THE UNIVERSITY OF TEXAS AT AUSTIN
CENTER FOR TRANSPORTATION RESEARCH

TECHNICAL REPORT 0-6873-2
TxDOT PROJECT NUMBER 0-6873

Development and Calibration of a Laser-based Highway Deflection Measuring System

Joaquin B. Hernandez
Suyash Vishnoi
Christian Claudel
Jorge A. Prozzi

July 2019; Published September 2020

http://library.ctr.utexas.edu/ctr-publications/0-6873-2.pdf

Technical Report Documentation Page

1. Report No.
FHWA/TX-19/0-6873-2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Development and Calibration of a Laser-based Highway Deflection Measuring System

5. Report Date
Submitted July 2019; Published September 2020

6. Performing Organization Code

7. Author(s)
Hernandez, J. B., V. Suyash, C. Claudel, and J. A. Prozzi

8. Performing Organization Report No.
0-6873-2

9. Performing Organization Name and Address
Center for Transportation Research
The University of Texas at Austin
3925 W. Braker Lane, 4th Floor
Austin, TX 78759

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
0-6873

12. Sponsoring Agency Name and Address
Texas Department of Transportation
Research and Technology Implementation Division
P.O. Box 5080
Austin, TX 78763-5080

13. Type of Report and Period Covered
Technical Report
September 2018–July 2019

14. Sponsoring Agency Code

15. Supplementary Notes
Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

16. Abstract

Despite recent developments, the measurement of actual pavement surface deflections under the axles of a moving truck at a network level remains a challenge in pavement engineering. Several pieces of equipment have been developed in the past decade but they fail to accurately measure actual deflection. Most of them measure indicators of deflection or deflection averages over a given distance. Pavement engineers recognize the usefulness of pavement deflection data to evaluating and monitoring pavement response and performance, assessing pavement structural conditions, and serving as an input for rehabilitation design. Network-level studies of pavement deflections are possible but have major limitations such as the lack of accurate measurements (actual deflection is not measured at any point), the inherent complexity, and safety concerns. Some accurate devices are available but they work at very low speed, posing safety concerns. The purpose of this project is to develop a laser-based highway deflection measuring system for network-level studies that continuously moves along the pavement at speeds of at least 10 mph.

17. Key Words
Surface pavement deflection, network level, laser

18. Distribution Statement
No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161; www.ntis.gov.

19. Security Classif. (of report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of pages
111

22. Price

THE UNIVERSITY OF TEXAS AT AUSTIN
CENTER FOR TRANSPORTATION RESEARCH

Development and Calibration of a Laser-based Highway Deflection Measuring System

Joaquin B. Hernandez
Suyash Vishnoi
Christian Claudel
Jorge A. Prozzi

CTR Technical Report:	0-6873-2
Report Date:	July 2019; Published September 2020
Project:	0-6873
Project Title:	True Road Surface Deflection Measuring Device
Sponsoring Agency:	Texas Department of Transportation
Performing Agency:	Center for Transportation Research at The University of Texas at Austin

Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

Disclaimers

Author's Disclaimer: The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration (FHWA) or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation.

Patent Disclaimer: There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine manufacture, design or composition of matter, or any new useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States of America or any foreign country.

Engineering Disclaimer

NOT INTENDED FOR CONSTRUCTION, BIDDING, OR PERMIT PURPOSES.

Research Supervisor: Dr. Jorge A. Prozzi

Acknowledgments

The authors express appreciation to all personnel from the Texas Department of Transportation that contributed to this research project and provided the research team with useful guidance and continuous support. In particular, we would like to thank Rocio Perez and Wade Odell from the Research and Technology Implementation (RTI) Office. In addition, the following people are acknowledged for their technical contribution and advice: Magdy Mikhail and Jenny Li (Maintenance Division); Gisel Carrasco (Austin District); and Harry Lorton and Butch Ricke (Flight Service Facility).

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Background

Surface pavement deflection measurements began more than 60 years ago with the development of the Benkelman Beam as part of the AASHO Road Test [1, 2]. Highway pavement engineers have always recognized the usefulness of pavement deflection data for evaluating and monitoring pavement response (e.g., maximum deflection) and pavement structural conditions (e.g., backcalculation of layer moduli) [3]. The Texas Department of Transportation (TxDOT) is one of the leading agencies in using deflection data as well as developing and implementing devices to measure pavement deflections. Two recent examples of TxDOT-sponsored device development are the rolling dynamic deflectometer (RDD) [4, 5] and the total pavement acceptance device (TPAD) [6]. While these two devices have extensively demonstrated their usefulness and capabilities, they do not measure actual pavement deflection under a fully loaded standard axle load and they operate at very low speeds. While they provide a deflection measure that can be used to assess pavement condition, this deflection measure cannot be used for backcalculation of layer moduli and rehabilitation design. Therefore, the objective of this project was the development of a novel sensing system capable of accurately measuring pavement deflection for network-level applications at higher speeds than the RDD and the TPAD can achieve. It is estimated that the system should be accurate to 3.9 mils (100 microns) and the target speed was 10 mph.

1.2. What the Researchers Did

The researchers investigated high-performance lasers rangefinders in combination with a high-precision ring laser gyroscope (RLG) for monitoring the deflection caused by a truck driving on a paved highway. The sensor system originally developed used at least three measurement lasers: two lasers monitor the undeflected pavement surface, while the remaining lasers monitor the deflection caused by the loaded truck axle. The final system contains four laser heads for optimization purposes. While the laser measurements are highly accurate, no current technology can determine the absolute position of a system with respect to an inertial frame of reference—at least, not with the required precision (i.e., 3.9 mils, or 100 microns) and during the time required by both lasers to cover the deflection bowl or deflection basin. These considerations mandated a physical coupling between sensors in order to reduce the degrees of freedom. The simplest mechanical coupling, a perfectly rigid beam, allows only six degrees of freedom (i.e., three degrees of freedom in translation and three in rotation). In practice, changes in distances between points of the rigid beam (deformations) that do not exceed a fraction of the required accuracy could be tolerated. In addition, the location of the sensors with respect to a common moving frame is necessary for analysis and deflection estimation. This represents a major challenge for the mechanical fabrication of the beam in order to verify the rigid body assumption while accounting for the location accuracy evaluated at a micron level. So far, with the limitations of the fabrication machinery to reach that accuracy, the researchers developed an algorithm using the three lasers to estimate these fabrication uncertainties within a calibration process before performing actual measurements.

Furthermore, the problem of determining the deflection of the pavement using the distance measurements generated by the lasers is, however, ill-posed (rank deficient matrix), unless a measurement of the orientation of the beam is available. The rotation of the beam around its center can be generated by a high-precision RLG, which precisely tracks the orientation of the beam with respect to the Geocentric frame. Thus, the translational movement is estimated using two adjacent lasers through an optimization algorithm. Hence, the system combines three laser distance sensors with an RLG.

1.3. How They Developed It

After the selection of the beam and the laser distance sensors, the deflection measurement system was developed through several steps. The experimental validation in each step helped to identify the limitations of the current prototype and identify possible solutions to improve it. The main steps are summarized as follows:

- Step 1: Deflection estimation by correlation of measurement.
- Step 2: Deflection estimation using computer vision techniques for matching. These technique accounts for speed variations and vehicle change of direction.
- Step 3: Implementation of the gyroscope to measure the orientation of the beam.
- Step 4: Frequency analysis to demonstrate that the sensors do not have any internal vibrations that might affect the accuracy of the deflection estimation.
- Step 5: Development of a new acquisition interface for better data filtering and synchronization.
- Step 6: Installation of a third laser in the undeflected outside the deflected area to measure the translational movement of the beam.
- Step 7: Development of an algorithm to estimate the translational movement of the beam due to the truck vibrations and the road structure.
- Step 8: Modification of the beam to ensure alignment of the laser and the gyroscope with respect to a common plane with a micron accuracy.
- Step 9: Manual measurement of the misalignment due to the mechanical fabrication uncertainty.

- Step 10: Development of an algorithm to estimate the misalignment by including a fourth laser after failure of manual measurement to achieve the required accuracy.

1.4. What They Found

In addition to evaluating the sensors and developing the acquisition system, the research team has focused on the problem of matching the data generated by the different laser distance sensors. The team investigated several image-matching techniques to correlate the data sections generated by the different laser distance sensors used in the system. They have determined that accurate matching could be achieved in all scenarios covered by the tests using a matching algorithm. The team also developed an algorithm to account for different external vibrations affecting the beam. They identified the limitations of each prototype through experimental validation. They have addressed the theoretical and practical problems gradually to achieve the required precision, i.e., 3.9 mils (100 microns). To date, the main issue to be solved is to ensure the rigid body assumption as the system is mounted onto different vehicles.

1.5. What This Means

This research project developed and implemented a laser-sensor system capable of capturing pavement deflections under a moving vehicle under different loads and speeds. The deflections can be measured using a combination of laser distance sensors and an RLG's measurements. The surface deflection of a pavement structure is an important property that can be used to address many of TxDOT's needs, such as the following: 1) measurement of road condition at the network level, 2) early indication of pavement deterioration, and 3) input for layer moduli backcalculation. Therefore, a system that can allow TxDOT to measure surface pavement deflections without the need to close the highway to traffic yields multiple benefits. For example, the Maintenance Division would be able to use this device to determine the condition of the pavement and use this information in Pavement Analyst to better manage the state's highway network and to make better informed maintenance and rehabilitation decisions. This information would directly support the 4-Year Plan. In addition, backcalculated layer moduli from surface deflections could be used by the Construction Division.

Chapter 2. Deflection Measuring Device (DMD): Mechanics

2.1. Rigid Beam

An essential feature of the system was to have a mounting system for a vehicle and a beam that was stiff enough to behave as a rigid body. To ensure that no more than six degrees of freedom could be observed, the response of the beam was studied through finite element analysis. If the system did behave as a non-rigid body, it would affect the relative positioning of one laser-sensor unit with respect to the other laser-sensor unit, creating a greater level of complexity in the data processing.

The main body of the deflection measuring device (DMD) is a hollow aluminum beam that the research team determined to be a good fit, based on the finite element analysis, modal analysis, and thermal analysis. Table 2-1 lists the characteristics of the beam and Figure 2-1 illustrates the beam's dimension.

Table 2-1 Selected beam's material property and dimensions

Characteristics of beam	
Material	Aluminum 6061-T6
height	8 in (203.2 mm)
Width	6 in (152.4 mm)
Length	120 in (3,048.0 mm)
Thickness	0.375 in (9.5 mm)
Weight	119 lb (54.0 kg)
Young's modulus	71.0 Gpa (10,290 ksi)
Coeff. Thermal Expansion	13.1 µin/in-°F (23.6 µm/m- °C)

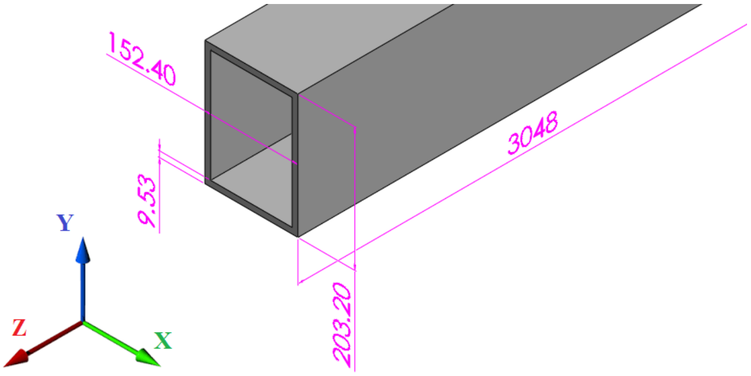


Figure 2-1 Beam with dimensions in millimeters

The height and width of the beam were selected based on the mentioned design considerations as well as what worked best with the initial testing vehicle, which in the early stage of the project was a Ford F-150 pickup truck. The dimensions given in Table 2-1 are for the current beam.

2.2. Beam: Finite Element Analysis

After the selection of an adequate aluminum beam, a series of finite element analyses were performed to characterize the natural frequencies and potential stresses, strains, and deformations. The software ANSYS 2016 was used to analyze the beam with the above dimensions [7]. The maximum static deflection was calculated based on a simply supported beam, using the external forces of the beam as its own weight and gravity. The equation for a beam supported at both ends with a uniform load over the entire span was used to calculate the theoretical maximum deflection, resulting in 2.9 mils (74 microns). It was further verified by ANSYS, resulting in 2.99 mils (76 microns), as shown in Figure 2-2; the difference was seen to be the Young's modulus (*E*) used in ANSYS for 6061-T6 aluminum.

$$\Delta Max = \frac{5WL^4}{384EI} \tag{2.1}$$

where,

W = Uniform load of beam

L = Length of beam

E = Young's modulus

I = Moment of inertia

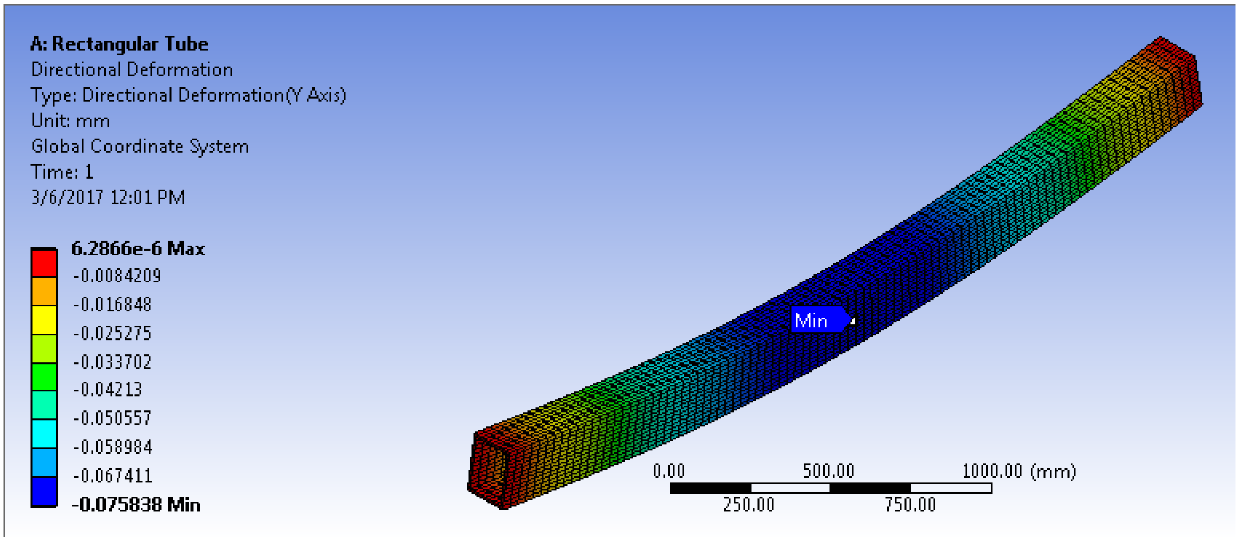


Figure 2-2 Maximum beam deflection under static analysis

After the static analysis, a modal analysis was performed on the aluminum beam. For the modal analysis the equation of motion was applied (Equation 2.2):

$$[M]\{\ddot{u}\} + [\boldsymbol{C}]\{\dot{u}\} + [\boldsymbol{K}]\{u\} = \{\boldsymbol{F}\} \tag{2.2}$$

where,

M = Mass matrix

C = Damper matrix

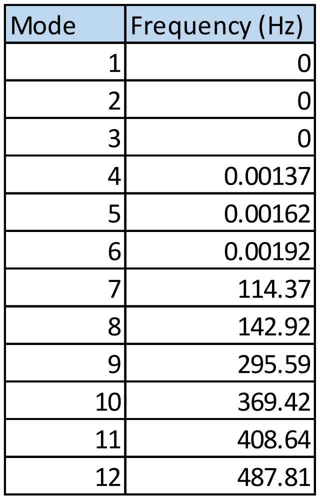
K = Stiffness matrix

F = External forces

The external forces, F, were not considered in this analysis, as the only properties accounted for were the modal analysis of the beam and its structure. The analysis results were used to understand the likely behavior that the beam would experience if exited at a given frequency. However, as mentioned earlier, a mode is not a structural response due to input load. As a result, the ranges of frequencies to avoid and the locations of nodes that could be used for sensor placement were determined.

ANSYS was used to calculate the modal analysis of the rectangular hollow tube. For the analysis, 30,000 nodes were used, a quantity considered sufficient to obtain an approximation of the natural frequency of the selected beam. The specification of the beam's material was that of 6061-T6 aluminum. The total nodes in this analysis was 30,364 with 4,464 elements. For the setup, there were no constraints; the analysis was based on a free-free beam. From such analysis, the first six modes should result in 0 or close to 0 Hz, as this means the structure translates and rotates freely without any internal stress. Table 2-2 shows the first 12 modes for the beam and the corresponding frequencies. From modes 0 to 3 the frequency was 0, while for modes 4 to 6 they were fairly close to zero. Mode 7 is substantially higher and has a bending behavior that makes the extreme end of the beam rotate around the y-axis. Therefore, the frequency to observe is mode 7 with a natural frequency of around 114 Hz, as seen in Figure 2-3.

Table 2-2 Aluminum beam's first 12 modes and corresponding frequency



Mode	Frequency (Hz)
1	0
2	0
3	0
4	0.00137
5	0.00162
6	0.00192
7	114.37
8	142.92
9	295.59
10	369.42
11	408.64
12	487.81

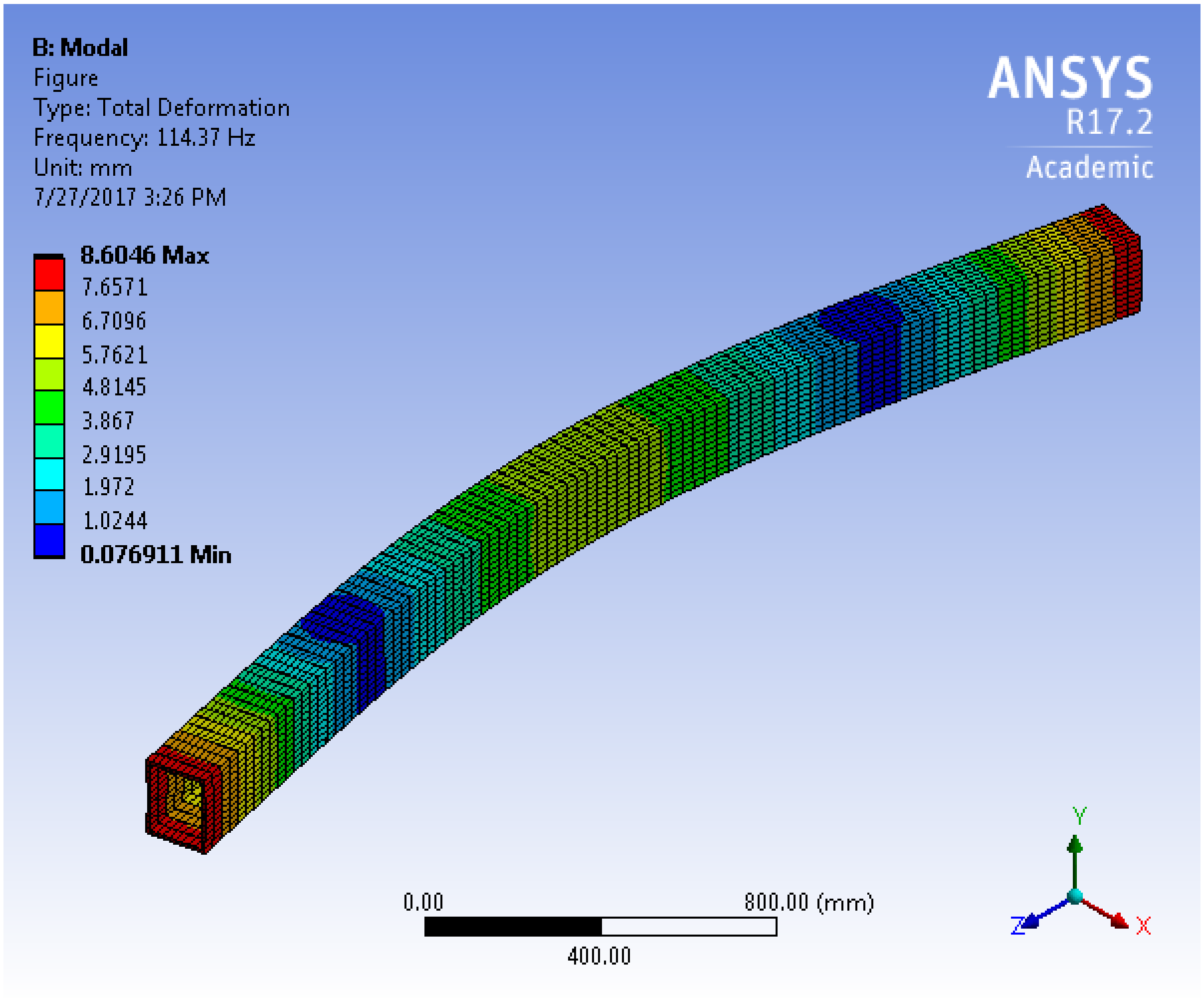


Figure 2-3 Total deformation by mode 7 under modal analysis

The results were satisfactory because in our laboratory test and field tests we did not approach the frequency of 114.0 Hz. Therefore, we did not have complications with any deformations caused by reaching the natural frequency of the aluminum beam. Previous experiments using the F-150 found that the power spectral density of the truck vibration was in the range of 8 to 10 Hz [8]; therefore, we were safe to proceed.

2.3. Initial Validation Testing

The term *DMD* is used to refer to the deflection measurement system developed as part of this project. As originally installed on an F-150 pickup truck, this system includes the following components: aluminum beam, two to three laser-sensors, two accelerometers, and one gyroscope. The system is attached to the pickup truck by three brackets that extend from the side of the vehicle. These three brackets cradle the DMD system to maintain it parallel to the moving direction. The laser-sensors, used for measuring road surfaces, are connected to the beam by machined metal plates. Each plate has two slots to allow height adjustment of the laser-sensor by fixing a bolt within the slot. The change in height is with respect to the beam on which the plate is mounted.

The total height adjustment is approximately 2.75 inches (70 mm). The main body of the DMD, which all the other instrumentations are fixed to, is the beam that was previously discussed in Section 2.2. The beam and its CAD model are presented in Figure 2-4. [9]

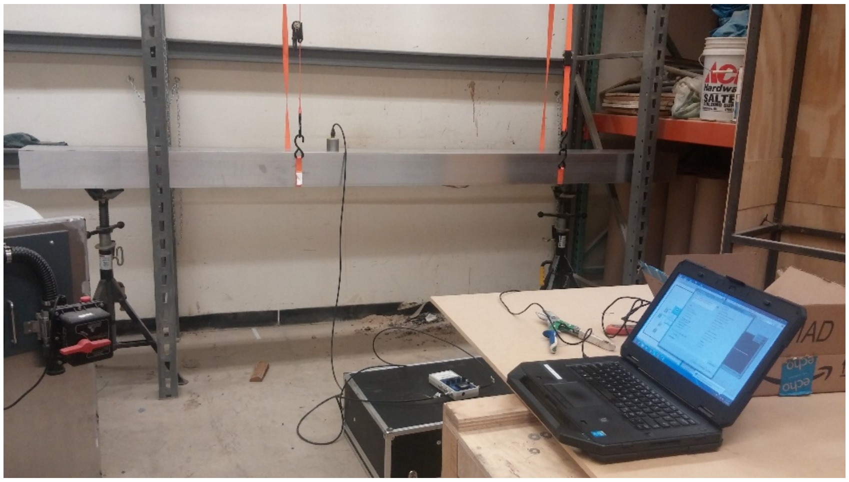


Figure 2-4 The aluminum beam used in DMD equipped with an accelerometer

As the pickup truck travels along a pavement, the vibrations caused by road surface profile would be transferred to the DMD through the vehicle's suspension and brackets that hold the beam to the truck. To decrease these vibrations, different types of vibration absorbers, such as polymers and foams, were used. To measure the beam vibration, two measuring devices were used. The first device comprises two accelerometers, both triaxial, placed at the front and rear of the beam. These accelerometers are used to measure the vertical acceleration of the beam. The second device is a gyroscope, which measures the angular rotation of the beam, installed at the center of the beam (Figure 2-5). The accelerometers are used to validate the data which is captured by the gyroscope.

