFERENCES

- [1] D. Sundaram, V. Yang, and R. A. Yetter, "Metal-based nanoenergetic materials: Synthesis, properties, and applications," *Prog. Energy Combust. Sci.*, vol. 61, pp. 293–365, Jul. 2017.
- [2] D. S. Sundaram, P. Puri, and V. Yang, "Pyrophoricity of nascent and passivated aluminum particles at nano-scales," *Combust. Flame*, vol. 160, no. 9, pp. 1870–1875, Sep. 2013.
- [3] J. Y. Malchi, T. J. Foley, and R. A. Yetter, "Electrostatically Self-Assembled Nanocomposite Reactive Microspheres," *ACS Appl. Mater. Interfaces*, vol. 1, no. 11, pp. 2420–2423, Nov. 2009.
- [4] P. Chakraborty and M. R. Zachariah, "Do nanoenergetic particles remain nano-sized during combustion?," *Combust. Flame*, vol. 161, no. 5, pp. 1408–1416, May 2014.
- [5] T. Hawa and M. R. Zachariah, "Development of a phenomenological scaling law for fractal aggregate sintering from molecular dynamics simulation," *J. Aerosol Sci.*, vol. 38, no. 8, pp. 793–806, 2007.
- [6] B. Derby, "Inkjet printing ceramics: From drops to solid," *J. Eur. Ceram. Soc.*, vol. 31, no. 14, pp. 2543–2550, Nov. 2011.
- [7] J. Perelaer, P. J. Smith, D. Mager, D. Soltman, S. K. Volkman, V. Subramanian, J. G. Korvink, and U. S. Schubert, "Printed electronics: the challenges involved in printing devices, interconnects, and contacts based on inorganic materials," *J. Mater. Chem.*, vol. 20, no. 39, pp. 8446–8453, 2010.
- [8] M. Singh, H. M. Haverinen, P. Dhagat, and G. E. Jabbour, "Inkjet Printing—Process and Its Applications," *Adv. Mater.*, vol. 22, no. 6, pp. 673–685, Feb. 2010.
- [9] B.-J. de Gans, P. C. Duineveld, and U. S. Schubert, "Inkjet Printing of Polymers: State of the Art and Future Developments," *Adv. Mater.*, vol. 16, no. 3, pp. 203–213, Feb. 2004.
- [10] F. G. Zaugg and P. Wagner, "Drop-on-Demand Printing of Protein Biochip Arrays," *MRS Bull.*, vol. 28, no. 11, pp. 837–842, Nov. 2003.
- [11] A. K. Murray, W. A. Novotny, T. J. Fleck, I. E. Gunduz, S. F. Son, G. T.-C. Chiu, and J. F. Rhoads, "Selectively-deposited energetic materials: A feasibility study of the piezoelectric inkjet printing of nanothermites," *Addit. Manuf.*, vol. 22, pp. 69–74, Aug. 2018.
- [12] E. R. Westphal, A. K. Murray, M. P. McConnell, Trevor J. Fleck, G. T.-C. Chiu, J. F. Rhoads, I. E. Gunduz, and S. F. Son, "The Effects of Confinement on the Destructive Performance of Printed Nanothermites," *Propellants Explos. Pyrotech.*, In Review.
- [13] Alexander S. Tappan, J. Patrick Ball, and James W. Colovos, "Inkjet Printing of Energetic Materials," presented at the Materials Research Society Fall Meeting, Boston, MA, 28-Dec-2011.

