Body Armor Shape Sensing with Fiber Optic Sensors

Frederick Alexander Seng
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Electrical and Computer Engineering Commons

BYU ScholarsArchive Citation
Seng, Frederick Alexander, "Body Armor Shape Sensing with Fiber Optic Sensors" (2018). All Theses and Dissertations. 6950.
https://scholarsarchive.byu.edu/etd/6950

This Dissertation is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Body Armor Shape Sensing with Fiber Optic Sensors

Frederick Alexander Seng

A dissertation submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Stephen M. Schultz, Chair
Aaron Hawkins
Greg Nordin
Daniel Smalley
Brian Jensen

Department of Electrical and Computer Engineering
Brigham Young University

Copyright © 2018 Frederick Alexander Seng
All Rights Reserved
ABSTRACT

Body Armor Shape Sensing with Fiber Optic Sensors

Frederick Alexander Seng
Department of Electrical and Computer Engineering, BYU
Doctor of Philosophy

In this dissertation, the rate of the BFD during body armor impact is characterized with fiber Bragg gratings for the first time ever. The depth rate is characterized using a single fiber optic sensor, while the entire shape rate can be characterized using multiple fiber optic sensors. This is done with a final depth accuracy of less than 10% and a timing accuracy of 15% for BFDs as deep as 50 mm and impact event of less than 1 millisecond.

The shape sensing method introduced in this dissertation is different from traditional fiber optic sensor shape reconstruction methods in the fact that strain from the kinetic friction regime is used rather than the static friction regime. In other words, information from the fiber optic sensors slipping is used to reconstruct the shape in this work, whereas strain from the fiber optic sensor remaining fixed to a reference is used for typical fiber optic shape sensing purposes.

Keywords: fiber optic sensors, body armor testing, fiber Bragg gratings
ACKNOWLEDGEMENTS

I would like to thank North Carolina State University for their collaboration on this project. Drew, Tyler, Alex, Dr. Pankow and Dr. Peters, you’ve all been great to work with. I’ve definitely learned a lot from all the meetings that we’ve had. Thank you for performing all the ballistics testing for this contract, and all the feedback you’ve given me during the meetings we’ve had over years.

I would also like to thank the Army Test Center in Aberdeen, Maryland for letting us test at their facility. Those tests were invaluable for our project because it allowed us to calibrate our efforts against current test methods and standards. I would also like to thank the Test Resource Management Center and the Test and Evaluation Science and Technology Group for their funding into this project.

Thank you to my mentors Nikola Stan and Spencer Chadderdon for always encouraging me to try to improve a little more. Thank you to Nikola Stan, LeGrand Shumway, Rex King, and Alec Hammond for not only being peers, but reliable friends that I could count on. Thank you to Alexander Petrie, Chad Josephson, Kevin Cuzner, Jessica Hatton, Ivann Velasco, and Helaman Johnston for all your help with my projects.

Thank you to Dr. Selfridge for your encouragement and collaboration on research. Thank you to Dr. Jensen for always being willing to brainstorm on mechanical engineering concepts, and thanks to the rest of my committee: Dr. Smalley, Dr. Nordin, and Dr. Hawkins for all the awesome classes and support during the years. Thank you to Dr. Wilde for helping make Electrical Engineering the most enjoyable coursework I’ve ever had in my life. And thank you to all the professors I had at BYU for providing an unforgettable education experience.
Thank you Kris Bess for being an awesome graduate program manager that the students can also call a really good friend. Thank you Janalyn Mergist for being a great undergraduate advisor. And thank you Mark Ormsby for taking care of my paychecks over the years.

Thank you Dr. Schultz for being an excellent mentor and friend. My graduate school experience was unforgettable, and could not have been possible without your help and support. Thanks for all the brainstorming sessions and guidance over the years; working with you has been invaluable in terms of my professional and personal development.

Thank you to all my family: Mom, Dad, thanks for paying for my mortgage and always believing in me and supporting me through anything. Thank you to my in-laws for all your help and support.

Finally, thank you Wendy, we made it.
TABLE OF CONTENTS

1 Introduction ............................................................................................................................. 1
  1.1 Body Armor Testing .......................................................................................................... 1
  1.2 Contributions and Dissertation Outline ............................................................................ 6
2 Background ............................................................................................................................ 10
  2.1 Basic Body Armor Dynamics ........................................................................................ 10
  2.2 Fiber Optic Sensors .......................................................................................................... 16
  2.3 Shape Sensing with Fiber Optic Sensors ....................................................................... 17
3 High Repetition-Rate Full-Spectrum Interrogation of Fiber Bragg Gratings ........................ 21
  3.1 High Speed Interrogation for Fiber Bragg Gratings ....................................................... 21
    3.1.1 Current Technology ................................................................................................ 22
    3.1.2 High-Speed Full-Spectrum Interrogation for Fiber Bragg Gratings ....................... 24
  3.2 High Repetition-Rate Measurement Using a Hopkinson Bar ........................................ 26
    3.2.1 Measurement Setup ................................................................................................. 32
  3.3 Measurement Results ..................................................................................................... 34
  3.4 Strain Calculation ........................................................................................................... 41
  3.5 Summary ........................................................................................................................ 43
4 Survivability of Fiber Bragg Gratings Under Impact Conditions ......................................... 45
  4.1 Experimental Setup ........................................................................................................ 45
  4.2 Fiber Survivability Under Impact Events....................................................................... 48
    4.2.1 Low Velocity Coating Qualification ....................................................................... 52
    4.2.2 Offset Specimens .................................................................................................... 53
    4.2.3 Directly Impacted Specimens ................................................................................. 60
  4.3 Survivability at Higher Dynamics .................................................................................. 64
    4.3.1 Silicone Sensing Layer ........................................................................................... 64
  4.4 Summary ........................................................................................................................ 68
5 Strain to Back face Deformation Algorithm for Fiber Bragg Gratings .................................. 71
  5.1 Kinetic Friction Strain to BFD Algorithm Overview ........................................................ 71
    5.1.1 Single Grating Strain to BFD Calculation Algorithm ............................................. 71
    5.1.2 Results ..................................................................................................................... 75
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clay Backing</td>
<td>76</td>
</tr>
<tr>
<td>Ballistics Gel Testing</td>
<td>79</td>
</tr>
<tr>
<td>5.2 Full Dynamic Backface Deformation Reconstruction with a Fiber Bragg Grating Array Under Kinetic Conditions</td>
<td>82</td>
</tr>
<tr>
<td>5.2.1 FBG Multiplexing with Peak Crossover</td>
<td>83</td>
</tr>
<tr>
<td>5.2.2 Peak Tracking with the Nearest Neighbor Algorithm</td>
<td>86</td>
</tr>
<tr>
<td>5.2.3 Results for Full BFD Reconstruction with an FBG array</td>
<td>88</td>
</tr>
<tr>
<td>5.2.4 Summary for Full BFD Reconstruction with an FBG Array</td>
<td>88</td>
</tr>
<tr>
<td>5.3 Stick-Slip Compensation for Strain to BFD</td>
<td>91</td>
</tr>
<tr>
<td>5.3.1 Stick Slip Compensation</td>
<td>92</td>
</tr>
<tr>
<td>5.3.2 Timing Compensation in Silicone</td>
<td>98</td>
</tr>
<tr>
<td>5.3.3 Results</td>
<td>100</td>
</tr>
<tr>
<td>5.4 Advanced Considerations</td>
<td>101</td>
</tr>
<tr>
<td>5.4.1 Test 1</td>
<td>102</td>
</tr>
<tr>
<td>5.4.2 Test 2</td>
<td>107</td>
</tr>
<tr>
<td>5.5 Conclusion</td>
<td>110</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>112</td>
</tr>
<tr>
<td>6.1 Contributions</td>
<td>112</td>
</tr>
<tr>
<td>6.2 Strain to BFD Algorithm Based on Kinetic Friction</td>
<td>115</td>
</tr>
<tr>
<td>6.3 Sensor Array to Shape Algorithm</td>
<td>115</td>
</tr>
<tr>
<td>6.4 Silicone Dynamics Compensation for the Strain to BFD Algorithm</td>
<td>116</td>
</tr>
<tr>
<td>6.5 Future Work</td>
<td>116</td>
</tr>
<tr>
<td>6.5.1 Peak Crossover Tracking under High Dynamic Conditions</td>
<td>117</td>
</tr>
<tr>
<td>6.5.2 Improved Optical Sensing of Electric Fields in Harsh Environments</td>
<td>118</td>
</tr>
<tr>
<td>6.5.3 Proper Characterization of Nonlinear Strain Transfer Effects in Silicone</td>
<td>118</td>
</tr>
<tr>
<td>6.5.4 Body Armor Shape Sensing at Extremely High Dynamics</td>
<td>119</td>
</tr>
<tr>
<td>6.5.5 SCOS Antenna Design with Advanced Antenna Design Concepts</td>
<td>119</td>
</tr>
<tr>
<td>Appendix A. Section A1 Graphical User Interface Code</td>
<td>125</td>
</tr>
<tr>
<td>Appendix A. Section A.2 Image Processing and Nearest Neighbor Algorithm</td>
<td>155</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 4-1. Test matrix for the 12 specimens tested where the FBG is vertically offset 1 cm from the impact location and the 6 tests where the FBG is directly impacted by the projectile.... 51
Table 4-2. Deformation and fiber offset data for the tested specimens with an offset fiber........ 60
Table 4-3. Deformation and fiber offset data for the tested specimens with a directly impacted fiber....................................................................................................................................... 61
Table 4-4. Material property summaries for Smooth-Sil 950 and Sorta-Clear 40................. 64
Table 4-5. Comparison between depths obtained with and without the different sensing layers. 66
Table 4-6. Pullout test results for measuring the pullout force due to application of different lubricant types................................................................................................................................................ 67
Table 4-7. Test results comparing silicone embedded FBG strain response with and without coating. .................................................................................................................................................. 68
LIST OF FIGURES

Figure 1-1. Body armor distributes the energy of the projectile through a larger portion of the material. ................................................................................................................................... 2
Figure 1-2. Severe injury and even death can be induced on the body armor user through blunt force trauma. ................................................................. 2
Figure 1-3. Current body armor testing involves placing clay backing behind the body armor and then measuring the deformation of the clay after the impact. 3

Figure 2-1. Example of high speed imaged BFD. ................................................................. 12
Figure 2-2. a) Simulated deceleration, b) simulated velocity, and c) simulate displacement over time. ........................................................................ 13
Figure 2-3. (left) (blue)Actual simulated deceleration (red) down sampled deceleration (10 samples), (right) (blue) actual simulated displacement (red) reconstructed displacement from down sampled deceleration. ....................................................................................................................................... 14
Figure 2-4. Accuracy as a function of downsampling %. .......................................................... 15
Figure 2-5. Decay time for deceleration compared to the repetition rate required from the interrogator for a certain reconstructed position accuracy. ...................................................... 15
Figure 2-6. A fiber Bragg grating consists of a periodic change in the refractive index of the core. This periodic change reflects a specific wavelength called the Bragg wavelength, \( \lambda_B \). 16
Figure 2-7. Reflection spectrum for an FBG. The peak of the reflected spectrum \( \lambda_B \) will change due to thermal and strain effects on the FBG. ................................................................................................................................. 16
Figure 2-8. Custom fibers such as multi core fibers are special made for shape sensing. 17
Figure 2-9. LUNA is one of many companies that offer fiber optic shape sensing solutions. 18
Figure 2-10. Common applications for shape sensing with FBGs include airplane wings and windmill blades. .......................................................................................................................... 19
Figure 2-11. Typical shape sensing with fiber optic sensors involves adhering the fiber optic sensors to a surface with a known neutral reference. ................................................................. 20

Figure 3-1. A model SLE-101 swept laser source from Insight Photonics Solutions is capable of sweeping ......................................................................................................................................... 24
Figure 3-2. Optical setup for full-spectrum high-speed interrogation of a fiber Bragg grating consisting of a swept laser source, a fiber optic circulator, a photodetector (PD), and an oscilloscope (OSCOPE). ........................................................................................................................................ 25
Figure 3-3. (a) A rising/falling clock edge initiates (b) a new sweep linear in wavelength. (c) The time domain waveform is converted into (d) a time varying wavelength spectrum which can be represented by (e) a false color representation. ................................................................................................. 26
Figure 3-4. Optimization procedure for determining the strain gradient across the FBG. An initial assumption is made for a strain profile which is fed into the transfer matrix. The variance between the measured spectra and the simulated spectra are compared and the strain profile is altered until the variance is minimized. .................................................................................................................. 26
Figure 3-5. A second order approximation is first made on the strain profile, this second order approximation is then used as a starting point for individual strain point optimization. ...... 31
Figure 3-6. (left) (blue) actual measured distorted FBG spectrum and (red) optimized FBG strain profile using a second order polynomial approximation. ................................................................. 32
Figure 3-7. (left) (blue) actual measured distorted FBG spectrum and (red) optimized FBG strain profile.................................................................................................................................. 32
Figure 3-8. The split Hopkinson tensile bar consists of two bars holding a tapered specimen in the middle. Stress waves in the bars produce displacements in the specimen resulting in strain. The FBG is mounted across the tapered aluminum specimen to monitor the strain across the specimen over time. ........................................................................................................... 33
Figure 3-9: The DIC software allows for strain profile reconstruction by tracking a speckle pattern along the surface of the specimen. This strain profile was measured at 235 µs. .... 35
Figure 3-10: (solid blue line) Measured Strain Profile from DIC at 235 µs and (dashed red line) Optimized Strain Profile for 230 µs. The strain profiles from the FBG and DIC agree with each other until the peak splitting phenomenon. ........................................................................ 36
Figure 3-11: False Color Representation of Captured FBG Spectra over Time. Full-spectrum high-speed interrogation allows the spectrum deformations to be captured. These deformations can later be analyzed to deduce the strain profile across the FBG. ............. 37
Figure 3-12: Measured Percent Strain on the FBG from the Strain Gauges (solid red line), DIC (dashed blue line) and FBG (dot dashed black line). The percent strain over time from the FBG agrees with the percent strain over time deduced by the DIC and strain gauges, this verifies that an FBG is a reliable tool for Hopkinson bar interrogation. ................................. 37
Figure 3-13: Measured Average Strain Rate from the FBG Using Peak Detection on the Measured Spectra: Strain Gauges (solid red line), DIC (dashed blue line) and FBG (dot dashed black line). The highest strain rate achieved is approximately 500 s⁻¹. ......................... 38
Figure 3-14: The left column shows (solid blue line) the measured spectrums and (dashed red line) the optimized spectrums over 10 µs intervals. The right column shows the optimized strain profiles. The strain discontinuities shown at 240 µs and 250 µs indicate localized material failure which is important in material analysis......................................................... 39
Figure 3-15: (top) High speed camera image corresponding to 240 µs from the FBG measurement where a crack is first detected by the FBG, and (middle) high speed camera image corresponding to 305 µs where the crack first manifests itself from the high speed camera video images. (bottom) The broken fiber ends can be seen at 1395 µs on the high speed camera video images. ........................................................................................................ 40
Figure 3-16. a) Original FBG wavelength spectrum over time; c) sliced section of FBG spectrum over time shown with (red dots) start of tracked areas and (orange dots) end of tracked areas; c) purple dot indicates largest area found; d) two FBG strain profiles tracked over time. 41
Figure 3-17. The area based peak tracking method finds the n largest areas above a threshold for each frame corresponding to n gratings, and then compiles the largest areas over all frames to deduce the wavelength profile over time................................................................. 41

Figure 4-1. a) The gas gun setup shoots 12.7 cm diameter ball bearings using compressed gas cylinders. b) A box with either clay or ballistics gel is mounted into the observation chamber and c) A side view camera can be used to image through the ballistics gel to monitor the BFD over time. ................................................................................................................. 46
Figure 4-2. The fiber optic sensor is sewn onto a Kevlar sensing layer to dynamically sense body armor BFD.

Figure 4-3. The sensing layer is placed between the shoot pack and the ballistic gel box.

Figure 4-4. High-speed camera imaging allows for BFD tracking over time.

Figure 4-5. Fiber tensile strength testing was done by supergluing one end of the fiber Bragg grating onto a micron alignment stage and the other end to a fixed platform.

Figure 4-6. A drawtower FBG went out of the interrogation band at 3.57% strain. At 6.53% strain the superglue holding the FBG gave out.

Figure 4-7. Initial images of the (a) ORMOCER coated fiber, (b) ORMOCER-T coated fiber, and (c) 80 μm diameter fibers and final images of the (d) ORMOCER coated fiber, (e) ORMOCER-T coated fiber, and (f) 80 μm fibers before and after being impacted repeatedly with a projectile travelling 120 m/s.

Figure 4-8. Strains output by the FBGs written in ORMOCER® coated fiber.

Figure 4-9. Strains output by the FBGs written in ORMOCER®-T coated fiber.

Figure 4-10. Strains output by the FBGs written in 80 μm diameter fiber.

Figure 4-11. (a) a photograph of the sensing layer with stitched on optical fiber where the black segments indicate the stitch locations and (b) a schematic of the sensing layer showing dimensions and stitch locations.

Figure 4-12. (a) Photograph of the clay box after impact showing the deformation profile from the ballistic impact with dimensions in cm and (b) a cross section of the 3D scan of the deformation profile.

Figure 4-13. Maximum strain measured by the FBG in each test as a function of deformation at the FBG location.

Figure 4-14. FBG strain response for the four directly impacted specimens.

Figure 4-15. False color full spectrum plot for the FBG in test 15.

Figure 4-16. The FBGs are embedded in a silicone sensing layer to survive better at higher dynamics. (left) smooth sil silicone tears during the impact event but (right) sorta clear silicone stays impact during the event.

Figure 4-17. A pullout test was done to determine the best lubricant to be used in conjunction with the silicone sensing layer.

Figure 4-18. Strain profiles compared for FBGs embedded in silicone with no Vaseline and FBGs embedded in silicone with Vaseline.

Figure 5-1. (a) Initially the projectile causes a normal force that pushes the optical fiber onto the adjacent layers. (b) The static friction causes the optical fiber to stretch as the backing material deforms. (c) When the force of the stretched optical fiber exceeds the static friction force the optical fiber starts to slip.

Figure 5-2. a) Tight sew configuration and b) loose sew configuration in Kevlar.

Figure 5-3. Strain for a) tight sewn specimen under dynamic impact and b) loose sewn specimen under dynamic impact.

Figure 5-4. (blue) Maximum BFD acceleration based off high speed imaging and (red) strain profile deduced from tightly sewn FBG.
Figure 5-5. Reflection spectrum from an FBG during a 9 mm FMJ round shot on 50 layer Kevlar shoot pack in clay.

Figure 5-6. (a) (line) BFD acceleration (circles) data points, (b) BFD velocity, and (c) BFD.

Figure 5-7. Displacement over time from a 9 mm FMJ shot into a 50 layer Kevlar shoot pack with clay backing material.

Figure 5-8. Strain profile from FBG during the 12.7 mm ball bearing impact into body armor with ballistics gel backing.

Figure 5-9. (a) BFD acceleration, (b) BFD velocity, and (c) BFD.

Figure 5-10. Displacement over time determined from the (dashed) FBG and (solid) high-speed imaging.

Figure 5-11. Normalized displacement over time determined from the (dashed) FBG and (solid) high-speed imaging.

Figure 5-12. 3 FBGs are multiplexed on the same sensor channel line each with a different offset from the center of impact to reconstruct the full BFD shape over time.

Figure 5-13. It is difficult to associate peaks when multiple peaks cross over one another.

Figure 5-14. (left) Original image and (right) edge tracked image.

Figure 5-15. (left) Original contour plot from 3 FBG data, (middle) edge tracked image points and (right) scattered edge tracked data points.

Figure 5-16. The edge tracked data points are linked with a nearest neighbor algorithm in a reverse order to allow for the estimation algorithm to properly determine the appropriate nearest neighbor.

Figure 5-17. Displacements for each sensor are reconstructed, where (top) is taken directly from Chapter 3 and shows the BFD deduced from (dashed) the FBG compared to the BFD deduced by (solid) high speed imaging. By compiling these displacements spatially over time, it is possible to reconstruct the full BFD shape over time.

Figure 5-18. (red) interpolated BFD shapes from (white) FBG locations on BFD. 3 FBGs are located on one side of the BFD and mirrored to generate 6 data points.

Figure 5-19. (solid) Deceleration deduced from high speed imaging and (dashed) strain deduced from FBG.

Figure 5-20. (solid) Displacement deduced from high speed imaging and (dashed) strain to BFD algorithm.

Figure 5-21. a) Identify final stick-regime, b) double differentiate stick regime and scale until monotonic with previous data, d) vertically shift rest of data until final data point has 0 amplitude. e) scale final data portion until graph is monotonic.

Figure 5-22. Process for static friction compensation.

Figure 5-23. (solid) reshaped strain profile and (dashed) actual acceleration profile from high speed camera.

Figure 5-24. Comparison between the deduced displacement profiles after stick-slip compensation.

Figure 5-25. (solid) Actual deceleration profile deduced from high speed imaging and (dashed) deduced deceleration profile deduced from reshaped FBG data. a) non-normalized b) normalized.
Figure 5-26. (dashed) Calculated displacement profile and (solid) actual displacement profile. a) non-normalized displacement profiles, b) normalized displacement profiles, c) non-normalized time scaled displacement profiles and d) normalized time scaled displacement profiles. .................................................................................................................................. 99

Figure 5-27. a) Comparison between the calculated displacement and the actual displacement for 6 different tests, b) displacement percent error, c) timing percent error, and d) actual timing comparison ........................................................................................................................................... 100

Figure 5-28. There are 2 FBGs in the sensing mat. One in the center and one at the very edge. 103

Figure 5-29. (left) original wavelength data, and (right) deduced strain profiles from FBG data. ..................................................................................................................................................... 103

Figure 5-30. (blue) strain profile deduced from FBG1, (red) strain profile deduced from FBG2, and (yellow) strain profile deduced from the subtraction between the two profiles. ........ 105

Figure 5-31. (blue) actual deceleration profile deduced from high speed imaging, (red) deceleration profile deduced from FBG. (left) normalized deceleration profiles (right) non-normalized deceleration profiles. ........................................................................................................ 106

Figure 5-32. (blue) Reconstructed displacement determined by FBG and (red) displacement determined by high speed imaging. ........................................................................................................ 106

Figure 5-33. (left) non normalized displacement comparison for double differentiation method and (right) normalized displacement comparison. .............................................................................. 107

Figure 5-34. FBG setup for test 2. 1 FBG is located 1.5 cm away from the center, and the other FBG is located 2.5 cm away from the center on the other side of the silicone mat. ........ 107

Figure 5-35. (left) contour plot of the original data and (right) comparison of the two strain profiles over time. ................................................................................................................ 108

Figure 5-36. Double differential method combined with absolute difference method. (blue) displacement reconstructed from FBG, and (red) displacement determined from high speed imaging. (left) non normalized displacement comparison for absolute difference method and (right) normalized displacement comparison. .............................................................................. 109

Figure 5-37. Double differential method only. (blue) displacement reconstructed from FBG, and (red) displacement determined from high speed imaging. (left) non normalized displacement comparison for double differentiation method and (right) normalized displacement comparison.......................................................................................................................... 110
1 INTRODUCTION

Modern body armor is extremely effective at stopping projectiles from penetrating the wearer, but users can still suffer from serious injuries known as blunt force trauma. Blunt force trauma can come in the form of broken bones, severe organ injuries, internal bleeding, and so on. It is caused by the large deformation of the body armor during impact, and the speed of the deformation which delivers force to the wearer during the impact. As a result there is a focus on redesigning modern body armor to prevent blunt force trauma, along with testing and evaluating how effective body armor is at mitigating blunt force trauma.

1.1 Body Armor Testing

Body armor is becoming thinner, lighter in weight and more flexible. While the use of high impact strength deformable composite materials decreases the likelihood of ballistic penetration, such materials distribute the impact energy through a larger portion of the material. Blunt force trauma due to back face deformation (BFD) of the armor into the human body are therefore becoming more prevalent [1].

Figure 1-1 shows that body armor typically consists of a hard plate in front of a Kevlar layer in order to stop a projectile from penetrating the wearer. Sometimes body armor only consists of Kevlar. The body armor stops the projectile by distributing the energy of the projectile through a larger portion of the material. Figure 1-2 shows that there is still severe
injury from blunt force trauma despite the fact that the projectile did not penetrate the user. These injuries can be very severe and cripple an individual for the rest of his/her life.

Figure 1-1. Body armor distributes the energy of the projectile through a larger portion of the material.

Figure 1-2. Severe injury and even death can be induced on the body armor user through blunt force trauma.

Figure 1-3 shows that the current state of the art T&E for BFD involves placing a clay backing behind the armor and then measuring the deformation of the clay after impact [2]-[3]. The standard clay that is used is Roma Plastilina #10 because it has a good correlation with soft human tissue [2]. In the testing standard created by the National Institute of Justice the
maximum BFD needs to be less than 44 mm with an accuracy of 1 mm [3]. There are different methods used to measure the BFD left in the clay. Two common measurement techniques (laser profilometer and digital caliper) were compared in a recent National Academies Review [3]. Alternative backing materials have been proposed, including ballistic gel [3].

Figure 1-3. Current body armor testing involves placing clay backing behind the body armor and then measuring the deformation of the clay after the impact.