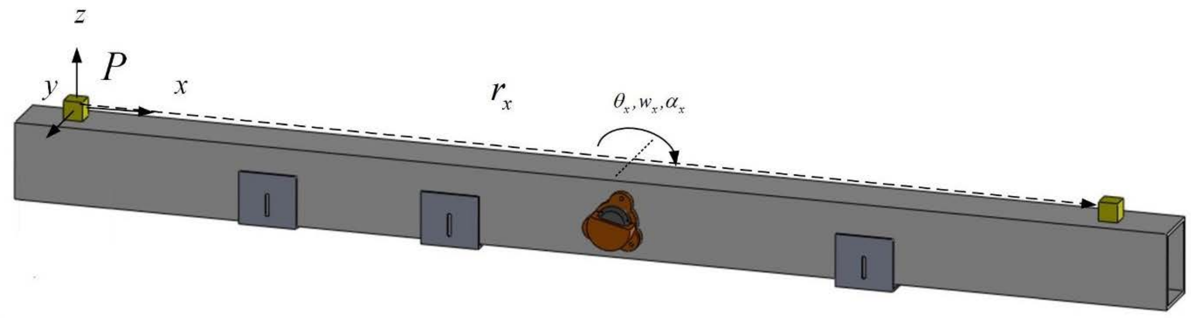


Figure 2-5 The CAD model of the gyroscope, middle, used for measuring the beam rotation and the accelerometers at the ends; the three brackets are also seen on the bottom

For connecting the gyroscope to the beam, a mount was designed in Solidwork. The first prototype of the gyroscope mount was made by Ultimaker 3D printer [10]. PLA plastic was used to print the mount. The prototype worked for initial testing and interfacing but was not adequate for field testing—PLA plastic is a soft material and it did not provide enough reaction mass for the dither motor within the gyroscope. The first mount prototype is presented in Figure 2-6. The mount was connected to the middle of the beam by a cyanoacrylate adhesive. Later development lead to a square mount of aluminum with a cavity for the gyroscope to fit in. Threaded inserts were used in the mount to reinforce the points where machine screws are used to fix the gyroscope.

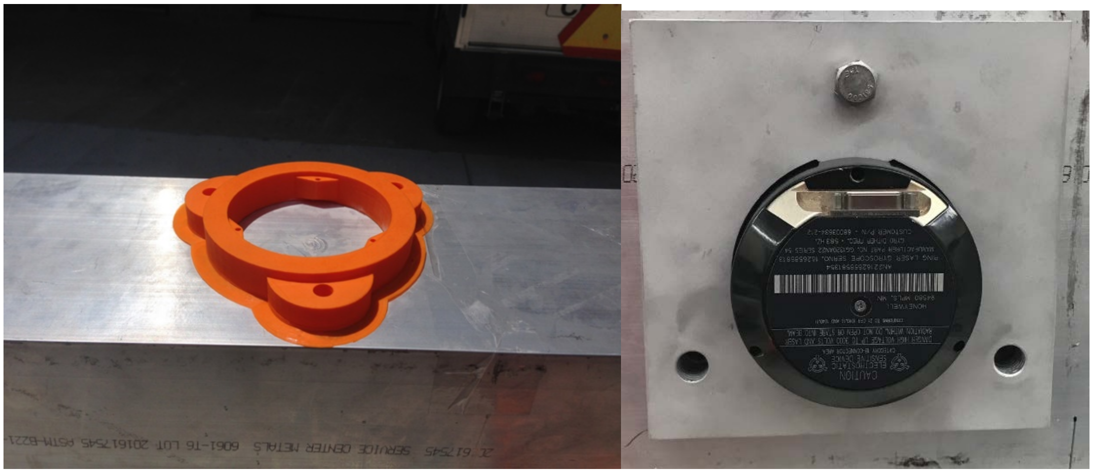


Figure 2-6 Prototype mounts: (Left) gyroscope mount prototype; (Right) aluminum gyroscope mount

To support the beam to the F-150 pickup truck, the passenger's side nerf bar was removed exposing three flat brackets that extrude from the chassis of the pickup truck. These brackets were used to support the beam. A combination of an L-bracket and C-channel were used to support the beam to the side of the pickup truck. The steel C-channels serves to hold the beam in place (cradle position), while the L-channel connects the C-channel and to the bracket extending from the truck. The C-channel and L-bracket were machined to have holes and slots, as shown in Figure 2-7. The slot on the side of the C-channel serves as a safety measure to prevent the beam from moving in the transverse direction (x-axis). The slot allows only vertical displacement of the beam. The complete setup can be seen in Figure 2-8, the L-brackets and C-channels are fastened by 3/8-16 inch machine screws, locking washers, and washers.

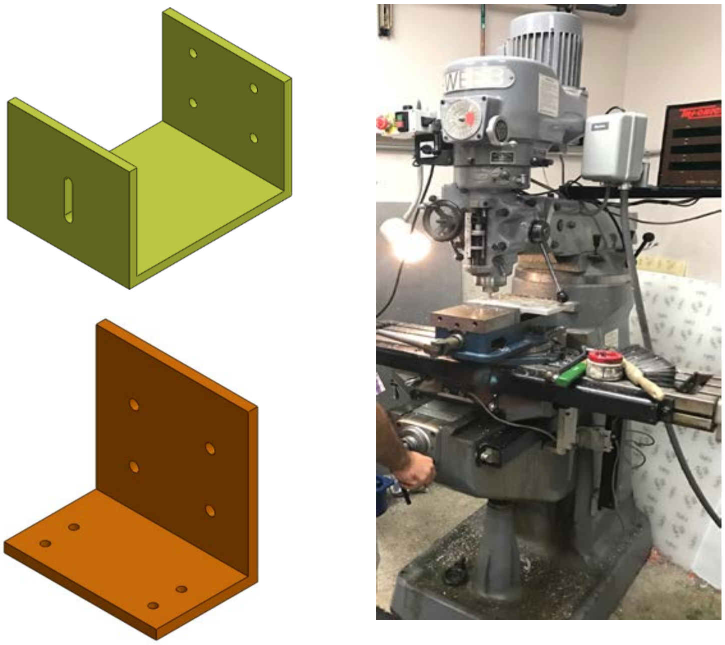


Figure 2-7 The L-bracket and C-channel used to support the beam on the pickup truck

The assembly of these two parts and their CAD simulation as a unit is presented in Figure 2-8.

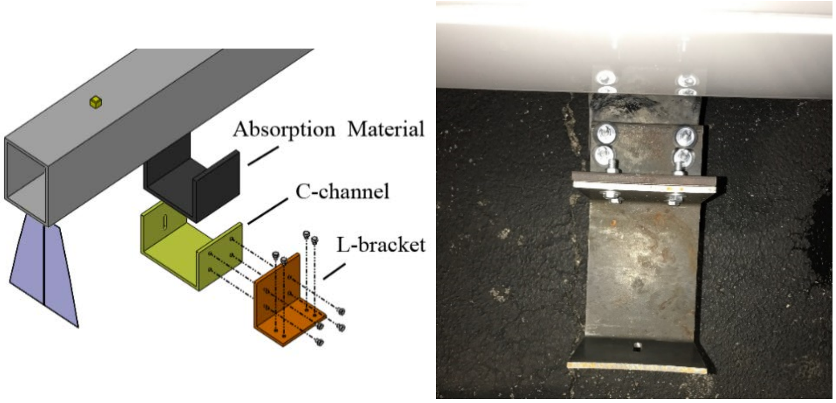


Figure 2-8 Complete setup: (Left) the L-bracket and C-channel unit that support the beam and connect to the pickup truck; (Right) L-bracket fastened to pickup truck

For measuring the deflection caused by a wheel load, laser-sensors were utilized. These laser-sensors are connected to the beam with an adjustable plate. More details can be found in Appendix A. For making the plates, milling and drilling machines were used to add holes for fixing the laser-sensor and slots to adjust for the height of the plate on the beam; the equipment used is illustrated in Figure 2-9.

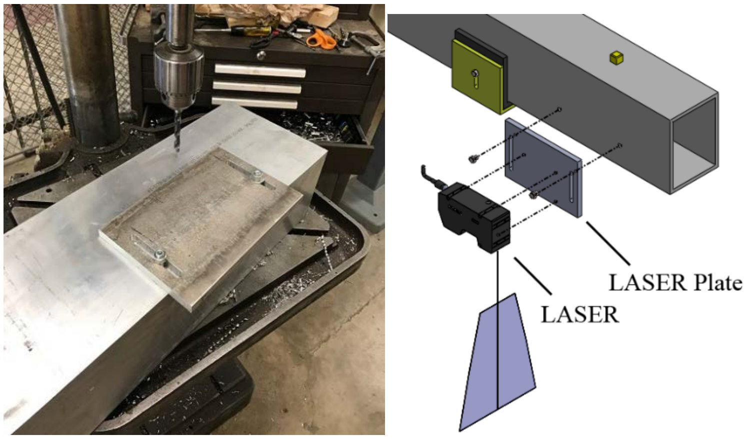


Figure 2-9 The fabrication of the plate and schematic of laser-sensor assembly

Finally, all the parts were assembled to create the DMD system. The beam was placed on supporters connected to the F-150 pickup truck. The CAD model of the beam and supporters are provided in Figure 2-10.

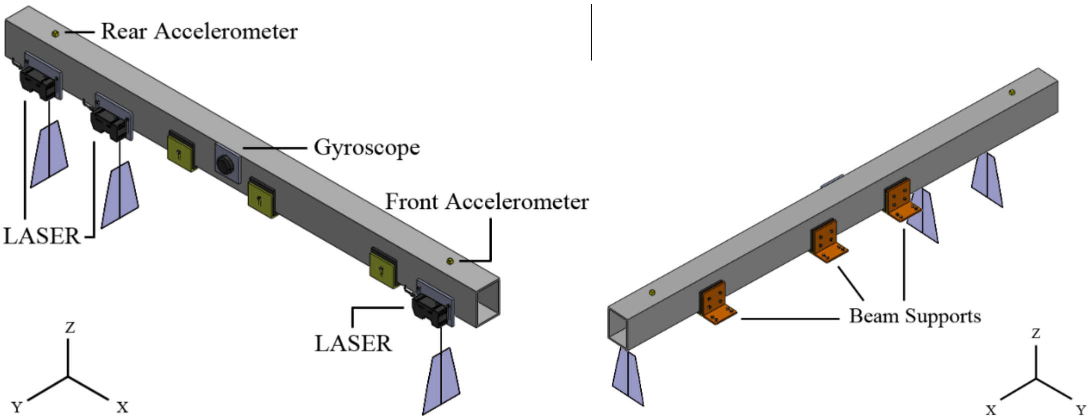


Figure 2-10 DMD system with sensors and mounts, shown on both sides

After fabricating and assembling all the parts, preliminary testing was conducted at the Pickle Research Campus of the University of Texas at Austin. Figure 2-11 presents the beam, supporters, and all the other parts that have been connected to the truck to comprise the DMD.

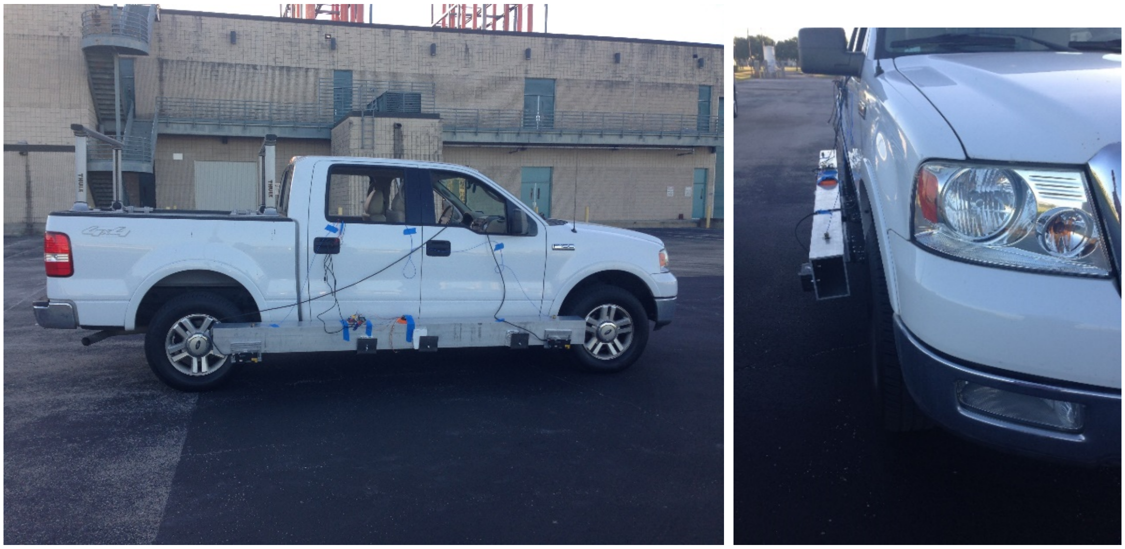


Figure 2-11 The F-150 pickup truck with DMD

To dampen the vibration of the truck (which is transferred to the beam through the supporters), vibration absorbers were used. The next section discusses the investigation of the effects of three different vibration absorbers, including two types of rubbers and foam.

2.3.1. Vibrational Damping: Sorbothane

To dampen the vibration of the vehicle, the first tested material was Sorbothane [11], placed between the beam and the C-channel. This type of rubber material combines some of the properties of rubber, silicone, and other elastic polymers. It is considered to be a good vibration dampening material, and highly durable. Sorbothane is a visco-elastic material, meaning that it exhibits properties of both liquids (viscous solutions) and solids (elastic materials).

Sorbothane was compared to polystyrene foam [12, 13] and nitrile butadine rubber by placing three axis accelerometers and measuring the vibration transferred from the vehicle to the beam with the use of these three materials. The most desirable results came from the Sorbothane material, showing the lowest vibration: 0.1 g compared to 0.2 g.

2.4. Deflection Measurement with Commercial Trucks

In the early stage of the project, as discussed in Section 2.3, the DMD was mounted on an F-150 pickup truck connected to three supports extruding from the passenger side. The supports meant for a nerf bar were used as the connecting points for a bracket that extended out to the side of the vehicle to cradle the DMD parallel to the F-150. With this configuration, the algorithm for matching the laser-sensor scan was validated as well as the measurements of the rotation angle with the gyroscope. The DMD included three laser-sensors, one gyroscope, and two accelerometers. The hardware used to connect to the F-150 pickup truck was a C-channel bracket and an L-bracket, as seen in Figure 2-12.

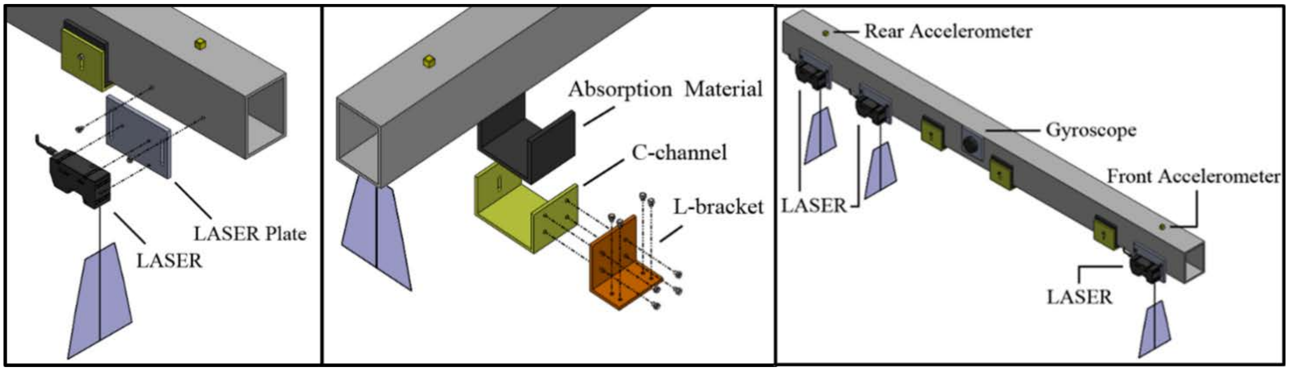


Figure 2-12 Initial mounting method for pickup truck

To get measurements of deflection, a vehicle that could apply a larger load on the pavement was necessary. The research team then transitioned to a larger truck. The truck used in the project was the largest available without requiring a commercial driver license. The vehicles rented were two-axle 26-foot box trucks from rental companies in Austin, Texas. The researchers used a portable wheel load scale from Haenni [14] to find the curb weight, which was 16,475 lbs (7,473 kg), and the specific load for each wheel and axle (Figure 2-13). The rear axle had dual wheels and weighed 5,275 lbs (2,393 kg) on the rear driver side and 4,975 lbs (2,257 kg) on the rear passenger side for a combined rear axle weight of 10,250 lbs (4,649 kg).

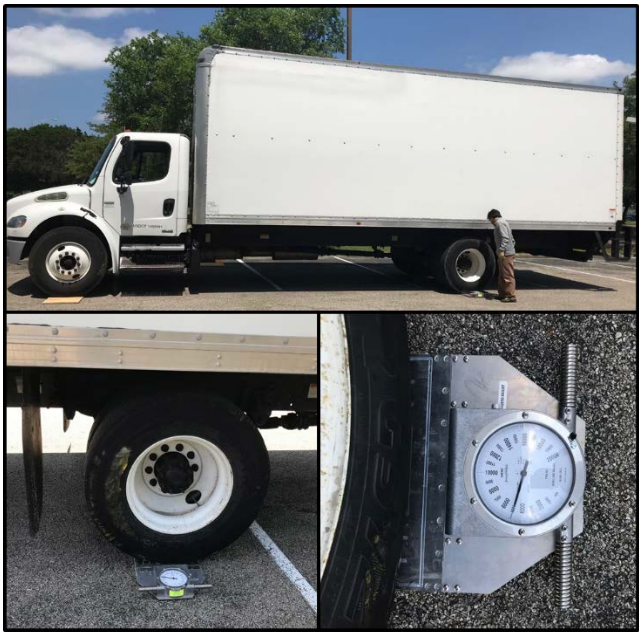


Figure 2-13 Obtaining the weight of the commercial vehicle

As part of the agreement with the rental company, the truck was not to be modified, which limited ways of mounting the DMD frame underneath the truck. To solve this problem, the research team looked to the undercarriage of the cargo box in the trucks, which had I-beams, or cross-members, that spanned the width of the cargo box (Figure 2-14). The general dimensions (spacing, height, thickness) of these I-beams varied with truck model, year, and rental company. Therefore, the mounting system to be developed had to adapt to different trucks. This situation posed some new and unexpected challenges and limitations. Having to work with a different vehicle each month meant that the research team had to modify algorithms every month to adapt to the specific vibrations of each new vehicle. It is strongly recommended that, in future research, a dedicated vehicle should be made available so the measuring beam could be permanently attached, thus obviating many of the modification issues faced in this project.

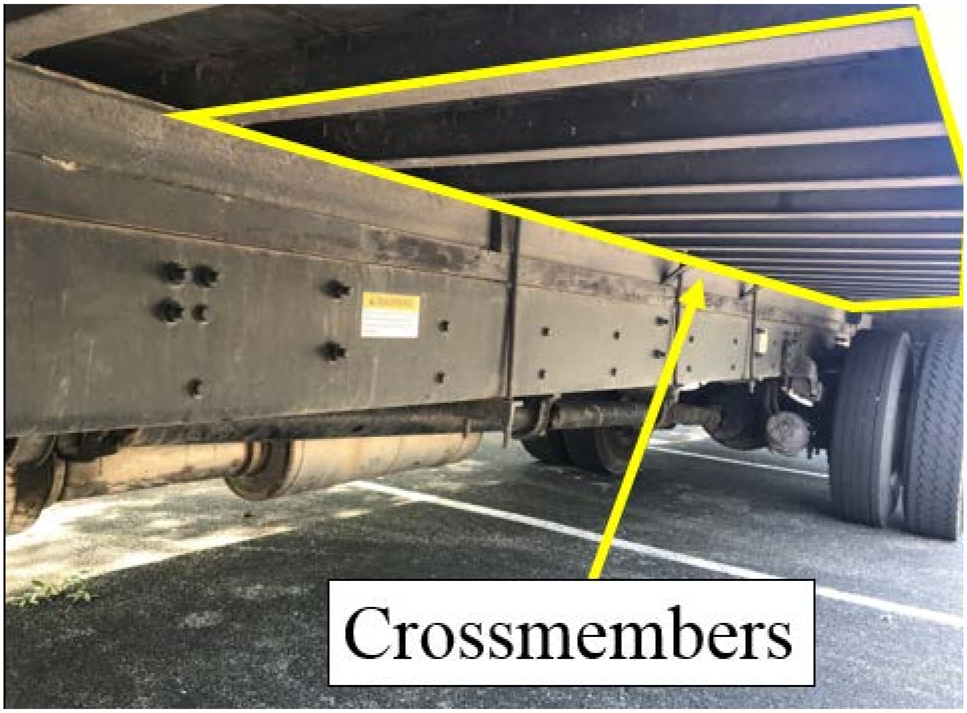


Figure 2-14 View of the undercarriage of a truck

The next step was the development of a system that could adjust to these variables. With no off-the-shelf product to solve the problem, custom fabrication of a frame that would hold the DMD underneath a box truck was pursued. The frame was fabricated at The University of Texas at Austin by the research team to have the flexibility to adjust the height of the DMD with respect to the pavement surface, as well as attaching the frame to any box truck undercarriage that has I-beams. The components of the frame are aluminum extrusion profiles of 60x60 mm that use T-slots, gussets, and quick connection bolts, making it easy to disassemble the frame or to modify as needed. The contact area between the beam and the frame use the viscoelastic rubber pads from Sorbothane to limit the amount of vibration transferred from the engine, drive shaft, exhaust, or other components. Figure 2-15 provides the CAD model and Figure 2-16 shows the DMD installed on a commercial truck.

