SANDIA REPORT
SAND2022-0187
Printed January 2022

Sandia National Laboratories

Sandia Optical Fringe Analysis Slope Tool (SOFAST) Improvement Effort: Final Report

Braden Smith and Randy C. Brost

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

The Sandia Optical Fringe Analysis Slope Tool (SOFAST) is a tool that has been developed at Sandia to measure the surface slope of concentrating solar power optics. This tool has largely remained of research quality over the past few years. Since SOFAST is important to ongoing tests happening at Sandia as well as an interest to others outside Sandia, there is a desire to bring SOFAST up to professional software standards. The goal of this effort was to make progress in several broad areas including: code quality, sample data collection, and validation and testing. During the course of this effort, much progress was made in these areas. SOFAST is now a much more professional grade tool. There are, however, some areas of improvement that could not be addressed in the timeframe of this work and will be addressed in the continuation of this effort.

ACKNOWLEDGEMENTS

We thank the following individuals for their help with this effort:

- Roger Buck
- Robert Crandell
- Luis Garcia Maldonado
- Micah Mann
- Nathan Schroeder

We thank the DOE Solar Energy Technology Office (SETO) for their financial support.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

CONTENTS

LIST OF FIGURES

This page left blank

EXECUTIVE SUMMARY

The Sandia Optical Fringe Analysis Slope Tool (SOFAST) is a tool for measuring the optical precision of concentrating solar mirrors. It uses deflectometry to obtain a high-resolution map of mirror surface slope, and then uses this to generate a variety of analytic products characterizing optical error. It was produced at Sandia by Chuck Andraka and colleagues circa 2013 [1], [2].

In this project, SOFAST was improved in several broad areas: code quality, sample data collection, and validation and testing. Improvements made in these areas has made SOFAST a more professional code, easier for developers to work with, and a more validated tool.

The quality of the written MATLAB code was greatly improved. In the beginning of this effort, SOFAST was in a form that was not easy to maintain or extend and did not easily allow for validation testing. The code now is modular and is easier for developers to use, extend, and validate.

A large suite of sample data has also been collected as part of this improvement effort. This is a crucial step for the development of any professional code. Having a suite of sample data ensures that the updated SOFAST code is behaving exactly the same as in the beginning of this effort. With automated code tests now written that reference this sample data, the code's processing fidelity can be validated to high accuracy with the click of a button. This provides a solid basis for future code improvement effort, because changes can be made with confidence that core functionality remains correct.

We also made progress in validating the code. We performed several tests to study the system's repeatability and accuracy. Regarding repeatability, we learned that projector output variation must be managed to achieve high repeatability. More testing is needed in the future to fully validate this tool.

ACRONYMS AND DEFINITIONS

Abbreviation	Definition
SOFAST	Sandia optical fringe analysis slope tool
GUI	Graphical user interface
NSTTF	National solar thermal test facility
BCS	Beam characterization system
AIMFAST	Alignment implementation for manufacturing using fringe analysis slope technique

1. INTRODUCTION

The Sandia Optical Fringe Analysis Slope Tool (SOFAST) is a tool for measuring the optical precision of concentrating solar mirrors. It uses deflectometry to obtain a high-resolution map of mirror surface slope, and then uses this to generate a variety of analytic products characterizing optical error. It was produced at Sandia by Chuck Andraka and colleagues [1], [2].

In this project, SOFAST was improved in several broad areas: code quality, sample data collection, and validation and testing. Improvements made in these areas has made SOFAST a more professional code, easier for developers to work with, and a more validated tool.

The initial state of SOFAST at the beginning of this effort is referred to SOFAST 1.0, while the improved version of SOFAST produce by this effort is referred to as SOFAST 2.0.

The physical setup consists of a camera that images the reflection of a screen seen in the mirror facet under test. A projector displays images on the screen, and the reflections of the projected images are captured by the camera after being distorted by the mirror. SOFAST processes these images and calculates a high-resolution surface slope map. This surface slope map is processed to calculate focal length, RMS slope error, and other statistics of interest.

SOFAST is written in MATLAB and was developed at Sandia National Laboratories. During its development phase, SOFAST was kept as a research grade tool, which suited the National Solar Thermal Test Facility's (NSTTF) needs at the time. The goal of the SOFAST improvement effort is to bring the code up to professional software engineering standards, making it easier for others to use or license the code, and easier to extend to incorporate new functionality. The following sections describe areas of improvement pursued under this project. In some cases, not all desired improvements were able to be completed and are thus listed as future work.