However, there are several primary deficiencies in the BFD measurement. (1) The response of the clay is highly variable, depending upon factors such as strain rate, shear and thermal loading history. These variations lead to significant variations in the final BFD measurement [1]. (2) The measurement of the BFD in clay is based on the assumption that all of the deformation in the clay is plastic, therefore the post-impact BFD matches the maximum
BFD. However, measurements have shown that the clay backing can often significantly underestimate the maximum BFD [1]. While the use of primarily elastic backing materials, such as ballistic gel, could alleviate some of this variation, they require high-speed imaging throughout the ballistic impact, which is difficult to implement or expected to be cost-ineffective [1]. There are two existing methods for performing dynamic measurements namely flash x-ray cineradiography and digital image correlation (DIC) [4]. The National Academies report stated that the flash x-ray approach lacks the necessary spatial and temporal resolution [5]. The DIC method has the potential to achieve the necessary temporal and spatial resolution but requires line of sight, which means that it cannot be used with any form of standard backing material. Finally, the National Academies Review panel determined that depth of the BFD alone is an inadequate indicator of injury probability. This assessment was also backed up by other independent research [6]-[7]. Current instrumented anatomical surrogates are not detailed enough to enable further correlation between BFD and specific events [1]. These deficiencies drive the need for high-spatial resolution, high-speed, and high-fidelity BFD measurements.

Current alternative dynamic body armor testing techniques consist of placing strain or pressure sensors in or behind the body armor to determine the performance of the body armor over time. For example, by weaving Nichrome wire into Kevlar body armor, the local and global strains over the course of impact can be determined [8]-[9]. These strains can then be used to assess the performance of the body armor and necessary improvements can be made. Another example is where pressure sensors are placed in the backing material behind the body armor [10]-[11]. The pressure over time experienced by the backing material can be recorded and analyzed, and the effectiveness of the body armor can be assessed.
This dissertation shows how to test the rate of BFD with fiber optic strain sensors. The fiber optic sensing element that is used is the fiber Bragg grating (FBG). The measurement is done by placing the FBG behind the body armor and then using the strain information to reconstruct the shape of the BFD over time.

Three challenges had to be overcome to accomplish this task. Firstly, the interrogation of the optical fiber sensors is increased to meet the dynamics of the ballistic impact. Secondly, the FBG is configured to survive an impact event when placed behind body armor. The survivability is attained by allowing the optical fiber to slip during the impact. Thirdly, a new shape sensing algorithm based on friction dynamics is developed.

Due to the capability to measure the dynamic BFD with the clay backing in place, the FBG sensor results can also be used as calibration for the clay backing material under different environmental conditions, prior to their use as backing materials in ballistic range testing. Ultimately, decreasing the uncertainty of the BFD criterion could lead to lighter weight armor designs, and therefore increase soldier mobility [1]. Finally, the FBG sensor network could be applied to specialized systems in Army Research Laboratory facilities and to increase the sensor density in instrumented anatomical surrogates.

There are a variety of advantages of the proposed integrated FBG sensor network: (1) It does not require a direct view of the surface; (2) It has an adjustable measurement range (i.e. it can measure small elastic deformations up to large plastic deformations); (3) It has a high maximum repetition-rate (up to 300 kHz); (4) High spatial density and resolution of the measurements can be achieved by multiplexing sensors, without the need for a separate channel per sensor; (5) the sensor network can be integrated directly into soft body armor or be used with any backing material; and (6) the sensor network could be implemented into an anthropomorphic
test module. These advantages are complementary to the post-impact BFD measurements in clay backing material and in-situ DIC measurements because the FBG sensor network has a lower spatial resolution. The full-surface imaging possible with the clay backing will provide the information for the reduced order structural model, which will further enable the fused measurement systems to attain high-resolution and high repetition-rate BFD measurements.

1.2 Contributions and Dissertation Outline

This dissertation describes in detail how to shape sense the BFD of body armor with fiber optic sensors. The dissertation is divided up into five chapters. The first chapter talks about the background for body armor dynamics and FBGs. The next three chapters cover the three technical challenges that were overcome high-speed interrogation, optical fiber survivability, and the FBG to strain algorithm. The final chapter is the summary and future work for the project.

A summary of my contributions are the following:

1. I developed a high-speed full-spectrum interrogation system using commercial components.


2. I helped develop a method to ensure the fiber optic sensor survives a direct impact when placed behind body armor.


3. I developed a back face deformation depth sensing algorithm based on kinetic friction induced on a single FBG.


4. I extended the back face deformation depth sensing algorithm to an array of FBGs enabling the entire back face deformation to be analyzed.

5. I developed a method to compensate for stick-slip friction to allow for back face deformation depth sensing when FBGs are embedded in a silicone sensing layer.

6. (Other Contributions) I have made contributions to slab coupled optical sensor (SCOS) development which allows for optical sensing of electric fields in harsh environments. These contributions can be found in:

F. Seng, "An Exploration in Fiber Optic Sensors" (2016). All Theses and Dissertations. 6101. https://scholarsarchive.byu.edu/etd/6101


7. I developed dipole antennas for fiber optic electric field sensors which flip the directional sensitivity and enhance the overall sensitivity.

This chapter discusses the background for this research. The background includes both an overview of the BFD of body armor and the operation of fiber Bragg gratings.

2.1 Basic Body Armor Dynamics

This section describes how to model the BFD of body armor as a first order system. The results of this section are an ideal plot of the BFD and an estimate for the required interrogation speed. The estimate for the speed is based off high speed side imaging done in this work and high speed imaging done in other work [12].

The body armor decreases the speed of projectile. So the initial velocity of the BFD is the same as the projectile velocity immediately before impacting the body armor. The body armor then decreases the velocity of the projectile. A simple model of the projectile dynamics is to model the velocity of the BFD as a first order linear system as given by

$$T \frac{dv}{dt} + v = 0,$$

where $T$ is the system time constant. The value of $T$ depends on the specific body armor.

The solution of Eq. (2-1) is given by

$$v(t) = v_0 \exp \left( -\frac{t}{T} \right).$$
The BFD is calculated from the velocity by taking the integral as given by
\[ x(t) = \int v_0 e^{-\frac{t}{T}} dt. \]  
(2-3)

Taking the integral and using the initial condition of \( x(0)=0 \) results in
\[ x(t) = T v_0 \left( 1 - e^{-\frac{t}{T}} \right). \]  
(2-4)

The acceleration can also be calculated from the velocity by taking the derivative of the velocity as given by
\[ a(t) = -\frac{v_0}{T} e^{-\frac{t}{T}}. \]  
(2-5)

High speed imaging into ballistics gel is one current method for capturing the full dynamics of the BFD. However, this method is expensive and difficult to implement. Ballistics gel is also elastic and therefore recovers its shape after impact, unlike clay which retains its shape which allows for a full analysis of the shape the BFD leaves in the backing material.

Work has been done with high speed imaging to get an estimate of the system time constant \( T \). Figure 2-1 shows an example of a frame taken from high speed imaging. It is possible to gauge the BFD over time by compiling the BFD tracked from individual frames over time.

Figure 2-1 shows the BFD deceleration measured with high-speed imaging and the correspond fit. The fit results in a time constant of \( T=0.28 \) milliseconds. Using this model it is possible to calculate the minimum sample rate required to reconstruct a BFD based off the simulation. As will be explained later the system measures the acceleration of the BFD and then the acceleration is integrated to get velocity, and then the velocity is integrated to get the BFD.

Figure 2-2 shows a) the simulated deceleration profile, b) the simulated velocity profile, and c) the simulated displacement profile using a time increment of \( \Delta t=1 \) \( \mu s \) and a time constant...
of $T=0.28$ ms. Figure 2-3 shows the concept of accuracy and down sampling illustrated on a data set. Suppose that the decay time was 0.28 milliseconds and that the repetition rate was 5 kHz. It can be seen that there is missing information on (left)(red) the sampled curve compared to (left)(blue) the actual curve. As a result of this missing information (right)(red) the reconstructed position over time has errors compared to (right)(blue) the actual position over time by 10%.

Figure 2-1. Example of high speed imaged BFD.
Figure 2-2. a) Simulated deceleration, b) simulated velocity, and c) simulate displacement over time.
Figure 2-4 shows the reconstructed position accuracy from deceleration as a function of down sampled percentage. Specifically,

1. The simulated exponential deceleration is sampled at a specific rate.

2. Numerical integration is used to get the velocity using cumtrapz in MATLAB.

3. Numerical integration of velocity is performed once again using cumtrapz to get the position over time.

4. The calculation of the final position from the downsampled deceleration is compared to the calculated position when the deceleration is not downsampled. The error between the two is taken for a given sample rate. As can be seen, 80% error is achieved with only 4 sample points.

Figure 2-3. (left) (blue) Actual simulated deceleration (red) down sampled deceleration (10 samples), (right) (blue) actual simulated displacement (red) reconstructed displacement from down sampled deceleration.

Figure 2-5 shows the required repetition rate from the high speed interrogator for a certain decay time of a deceleration signal. As can be seen, the faster the decay time the more sample points are required to properly reconstruct the position over time. The decay times in this
work are approximately 0.28 ms, therefore a repetition rate of about 10 kHz is required for a
90% accuracy for the final BFD. There are flat sections because an accuracy threshold is
specified per line plot. In otherwords, the lines in Figure 2-5 show the minimum sample rate per
decay time to achieve a certain accuracy.

Figure 2-4. Accuracy as a function of downsample %.

Figure 2-5. Decay time for deceleration compared to the repetition rate required from the
interrogator for a certain reconstructed position accuracy.
2.2 Fiber Optic Sensors

With the rise of modern infrastructure and systems, testing and evaluation of specific components and systems such as body armor is becoming more important. Fiber optic sensors [13]-[20] are ideal for these applications due to numerous advantages such as their compact, dielectric and lightweight nature [21]-[27].

Figure 2-6. A fiber Bragg grating consists of a periodic change in the refractive index of the core. This periodic change reflects a specific wavelength called the Bragg wavelength, $\lambda_B$.

Figure 2-7. Reflection spectrum for an FBG. The peak of the reflected spectrum $\lambda_B$ will change due to thermal and strain effects on the FBG.

This work focuses on the application of a fiber optic strain sensor known as the fiber Bragg grating (FBG) to body armor testing. FBGs are extremely lightweight and compact, making them ideal for nonintrusive measurements. Figure 2-6 shows that an FBG is a section of...
optical fiber with a sinusoidal periodic variation in the refractive index of the core. The FBG reflects a specific wavelength as given by [27]

\[ \lambda_B = 2n_e \Lambda, \]  

where \( \lambda_B \) is the reflected wavelength called the Bragg wavelength, \( n_e \) is the effective refractive index of the optical fiber fundamental mode and \( \Lambda \) is the grating period.

Figure 2-7 shows the reflection spectrum of an FBG used in this work. The peak of the reflected spectrum corresponds to the Bragg wavelength given by Equation (2-9). A strain applied to the FBG causes a change in both the grating period \( \Lambda \), and the effective index of refraction of the fiber mode \( n_e \), resulting in a shift of the reflection spectrum. The shift in the Bragg wavelength with respect to strain is given by [28]

\[ \frac{\Delta \lambda_B}{\varepsilon} = 1.2 \frac{nm}{m\varepsilon}. \]  

(2-10)

2.3 Shape Sensing with Fiber Optic Sensors

Figure 2-8. Custom fibers such as multi core fibers are special made for shape sensing.
Shape sensing with fiber Bragg gratings is a current area of research with many research groups developing new methods to interrogate hundreds or thousands of FBGs simultaneously. Custom made fibers such as the tri-core fiber shown in Figure 2-8 are special made for shape sensing, and new applications for shape sensing. Figure 2-9 shows that due to a rise in the demand for such sensors, companies such as FBGS, OFS fitel, and LUNA have developed fiber optic shape sensing technology commercially available for clients with such a need.

Figure 2-9. LUNA is one of many companies that offer fiber optic shape sensing solutions.

Figure 2-10 shows that common applications for shape sensing with FBGs include airplane wings, windmill blades, or bridges for structural health monitoring [29]-[31]. By embedding FBGs into these structures it is possible to determine the fatigue of the structure by
its shape. NASA has been developing shape sensing methods for a variety of different systems such as aeronautics and launch vehicles [32]. There has also been an increased demand for shape sensing for surgical endoscopes or other medical applications by the medical industry which has led to an increase in demand for fiber optic shape sensors [33].

Figure 2-10. Common applications for shape sensing with FBGs include airplane wings and windmill blades.

Figure 2-11 shows typical FBG shape sensing methods involve bonding the fiber directly onto the structure being monitored and then using some known numerical strain to shape method to reconstruct the shape over time. In this particular case the fiber optic sensor is bonded to the surface of a beam. The strain on the fiber optic sensor will be determined by how far it is from the neutral axis of the beam and the curvature of the beam. Therefore, by knowing what the strain on the fiber optic sensor is, and by knowing what the distance from the neutral axis is, it is possible to deduce the curvature of the beam.

Unfortunately these current methods are not applicable to dynamic body armor testing due to the high dynamics of the body armor. Efforts to implement these methods have resulted in the optical fiber shattering under impact. As a result this dissertation introduces the development
of a new method for shape sensing impact based events via the kinetic friction induced on the FBG under an impact event.

Figure 2-11. Typical shape sensing with fiber optic sensors involves adhering the fiber optic sensors to a surface with a known neutral reference.
3 HIGH REPETITION-RATE FULL-SPECTRUM INTERROGATION OF FIBER BRAGG GRATINGS

This section introduces the high-speed full-spectrum interrogation method for FBGs used in this work and illustrates its application on a split Hopkinson bar. Different wavelength to strain demodulation techniques are also introduced in this work. Specifically, optimization with the transfer matrix and fmincon to deduce the strain profile across an FBG is discussed, as well as a fast method to demodulate strain information necessary for strain to BFD reconstruction from an FBG under a dynamic impact.

3.1 High Speed Interrogation for Fiber Bragg Gratings

High repetition-rate is required for FBG interrogation under dynamic body armor testing. The Background Chapter shows that the repetition-rate of the interrogation needs to be over 10’s of kHz to accurately measure the BFD for a ballistic impact. In addition to the high repetition-rate, the interrogation system also needs to measure the full spectrum of the FBG because of the distortion in the reflection spectrum of the FBG. The full spectrum distortion of the FBG contains strain gradient information which is crucial to the algorithm later introduced to reconstruct the BFD over time. This section describes the high repetition-rate full-spectrum FBG interrogation method used in this work.
3.1.1 Current Technology

High speed interrogation of FBGs can be performed through filtering of the FBG response signal through a narrow-bandwidth Fabry-Perot filter and applying a follower circuit to track the peak wavelength response of the FBG sensor. A commercial version of this approach is the Micron Optics sm690 interrogator which provides 2 MHz interrogation of up to four FBG sensors [34]. Separate channels of the interrogator must be used for each sensor due to the following filter. The advantage to this instrument is the fast data acquisition rate. The disadvantages are the need to build an additional follower circuit for each additional sensor to be monitored and the possibility of wavelength hopping due to changes in the FBG reflected spectrum due to non-uniformities in the local strain distribution. Wavelength hopping occurs when non-uniform strain along the FBG sensor distorts the reflected spectra such that a single peak is split into multiple peaks [35]. At the time at which the splitting first occurs, the direction of the peak wavelength follower is highly sensitive to local perturbations. Therefore, the peak wavelength interrogator may appear to rapidly change response or may not produce a repeatable response.

A second approach to measure the strain of a FBG sensor array is to illuminate the FBG sensors with a broadband source and to project the reflected spectra through a diffraction grating. The output of the grating is then projected onto a CCD array and the individual FBG peak wavelengths are determined from the pixel locations with maximum local intensities. A commercial system applying this method has been developed by Technobis. This system can interrogate a maximum of 32 FBG sensors at a maximum data acquisition rate of 19.3 kHz [36]. For this system, the data acquisition rate is limited by the response delay time of the CCD array. Mueller et al. [37] replaced the CCD array with a position sensitive detector (PSD), which can
detect the centroid location of a light spot projected onto its surface. Due to the reduced delay
time of the PSD, interrogation of multiple FBG sensors can theoretically be performed up to 300
kHz. To date, however the research group has only demonstrated the technology at interrogation
rates up to 12 kHz for four acquisition channels. While these instruments collect the full-spectral
response of the FBGs, this information is not saved and therefore no information on the local
strain non-uniformity is obtained. The output of these interrogators is therefore subject to the
same risk of peak wavelength hopping as the Micron Optic sm690.

An existing alternative to measuring strain components at discrete locations and applying
a structural model for shape reconstruction is to measure local curvatures directly. By measuring
the local curvatures at a high density of sensor locations, accurate shape reconstruction can be
achieved without the need for a prior structural response model. Luna Innovations has a
commercially available system based on multiple core optical fibers [38],[41]. The Luna
Innovations Shape Sensing System applies optical frequency domain reflectometry to interrogate
the large number of FBG sensors required. Ko and Richards [39]-[40] applied the same optical
frequency domain reflectometry method to interrogate FBG strain sensor arrays in standard
optical fibers, for shape reconstruction of the flexible wing on the NASA Ikhana unpiloted aerial
vehicle. While each of these examples provides excellent shape reconstruction capabilities, the
scanning of a large number of FBG sensors through optical frequency domain reflectometry
induces a significant delay in the data acquisition process. For this reason the maximum data
acquisition rate is less than 1 kHz and not suitable for the impact testing environment.
Additionally, no full-spectral information on the response of the FBG sensors is obtained.

There are different methods for high-speed FBG interrogation [43]-[44]. Many of them
are based off averaging the motion of the FBG peak either through edge filtering or from
centroid tracking. However, under high dynamic events there are heavy peak distortions which make edge detection and centroid tracking inaccurate. For this work, high repetition-rate full-spectrum interrogation is used to interrogate the FBG to ensure high-speed capture of the FBG waveform under dynamic events with peak distortions.

3.1.2 High-Speed Full-Spectrum Interrogation for Fiber Bragg Gratings

Figure 3-1 shows that a model SLE-101 swept laser source from Insight Photonics Solutions is used to interrogate the FBGs in this work [45]-[46]. This solid state swept laser source can sweep at a rate of 90 kHz with an 80 nm sweep band centered on 1560 nm. It has a 0.4 Ghz/0.961 pm laser linewidth with a maximum output power of 10 mW. One primary advantage of this laser is that it is all solid state, allowing for a wide range of custom sweep settings and very linear sweeps.
Figure 3-2 shows that the swept laser source feeds into a fiber optic circulator. The reflected spectrum from the FBG is captured by a photodetector (PD) as a time domain signal, where the time is directly related to the sweeping wavelengths of the source. The PD converts the optical power from the FBG spectrum into an electrical voltage. The time domain waveform is then captured by an oscilloscope (OSCOPE).

Figure 3-2. Optical setup for full-spectrum high-speed interrogation of a fiber Bragg grating consisting of a swept laser source, a fiber optic circulator, a photodetector (PD), and an oscilloscope (OSCOPE).

Figure 3-3 shows the process diagram for converting the captured time domain FBG waveform from the oscilloscope into a wavelength domain waveform. A linear sweep in wavelength is initiated on every rising and falling edge of the captured clock signal. By knowing the starting sweep wavelength and the ending sweep wavelength, the wavelength spectrum is measured by linearly mapping the captured data points to the linearly swept wavelength. By plotting a false color representation of the captured FBG reflection spectra over time, it is possible to observe the spectrum shifts over time.
One of the most powerful uses of a fiber Bragg grating is to deduce entire strain gradients along the grating. Typical strain sensors allow for a single measurement which averages the strain across the sensor, whereas fiber Bragg gratings allow for information from a strain gradient distributed across the grating to be manifest in the wavelength spectrum. Previous work describes testing high-speed full-spectrum interrogation of FBGs on a Hopkinson bar. The strain gradients deduced from the full spectra of the FBG were verified against high speed imaging.

This section describes the optimization algorithm used to deduce the strain gradients in detail. To obtain strain profile information along the FBG from the measured reflection spectrum, an optimization procedure was applied that involves generating simulated deformed
reflection spectra given strain profiles. These predicted reflection spectra are then compared against the actual measured spectrum.

Many different optimization algorithms have been developed to obtain the distributed strain profile through an FBG [35],[47]. An efficient approach for calculating the reflected spectrum of a FBG due to a nonlinear strain profile along an FBG is through the use of the transfer matrix method, where the FBG is split up into many small segments with constant properties. In this work, the modified transfer matrix formulation is used to account for the fact that high strain gradients are expected [48].

The transfer matrix of a single $m_{th}$ section of the grating is given by

$$F_m = \begin{bmatrix} \cosh(\Omega \Delta z) - j \frac{\xi}{\Omega} \sinh(\Omega \Delta z) & -j \frac{K}{\Omega} \sinh(\Omega \Delta z) \\ j \frac{K}{\Omega} \sinh(\Omega \Delta z) & \cosh(\Omega \Delta z) + j \frac{\xi}{\Omega} \sinh(\Omega \Delta z) \end{bmatrix},$$  \hspace{1cm} (3-1)

where $\Delta z$ is the segment length, and

$$\Omega = \sqrt{K^2 - \xi^2},$$ \hspace{1cm} (3-2)

where $K$ is the ac coupling coefficient and $\xi$ is the dc self-coupling coefficient. The ac coupling coefficient is defined as

$$K = \frac{\pi}{\lambda} s \Delta n g(z),$$ \hspace{1cm} (3-3)

where $s$ is the fringe visibility, $\Delta n$ is the refractive index contrast of the grating, $g(z)$ is the change in the index contrast due to apodization during manufacturing and is commonly assumed to be a Gaussian of the form

$$g(z) = \exp \left( -\frac{4\ln(2)(z-\frac{L}{2})^2}{\rho^2} \right),$$ \hspace{1cm} (3-4)
where $\rho$ is the apodization constant, and $L$ is the total length of the grating. The dc self-coupling coefficient is defined as

$$\xi = \frac{2\pi}{\lambda} n_e - \frac{\pi}{\Lambda},$$

(3-5)

where $\lambda$ is the free space wavelength, $n_e$ is the effective refractive index of the optical fiber fundamental mode, and $\Lambda$ is the effective grating period.

The grating period varies along the length of the grating with the applied strain $\varepsilon(z)$ and the strain gradient $\varepsilon'(z)$ as given by

$$\Lambda(z) = \Lambda_0 [1 + \varepsilon(z) + z\varepsilon'(z)].$$

(3-6)

Similarly, the strain also creates a variation in $n_e$ as given by

$$n_e(z) = n_{e0} - p_e n_{e0} (\varepsilon(z) + z\varepsilon'(z)),$$

(3-7)

where $p_e$ is the photoelastic constant of the FBG which for this work is 0.22 and $n_{e0}$ is the effective refractive index before any strain is applied.

Defining $F_m$ as the matrix of the $m_{th}$ section of the grating, the matrix $F$ for the combined grating is given by

$$F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} = F_m F_{m-1} \ldots F_2 F_1,$$

(3-8)

and from $F$ the reflection coefficient of the FBG is given by

$$R = \left| \frac{f_{11}}{f_{12}} \right|^2.$$  

(3-9)

The optimization algorithm then computes the difference between the optimized and measured spectra as given by

$$ f_{\text{merit}} = \sum_{j=1}^{s} [R_{\text{opt},j}(\lambda_j) - R_{\text{meas},j}(\lambda_j)]^2. $$

(3-10)
where $R_{opt,j}(\lambda_j)$ is the optimized FBG reflection spectrum and $R_{meas,j}(\lambda_j)$ is the measured FBG reflection spectrum.

Figure 3-4. Optimization procedure for determining the strain gradient across the FBG. An initial assumption is made for a strain profile which is fed into the transfer matrix. The variance between the measured spectra and the simulated spectra are compared and the strain profile is altered until the variance is minimized.

Figure 3-4 shows the block diagram for the algorithm to optimize an FBG spectrum to an unknown strain profile. An initial strain profile along the FBG is assumed which is used to generate a transfer matrix for the FBG, and therefore a predicted FBG spectrum. In this paper, the built-in MATLAB optimization method fmincon is used [49].

If the difference between the two spectra is within a certain tolerance, then the optimization algorithm deems the optimized FBG strain profile as acceptable and returns the
strain profile to the user. Otherwise, it repeatedly mutates the strain profile and continues to compare simulated and measured FBG spectra until the tolerance is reached.

The FBG used in this work is broken up into 90 individual pieces and the strain at each piece is evaluated via optimization. fmincon defaults to an interior point algorithm, as a result, it is important that it have the correct starting point. In other words, the algorithm needs the correct constraints in order to find the correct solution. This constrained process is shown in Figure 3-5 where the optimization procedure introduced in this paper begins by optimizing the strain profile across the FBG as a second order polynomial given by

$$
\varepsilon(z) = az(t)^2 + bz(t) + c,
$$

where \(a\), \(b\), and \(c\) are the three parameters to be optimized for. Large strain gradients are encountered during dynamic events, so \(a\) and \(b\) are typically constrained between \(-10^6\) and \(10^6\), \(c\) is usually constrained to be within \(5\times10^{-3}\) strain of the centroid of the FBG spectrum. A random number generator generates the initial values \(a\), \(b\) and \(c\) for fmincon within the constrained ranges. A loop reiterates the process until an acceptable solution is found. Ideally, this loop would reiterate until a number of solutions are found, and the best second order polynomial approximation solution would then be chosen.