- [14] S. R. Turns, *An Introduction to Combustion: Concepts and Applications*, 3 edition. New York: McGraw-Hill Education, 2011.
- [15] D. Wei, R. Dave, and R. Pfeffer, "Mixing and Characterization of Nanosized Powders: An Assessment of Different Techniques," *J. Nanoparticle Res.*, vol. 4, no. 1–2, pp. 21–41, Apr. 2002.
- [16] J. Perelaer, P. J. Smith, E. van den Bosch, S. S. C. van Grootel, P. H. J. M. Ketelaars, and U. S. Schubert, "The Spreading of Inkjet-Printed Droplets with Varying Polymer Molar Mass on a Dry Solid Substrate," *Macromol. Chem. Phys.*, vol. 210, no. 6, pp. 495–502, Mar. 2009.
- [17] A. S. Tappan, J. Cesarano, III, and J. N. Stuecker, "Nanocomposite thermite ink," US8048242B1, 05-Apr-2007.
- [18] L. L. Wang, Z. A. Munir, and Y. M. Maximov, "Thermite reactions: their utilization in the synthesis and processing of materials," *J. Mater. Sci.*, vol. 28, no. 14, pp. 3693–3708, Jan. 1993.
- [19] T. J. Fleck, "Additive Manufacturing of Energetic Materials and Its Uses in Various Applications," M.S.M.E., Purdue University, United States -- Indiana, 2017.
- [20] "Segmenting Coins...a Tutorial on Blob Analysis," *MATLAB File Exchange Pick of the Week*, <https://blogs.mathworks.com/pick/2009/11/06/segmenting-coinsa-tutorial-on-blob-analysis/>
- [21] V. E. Sanders, B. W. Asay, T. J. Foley, B. C. Tappan, A. N. Pacheco, and S. F. Son, "Reaction Propagation of Four Nanoscale Energetic Composites (Al/MoO<sub>3</sub>, Al/WO<sub>3</sub>, Al/CuO, and Bi<sub>2</sub>O<sub>3</sub>)," *J. Propuls. Power*, vol. 23, no. 4, pp. 707–714, 2007.
- [22] R. R. Nellums, B. C. Terry, B. C. Tappan, S. F. Son, and L. J. Groven, "Effect of Solids Loading on Resonant Mixed Al-Bi<sub>2</sub>O<sub>3</sub> Nanothermite Powders," *Propellants Explos. Pyrotech.*, vol. 38, no. 5, pp. 605–610, Oct. 2013.
- [23] R. Ramachandran, V. S. Vuppuluri, T. J. Fleck, J. F. Rhoads, I. E. Gunduz, and S. F. Son, "Influence of Stoichiometry on the Thrust and Heat Deposition of On-Chip Nanothermites," *Propellants Explos. Pyrotech.*, vol. 43, no. 3, pp. 258–266, Mar. 2018.
- [24] S.F. Son, B. W. Asay, and G. A. Risha, "Combustion of Nanoscale Al/MoO<sub>3</sub> Thermite in Microchannels," *J. Propuls. Power*, vol. 23, no. 4, pp. 715–721, Jul. 2007.
- [25] T. Leach, "Effect of Structural Heat Conduction on the Performance of Micro-Combustors and Micro-Thrusters," University of Maryland, 2005.
- [26] T. T. Leach, C. P. Cadou, and G. S. Jackson, "Effect of structural conduction and heat loss on combustion in micro-channels," *Combust. Theory Model.*, vol. 10, no. 1, pp. 85–103, Feb. 2006.
- [27] S. Ducruix, T. Schuller, D. Durox, bastien Candel, and S. Candel, "Combustion Dynamics and Instabilities: Elementary Coupling and Driving Mechanisms," *J. Propuls. Power*, vol. 19, no. 5, pp. 722–734, 2003.

- [28] H. Wang and M. Frenklach, "Detailed reduction of reaction mechanisms for flame modeling," *Combust. Flame*, vol. 87, no. 3, pp. 365–370, Dec. 1991.
- [29] J. Li, S. K. Chou, Z. Li, and W. Yang, "Development of 1D model for the analysis of heat transport in cylindrical micro combustors," *Appl. Therm. Eng.*, vol. 29, no. 8, pp. 1854–1863, Jun. 2009.
- [30] A. Varma, A. S. Mukasyan, and S. Hwang, "Dynamics of self-propagating reactions in heterogeneous media: experiments and model," *Chem. Eng. Sci.*, vol. 56, no. 4, pp. 1459–1466, Feb. 2001.
- [31] R. S. Benson, W. J. D. Annand, and P. C. Baruah, "A simulation model including intake and exhaust systems for a single cylinder four-stroke cycle spark ignition engine," *Int. J. Mech. Sci.*, vol. 17, no. 2, pp. 97–124, Feb. 1975.
- [32] S. F. Son and M. Q. Brewster, "Radiation-Augmented Combustion of Homogeneous Solids," *Combust. Sci. Technol.*, vol. 107, pp. 127–154, 1995.
- [33] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2011.
- [34] S. Torquato, *Random Heterogeneous Materials - Microstructure and Macroscopic Properties*. Springer, 2005.
- [35] M. J. Mezger, K. J. Tindle, M. Pantoya, L. J. Groven, and D. Kalyon, *Energetic Materials: Advanced Processing Technologies for Next-Generation Materials*. CRC Press, 2017.
- [36] R. Singh, "Predictions of Effective Thermal Conductivity of Complex Materials," in *Heat Transfer in Multi-Phase Materials*, 2011, pp. 237–273.
- [37] N. W. Piekiet, L. Zhou, K. T. Sullivan, S. Chowdhury, G. C. Egan, and M. R. Zachariah, "Initiation and Reaction in Al/Bi<sub>2</sub>O<sub>3</sub> Nanothermites: Evidence for the Predominance of Condensed Phase Chemistry," *Combust. Sci. Technol.*, vol. 186, no. 9, pp. 1209–1224, Sep. 2014.
- [38] J. A. Puszyński, C. J. Bulian, and J. J. Swiatkiewicz, "Processing and Ignition Characteristics of Aluminum-Bismuth Trioxide Nanothermite System," *J. Propuls. Power*, vol. 23, no. 4, pp. 698–706, Jul. 2007.
- [39] S. Fischer and M. Grubelich, "A survey of combustible metals, thermites, and intermetallics for pyrotechnic applications," in *32nd Joint Propulsion Conference and Exhibit*, 0 vols., American Institute of Aeronautics and Astronautics, 1996.
- [40] C. Altavilla and E. Ciliberto, *Inorganic Nanoparticles: Synthesis, Applications, and Perspectives*. CRC Press, 2016.
- [41] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 690–698, May 1987.