2. AREA 1: CODE QUALITY

2.1. Initial state of SOFAST 1.0

Written Code: The SOFAST 1.0 MATLAB code was of research quality at the beginning of this effort. The code sometimes lacked inline comments and documentation where needed. The code package lacked modularity; the main code file was a monolithic 11,000-line script that called some sub-functions. It was not possible to call a single function of SOFAST, which is crucial for code testing and validation. Data passed from function-to-function within the script also lacked modularity; all data was contained in the MATLAB graphical user interface (GUI) handles variable. Although this was convenient for fast data transfer between functions, there was little organization of the data fields within the data structure, making development difficult. The code suite also had only been fully tested with an old version of MATLAB, version 2014a, a legacy version of MATLAB that most users do not have installed.

Version control: There was no formal form of version control for SOFAST 1.0. Often the comments in the MATLAB code functioned as a type of documentation of previous versions, where old versions of the code were commented out and replaced with new code. This did provide a record of changes to the code, but left the code cluttered.

Utility for high volume CSP mirror fabrication: SOFAST 1.0 existed as a GUI, with many of the functions necessary for data collection/processing baked into the GUI script. In this form, SOFAST could not be run as a scripted code; the user was required to physically click buttons on the GUI. This type of code organization prohibited the code from being automated to run on large batches of data or from running autonomously.

System Setup: The setup and calibration procedure for SOFAST is very time and labor intensive. An extensive calibration process is required for the camera and projector systems. Documentation does exist for this procedure, but it lacks any figures and is, in some cases, hard for new users to follow.

2.2. Improvements Made

Written Code: The SOFAST MATLAB code has undergone significant improvements. Inline comments have been added and they now act as a form of documentation as to what processing steps are being performed. Variable names, where appropriate, have been changed from arbitrary to descriptive names to further help developers understand what the code is doing.

The internal structure of the code was also improved. All internal processing steps are now contained in separate MATLAB functions. SOFAST now exists as a MATLAB class, which only calls functions individually as needed. Another improvement was made to the internal structure of the data variables. Within the MATLAB class, data variables are now separated into six MATLAB structures with the following descriptive names: Physical Setup, Facet, Camera, Measurements, Calculations, and Miscellaneous. Each MATLAB structure is then populated with descriptive field names. This better separates the data within SOFAST, making it cleaner and easier for developers to access. This new code structure also allows for better validation testing to occur during code development to ensure the way the code processes data remains the same.

Progress was also made in making SOFAST compatible with the most up-to-date version of MATLAB. The analysis code (this is all the code that doesn't rely on communicating with the cameras) was developed and thoroughly tested on MATLAB 2019b. The analysis code has also been tested on MATLAB version 2021b with success, but a thorough validation of using MATLAB 2021b has not been performed yet.

Version Control: Old comments have been removed that previously documented what code changes have been made by Sandia developers. These comments have been replaced with a professional form of version control, GitLab. GitLab is a code repository system that automatically records all changes made to the code by developers. Code updates are developed on GitLab "branches." When development on a branch is finalized, the branch is "merged" with the "master branch," which contains the most up-to-date version of the code. This merging process is well controlled, and can only happen after being reviewed by the project's designated maintainers. The entire history of the code's development is stored in its repository. Users can clone the code repository to their computers or review old versions of the code using GitLab's user interface.

Utility for high volume CSP mirror fabrication: SOFAST 2.0 is now a fully scriptable code. In other words, the user is not required to click buttons on the GUI, but can be run as either a MATLAB command line program, or can be run as an automatic script. In the process of converting SOFAST to a scriptable code, SOFAST naturally became more modular. Each processing block is not contained in its own function.

System Setup: The instructions documenting the setup and calibration of SOFAST have been significantly improved. The instructions are easier to follow, more verbose and contain many figures of the calibration process. The audience is also more encouraged to copy NSTTF's hardware setup exactly, instead of using custom hardware that has not been tested by NSTTF.

2.3. Future Work

Currently, SOFAST is a MATLAB tool. This requires the user to purchase a MATLAB license as well as two processing toolboxes. This and inherent MATLAB language limitations drive a desire to convert all of SOFAST to Python (an open source programming language). Python is not only attractive because it is free and open source, but it also has more sophisticated capability to implement complex data structures and is regarded as a more agile and functional programming language. Converting SOFAST to Python would significantly improve its extensibility, and also make it more compatible with other codes in the Sandia Concentrating Solar Optics Laboratory.