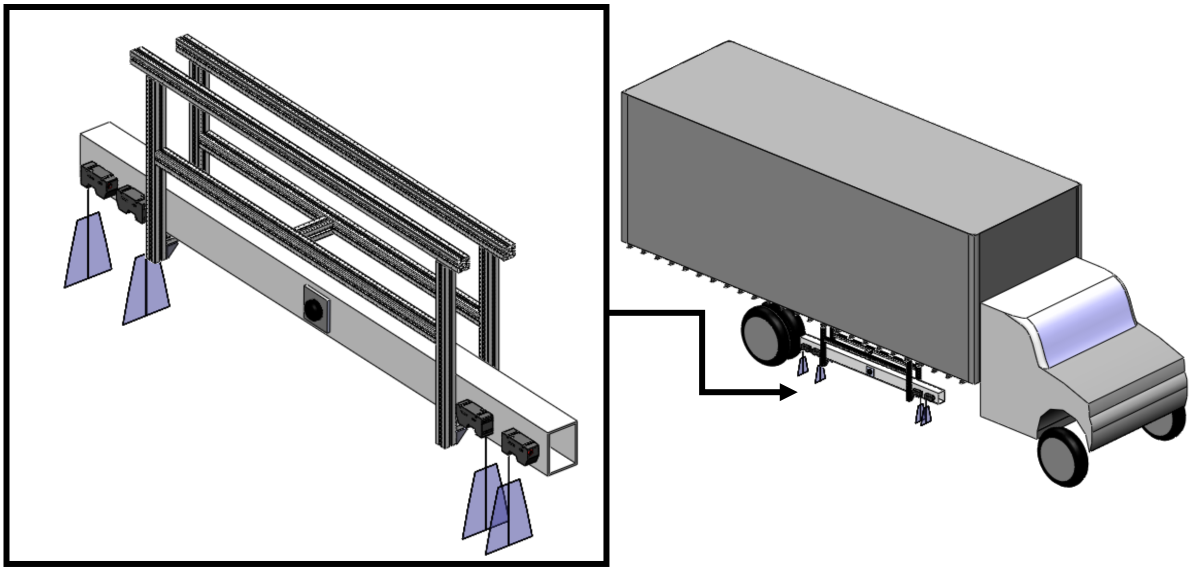


Figure 2-15 CAD model of frame and DMD fixed to the undercarriage of the box truck

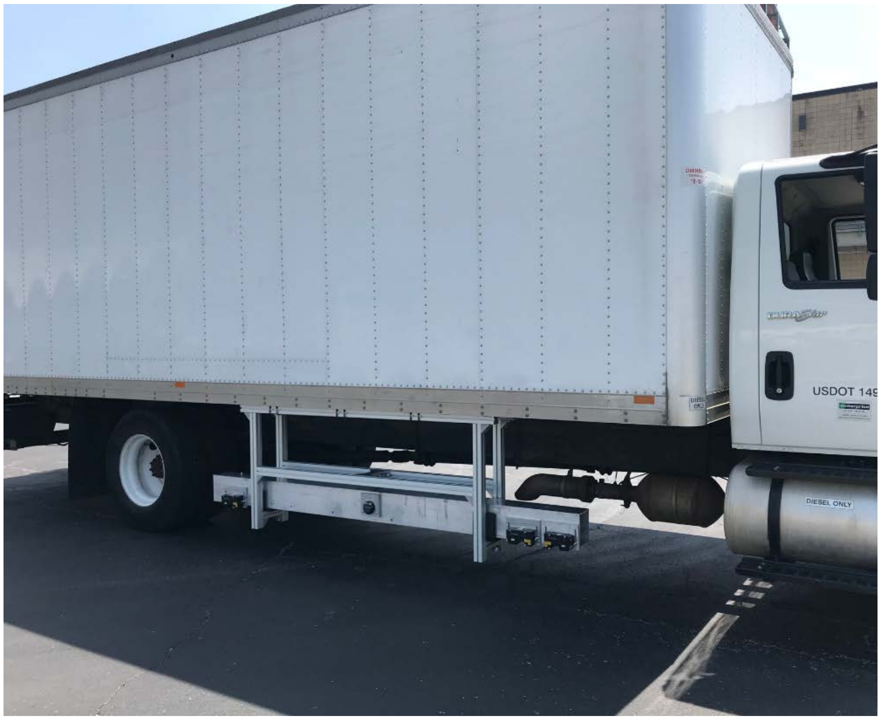


Figure 2-16 Commercial truck with DMD installed

The custom bracket fabricated utilizes large square U-bolts and L-brackets to affix the frame to the I-beams of the truck. To ensure the safety of the DMD, twelve of these brackets were made and used in pairs on the I-beams (Figure 2-17). The nuts on the bracket are torqued to specification to provide a clamp load of 5,588 lbs (2,535 kg).

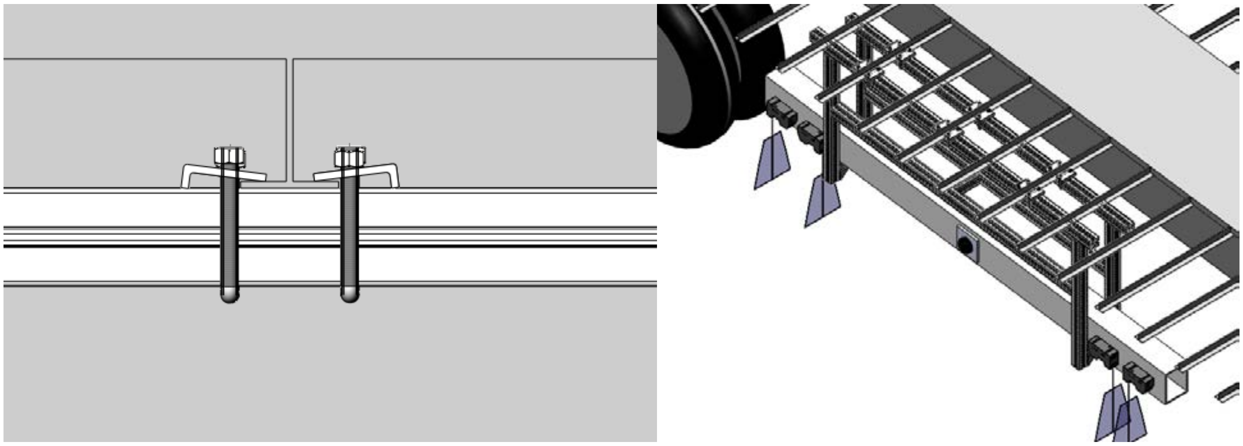


Figure 2-17 Fabricated mounts to fasten frame to the crossmember of the truck

Using components commonly used for heavy-duty applications, the frame was overdesigned for safety, rigidity, and mass to prevent undesirable vibrations. The weight of the overall system (frame and beam) is 200 lbs (91 kg). The fasteners used in the mounting frame are vibration-resistant. Field tests for evaluating the performance were carried out to analyze the behavior of the system and observe for vibrational loosening of nuts or bracket at different driving speeds (5, 10 and 15 mph). The overall system performed well and no vibrational loosening was observed after a week of extensive testing. With the flexibility of the system, the DMD can be positioned in the middle of the box truck far from the tires (undeflected pavement). This configuration is ideal when performing optimizations of the system, as presented in later chapters. Moreover, the DMD can be moved closer to the leading edge of the rear tires to measure the deflection at the leading edge.

A second setup (Figure 2-18, image B) was designed to allow for two laser-sensors to be positioned adjacent to the rear wheel by shifting the frame outside of the box truck. This maximizes the distance the laser-sensors are located within the deflection basin. Laser-sensor Head B measures the peak of the deflection while the second laser-sensor, Head D, would be within the deflection basin.

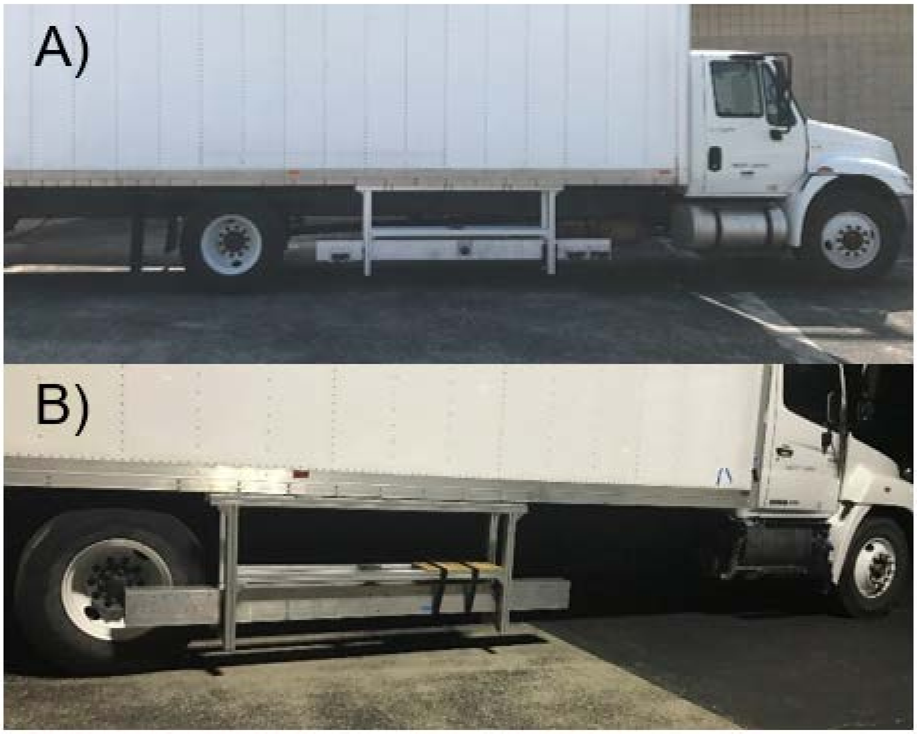


Figure 2-18 DMD locations: A) DMD positioned away from deflection basin, B) DMD positioned to the rear in the deflection basin.

With the DMD’s use on large trucks, the number of sensors changed as well; originally the DMD was equipped with one gyroscope, two accelerometers, and two laser-sensors. The number of laser-sensors was eventually increased to three to validate the matching in the F-150 pickup truck. When the research group transitioned to a larger vehicle to measure deflection, an additional laser-sensor was implemented. Four laser-sensors were now employed: two laser-sensors very close together in the front and two in the rear to measure the peak of the deflection; the second was placed within the deflection basin. The accelerometers were removed from the DMD system as the data from the gyroscope out-performed the accelerometers. These changes resulted in an updated deflection measurement software (DMS) developed in LabVIEW.

Chapter 3. Sensors and Software

3.1. Rotational Measurement: Gyroscope

The Honeywell GG1320 Ring Laser Gyroscope (RLG) possesses the level of precision and accuracy that makes the device suitable for military applications (Figure 3-1). The GG1320 requires a dedicated power supply to operate at a certain voltage. On every positive edge of request signals, a data packet containing status, angle, and other device information is sent at a rate of one megabaud. The seller was unable to provide support for this product and the work performed in this project to integrate it into the DMD system was a custom solution.

The RLG was selected because of its resistance to vibrations and other mechanical noise that do not otherwise cause the RLG to rotate along its measurement axis, or cause acceleration measurements to be mistaken for rotations, as is the case with MEMS gyroscopes. The drift of the device is on the order of 3 millidegrees per hour of integration. The maximum rotation rate that the RLG can reliably measure without saturation is 900 deg/sec.

Initial attempts to mount the RLG to the aluminum beam used a 3D-printed housing made from PLA plastic. The RLG would work intermittently when free from constraints and stop working when bolted to the aluminum beam. It was later discovered that these sensors need a sufficiently rigid support for the internal dither motion. The dither motor in the RLG is a high-Q mechanical resonator that is dependent on the external mount in which it is fixed. To prevent performance degradation, an aluminum housing was fabricated to provide a moment-of-inertial about the RLG's input axis greater than greater than 43.0E-04 kg-m^2 [15]. To interface with the RLG an Arduino Mega board [16] was integrated between the RLG and the computer to format the data packet from the RLG. The one-megabaud speed that the RLG transmits data is more than what a typical laptop can handle. Therefore, the data packet compiled by the Arduino sent the unitless angle output and the status, discarding other unnecessary information packets.

At one megabaud, the 6-byte (48 bits) packet are prepared using 8-N-1 asynchronous protocol by the gyroscope, consists of Status, Tag ID, LSB angle, MSB angle, Tag Data Byte, and a checksum. Bits 7 and 6 of the Status byte indicates which state the RLG is currently in. "00" indicates WARNING state. This state occurs when the internal laser power is too low, the dither is operating outside performance limits, or the software self-check detects an error. Data acquired in this state may be unusable. "01" indicates "No Computed Data" state. This is the RLG's initial state, and remains in this state until the startup sequence is complete. "10" indicates CAUTION state. This state occurs when built-in-test (BIT) faults involving discharge current, high voltage, or temperature occur. Although an output is still provided, the performance may decline. The final state is indicated by "11", NORMAL state. This state indicates the gyroscope system is healthy and reporting accurate data.

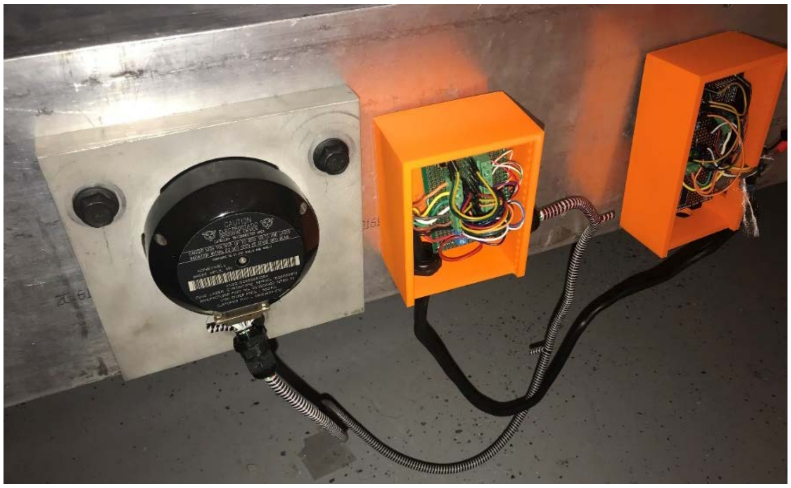


Figure 3-1 Honeywell GG1320 AN22

The angle output of the data packet is created by joining bytes 3 and 4 to form a 16-bit theta measurement. The raw output of the RLG is initially unit-less until the scale factor is applied, generating an output in arc-seconds. The scale factor of the GG1320 is 1.113065 arc-seconds per gyro count or 1,164,352 gyro counts per revolution. Therefore, the output is a relative measurement rather than an absolute description of orientation.

3.2. Displacement Measurement: Laser-sensor

The laser-sensors used in this project were LJ-V7300, developed by Keyence Corporation [17]. The reference distance, height from surface of interest to the laser, is 300 mm and the measurement range can be ± 145 mm from the 300 mm reference. The laser emission is classified as a Class 2, thus not requiring personal protective equipment. More information about the lasers' characteristics is listed in Table 3-1.

Table 3-1 Specification of LJ-V7300

Laser-Sensor	
Reference Distance	11.81 in (300 mm)
Measurement Range	5.7 in (145 mm)
X - Axis (width)	
Near side	4.3 in (110 mm)
Reference Distance	7.1 in (180 mm)
Far side	9.5 in (240 mm)
Output	4.8 mW
Repetability	
Z - axis	0.2 mils (5 µm)
X - Axis (width)	2.4 mils (60 µm)
Profile data interval	11.8 mils (300 µm)
Class	Class II

3.3. Accelerometer

A high sensitivity, three-axis accelerometers (PCB 356B18) connects into a data acquisition (DAQ) system that operates in a plug-and-play manner [18, 19]. The accelerometer has three outputs, each output corresponding to an axis. The signals first passed through a three-channel signal conditioner to adjust the signals before going into the DAQ [20], both devices can be seen in Figure 3-2. For the accelerometer, the minimum sampling rate was 1.652 kHz. The characteristics of the involved equipment can be seen in Table 3-2.

Table 3-2 Specifications of accelerometer, conditioner, and DAQ

Accelerometer: PCB 356B18	
Sensitivity	(± 10%) 1000 mV/g (102 mV/(m/s2)
Measurement Range	± 5 g pk (± 49 m/s2 pk)
Frequency Range	(± 5%) 0.5 to 3000 Hz
Conditioner	
Channels	3
Gain	x1, x10, x100
Frequency Range	0.15 to 100,000 Hz [x1, x10 gain]
Data Acquisition	
Channels	4
Max Sampling Rate	51.2 kS/s/Channel
Analog Input Voltage	± 5 V
ADC Resolution	24 bits

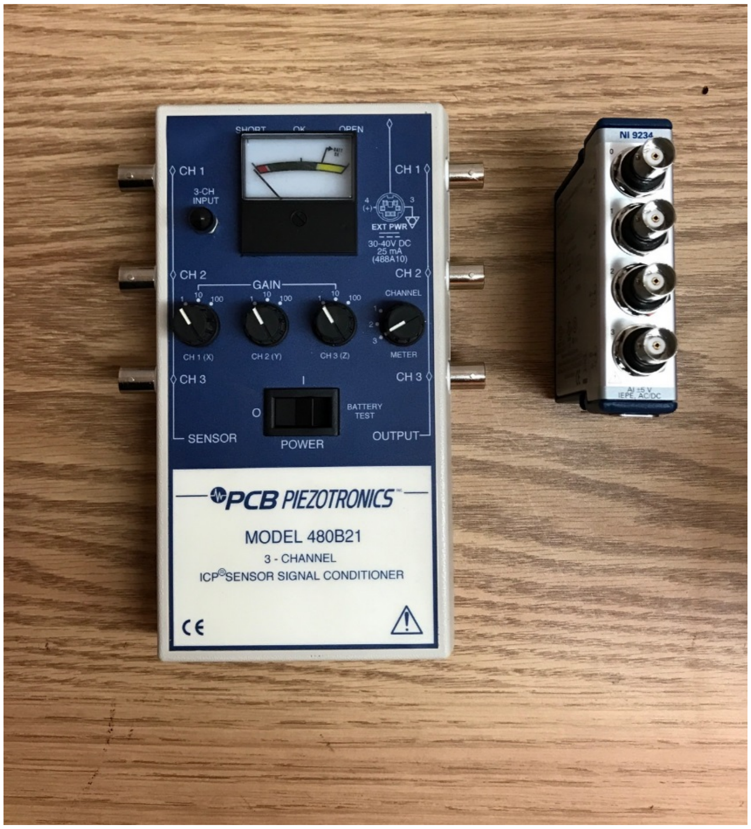


Figure 3-2 Conditioner on the left and DAQ on the right

3.4. Deflection Measurement Software (DMS)

The software was written in LabVIEW [21] and updated from the previous version, which operated only three laser-sensors and a gyroscope. The latest version operates four laser-sensors and the gyroscope, and has the option to operate accelerometers. The program and sensors are operated from a single Windows-based laptop via Ethernet and two USB connections. There are two Keyence controllers (each controller communicates with two laser-sensors); one controller outputs data through Ethernet and the second uses USB connection. The transmission speeds for both Ethernet and USB were observed to be equivalent. The custom gyroscope board used a USB connection to supply the board 5V and to transmit the data to the laptop. The measurements for the four laser-sensors are saved into a single TDMS (Transition Minimized Differential Signaling) file and the gyroscope data is saved to a separate TDMS file. Information on the proper operation of the system is displayed to the user while measurements are collected, including real-time profiles, buffer rate, elapsed time, number of profiles collected, and status of the laser-sensor controller. A sampling rate of 1.0 kHz was used for the first half of Phase 2 and it was later re-programmed to handle 4.0 kHz. The laptop used was the limiting factor, which reached data transmission problems past 4.0 kHz.

The software was designed to be intuitive. A start button is used to establish communication with the Keyence controllers and a second start button to start monitoring/gathering data from all connected sensors (Figure 3-3). Data is displayed as it is collected, with height values from the respective laser-sensors.

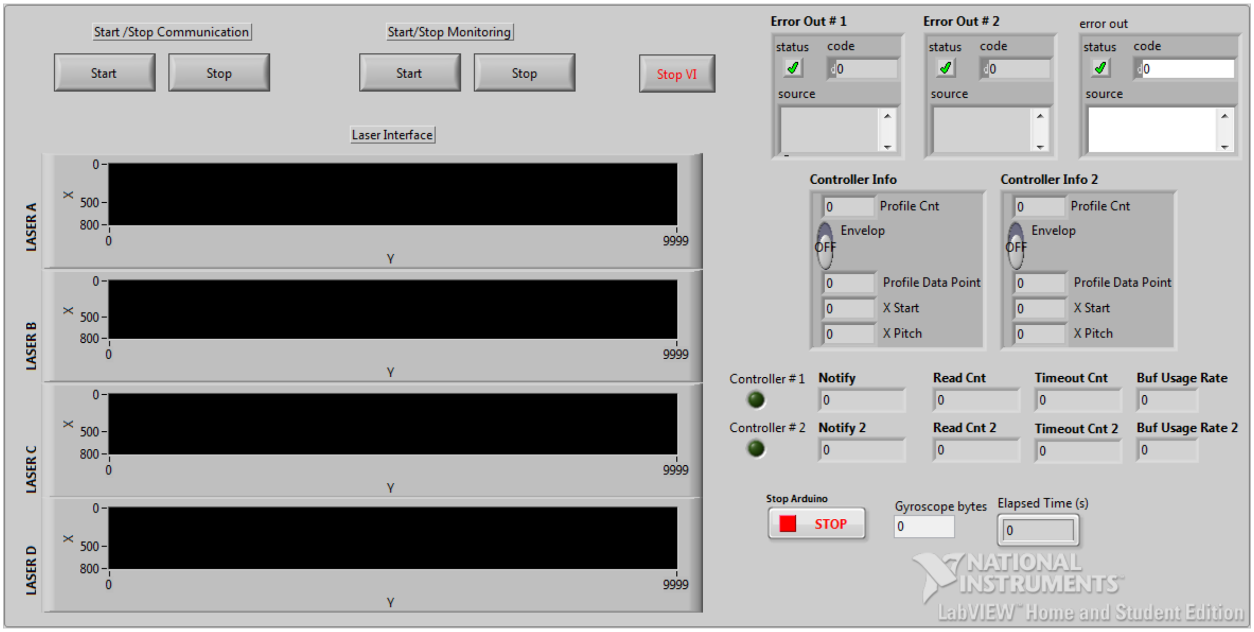


Figure 3-3 DMS: Graphical user interface

Chapter 4. Field Tests for Data Processing Development

During the months of July to November 2018 and January to August of 2019, extensive evaluation of the system was performed at the Pickle Research Campus under a range of operating speeds, axle loads, pavement types and pavement temperatures. A single test was conducted at TxDOT's Flight Service Facility (FSF) at the Austin-Bergstrom International Airport (ABIA) to test a rigid pavement. The load of the rear axle of the two-axle truck varied between approximately 4,500 and 5,500 lbs.

4.1. Flexible Pavement Testing

The research team conducted experiments on two asphalt pavement sites at Pickle Research Campus: Creativity Trail and Innovation Boulevard. Roads were selected based on pavement homogeneity, section length, and traffic level. Short-length testing was performed in an area behind the Microelectronics Research Center, which had two different asphalt sections and a concrete slab. All of the evaluations were performed after-hours to limit interruptions (commuters, pedestrians, public busses) while testing. The specific locations are shown in Figure 4.1.

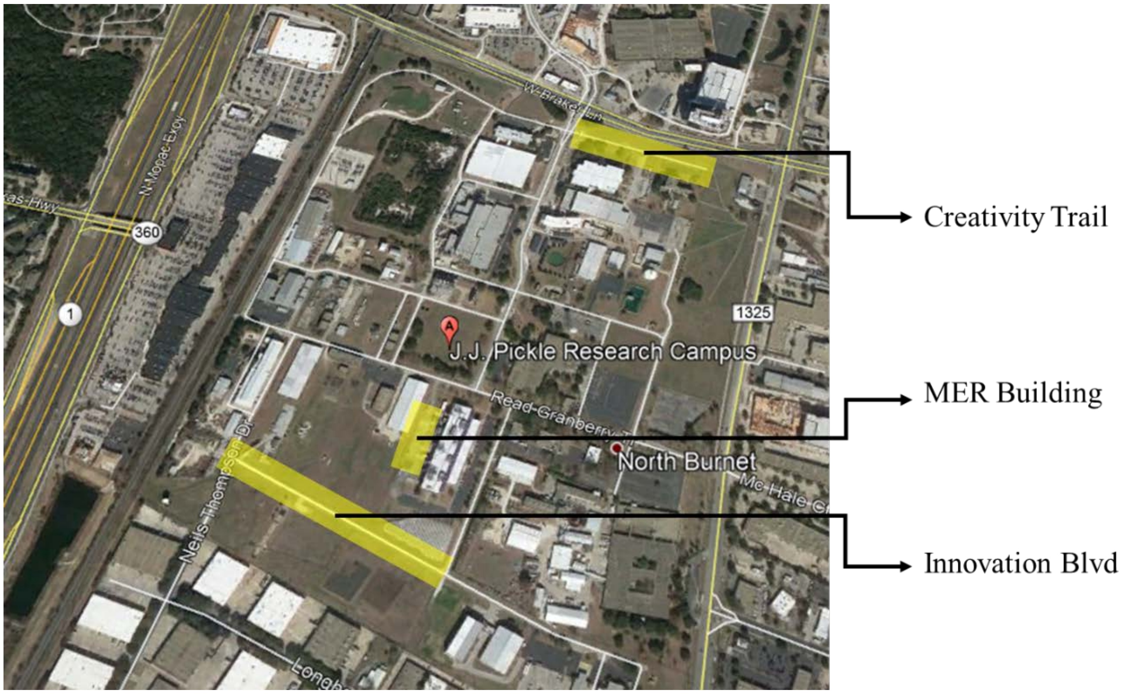


Figure 4-1 Field test locations within Pickle Research Campus

The road test sections were prepared with a fluorescent string secured to the pavement (extending the full length of the road to be tested) to help the driver maintain a consistent path for each test iteration. The driver viewed the string via a real-time video broadcast transmitted from a GoPro camera that was secured to the front bumper.