The solution from the second order approximation is shown in Figure 3-6 where the left hand figure shows (blue) the measured waveform from an actual experiment and (red) shows the waveform generated from the second order polynomial approximation shown on the right hand figure. This strain profile is then used as the initial starting point for the second part of the optimization procedure where all 90 points are optimized for individually. The result from the second part of the optimization procedure is shown in Figure 3-7. The validity of this optimization method is shown in previous work \[50\]. However, this optimization procedure is
time prohibitive enough for a single frame, taking up to 30 minutes. Optimization for a whole set of 50 or so frames obtained from an FBG under ballistic impact takes approximately 2 days to generate all strain profiles for all frames when processed via parallel processing on all 4 cores of an Intel i-5 3.1 Ghz processor. Although optimization is not used for the later portions of this work to demodulate strain from the FBG, the strain gradient correspondence to the wavelength spread of the FBG will be used extensively.

Figure 3-5. A second order approximation is first made on the strain profile, this second order approximation is then used as a starting point for individual strain point optimization.
3.2.1 Measurement Setup

In this work a split Hopkinson tensile bar is used to pull a specimen in tension [51]-[54]. Figure 3-8 shows an illustration of a split Hopkinson tensile bar that is capable of applying high...
tensile, compressive, or torsional strain rates \((10^2 \text{ to } 10^4 \text{ s}^{-1})\) depending on how the ends of the bars are displaced.

Figure 3-8. The split Hopkinson tensile bar consists of two bars holding a tapered specimen in the middle. Stress waves in the bars produce displacements in the specimen resulting in strain. The FBG is mounted across the tapered aluminum specimen to monitor the strain across the specimen over time.

The system works by firing the striker into the anvil at the end of the incident bar. This causes a compressive wave that is reflected back as a tensile wave when it reaches the end of the anvil. This tensile wave then travels all the way down the length of the incident bar. When it reaches the specimen some of the stress pulse is transmitted through the specimen and some is reflected back based on the mismatch of the materials. The stress wave that goes into the specimen is then transmitted to the transmitted bar.

Both the incident and transmitted bars are made out of maraging steel while the specimen used was machined from 6061 aluminum to a gauged diameter of 0.51 cm and a gaged length of 2.5 cm. The bar used was a \(\frac{3}{4}\) inch tensile bar, the striker was 12 inches long and the firing pressure was 30 psi.
Figure 3-8 shows that the FBG is glued onto the specimen along the length of the specimen, therefore measuring the strain on the surface of the specimen. In this work, a micron os1100 FBG is used with a theoretical strain limit of around 5 millistrain. The FBG was glued to the specimen using M-Bond AE-10 produced by Micro Measurements. M-Bond AE-10 is a room temperature cure two part epoxy capable of surviving up to 10% elongation.

3.3 Measurement Results

To verify the results found from the FBG interrogation, strains were also measured by electrical strain gauges on both the incident and transmitted bars. Additionally, a Photron Fastcam SA-X2 took high speed video of the specimen during the test. The Fastcam SA-X2 was set to take images at 100k frames per second, and VIC-2D software by Correlated Solutions was used to conduct the DIC analysis to measure the strain along the bar.

There is an estimated 5 µs time difference between the captured high speed video image frames and the FBG wavelength time frames. Figure 3-9 shows a screen capture from high speed video. The frame corresponds to 235 µs, which is close to FBG spectrum measured at 230 µs. Figure 3-9 also provides the DIC strain measurement.

Figure 3-10 (solid) shows a line profile of the DIC strain measurement for the frame at time 235 µs along with (dashed) the FBG strain measurement corresponding to the time 230 µs. The optimization method described in Section 2.3 was used on the FBG spectra to determine the strain profile across the grating. To reduce the effects of the noise, a discrete cosine transform to represent 99% of the power from the data is applied [55]. Figure 3-10 shows that the new strain measurement based on high-speed FBG interrogation agrees well with the existing DIC measurement.
Figure 3-9: The DIC software allows for strain profile reconstruction by tracking a speckle pattern along the surface of the specimen. This strain profile was measured at 235 µs.

Figure 3-9 shows that the virtual DIC strain gauge is located in the center of the specimen for best tracking. The FBG is located directly at the top of the specimen, where DIC tracking would yield inaccurate data on a surface that is not directly facing the camera. The DIC virtual strain gauge also averages over 10 pixels, which would not allow it to measure very discrete changes in the strain profile along the specimen. This is a main advantage of using a high-speed full-spectrum FBG interrogator, in which a line of sight is not needed in order to deduce localized strain profiles.

Figure 3-11 shows a false color image of the captured FBG wavelength spectra over time. The high-speed full-spectrum interrogator has a repetition rate of 100 kHz. This means that every 10 µs a new FBG reflection spectrum is measured. There are FBG spectra in which the shape stays approximately the same but is simply shifted in wavelength. This uniform shift corresponds to uniform strain across the grating. However, there are also regions in which the spectrum is distorted because of non-uniform strain. The average strain across the FBG can be determined by finding the centroid of each reflection peak and then multiplying it by the strain sensitivity of 1.2 pm/µε. This averaging approach is called peak tracking.
Figure 3-10: (solid blue line) Measured Strain Profile from DIC at 235 µs and (dashed red line) Optimized Strain Profile for 230 µs. The strain profiles from the FBG and DIC agree with each other until the peak splitting phenomenon.

Figure 3-12 shows the strain at the location of the FBG using three different measurement methods. These methods are (dot dashed black line) using peak tracking on the measured FBG spectrums, (solid red line) estimating the average strain using the electrical strain gauges placed on the transmitted and incident bars (see Figure 3-8), and (dashed blue line) using DIC average over the grating location. As can be seen, all three methods roughly agree on the strain profile over time.

The region in Figure 3-12 around 40 µs corresponds to the time when the tensile wave reaches the location of the FBG. Around 50 µs the sample reaches a local peak and something starts to slip and holds a constant load until 220 µs when the sample is loaded again. Figure 3-13 shows the strain rate from the three measurement methods. The overall trends illustrated by the graphs agree with each other.
Figure 3-11: False Color Representation of Captured FBG Spectra over Time. Full-spectrum high-speed interrogation allows the spectrum deformations to be captured. These deformations can later be analyzed to deduce the strain profile across the FBG.

Figure 3-12: Measured Percent Strain on the FBG from the Strain Gauges (solid red line), DIC (dashed blue line) and FBG (dot dashed black line). The percent strain over time from the FBG agrees with the percent strain over time deduced by the DIC and strain gauges, this verifies that an FBG is a reliable tool for Hopkinson bar interrogation.

The high-speed full-spectrum interrogation method not only allows for the peak shift to be detected, but also allows for deformations in the FBG spectrum to be analyzed. These
deformations need to be analyzed to understand strain profiles and strain gradients across the FBG, this was not possible with conventional peak tracking.

Figure 3-13: Measured Average Strain Rate from the FBG Using Peak Detection on the Measured Spectra: Strain Gauges (solid red line), DIC (dashed blue line) and FBG (dot dashed black line). The highest strain rate achieved is approximately 500 s\(^{-1}\).

During the first 50 µs the FBG spectra stays approximately uniform. Between 50 µs and 200 µs some spectra are uniform while others are distorted. After the reflected tensile wave reaches the FBG around the time of 200 µs the spectra remain distorted.

Figure 3-14 shows five of the FBG spectra from time 210 µs to 250 µs. The right side of Figure 3-14 shows the calculated strain profiles. Using the FBG strain measurement conclusions about the cause of the failure point can be deduced. For example, at 4 mm along the grating length a discrete change in the strain occurs. It is assumed that this discrete change in the strain causes a break in the optical fiber. The exact cause of the discrete jump in strain is unknown.
However, it could be part of the glue breaking off of the aluminum bar resulting in different strain values on the two halves of the grating [56].

Figure 3-14: The left column shows (solid blue line) the measured spectrums and (dashed red line) the optimized spectrums over 10 µs intervals. The right column shows the optimized strain profiles. The strain discontinuities shown at 240 µs and 250 µs indicate localized material failure which is important in material analysis.
Figure 3-15 (top) shows the screen capture from 245 µs where a crack was first detected by the FBG and Figure 3-15 (middle) shows a break point on the glue that can be seen at 305 µs on the high speed camera video frames. Figure 3-15 also shows that the assumed FBG location overlaps the section where the glue broke off, which creates a discrete jump in the strain profile and peak splitting as is seen in Figure 3-14.

The fiber doesn’t separate from the specimen until a few stress pulses later at around 1350 µs. Figure 3-15 (bottom) shows the fiber separation from the specimen at 1395 µs where the broken fiber ends can be seen. It is assumed that the sharp strain gradient is what caused the break in the fiber at the FBG location.

Figure 3-15: (top) High speed camera image corresponding to 240 µs from the FBG measurement where a crack is first detected by the FBG, and (middle) high speed camera image corresponding to 305 µs where the crack first manifests itself from the high speed camera video images. (bottom) The broken fiber ends can be seen at 1395 µs on the high speed camera video images.
The FBG is able to detect localized strain discontinuities at the moment of the event without a direct line of sight. The FBG used in this work was surface mounted to the specimen, but it has been shown that FBGs can be embedded inside Hopkinson bar specimen. This would allow for a direct internal strain profile analysis over time on the specimen during a Hopkinson bar test or any other high speed event.

### 3.4 Strain Calculation

![Figure 3-16](image.png)

Figure 3-16. a) Original FBG wavelength spectrum over time; c) sliced section of FBG spectrum over time shown with (red dots) start of tracked areas and (orange dots) end of tracked areas; c) purple dot indicates largest area found; d) two FBG strain profiles tracked over time.
In addition to measuring the full-spectrum, the interrogation system needs to track the FBG peak as a function of time. Figure 3-16a shows that a ballistic impact causes the wavelength spectrum to smear. The smearing is caused by the strain across the grating being non-uniform. The non-uniform strain in combination with multiple gratings makes it difficult to track the peak. Optimization through the transfer matrix method \[45],[35] can be used to determine the non-uniform strain. This allows for entire strain profiles along the FBG to be demodulated. However, as discussed in the previous section of this work, this method is prohibitively time consuming.

Under a dynamic impact, many traditional FBG peak tracking methods are no longer applicable due to SNR problems from the peak spread. Sometimes the FBG spectrum is below the noise floor. However, it is still possible to track the peak of an FBG spectrum using an area based tracking approach.

The method for this area based peak tracking approach is shown in Figure 3-17. For each frame, the algorithm finds all continuous data points above a threshold defined as an area. The value for each area is then computed and the \( n \) largest areas corresponding to \( n \) gratings is found and compiled for all frames to build up the wavelength profile over time.

Figure 3-16b illustrates this process with the spectra for 2 FBG at a period corresponding to the red line in a. The area based tracking approach first finds (red) all points at the beginning of an area and (yellow) all points at the end of an area. To ensure the algorithm is usable across datasets, the threshold is set to the mean value of a dataset period. The algorithm then finds the largest area out of all the discovered areas shown from (purple) in c. The lowest wavelength
point of the largest area (orange) which corresponds to the lowest strain on the grating is then used for BFD depth reconstruction.

Figure 3-16d shows that by area tracking all FBG spectra over time, it is possible to obtain strain information over time from multiple FBGs by searching for the $n$ largest areas in the list where $n$ corresponds to the number of FBGs.

![Diagram](image)

Figure 3-17. The area based peak tracking method finds the $n$ largest areas above a threshold for each frame corresponding to $n$ gratings, and then compiles the largest areas over all frames to deduce the wavelength profile over time.

### 3.5 Summary

This work presents a method for analyzing dynamic, localized strain distributions through the use of a single FBG sensor. By using a swept laser source with a sweep repetition rate of 100 kHz, the entire deformation of the FBG spectrum can be captured to analyze strain gradients across the material as a function of position and time. A tensile Hopkinson bar specimen produced strain rates up to $500 \text{ s}^{-1}$ and DIC measurements were used to validate the accuracy of the FBG measurements.
The transfer matrix optimization method is used on the deformed spectrum of the FBG. It is possible to deduce the strain gradient across the FBG, and predict where a specimen will fail or tell where a specimen has failed. This method does not require a full view of the specimen or strain gauges.

The FBG used in this work was surface mounted to the Hopkinson bar specimen, however, FBG’s offer the ability to be both surface mounted or embedded allowing for strain profile analysis without a direct line of sight during high speed events.

Optimization is time prohibitive for large data sets, as a result an area based peak tracking method is introduced in this work. The area based peak tracking method allows for strain from distorted spectra to be demodulated, and is used intensively later in the strain to BFD algorithm in this work.
4 SURVIVABILITY OF FIBER BRAGG GRATINGS UNDER IMPACT CONDITIONS

The fiber Bragg grating optical fiber strain sensor is the most developed optical fiber sensor and can be purchased from a number of vendors. In this work the FBG is sewn onto a Kevlar sensing layer under lower ballistic dynamics and embedded into silicone for higher ballistic dynamics. A common application for FBGs is shape sensing, which is under research from a range of different organizations. This section explains the setup used for ballistics testing with FBGs, and analyzes the survivability of FBGs when the FBG is sewn onto a Kevlar sensing layer and placed between body armor and clay under a ballistic impact. This Chapter discusses how to overcome the necessary challenges to allow for the FBG to survive a ballistic impact.

4.1 Experimental Setup

Figure 4-1 shows the primary gas gun and imaging setup used for impact testing. A compressed gas cylinder is used to shoot either a 12.7 mm diameter 8.24 gram ball bearing or a 124 grain 9 mm round nose full metal jacket (FMJ) round. The projectile speed is measured before impact using photogates, and shots over 400 m/s can be performed. The containment chamber has viewing windows for imaging during impact. Although the standard test backing is clay, a high-speed camera can be used to image through these windows to estimate the actual BFD over time in ballistics gel.
Figure 4-1. a) The gas gun setup shoots 12.7 cm diameter ball bearings using compressed gas cylinders. b) A box with either clay or ballistics gel is mounted into the observation chamber and c) A side view camera can be used to image through the ballistics gel to monitor the BFD over time.

Figure 4-2 shows that in this work, the fiber optic sensor is sewn onto a Kevlar sensing layer. Figure 4-3 shows that the sensing layer is then placed between a shoot pack and the backing material. The standard backing material for ballistics testing is Roma Plastilina No. 1 clay, which is used in the National Institute of Justice (NIJ) body armor testing standards [57]. In some tests in this work the ballistics gel used is a NATO standard (20%) synthetic ballistics gel sourced from Clear Ballistics™. For this work a 50 layer shoot pack was used with the clay
backing material, and a 30 layer Kevlar shoot pack was used with the ballistics gel backing material.

Figure 4-2. The fiber optic sensor is sewn onto a Kevlar sensing layer to dynamically sense body armor BFD.

Figure 4-3. The sensing layer is placed between the shoot pack and the ballistics gel box.
In the tests using ballistics gel as the backing material, the side view of the BFD is recorded using a high-speed camera during the impact to provide a BFD reference for the single and multiple FBG measurement. The high-speed camera is a Photron Fastcam SA-X2, which was used at a frame rate of 100,000 frames per second and a resolution of 384 by 264 pixels.

Figure 4-4 shows how the gray-scale high-speed camera images are converted into binary images. The white pixel with the maximum distance from the bottom edge of each frame is the BFD for each camera frame. The BFD calculated from each camera frame is then mapped to the BFD over time. The image processing method involves several steps for which details can be found in [58].

![High-speed camera imaging](image)

Figure 4-4. High-speed camera imaging allows for BFD tracking over time.

### 4.2 Fiber Survivability under Impact Events

High strength draw tower gratings are used to address the problem in this project that has to do with sensor survivability. High strength draw tower gratings are able to withstand much more strain than traditional FBGs. Typical FBG manufacturing processes compromise the strength of the optical fiber by stripping off the protective jacket which exposes the optical fiber to water ingress and other impurities. The stripping process itself introduces surface defects to the optical fiber which leads to a degradation of the strength of the optical fiber.
To ensure FBG survival, high strength draw tower gratings [59]-[60] from FBGS (Geel, Belgium) are used. These gratings are written into low bend loss optical fiber during the fiber drawing process to ensure that no surface defects are introduced to the optical fiber. The fibers are then coated with an organically modified ceramic (Ormocer) to prevent surface defects and to allow for effective strain transfer from the coating to the sensor. The draw tower gratings have a breaking force around 60 N compared to standard gratings with a breaking force around 10 N [42].

In house experiments were done with ordinary FBGs and drawtower FBGs to compare the tensile strength of the two products. Figure 4-5 shows the setup used for testing the tensile strength of the two types of FBGs. One end of either FBG is superglued to a micron motion stage, and the other end is glued to a fixed mount. A digital caliper is superglued between the two stages to measure the percent strain induced on the FBG. A traditional FBG produced via strip and recoat managed to sustain 1.5% strain before failure. Figure 4-6 shows the test result for a drawtower FBG. At 3.57% the drawtower FBG spectrum went out of the interrogation band but the FBG was still intact. At 6.53% strain the superglue holding down the drawtower FBG gave out, but the FBG was still intact.

The draw tower FBGs from FBGS come in different specifications. To compare the FBG response to different fiber coatings and optical fiber diameters, the different fibers, 125 μm diameter ORMOCER coated fiber, 125 μm diameter ORMOCER-T coated fiber, and 80 μm diameter ORMOCER coated fiber were repeatedly impacted at velocities of 120 m/s to better understand the response of the coatings to impacts. Between each impact, the fibers were removed from the Kevlar sensing layer and photographed using a high depth of field microscope before being reintegrated into the sensing layer. Impacting the fiber at a lower impact velocity...
and photographing the fibers between each impact allowed the development of the damage in the optical fiber coatings to be observed.

Figure 4-5. Fiber tensile strength testing was done by supergluing one end of the fiber Bragg grating onto a micron alignment stage and the other end to a fixed platform.

Figure 4-6. A drawtower FBG went out of the interrogation band at 3.57% strain. At 6.53% strain the superglue holding the FBG gave out.
Table 4-1. Test matrix for the 12 specimens tested where the FBG is vertically offset 1 cm from the impact location and the 6 tests where the FBG is directly impacted by the projectile.

<table>
<thead>
<tr>
<th>Test</th>
<th>Fiber Diameter (μm)</th>
<th>Fiber Coating</th>
<th>Bragg Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1533.93</td>
</tr>
<tr>
<td>2</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1564.08</td>
</tr>
<tr>
<td>3</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1548.62</td>
</tr>
<tr>
<td>4</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1550.74</td>
</tr>
<tr>
<td>5</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1550.77</td>
</tr>
<tr>
<td>6</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1550.38</td>
</tr>
<tr>
<td>7</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1535.53</td>
</tr>
<tr>
<td>8</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1550.45</td>
</tr>
<tr>
<td>9</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1549.84</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1550.74</td>
</tr>
<tr>
<td>11</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1544.72</td>
</tr>
<tr>
<td>12</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1550.59</td>
</tr>
<tr>
<td>13</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1561.00</td>
</tr>
<tr>
<td>14</td>
<td>125</td>
<td>ORMOCER®</td>
<td>1573.80</td>
</tr>
<tr>
<td>15</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1550.77</td>
</tr>
<tr>
<td>16</td>
<td>125</td>
<td>ORMOCER®-T</td>
<td>1550.38</td>
</tr>
<tr>
<td>17</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1549.84</td>
</tr>
<tr>
<td>18</td>
<td>80</td>
<td>ORMOCER®</td>
<td>1550.74</td>
</tr>
</tbody>
</table>

Once the survivability of the coating was observed under low velocity impacts, 18 tests were conducted with projectile velocities of approximately 285 m/s. The 18 tests were separated into two groups: 12 specimens were initially tested with the optical fiber offset from the impact location by approximately 1 cm and 6 specimens were tested where the fiber was not offset from the impact location. After these specimens were tested, an additional six specimens, two of each fiber diameter and coating configuration, were tested where the fiber was not offset from the projectile location. The 12 offset tests contained 4 specimens of each fiber type, and the 6 directly impacted tests consisted of 2 specimens of each type. The 18 tests are outlined in Table 4-1.
4.2.1 Low Velocity Coating Qualification

The 80 μm ORMOCER coated and 125 μm ORMOCER-T coated fibers survived all five impacts, but the 125 μm ORMOCER coated fiber fractured after the third impact. Photographs of the three fibers before they were impacted are shown in Figure 4-7(a-c) and the conclusion of the tests are shown in Figure 4-7(d-f). The images in Figure 4-7(e) and (f) are of the 125 μm

Figure 4-7. Initial images of the (a) ORMOCER coated fiber, (b) ORMOCER-T coated fiber, and (c) 80 μm diameter fibers and final images of the (d) ORMOCER coated fiber, (e) ORMOCER-T coated fiber, and (f) 80 μm fibers before and after being impacted repeatedly with a projectile travelling 120 m/s.
diameter, ORMOCER-T coated fiber and the 80 μm diameter, ORMOCER coated fiber, respectively, after five impacts. Because the 125 μm diameter ORMOCER coated fiber fractured after 3 impacts, the image in Figure 4-7(d) was taken after the third impact. Not only did the ORMOCER coated fiber fracture, but the ORMOCER coating photographed in Figure 4-7(d) shows severe cracking with sections of the coating missing exposing the fiber. Besides the coating fracture, the optical fiber in Figure 4-7(d) looks otherwise undamaged. After five impacts, the ORMOCER coating on the 80 μm diameter fiber was severely cracked. The least amount of damage was observed in the softer ORMOCER-T coated, 125 μm diameter fiber seen in Figure 4-7(e). While surface abrasion is present after five impacts, no large cracks in the coating are observed.

4.2.2 Offset Specimens

Figure 4-8. Strains output by the FBGs written in ORMOCER® coated fiber.
Figure 4-9. Strains output by the FBGs written in ORMOCER®-T coated fiber.

Figure 4-10. Strains output by the FBGs written in 80 μm diameter fiber.

Strains as a function of time measured by the FBGs in 125 micron, Ormocer coated fibers, 125 micron, Ormocer-T coated fibers, and 80 micron diameter fibers are given in Figure 4-8, Figure 4-9 and Figure 4-10 respectively. The negative, compressive strain is unexpected given the general tensile loading of the Kevlar fabric. The strain returning to zero approximately
2 ms after the beginning of the impact event shows that the optical fiber is slipping in the fabric relieving any applied strain.

All 12 of the FBGs offset from the center of the impact survived the impact event. The duration of the impact event is between 1.5 ms and 2 ms. After the impact event, the FBG measured strains for all optical fiber configurations returns to zero. During the impact event, the FBG response, regardless of fiber coating or diameter, exhibits a sudden tensile strain on impact followed immediately by a strain release as the optical fiber slips in the stitch. With the exception of the third 80 μm diameter fiber test, the peak of the initial strain spike is the maximum strain measured by the FBG for a given impact.

Slipping is activated by the force on the fiber reaching the threshold that overcomes static friction between the fiber, the Kevlar, and the stitching. Once this threshold is reached, the movement of the fiber is governed by kinetic friction which has a lower friction coefficient than static friction resulting in less strain transferred from the Kevlar to the fiber. After the initial period of strain relaxation as the optical fiber slips, the FBG measures a second period of tensile strain which is most noticeable in the 80 μm diameter fiber response. The secondary tensile strain region is due to static friction again becoming the dominant force on the optical fiber.

While the 80 μm FBG strain output seen in Figure 4-10 clearly displays this peak, relaxation, and secondary peak behavior for all four tests, the FBGs written in 125 μm diameter fiber do not exhibit this behavior as cleanly. Rather, the FBGs in 125 μm diameter fiber experiences regions of increased compressive strain that are not present in the response of the FBGs written in 80 μm diameter fiber. The FBGs are integrated on the impact side of the sensing layer; therefore, as the sensing layer deforms, the FBG is located on the concave side of the
deformation profile resulting in a compressive bending strain on the FBG. A FBG written in 80 μm diameter fiber is less sensitive to this compressive strain because the FBG is closer to the neutral axis of the bend due to the smaller diameter of the 80 μm fiber.

Figure 4-11. (a) a photograph of the sensing layer with stitched on optical fiber where the black segments indicate the stitch locations and (b) a schematic of the sensing layer showing dimensions and stitch locations.

After the impact, the specimen was removed from the clay block. The actual offset between the center of the impact and the optical fiber, as signified by the labeled offset in Figure
was measured using a pair of calipers, and the deformation profile scanned using the Occipital Structure Sensor.

A photograph of the Roma Plastilina No. 1 clay after the impact event of ORMOCER coated fiber Test 1 is shown in Figure 4-12 along with the cross section of the 3D scan produced by the Structure Sensor. The measured deformation dimensions for the 12 specimens are given in Table 4-2. The deformation depth and width are measured both by hand, using a pair of calipers, and calculated from the profiles given by the 3D scans. The hand measured depths and the depths measured by the 3D scan are less than 1 mm different, and the hand measured depths are presented in Table 4-2. The deformation at the FBG is determined by calculating the depth of the deformation profile measured by the 3D scan at the FBG offset location for each test.

The depth of the deformation at the FBG location, determined from the 3D scans for each test, varied considerably between the different specimens. This variation is common with ballistic testing, but makes direct comparison between the specimens difficult. Therefore, the maximum strain measured by the integrated FBGs across all 12 tests is plotted as a function of deformation depth at the FBG in Figure 4-13. The strain is plotted against the deformation at the FBG location rather than the maximum deformation because the FBG is only able to measure the strain corresponding to the deformation the grating experiences. The measured offsets of the FBGs across the 12 tests is between 6.19 mm and 15.57 mm from the center of the impact. The difference in offset combined with variations in the deformation profile, results in a 14.7 mm variation in the FBG offset depth. The maximum strain measured by the FBGs is fairly constant within each fiber type for deformation depths greater than 15 mm. The test with the smallest maximum strain value, ORMOCER-T Test 1, also had the smallest deformation depth at the FBG, 11.4 mm. Additionally, the strain profile of this test, as seen in Figure 4-9 is dominated by
a compressive strain region through the entire duration of the impact event. The force of the projectile on the sensing layer induces a small tensile strain on the FBG since the FBG is offset by a large, 15.57 mm, distance. However, the compressive strain resulting from the concave bend experienced by the fiber still affects the strain response of the FBG, and is the dominant feature.