SOFAST 2.0 currently has no working GUI. This is potentially a drawback for users that do not have the expertise or time to run SOFAST as a command line tool. In the future, a GUI will be built to interface with SOFAST 2.0. This improvement may come after SOFAST is converted to Python, since MATLAB and Python GUIs are not translatable.

The setup/calibration sequence currently takes several hours with multiple people. It is highly desirable to shorten the setup time for SOFAST. Specifically, the projector distortion calibration takes the most time and resources. In the future, we plan to shorten this step by moving from a manual method of measuring image distortion to a photogrammetric method.

It is also desirable for SOFAST to be validated on the current version of MATLAB. Future work will pursue testing all functions, including the camera control functions, with the current release of MATLAB.

3. AREA 2: SAMPLE DATA COLLECTION

Validation data is an important part of software packages. A validation dataset is a set of measurement data and accompanying processed output data created by running the input measurement data through the analysis software. Validation data is necessary to have during code development both to verify correct functionality, and then ensure that correct functionality is maintained as the code is modified and extended.

3.1. Initial State of SOFAST 1.0

SOFAST 1.0, being of research quality, did not contain a curated set of sample data. SOFAST was informally tested and validated during its development, but we did not find any saved validation datasets. SOFAST 1.0 collected data on many test mirror facets; the results of characterizing a facet after code development were likely compared "by eye" to previous results to ensure consistency.

3.2. Improvements Made

Sample data was an area of great improvement for SOFAST 2.0. A set of sample data was collected by inserting lines of code into a clone of the SOFAST 1.0 MATLAB script. These lines of code were strategically placed in the internal processing functions of SOFAST to access all intermediate steps' inputs and outputs. These intermediate values for a given SOFAST run were then automatically saved to a data file. This modified version of SOFAST 1.0 was run to characterize a heliostat facet under different conditions, and with different processing options enabled. The goal was to exercise as much of the code as possible and to flush out as many potential bugs as possible.

An extensive suite of unit tests and integration tests were also written as part of SOFAST 2.0 development using MATLAB's built-in unit test functionality. Unit tests are scripts that are used to validate that a specific function is behaving as expected. Generally, a unit test will load pre-calculated input parameters from the collection of sample data, run one single function using those input parameters, and then finally compare the function's outputs to the expected outputs stored in the sample data collection. If the calculated outputs and expected outputs are different, this is communicated to the user. Integration tests are similar to unit tests, but instead of testing one single function at a time, an integration test will test a string of functions as they are "integrated" together. The SOFAST 2.0 sample data testing suite contains both unit tests and integration tests. Using MATLAB's built-in unit test functionality, unit tests were created to test each of SOFAST's processing steps individually. Integration tests were also written to test the code as a whole. All tests can be run with the click of a button, and a report of any failing functions is generated. If any bugs are introduced to SOFAST 2.0 during development, the offending functions will show up in MATLAB's unit test report. This is a crucial step in the improvement of SOFAST as it provides a way to check that development of the code has not changed the way SOFAST characterizes heliostat facets.

3.3. Future Work

Future work in this area includes growing the collection of sample data. There are processing options relating to refining the characterization of the mirror facet that have not been exercised in the current set of sample data. Ideally, all of the processing options of SOFAST 1.0 will be tested so that code testing touches all lines of the code.

4. AREA 3: VALIDATION AND TESTING

4.1. Initial State of SOFAST 1.0

The absolute accuracy of SOFAST 1.0 has not been thoroughly tested, although there has been some work in estimating the relative accuracy of SOFAST measurements [3]. This was in part because the intermediate processing steps of SOFAST were not available, so the source of any inaccuracies could not be pinpointed. This was also due to the lack of an available reference standard. Ideally, the surface slopes of a reference standard, as measured by SOFAST, would be compared to the known surface slope of the reference standard. Then the source of any errors would be determined by analyzing the inputs and outputs of intermediate processing steps.

4.2. Improvements Made

Validation and testing is another area of SOFAST that has seen great improvement. Several tests have been performed on SOFAST 2.0 to characterize its accuracy and repeatability. The details of these tests are listed below.