Installing the DMD on the truck's box required at least three members of the research team. The installation process involved the use of two car jacks placed at each end of the frame to lift the frame upwards onto the undercarriage. At this point the researchers measured the frame with respect to the undercarriage, to align and set the DMD parallel to the truck. Once aligned, the researchers installed the custom brackets to fix the frame in place. The car jacks were removed and the DMD beam slid into place within the frame. The vibration absorption material would also be installed at this point, inserted at all contact areas, between the DMD beam and the frame. The sensors (laser-sensor and gyroscope) were fixed directly onto the beam and torqued to specifications. At the early stages of field testing, the sensors received power from the vehicle; however, this presented fluctuations in voltage causing the gyroscope to be susceptible and output erroneous readings. To have a steady source of power for the sensors, a lithium portable battery was thereafter used to power the four laser-sensors and the gyroscope. Measurements of the sensor physical relation to a reference were recorded for improving the matching algorithm. The pitch angle was measured with a high-accuracy, 0.001°, with a digital level. The vertical distance from the top surface of the laser-sensor encasing was measured to the top plane of the beam with a digital caliper with an accuracy of 0.79 mils (0.02 mm). The horizontal position of each laser-sensor was measured with respect to the center of the beam where the gyroscope was located.

For collecting data, one researcher would be responsible for driving the vehicle at a steady speed and another researcher would perform the calibration and operate the DMS. The researcher would start the DMS software to initiate data collection with the laptop inside the cabin. Once the software started, the calibration was manually performed at the beginning of every test and consisted of applying three impacts to the DMD beam to cause three peaks in all the sensors' measurements. These three peaks observed in the data were used to account for transmission delays from the controllers and the gyroscope. The researcher driving then maintained a constant speed, between 4.5 and 11 mph (2 to 5 m/s).

4.2. Rigid Pavement Testing

During the winter months (December 2018 and January 2019), the research team visited TxDOT's FSF at ABIA in order to calibrate the system and resolve the misalignment of the sensors. By operating on a rigid pavement with very small deflections, it was possible to calibrate the system and estimate the physical parameters of the sensors through the optimization process. The second objective was to keep the DMD system intact after testing, as in the past the system would be disassembled completely for storage after field tests. The original procedure was to unmount the sensors for storage as well as the other hardware. It was later concluded that for every time the system was disassembled, it would result in different misalignment readings. Therefore, after test completion the DMD was unmounted from the vehicle but the sensors and wiring were not removed. This was the most significant drawback for the research team during the last part of the project. For that reason, it is recommended that the beam be mounted onto a dedicated tuck.

TxDOT's FSF, shown in Figure 4-2, provided the necessary rigid pavement to evaluate the best possible scenario for a surface with very small deflections. Several measurements were taken along a 350-ft (107 m) path made of jointed concrete pavement. The slabs were 12.5 ft (3.8 m) in both length and width, with an 8-in. (20.3 cm) thickness.

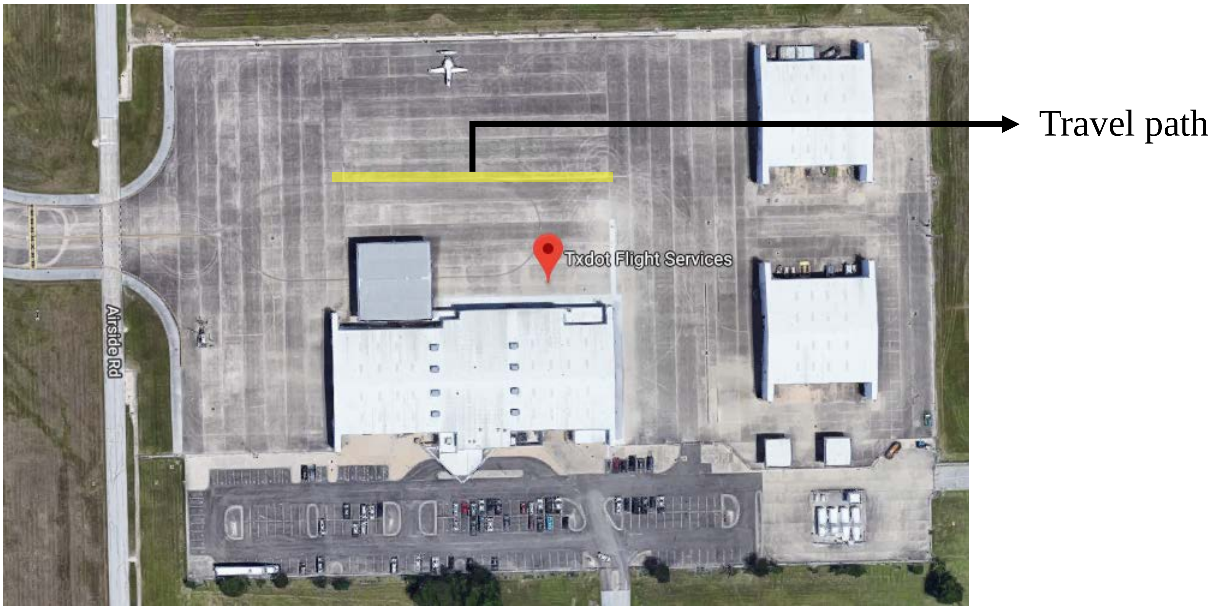


Figure 4-2 TxDOT's Flight Service Facility

A similar field test procedure mentioned in Section 4.1 was followed at the FSF. The average velocity of the DMD for this test was 3.4 to 4.5 mph (1.5 to 2 m/s). Six measurements were taken in all, three in each direction along the travel path (Figure 4-3). After completion of the tests, the DMD frame was removed from the undercarriage and was stored with the sensors and wiring intact.

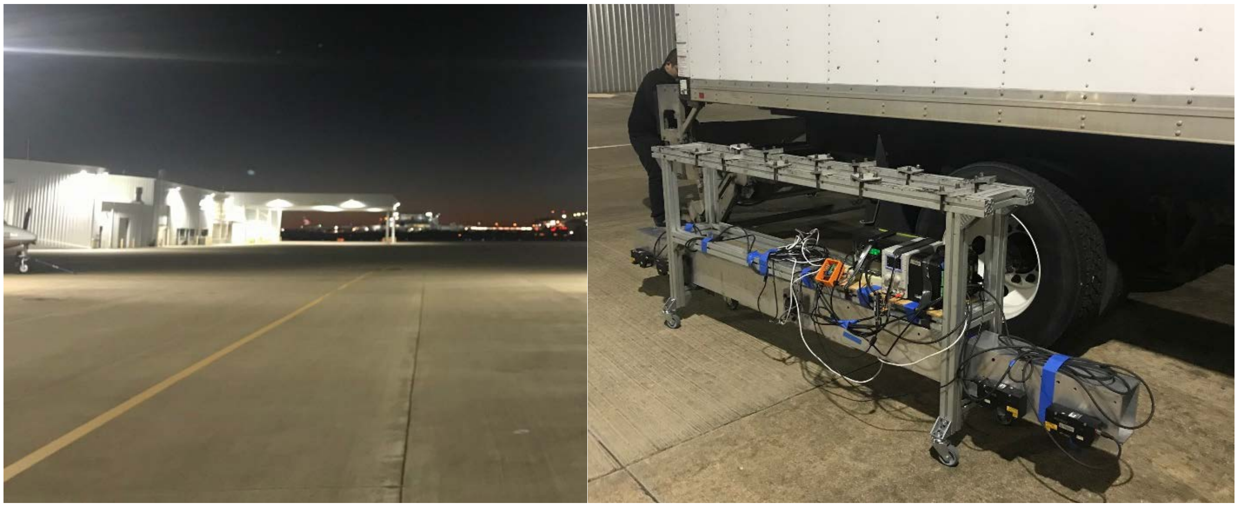


Figure 4-3 Field testing: (Left) section measured with FSF; (Right) DMD assembled with wiring and sensors

Chapter 5. Data Processing for Deflection Estimation

The deflection measurement software (DMS) presented in the previous chapter allowed for the continuous collection of data from multiple laser-sensors along with the gyroscope. This chapter presents the data processing approach. Four laser-sensors were used to obtain the images from the pavement surface with the corresponding rotation of the beam around it center on the vertical plane. Two laser-sensors termed Head A and Head C were placed in the front of the DMD main body (undeflected area) while two other laser-sensors, Head B and Head D, were placed in the rear of the beam (deflected area). The Heads A, C, D, and B are placed, respectively, at 51.7 in. (1,314 mm), 47 in. (1,194 mm), 40.2 in. (1,022 mm), and 57.6 in. (1,463 mm) from the center of the 120-in. (3,050 mm) beam (Figure 5-1).

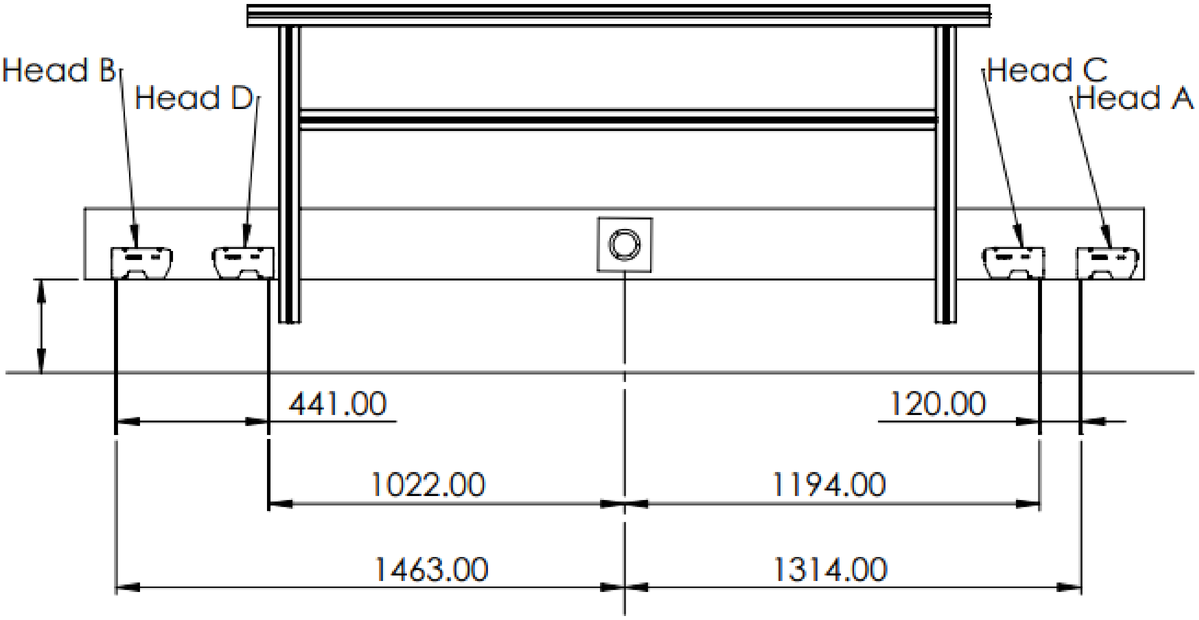


Figure 5-1 Drawing of sensor position on DMD in millimeters

The objective of the image analysis was to compare the distance from the beam to the pavement surface measured by the laser-sensors Heads A and B to estimate the surface deflection by obtaining the difference. However, to do so, it was necessary to first develop a processing system to filter the data, match the images from all laser-sensors, and correct for the external disturbances affecting the system. To be able to extract the useful information, the acquired data was processed through several steps. First, the laser-sensor and gyroscope raw data is loaded, converted and filtered. Then, the data is synchronized and sectioned. Once the data is ready for processing, the data is corrected to account for the rotational movement of the beam. After that, a matching algorithm was applied on the laser-sensor data to locate similar areas for comparison. After matching, the matches of Heads A, C, and D are used to estimate the vertical displacement of the beam and the misalignment of the sensors on the beam. Finally, the pavement deflection is estimated based on comparison of Head B and Head A matches after accounting for the external disturbances. The main steps are depicted in the flowchart in Figure 5-2.

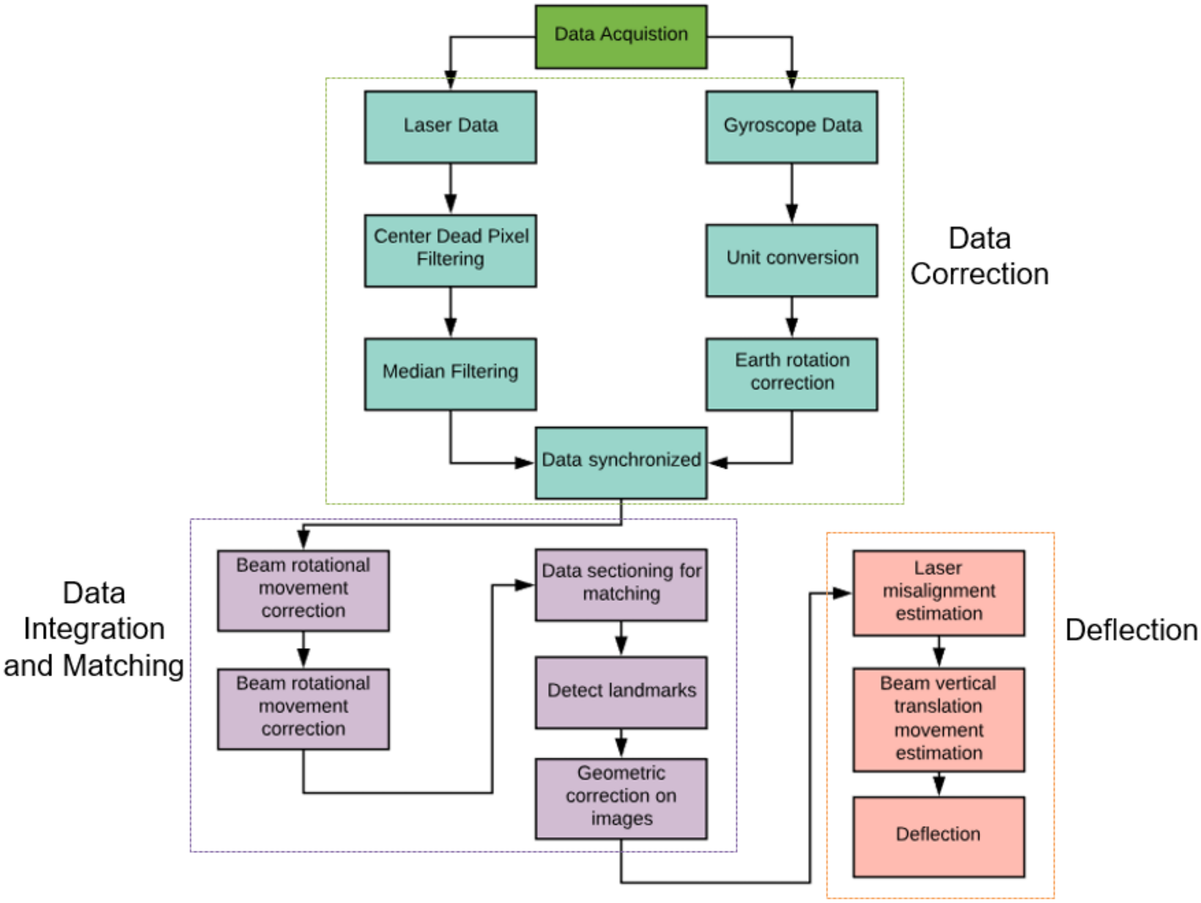

Data Acquistion
Laser Data
Gyroscope Data
Center Dead Pixel Filtering
Unit conversion
Median Filtering
Earth rotation correction
Data synchronized
Data Correction
Data Integration and Matching
Beam rotational movement correction
Beam rotational movement correction
Data sectioning for matching
Detect landmarks
Geometric correction on images
Laser misalignment estimation
Beam vertical translation movement estimation
Deflection
Deflection

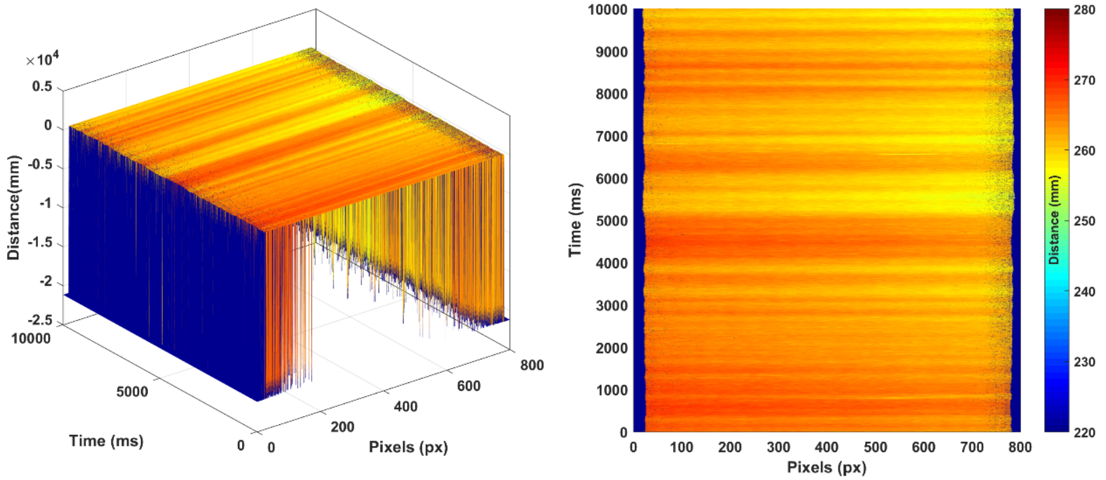
Figure 5-2 Processing flow chart

5.1. Data Correction: Laser-Sensors

Initially, data was acquired using the DMS user interface. Data was saved into a Transition Minimized Differential Signaling (TDMS) file and then imported into MATLAB for filtering [22].

5.1.1. Laser Loading Data

The four-laser-sensor data, saved in a TDMS file, is imported into MATLAB as a structure where the laser-sensor measurements are gathered under different groups. The time series of each pixel is saved under a specific group subfield. The data of each laser-sensor head is then concatenated into a two-dimensional MATLAB matrix of 800 columns. Each one contains the measurements of a pixel that was acquired at a 1.0-kHz rate. Figure 5-3 presents the acquired data of one of the heads along 10 seconds. The data represents the vertical distance in millimeters from a reference line located 11.8 in (300 mm) below the laser-sensor. A 3D view of the data is presented in Figure 5.3 (a) with a scaled 2D image in Figure 5.3 (b). The data set presented columns of dead pixels on the edges with some dead pixels in within the data. The filtering process to extract the pavement scan from the acquired data is explained in detail in the next subsections.

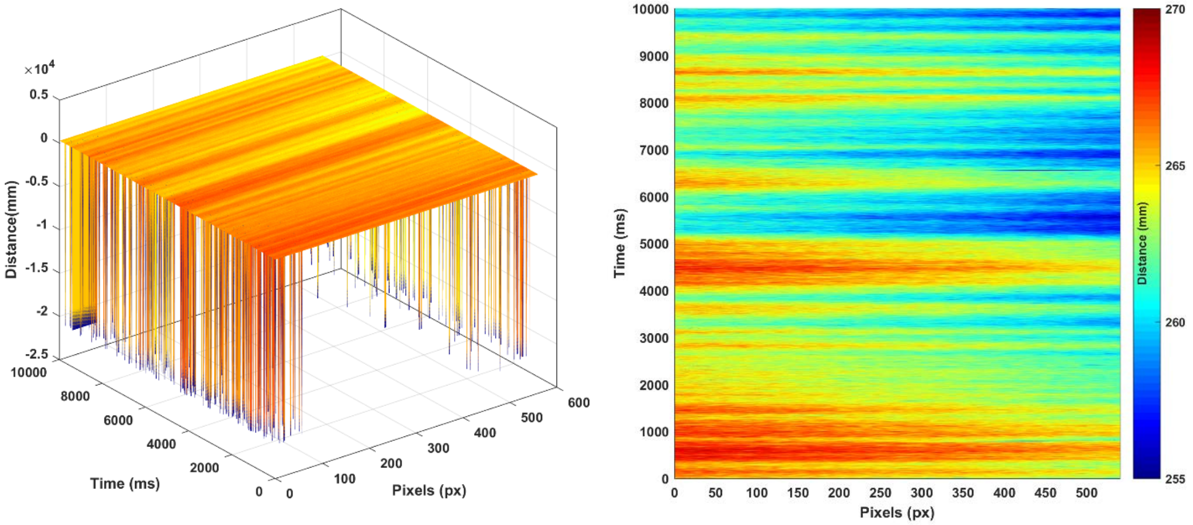


(a) 3-dimensional view *(b) 2-dimensional view*

Figure 5-3 Laser-sensor raw data for 10 seconds

5.1.2. Edges' Dead-Pixels Filtering

Due to the configuration of the laser-sensor triangulation system, the laser reflection at the edges is not properly captured. Therefore, portions of the signal are not received back by the camera. These points are set to a fixed low off-scale value by the laser-sensor interfacing software. They appear as dead pixels on the edges of the data, shown as dark blue areas in Figure 5-3. Thereafter the data matrices were resized to eliminate the columns corresponding to the dead-pixel areas. The results of the edges' dead-pixel filtering are presented in Figure 5-4. The head data was resized to 540 pixels wide.



(a) 3-dimensional view *(b) 2-dimensional view*

Figure 5-4 Edges dead-pixels results

5.1.3. Center Dead-Pixels filtering

As presented in the previous subsection, the acquired data also contains dead pixels in the middle due to deviated reflection of the laser light. This might also be due to sharp edges in cracks and aggregates on the pavement. A second noise filter was designed to detect the central dead pixels and replace them by the mean value of the non-dead pixels in the surrounding scanlines. The obtained results are satisfactory without loss of information or alteration to the general structure of the scanned pavement. Figure 5-5 shows the outputs of the second filtering step.

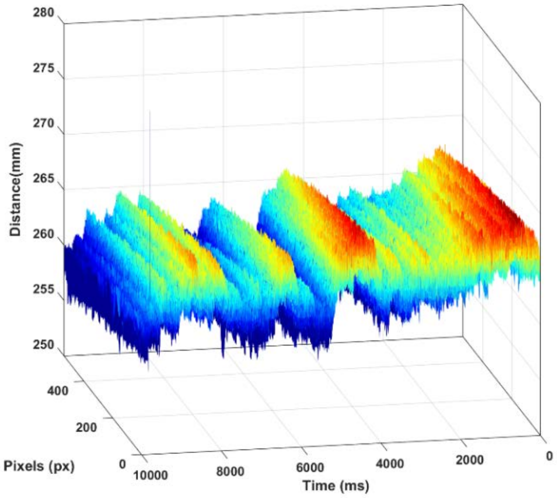


Figure 5-5 Dead-pixels filtering

5.1.4. Median Filtering

The last filtering step was aimed at filtering noise impulses using a median filter. The median filter is widely used in image pre-processing as it preserves the signal while removing the spikes. Several trials were performed to evaluate the window size that would eliminate the spikes with minimal impact on the signal. A filter size of seven-by-seven was applied along each profile. It runs point by point through the profile and replaces the distance measurement of the current pixel by the median of the filter window. The filter window consists of seven entries in each direction and is centered at the point of interest. The obtained results are illustrated in Figure 5-6.