Figure 4-12. (a) Photograph of the clay box after impact showing the deformation profile from the ballistic impact with dimensions in cm and (b) a cross section of the 3D scan of the deformation profile.
Figure 4-13 shows that the FBGs written in 80 μm diameter fiber measured higher maximum strains than the FBGs written in 125 μm diameter fiber for similar deformations at the FBG. Since the maximum strain is a function of the initial strain spike on the fiber and because the initial force on the fiber is likely a pure tensile load, the smaller diameter 80 μm fiber is more sensitive to this load than the 125 μm diameter fiber. The coating, ORMOCER or ORMOCER-T, has little effect on the maximum strains measured by the FBG as there is no significant difference in the maximum strains for the two 125 μm diameter fiber types for similar measured deformation depths.

The maximum strain corresponds to the initial strain for 11 of the 12 tests shown. The exception is test 3 of the 80 μm fiber. This test corresponds to the 80 μm data point at the 21.9 mm depth and has the highest strain value plotted in Figure 4-13. The maximum strain value for this test was 2.00%, whereas the magnitude of the initial strain spike is 1.69%. This value of 1.69% strain is consistent with the other 80 μm data points.
Table 4-2. Deformation and fiber offset data for the tested specimens with an offset fiber.

<table>
<thead>
<tr>
<th>Test</th>
<th>Proj. Vel. (m/s)</th>
<th>Def. Depth (mm)</th>
<th>Def. Width (mm)</th>
<th>Meas. Offset (mm)</th>
<th>Def. at FBG (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>285</td>
<td>34.5</td>
<td>35.0</td>
<td>10.77</td>
<td>19.8</td>
</tr>
<tr>
<td>2</td>
<td>282</td>
<td>37.5</td>
<td>38.0</td>
<td>10.08</td>
<td>17.2</td>
</tr>
<tr>
<td>3</td>
<td>273</td>
<td>37.5</td>
<td>37.5</td>
<td>9.38</td>
<td>26.1</td>
</tr>
<tr>
<td>4</td>
<td>278</td>
<td>36.5</td>
<td>38.5</td>
<td>14.82</td>
<td>13.8</td>
</tr>
<tr>
<td>5</td>
<td>285</td>
<td>34.0</td>
<td>37.0</td>
<td>15.57</td>
<td>11.4</td>
</tr>
<tr>
<td>6</td>
<td>285</td>
<td>35.0</td>
<td>38.0</td>
<td>9.58</td>
<td>18.9</td>
</tr>
<tr>
<td>7</td>
<td>281</td>
<td>40.0</td>
<td>37.0</td>
<td>10.97</td>
<td>23.1</td>
</tr>
<tr>
<td>8</td>
<td>269</td>
<td>32.5</td>
<td>37.0</td>
<td>10.95</td>
<td>19.1</td>
</tr>
<tr>
<td>9</td>
<td>285</td>
<td>32.5</td>
<td>35.0</td>
<td>6.19</td>
<td>25.6</td>
</tr>
<tr>
<td>10</td>
<td>285</td>
<td>34.5</td>
<td>34.0</td>
<td>9.71</td>
<td>25.7</td>
</tr>
<tr>
<td>11</td>
<td>282</td>
<td>36.0</td>
<td>37.0</td>
<td>7.20</td>
<td>21.7</td>
</tr>
<tr>
<td>12</td>
<td>273</td>
<td>32.3</td>
<td>38.5</td>
<td>13.36</td>
<td>15.3</td>
</tr>
</tbody>
</table>

4.2.3 Directly Impacted Specimens

Since all 12 of the initial FBGs survived the impact event, a second set of 6 specimens were tested where the optical fibers were integrated along the centerline of the sensing layer with the FBG located at the center of the sensing layer. In this configuration, the FBG and optical fiber was directly impacted by the projectile. The second set of 6 specimens consisted of two specimens each of ORMOCER coated 125 μm diameter fiber, ORMOCER-T coated 125 μm diameter fiber, and ORMOCER coated 80 μm diameter fiber. During the tests, both the specimens with ORMOCER coated FBGs written in 125 μm diameter fiber broke at the impact location at the moment of impact. Given the images of the fiber coatings after repeated low
velocity impacts in Figure 4-7, the fragility of the 125 μm diameter, ORMOCER coated fibers compared to the other fiber types was not surprising. For the second ORMOCER coated, 125 μm diameter fiber specimen, there was a triggering issue which prevented the measuring of projectile velocity. The shot data for the six specimens with directly shot FBGs are shown in Table 4-3.

Table 4-3. Deformation and fiber offset data for the tested specimens with a directly impacted fiber.

<table>
<thead>
<tr>
<th>Test</th>
<th>Proj. Vel. (m/s)</th>
<th>Def. Depth (mm)</th>
<th>Def. Width (mm)</th>
<th>Meas. Offset (mm)</th>
<th>Def. at FBG (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>125 μm ORM.</td>
<td>13 270</td>
<td>26.8</td>
<td>37.0</td>
<td>5.62</td>
<td>23.2</td>
</tr>
<tr>
<td></td>
<td>14 N/A</td>
<td>27.3</td>
<td>38.5</td>
<td>2.63</td>
<td>26.3</td>
</tr>
<tr>
<td>125 μm ORM-T</td>
<td>15 267</td>
<td>27.4</td>
<td>36.8</td>
<td>1.64</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td>16 272</td>
<td>27.0</td>
<td>37.8</td>
<td>0.97</td>
<td>26.9</td>
</tr>
<tr>
<td>80 μm ORM.</td>
<td>17 265</td>
<td>27.7</td>
<td>36.5</td>
<td>4.27</td>
<td>25.8</td>
</tr>
<tr>
<td></td>
<td>18 266</td>
<td>27.2</td>
<td>36.9</td>
<td>4.61</td>
<td>25.5</td>
</tr>
</tbody>
</table>

While both of the ORMOCER-T coated FBGs survived the impact, the center of the impact in ORMOCER-T Test 15 was 10.64 mm horizontally offset from the center of the specimen. The offset depth for ORMOCER-T Test 15 in Table 4-3 is the depth measured at the FBG location 10.64 mm offset horizontally. The grating in Test 15 survived and output data throughout the test; however, because the FBG was not hit directly, the data cannot be directly compared to the other directly impacted FBG cases. The FBG response was unable to be measured in ORMOCER-T Test 16 after 0.0648 ms post-impact due to the optical fiber breaking outside of the specimen due to issues with how the fiber was run out of the test fixture rather than the ballistic impact. Fortunately, the fiber broke after the FBG had measured the peak strain.
The strain responses for the three directly impacted FBGs that survived the impact event are shown in Figure 4-14.

The strain response of ORMOCER-T coated fiber in Test 16 and both 80 μm specimens have a similar response consisting of an initial increase in tensile strain to the maximum value followed by strain relaxation. The strain briefly plateaus for between 0.1 ms and 0.4 ms before gradually relaxing to 0%. Unlike the offset test specimens, none of the FBG responses seen in Figure 4-14 exhibit regions of total compressive strain. Due to the projectile impacting the edge of the FBG in Test 15, the reflected FBG spectrum exhibited significant peak spreading, with FBG spectra being as high as 7 nm in width. This behavior is seen in Figure 4-15. Peak spreading of such magnitude was not observed in any other test. The FBG response in Figure 4-15 shows how more data can be extracted from the full FBG spectrum compared to peak wavelength tracking which would not capture the peak spreading. Additionally, using the full FBG spectrum allows strain data across the length of the FBG to be measured.
Unlike the indirect FBG tests seen in Figure 4-8, Figure 4-9, and Figure 4-10, where highest strains were observed in the 80 μm fiber FBGs, the strain measured by the FBGs written in the ORMOCER-T coated fiber is about 0.5% higher than the strains measured by the FBGs written in 80 μm diameter fiber. Additionally, the maximum strain measured in both ORMOCER-T tests is similar, with the maximum strain measured in test 1 having 88.6% of the value of the maximum strain measured in test 2, even though the deformation depth experienced by the FBG in test 1 is 68% of the deformation depth experienced by the FBG in test 2. This signifies that when the optical fiber is directly impacted, there is localized strain in the fiber within the impact region. This is consistent with current literature on fabric science: that parts of a yarn slip, while parts of a yarn stick under impact conditions. Given this, it is crucial that the entire spectrum of the FBG is captured so that these strain gradients can be discerned along the FBG.
4.3 Survivability at Higher Dynamics

4.3.1 Silicone Sensing Layer

While the Kevlar sensing layer allows for dynamic BFD reconstruction over time, the sensors don’t survive a direct impact at high dynamics. Figure 4-16 shows that at higher dynamics the FBGs are embedded in silicone. This allows for the optical fiber to survive a direct impact when the sensing layer is placed between the body armor and sensing layer. Two different types of silicone were investigated, Smooth-Sil 950 and Sorta-Clear 40, their material properties are summarized in Table 4-4.

The Smooth-Sil tears during impact and is opaque which makes it difficult to identify the sensor location within the sensing layer. The optical fiber embedded in the silicone survives the entire impact event even though the Smooth-Sil tears during impact. However, the tearing silicone layer introduces anomalies into the collected FBG strain data which leads to inconsistent data from the sensing layer, this inconsistent data could lead to large errors in the BFD calculation algorithm. As a result Sorta-Clear silicone is used for body armor shape sensing at higher dynamics.

Table 4-4. Material property summaries for Smooth-Sil 950 and Sorta-Clear 40.

<table>
<thead>
<tr>
<th>Silicone</th>
<th>Modulus (MPA)</th>
<th>Tensile Strength (MPA)</th>
<th>Strain at failure (%)</th>
<th>Die B Tear Strength (pli)</th>
<th>Shore A Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth-Sil 950</td>
<td>1.88</td>
<td>5</td>
<td>320</td>
<td>155</td>
<td>50</td>
</tr>
<tr>
<td>SORTA-Clear 40</td>
<td>0.621</td>
<td>5.52</td>
<td>400</td>
<td>120</td>
<td>40</td>
</tr>
</tbody>
</table>
The effects of different sensing layers on the final BFD are shown in Table 4-5. The Kevlar sensing layer was found to have little impact on the maximum BFD and volume. The 2 mm thick silicone sensing layers (Sorta-Clear 40 and Smooth Sil 950) showed the largest reduction in depth and volume when compared to the baseline. Reducing the silicone thickness to 1 mm minimized the sensing layer’s impact while improving fiber survivability when compared to Kevlar sensing layers.

Several different lubricative coatings were tested to determine if the coating could reduce friction between the silicone and fiber. Reduced friction means that there is a lower force necessary to activate slipping between the fiber and silicone, a lower slip force translates into a lower strain on the optical fiber. Figure 4-17 shows that these different coatings were evaluated by a pullout test apparatus to measure the slipping force, where the pullout forces were measured using a force scale. These pullout tests are summarized in Table 4-6 where all the coatings were found to decrease the pullout force. The polishing wax performed the best; however, the wax required a temperature load to set. Vaseline, which did not require a temperature load was
chosen. Vaseline is easy to apply consistently, readily available and inexpensive, so it was chosen.

Table 4-5. Comparison between depths obtained with and without the different sensing layers.

<table>
<thead>
<tr>
<th></th>
<th>Average Velocity (m/s)</th>
<th>Average Width (mm)</th>
<th>Average Depth (mm)</th>
<th>Average Volume (cm³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Sensing Layer</td>
<td>359.78</td>
<td>47.75</td>
<td>34.73</td>
<td>48.49</td>
</tr>
<tr>
<td>Sorta-Clear 40 (2 mm)</td>
<td>364.69</td>
<td>52.42</td>
<td>32.72</td>
<td>41.47</td>
</tr>
<tr>
<td>Smooth Sil 950 (2 mm)</td>
<td>366.75</td>
<td>55.18</td>
<td>33.42</td>
<td>41.43</td>
</tr>
<tr>
<td>Sorta-Clear 40 (1 mm)</td>
<td>364.42</td>
<td>53.66</td>
<td>34.12</td>
<td>47.05</td>
</tr>
<tr>
<td>Smooth Sil 950 (1 mm)</td>
<td>370.27</td>
<td>54.75</td>
<td>36.78</td>
<td>46.95</td>
</tr>
<tr>
<td>Kevlar</td>
<td>376.32</td>
<td>56.53</td>
<td>34.41</td>
<td>44.51</td>
</tr>
</tbody>
</table>

Figure 4-17. A pullout test was done to determine the best lubricant to be used in conjunction with the silicone sensing layer.
Table 4-6. Pullout test results for measuring the pullout force due to application of different lubricant types.

<table>
<thead>
<tr>
<th></th>
<th>No Coating</th>
<th>SuperLube</th>
<th>TriboGel</th>
<th>Vaseline</th>
<th>Polishing Wax (heated to 90°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pullout Force 1st</td>
<td>22.17</td>
<td>16.09</td>
<td>12.95</td>
<td>7.55</td>
<td>5.30</td>
</tr>
<tr>
<td>Direction (N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pullout force 2nd</td>
<td>11.18</td>
<td>10.30</td>
<td>13.64</td>
<td>6.97</td>
<td>5.30</td>
</tr>
<tr>
<td>Direction (N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percent Difference</td>
<td>65.88</td>
<td>43.87</td>
<td>5.17</td>
<td>8.11</td>
<td>0.00</td>
</tr>
<tr>
<td>(%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Force Reduction 1st</td>
<td>NA</td>
<td>27.43</td>
<td>41.59</td>
<td>65.93</td>
<td>76.11</td>
</tr>
<tr>
<td>Direction (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>Fiber broke during 2nd pull before slipping occurred</td>
<td></td>
<td></td>
<td></td>
<td>heated for 3 minutes with heat gun</td>
</tr>
</tbody>
</table>

Figure 4-18. Strain profiles compared for FBGs embedded in silicone with no Vaseline and FBGs embedded in silicone with Vaseline.
Table 4-7. Test results comparing silicone embedded FBG strain response with and without coating.

<table>
<thead>
<tr>
<th>Test</th>
<th>Vaseline</th>
<th>Projectile Velocity (m/s)</th>
<th>Deformation Depth (mm)</th>
<th>Deformation Width (mm)</th>
<th>Measured Offset (mm)</th>
<th>Maximum Strain (%ε)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No</td>
<td>408</td>
<td>38.9</td>
<td>52.0</td>
<td>4.28</td>
<td>5.92</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>396.9</td>
<td>37.5</td>
<td>55.3</td>
<td>6.32</td>
<td>5.25</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>399.4</td>
<td>30.6</td>
<td>58.7</td>
<td>0.00</td>
<td>5.13</td>
</tr>
</tbody>
</table>

Figure 4-18 and Table 4-7 show the different strain responses for an FBG embedded in silicone with and without vaseline coating. All specimens used are in Sorta-Clear silicone. The presence of vaseline reduced the strain by 13.3 % from 5.92 % to 5.13 %. Something interesting to note with vaseline coated fiber is the faster response time and negative strain region. Overall the vaseline coated fiber was found to exhibit less strain indicating that it would survive an impact event better than a fiber that wasn’t coated, as a result all subsequent tests consist of vaseline coated optical fibers embedded in silicone.

4.4 Summary

This work has shown that when integrated onto a Kevlar sensing layer, FBG sensors are able to survive a ballistic impact event and output justifiable strains given the dynamics of the impact event and how the fibers are integrated into the sensing layer. The strains are reasonable both for FBGs that are offset from the impact location and directly impacted. Strain data was collected at 90 kHz throughout the impact event.
The choice of fiber coating did not significantly affect the strain response of the grating, though it did affect the fiber survivability. The stiffer ORMOCER coating was expected to transfer more strain from the impact to the underlying FBG; however, this was shown not to be the case as the different coatings have little influence on the response of an FBG written in 125 μm diameter fiber. On the other hand, the repeated low velocity ballistic tests showed stiffer ORMOCER coating was more susceptible to fracture and crack development during high strain rate events. The reduced hardiness of the ORMOCER coating was reflected in the increased fragility of the ORMOCER coated fiber to direct high speed impacts. Using a smaller diameter fiber did improve the survivability of the coating and the fiber.

The diameter of the optical fiber also affected FBG strain output in two ways. First, when the fiber was not directly impacted, strains measured by FBGs written in 80 μm were higher than the strains measured by FBGs written in 125 μm diameter fibers; however, when the fiber was directly impacted, the influence of optical fiber diameter decreased and the fiber offset depth was dominant. However, due to their low bending sensitivity, there was also less information present in the full spectrum data output by the 80 μm diameter fibers. All the fibers were integrated into a sensing layer on the impact side of the Kevlar and they experienced a concave curvature during the impact event. This negative strain effects of this concave curvature could not be observed in the 80 μm diameter fiber response.

While the Kevlar sensing layer works well for lower dynamic ballistic tests, the fiber optic cable fails at higher dynamics under a direct impact. Ballistic testing can range from speeds below 400 m/s all the way to 1200 m/s, as a result it is important to fabricate a sensing layer that can survive a wide range of ballistic environments with different types of body armor. The first fiber optic failures were noticed at the Army Test Center (ATC) in Aberdeen, where directly
impacted fibers between the shoot pack and clay failed at the very start of the test. This is because the dynamics at ATC are much higher than the dynamics previously used at NCSU.

A number of changes were made to the NCSU ballistics setup to better mimic the high dynamics under a real ballistics test including that the number of shoot pack Kevlar layers used were reduced, and that full metal jacketed (FMJ) 9 mm rounds are now used that travel approximately 425 m/s. With a decreased number of layers the deformation depth increased from 44.42 mm (50 layers) to 55.99 mm (30 layers), and the deformation width decreased from 63.18 mm (50 layers) to 54.99 mm (30 layers).

The dynamics were tested on FBGs sewn into a Kevlar layer where it was found that the new modified dynamics caused the fibers to fail during a direct impact. This indicated that the new dynamics are a good representative of the dynamics happening at ATC.

Now the optical fibers are embedded in silicone to ensure survivability at these higher dynamics. Two different types of silicone were investigated and a Vaseline layer is added between the optical fiber and silicone to help the fiber stay in kinetic friction longer during the impact event.
5 STRAIN TO BACK FACE DEFORMATION ALGORITHM FOR FIBER BRAGG GRATINGS

This section describes the strain to shape algorithm used for body armor testing. Typical strain to shape algorithms depend on the strain of the optical fiber with respect to a neutral axis. As a result these algorithms rely on strain information when the FBG is held under completely static friction conditions. Unfortunately these methods are difficult to implement at high dynamics such as body armor shape sensing. Efforts to implement such methods onto body armor resulted in failure of the neutral axis reference as well as the FBG. As a result a new shape sensing method based on allowing the fiber to slip is developed in this work. This slip method requires a single FBG to sense the depth of the BFD over time by using boundary conditions on time and depth rather than spatial boundary conditions to a neutral reference. In other words, the new strain to BFD algorithm utilizes the high dynamics of the system under test rather than referencing known spatial coordinates.

5.1 Kinetic Friction Strain to BFD Algorithm Overview

5.1.1 Single Grating Strain to BFD Calculation Algorithm

Figure 5-1 shows that when the projectile impacts the body armor there is a normal force that pushes the optical fiber against the adjacent layers. The static friction causes the optical fiber
to strain as the backing material is deformed. When the tensile force on the stretched optical fiber exceeds the static friction force then the optical fiber starts to slip.

![Figure 5-1](image)

Figure 5-1. (a) Initially the projectile causes a normal force that pushes the optical fiber onto the adjacent layers. (b) The static friction causes the optical fiber to stretch as the backing material deforms. (c) When the force of the stretched optical fiber exceeds the static friction force the optical fiber starts to slip.

When the optical fiber is slipping, Coulomb’s friction law states that the magnitude of the kinetic friction force is independent of the slipping speed. This means that the strain on the optical fiber is entirely dependent on the normal force, which is proportional to the deceleration of the BFD as given by

\[ \varepsilon(t) = C \ a(t) , \]  

(5-1)

where \( \varepsilon \) is the strain on the FBG, \( a(t) \) is the acceleration of the BFD, and \( C \) is an unknown constant relating strain to acceleration.

For Equation (5-1) to be valid, the optical fiber must be slipping, this can be achieved by using the correct sew configuration. Figure 5-2a shows a tight sewn FBG configuration and Figure 5-2b shows a loose sewn Kevlar configuration. The tight sew configuration involves multiple loops around the fiber per sew point. The loose sew configuration consists of a single loop that functions more like a guide for the fiber.
Figure 5-3a shows the strain deduced from a tightly sewn FBG and Figure 5-3b shows the strain deduced from the loosely sewn FBG. Under ideal slipping conditions, the FBG strain profile should resemble the actual deceleration of the BFD. When the FBG is sticking the strain increases and when the FBG is slipping the strain decreases. In the case of the tight sew the FBG noticeably transitions from the slip regime back to the stick regime at various times along the impact.

As a result the slipping condition is not met with the tight sewn FBG resulting in Equation (5-1) only being valid for a portion of the impact. Equation (5-1) yields a 350% displacement prediction error when used with data from a tight sew compared to less than 10% error when used with data from a loose sew. This error is illustrated in Figure 5-4, where the blue line is the deceleration of the projectile deduced from high speed imaging and the red line is the deceleration of the projectile deduced from the FBG strain profile. As can be seen, the deceleration profile from the FBG only resembles the actual deceleration for the initial slip. After multiple stick slip processes, the strain in the fiber no longer accurately represents the deceleration.

If the FBG is loosely sewn then once it starts to slip it stays in the kinetic friction regime. With continuous slipping the unknown constant $C$ is solved using the fact that the final velocity of the BFD is zero and the initial velocity at the start of slip is the impact velocity of the projectile, which is independently measured during the body armor testing. The BFD velocity at time $t$ is given by

$$v(t) = v_o + \int_0^t C \varepsilon(\tau) d\tau,$$

where $v_o$ is the impact velocity of the projectile and $t=0$ at the start of slip. Since the final
velocity is zero, the unknown constant is found to be

\[ C = \frac{v_0}{-\int_{0}^{\infty} \varepsilon(\tau) d\tau} \quad \text{(5-3)} \]

In practice, the integration of the strain is not taken out to infinity but rather to the time of maximum BFD determined by the FBG, this is when the FBG spectrum returns to its original Bragg wavelength value. The BFD is then determined by integrating the velocity resulting in

\[ x(t) = \int_{0}^{t} [v_0 + C \int_{0}^{t} \varepsilon(\tau) d\tau] d\alpha. \quad \text{(5-4)} \]

Figure 5-2. a) Tight sew configuration and b) loose sew configuration in Kevlar.

A loose sew ensures that the FBG survives a ballistic impact by allowing the optical fiber to slip as much as possible during impact, and the loose sew also causes the friction to be dominated by the force between the projectile and backing material rather than by the stitching. This allows for the FBG BFD sensing method to be properly implemented. The MATLAB code for implementing the area based peak tracking algorithm along with this strain to BFD algorithm can be found in Appendix section A.1 in the form of a graphical user interface (GUI).
Figure 5-3. Strain for a) tight sewn specimen under dynamic impact and b) loose sewn specimen under dynamic impact.

Figure 5-4. (blue) Maximum BFD acceleration based off high speed imaging and (red) strain profile deduced from tightly sewn FBG.

5.1.2 Results

To test a single FBG’s ability to measure the BFD over time two different ballistics tests were done. The first test was an impact test shot into clay with a 9 mm FMJ round. This test was done to show that this method is applicable to current NIJ body armor testing standards [57]. The
second test is a ball bearing with clear ballistics gel as the backing material. The displacement over time determined by the FBG is then compared with the actual displacement over time determined by the high-speed imaging.

**Clay Backing**

To show that this BFD measurement method is applicable with existing body armor testing standards the FBG method was performed with a clay backing. For these tests, a 9 mm FMJ round was fired with a speed of 372 m/s into a 50 layer shoot pack placed in front of the sensing layer. For the first time, the dynamics during a shot into clay can be analyzed and discussed.

Figure 5-5 shows the FBG spectrum over time captured from the high-speed interrogator. The impact occurs at time $t=0$ ms when there is a dramatic increase in the FBG reflection peak. The FBG reflection peak then starts to decrease when the optical fiber starts slipping.

We observe that the reflected spectrum broadens once the impact starts and there is not a well-defined Bragg wavelength from which to calculate the strain $\varepsilon$. Therefore, we consider the variation in friction along the FBG length. It is known that the friction between yarns in fabrics cannot simply be defined by Coulomb’s friction law. For a given yarn of the fabric stick-slip modelling is more appropriate to account for the transition between the kinetic and static friction regimes [61].