4.2.1. Frame averaging test

This test characterized the amount at which the noise in the camera (read noise, shot noise, dark current shot noise, etc.) affected the result of a SOFAST measurement (reported focal length, RMS error, etc.). This test was performed by capturing 25 SOFAST measurements of a single stationary mirror. Then, 25 more measurements of the same stationary mirror were captured but each single raw camera image used for internal calculations was actually five images averaged together. The theory was that averaging five images together would reduce the amount of noise in the image and give a more stable focal length measurement.

The results, shown in Figure 1, were not as expected. The widths of the distributions of reported focal lengths were not smaller for the five frame average cases. In fact, the y focal length distribution was larger in the five frame average case. This led us to conclude that sensor noise was not the largest driver of measurement error and variability. This led us to conduct the second test, described below.

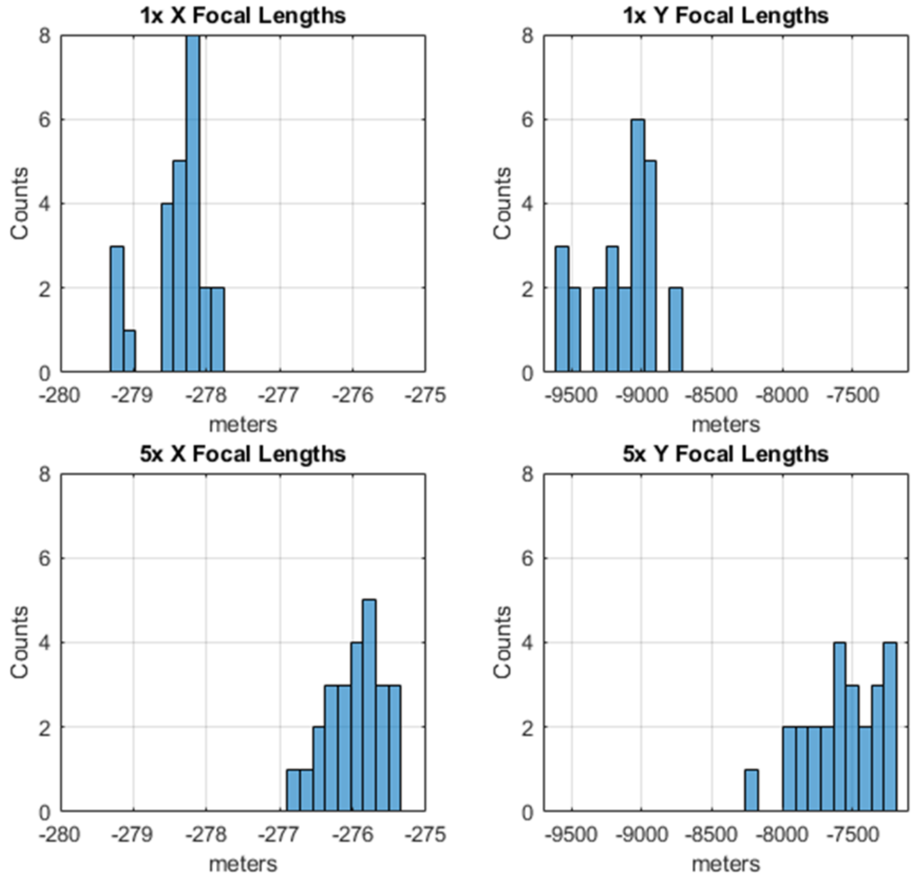

Figure 1: Histograms of reported focal lengths for one and five frame averages. (The test mirror was nominally flat, but imperfect; the negative focal length values indicate a slightly convex mirror surface.)

4.2.2. Projector warm-up test

This test quantified how the projector warming up affected a SOFAST measurement (reported focal length, RMS error, etc.). The theory was that as the projector warms up, the image projected on the screen changes. These changes could be from elements in the projector expanding as they warm up, or from the projector's light source getting brighter as it heats up. Since SOFAST calibrates the response between the projector and the camera and deduces mirror slope based on grey scale intensity, a change in projector intensity would introduce an error into the measurement.

Measurements were taken of a stationary mirror every few seconds for an hour. Measurements started a few minutes after the projector first turned on, and no recalibrations of the projector-camera intensity response were taken. The projector used in this experiment uses a filament light source internal to the projector. Thus, this type of projector is especially susceptible to intensity variations as a function of temperature. As shown in Figure 2, the focal lengths shifted significantly as a function of time. Even after an hour, the rate of change of the focal lengths showed no signs of slowing down.