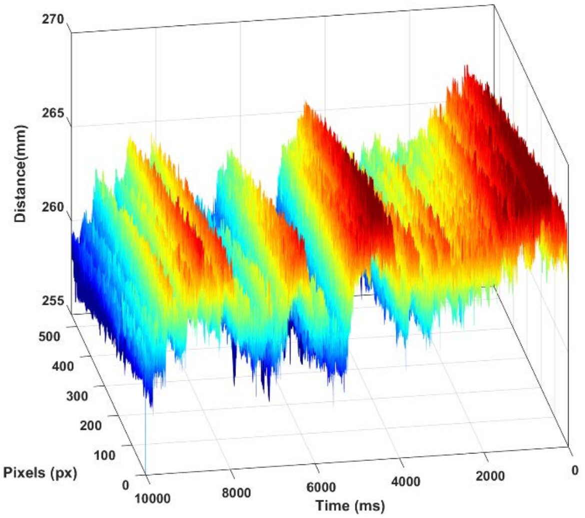


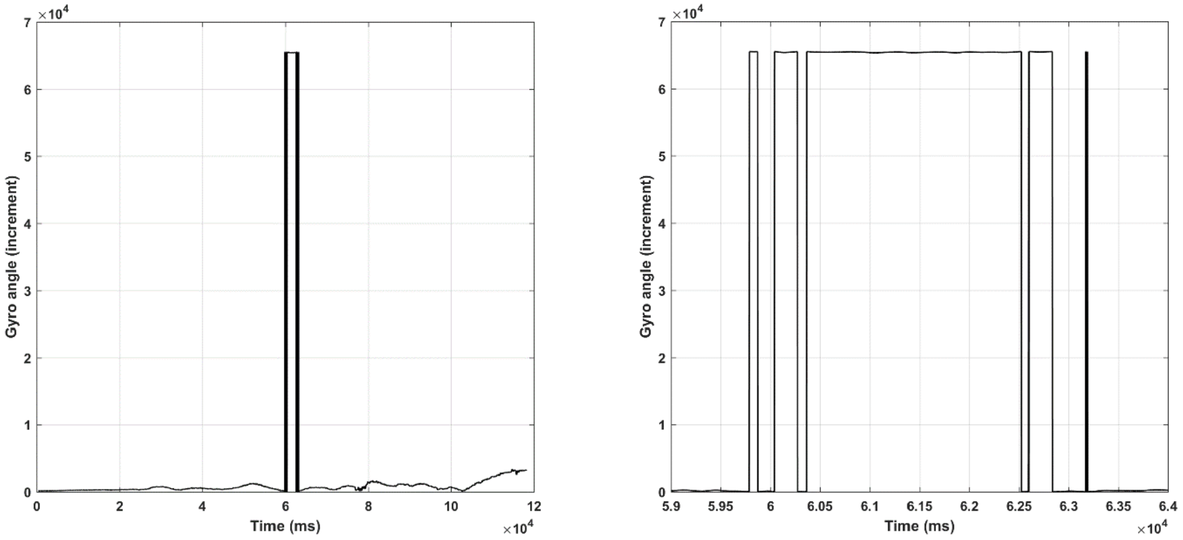
Figure 5-6 Median filtering results

5.2. Data Correction: Gyroscope

The gyroscope data was also acquired using the DMS user interface and saved in a TDMS file.

5.2.1. Gyroscope Data Loading

The data was imported into MATLAB as a vector of strings. Each string contains the gyroscope measurement with the corresponding reading time. Figure 5-7 presents the acquired gyroscope data along 120 seconds at a 500 Hz frequency. The data represents the change in the gyroscope angle with respect to the initial plane in terms of unit increment. As observed, the data presents a wrap-up due to negative changes or high values in the gyroscope angle. This is due to the fact that the increments written on 16 bits can only vary between 0 and 65,535.



(a) Gyroscope data along 120 seconds *(b) zoom in the gyroscope data*

Figure 5-7 Gyroscope raw data

5.2.2. Wrap-up Correction and Angle Unit Conversion

As shown in the previous subsection, it was observed that the gyroscope measurement had to be preprocessed to extract the angle variation. First the wrap-up due to the bits' limitation was corrected. Then, the angle was converted to degrees following the manufacturer equation, given by:

$$Gyro(°) = Gyro(increment) * \frac{1.113065}{3600} \tag{5.1}$$

where,

Gyro (°)	=	Angle in degrees
Gyro (increment)	=	Raw value from the gyroscope

Figure 5-8 illustrates the resulting gyro angle from the data preprocessing. However, from the initial period corresponding to the stationary scanning, the angle drifts instead of being constant. This is due to the earth's rotation that is measured with the gyroscope.

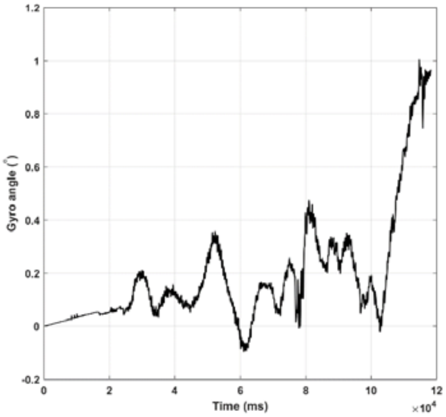


Figure 5-8 Gyroscope angle

5.2.3. Earth Rotation Correction

Based on the geographical location of the experiment, the gyroscope angle is corrected to eliminate the earth's rotation measured in addition to the beam rotational movement (Figure 5-9). For the Innovation Boulevard site at UT's Pickle Research Campus, the correction was done using the following equation:

$$Gyro(°) = Gyro(°) - 3.5 * 10^{-6} * (Gyro\ time(ms) - Gyro\ ini\ time(ms)) \qquad (5.1)$$

where,

Gyro (°) = Angle in degree

$3.5 * 10^{-6}$ = Value subject to the direction the system is facing (can be computed from pitch, roll and heading angle)

Gyro time = Sampling time of the gyroscope in milliseconds

Gyro ini time = Gyroscopes initial time in milliseconds

The earth's rotation equation contains a value that is dependent on the direction the gyroscope is facing. This can be seen in Figure 5-9 in the black line plot where the first two seconds should be a horizontal line (as seen in the red line plot)—but due to the earth's rotation, it has a positive slope. The constant value is found by identifying the slope of the line and subtracting it from the gyroscope measurement.

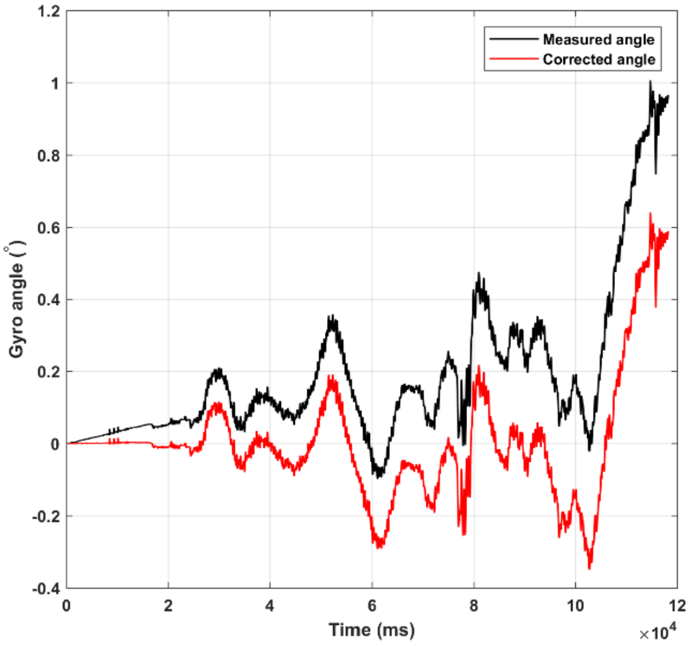


Figure 5-9 Corrected gyroscope angle

5.3. Data Integration and Matching

Once the laser-sensors and the gyroscope data are loaded and preprocessed, all sensors' measurements are synchronized based on their reading time. Then the laser-sensor data is corrected to account for the beam rotational movement using the gyroscope measurements as shown in Figure 5-10.

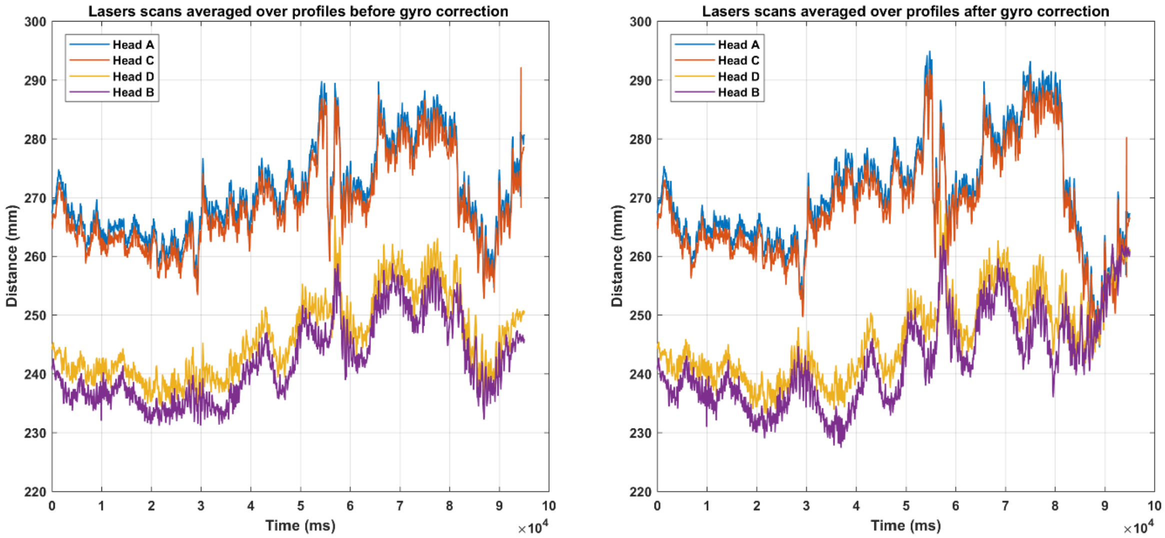


Figure 5-10 Beam rotational movement correction

5.3.1. Data Matching

At this time, the deflection measurement device consists of at least four laser-sensors, two heads to collect data from an undeflected area while the two other heads collect data from a deflected area. The fourth laser-sensor, Head D, has also been installed but this is used for calibrations purposes.

To measure the surface deflection of the road caused by a heavy vehicle, the distance from the beam to the pavement surface is measured by both Heads A and B and then compared after disturbance corrections. Therefore, locating similar areas in the measurement acquired from the laser-sensor is critical for the deflection estimation. However, trajectory changes and speed variations made the matching more challenging. To achieve this objective, the A matching algorithm was selected because it is widely used in computer vision applications [23]. It is a patented local feature detector and descriptor that is robust to image transformations. The algorithm consists of three main steps: interest point detection, local neighborhood description, and points matching.

5.3.2. Contrast Enhancing

To match the pavement data from different laser-sensors, each data set was subdivided to sections of a few thousand profiles. The submatrices are converted to ".png" images to be analyzed. Then, the features on images are enhanced through colors adaptive equalization filter. To illustrate the matching steps in this section, 3,000 profiles of 400 pixels from Head A data with 4,000 profiles of 400 pixels from Head C data were considered.

5.3.3. Features Points Detection

Each pixel on the two images is analyzed and its gradient variations are defined in all directions. Feature points are defined by comparing the gradients variations to a defined threshold. Then, points with similar surrounding variations are matched, together representing the same feature on two different images as shown in Figure 5-11.

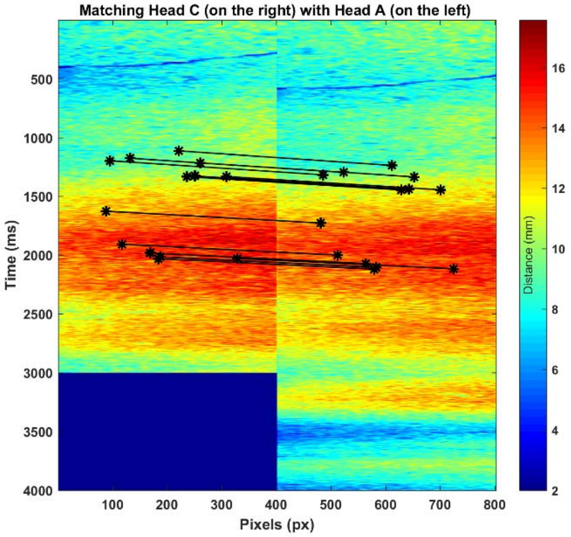


Figure 5-11 Feature matching

It is important to mention that features detection is also performed on pavement sections without significant features such as cracks or loose aggregates. The choice of a section with a clear abnormality was just for visibility and better appreciation of the procedure.

5.3.4. Geometric Correction

Based on the previous results, a geometric correction to the data sections based on the matched points using the A algorithm was implemented (Figure 5-12). The measurements from Heads A and C were matched after scale and rotation correction based on the location of the points of interest.

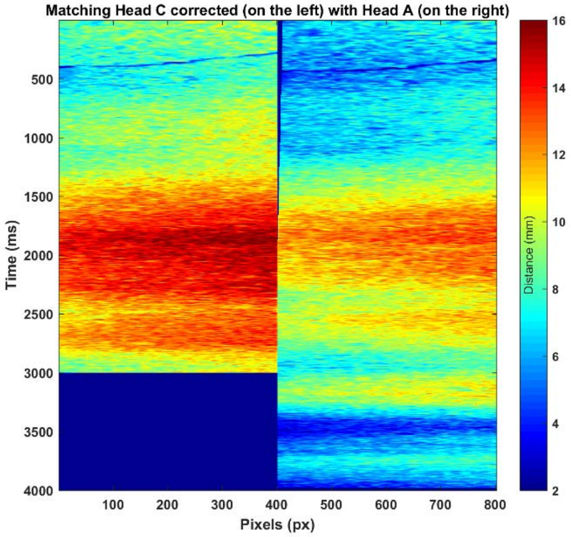


Figure 5-12 Matching geometric correction

A geometric transformation matrix is calculated for each section of the image based on the coordinates of the matched features. The resulting geometric transform matrix obtained is a combination of rotation, translation and scaling matrices.

The function implements an optimization problem to solve for the transformation matrix given the fixed point (or the features from the image of Head A), and the moving points (or the matching features from images of Head B, C and D). Once obtained, the transformation matrix is converted to a 2D transformation object in MATLAB and sent to another function to apply the transform on the section from the second image.

The geometric transformation matrix obtained from solving the optimization problem uses 15 best matched points from a section and is applied to that entire section. The imwarp() *(9)* function of MATLAB was used to perform this operation in this algorithm. The function takes in as input the section from the second image and its affine transformation object created previously, and outputs the transformed section. These transformed sections together form the new image which is the matched version of the original image and has the same points as image A at the same coordinate positions. This step does produce some dead pixel areas in the matched version of the image because of the transformations that are applied to match it with image A. These dead pixels are cropped before further analysis, an explanation is given in section 5.3.6.

The matching procedure is applied along the data time series of Heads C, D and B. An example of the resulting matched sections to Head A are presented in Figure 5-13.

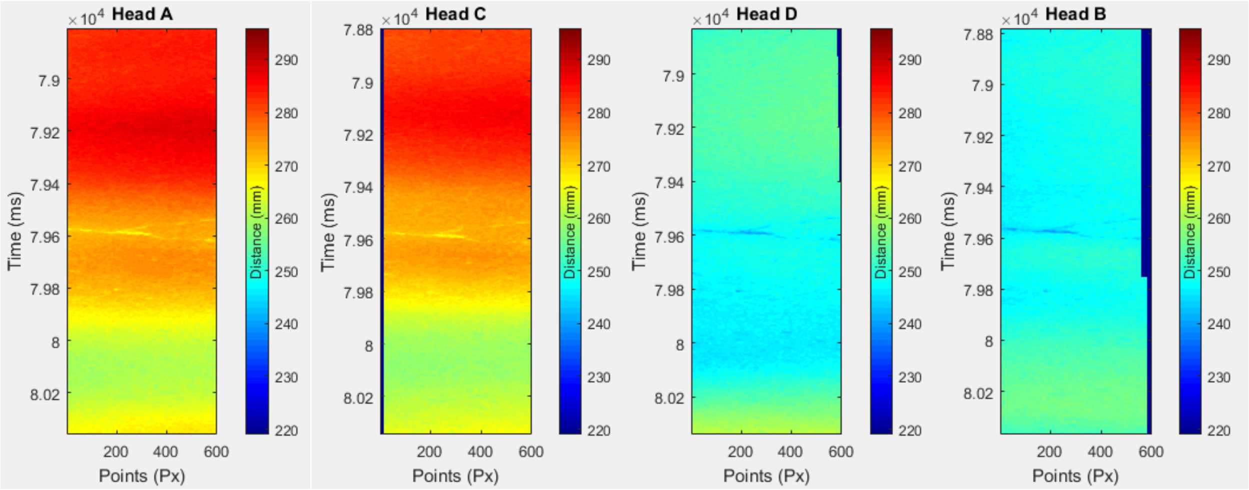


Figure 5-13 Example of a crack in the asphalt seen in all plots after matching

5.3.5. Matching Fine Tuning

The matching algorithm performs very well in pavement with uniform surface characteristics with no abnormalities. However, due to sectioning the images of the laser-sensor data into smaller images, there are unaccounted errors. To check the goodness of the matched images, the mean absolute error (MAE) was used. By taken the sum of the absolute differences between two profiles and averaging it over the number of points in the profile. The overall error is taken as the median of the MAE for all the profiles in the image.

5.3.5.1. Adjustments

1. Tuning the section/subsection size parameters of the matching algorithm

   As discussed earlier, the matching algorithm works by finding features in the images to match the reference image (Head A) and transforms the moving images. If the image does not have any good features to match the two images, the algorithm fails to match the images correctly. In this case, it attempts to finds features with similar surrounding variations without a limit on the threshold. This scenario results in poorer matching. From experiments, this has occurred in the FSF tests mentioned in Chapter 4.2 and when selecting a small section size for the matching. For FSF, it occurred because the concrete surface did not have unique features, only longitudinal tining. For the case of small sections, the operator has control over the size of the section but selecting a small size project restricts the feature search to a small area. On the other hand, a section size that is too large will lead to lower matching accuracy by applying a single geometrical transformation to a large image. This is also unacceptable, especially in cases where the truck is known to have variations in speed while scanning the pavement. Hence, there is an optimal section size.

   Besides the section size, the subsection size also plays an important role in the accuracy of matching. The closer the subsection size is to the section size, the less overlapped are the consecutive sections for matching. The less the overlap between sections, the more distinct features the algorithm selects for matching. This sometimes results in significantly different transformations for consecutive sections resulting in highly distorted images. Coupled with a small section size, a large subsection size can make the matched image almost unusable by letting the algorithm select distinct bad features every time. On the other hand, the smaller the subsection size is, the more the overlap between sections and more similar features will be selected in consecutive sections. This results in similar geometric transformations for consecutive sections thus preventing distortion. However, too large overlaps in the sections while matching sometimes results in overlaps in the matched image too, which lowers the accuracy of the matching especially at the junctions of the sections where the same profiles are repeated.

   To select the best section and subsection size, a preliminary matching is performed with different settings and the errors is minimized at each iteration.

2. Adding a profile-profile matching routine to the existing process

   Following what has been described about the geometric transformations, the algorithm does not evaluate if the individual profiles are exactly matched. Instead it aims at best matching the features that were selected and, hence, the section as a whole. Intuitively, this means that the matching can be improved for each section.

To confirm the idea, several different sections were analyzed from the images of Head A (A) and Head C (C) focusing on one profile at a time. One such section is shown in Figure 5-14. Several data sets and section sizes showed that setting the line search limit to five pixel shift is necessary to improve the accuracy within the sections. How this creates a problem is illustrated by Figure 5-15, which shows a profile from A plotted with a profile from C, after shifting the profile two pixels the correct profiles were compared. The error values are significantly different between the two profiles especially considering that we need to estimate deflections. Such discrepancies can be found throughout the data.

To account for mismatched profiles in the section, a routine is added to the existing process that uses the images outputted from the matching algorithm. To match the image of A with image of C perfectly, that is, profile by profile, the algorithm checks the first to the last profile in A and compares the profiles from C. The algorithm has a search limit for the profile shift so as not to match profiles that are physically incorrect. The appropriate profiles are selected by minimizing the MAE between A and C. After line searching every profile in A and selecting the appropriate profile in C, the best possible solution of C can be compared to A.

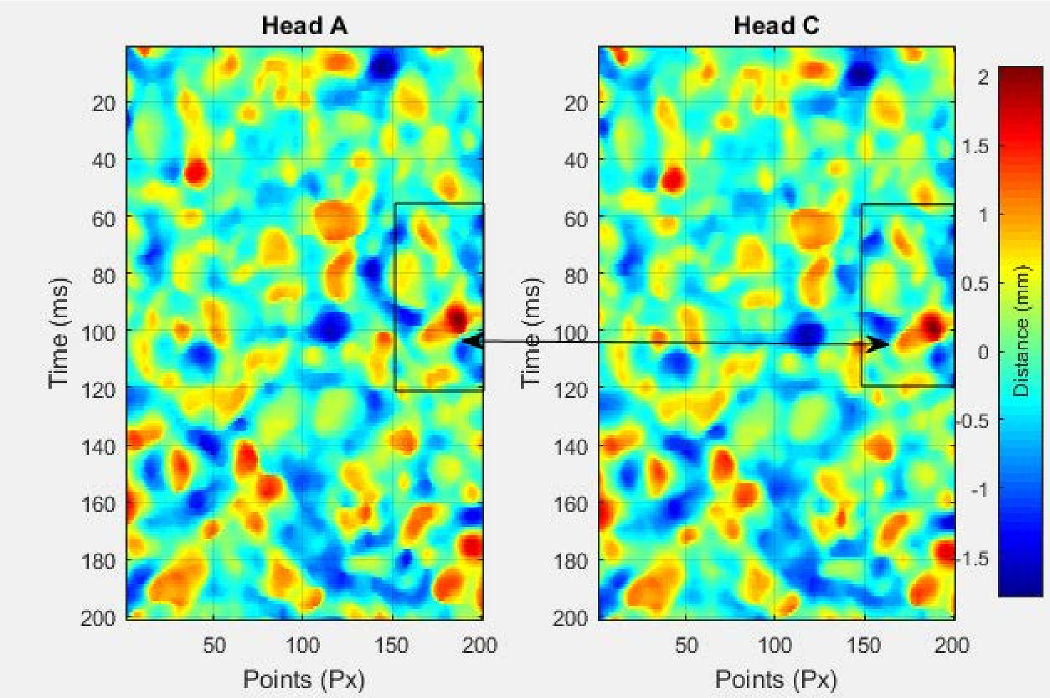


Figure 5-14 Matched sections showing individual profile shift

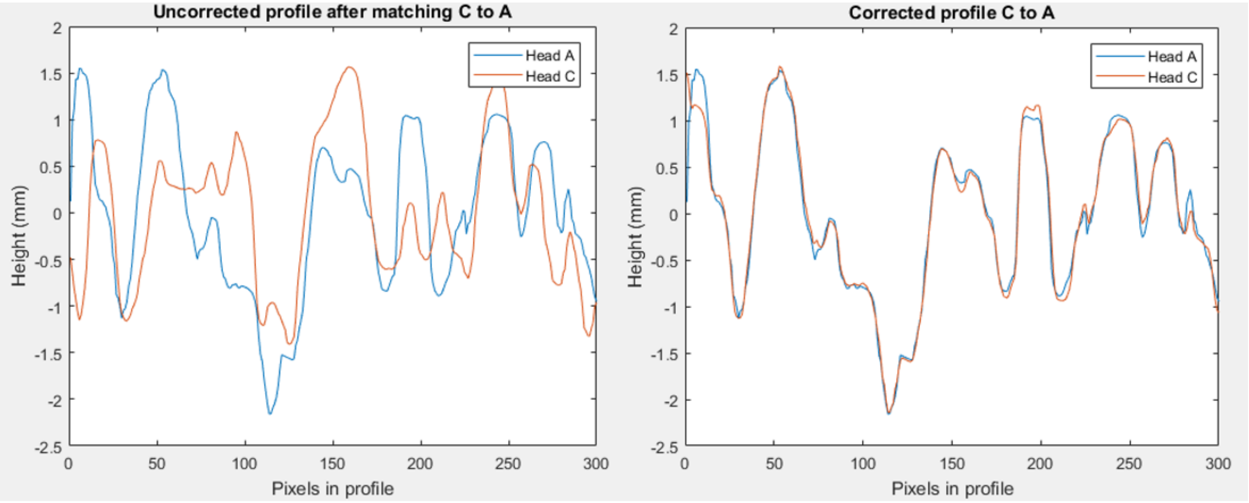


Figure 5-15 Matched profiles: (Left) from matching algorithm; (Right) after profile-profile matching

3. <u>Increasing the sampling rate of the laser scanners</u>

   For the majority of the field tests performed for calibrating the algorithm, the sampling rate for the laser-sensors was 1.0 kHz and the truck speed was 2 m/s with +/- 0.2 m/s uncertainty. Under this configuration the profiles are approximately spaced by 2 mm. Given that the four laser-sensor's scan the pavement independently, it is possible that the different lasers may scan a different profile from within that 2 mm region of the pavement. In sampling at this rate, it was assumed that the pavement does not vary significantly within the 2 mm. A spacing greater than 2 mm increases the error for profile by profile matching.