Portions of the yarn are sticking while portions of the yarn are slipping. When reconstructing the BFD based off this strain consideration, it is important to identify the regions of the FBG that are sticking compared to the regions that are sticking, since both friction regimes yield information that must be interpreted differently.
Initially the force exerted on the optical fiber increases with strain according to Hooke’s law. Whenever the force exceeds the static friction then the section of the FBG slips. Therefore, the section of the FBG that is slipping has less strain than the section that is sticking. If an FBG has sections with different amount of strain then the width of the reflection peak will increase. Figure 5-5 shows the increase in the reflection peak width caused by the stick-slip phenomenon. The lowest wavelength of the reflection peak corresponds to the strain of the slipping portion. Since the algorithm is based off of slipping, the lowest wavelength of the FBG reflection peak is used to determine the strain.

Figure 5-3b shows the strain as a function of time, which is determined by finding the lowest wavelength of the reflection peak, subtracting it from the initial reflection wavelength and then dividing the wavelength shift by the gauge factor as given by

$$\varepsilon(t) = \frac{\lambda(t) - \lambda_B}{M_B t},$$

(5-5)

where $\lambda_B$ is 1536.5 nm in this test.
The proportionality constant for this test is found using Equation (5-3) as $C = -3261 \text{ (m/(s}^2\text{)})$ where $v_o = 372 \text{ m/s}$ is the projectile impact velocity and integration is performed from 0 ms to 0.42 ms. This is determined to be the time to maximum deformation because this is when the lowest $\lambda_B$ returns to its original value. Figure 5-6a shows the resulting BFD acceleration, which is given by $a(t) = -3261 \left( \frac{m}{s^2} \right) \varepsilon(t)$. Figure 5-6b shows the BFD velocity that is determined by integrating Figure 5-6a. Similarly, Figure 5-6c shows the BFD that is determined by integrating Figure 5-6b.

Figure 5-7 shows zoomed in time history of the resulting BFD deflection (i.e. Figure 5-6c). As can be seen, the time to maximum BFD is approximately 0.4 milliseconds. The measured final displacement in clay was 33.4 mm, and the FBG measurement deduced the maximum BFD to be 30.0 mm resulting in a 10% error.

Figure 5-6. (a) (line) BFD acceleration (circles) data points, (b) BFD velocity, and (c) BFD.
Figure 5-7. Displacement over time from a 9 mm FMJ shot into a 50 layer Kevlar shoot pack with clay backing material.

**Ballistics Gel Testing**

To demonstrate the ability of the FBG to correctly deduce the position and timing of the BFD over time, another test was done with the gas gun launching a 12.7 mm diameter, 8.24 gram ball bearing at 260 m/s with a ballistics gel backing material. Ballistics gel is used in this test instead of clay so that the BFD can be independently measured over time with high-speed imaging. Figure 5-8 shows the deduced strain profile over time.

The proportionality constant for this test is $C = -1853.7 \text{ (m/}(s^2\varepsilon))$. Integration is performed from 0 milliseconds to 1.2 milliseconds. Figure 5-9 shows the resulting BFD acceleration, velocity, and BFD.

Figure 5-10 shows the BFD over time determined by high-speed imaging (solid) compared to the BFD over time determined by the FBG (dashed). The final BFD determined by the FBG is 43.99 mm while the actual displacement determined by the high-speed imaging is 41.50 mm, resulting in an error of 5.5%.
Figure 5-8. Strain profile from FBG during the 12.7 mm ball bearing impact into body armor with ballistics gel backing.

Figure 5-9. (a) BFD acceleration, (b) BFD velocity, and (c) BFD.

Figure 5-10 shows that the BFD calculated from the FBG measurement has a faster rise time than the high-speed imaging measurement. Both of the measurements resemble the response of a first order damped system with a step input function, which is consistent with other BFD measurements that have been made with a homogeneous backing material [10]. Therefore,
to estimate the timing error both BFD measurements can be fit to a first order system. Both measurements are normalized and then fit to a first order system given by

\[ x(t) = 1 - e^{-\frac{t}{T}}, \quad (5-6) \]

where \( T \) is the time constant.

---

**Figure 5-10.** Displacement over time determined from the (dashed) FBG and (solid) high-speed imaging.

**Figure 5-11.** Normalized displacement over time determined from the (dashed) FBG and (solid) high-speed imaging.
Figure 5-11 shows the two normalized measurements with the corresponding fit. The resulting time constant for the imaging measurement is $T_{image}=0.66 \text{ ms}$ and the time constant for the FBG method is $T_{FBG}=0.5571 \text{ ms}$. The FBG method has 15.5% error in timing constant when compared against the imaging measurement.

5.2 Full Dynamic Backface Deformation Reconstruction with a Fiber Bragg Grating Array under Kinetic Conditions.

The BFD method for the single grating described in the previous section can be extended to an FBG sensor array to sense the full shape of the BFD over time. The sensor array is shown in Figure 5-12, where all the sensors are multiplexed on the same channel line and each sensor has a different offset distance from the center of impact. The BFD is then reconstructed from the strain at each sensor location by calibrating the sensor array off of the closest FBG since the initial conditions for the closest FBG best estimate the initial velocity of the projectile.

The calibration constant $C$ is deduced from the velocity and strain profile of the FBG closest to the impact location. The displacement for the closest FBG is then calculated off the BFD algorithm for a single FBG, and then $C$ is applied to subsequent gratings to determine their displacement over time. The final shape is reconstructed by interpolating over the displacements from multiple gratings at multiple locations along the sensing layer.

The final velocity for all FBG locations is zero; therefore, the initial speeds at the different grating locations are determined based off their area

$$v_0 = \int_0^t C \epsilon(\tau) \, d\tau,$$  \hspace{1cm} (5-7)
and since the deceleration profiles for the gratings at each location has been related by $C$, the displacement at each location is reconstructed via the BFD algorithm. A spline fit is then used to interpolate between the spatial locations of the displacement points per time frame to reconstruct the whole shape over time.

Figure 5-12. 3 FBGs are multiplexed on the same sensor channel line each with a different offset from the center of impact to reconstruct the full BFD shape over time.

5.2.1 FBG Multiplexing with Peak Crossover

Since full BFD shape reconstruction requires multiple FBGs it is important to have an algorithm that can demodulate a number of FBGs. This problem becomes especially difficult when there are multiple FBGs multiplexed on the same line and their peaks overlap. This is a current research area in fiber Bragg gratings and is called the peak crossover problem [62]. Figure 5-13 shows that this problem involves keeping track of multiple FBGs that are multiplexed on a single line but overlap one another during the measurement process. In other
words, the problem is concerned with correctly detecting and associating multiple FBG peaks when they move through each other.

One advantage of the system is post processing. This allows for all time relevant data to be taken into account rather than simply the data prior to a current frame of reference. As a result the entire data set should be evaluated together to maximize the benefit. The most common method to analyze data from the entire high speed full spectrum interrogation system is with a contour plot and one of the easiest ways to extract data from a contour plot is with image processing.

Figure 5-13. It is difficult to associate peaks when multiple peaks cross over one another.

Image processing can be used to track relevant features from a plot, image gradients, threshold data, etc. For this research the lowest edge of the FBG spectrum is required, therefore the Canny edge tracking algorithm in MATLAB is used. Figure 5-14 shows that edge tracking algorithms locate sharp boundaries on images by turning the largest contrast gradients into data
points, which is ideal for this application since the strain to BFD algorithm makes use of the lowest FBG wavelength.

Figure 5-14. (left) Original image and (right) edge tracked image.

Figure 5-15. (left) Original contour plot from 3 FBG data, (middle) edge tracked image points and (right) scattered edge tracked data points.

Figure 5-15 (left) shows the original contour plot from the 3 FBG data. In this case the wavelength situated closest to the impact has been impacted and therefore crosses over the two other peaks. Figure 5-15 (middle) shows the edge tracked image points from the 3 FBG array represented as a binary line plot and Figure 5-15 (right) shows the edge tracked data points
represented as a scatter plot. Image tracking is able to convert edge information from the FBG contour plot into data points, then a nearest neighbor algorithm can link the points spaced close together to recover the FBG strain profiles and allow for peak crossover to be resolved.

### 5.2.2 Peak Tracking with the Nearest Neighbor Algorithm

Nearest neighbor algorithms allow for the next closest point to a current data point to be identified in a given data set. In the case of the edge tracked data shown in Figure 5-16 (right), this data would correspond to the next closest strain point in time, which is the next closest strain point from the same sensor. Figure 5-16 shows the nearest neighbor algorithm linking data points from the individual sensors over time, where the blue dots are the data points from the sensors and noise, and the red line is the linked strain profile progress.

The start point is specified as the lowest wavelength point of each sensor at the end of the impact event. The strain profile is tracked from the end of the event to the start of the event for accuracy purposes. In other words, there is a large discontinuity at the start of the time of impact which would throw off the nearest neighbor algorithm. The tracked strain profiles can then be used to reconstruct the BFD shape over time.

The code for image tracking and the nearest neighbor algorithm is included in this dissertation in Appendix Section A2.
Figure 5-16. The edge tracked data points are linked with a nearest neighbor algorithm in a reverse order to allow for the estimation algorithm to properly determine the appropriate nearest neighbor.
5.2.3 Results for Full BFD Reconstruction with an FBG array.

The three strain profiles obtained from Figure 5-16 are used to reconstruct the displacement profiles shown in Figure 5-17, where the top graph shows the comparison between the deduced BFD using (dashed) the FBG and (solid) high speed camera imaging. The displacement profile is reconstructed for each sensor which has a spatial location along the sensing layer. These reconstructed displacements compiled at their correct spatial locations then lead to the shapes reconstructed in Figure 5-18. Where the red line is the reconstructed BFD. This reconstructed BFD is overlaid on the corresponding video frames from high speed imaging. The white points represent the FBG locations. There are 3 FBGs located on one side of the BFD, these 3 data points are mirrored about the origin in order to generate 6 data points. Accuracy could potentially be improved by incorporating more FBGs onto the sensing layer.

5.2.4 Summary for Full BFD Reconstruction with an FBG Array

In summary it is possible to use multiple FBGs to reconstruct the full BFD shape over time. Inaccuracies in the final reconstructed shape could be because the algorithm assumes a non-angled displacement into the backing material. In other words, the FBGs could be displacing at an angle rather than straight into the BFD, which would change their spatial locations over time. By using a full shape reconstruction method over time the overall force delivered from the body armor to the wearer over time can be analyzed. The spatial as well as temporal performance of the body armor can also be analyzed.
Figure 5-17. Displacements for each sensor are reconstructed, where (top) is taken directly from Chapter 3 and shows the BFD deduced from (dashed) the FBG compared to the BFD deduced by (solid) high speed imaging. By compiling these displacements spatially over time, it is possible to reconstruct the full BFD shape over time.
Figure 5-18. (red) interpolated BFD shapes from (white) FBG locations on BFD. 3 FBGs are located on one side of the BFD and mirrored to generate 6 data points.
Sewing the FBG onto a Kevlar sensing mat allows for the reconstruction of the BFD over time, but fails at higher dynamics in the sense that a directly impacted FBG will break. As a result it is necessary to develop a sensing layer that better protects the FBG. The next section talks about the development of a silicone sensing layer and modifications to the current shape sensing algorithm to account for nonlinear friction effects.

5.3 Stick-Slip Compensation for Strain to BFD

This section builds on the kinetic friction strain to BFD algorithm in this work to allow for high dynamic BFD sensing with a single fiber Bragg grating (FBG) embedded in silicone. The dynamic BFD of a ballistic impact could be deduced from a single FBG sewn into a Kevlar. However, the sensor fails at high dynamics, as a result the FBG is embedded in a silicone sensing layer for higher dynamic events.

The strain to BFD algorithm showed that the strain induced on the FBG via kinetic friction could be used to deduce the BFD. This was possible because Coulomb’s friction law deduced that the BFD deceleration was proportional to the strain on the FBG, therefore a double integration would yield the BFD.

By embedding the FBG in silicone, the strain induced on an FBG is no longer directly proportional to the deceleration of the BFD. Stick-slip friction resets the strain to acceleration relationship and filtering effects from the silicone render a time scale in the captured data. This work expands on the previous BFD algorithm by compensating for stick-slip effects and filtering effects from the silicone sensing layer.

Embedding the FBG in silicone induces stick-slip friction and filtering effects on the measured strain profile. Therefore, the previously determined strain to BFD algorithm can no
longer be directly applied to the measured strain profile. This section introduces a method to compensate for the stick-slip phenomenon induced on an FBG under a ballistic impact and the filtering effect from the silicone. All tests were done with the same gas gun setup described earlier with a 9 mm FMJ bullet and ballistics gel backing material for high speed camera imaging [58].

5.3.1 Stick Slip Compensation

It was demonstrated that strain induced on the FBG under a ballistic impact is directly related to the deceleration of the BFD over time. Under stick-slip conditions, the static regions no longer allow for such a direct relationship. This is illustrated in Figure 5-19, and Figure 5-20 where (solid) represents data from high speed imaging and (dashed) represents data from the FBG. As can be seen the strain represents the initial deceleration during the initial stick-slip portion, but no longer bears such a direct relationship after the second stick region. This leads to a final displacement error in the reconstructed BFD that is over 100%.

Typically the deflection on a membrane is directly related to the stress on the membrane [63], but there are no fixed boundaries on the silicone mat under the test conditions presented in this work. However, if the slip of the actual sensing layer against the backing material itself is considered negligible for the short period of time that the FBG is sticking, then the strain profile under static friction is directly related to the depth of the BFD. In other words, if it is assumed that all boundary conditions are held constant for the short period of time that the FBG is in the static friction regime and being stretched, then the amount of stretching induced on the FBG is proportional to the deflection of the BFD.
This concept is illustrated on Figure 5-21. With the assumption that the strain on an FBG in the static friction regime is proportional to the deflection of the BFD, and knowing that the strain on an FBG under the kinetic friction regime is proportional to the deceleration of the projectile, then the two strains can be interrelated by the following: since deceleration is the
double derivative of position, the strain from the static regime can be double differentiated to recover the strain from the kinetic regime.

From a) to b) the stick region of the measured strain signal $\varepsilon(t)$ is identified as the rising region. From b) to c) this stick region double differentiated to reconstruct the strain for that region if kinetic friction $F_k$ conditions were observed. This differentiated region is then scaled via a constant $\beta$ to account for the boundary conditions of the FBG under stick conditions

$$\beta \frac{d^2\varepsilon(t)}{dt^2} = F_k,$$

where $\beta$ takes into account the changing boundary conditions of the FBG under stick conditions. In other words for a given FBG strain data set over time, the initial stick portion may not have the same stick points as the second stick portion. $\beta$ is determined such that the scaled section must be monotonic with data before the section. The stick regions often experience large peak spreading and as a result have low SNR. To reduce noise during the differentiation process the static region undergoing double differentiation is fit to an inverse exponential prior to double differentiation. Often times there is more than just a single stick slip region as a result the final static region of the strain profile is first identified, and fit to an inverse exponential with the function

$$Ae^{-Bt} + C,$$

where $A$, $B$, and $C$ are the parameters to be fitted. The region is then double differentiated and scaled until it is monotonic with prior data.

Furthermore, a condition for the strain profile in the kinetic friction regime is that the final data point is equal to 0. In other words, no hysteresis is present in the system. As a result
the data after the current region is then shifted until the final data point is 0, and then scaled down until monotonic with the current region.

The algorithm is summarized in Figure 5-22. This process is then repeated for all static friction regions with the condition that $B$ and $C$ in Equation (5-9) are held constant for subsequent static region fits before the final static region on the same data set. $A$ takes into account the changing boundary conditions and the amount of strain transferred to the grating as a result of the changing boundary conditions. However, in general the timing of the deformation which is represented with $B$ and $C$, should remain the same between the static regions.

To summarize the algorithm process illustrated in Figure 5-21, a) shows the strain profile of an FBG tight sewn onto Kevlar under ballistic impact to induce the stick-slip phenomenon. Where a)->b) shows the identification of the final static region, b)->c) shows the double differentiation and scaling step for the current region. d)->e) shows the down shifting step for the data after the current region and e) -> f) shows the scaling for the data after the current region.

Figure 5-23 and Figure 5-24 shows the result from this reshaping process where (dashed) shows the actual acceleration profile determined from high speed imaging and (solid) shows the acceleration profile and displacement profile determined from reshaping the FBG strain profile. This new reshaped strain profile is then ran through the strain to BFD algorithm. By applying the reshape process to the original stick-slip strain profile the error between the actual and calculated final displacements have been reduced by an order of magnitude.
Figure 5-21. a)→b) identify final stick-regime, b)→c) double differentiate stick regime and scale until monotonic with previous data, d)→e) vertically shift rest of data until final data point has 0 amplitude. e)→f) scale final data portion until graph is monotonic.
Figure 5-22. Process for static friction compensation.

Figure 5-23. (solid) reshaped strain profile and (dashed) actual acceleration profile from high speed camera.
5.3.2 **Timing Compensation in Silicone**

Figure 5-25. (solid) Actual deceleration profile deduced from high speed imaging and (dashed) deduced deceleration profile deduced from reshaped FBG data. a) non-normalized b) normalized.

The double differentiation algorithm works well for a tight sewn FBG in Kevlar because of hard contact interfaces. In other words the spring constants of the surrounding materials can be considered negligible. Figure 5-25 shows that a filtering effect can be observed when the FBG is embedded in 1mm thick silicone where a) shows the non-normalized deceleration plots and b)
shows the normalized deceleration plots. As can be seen the filtering effect from the silicone essentially time scales the actual deceleration profile. As a result, calibrating this filtered strain profile to the initial velocity of the projectile leads to a lower amplitude deceleration profile over time.

Figure 5-26. (dashed) Calculated displacement profile and (solid) actual displacement profile. a) non-normalized displacement profiles, b) normalized displacement profiles, c) non-normalized time scaled displacement profiles and d) normalized time scaled displacement profiles.

Figure 5-26 shows the reconstructed BFD profiles over time where (dashed) shows the BFD profile deduced from the FBG and (solid) shows the BFD profile deduced from the high speed imaging, with a) showing the non-normalized profiles and b) showing the normalized profiles. a) shows that there is a 2% displacement error between the final displacement deduced from high speed imaging and the FBG, and b) shows that there is a 40% 90% rise timing error between the two profiles. c) and d) show that this timing error is alleviated using a simple time
scale on the displacement profile. This calibrated time scale constant can then be applied to subsequent measurements to obtain timing errors of less than 20%. This time scale error can be reduced even further by calibrating the time constant to a series of measurements rather than just a single measurement.

5.3.3 Results

Figure 5-27. a) Comparison between the calculated displacement and the actual displacement for 6 different tests, b) displacement percent error, c) timing percent error, and d) actual timing comparison.

Figure 5-27 shows the results of applying the stick-slip compensation algorithm to the strain profile prior to running the strain profile through the time scaled strain to BFD algorithm. a) and b) show that the tests have less than 10% final displacement error with the exception of 2 tests. One that had an enlarged silicone cavity and the other had the impacted fiber offset by 1 cm, these had final displacement errors less than 20%. c) shows the timing error % for each test.
prior to timing correction, the % errors range from 43% to 54%, and d) shows that the timing error tracks well between tests. The 90% rise time error characterization did not apply well to the 418 m/s test.

In summary this section expands on previous work by introducing methods to compensate for stick-slip and filtering effects on the measured strain profile from the FBG when embedded in silicone. To account for the stick-slip behavior the double derivative of the stick portion is taken and then scaled with a constant. A time scale is then applied to the BFD in the displacement domain to compensate for filtering effects.

The compensated strain profiles were then ran through the strain to BFD algorithm introduced in the previous work. This was demonstrated on 6 different tests in ballistics gel at different speeds to generate less than 10% final displacement error and less than 20% final timing error after compensation.

5.4 Advanced Considerations

A number of efforts have been made in order to characterize the filtering effect that the silicone has on the normal force translation from the projectile to the FBG. Specifically a number of inverse filters including Chebyshev filters, Butterworth filters, and even generic filters via optimization for filter coefficients were investigated to try to obtain a proper inverse filter for the silicone. However, there doesn’t seem to be any direct inverse filter that can properly characterize all the nonlinear effects that the FBG experiences. This makes sense to the extent that the FBG signal is not due to a single measurand i.e. there is different information from both the static friction regime as well as the kinetic friction regime so technically there are 2 filtered signals instead of just one. As a result, this section explains the results obtained when the
nonlinear silicone effects on the FBG are approximated to be linear such that linear noise reduction methods can be used.

A well-known method for reducing linear effects on a sensor or a system such as common mode noise or localized noise is by using a differential/push-pull configuration. This is where the difference between a reference sensor to capture the noise and an actual sensor to capture the noise and the data are used. The signals from the two sensors are then subtracted to obtain the actual signal that was supposed to be measured. This report illustrates this concept on 2 separate tests, each involving 2 FBG sensors.

5.4.1 Test 1

Figure 5-28 shows that there are 2 FBGs for the first test discussed in this section, 1 in the middle of the silicone sensing layer and 1 near the edge of the sensing layer. Figure 5-29 shows that despite this difference in location, both FBGs seem to undergo the same general strain behavior i.e. they stick and slip at approximately the same time. It is assumed that the FBG near the edge of the sensing layer is not involved in the BFD i.e. its BFD displacement is negligible. Therefore it can be assumed that the FBG near the edge of the sensing layer experiences minimal normal force, and that its strain is mainly due to nonlinear effects from the sensing layer i.e. the spring constant of the silicone mat, Striebeck friction from the Vaseline layer, or stick-slip effects etc. As a result, it is assumed that the spring constant of the optical fiber is stiff enough such that any nonlinear effects rendered on the fiber are consistent through the whole fiber.
Figure 5-28. There are 2 FBGs in the sensing mat. One in the center and one at the very edge.

Figure 5-29. (left) original wavelength data, and (right) deduced strain profiles from FBG data.

Coulomb’s law states that

\[ F_k = \mu_k N, \]  

(5-10)

where \( F_k \) is the kinetic friction sliding force, \( \mu_k \) is the kinetic friction coefficient, and \( N \) is the normal force. This relates to the amount of strain placed on an FBG via Hooke’s and Newton’s laws as rewriting (5-10) to be
\[ k\Delta x = \mu_k ma, \]  
(5-11)

where \( m \) is the mass of the BFD, \( k \) is the spring constant of the optical fiber and \( a \) is the deceleration of the projectile. As a result sewing FBGs into Kevlar which rendered the FBGs to stay in kinetic friction allowed for a direct relationship between strain and deceleration.

Approximating the nonlinear effects into a linear term \( C \), then we can approximate the strain induction equation for the FBG in silicone as

\[ k\Delta x = -m_{eff(FBG)}(t)a_{FBG}(t) + C. \]  
(5-12)

Therefore if we assume the nonlinear effects to be the same on both FBGs, then

\[ C_{FBG1} = C_{FBG2}, \]  
(5-13)

and since the FBG2 is near the edge of the impact FBG2 experiences minimal normal force, i.e.

\[ \mu_k m_{eff(bfd2)} a_{bfd2}(t) \approx 0, \]  
(5-14)

then the strain on FBG2 at the edge of the impact zone can be written as

\[ k\Delta x_2 = C_{FBG2}. \]  
(5-15)

Therefore taking the difference between the two strain profiles

\[ k\Delta x_1 - k\Delta x_2 = C_{FBG1} + \mu_k m_{eff(bfd_1)} a_{bfd_1}(t) - (C_{FBG2} + \mu_k m_{eff(bfd_2)} a_{bfd_2}(t)), \]  
(5-16)

gives

\[ k\Delta x_1 - k\Delta x_2 = \mu_k m_{eff(bfd_1)} a_{bfd_1}(t). \]  
(5-17)

Therefore a subtraction of the two strain profiles gives a direct relation to the deceleration of the BFD. Now of course this is not true for the static regions, which is why both strain profiles are
first run through the double differential algorithm to approximate the kinetic friction profile for when the strain profiles are in static strain.

Figure 5-30 shows (blue and red) the two strain profiles deduced from the 2 FBGs and (yellow) the strain profile deduced from taking the absolute difference between the two profiles from the FBGs. Figure 5-31 shows the resulting deceleration comparisons between the camera and the FBG with (left) showing the comparisons normalized and (right) showing the comparisons non-normalized. Taking the absolute difference between the two strain profiles from the two FBG channels yields a much more accurate representation of the actual deceleration over time.

![Graph of strain vs time](image)

Figure 5-30. (blue) strain profile deduced from FBG1, (red) strain profile deduced from FBG2, and (yellow) strain profile deduced from the subtraction between the two profiles.

It should be noted that the double differentiation method is only really an approximation of the kinetic strain over time. This is so because there are not many data points for the regions where the double differentiation method is applied. Even so, it relies on a good number of
assumptions that lead to further inaccuracies. It can also be seen that the difference between the two strain profiles compiles noise and compiles any errors from the double differentiation method. This is potentially another source of error in the final calculated BFD depth.

Figure 5-31. (blue) actual deceleration profile deduced from high speed imaging, (red) deceleration profile deduced from FBG. (left) normalized deceleration profiles (right) non-normalized deceleration profiles.

Figure 5-32 (left) shows a comparison between the FBG deduced displacement profile and the camera deduced displacement profile. There is a 15% final displacement error between the two measurement methods. Looking at the normalized profiles shown in (right), it can be seen that the timing error is almost completely eliminated by using the absolute difference method between the two strain profiles. Figure 5-33 shows the non-normalized and normalized reconstructed BFD over time after only applying the double differential method on a single FBG (the closest one). The displacement error is much less, but the timing error is significant.
Figure 5-32. (blue) Reconstructed displacement determined by FBG and (red) displacement determined by high speed imaging.

Figure 5-33. (left) non normalized displacement comparison for double differentiation method and (right) normalized displacement comparison.