From this, we conclude that the major source of variability in between SOFAST measurements is "warm-up error." This error increases as more time passes between the last projector-camera intensity response calibration and the measurement.

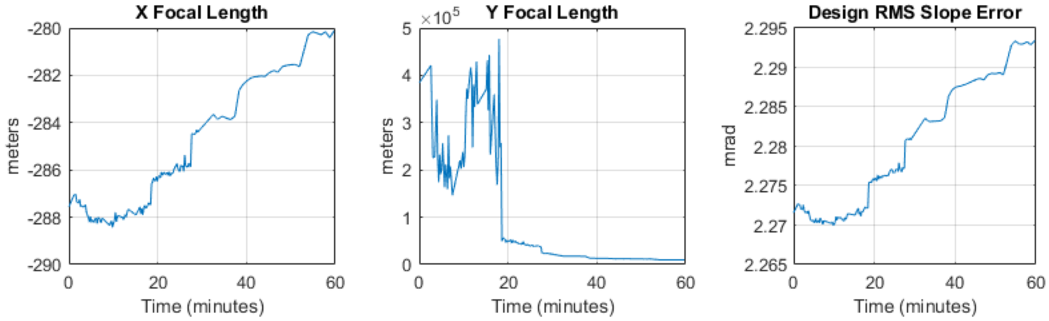

Figure 2: Focal length shift over time as the projector warmed up

4.2.3. Facet Focus Experiment

This experiment studied the accuracy of SOFAST's slope measurement. An NSTTF heliostat facet was first characterized by SOFAST. This facet was then brought to the heliostat field near solar noon and imaged the sun onto the side of the solar tower. The irradiance pattern was captured by NSTTF's Beam Characterization System (BCS). Next, the heliostat facet was adjusted to produce a near circular image of the sun on the side of the solar tower. The facet was then characterized by SOFAST again, post-adjustment. The theory is that if the image of the sun as predicted by SOFAST's surface slope characterization matched the image of the sun captured by BCS, this would serve as an absolute validation of SOFAST's accuracy.

The SOFAST surface slope data and the BCS data were processed and compared; the results are summarized in Figure 3. These results show general agreement between the irradiance pattern predicted by SOFAST's surface slope characterization and the irradiance pattern measured by BCS. This indicates that the slope map measured by SOFAST yields a prediction of optical performance consistent with observed results. A more in-depth test is needed with fewer unknown variables to achieve a full absolute accuracy validation, but these are promising preliminary results.