   To account for this type of error, the DMS was re-programmed to handle up to 4.0 kHz sampling rate from the four laser-sensors. The truck speed was maintained at 2 m/s, reducing the 2 mm region to 0.5 mm. By reducing the profile spacing region, the matching had a greater resolution and greater accuracy. The dataset with the 4.0 kHz was evaluated to observe the differences and the improvement.

5.3.5.2. Results

The results obtained by applying the above methodology is presented below and compared to the results from the original process. Images of Head A and Head C were used to explain the solution but the remaining images showed similar results.

5.3.5.2.1. Optimizing Section/Subsection Size

Table 5-1 presents the error values for the different section and subsection sizes tried with the matching algorithm.

Table 5-1 Error values corresponding to different section/subsection sizes

Section/Subsection combination	Median MAE (mm)
1000/1000 (original)	0.1732
1000/100	0.1700
500/500	0.1718
500/250 (optimal)	0.1461
500/50	0.1600
100/100 (highly distorted)	0.1238

One can notice that the error decreases as the section size is smaller, this is because the matching algorithm has more flexibility to apply the transformations. However, the images with very small section sizes, as 100 profiles, are highly distorted and do not provide enough data which is needed later for deflection estimation. The same situation also occurs when using larger subsection sizes which are almost equal to the section size. The optimal combination of section/subsection size that not only provided a significantly lower error but also provided less distortions is 500/250 with an error value of about 0.146 mm.

5.3.5.2.2. Profile-Profile Matching

This algorithm searches five profiles above and below each matched profile obtained from the matching algorithm to find the one with the least error. When considering all the profiles in the data set, 120 meters of pavement, the overall error changed from 0.146 mm to 0.119 mm for the selected section/subsection size of 500/250.

5.3.5.2.3. Effect of Sampling Rate

The increased sampling rate significantly brought the error in matching down from around 0.119 mm to around 0.07mm, or 70 microns, which is lower than the target set as the research objectives, therefore, it was considered very successful. Other analysis was performed to evaluate how the sampling rate affects the synchronization and whether data is lost. This analysis is described in Appendix A.

5.3.6. Distance Estimation

As introduced at the beginning of this chapter, the objective of the data processing algorithm is to estimate the deflection of the pavement surface by comparing the distance measured from the beam to the pavement surface by Head A to the one measured by Head B. After running the matching algorithm with the corresponding geometric image correction, the resulting corrected profile of Heads C, D and B presented some dead areas due to the correction process, as illustrated in Figure 5-13. Therefore, before proceeding to estimate the average measured distances A, C, D and B, the matches were resized to keep only the areas with useful information. Then, an average distance profile for each head was estimated. For that, the moving average algorithm was considered with a window of three profiles. For each head, the average distance was computed by averaging the measurements of a centralized window of three profiles considering all pixels along each profile.

Figure 5-16 presents the estimated distances of Heads A, C, D, and B for the pavement sections matched in Figure 5-16. A series of examples is presented in Appendix D.

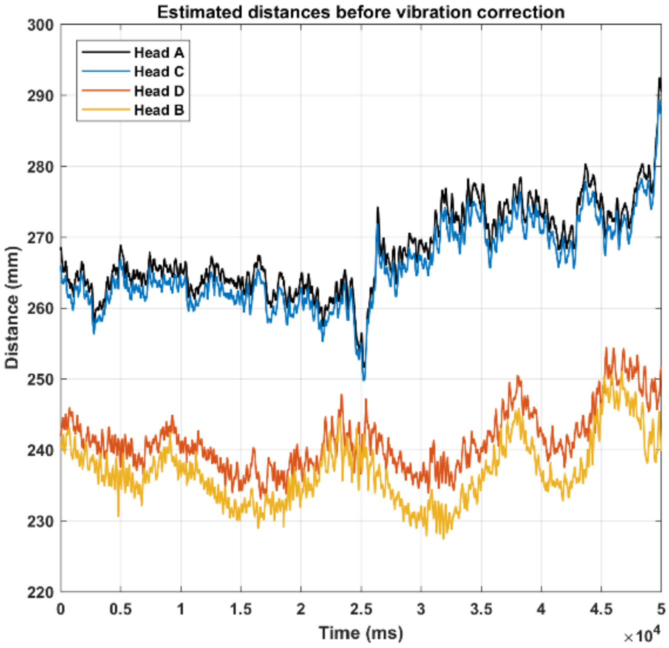


Figure 5-16 Measured distances

By analyzing different sections from the profiles of Figure 5-16, it was concluded that the proposed data processing algorithm matched the areas from Heads C, D and B to the respective areas from Head A. Indeed, the details of pavement structure were recovered which proved the efficiency of the matching and the distortion correction.

The previous results demonstrated that the performance of the matching algorithm is satisfactory to match the pavement profiles from the different laser-sensors within the desire accuracy. However, the pavement deflection estimation would require further data corrections. It was also concluded that the measured data is affected by some errors due to the mechanical structure and the vehicle vibrations. This problem has been exacerbated as the research team does not have a project-dedicated vehicle so a new set of corrections have to be developed every time that testing is performed. In the original proposal, the research team proposed the purchase of a dedicated vehicle but this was later removed at the request from TxDOT’s RTI Office.

5.4. External Errors Correction for Deflection Estimation

After several experimental tests, the different sources of external disturbances affecting the laser-sensors measurements were finally identified. As mentioned previously, the main challenge in this research project is defining a reference or datum for measurement to be able to compare the different laser-sensors scans. The research team was able to correct for the rotational movement of the beam by including a gyroscope measurement. However, to reach the required accuracy, the team also needed to estimate the vertical translational movement by introducing a third laser-sensor (Head C) placed close to the front head (Head A) in the undeflected area. Another source of error was the misalignment of the different sensors which is due to the uncertainty in the mechanical fabrication.

5.4.1. Vertical Vibration Estimation

The laser-sensor, Head C, was changed several times to find the correct distance away from the center of the beam. After several iterations, it was decided to place the sensor very close to Head A to measure the instantaneous vertical vibration of the beam. This was done in order to compute the accumulated distance that the beam moved vertically between the time where the two Heads A and C scanned the same area. From these consecutive accumulated distances, the instantaneous beam vertical translational movement is estimated as shown in Figure 5-17. The optimization problem was solved with a quadratic method and the comparative results between the corrected distances are presented in Figure 5-18.

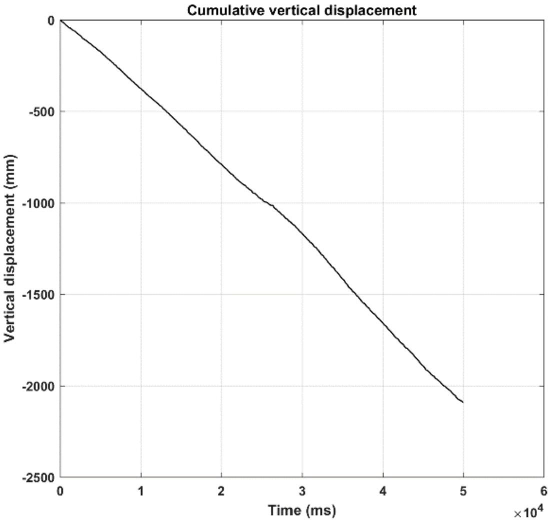


Figure 5-17 Beam vertical displacement

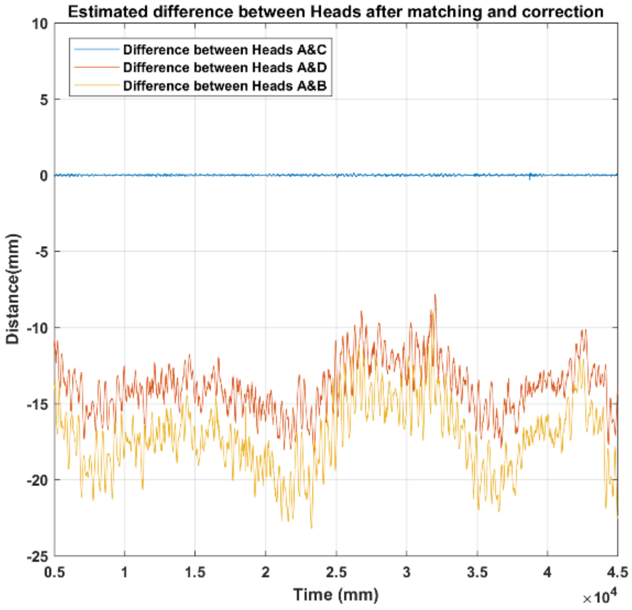


Figure 5-18 Differences between head distances

From the error graph in Figure 5-18, the obtained results are far from the expected 100 microns. However, it can be observed that correcting for the misalignment is mandatory to be able to measure the deflection properly with the correct magnitude.

5.4.2. Misalignment Estimation

From the experimental results that were obtained after correcting for the translational and rotational movement of the beam, it was concluded that the misalignment of the different sensors with respect to a common plane must be defined with a micron accuracy. For this determination, simulated errors were introduced to the data to observe the affect the misalignment has on the final deflection error. More importantly, this was also done to define if the errors are linear or non-linear. The simulated errors included the six degrees of freedom of each laser-sensor. The quadratic optimization was sensitive to any height changes in A and C but it was concluded that the errors were linear, which is a significant benefit because these errors are easily corrected. It was confirmed that the failure to measure misalignments to a micron precision results in errors of up to 20 mm as seen in Figure 5-18. Yet, given the linear behavior of the misalignment, the misalignment values can be found numerically.

5.5. Advancing the Optimization

The purpose of the field test in the FSF was to use a different optimization, resolve the unaccounted misalignments and to improve the results of the vertical displacement of the center of the beam. The distance between Head C and A was an important distance that had a large effect on the result of the vertical displacement of the beam. The closer Head C is to A causes sensitivity to noise, therefore Head C was placed 153 mm further apart of A.

The data collected from FSF was cropped to a time frame where the velocity of the vehicle was constant, with a 0.5% error, to have consistent time differences between the sensors, as seen in Figure 5-19. The FSF region where we collected data had jointed concrete pavement with a featureless surface. The matching algorithm was not capable of distinguishing unique features in the concrete and the data had to be analyzed differently by considering the distance between the concrete joints.

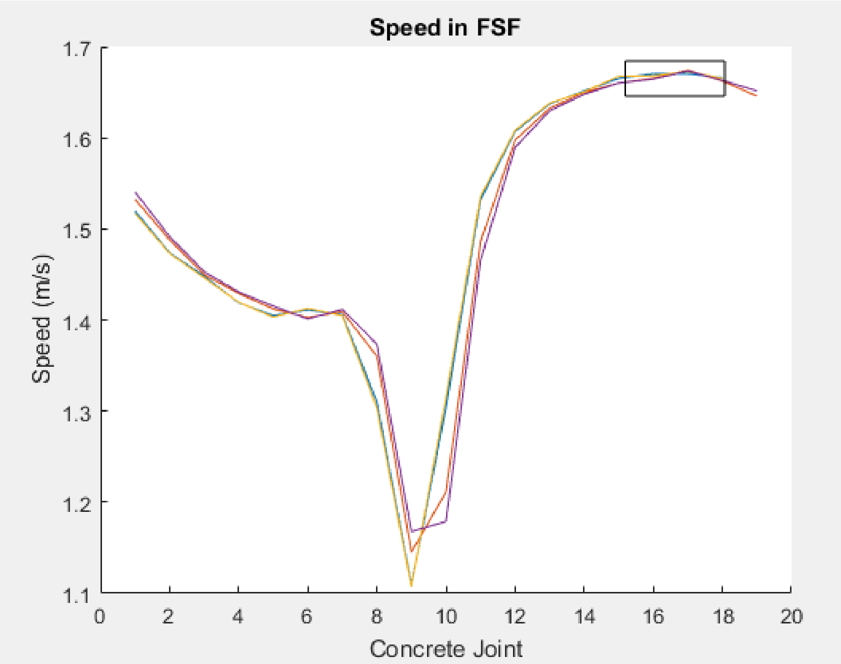


Figure 5-19 Speed data to identify section where the speed remained constant

The order of the laser-sensors and the time it took for a certain laser unit, or head, to pass by the first head (Head A), as Figure 5-20 illustrates. Head A is first and then, 70 ms after, Head C passes the same feature. It can be observed in the plot with the distinctive jumps/drops caused by the concrete joints. Furthermore, the two laser-sensors in the rear of the DMD (Head D/B) can be seen passing the same feature almost two seconds later. From the same graph (Figure 5.18) it was learned how the height measured by each laser behaves when there is little beam rotation compared to pavement with large wavelengths.

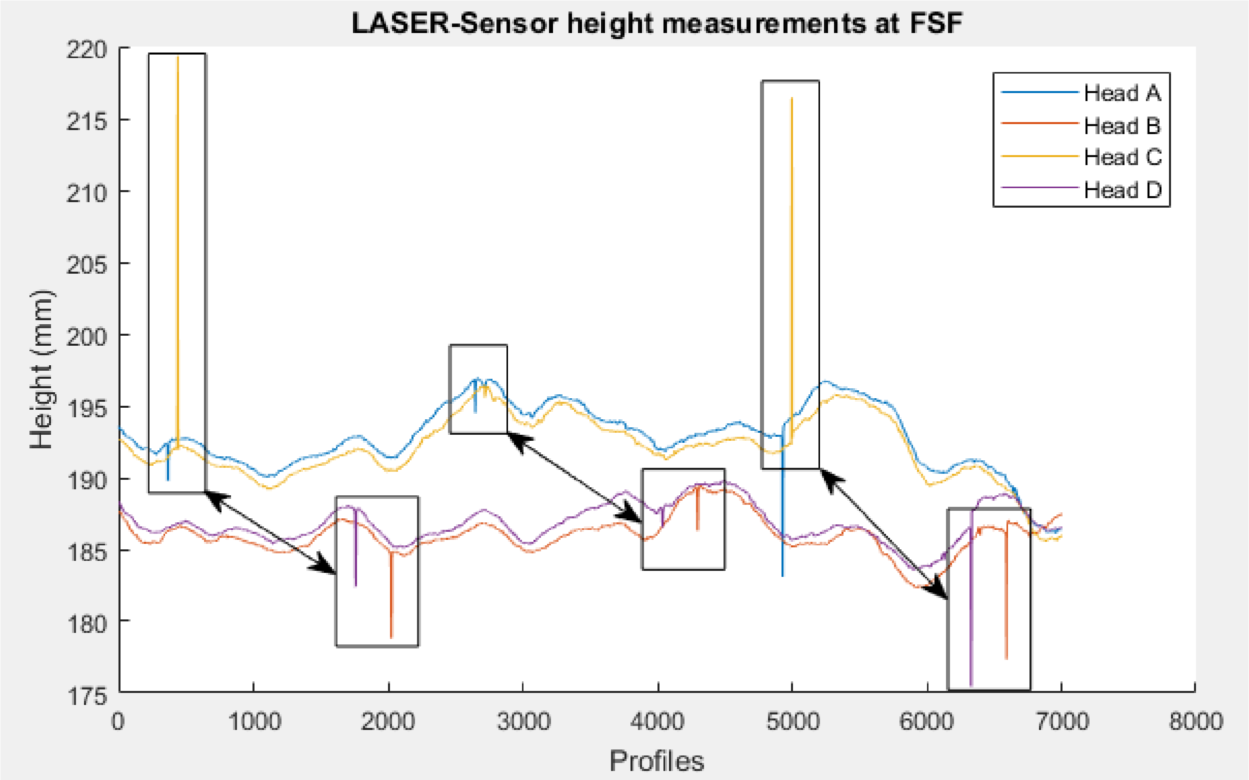


Figure 5-20 Head A and C see a feature first and ~ 1.5 seconds later Head D and B pass by the same feature; peaks are concrete joints

The analytical solution used previously relied on the minimum norm solution of the underdetermined system to compute the time series of the vertical velocity of the center of the beam from Head A and Head C measurement data. The objective function was to minimize the norm of x, $\| x \|$, subject to Equation 5.1 to find the best approximation for the velocity, x. Knowing the time in milliseconds, a, that it took Head C to get to Head A and knowing the displacement difference between the two laser-sensors, b, the velocity can be reconstructed. The notation in the following equation use two letters referring to the two sensors used; Head A and Head C are denoted as AC. The physical equation is expressed as Equation 5.2, where the displacement is equaled to the velocity multiplied by the time. In the minimum norm, to find the minimized solution, Equation 5.3 was used.

$$a \cdot x = b \tag{5.1}$$

where,

a = Matrix of the time difference in milliseconds between sensors

x = Unknown vector of vertical velocities

b = Vector of the displacement between sensors

$$Hc - Ha = \sum_{i=0}^{time\ between\ C\ and\ A} v_i \cdot t \qquad (5.2)$$

where,

Hc = Height of laser-sensor C

Ha = Height of laser-sensor A

V_i = Velocity

T = Time

The least norm solution to Equation 5.1 is computed using the classical formula:

$$X_{lm} = a^T (a \cdot a^T)^{-1} \cdot b \qquad (5.3)$$

where,

X_{im} = Resulting velocity

a = Matrix of time

b = Vector of displacement

After finding the velocity based on Head A and C, the displacement can be reconstructed and compared with the original displacement obtained by the two laser-sensors. The results can be observed in Figure 5-21. Furthermore, the unknown misalignment values are accounted for in the matrix for time and are solved for by the optimization.

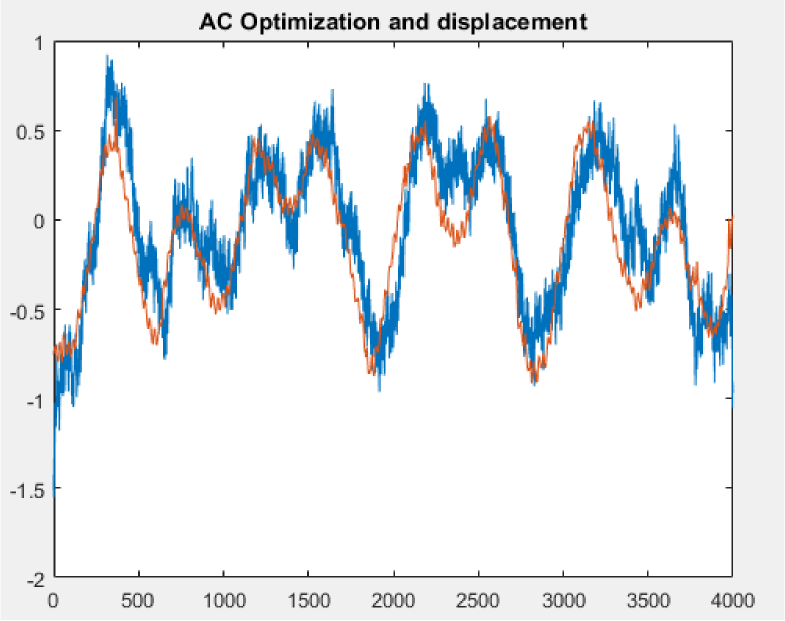


Figure 5-21 Head A and C used to reconstruct displacement (red) and compared to actual (blue)

To reconstruct the data with a numerical solution for the vertical movement and the misalignments, more equations were added: 1) Head A and Head C, 2) Head A and Head D, and 3) Head A and head B, notated AD, AC, and AB respectively. A larger data set was used from the flexible pavement testing to observe how the new optimization improves the deflection estimation. The optimization uses the matrices and vectors structured from the minimum norm mentioned in Equation 5.1. The equation to minimize can be seen in Equation 5.4.

$$\| A_{AC}X - b_{AC} \|^2 + \| A_{AD}X - b_{AD} \|^2 + \| A_{AB}X - b_{AB} \|^2 \tag{5.4}$$

where,

A = Matrix of Head A and Head C, D, and B; time

B = Vector of Head A and Head C, D, and B; displacement

X = Velocity

One of the issues associated with least squares solutions to overdetermined problems is their lack of regularity. The solutions tend to exhibit very large variations over time, which are unrealistic, since the rate of change of the velocity is equal to the vertical acceleration, which is bounded since the maximum reaction forces are limited by structural constraints. To minimize the variations of the vertical velocity over time, we added a regularization term corresponding to the L1 norm of the variations of the velocity over time. This approach is similar to Compressed Sensing or LASSO. The problem was solved with CVX, a package for specifying and solving convex programs [24, 25]. Figure 5-22 illustrates how adding the regularization term to the equation reduces the noise.

The regularization term can be written as follows:

$$\lambda \cdot \sum_{i=2} | V_i - V_{i-1} | \tag{5.5}$$

where,

λ = Parameter to control the weight of the regularization term

V = Velocity

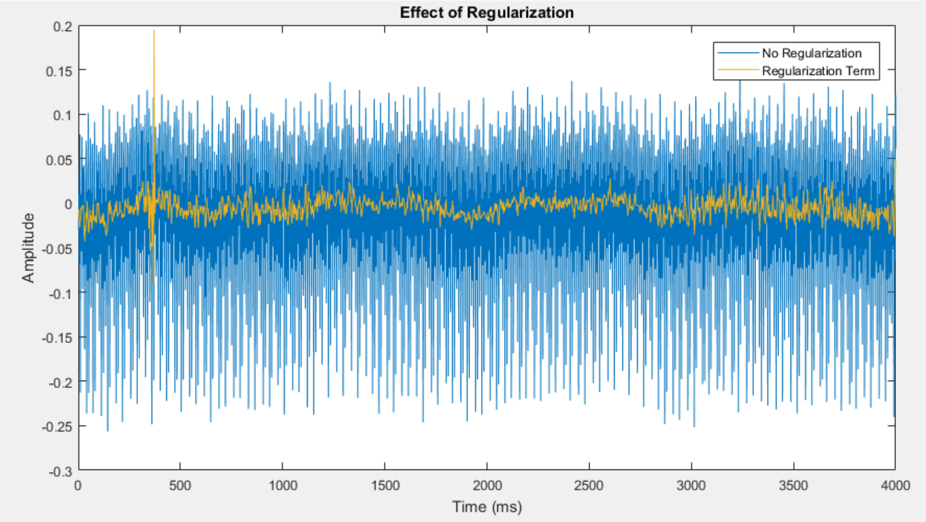


Figure 5-22 Adding the regularization term to the equation reduces the noise.