5.4.2 Test 2

Figure 5-34 shows the FBG sensor setup for the next test discussed in this work. For this test 1 FBG was offset from the center of impact by 1.5 cm, while the other FBG was offset by 2.5 cm. Figure 5-35 shows the raw data and the deduced strain profiles from the two tests. As can be seen, again the two FBGs exhibit generally the same strain behavior. So the absolute
difference is taken between the two in order to gain a better estimate of the actual deceleration of
the projectile. Note that both strain profiles are first ran through the double differentiation
algorithm prior to taking the absolute difference between the two.

Figure 5-34. FBG setup for test 2. 1 FBG is located 1.5 cm away from the center, and the other
FBG is located 2.5 cm away from the center on the other side of the silicone mat.

Figure 5-35. (left) contour plot of the original data and (right) comparison of the two strain
profiles over time.
Figure 5-36 shows the non-normalized and normalized displacement profiles between the two calculated displacements after running both strain profiles through the absolute difference method. The final displacement error is about 40%, due to where the two sensors were located. The difference between the two sensors would yield the displacement difference between the two sensors if the calibration were correct. The calibration based off the velocity is incorrect in this case since it relies on one sensor being at the impact location and one sensor being at the edge of the impact location and this leads to large errors.

Figure 5-37 shows the non-normalized and the normalized displacement comparisons to when only the double differentiation method is used on the grating closest to the center of impact. As can be seen the error is smaller, but there is still a timing error of around 50%.

Figure 5-36. Double differential method combined with absolute difference method. (blue) displacement reconstructed from FBG, and (red) displacement determined from high speed imaging. (left) non normalized displacement comparison for absolute difference method and (right) normalized displacement comparison.
Figure 5-37. Double differential method only. (blue) displacement reconstructed from FBG, and (red) displacement determined from high speed imaging. (left) non normalized displacement comparison for double differentiation method and (right) normalized displacement comparison.

5.5 Conclusion:

This work shows that the dynamic BFD of an impact over time can be determined using a single FBG without the need to bond the FBG. The FBG was held to a single layer of woven Kevlar using a loose sew. By allowing the optical fiber to slip it was able to measure a relatively large BFD without breaking. The single FBG measurement technique relies on strain being proportional to the back face acceleration because of the kinetic friction.

This method was tested in ballistics gel along with high-speed imaging to obtain a final displacement error of 5.5% and a timing error of approximately 15.5% when the rise time from the FBG is compared to the actual rise time deduced from high-speed imaging. The FBG was also used to deduce the BFD over time with a clay backing. The time to maximum BFD was determined to be approximately 0.4 milliseconds with a final deduced displacement error of 10%.
These new measurements not only characterize the BFD from a projectile impact test, but also allow the rate of deformation to be analyzed. This would give a better estimate for the lethality of the shot since the amount of kinetic energy delivered to the backing material along with many other parameters can be determined. Errors due to elastic recovery would be expected to be higher, which is why the time dependent measurements are useful.

A single FBG gives insight as to how the BFD depth progresses over time, and this algorithm can be extended to multiple FBGs to determine the full BFD shape over time.

Future work should look into properly characterizing the strain transfer filtering effect from the silicone sensing layer to the optical fiber. If a proper filter is characterized there is no need for time scaling or differentiation. Future work should also look into multiplexing multiple sensor arrays on the silicone sensing layer. This way a 2D shape can be reconstructed with the silicone sensing layer at higher dynamics. Finally, the survivability limit for silicone should be investigated to see what the highest dynamics a silicone sensing layer can withstand. From there, another sensing layer capable of allowing the optical fiber to survive at even higher dynamics may need to be investigated.
6 CONCLUSION

This dissertation shows how to use fiber Bragg gratings coupled with high speed full spectrum interrogation to dynamically shape sense the BFD of body armor. For the first time it is possible to analyze the rate of deformation of body armor, which can potentially be used to reduce blunt force trauma via more efficient body armor designs.

6.1 Contributions

As mentioned in the introduction, the main contributions in this work are outlined as follows:

1. I developed a high-speed full-spectrum interrogation system using commercial components.


2. I helped develop a method to ensure the fiber optic sensor survives a direct impact when placed behind body armor.


3. I developed a back face deformation depth sensing algorithm based on kinetic friction induced on a single FBG.


4. I extended the back face deformation depth sensing algorithm to an array of FBGs enabling the entire back face deformation to be analyzed.

5. I developed a method to compensate for stick-slip friction to allow for back face deformation depth sensing when FBGs are embedded in a silicone sensing layer.

6. (Other Contribution) I have made contributions to slab coupled optical sensor (SCOS) development which allows for optical sensing of electric fields in harsh environments. These contributions can be found in:

Seng, Frederick Alexander, "An Exploration in Fiber Optic Sensors" (2016). All Theses and Dissertations. 6101. https://scholarsarchive.byu.edu/etd/6101


7. I developed dipole antennas for fiber optic electric field sensors which flip the directional sensitivity and enhance the overall sensitivity.
6.2 Strain to BFD Algorithm Based on Kinetic Friction

Research towards dynamic BFD sensing with FBGs has been researched at BYU for 3 years. The basic strain to BFD algorithm developed in this work is based on loosely sewing the FBG onto a Kevlar sensing layer. The sew points should act more of a guide than something to hold down the fiber onto the sensing layer. This dissertation shows that kinetic friction strain information induced on the optical fiber can be directly correlated to the deceleration of the BFD over time. This deceleration can be double integrated to obtain the position of the BFD over time.

6.3 Sensor Array to Shape Algorithm

The rate of the maximum BFD can be determined by a single FBG, but the full dynamic shape over time can be determined by an FBG array. This is done by assuming that all the FBGs in the array have the same strain to deceleration relationship and calibrating this relationship off of the FBG closest to the impact location. Determining the full shape is advantageous in body armor testing because it allows for an analysis of the full force and impact delivered to the wearer via the body armor. Full shape reconstruction is also useful because it allows for a full view of how the body armor reacts to a projectile over time.
6.4 Silicone Dynamics Compensation for the Strain to BFD Algorithm

A large assumption that the strain to BFD algorithm makes is that once the optical fiber enters into the kinetic friction regime, it will stay in the kinetic friction regime. This is not always true for when the optical fiber is sewn into Kevlar, especially at high dynamics; and stick-slip friction transitions are particularly noticeable when the FBG is embedded in Silicone. As a result, this work has derived a method for compensating stick-slip friction. This compensation is based on the relationship that the strain on the FBG represents deceleration of the BFD under kinetic friction conditions and position of the BFD under static friction conditions. As a result the static friction sections of the captured FBG strain profile are double differentiated to obtain an estimate of what the fully kinetic friction strain profile would have looked like.

Another issue with embedding the fibers into silicone are various filtering effects that have not yet been fully investigated. However, by approximating these filtering effects to be linear, then it is possible to reduce these filtering effects by subtracting out signals from an FBG at the center of impact and an FBG at the edge of impact where minimal normal force is experienced.

6.5 Future Work

This dissertation explores how to shape sensethe dynamic BFD of body armor with fiber optic sensors. Although a strain to BFD algorithm has been developed in this work, and although the FBG has been proven to survive at ATC’s base ballistics dynamics, much more work needs to be done to make a robust strain to BFD sensing method with optical fibers. This will involve developing methods to ensure that the FBG survives at even higher dynamics, perhaps past 600
m/s, and potentially even develop a new strain to BFD algorithm such that the algorithm can be implemented in use with hard body armor.

### 6.5.1 Peak Crossover Tracking under High Dynamic Conditions

A popular area of research with FBGs is keeping correct wavelength profile to FBG association when the FBG peaks crossover. This is already difficult enough when the FBG peaks don’t distort as they cross. But the FBG peaks undergo severe smearing under high dynamics testing with FBGs such as body armor testing. As a result it is important to develop a robust method for peak tracking. This work has introduced the concept of using the Canny edge method coupled with a nearest neighbor algorithm to track the lowest wavelength of an FBG, but this method requires heavy tuning from the user. One possible method to overcome this heavy tuning requirement is to develop a good signal estimator, such that the signal estimator could be combined with the nearest neighbor algorithm to determine what data points to link.

Recursive Ransac is a potential algorithm to be used as an estimator. Although this algorithm is typically used to track UAVs, preliminary tests have shown this algorithm to be particularly useful when tracking FBG peaks crossing over granted the peaks do not distort. Now that the Canny image processing method allows for data point extraction from distorted FBG spectra, it could be possible to feed these data points into the recursive RRANSAC algorithm to allow for better lowest wavelength tracking.

Another potential algorithm being developed by Helaman Johnston is the use of function fitting to estimate the data track association. This is done by using a function fit on the wavelength profiles on the numerous FBG spectra and using that function fit to determine association when the FBG peaks crossover.
6.5.2 Improved Optical Sensing of Electric Fields in Harsh Environments

As previously pointed out in my master’s thesis, currently the push-pull SCOS can handle signals in the KV range, but has difficulties handling lower signals. Dipole antennas enhance the sensitivity of the SCOS but at the same time introduce metal into the system under test which could intrude on the electric field being measured. One method is to swap out the filters in the push-pull SCOS interrogation system for two polarization maintaining optical fiber circulators. The configuration should be as follows.

1. Launch 2 lasers into port 1 for both the optical fiber circulators.

2. Hook port 2 of both optical fiber circulators into either end of the push-pull SCOS.

3. Hook port 3 of both optical fiber circulators into photodetectors.

This way signal splitting and filtering is unnecessary, which allows for more power into the optical carrier to potentially provide a much higher signal to noise ratio. The phase modulator should still be used in conjunction with the push-pull configuration with one phase modulator per optical carrier or a single broadband phase modulation scheme for the entire system.

6.5.3 Proper Characterization of Nonlinear Strain Transfer Effects in Silicone

The nonlinear strain transfer effects in silicone are still not fully understood. It is hypothesized with there is Stribeck friction due to the lubricant used and some sort of dampening occurring with the softness of the silicone. Future work should focus on developing a method to characterize the filtering effects of any generic sensing layer material. This way it would be easier to modify the algorithm for higher dynamics without having to develop heuristics.
6.5.4 Body Armor Shape Sensing at Extremely High Dynamics

Ballistics testing can range from less than 400 m/s up to 1200 m/s. To make the entire system fully robust, and to fully be able to test a wide range of body armors, higher dynamics should be explored. This will involve making significant modifications to the current gas gun such that it can fire projectiles at much higher speeds. This will most likely also involve finding a new method for guaranteeing sensor survivability at these higher dynamics.

6.5.5 SCOS Antenna Design with Advanced Antenna Design Concepts

The dipole antenna made for SCOS is a very basic dipole antenna consisting of two metal conductors. It is so basic that the final design is formed using a razor blade. It is possible to refine this antenna design using advanced antenna design concepts and advanced fabrication processes such that more gain from the antenna and a higher extinction ratio between the two sensitive directions is achieved with a lower overall physical area. This way a minimal amount of metal is introduced to the system under test with a large signal boost for the sensor.
REFERENCES


function [] = GUI_4()

% Author: Haderache Menhal / Frederick Seng
% Date: 4/3/2018
SCR = get(0,'Screensize');

% aesthetics
S.fh = figure('units','pixels',...
    'position',[10*SCR(1,3)/1920 10*SCR(1,4)/1100 1200*SCR(1,3)/1920 1200*SCR(1,4)/1200],...
    'menubar','none',...
    'name','HISS',...
    'resize','off',...
    'numbertitle','off',...
    'name','GUI_4');

% axes

%-----------------------------output axes for data-----------------------------
S.ax(1) = axes('units','pixels',...
    'position',[150*SCR(1,3)/1920 300*SCR(1,4)/1200 300*SCR(1,3)/1920 230*SCR(1,4)/1200]);
S.ax(2) = axes('units','pixels',...
    'position',[150*SCR(1,3)/1920 600*SCR(1,4)/1200 300*SCR(1,3)/1920 230*SCR(1,4)/1200]);
S.ax(3) = axes('units','pixels',...
    'position',[600*SCR(1,3)/1920 300*SCR(1,4)/1200 300*SCR(1,3)/1920 230*SCR(1,4)/1200]);
S.ax(4) = axes('units','pixels',...
    'position',[600*SCR(1,3)/1920 600*SCR(1,4)/1200 300*SCR(1,3)/1920 230*SCR(1,4)/1200]);

%----------------------output axes for logos----------------------
S.ax(5) = axes('units','pixels',...
    'position',[700*SCR(1,3)/1920 0*SCR(1,4)/1200 300*SCR(1,3)/1920 230*SCR(1,4)/1200]);
matlabImage1 = imread('BYU_logo.jpg');
image(matlabImage1)
axis offaxis image
S.ax(6) = axes('units','pixels',...
    'position',[935*SCR(1,3)/1920 0*SCR(1,4)/1200 300*SCR(1,3)/1920
    230*SCR(1,4)/1200]);
matlabImage = imread('ncsu-logo.jpg');
image(matlabImage)
axis off
axis image

%----------------------- output axes for general data viewing------------
S.ax(7) = axes('units','pixels',...
    'position',[400*SCR(1,3)/1920 300*SCR(1,4)/1200 750*SCR(1,3)/1920
    575*SCR(1,4)/1200]);
S.ax(8) = axes('units','pixels',...
    'position',[400*SCR(1,3)/1920 300*SCR(1,4)/1200 750*SCR(1,3)/1920
    575*SCR(1,4)/1200]);

%--------------------------- input text boxes and dropdowns -------------
directoryfiles=dir;
directoryfilesstring=char(directoryfiles.name);
directoryfilesstring=directoryfilesstring(3:end,:);
S.ed1 = uicontrol('style','edit',...
    'units','pix',...
    'position',[10*SCR(1,3)/1920 950*SCR(1,4)/1200
    190*SCR(1,3)/1920 20*SCR(1,4)/1200],...
    'min',0,'max',1,...
    'string',['Enter waveform file'],...
    'fontweight','bold',...
    'horizontalalign','center',...
    'fontsize',11*SCR(1,4)/1200);
S.ed2 = uicontrol('style','edit',...
    'units','pix',...
    'position',[210*SCR(1,3)/1920 950*SCR(1,4)/1200
    180*SCR(1,3)/1920 20*SCR(1,4)/1200],...
    'min',0,'max',1,...
    'string',['Enter clock file'],...
    'fontweight','bold',...
    'horizontalalign','center',...
    'fontsize',11*SCR(1,4)/1200);
S.ed3 = uicontrol('style','edit',...
    'units','pix',...
    'position',[400*SCR(1,3)/1920 950*SCR(1,4)/1200
    190*SCR(1,3)/1920 20*SCR(1,4)/1200],...
    'min',0,'max',1,...
    'string',['Enter data valid file'],...
    'fontweight','bold',...
    'horizontalalign','center',...
    'fontsize',11*SCR(1,4)/1200);
S.ed4 = uicontrol('style','edit',...
    'units','pix',...
'position',[10*SCR(1,3)/1920 900*SCR(1,4)/1200]
'190*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...
'string',
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',11*SCR(1,4)/1200);

S.ed5 = uicontrol('style','edit',...
'units','pix',...,...
'position',[210*SCR(1,3)/1920 900*SCR(1,4)/1200]
'220*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...
'string',
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',11*SCR(1,4)/1200);

S.ed6 = uicontrol('style','edit',...
'units','pix',...,...
'position',[440*SCR(1,3)/1920 900*SCR(1,4)/1200]
'220*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...  % This is the key to multiline edits.
'string',
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',11*SCR(1,4)/1200);

S.ed7 = uicontrol('style','edit',...
'units','pix',...,...
'position',[610*SCR(1,3)/1920 950*SCR(1,4)/1200]
'220*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...  % This is the key to multiline edits.
'string',
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',11*SCR(1,4)/1200);

S.ed8 = uicontrol('style','edit',...
'units','pix',...,...
'position',[300*SCR(1,3)/1920 200*SCR(1,4)/1200]
'220*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...  % This is the key to multiline edits.
'string',
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',11*SCR(1,4)/1200);

S.ed9 = uicontrol('style','edit',...
'units','pix',...,...
'position',[300*SCR(1,3)/1920 200*SCR(1,4)/1200]
'220*SCR(1,3)/1920 20*SCR(1,4)/1200],...
'min',0,'max',1,...  % This is the key to multiline edits.
'string',
'fontweight','bold',...
'horizontalalign','center',...
S.ed10 = uicontrol('style','edit','
   'units','pix','
   'position',[10*SCR(1,3)/1920 20*SCR(1,4)/1200
   220*SCR(1,3)/1920 20*SCR(1,4)/1200],
   'min',0,'max',1,... % This is the key to multiline edits.
   'string',['Enter start data point'],...
   'fontweight','bold',...
   'horizontalalign','center',...
   'fontsize',11*SCR(1,4)/1200);

S.ed11 = uicontrol('style','edit','
   'units','pix','
   'position',[10*SCR(1,3)/1920 20*SCR(1,4)/1200
   220*SCR(1,3)/1920 20*SCR(1,4)/1200],
   'min',0,'max',1,... % This is the key to multiline edits.
   'string',['Enter end data point'],...
   'fontweight','bold',...
   'horizontalalign','center',...
   'fontsize',11*SCR(1,4)/1200);

S.ed12 = uicontrol('style','edit','
   'units','pix','
   'position',[10*SCR(1,3)/1920 20*SCR(1,4)/1200
   220*SCR(1,3)/1920 20*SCR(1,4)/1200],
   'min',0,'max',1,... % This is the key to multiline edits.
   'string',['Enter offset data point'],...
   'fontweight','bold',...
   'horizontalalign','center',...
   'fontsize',11*SCR(1,4)/1200);

S.ed13 = uicontrol('style','edit','
   'units','pix','
   'position',[300*SCR(1,3)/1920 20*SCR(1,4)/1200
   220*SCR(1,3)/1920 20*SCR(1,4)/1200],
   'min',0,'max',1,... % This is the key to multiline edits.
   'string',['Enter new end data point'],...
   'fontweight','bold',...
   'horizontalalign','center',...
   'fontsize',11*SCR(1,4)/1200);

%-----------------list current files in folder for data partition function-
S.listfiles = uicontrol('Style', 'pop','
   'String', {directoryfilesstring},...
   'Position', [20*SCR(1,3)/1920 850*SCR(1,4)/1200 200*SCR(1,3)/1920
   100*SCR(1,4)/1200]);

%----------------display notice for calculation procedures ---------------
S.dispcalc = uicontrol('style','text','
   'units','pix','
   'position',[610*SCR(1,3)/1920 800*SCR(1,4)/1200
   220*SCR(1,3)/1920 50*SCR(1,4)/1200],
   'min',0,'max',1,... % This is the key to multiline edits.
   'string',['Calculating....'],...
   'fontweight','bold',...
S.wrongfile=uicontrol('style','text',...
'units','pix',...
'position',[0*SCR(1,3)/1920 500*SCR(1,4)/1200 220*SCR(1,3)/1920 50*SCR(1,4)/1200],...
'min',0,'max',1,... % This is the key to multiline edits.
'string',{'Wrong file type'},...
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',20*SCR(1,4)/1200);

% display choice for functionality.
S.dispchoice = uicontrol('style','text',...
'units','pix',...
'position',[500*SCR(1,3)/1920 1050*SCR(1,4)/1200 220*SCR(1,3)/1920 100*SCR(1,4)/1200],...
'min',0,'max',1,... % This is the key to multiline edits.
'string',{'Choose functionlity'},...
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',20*SCR(1,4)/1200);

% dropdown for user to select file and rename
S.dispchoice = uicontrol('style','text',...
'units','pix',...
'position',[500*SCR(1,3)/1920 1050*SCR(1,4)/1200 220*SCR(1,3)/1920 100*SCR(1,4)/1200],...
'min',0,'max',1,... % This is the key to multiline edits.
'string',{'Choose functionlity'},...
'fontweight','bold',...
'horizontalalign','center',...
'fontsize',20*SCR(1,4)/1200);

% pushbutton options
S.pbsiliconecase = uicontrol('style','push',...
'units','pix',...
'position',[10*SCR(1,3)/1920 1000*SCR(1,4)/1200 380*SCR(1,3)/1920 40*SCR(1,4)/1200],...
'HorizontalAlignment','left',...
'string','BFD algorithm',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.pbsilicone = uicontrol('style','push',...
'units','pix',...
'position',[400*SCR(1,3)/1920 10*SCR(1,4)/1200 380*SCR(1,3)/1920 40*SCR(1,4)/1200],...
'HorizontalAlignment','left',...
'string','Calculate',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.pbpartitiondata = uicontrol('style','push',...
S.pbdatapartition = uicontrol('style','push',...
'units','pix',...
'position',[100*SCR(1,3)/1920 850*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string','Partition data',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.pbdatarename = uicontrol('style','push',...
'units','pix',...
'position',[100*SCR(1,3)/1920 450*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string','Rename data',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.pbcalc = uicontrol('style','push',...
'units','pix',...
'position',[400*SCR(1,3)/1920 10*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string','Calculate',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.pbkevlarcase = uicontrol('style','push',...
'units','pix',...
'position',[10*SCR(1,3)/1920 1000*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string','Kevlar algorithm',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

% silicone algorithm push buttons

S.step1 = uicontrol('style','push',...
'units','pix',...
'position',[100*SCR(1,3)/1920 100*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string','Step1',...
'fontsize',14*SCR(1,4)/1200,'fontweight','bold');

S.step2 = uicontrol('style','push',...
'units','pix',...
'position',[100*SCR(1,3)/1920 300*SCR(1,4)/1200...
'HorizontalAlign','left',...
'string', 'Step2', ...
'fontsize', 14*SCR(1,4)/1200, 'fontweight', 'bold');

S.step3 = uicontrol('style', 'push', ...
'units', 'pix', ...
'position', [100*SCR(1,3)/1920 500*SCR(1,4)/1200 ...
'HorizontalAlign', 'left', ...
'string', 'Step3', ...
'fontsize', 14*SCR(1,4)/1200, 'fontweight', 'bold');

S.step4 = uicontrol('style', 'push', ...
'units', 'pix', ...
'position', [100*SCR(1,3)/1920 700*SCR(1,4)/1200 ...
'HorizontalAlign', 'left', ...
'string', 'go', ...
'fontsize', 14*SCR(1,4)/1200, 'fontweight', 'bold');

S.txtin(1) = uicontrol('style', 'edit', ...
'units', 'pix', ...
'position', [10*SCR(1,3)/1920 500*SCR(1,4)/1200 ...
'min', 0, 'max', 1, ...
'string', 'Rename File', ...
'fontweight', 'bold', ...
'horizontalalign', 'center', ...
'fontsize', 11);

%% initially set everything that should be off to off
set(S.dispcalc, 'visible', 'off');
set(S.txtin(1), 'visible', 'off');
set(S.pbdatapartition, 'visible', 'off');
set(S.pbdatarename, 'visible', 'off');
set(S.ax(1), 'visible', 'off');
set(S.ax(2), 'visible', 'off');
set(S.ax(3), 'visible', 'off');
set(S.ax(4), 'visible', 'off');
set(S.ax(7), 'visible', 'off');
set(S.ax(8), 'visible', 'off');
set(S.ed1, 'visible', 'off');
set(S.ed2, 'visible', 'off');
set(S.ed3, 'visible', 'off');
set(S.ed4, 'visible', 'off');
set(S.ed5, 'visible', 'off');
set(S.ed6, 'visible', 'off');
set(S.ed7, 'visible', 'off');
set(S.ed8, 'visible', 'off');
set(S.ed9, 'visible', 'off');
set(S.ed10, 'visible', 'off');
set(S.ed11, 'visible', 'off');
set(S.ed12, 'visible', 'off');
set(S.ed13, 'visible', 'off');
set(S.pbcalc, 'visible', 'off');
set(S.wrongfile, 'visible', 'off');
function S = varargin{3};

L = get(S.listfiles,{'string'});
L = strsplit(L(1,1));
toparse = get(S.listfiles,{'Value'});
filetoget = strsplit(toparse,1);
charfiletoget = char(filetoget);
stopuntilfound = 0;
ii = 0;
sizecharfiletoget = size(charfiletoget);
sizecharfiletoget = sizecharfiletoget(1,2);
while stopuntilfound == 0
    if(strcmp(charfiletoget(1,sizecharfiletoget-ii),' '))
        ii = ii + 1;
    else
        stopuntilfound = 1;
    end
end
checkfiletype = charfiletoget(sizecharfiletoget-ii-3:sizecharfiletoget-ii);

if(strcmp(checkfiletype,'.bin'))
    renamestring = char(get(S.txtin(1),'string'))
    stopuntilfound = 0;
    ii = 0;
    sizecharfiletoget = size(renamestring);
    sizecharfiletoget = sizecharfiletoget(1,2);
    while stopuntilfound == 0
        if(strcmp(renamestring(1,sizecharfiletoget-ii),' '))
            ii = ii + 1;
        else
            stopuntilfound = 1;
        end
    end
end
checkfiletype=charfiletoget(sizecharfiletoget(ii-3:sizecharfiletoget(ii));
if(strcmp(checkfiletype,'.bin'))
    movefile(charfiletoget,renamestring);
directoryfiles=dir;
directoryfilesstring=char(directoryfiles.name);
directoryfilesstring=directoryfilesstring(3:end,:);
S.listfiles = uicontrol('Style', 'pop', ...
    'String', [directoryfilesstring],...
    'Position', [20 850 200 100]);
set(S.wrongfile,'visible','off');
else
    set(S.wrongfile,'visible','on');
end
else
    set(S.wrongfile,'visible','on');
end