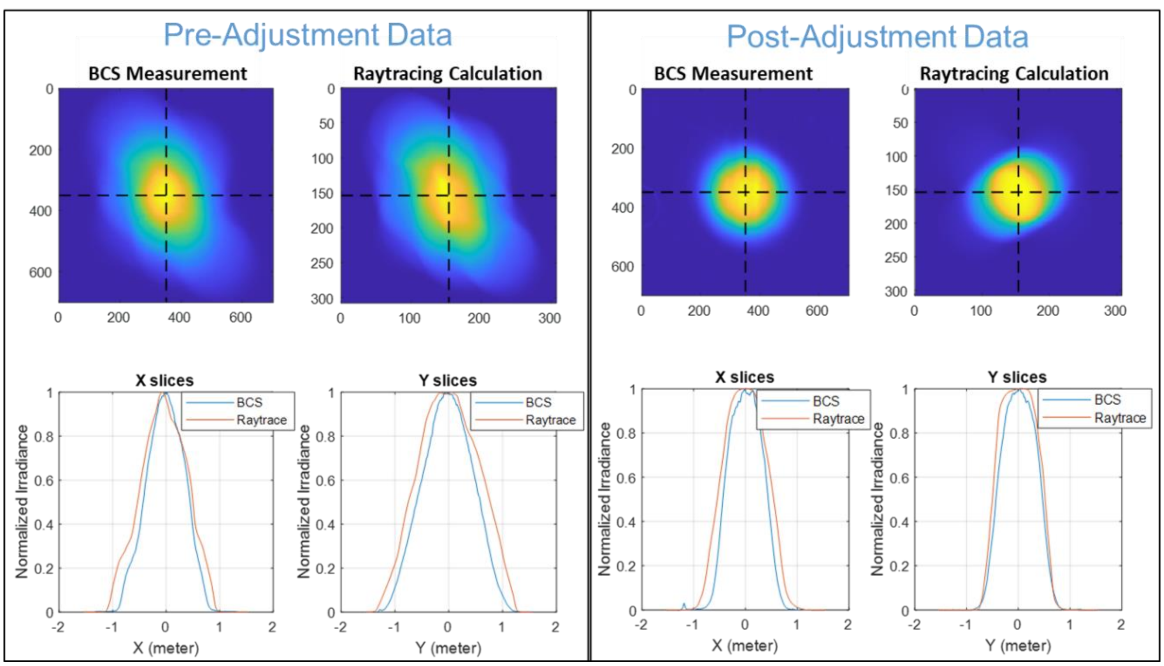

Figure 3: Results of NSTTF Facet Focus Experiment

4.3. Future Work

Other SOFAST 2.0 tests are planned in the future. These tests include:

1) *Continue the "Facet Focus Experiment," described in 4.2.3, to demonstrate SOFAST's utility in focusing heliostat facets.* The goal of this experiment would be to use SOFAST real-time while adjusting an NSTTF heliostat facet's shape to achieve a desired focal length. This would be an important demonstration and proof of concept of SOFAST's utility in heliostat adjustment.

2) *Further the "Projector warm-up test" to include other types of projectors.* The projector examined in the test described in 4.2.2 used a heated filament as a light source. The output light intensity in these projectors is especially susceptible to changes in temperature. Laser based projectors are expected to be less sensitive to temperature changes. Therefore, a future experiment will be to repeat the projector warm-up experiment with a laser based projector, such as the Canon WUX5800 projector currently owned by NSTTF. This experiment will validate the theory that laser projectors, such as the WUX5800, are indeed better to use with SOFAST. With this, Sandia can make more informed and confident recommendations to others who use SOFAST.

3) *Conduct a pose calculation accuracy test.* One important improvement from SOFAST 1.0 is the ability to calculate a mirror's absolute pointing angle (azimuth and elevation angle relative to the normal of the projector screen). Although this functionality has been developed, it has not been thoroughly tested and validated. Thus, NSTTF plans to conduct a test where many measurements are collected of the same mirror in the same location, but in various poses. Analyzing the variability in the reported pointing angles will produce estimates of the absolute accuracy as well as the repeatability of SOFAST's pointing calculation. This is an important result to have as current and future projects at NSTTF will rely on SOFAST's ability to calculate a mirror's absolute pose.

4) *Multi-system comparison.* We currently have two copies of the SOFAST system, and will compare their output in measuring the same mirror.

5) *Comparison against a reference standard.* We would like to acquire a reference mirror that dimensionally stable and precisely produced and measured using a high-accuracy technique such as interferometry. This will enable us to evaluate SOFAST's performance against known standard.

6) *Comparison against alternative measurement methods.* Other techniques, such as laser-based methods, may be used to measure CSP mirrors, albeit more slowly. We would like to implement these techniques and compare their measurements with SOFAST output.

5. OTHER FUTURE WORK

Future work for NSTTF includes adding support for Alignment Implementation for Manufacturing using Fringe Analysis Slope Technique (AIMFAST). AIMFAST is a tool developed at Sandia that is used for accurate canting adjustment of multi-faceted CSP mirrors [4]. AIMFAST uses many of the same functions that SOFAST uses, and the code is largely in the same state as SOFAST 1.0. In the future, NSTTF will work towards making AIMFAST compatible with the updated SOFAST functions as well as bringing AIMFAST up to the same coding standards as SOFAST 2.0.

REFERENCES

[1] Andraka, C. E., Sadlon, S., Myer, B., Trapeznikov, K., Liebner, C (2014). Rapid Reflective Facet Characterization Using Fringe Reflection Techniques. *Journal of Solar Energy Engineering*.

[2] Yellowhair, J., Andraka, C.E., & Finch, N.S. (2011). Heliostat Focal Length Impacts on SOFAST Sensitivity. *Environmental Science*.

[3] Finch, N.S., & Andraka, C.E. (2011). Uncertainty Analysis and Characterization of the SOFAST Mirror Facet Characterization System. *Journal of Solar Energy Engineering*.

[4] Andraka, Charles & Yellowhair, Julius. (2019). AIMFAST for heliostats: Canting tool for long focal lengths. Presented at *SolarPACES 2018*. AIP Conference Proceedings. Volume 2126, 030003. https://doi.org/10.1063/1.5117515.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	01977	sanddocs@sandia.gov

This page left blank

This page left blank

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.