After obtaining the velocity of the center of the beam, the velocity was integrated to obtain the vertical movement of the beam (Figure 5-23). The slope of this vertical movement is contributed to the initial angle for the optimization and the slope of the road profile.

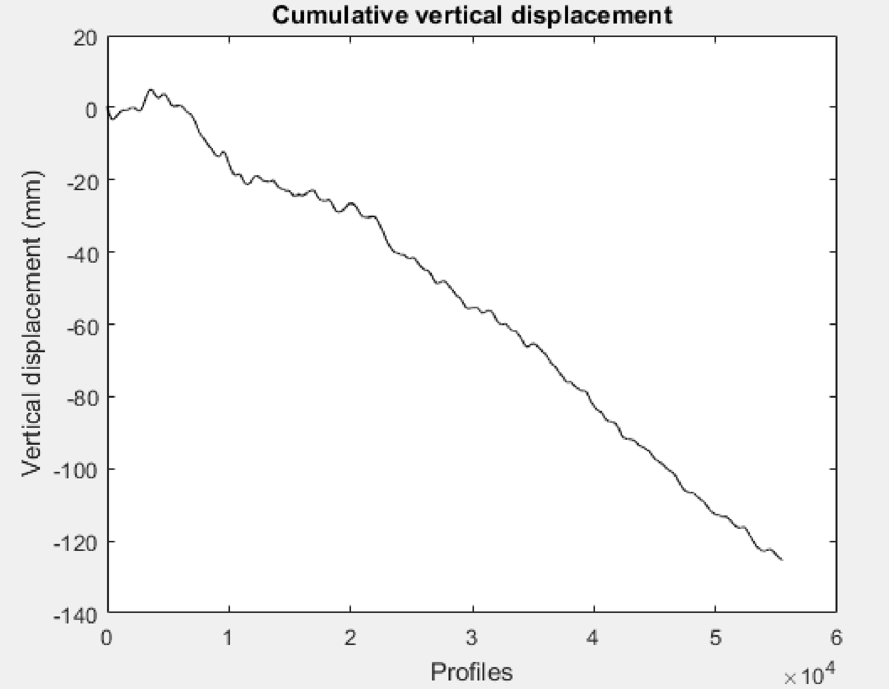
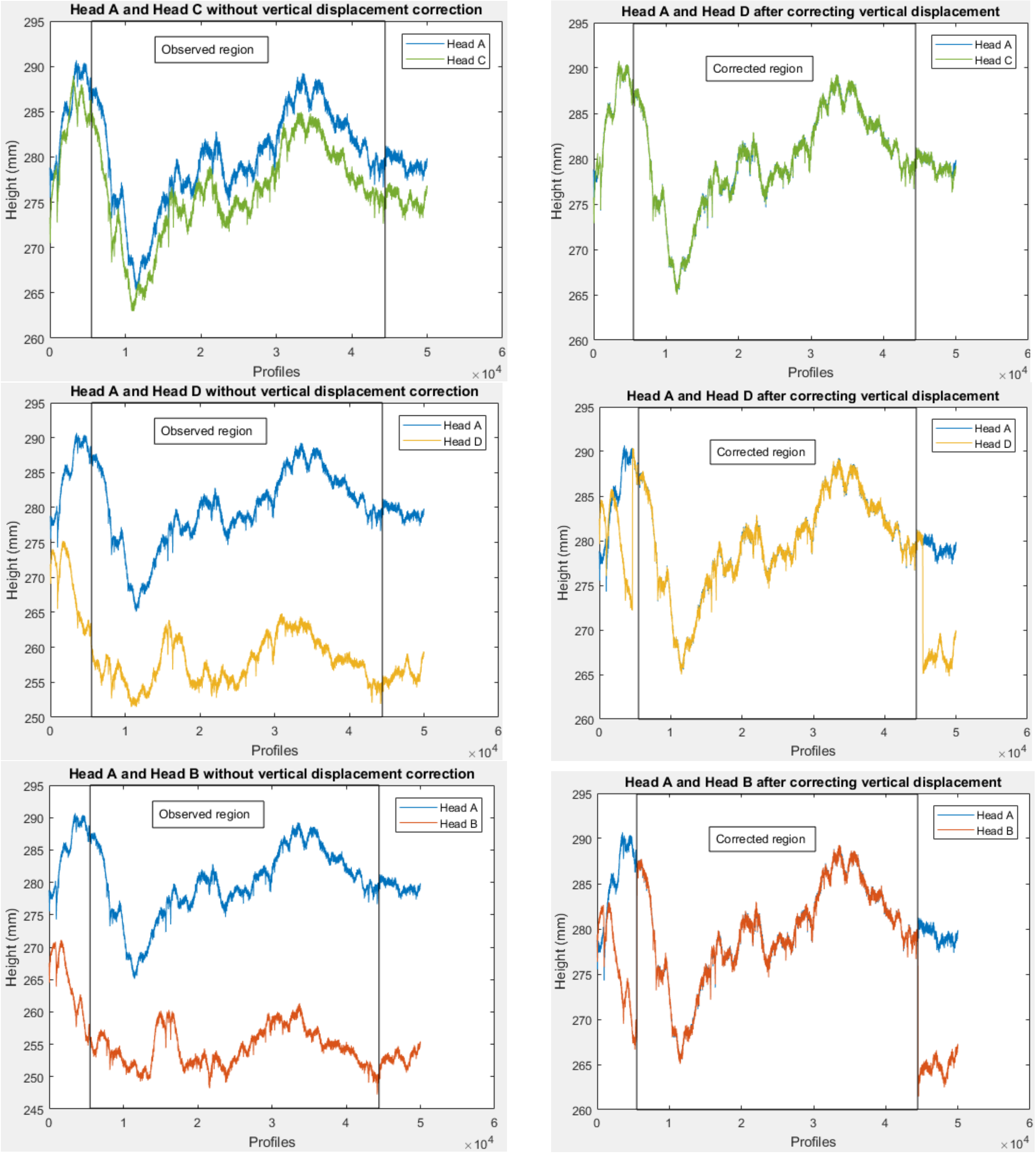


Figure 5-23 Reconstruction of the center of the beam's vertical displacement

The left images of Table 5-2 show that after matching, the features are seen but there is an observable difference in heights that changes with time. This height difference is the result of the vertical movement of the beam which after incorporating the correction described above into the height measured by each laser-sensor results in the corrected images on the right. The observed region is the surface area that all laser-sensors observed at the different times, where the height data that does not overlap for the first and last laser-sensors are neglected.

Table 5-2 Optimization corrections for Head C, Head D, and Head B: uncorrected on the left and corrected on the right



Head A and Head C without vertical displacement correction

Head A and Head D after correcting vertical displacement

Head A and Head D without vertical displacement correction

Head A and Head D after correcting vertical displacement

Head A and Head B without vertical displacement correction

Head A and Head B after correcting vertical displacement

Each laser-sensor was independently corrected with the results from the estimated vertical displacement. The results were then subtracted from the laser-sensors that were located in the deflection basin (Head D and B) to the laser-sensors in the undeflected region (Head A and C). The difference results in the pavement deflection with time (Figure 5-24). The frequency that is seen in the figure is a re-occurring noise that is introduced into the system. The source of this frequency has not been found yet due to not having a project-dedicated vehicle. The frequency changes with different rental trucks. The range that has been observed from November 2018 to August 2019 ranges from 2.2 to 3.6 Hz. The root mean square (RMS) of the amplitude shown below are: 0.0464 mm, 0.0781 mm, and 0.0450 mm for Heads A-C, A-D and A-B, respectively. These signifies that any deflection above this noise can be captured by the DMD. The results are without placing the DMD in the deflection basin. Future test will observe how these three frequencies affect the results and the deflections.

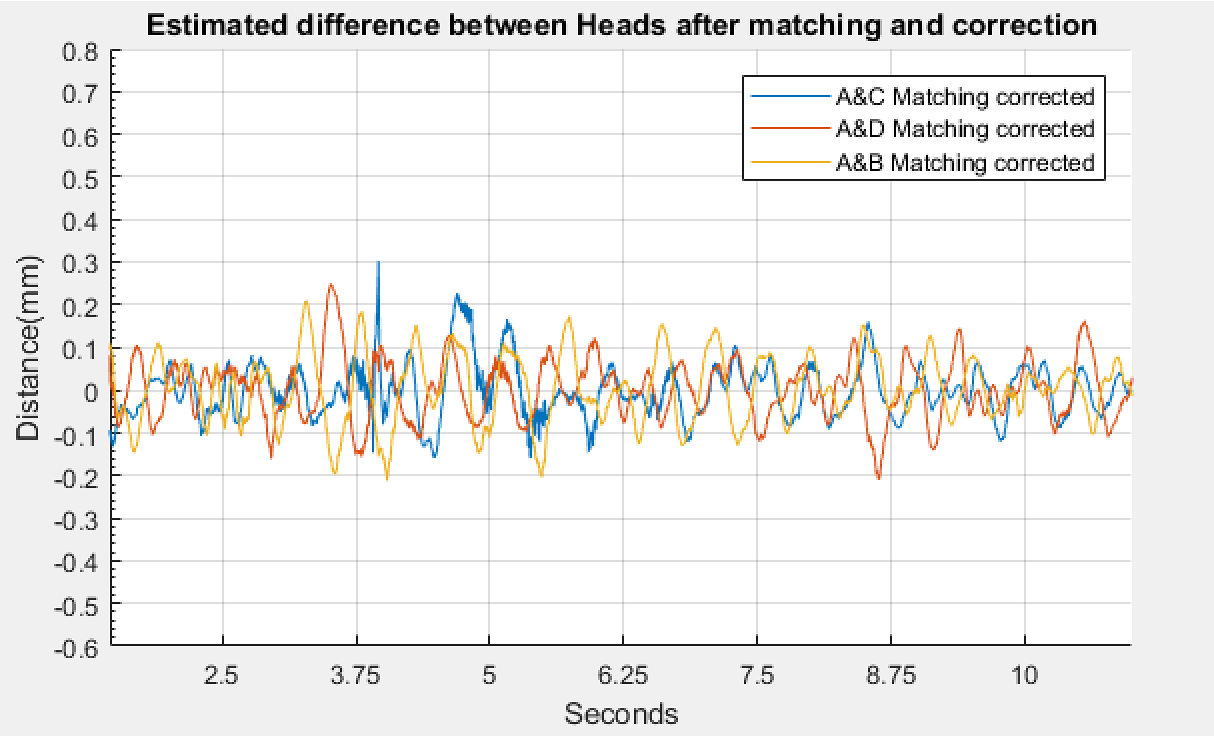


Figure 5-24 Laser-sensor height differences to A; deflection measurement

Chapter 6. Preliminary Conclusions and Recommendations

6.1. Conclusions and Recommendations

The main technical objective of this research project was the development of a system that allows the continuous measurement of surface pavement deflections under a moving axle load. During the first phase of the project, a series of sensors were evaluated and, as a result, it was concluded that scanning laser technology offered the best option to achieve the technical objectives of the project. This report describes the work performed under Phase 2 of the project, which involved the implementation of the identified technologies into a prototype.

The system components, the DAQ algorithms and code, and the data processing techniques have been described in Chapters 2 to 5 of this research report. As the various technologies were described, some unresolved issues were also highlighted. Those issues are currently being addressed and the work could be continued in the future if Phase 3 of the project were to be activated.

To summarize, the deflection measuring system developed during this project consists of a minimum of four laser-sensors and one RLG that are attached to a rigid beam. To date, an aluminum beam has been used to demonstrate the feasibility of the instruments selected. In the future, a lighter and more rigid beam should be used, made of a material that offers a better height-to-weight ratio. It is contemplated that this beam will be custom made. For the construction of the beam, the research team requires a project-dedicated vehicle so the beam can be developed to fit the mechanical characteristics of the vehicle and attenuate systematic vibrations. The beam fabrication should consider the unique mechanical characteristics of a dedicated vehicle. Adapting the measuring beam to different vehicles for testing during this project proved to be ineffective and inefficient. Many of the problems and delays encountered during the past 18 months of the project could have been avoided had a dedicated vehicle been available.

In terms of data acquisition system, to minimize costs, all hardware components are off-the-shelf components so they are easy to purchase. The data acquisition algorithms were developed in-house and will be compiled for the final user (Appendix B). Furthermore a User Manual and some additional examples are given in Appendix C and Appendix D, respectively. One issue to be resolved in the future relates to the triggering system that allows activating all instruments at once while allowing the collection of individual instruments' data at compatible rates. This particular issue is currently being address by the research team.

Finally, the data processing system, which is the most challenging and novel aspect of this project, has been developed and preliminarily tested using the prototype measuring beam. As it was demonstrated and shown in Chapter 5, the performance of the matching algorithm is satisfactory to recover the pavement deflection above the targeted accuracy of 100 microns.

6.2. The Work Ahead

During Phase 2 of the project, the work focused on the development of the prototype measuring system, the integration of the measuring system onto a testing vehicle, and the testing and calibration of the integrated system in the field. However, the integration of the measuring system with the testing vehicle was not entirely satisfactory because the difficulties associated with the lack of a project-dedicated vehicle. The system had to be modified every month to accommodate the unique characteristics of the particular rental vehicle. Several modifications were implemented that partially addressed this problem but it is not an optimal situation. The prototype measuring system was built in isolation on a provisional beam and was tested on several rental vehicles. During the last month of the project (August 2019), the system will be evaluated again before delivery at the Pickle Research Campus under a range of axle loads, operating speeds, and pavement temperatures.

References

[1] AASHTO T256-01-UL (2001), *Standard Method of Test for Pavement Deflection Measurements,* American Association of State Highway and Transportation Officials, Washington, D.C., 2001.www.transportation.org/home/organization

[2] ASTM D4695-03 (2015), *Standard Guide for General Pavement Deflection Measurements*, ASTM International, West Conshohocken, Pennsylvania, 2015.

[3] E. Levenberg, *Analysis of Pavement Response to Subsurface Deformations*, Computers and Geotechnics, Volume 50, 2013, Pages 79-88, ISSN 0266-352X, http://dx.doi.org/10.1016/j.compgeo.2012.12.011.

[4] J. A. Bay, and K. H. Stokoe (1998), *Development of a rolling dynamic deflectometer for continuous deflection testing of pavements,* Geotechnical engineering report GR98-7, Center for Transportation Research, Bureau of Engineering Research, the University of Texas at Austin, Austin, Texas, 1998.

[5] J. L. Lee, K. H. Stokoe, and J. A. Bay (2005), *The rolling dynamic deflectometer: A tool for continuous deflection profiling of pavements,* In *Proceedings of the 16th International Conference on Soil Mechanics and Geotechnical Engineering: Geotechnology in Harmony with the Global Environment* (Vol. 3, pp. 1745-1748), 2005.

[6] K. H. Stokoe, J.-S. Lee, M. Lewis, R. Hayes, T. Scullion and W. Liu (2013), CTR technical report no. 0-6005-3, Center for Transportation Research, Bureau of Engineering Research, the University of Texas at Austin, Austin, Texas, 2013.

[7] ANSYS official website: http://www.ansys.com/

[8] J. A. Prozzi, C. Claudel, P. Pasupathy, A. Verma, and J. B. Hernandez (2017), *Development of Laser Based Sensor to Measure True Road Surface Deflection*, CTR technical report 0-6873-1, University of Texas at Austin, Center for Transportation Research, Texas, 2017.

[9] SolidWorks official website, http://www.solidworks.com/

[10] Ultimaker 3D Printers official website https://ultimaker.com/

[11] Material Properties of Sorbothane, http://www.sorbothane.com/technical-data-sheet.aspx

[12] Q. H. Shah and A. Topa (2014), *Modeling Large Deformation and Failure of Expanded Polystyrene Crushable Foam Using LS-DYNA*, Modelling and Simulation in Engineering, volume 2014, Article ID 292647, pp. 1-7.

[13] G.A. Athanasopoulos, P.C. Pelekis, and V.C. Xenaki (1999), *Dynamic Properties of EPS Geofoam: An Experimental Investigation,* Volume 6 Issue 3, January 1999, pp. 171-194.

[14] HAENNI official website: https://www.haenni-scales.com/

[15] Honeywell official website: https://aerospace.honeywell.com/en/products/navigation-and-sensors/gg1320an-digital-ring-laser-gyroscope

[16] Arduino Mega board website, https://store.arduino.cc/usa/arduino-mega-2560-rev3

[17] Keyence official website: http://www.keyence.com/

[18] PCB Accelerometer website: http://www.pcb.com/Products/model/356B18

[19] NI DAQ website: http://www.ni.com/en-us/support/model.ni-9234.html

[20] PCB Conditioner website: http://www.pcb.com/Products/model/480B21

[21] LabVIEW official website: http://www.ni.com/en-us/shop/labview.html

[22] Matlab official website: https://www.mathworks.com/products/matlab.html

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool (2008), *Speeded-Up Robust Features (SURF)* Computer Vision and Image Understanding, Volume 110, Issue 3, Pages 346-359, June 2009

[24] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, September 2013.

[25] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008. http://stanford.edu/~boyd/graph_dcp.html.

Appendix A. Sensor Complications and Findings

Laser-Sensor Interaction

When designing the beam and the placement of the laser-sensors, there was a particular trade-off between the researcher's requirements and the laser-sensor. The trade-off was the spacing between laser-sensors that was needed for estimations performed by the algorithm. The shorter the space between two laser-sensors, the better the estimation.

To measure the initial angle of the DMD system, two laser-sensors were to be positioned as close apart as optically possible on the beam. The two laser-sensors were opposing from the side from which the laser emits light, also termed as head-to-head. To understand how it worked, the triangulation principle of the laser-sensor for collecting data is explained next.

The laser emits a line light on the surface and the camera from the same laser-sensor captures an image of the surface area. If the two laser-sensors are placed too close, the camera will pick up signal from the opposing laser-sensor, causing interference.

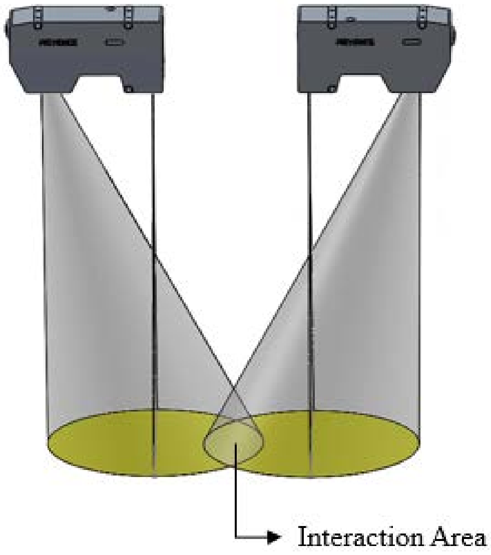


Figure A-1 Laser-sensors in the head-to-head configuration

In the data, this interference is seen as spikes or abnormal height values. The spacing is also dependent on the height of the laser-sensor with respect to the surface; the further away the laser-sensor is from the surface, the larger the area captured by the camera. The opposite occurs if the laser-sensor is placed closer to the surface, the camera area is small, allowing for the spacing of the two laser-sensors to be closer apart. Finally, the pavement surface, or pavement profile, contributed to the problem. If the aggregates on the pavement were very coarse, the reflection of the laser light would reflect to the camera of the other laser-sensor. Also, if the pavement has a large feature such as a speed bump, this can cause the two lasers to interfere because of the fast change in surface height. While it is important to keep the two lasers as close as possible, it was also important to take into account these factors when setting the distance between the laser-sensors.

Laser-Sensor Mounting Plate

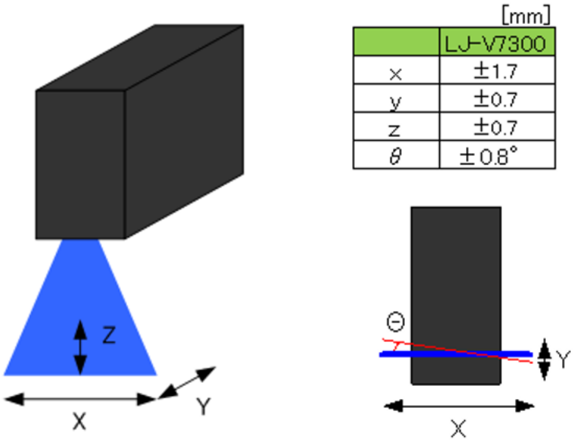
The first prototype of the DMD, Section 2.3, included aluminum plates that attached between the laser-sensor and the beam. These plates allowed for adjusting the pitch angle and the height of each laser. It was later found that these plates were prone to shifting from sudden vertical vibrations. The aluminum plates were removed and the laser-sensors were fixed directly to the beam, Section 2.4.

Laser-Sensor Mounted Directly to Beam

The beam was drilled to fix in place the laser-sensors. The research group performed their own machining on the beam. The hole positioning tolerance achieved based on the cumulative variations of several manufacturing variables (machine rigidity, drill fixture, manual location technique) resulted in 1.0 mm tolerance. An error of no greater than 1.0 mm signified a pitch angle in the laser-sensors head when placed on the beam that was to introduce an unknown angle whenever the laser-sensor was fixed on the beam. The pitch angle signified a greater or smaller spacing in the head-to-head configuration of the two lasers which caused interference. The pitch angle also meant the laser-sensor was scanning slightly different texture due to the angle when compared to other laser-sensors on the beam that have smaller or no pitch angle. As a solution, several measurements were recorded when the sensors were fixed to the beam

Laser-Sensor Encasing

The exact position of the emitted laser line is challenging to physically measure and cannot be assumed to be aligned with the unit's encasing. Therefore, all the laser-sensors are considered unique as they each have different positioning of the laser diode inside the encasing. The manufacturer has tolerances on four parameters: X, Y, Z, and theta.



[mm]

	LJ-V7300
x	±1.7
y	±0.7
z	±0.7
θ	±0.8°

Figure A-2 Laser-sensor unit and the maximum deviation from manufacturer

Laser-Sensor Resonance Frequency

Several indoor tests were performed to evaluate the sensors' internal frequency. For the laser-sensors a shaker table was used in conjunction with a waveform generator to control the vibration. An external frequency was generated ranging from 0.1 Hz to 15.0 Hz, in increments of 0.1 Hz. When the laser data was processed, if any frequency within the range showed a higher amplitude, it would have signified that an external frequency would have an impact on the heights measured. The laser-sensor was evaluated on its three different axis. The results showed no external frequency within 0.1 Hz to 15.0 Hz that affected the height measurement. The sensor's manual also states it has a vibration resistance of 10 to 57 Hz.