% plot out the selected data when this button is pressed
function[] = pb_calculatedatapart(varargin)
S = varargin{3};
L = get(S.listfiles,{'string'});
L={L{1,1}};
toparse=get(S.listfiles,{'Value'});
filetoget=L{1,1}(cell2mat(toparse),1);
charfiletoget=char(filetoget);

stopuntilfound=0;
ii=0;
sizecharfiletoget=size(charfiletoget);
sizecharfiletoget=size(charfiletoget(1,2));
while stopuntilfound==0
    if(strcmp(charfiletoget(1,sizecharfiletoget(ii)),' '))
        ii=ii+1;
    else
        stopuntilfound=1;
    end
end

checkfiletype=charfiletoget(sizecharfiletoget(ii-3:sizecharfiletoget(ii))
if(strcmp(checkfiletype,'.bin'))
    [t,V] = importAgilentBin(filetoget);
    axes(S.ax(7))
    plot(t,V,'ButtonDownFcn',@linecallbackpartition);
    figure(100)
    plot(t,V,'ButtonDownFcn',@linecallbackpartition);
    xlabel('Time (s)');
    ylabel('Measured Voltage (V)');
    grid on
    set(S.wrongfile,'visible','off');
else
set(S.wrongfile,'visible','on');

end

% plot out data when figure is selected <- functional only on later
% versions of MATLAB
function[]=linecallbackpartition(hObject,~)
figure(100)
plot(hObject.XData,hObject.YData);
xlabel('Time (s)');
ylabel('Measured Voltage (V)');
grid on

% contour data when figure is selected <- functional only on later versions
% of MATLAB.
function[]=linecallbackpartitioncontour(hObject,~)
figure(100)
%plot(hObject.XData,hObject.YData);
contourf(hObject.XData,hObject.YData,hObject.ZData,'edgecolor','none');
xlabel('datapoint');
ylabel('Measured Voltage (V)');
grid on

% function for when the data partition function choice button is pressed
function[]=pb_calldatapart(varargin)
% set up layout
S=varargin{3};
set(S.dispcalc,'visible','off');
set(S.txtin(1),'visible','off');
set(S.pbdatapartition,'visible','off');
set(S.pbdatarename,'visible','off');
cla(S.ax(7),'reset');
set(S.ax(1),'visible','off');
set(S.ax(2),'visible','off');
set(S.ax(3),'visible','off');
set(S.ax(4),'visible','off');
set(S.ax(7),'visible','off');
set(S.ed1,'visible','off');
set(S.ed2,'visible','off');
set(S.ed3,'visible','off');
set(S.ed4,'visible','off');
set(S.ed5,'visible','off');
set(S.ed6,'visible','off');
set(S.ed7,'visible','off');
set(S.ed8,'visible','off');
set(S.ed9,'visible','off');
set(S.ed10,'visible','off');
set(S.ed11,'visible','off');
set(S.ed12,'visible','off');
set(S.ed13,'visible','off');
set(S.pbcalc,'visible','off');
set(S.wrongfile,'visible','off');
set(S.pbsilicone,'visible','off');
set(S.listfiles,'visible','off');
set(S.step1,'visible','off');
set(S.step2,'visible','off');
set(S.step3,'visible','off');
set(S.step4, 'visible', 'off');
set(S.ax(8), 'visible', 'off');
set(S.pbdatapartition, 'visible', 'on');
set(S.listfiles, 'visible', 'on');
set(S.ax(7), 'visible', 'on');
set(S.txtin(1), 'visible', 'on');
set(S.pbdatarenamedata, 'visible', 'on');
set(S.ax(7), 'visible', 'on');

% layout functionality for when silicone button is pressed
function [] = pb_callsilicone(varargin)
S = varargin(3);
set(S.dispcalc, 'visible', 'off');
set(S.txtin(1), 'visible', 'off');
set(S.pbdatapartition, 'visible', 'off');
set(S.pbdatarenamedata, 'visible', 'off');
set(S.ax(1), 'visible', 'off');
set(S.ax(2), 'visible', 'off');
set(S.ax(3), 'visible', 'off');
set(S.ax(4), 'visible', 'off');
set(S.ax(7), 'visible', 'off');
set(S.ed1, 'visible', 'off');
set(S.ed2, 'visible', 'off');
set(S.ed3, 'visible', 'off');
set(S.ed4, 'visible', 'off');
set(S.ed5, 'visible', 'off');
set(S.ed6, 'visible', 'off');
set(S.ed7, 'visible', 'off');
set(S.ed8, 'visible', 'off');
set(S.ed9, 'visible', 'off');
set(S.ed10, 'visible', 'off');
set(S.ed11, 'visible', 'off');
set(S.ed12, 'visible', 'off');
set(S.ed13, 'visible', 'off');
set(S.pbcalc, 'visible', 'off');
set(S.wrongfile, 'visible', 'off');
set(S.pbdatapartition, 'visible', 'off');
set(S.listfiles, 'visible', 'off');
set(S.listfiles, 'visible', 'off');
set(S.step1, 'visible', 'off');
set(S.step2, 'visible', 'off');
set(S.step3, 'visible', 'off');
set(S.step4, 'visible', 'off');
set(S.ed1, 'visible', 'off');
set(S.ed2, 'visible', 'on');
set(S.ed3, 'visible', 'on');
set(S.ed4, 'visible', 'on');
set(S.ed5, 'visible', 'on');
set(S.ed6, 'visible', 'on');
set(S.ed7, 'visible', 'on');
set(S.ed8, 'visible', 'on');
set(S.ed9, 'visible', 'on');
set(S.ed10, 'visible', 'on');
set(S.ed11, 'visible', 'on');
set(S.ed12, 'visible', 'on');
set(S.ed13, 'visible', 'on');
set(S.ed11 , 'visible', 'on');
set(S.ed12 , 'visible', 'on');

% step 1 should read in the files and parameters and then plot out the data
% to determine the start point and end point for record player.

function [] = pb_step1calc(varargin)
S = varargin(3);
    set(S.dispcalc , 'visible', 'on');
[t,V] = importAgilentBin(char(get(S.ed2,'string'))); %start sweep signal
[t1,V1] = importAgilentBin(char(get(S.ed3,'string'))); %data valid waveform
[t2,V2] = importAgilentBin(char(get(S.ed1,'string'))); %waveform

V=V;

%start off slow, see if it works for like 3 repetitions first.
% we first break up everything and load them into a 2x 2 array
ii=1;
jj=0; %this will be the number for counting the number of sample points
in one time frame.
startnewsweep=0;
timeholder=zeros(2,2);
timeholderrow=1; %start off on the first row of the timeholder matrix
startingclock=V(1,1);
updownboolean=0;
% the updown boolean will help us determine what the next threshold we
% should look for is. i.e. shoudl we look for an edge that is greater
% than or less than?
if(startingclock<1.500)
    updownboolean=0;% this is if we are starting on the low clock edge
else
    updownboolean=1;%this is if we are starting on the high clock edge
end
endpoint=size(t);

gettimeperiod=0; %to match time to wavelength, we need to know the sweeep
time

%go through all the points

timeperiodstart=0;
timeperiodend=0;

h = waitbar(0,'Please wait...');
%go through all the points
while gettimeperiod<2
    %while we have not hit a threshold condition keep appending points
    %to the current location in the 2D matrix.
    timeholdercolumn=1; %start off on the first column on the timeholder
column
    startnewsweep=0;

    while (startnewsweep < 0.5) && (ii < endpoint(1,1))
        timeholder(timeholderrow,timeholdercolumn)=V2(ii);

        startnewsweep=startnewsweep+0.5;

        ii=ii+1;
end

delay=hh:mm:ss

while gettimeperiod<2
    %while we have not hit a threshold condition keep appending points
    %to the current location in the 2D matrix.
    timeholdercolumn=1; %start off on the first column on the timeholder
column
    startnewsweep=0;

    while (startnewsweep < 0.5) && (ii < endpoint(1,1))
        timeholder(timeholderrow,timeholdercolumn)=V2(ii);

        startnewsweep=startnewsweep+0.5;

        ii=ii+1;
énd
}
ii=ii+1;

if(updownboolean==0 && V(ii)>1.500)
    updownboolean=1;
    startnewsweep=1;

    if(gettimeperiod==0) %we are ready to start seeing what the
time period is
        gettimeperiod=1;
        timeperiodstart=t(ii);
    else
        if gettimeperiod==1 %otherwise we want to see what the
end of the time period is
            timeperiodend=t(ii);
            gettimeperiod=2;
        end
    end
else
    if(updownboolean==1 && V(ii)<1.500)
        updownboolean=0;
        startnewsweep=1;

        if(gettimeperiod==0) %we are ready to start seeing what the
time period is
            gettimeperiod=1;
            timeperiodstart=t(ii);
        else
            if gettimeperiod==1 %otherwise we want to see what the
end of the time period is
                timeperiodend=t(ii);
                gettimeperiod=2;
            end
        end
end

if gettimeperiod==1
    jj=jj+1;
end

if(V(ii)>1.500)
    V(ii);
end

timeholdercolumn=timeholdercolumn+1;

waitbar(ii / endpoint(1,1))
end

n=1;
timeholderrow=timeholderrow+1;
gettimeperiod
end
jj=jj

timeholderpermenant=zeros(round(endpoint(1,1)/jj),(jj)+1);
errorholderpermenant=zeros(round(endpoint(1,1)/jj),(jj)+1);
alternator=0;
temp=zeros(1,1);
timeholderrow=1;
offsetter=1

if(V(1:ii)<1.500)
    updownboolean=0; % this is if we are starting on the low clock edge
else
    updownboolean=1; % this is if we are starting on the high clock edge
end

while ii<(endpoint(1,1)-jj*2)
    % we have to take care of an inconsistent sample time with the % clock
    if(updownboolean==1)
        if(V(ii+jj,1)>1.500)
            while V(ii+jj,1)>1.500
                ii=ii+1;
            end
        updownboolean=0;
        temp=V2(ii:(ii+jj),1);
        errortemp=V1(ii:(ii+jj),1);
        errorholderpermenant(timeholderrow,:)=transpose(errortemp);
        timeholderpermenant(timeholderrow,:)=transpose(temp);
        else
            while V(ii+jj,1)<1.500
                ii=ii-1;
            end
        updownboolean=0;
        temp=V2(ii:(ii+jj),1);
        errortemp=V1(ii:(ii+jj),1);
        errorholderpermenant(timeholderrow,:)=transpose(errortemp);
        timeholderpermenant(timeholderrow,:)=transpose(temp);
    end
    else if (updownboolean==0)
        if(V(ii+jj,1)>1.500)
            while V(ii+jj,1)>1.500
                ii=ii-1;
            end
        updownboolean=1;
        temp=V2(ii:(ii+jj),1);
        errortemp=V1(ii:(ii+jj),1);
        errorholderpermenant(timeholderrow,:)=transpose(errortemp);
        timeholderpermenant(timeholderrow,:)=transpose(temp);
    else
        while V(ii+jj,1)<1.500
            ii=ii+1;
        end
        updownboolean=1;
        temp=V2(ii:(ii+jj),1);
        errortemp=V1(ii:(ii+jj),1);
        errorholderpermenant(timeholderrow,:)=transpose(errortemp);
    end
end
timeholderpermenant(timeholderrow,:)=transpose(temp);
end
end
end
ii=ii+jj;
timeholderrow=timeholderrow+1;
waitbar(ii / endpoint(1,1))
end
close(h)
% check one sweep
% if samples within clock range are less than the number of points that
% there should be, then use the point range
timeperiod=timeperiodend-timeperiodstart;
% csvwrite('sampletestdatach2.dat',timeholderpermenant);
csvwrite('sampletesterrrorch2.dat',errorholderpermenant);

dlmwrite('sampletestdatach2.dat',timeholderpermenant, 'delimiter', ',', 'precision', 9);
dlmwrite('sampletesterrrorch2.dat',errorholderpermenant, 'delimiter', ',', 'precision', 9);
% moving forward be sure to ignore the first row in timeholder since
% that row is incomplete.
% now convert time to wavelength, the laser sweeps linear with
% frequency.

% error correction portion
segmentdata=csvread('sampletestdatach2.dat'); % read in waveform data
errodata=csvread('sampletesterrrorch2.dat'); % read in error data
sizeofdata=size(segmentdata);

% start with one file, correct its error and see if it works, do this by
% finding where the first threshold is less than 1.5, and then finding the
% next threshold where we are greater than 1.5 interpolate the data between
% these points using pchip, and then replace the data with the pchip data.

ii=1
jj=0
timeholdernew = zeros(0,0);

h = waitbar(0,'Please wait...');
threshold=1;
while ii<=sizeofdata(1,1)
    jj=1;
    while jj<=sizeofdata(1,1)
        if(errordata(ii,jj)<threshold)
            fixed=0;
            numberofpointstointerpolate=0;
            % take the starting point for the pchip interpolation
            interpolatestart=segmentdata(ii,jj);
            % where to start replacement in original data

if(jj>1)
    replacestartpoint=jj;
else
    replacestartpoint=jj;
end

%find the ending point for the pchip interpolation as well as
%the number of points you need to interpolate to make it work

while (errordata(ii,jj)<threshold) && (jj<=sizeofdata(1,2))
    numberofpointstointerpolate=numberofpointstointerpolate+1;
    jj=jj+1;

    if(jj>=sizeofdata(1,2))
        break;
    end
end

%take the ending point for the pchip interpolation
if(jj>=sizeofdata(1,2))
    break;
end
interpolateend=segmentdata(ii,jj);

%where to end replacement in original data
replaceendpoint=jj;

x=1:2;
y=[interpolatestart interpolateend];

xx=linspace(1,numberofpointstointerpolate,numberofpointstointerpolate);
p=pchip(x,y,xx);
k=1;
while replacestartpoint < replaceendpoint
    segmentdata(ii,replacestartpoint)=10000;
    replacestartpoint=replacestartpoint+1;
end

if(jj>=sizeofdata(1,2))
    break;
end
jj=jj+1;
end

waitbar(ii / sizeofdata(1,1))
ii=ii+1;

end

close (h)
finaloutput=zeros(1,1);

ii=1

h = waitbar(0,'Please wait...');
while (ii<sizeofdata(1,1))
    jj=1;
    finaloutputjj=1;
    while (jj<sizeofdata(1,2))
        if(segmentdata(ii,jj)<10000)
            finaloutput(ii,finaloutputjj)=segmentdata(ii,jj);
            finaloutputjj=finaloutputjj+1;
        end
    end
    ii=ii+1;
    jj=1;
end

140
jj=jj+1;
end

while (finaloutputjj<jj)
    %finaloutput(ii,finaloutputjj)=0;
    finaloutputjj=finaloutputjj+1;
end
    waitbar(ii / sizeofdata(1,1))
    ii=ii+1;
end

close (h);
csvwrite('testtwochtwo.dat',finaloutput);
dlmwrite('testtwochtwo.dat',finaloutput, 'delimiter', ',', 'precision', 9);

%% interpolation portion

inputtointerpolate=csvread('testtwochtwo.dat');
sizeinput=size(inputtointerpolate);
targetsize=sizeinput(1,2);
targetoutput=zeros(1,targetsize);

for ii=1:sizeinput(1,1)
    if(sum(abs(inputtointerpolate(ii,:)))==0)
        break
    end
    processing=inputtointerpolate(ii,:);
    idx=find(processing~=0);
    processing=processing(idx);
    sizeprocessing=size(processing);
    sizeprocessing=sizeprocessing(1,2);
    processingxaxis=linspace(0,sizeprocessing,sizeprocessing);
    interpolationaxis=linspace(0,sizeprocessing,targetsize);
    interpolatedprocessing=pchip(processingxaxis,processing,interpolationaxis);
    targetoutput(ii,:)=interpolatedprocessing;
end

dlmwrite('testtwochoneinterpolated.dat',targetoutput, 'delimiter', ',', 'precision', 9);

segmentdata=csvread('testtwochoneinterpolated.dat');
sizeofdata=size(segmentdata);
c=3*10^8;
minwavelength=str2num(char(get(S.ed5,'string')));
maxwavelength=str2num(char(get(S.ed6,'string')));
timeperiod=1;
minfreq=c/(minwavelength*10^(-9));
maxfreq=c/(maxwavelength*10^(-9));
fnew=linspace(minfreq,maxfreq,sizeofdata(1,2));
lambdanew=c./(fnew);
time*sizeofdata(1,1).*timeperiod;
timenew=linspace(0,sizeofdata(1,1)*timeperiod,sizeofdata(1,1));
datapoints=linspace(0,sizeofdata(1,1),sizeofdata(1,1));

%% contour figure to decide where to set your startpoint and endpoint.
% Breakpoint here to decide.

axes(S.ax(7))
contourf(datapoints,lambdanew.*(10^9),transpose(segmentdata),'edgecolor','none');%,'buttonDownFcn',@linecallbackpartitioncontour);
ylim([minwavelength maxwavelength]);
xlabel('datapoint')
ylabel('Wavelength (nm)');

figure(100)
contourf(datapoints,lambdanew.*(10^9),transpose(segmentdata),'edgecolor','none');%,'buttonDownFcn',@linecallbackpartitioncontour);
ylim([minwavelength maxwavelength]);
xlabel('datapoint')
ylabel('Wavelength (nm)');

set(S.dispcalc,'visible','off');

function createnew_fig(cb,evendata)
%cb is the handle of the axes that was clicked
%click on the whitespace within and axes and not on the line object
%copy the axes object to the new figure
hh = copyobj(cb,figure);
%for the new figure assign the ButtonDownFcn to empty
set(hh,'ButtonDownFcn',[]);
%resize the axis to fill the figure
set(hh,'Position',get(0,'DefaultAxesPosition'));

function[] = pb_step2calc(varargin)

S = varargin{3};
set(S.dispcalc,'visible','on');
%set(S.ax(8),'visible','on');
%set(S.ax(7),'visible','off');
segmentdata=csvread('testtwochoneinterpolated.dat');
sizeofdata=size(segmentdata);
c=3*10^8;
startpoint=str2num(char(get(S.ed8,'string'))); %startpoint for record player algorithm.
endpoint = str2num(char(get(S.ed9,'string'))); %endpoint for record player algorithm.
minwavelength=str2num(char(get(S.ed5,'string')));
maxwavelength=str2num(char(get(S.ed6,'string')));
timeperiod=1;
minfreq=c/(minwavelength*10^(-9));
maxfreq=c/(maxwavelength*10^(-9));
fnew=linspace(minfreq,maxfreq,sizeofdata(1,2));
\[
\text{lambdanew} = \frac{c}{(f_{\text{new}})};
\]

\[
\text{firstsearch} = \text{smooth}(\text{segmentdata}(1,:),1); \\
\text{idx} = \text{find(}\text{firstsearch} = \text{max(}\text{firstsearch})\text{);} \\
\text{originalwavelength1} = \text{lambdanew(idx);} \\
\text{positioncounter} = 1; \\
\text{strainmatrix} = \text{zeros}(1,1);
\]

\[
\text{for } \text{ii=startpoint:endpoint} \\
\quad \text{Vs} = \text{smooth}(\text{segmentdata(ii,:),1});
\]

\[
\text{usedvariable} = \text{Vs}'; \\
\text{newwavelength} = \text{find(}\text{usedvariable} = \text{max(}\text{usedvariable})\text{);} \\
\text{loweriterate} = \text{newwavelength}(1,1); \\
\text{higheriterate} = \text{newwavelength}(1,1); \\
\text{hold off}
\]

\[
\% \text{ start at the beginning of the matrix, and then work your way to the end} \\
\text{sizeusedvariable} = \text{size(}\text{usedvariable})\text{;} \\
\text{startplace} = \text{zeros}(2,2); \\
\text{placeholder} = \text{zeros}(1,4); \\
\text{peakfound} = 0; \\
\text{threshold} = \text{mean(}\text{usedvariable})\text{;} \% \text{this is a good value} \\
\]

\[
\% \text{ minimum number of points the record player has to stay above} \\
\% \text{ the threshold in order to be considered an actual spectrum} \\
\text{minnumbpoints} = 10; \\
\text{pointcount} = 0; \\
\text{peakvalid} = 0; \\
\]

\[
\text{pointcounter} = 0; \% \text{this variable will hold how many points the} \\
\text{record player swept across.} \\
\text{tempholder} = \text{zeros}(1,3); \% \text{third place holds number of points} \\
\text{crossed, first and second place hold wavelength values for start and finish.} \\
\text{peaksfound} = 1; \% \text{keeps track of how many peaks have been found} \\
\text{areaplotter}(1,\text{ii}) = \text{sum(}\text{abs(}\text{usedvariable})\text{));}
\]

\[
\% \text{ figure(2)} \\
\% \text{ plot(}\text{usedvariable})
\]

\[
\text{for } \text{jj}=1:\text{sizeusedvariable}(1,2) \\
\quad \% \text{find how many peaks there are and record their values} \\
\quad \text{if(}\text{(}\text{usedvariable}(1,\text{jj}) > \text{threshold}) \text{&& (}\text{peakfound} = 0)) \\
\quad \% \text{start counter.} \\
\quad \text{pointcounter} = 1; \\
\quad \text{tempholder}(\text{peaksfound},1) = \text{jj}; \\
\quad \text{peakfound} = 1;
\]

\[
\text{else}
\]
if((usedvariable(1,jj) < threshold) && (peakfound == 1))
    tempholder(peaksfound,2) = jj;
    tempholder(peaksfound,3) = pointcounter;
    peaksfound = peaksfound + 1;
    peakfound = 0;
else
    if((jj == sizeusedvariable(1,2)) && (peakfound == 1))
        tempholder(peaksfound,2) = jj;
        tempholder(peaksfound,3) = pointcounter;
        peaksfound = peaksfound + 1;
    else
        pointcounter = pointcounter + 1;
end
end

end

% third one holds number of points crossed, i.e. which areas are larger.
sizes = tempholder(:,3);
sizesizes = size(sizes);
firstone = find(max(sizes) == sizes);
sizefirstone = size(firstone);
sizefirstone = sizefirstone(1,1);
if(sizefirstone > 1)
    difffirstone = firstone - previousfirstone;
    idx = find(difffirstone == min(difffirstone));
    firstone = firstone(idx);
end

initialwavelengths = tempholder(:, 1);
finalwavelengths = tempholder(:, 2);

startplace(1, 1) = initialwavelengths(firstone, 1);
startplace(1, 2) = finalwavelengths(firstone, 1);

spreadmatrix(1, positioncounter) = (higheriterate - loweriterate);
lambdahigh = lambdanew(startplace(1, 1));
lambdalow = lambdanew(startplace(1, 2));
lambdaaverage = (lambdahigh);
strainaverage = (lambdaaverage - originalwavelength1) / (1.2 * 10^(-12)) * 10^(-6);
strainaveragematrix(1, positioncounter) = strainaverage;
wavelengthmatrix(1, positioncounter) = lambdaaverage;

positioncounter = positioncounter + 1;
previousfirstone = firstone;
ii;
end

dlmwrite('straindata.dat',strainaveragematrix,'delimiter',',','precision',9);
axes(S.ax(7))
plot(strainaveragematrix,'ButtonDownFcn',@linecallbackpartition);
xlabel('datapoint')
ylabel('strain(\epsilon)');

figure(100)
plot(strainaveragematrix,'ButtonDownFcn',@linecallbackpartition);

timestep=1/(str2num(char(get(S.ed4,'string')))*10^(3));

locationarray=zeros(1,1);
exitloop=0;
positioncounter=1;
% collect static start points and end points as well as the desired
% start calculation point.
while exitloop==0
    uiwait(gcf);
    if(strcmp(char(get(S.ed10,'string')),'done') &&
        strcmp(char(get(S.ed11,'string')),'done'))
        exitloop=1;
    else
        locationarray(positioncounter,1)=str2num(char(get(S.ed10,'string')));
        locationarray(positioncounter,2)=str2num(char(get(S.ed11,'string')));
        positioncounter=positioncounter+1;
    end
    y=1;
end

sizestrainmatrix=size(strainmatrix);
sizelocationarray=size(locationarray);
newtimeaxes=linspace(0,sizestrainmatrix*timestep,sizestrainmatrix*10^(3));
% now double differentiate what the user put in and run through BFD
% algorithm

% if there are no static regions to take into account default to Kevlar
% algorithm.

% first take into account if there is only one final bump to derivate
if((locationarray(1,1)==0) && (locationarray(1,2)==0))
end

class set(S.dispcalc ,'visible','off');
set(S.txtin(1),'visible','off');
set(S.pbdatapartition ,'visible','off');
set(S.pbdatarenome , 'visible','off');
cla(S.ax(7), 'reset');
set(S.ax(1) , 'visible','off');
set(S.ax(2) , 'visible','off');
set(S.ax(3) , 'visible','off');
set(S.ax(4) , 'visible','off');
set(S.ax(7) , 'visible','off');
set(S.ed1 , 'visible','off');
set(S.ed2 , 'visible','off');
set(S.ed3 , 'visible','off');
set(S.ed4 , 'visible','off');
set(S.ed5 , 'visible','off');
set(S.ed6 , 'visible','off');
set(S.ed7 , 'visible','off');
set(S.ed8 , 'visible','off');
set(S.ed9 , 'visible','off');
set(S.ed10 , 'visible','off');
set(S.ed11 , 'visible','off');
set(S.ed12 , 'visible','off');
set(S.ed13 , 'visible','off');
set(S.pbcalc , 'visible','off');
set(S.wrongfile , 'visible','off');
set(S.pbsilicone , 'visible','off');
set(S.listfiles , 'visible','off');
set(S.step1 , 'visible','off');
set(S.step2 , 'visible','off');
set(S.step3 , 'visible','off');
set(S.step4 , 'visible','off');
set(S.ax(1) , 'visible','on');
set(S.ax(2) , 'visible','on');
set(S.ax(3) , 'visible','on');
set(S.ax(4) , 'visible','on');
set(S.ed1 , 'visible','on');
set(S.ed2 , 'visible','on');
set(S.ed3 , 'visible','on');
set(S.ed4 , 'visible','on');
set(S.ed5 , 'visible','on');
set(S.ed6 , 'visible','on');
set(S.ed7 , 'visible','on');
set(S.pbcalc , 'visible','on');

newoffset = str2num(char(get(S.ed12,'string')));
newtimeaxes = newtimeaxes(1,newoffset:end) -
newtimeaxes(1,newoffset);
strainmatrix = strainmatrix(1,newoffset:end);
currentspeed = str2num(char(get(S.ed7,'string')));
%trim strain matrix and time axes based on previous plots
newtimeaxesoriginal=newtimeaxes;

calibrationfactor=currentspeed/sum(strainmatrix);
decelerationmatrix=calibrationfactor.*strainmatrix;

sizestrainmatrix=size(strainmatrix);
sizestrainmatrix=sizestrainmatrix(1,2);

speedmatrix=zeros(1,1);
for pp=1:sizestrainmatrix
    currentspeed=currentspeed-decelerationmatrix(1,pp);
    speedmatrix(1,pp)=currentspeed;
end

positionovertime=cumtrapz(newtimeaxes,speedmatrix);

axes(S.ax(4))
plot(newtimeaxes*1000,positionovertime*1000)
xlabel('time (ms)');
ylabel('displacement (mm)');
grid on

axes(S.ax(3))
plot(newtimeaxes*1000,speedmatrix)
xlabel('time (ms)');
ylabel('speed (m/s)');
grid on

axes(S.ax(2))
plot(newtimeaxes(1,1:end-1)*1000,diff(speedmatrix)./diff(newtimeaxes))
xlabel('time (ms)');
ylabel('deceleration (m/s^2)');
grid on

axes(S.ax(1))
plot(newtimeaxes*1000,strainmatrix)
xlabel('time (ms)');
ylabel('strain (\epsilon)');
grid on

else
    if (sizelocationarray==1)
        startpointfit=locationarray(1,1);
        endpointfit=locationarray(1,2);
    end
end
dlmwrite('timedata.dat',newtimeaxes(1,startpointfit:endpointfit),
'delimiter', ',', 'precision', 9);

dlmwrite('testagainst.dat',smooth(strainmatrix(1,startpointfit:endpointfit),2)
', 'delimiter', ',', 'precision', 9);

x0=[1,10000,0];
palala=fmincon(@logfitter,x0,[],[],[],[],[0 1000000 1000000 1000000 1000000]
); dlwrite('parameters.dat',palala(1,1) palala(1,2) palala(1,3]),
'delimiter', ',', 'precision', 9);

a=palala(1,1);
b=palala(1,2);
c=palala(1,3)

derivativeportion=a.*b.^2.*exp(-c*newtimeaxes(1,startpointfit:endpointfit));

% need automatic method to scale until differentiated section is
monotonic

desiredestimate=(strainmatrix(1,startpointfit-2) -
strainmatrix(1,startpointfit-1))+strainmatrix(1,startpointfit-1);

scalefactor=1;
strainmatrix(1,startpointfit:endpointfit)=(derivativeportion./(10^7)).*scale
factor;

while (strainmatrix(1,startpointfit)>desiredestimate)
strainmatrix(1,startpointfit:endpointfit)=(derivativeportion./(10^7)).*scale
factor;
    scalefactor=scalefactor-0.01;
end

% now need to make sure everything after scale section conforms
% i.e. shift down first

if(strainmatrix(1,end) >0)
    strainmatrix(1,end+1:end)=strainmatrix(1,end+1:end)-
    strainmatrix(1,end);
end

sizestrainmatrix=size(strainmatrix);
sizestrainmatrix=sizestrainmatrix(1,2);
% now check to make sure that everything is scaled down
correctly.
newestimate=(strainmatrix(1,endpointfit-1) -
strainmatrix(1,endpointfit)) +strainmatrix(1,endpointfit);
if(endpointfit<sizestrainmatrix)
scalefactor=1;
finalsectiontemp = strainmatrix(1, endpointfit + 1:end);
while (strainmatrix(1, endpointfit + 1) > newestimate)
    strainmatrix(1, endpointfit + 1:end) = finalsectiontemp .* scalefactor;
    scalefactor = scalefactor - 0.01;
end

% now run through BFD model.
% hide everything except for what you use to show all the data.

set(S.dispcalc, 'visible', 'off');
set(S.txtin(1), 'visible', 'off');
set(S.pbdatapartition, 'visible', 'off');
set(S.pbdataparams, 'visible', 'off');
cla(S.ax(7), 'reset');
set(S.ax(1), 'visible', 'off');
set(S.ax(2), 'visible', 'off');
set(S.ax(3), 'visible', 'off');
set(S.ax(4), 'visible', 'off');
set(S.ax(7), 'visible', 'off');
set(S.ed1, 'visible', 'off');
set(S.ed2, 'visible', 'off');
set(S.ed3, 'visible', 'off');
set(S.ed4, 'visible', 'off');
set(S.ed5, 'visible', 'off');
set(S.ed6, 'visible', 'off');
set(S.ed7, 'visible', 'off');
set(S.ed8, 'visible', 'off');
set(S.ed9, 'visible', 'off');
set(S.ed10, 'visible', 'off');
set(S.ed11, 'visible', 'off');
set(S.ed12, 'visible', 'off');
set(S.ed13, 'visible', 'off');
set(S.pbcalc, 'visible', 'off');
set(S.wrongfile, 'visible', 'off');
set(S.pbsilicone, 'visible', 'off');
set(S.listfiles, 'visible', 'off');
set(S.step1, 'visible', 'off');
set(S.step2, 'visible', 'off');
set(S.step3, 'visible', 'off');
set(S.step4, 'visible', 'off');
set(S.ax(1), 'visible', 'on');
set(S.ax(2), 'visible', 'on');
set(S.ax(3), 'visible', 'on');
set(S.ax(4), 'visible', 'on');
set(S.ed1, 'visible', 'on');
set(S.ed2, 'visible', 'on');
set(S.ed3, 'visible', 'on');
set(S.ed4, 'visible', 'on');
set(S.ed5, 'visible', 'on');
set(S.ed6, 'visible', 'on');
set(S.ed7, 'visible', 'on');
set(S.pbcalc, 'visible', 'on');

newoffset = str2num(char(get(S.ed12, 'string')));
newtimeaxes=newtimeaxes(1,newoffset:end)-
newtimeaxes(1,newoffset);
strainmatrix=strainmatrix(1,newoffset:end);

currentspeed=str2num(char(get(S.ed7,'string')));

%trim strain matrix and time axes based on previous plots
newtimeaxesoriginal=newtimeaxes;
calibrationfactor=currentspeed/sum(strainmatrix);
decelerationmatrix=calibrationfactor.*strainmatrix;
sizestrainmatrix=size(strainmatrix);
sizestrainmatrix=sizestrainmatrix(1,2);
speedmatrix=zeros(1,1);
for pp=1:sizestrainmatrix
    currentspeed=currentspeed-decelerationmatrix(1,pp);
    speedmatrix(1,pp)=currentspeed;
end

positionovertime=cumtrapz(newtimeaxes,speedmatrix);

axes(S.ax(4))
    plot(newtimeaxes*1000,positionovertime*1000)
xlabel('time (ms)');
ylabel('displacement (mm)');
grid on

axes(S.ax(3))
    plot(newtimeaxes*1000,speedmatrix)
xlabel('time (ms)');
ylabel('speed (m/s)');
grid on

axes(S.ax(2))
    plot(newtimeaxes(1,1:end-1)*1000,diff(speedmatrix)./diff(newtimeaxes))
xlabel('time (ms)');
ylabel('deceleration (m/s^2)');
grid on

axes(S.ax(1))
    plot(newtimeaxes*1000,strainmatrix)
xlabel('time (ms)');
ylabel('strain(\epsilon)');
grid on

else
    % if there is more than just one section to reshape, reshape the
    beginning and then loop reshape the rest
    startpointfit=locationarray(1,1);
    endpointfit=locationarray(1,2);
dlmwrite('timedata.dat', newtimeaxes(1, startpointfit:endpointfit), 'delimiter', ',', 'precision', 9);

dlmwrite('testagainst.dat', smooth(strainmatrix(1, startpointfit:endpointfit), 2), 'delimiter', ',', 'precision', 9);

x0=[1, 10000, 0];
palala=fmincon(@logfitter, x0, [], [], [], [], [0 0 0], [1000000 1000000 1000000])
dlmwrite('parameters.dat', [palala(1, 1) palala(1, 2) palala(1, 3)], 'delimiter', ',', 'precision', 9);

a=palala(1, 1);
b=palala(1, 2);
c=palala(1, 3);

derivatedportion=a.*b.^2.*exp(-b*newtimeaxes(1, startpointfit:endpointfit));

% need automatic method to scale until differentiated section is monotonic

desiredestimate=(strainmatrix(1, startpointfit-2) - strainmatrix(1, startpointfit-1)) + strainmatrix(1, startpointfit-1);

scalefactor=1;
strainmatrix(1, startpointfit:endpointfit)=(derivativedportion./(10^7)).*scale
factor;
while (strainmatrix(1, startpointfit)>desiredestimate)
strainmatrix(1, startpointfit:endpointfit)=(derivativedportion./(10^7)).*scale
factor;
scalefactor=scalefactor-0.01;
end

% now need to make sure everything after scale section conforms
% i.e. shift down first

if (strainmatrix(1, end) > 0)
strainmatrix(1, endpointfit+1:end)=strainmatrix(1, endpointfit+1:end) - strainmatrix(1, end);
end

% now check to make sure that everything is scaled down correctly.
newestimate=(strainmatrix(1, endpointfit-1) - strainmatrix(1, endpointfit)) + strainmatrix(1, endpointfit);
sizestrainmatrix=size(strainmatrix);
sizestrainmatrix = size(strainmatrix(1,2));
if(endpointfit < sizestrainmatrix)
    scalefactor = 1;
    finalsectiontemp = strainmatrix(1, endpointfit + 1:end);
    while (strainmatrix(1, endpointfit + 1) > newestimate)
        strainmatrix(1, endpointfit + 1:end) = finalsectiontemp .* scalefactor;
        scalefactor = scalefactor - 0.01;
    end
end

% now loop reshape the rest
for ii = 2:sizelocationarray
    startpointfit = locationarray(ii, 1);
    endpointfit = locationarray(ii, 2);
    dlmwrite('testagainst.dat', smooth(strainmatrix(1, startpointfit:endpointfit), 1)
        ', 'delimiter', ',', 'precision', 9);
    dlmwrite('timedata.dat', newtimeaxes(1, startpointfit:endpointfit),
        'delimiter', ',', 'precision', 9);
    palala = fminunc(@logfitteroffsetfinder, [a, 0])
    derivativedportion = palala(1, 1) .* b.^2 .* exp(-b*newtimeaxes(1, startpointfit:endpointfit));
    desiredestimate = (strainmatrix(1, startpointfit-2) -
        strainmatrix(1, startpointfit-1)) + strainmatrix(1, startpointfit-1);
    scalefactor = 1;
    strainmatrix(1, startpointfit:endpointfit) = (derivativedportion ./ (10^7)). * scale
        factor;
    while (strainmatrix(1, startpointfit) > desiredestimate)
        strainmatrix(1, startpointfit:endpointfit) = (derivativedportion ./ (10^7)). * scale
            factor;
        scalefactor = scalefactor - 0.01;
    end

    newestimate = (strainmatrix(1, endpointfit-1) -
        strainmatrix(1, endpointfit)) + strainmatrix(1, endpointfit);
    scalefactor = 1;
    finalsectiontemp = strainmatrix(1, endpointfit + 1:end);
    while (strainmatrix(1, endpointfit + 1) > newestimate)
        strainmatrix(1, endpointfit + 1:end) = finalsectiontemp .* scalefactor;
        scalefactor = scalefactor - 0.01;
    end
end

set(S.dispcalc, 'visible', 'off');
set(S.txtin(1), 'visible', 'off');
set(S.pbdatapartition, 'visible', 'off');
set(S.pbdatarename, 'visible', 'off');
cla(S.ax(7), 'reset');
set(S.ax(1), 'visible', 'off');
set(S.ax(2), 'visible', 'off');
set(S.ax(3) , 'visible', 'off');
set(S.ax(4) , 'visible', 'off');
set(S.ax(7) , 'visible', 'off');
set(S.ed1 , 'visible', 'off');
set(S.ed2 , 'visible', 'off');
set(S.ed3 , 'visible', 'off');
set(S.ed4 , 'visible', 'off');
set(S.ed5 , 'visible', 'off');
set(S.ed6 , 'visible', 'off');
set(S.ed7 , 'visible', 'off');
set(S.ed8 , 'visible', 'off');
set(S.ed9 , 'visible', 'off');
set(S.ed10 , 'visible', 'off');
set(S.ed11 , 'visible', 'off');
set(S.ed12 , 'visible', 'off');
set(S.ed13 , 'visible', 'off');
set(S.pbcalc , 'visible', 'off');
set(S.wrongfile , 'visible', 'off');
set(S.pbsilicone , 'visible', 'off');
set(S.listfiles , 'visible', 'off');
set(S.step1, 'visible', 'off');
set(S.step2, 'visible', 'off');
set(S.step3, 'visible', 'off');
set(S.step4, 'visible', 'off');
set(S.ax(1) , 'visible', 'on');
set(S.ax(2) , 'visible', 'on');
set(S.ax(3) , 'visible', 'on');
set(S.ax(4) , 'visible', 'on');
set(S.ed1 , 'visible', 'on');
set(S.ed2 , 'visible', 'on');
set(S.ed3 , 'visible', 'on');
set(S.ed4 , 'visible', 'on');
set(S.ed5 , 'visible', 'on');
set(S.ed6 , 'visible', 'on');
set(S.ed7 , 'visible', 'on');
set(S.pbcalc , 'visible', 'on');

currentspeed=str2num(char(get(S.ed7,'string')));
newoffset=str2num(char(get(S.ed12,'string')));
newtimeaxes=newtimeaxes(1,newoffset:end) -
newtimeaxes(1,newoffset);
strainmatrix=strainmatrix(1,newoffset:end);

%trim strain matrix and time axes based on previous plots
newtimeaxesoriginal=newtimeaxes;
calibrationfactor=currentspeed/sum(strainmatrix);
decelerationmatrix=calibrationfactor.*strainmatrix;
sizestrainmatrix=size(strainmatrix);
sizestrainmatrix=sizestrainmatrix(1,2);
speedmatrix=zeros(1,1);
for pp=1:sizestrainmatrix
    currentspeed=currentspeed-decelerationmatrix(1,pp);
speedmatrix(1,pp)=currentspeed;
end

positionovertime=cumtrapz(newtimeaxes,speedmatrix);

axes(S.ax(4))
plot(newtimeaxes*1000,positionovertime*1000)
xlabel('time (ms)');
ylabel('displacement (mm)');
grid on

axes(S.ax(3))
plot(newtimeaxes*1000,speedmatrix)
xlabel('time (ms)');
ylabel('speed (m/s)');
grid on

axes(S.ax(2))
plot(newtimeaxes(1,1:end-1)*1000,diff(speedmatrix)./diff(newtimeaxes))
xlabel('time (ms)');
ylabel('deceleration (m/s^2)');
grid on

axes(S.ax(1))
plot(newtimeaxes*1000,strainmatrix)
xlabel('time (ms)');
ylabel('strain (\epsilon)');
grid on
end
end
dlmwrite('timedata.dat',newtimeaxes(1,startpointfit:endpointfit),
'delimiter', ',', 'precision', 9);
dlmwrite('testagainst.dat',smooth(strainmatrix(1,startpointfit:endpointfit),2)
', 'delimiter', ',', 'precision', 9);
set(S.step4,'visible','off');
APPENDIX A. SECTION A.2 IMAGE PROCESSING AND NEAREST NEIGHBOR ALGORITHM

close all
clear all

segmentdata=csvread('finaloutput.dat');
sizeofdata=size(segmentdata);
c=3*10^8;
minwavelength=1529.57;
maxwavelength=1580.40;
timeperiod=1/(80.775*10^3);
minfreq=c/(minwavelength*10^(-9));
maxfreq=c/(maxwavelength*10^(-9));
fnew=linspace(minfreq,maxfreq,sizeofdata(1,2));
lambdanew=c./(fnew);
time=sizeofdata(1,1).*timeperiod;
timenew=linspace(0,sizeofdata(1,1)*timeperiod,sizeofdata(1,1));
datapoints=linspace(0,sizeofdata(1,1),sizeofdata(1,1));
%segmentdata=mag2db(abs(segmentdata));

%% contour figure to decide where to set your startpoint and endpoint.
Breakpoint here to decide.
h0= figure('PaperUnits', 'inches');
pos = get (h0, 'PaperPosition');
set (h0, 'PaperPosition', [pos(1) pos(2) 3.5 pos(4)/pos(3)*3.5] );
lambdanew=fliplr(lambdanew)
imagesc(datapoints.*timeperiod.*1000,fliplr(lambdanew .*(10^9)),transpose((segmentdata)));
h = gca;  % Handle to currently active axes
set(gca,'YDir','normal')
set(gca,'YTick',[]);
set(gca,'XTick',[]);
set(gca,'position',[0 0 1 1],'units','normalized')
saveas(h0,'testzerochannel2.png');

I = imread('testzerochannel2.png');

%% for corners
corners = detectFASTFeatures(rgb2gray(I),'MinContrast',0.001);
figure(2)
imshow(I); hold on;
plot(corners.selectStrongest(6000));

for regions
regions = detectMSERFeatures(rgb2gray(I));
figure(3)
imshow(I); hold on;
plot(regions,'showPixelList',true,'showEllipses',false);

for HOG features
[featureVector,hogVisualization] = extractHOGFeatures(rgb2gray(I));
figure(4);
imshow(I);
hold on;
plot(hogVisualization);

edge boundary conditions
BW=I;
h0= figure('PaperUnits', 'inches');
pos = get (h0, 'PaperPosition');
set (h0, 'PaperPosition', [pos(1) pos(2) 3.5 pos(4)/pos(3)*3.5] );
BW1 = edge(rgb2gray(I),'Canny',[],sqrt(6));
BW2 = edge(rgb2gray(I),'Prewitt');
imshow(I)
hold on
imshow(BW1);
print('-dpng', 'text.png', '-r300');
saveas(h0,'cannyprocessed.png');

convert to x and y coordinates
sizecanny=size(BW1);
sizecannyx=sizecanny(1,2);
sizecannyy=sizecanny(1,1);
placeholderx=zeros(1,1);
placeholdery=zeros(1,1);

counter=1;
for ii=1:sizecannyy
  for jj=1:sizecannyx
    if(BW1(ii,jj)==1)
      placeholderx(1,counter)=jj;
      placeholdery(1,counter)=ii;
      counter=counter+1
    end
  end
end
end
figure(1)
scatter(placeholderx,placeholdery);

%% find nearest neighbor and iteratively take out points. Start with the left most points.
startpointx=472;
startpointy=308;

%% eliminate all points after x=472
placeholderxtemp=placeholderx;
placeholderytemp=placeholdery;

idxfirsteliminate=find(placeholderx > 472);

placeholderxtemp(idxfirsteliminate)=[];
placeholderytemp(idxfirsteliminate)=[];

figure(1)
scatter(placeholderxtemp,placeholderytemp);
set(gca,'Ydir','reverse')
starttrackx=472;
startracky=308;

placeholdertemp(1,:)=placeholderxtemp;
placeholdertemp(2,:)=placeholderytemp;

[n,d]=knnsearch(placeholdertemp',[472,335],'k',2);
xtrack=zeros(1,1);
ytrack=zeros(1,1);
ypositionadder=0;
yvelocityadder=0;
firsttime=1;
foundx=472;

%% track lowest
writerObj = VideoWriter('lowestwavelength.avi');
writerObj.FrameRate = 10;
open(writerObj);

for ii=1:448
    currentframe=472-ii;
    if(foundx<currentframe)
        currentframe=foundx-1;
        wearehere=1
    end
end
idxfirsteliminate = find(placeholderxtemp > currentframe);

placeholderxtemp(idxfirsteliminate) = [];
placeholderytemp(idxfirsteliminate) = [];

if (firsttime == 1)
    starttrackx = 472;
    startracky = 341;
    firsttime = 0;
else
    starttrackx = foundx;
    startracky = foundy + ypositionadder + yvelocityadder;
end

placeholdertemp = [placeholderxtemp; placeholderytemp];

[n, d] = knnsearch(placeholdertemp', [starttrackx, startracky], 'k', 1, 'NSMethod', 'kdtree');

n = n(1, 1);

foundx = placeholderxtemp(1, n);
foundy = placeholderytemp(1, n);

xtrack(1, ii) = foundx;
ytrack(1, ii) = foundy;

if ii > 2
    ypositionadder = ytrack(1, ii) - ytrack(1, ii - 1);
yvelocityadder = 0;
end

figure(1)
scatter(placeholderxtemp, placeholderytemp);
hold on
plot(xtrack, ytrack, 'r', 'linewidth', 3);
set(gca, 'Ydir', 'reverse')
hold off

frame = getframe(gcf);
writeVideo(writerObj, frame);

ii
end

close(writerObj)

figure(1)
plot(xtrack, ytrack)
dlmwrite('trackeddatalowest.dat',[xtrack; ytrack], 'delimiter', ',', 'precision', 9);

%% track second lowest
firsttime=1
    placeholderxtemp=placeholderx;
    placeholderytemp=placeholdery;
xtrack=zeros(1,1);
ytrack=zeros(1,1);

writerObj = VideoWriter('secondlowestwavelength.avi');
writerObj.FrameRate = 10;
open(writerObj);

for ii=1:460
    currentframe=472-ii;
    if((foundx<currentframe) && (firsttime==0))
        currentframe=foundx-1;
        wearehere=1
    end

    idxfirsteliminate=find(placeholderxtemp > currentframe);
    placeholderxtemp(idxfirsteliminate)=[
    placeholderytemp(idxfirsteliminate)=[

    if(firsttime==1)
        starttrackx=472;
        startracky=282;
        firsttime=0;
    else
        starttrackx=foundx;
        startracky=foundy+ypositionadder+yvelocityadder;
    end

    placeholderxtemp=[placeholderxtemp; placeholderxtemp];
    placeholderytemp=[placeholderytemp; placeholderytemp];
    [n,d]=knnsearch(placeholderxtemp', [starttrackx, startracky], 'k', 1);
    n=n(1,1);

    foundx=placeholderxtemp(1,n);
    foundy=placeholderytemp(1,n);

    xtrack(1,ii)=foundx;
ytrack(1,ii)=foundy;

    if ii>2
        ypositionadder=ytrack(1,ii)-ytrack(1,ii-1);
yvelocityadder=0;
    end
figure(1)
scatter(placeholderxtemp,placeholderytemp);
hold on
plot(xtrack,ytrack,'r','linewidth',3);
set(gca,'Ydir','reverse')
hold off
frame = getframe(gcf);
writeVideo( writerObj,frame);

ii
end

close( writerObj)
dlmwrite('trackeddatasecondlowest.dat',[xtrack; ytrack], 'delimiter', ',', ',', 'precision', 9);

%% track highest

firsttime=1
placeholderxtemp=placeholderx;
placeholderytemp=placeholdery;
xtrack=zeros(1,1);
ytrack=zeros(1,1);

writerObj = VideoWriter('highestwavelength.avi');
writerObj.FrameRate = 10;
open(writerObj);

for ii=1:452

    currentframe=472-ii;

    if((foundx<currentframe) && (firsttime==0))
        currentframe=foundx-1;
        wearehere=1
    end

    idxfirsteliminate=find(placeholderxtemp > currentframe);

    placeholderxtemp(idxfirsteliminate)=[];
    placeholderytemp(idxfirsteliminate)=[];

    if(firsttime==1)
        starttrackx=472;
        starttracky=221;
        firsttime=0;
    else
        starttrackx=foundx;
        starttracky=foundy+ypositionadder+yvelocityadder;
    end

    placeholderxtemp=[placeholderxtemp; placeholderxtemp];
[n,d]=knnsearch(placeholdertemp',[starttrackx,startracky],'k',1);

n=n(1,1);
foundx=placeholderxtemp(1,n);
foundy=placeholderytemp(1,n);

xtrack(1,ii)=foundx;
ytrack(1,ii)=foundy;

if ii>2
    ypositionadder=ytrack(1,ii)-ytrack(1,ii-1);
yvelocityadder=0;
end

figure(1)
    scatter(placeholderxtemp,placeholderytemp);
    hold on
    plot(xtrack,ytrack,'r','linewidth',3);
    set(gca,'Ydir','reverse')
    hold off

    frame = getframe(gcf);
    writeVideo( writerObj,frame);

ii
end

close( writerObj)

dlmwrite('trackeddatahighest.dat',[xtrack; ytrack], 'delimiter', ',', 'precision', 9);