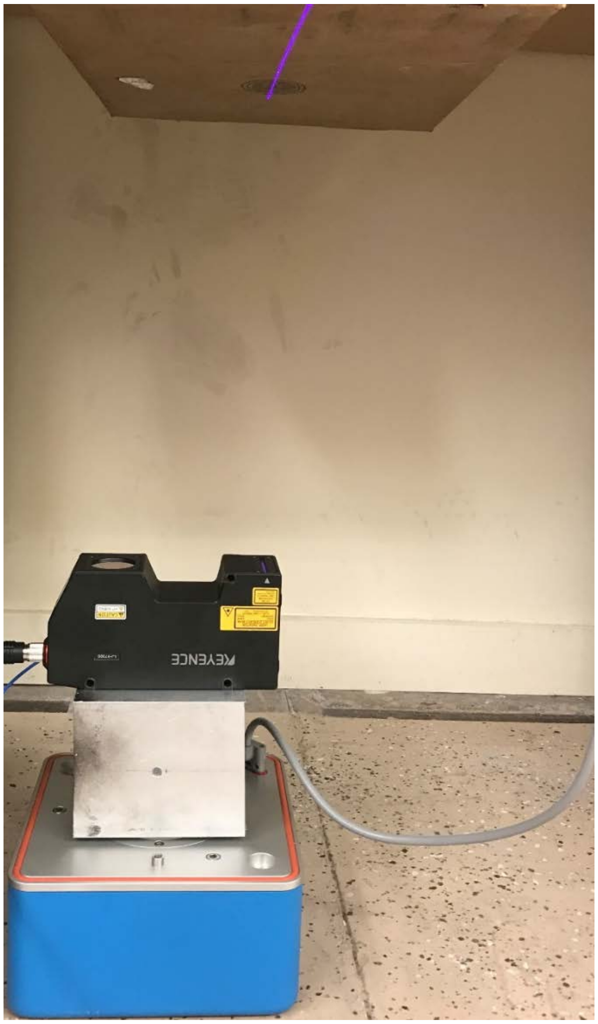


Figure A-3 Laser-sensor fixed to shaker table

Laser-Sensor and Gyroscope Synchronization

To assure proper performance of the sensors, an indoor test was conducted. The test consisted of having the DMD stationary in the laboratory while a researcher applied a slight push downwards in the front of the beam. The force applied was enough that all sensors observed this change in height/angle. The test ran for 60 seconds and the researcher applied the push downwards every 10 seconds. The purpose of this experiment was to use the laser-sensor heights from Head A and Head B to calculate the angle that the gyroscope should measure. The result should be a peak every 10 seconds without a phase shift. The occurrence would mean signs of problems in data transfer, and/or hardware. The results showed the 6 inflicted changes in the angle caused by the researcher within the 60 seconds. More importantly, there was not a noticeable phase shift from comparing the estimated angle from the laser-sensor to that of the gyroscope. The angle from the laser-sensors is seen as drifting from the 0 baseline and it is due to considering the first peak as the reference point. Slight movements in the beam while testing caused the drift observed. Using 4.0 kHz for sampling resulted in no observable data transfer/synchronization problem in 60 seconds. Anything above 4.0 kHz sampling rate has shown the current computer would be the limiting factor.

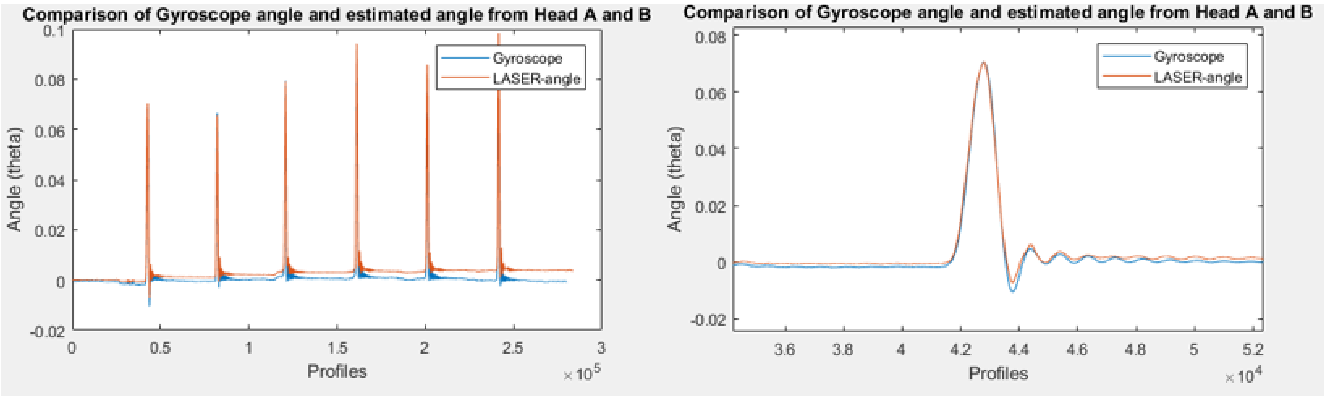


Figure A-4 Gyroscope and laser-sensor comparison: (Left) all the peaks from when beam was pushed downwards; (Right) first peak close-up showing the superposition of the angles

Appendix B. Deflection Estimation Code

Importing Data to MATLAB and trimming: Laser_Loading_Filtering.m

```
Pavement deflection measurement System
Data processing
  Author : Shahrazed Elmetennani
  Email  : shahrazed.elmetennani@gmail.com
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

#############################################################################

% Copyright (c) 2018. Shahrazed Elmetennani All rights reserved.
% Redistribution with or without modification is not allowed without consent
of the PIs

%% ###################LASERS DATA LOADING AND FILTERING #####################

%% Including subfunctions paths
addpath('SubFunctions\');
addpath('tdmsSubfunctions\');

%% Setting filtering windows size==========================================
Section_size_DeadPixelsFiltering = 10;
Section_size_MedianFiltering = 7;
%% ========================================================================

%% Converting TDMS file to Matlab structure
filename = 'LASERS.tdms';
LASERS=TDMS_getStruct(filename);

%% Loading data from structure to Matrices
% Head A
Raw_A_struct=LASERS.Head_A;
RawA1=Raw_A_struct.Untitled_1.data';
[nA,mA]=size(RawA1);
RawA=zeros(nA,800);
RawA(:,1)=RawA1;


parfor i=2:800
   name= 'Untitled_'+string(i);
   name=char(name);
   Fullname=Raw_A_struct;
   files = Fullname.(name);
   RawA(:,i)=[files.data'];
end

% Head B
```

```
Raw_B_struct=LASERS.Head_B;
RawB1=Raw_B_struct.Untitled_1.data';
[nB,mB]=size(RawB1);
RawB=zeros(nB,800);
RawB(:,1)=RawB1;


parfor i=2:800
   name= 'Untitled_'+string(i);
   name=char(name);
   Fullname=Raw_B_struct;
   files = Fullname.(name);
   RawB(:,i)=[files.data'];
end

% Head C
Raw_C_struct=LASERS.Head_C;
RawC1=Raw_C_struct.Untitled_1.data';
[nC,mC]=size(RawC1);
RawC=zeros(nC,800);
RawC(:,1)=RawC1;

parfor i=2:800
   name= 'Untitled_'+string(i);
   name=char(name);
   Fullname=Raw_C_struct;
   files = Fullname.(name);
   RawC(:,i)=[files.data'];
end

% Head D
Raw_D_struct=LASERS.Head_D;
RawD1=Raw_D_struct.Untitled_1.data';
[nD,mD]=size(RawD1);
RawD=zeros(nD,800);
RawD(:,1)=RawD1;

parfor i=2:800
   name= 'Untitled_'+string(i);
   name=char(name);
   Fullname=Raw_D_struct;
   files = Fullname.(name);
   RawD(:,i)=[files.data'];
end

%% Resizing and Rescaling
Ds=200;
De=600;
RawA1=RawA(:,Ds:De)+300;
RawB1=RawB(:,Ds:De)+300;
RawC1=RawC(:,Ds:De)+300;
RawD1=RawD(:,Ds:De)+300;

%% Filtering
%=========Dead pixels
```

```
ssDPF=Section_size_DeadPixelsFiltering;

[AF]=FilterDeadPixels(RawA1,ssDPF);
[BF]=FilterDeadPixels(RawB1,ssDPF);
[CF]=FilterDeadPixels(RawC1,ssDPF);
[DF]=FilterDeadPixels(RawD1,ssDPF);

%=========Median filtering
ssMF=Section_size_MedianFiltering;
A = medfilt2(AF, [ssMF ssMF]);
B = medfilt2(BF, [ssMF ssMF]);
C = medfilt2(CF, [ssMF ssMF]);
D = medfilt2(DF, [ssMF ssMF]);

%% Fliping
A= fliplr(A);
B= fliplr(B);

%% Plots
% ===============================
MA=mean(A');
MC=mean(C');
MD=mean(D');
MB=mean(B');
figure(2)
hold all
plot(MA);
plot(MC);
plot(MD);
plot(MB);
title('Lasers scans averaged over profiles')
legend('Head A', 'Head C', 'Head D', 'Head B')
xlabel('Time (ms)')
ylabel('Distance (mm)')
% ===============================
figure(3)
hold all
plot(MA-mean(MA));
plot(MC-mean(MC));
plot(MD-mean(MD));
plot(MB-mean(MB));
title('Lasers scans averaged over profiles and normalized')
legend('Head A', 'Head C', 'Head D', 'Head B')
xlabel('Time (ms)')
ylabel('Distance (mm)')
% ===============================
minA=min(min(A(A>0)));
minB=min(min(B(B>0)));
minC=min(min(C(C>0)));

zmin=min([minA,minB,minC]);

maxA=max(max(A(A>0)));
maxB=max(max(B(B>0)));
maxC=max(max(C(C>0)));
```

```
zmax=max([maxA,maxB,maxC]);

figure(1)
colormap(jet)
subplot(1,4,1),imagesc(A);title('Head A');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,2), imagesc(C);title('Head C');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,3), imagesc(D);title('Head D');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,4), imagesc(B);title('Head B');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);
```

Importing gyroscope data: Gyro_Loading_Correcting.m

```
Pavement deflection measurement System
Data processing
  Author : Shahrazed Elmetennani
  Email  : shahrazed.elmetennani@gmail.com
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

#############################################################################

% Copyright (c) 2018. Shahrazed Elmetennani All rights reserved.
% Redistribution with or without modification is not allowed without consent
of the PIs

%% ###################GYRO DATA LOADING AND CORRECTION ######################

%% Including subfunctions paths
addpath('SubFunctions\');
addpath('tdmsSubfunctions\');

%% Variable to tune
G_sign = 1;
G_slope = 3.5;          % slope caused by the earth rotation (To be
identified)
%It is equal to 3.2 for the innovation blvd
%It is equal to 3.5 for the creativity blvd
w = 3;

%% Converting TDMS file to Matlab structure
filename = 'Gyro.tdms';
Gyro=TDMS_getStruct(filename);

%% Loading from structure to a matrix of strings
GyroRaw=Gyro.Untitled.Untitled.data';
[nG,mG]=size(GyroRaw);
sizeG=max(nG,mG);
RawG=[" "];
parfor i=1:sizeG
  val=cell2mat(GyroRaw(i));
  RawG(i)=string(val);
end

%% Acquisition of data as double from the string
RawG(1:30)=[];
nT=length(RawG);
GyroData=[];
t=0;
k=0;
for i=1:nT
    k=k+1;
    Val=RawG(i);
    val=char(Val);
    if(val(w)=='*' && val(end)=='#')
        DD=find(val=='$');
        Gtime=string(val(w+1:DD-1));
        GyroData(k,1)=str2double(Gtime);
        Gangle=string(val(DD+1:end-1));
        GyroData(k,2)=str2double(Gangle);
    end

end
GyroData(find(GyroData(:,1)==0 & GyroData(:,2)==0),:)=[];

%% Wrap up correction
X=GyroData(:,2);
n=length(X);
Y=[];
Y(1) = X(1);
Thr=40000;                             %Ramp variation threshold

if Y(1)>Thr
    Bias=0;                             %Correction bias
else
    Bias=65536;
    Y(1)=Y(1)+Bias;
end

for i=2:n
    Xp=X(i-1);
    % Detection of the discontinuity
    if (Xp-X(i)>Thr)
            Bias=65536;           %Bias update for positive variation
    elseif (Xp-X(i)<-Thr)
           Bias=0;          %Bias update for negative variation
    end
    Y(i) = X(i)+ Bias;              %Corrected data entry
end

%% Conversion from increments to degrees
GyroData(:,3)=Y*1.113065/3600;
GyroData(:,3)=GyroData(:,3)-GyroData(1,3);

%% Earth rotation correction y=+3.52 e-03*t (time in ms) for innovation blvd.
GyroData(:,4)=GyroData(:,3)+G_sign*G_slope*10^(-6)*(GyroData(:,1)-
GyroData(1,1));


figure(4)
plot(GyroData(:,1),GyroData(:,3),'b');
hold on
plot(GyroData(:,1),GyroData(:,4),'r');
title('Gyro angle')
xlabel('Time (ms)')
ylabel('Angle ({^\circ})')
legend('Measured gyro angle','Gyro angle after earth correction')
```

Matching LASER-sensors: MatchingABCD.m

```
Pavement deflection measurement System
Data processing
  Author : Shahrazed Elmetennani
  Email  : shahrazed.elmetennani@gmail.com
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

#############################################################################

% Copyright (c) 2018. Shahrazed Elmetennani All rights reserved.
% Redistribution with or without modification is not allowed without consent
of the PIs

%% ###################MATCHING SECTIONS###############################
close all

%% Including subfunctions paths
addpath('SubFunctions\');
addpath('tdmsSubfunctions\');

%% =========== Parameters tuning ===================
%% *********** Section for matching
nS=16000;       %Start point
nE=74000;      %End point

%% *********** Matching sections size
%(These values are for speed 2-5 mile/h)
Section=900; %250
Subsection=900; %250
%500 section made it works bad
%1000/500 ok
%1000/750 looked better
%1000/1000 ok
%750/750 is bad
%1100/1100 is bad
%1200/1200 even worse

ShiftAB=3000;
ShiftAD=2000;
ShiftAC=1000;


%% *********** Acquisition delays
% with respect to Head B based on the vibration response induced by the
hammer
% (assesed manually so far)
Gyro_Delay=-870;    %Gyroscope acquisition delay
A_Delay=-95;        %Head A acquisition delay
C_Delay=-9;         %Head C acquisition delay
D_Delay=-85;        %Head D acquisition delay
```

```
%% *********** Lasers distances from the center of the beam
% Beam measurements evaluated manually
L = 3050;           % Length of the beam
%"O" being the center of the beam
OA=1200+114;        %Distance between O and A
OC=1200+114-120;    %Distance between O and C
OD=1200-178;        %Distance between O and D
OB=1200+263;        %Distance between O and B

%% *********** Lasers distances from the center of the laser itself
OL=38-12+50;

%% #########################################################
%% ===========  Loading data to match
Asec=A(nS+A_Delay:nE+A_Delay,:);    %Head A with acquisition delay corrected
Csec=C(nS+C_Delay:nE+C_Delay,:);    %Head C with acquisition delay corrected
Dsec=D(nS+D_Delay:nE+D_Delay,:);    %Head D with acquisition delay corrected
Bsec=B(nS:nE,:);                    %Head B with acquisition delay corrected

Gyroexp=GyroRe(nS+Gyro_Delay:nE+Gyro_Delay,:);  % Gyroscope with acquisition
delay corrected
Gyroexp=Gyroexp-Gyroexp(1);                     % initial angle of the
matching section is defined as reference (=zero)

%% ===========  Beam rotation angle correction
[nA,mA] = size(Asec);
[nC,mC] = size(Csec);
[nB,mB] = size(Bsec);
[nD,mD] = size(Dsec);

Asec=Asec+repmat(OA*sin((Gyroexp)/180*pi),1,mA);
Csec=Csec+repmat(OC*sin((Gyroexp)/180*pi),1,mC);
Dsec=Dsec-repmat(OD*sin((Gyroexp)/180*pi),1,mD);
Bsec=Bsec-repmat(OB*sin((Gyroexp)/180*pi),1,mB);

%% =========== Matching sections====================

%% *********** Matching C to A
Shift = ShiftAC;

[nA,mA] = size(Asec);
[nC,mC] = size(Csec);
[nD,mD] = size(Dsec);
[nB,mB] = size(Bsec);

Cout=zeros(nA,mA);
AoutC=zeros(nA,mA);
KAC=zeros(nA,1);
N=floor((nA-Shift-Section)/Subsection)

Frontsec=Asec;  %Front laser is head A
Backsec=Csec;   %Back laser is head C
```

```
Options.upright=false;
Options.tresh=0.0001;

for i=1:1:round(N)
    i
    [nA,mA] = size(Asec);

    %% -----section to match to load from the data
    Nstart=(i-1)*Subsection+1;  % start point
    Nend=Nstart+Section-1;      % end point

    Abis=[];
    Bbis=[];
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:)-
(min(min(Backsec(Nstart:Nend+Shift,:))))+2;
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:)-
(min(min(Frontsec(Nstart:Nend,:))))+2;

    %% -----Rescaling to save as an image
    maxA=max(Abis(:));
    maxB=max(Bbis(:));
    Aratio=64/maxA;
    Bratio=64/maxB;
    [nA,mA] =size(Abis);
    Amean=(Abis)*Aratio;
    [nB,mB] =size(Bbis);
    Bmean=(Bbis)*Bratio;

    %% -----Save the matrices as an PNG images
    imwrite(Amean,jet,['SecAC_A.png'],'BitDepth',8);
    imwrite(Bmean,jet,['SecAC_C.png'],'BitDepth',8);

    %% -----Loading of the images for processing
    [I1]=imread(['SecAC_A.png']);
    [I2]=imread(['SecAC_C.png']);

    %% -----Features color enhancement
    kernelw=24;
    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    %% -----Matching features
    % Get the Key Points
    Ipts1=OpenSurf(I1,Options);
    Ipts2=OpenSurf(I2,Options);
    % Put the landmark descriptors in a matrix
    D1 = reshape([Ipts1.descriptor],64,[]);
    D2 = reshape([Ipts2.descriptor],64,[]);

    % Find the best matches
    err=zeros(1,length(Ipts1));
    cor1=1:length(Ipts1);
    cor2=zeros(1,length(Ipts1));
    for k=1:length(Ipts1)
        distance=sum((D2-repmat(D1(:,k),[1 length(Ipts2)])).^2,1);
        [err(k),cor2(k)]=min(distance);
    end

    % Sort matches on vector distance
    [err, ind]=sort(err);
    cor1=cor1(ind);
    cor2=cor2(ind);
    minA=min(min(Abis(Abis>0)));
    minB=min(min(Bbis(Bbis>0)));
    zmin=min(minA,minB);
    maxA=max(max(Abis(Abis>0)));
    maxB=max(max(Bbis(Bbis>0)));
    zmax=max(maxA,maxB);

    %% -----Make vectors with the coordinates of the best matches
    Pos1=[[Ipts1(cor1).y]',[Ipts1(cor1).x]'];
    Pos2=[[Ipts2(cor2).y]',[Ipts2(cor2).x]'];
    Pos1=Pos1(1:min(length(cor1),15),:);
    Pos2=Pos2(1:min(length(cor1),15),:);
    POS=[Pos1(:,1)-Pos2(:,1)];
    [TF,lower,upper,center] = isoutlier(POS,'quartiles');
    k=0;
    Pos1(find(TF~=0),:)=[];
    Pos2(find(TF~=0),:)=[];

    %% -----Compute the delay time
    Shifty=[Pos1(:,1)-Pos2(:,1)];
    YShift=[round(mean(Shifty))];

    %% -----Geomtric correction of the images
    tform = fitgeotrans(fliplr(Pos2),fliplr(Pos1),'similarity'); %
fiteotrans(movingpoints,fixedpoints, transofmration type)
    outputView = imref2d(size(Bbis));
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:);
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:);
    recovered =
imwarp(Bbis,tform,'SmoothEdges',false,'OutputView',outputView);

    Cout((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=recovered(round(Section
/2)-round(Subsection/2)+1:(round(Section/2)+round(Subsection/2)),:);
    AoutC((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=Abis(round(Section/2)-
round(Subsection/2)+1:round(Section/2)+round(Subsection/2),:);
```

```
    KAC((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2))=YShift*ones(length((i-
1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2)),1);
end

%% *********** Matching D to A
Shift = ShiftAD;

[nA,mA] = size(Asec);
[nC,mC] = size(Csec);
[nD,mD] = size(Dsec);
[nB,mB] = size(Bsec);

Dout=zeros(nA,mA);
AoutD=zeros(nA,mA);
KAD=zeros(nA,1);
N=floor((nA-Shift-Section)/Subsection)

Frontsec=Asec;  %Front laser is head A
Backsec=Dsec;   %Back laser is head D

Options.upright=false;
Options.tresh=0.0001;
for i=1:1:round(N)
    i
    [nA,mA] = size(Asec);

    %% -----section to match to load from the data
    Nstart=(i-1)*Subsection+1;  % start point
    Nend=Nstart+Section-1;      % end point

    Abis=[];
    Bbis=[];
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:)-
(min(min(Backsec(Nstart:Nend+Shift,:))))+2;
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:)-
(min(min(Frontsec(Nstart:Nend,:))))+2;

    %% -----Rescaling to save as an image
    maxA=max(Abis(:));
    maxB=max(Bbis(:));
    Aratio=64/maxA;
    Bratio=64/maxB;
    [nA,mA] =size(Abis);
    Amean=(Abis)*Aratio;
    [nB,mB] =size(Bbis);
    Bmean=(Bbis)*Bratio;

    %% -----Save the matrices as an PNG images
    imwrite(Amean,jet,['SecAD_A.png'],'BitDepth',8);
    imwrite(Bmean,jet,['SecAD_D.png'],'BitDepth',8);
    %% -----Loading of the images for processing
    [I1]=imread(['SecAD_A.png']);
    [I2]=imread(['SecAD_D.png']);

    %% -----Features color enhancement
    kernelw=24;
    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    %% -----Matching features
    % Get the Key Points
    Ipts1=OpenSurf(I1,Options);
    Ipts2=OpenSurf(I2,Options);

    % Put the landmark descriptors in a matrix
    D1 = reshape([Ipts1.descriptor],64,[]);
    D2 = reshape([Ipts2.descriptor],64,[]);

    % Find the best matches
    err=zeros(1,length(Ipts1));
    cor1=1:length(Ipts1);
    cor2=zeros(1,length(Ipts1));
    for k=1:length(Ipts1)
        distance=sum((D2-repmat(D1(:,k),[1 length(Ipts2)])).^2,1);
        [err(k),cor2(k)]=min(distance);
    end

    % Sort matches on vector distance
    [err, ind]=sort(err);
    cor1=cor1(ind);
    cor2=cor2(ind);
    minA=min(min(Abis(Abis>0)));
    minB=min(min(Bbis(Bbis>0)));
    zmin=min(minA,minB);
    maxA=max(max(Abis(Abis>0)));
    maxB=max(max(Bbis(Bbis>0)));
    zmax=max(maxA,maxB);

    %% -----Make vectors with the coordinates of the best matches
    Pos1=[[Ipts1(cor1).y]',[Ipts1(cor1).x]'];
    Pos2=[[Ipts2(cor2).y]',[Ipts2(cor2).x]'];
    Pos1=Pos1(1:min(length(cor1),15),:);
    Pos2=Pos2(1:min(length(cor1),15),:);
    POS=[Pos1(:,1)-Pos2(:,1)];
    [TF,lower,upper,center] = isoutlier(POS,'quartiles');
    k=0;
    Pos1(find(TF~=0),:)=[];
    Pos2(find(TF~=0),:)=[];
    %% -----Compute the delay time
    Shifty=[Pos1(:,1)-Pos2(:,1)];
    YShift=[round(mean(Shifty))];

    %% -----Geomtric correction of the images
    tform = fitgeotrans(fliplr(Pos2),fliplr(Pos1),'similarity');
    outputView = imref2d(size(Bbis));
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:);
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:);
    recovered =
imwarp(Bbis,tform,'SmoothEdges',false,'OutputView',outputView);

    Dout((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=recovered(round(Section
/2)-round(Subsection/2)+1:(round(Section/2)+round(Subsection/2)),:);
    AoutD((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=Abis(round(Section/2)-
round(Subsection/2)+1:round(Section/2)+round(Subsection/2),:);
    KAD((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2))=YShift*ones(length((i-
1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2)),1);
end

%% *********** Matching B to A
Shift = ShiftAB;

[nA,mA] = size(Asec);
[nC,mC] = size(Csec);
[nD,mD] = size(Dsec);
[nB,mB] = size(Bsec);

Bout=zeros(nA,mA);
AoutB=zeros(nA,mA);
KAB=zeros(nA,1);
N=floor((nA-Shift-Section)/Subsection)

Frontsec=Asec;  %Front laser is head A
Backsec=Bsec;   %Back laser is head B

Options.upright=false;
Options.tresh=0.0001;

for i=1:1:round(N)
    i
    [nA,mA] = size(Asec);

    %% -----section to match to load from the data
    Nstart=(i-1)*Subsection+1;  % start point
    Nend=Nstart+Section-1;      % end point

    Abis=[];
    Bbis=[];
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:)-
(min(min(Backsec(Nstart:Nend+Shift,:))))+2;
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:)-
(min(min(Frontsec(Nstart:Nend,:))))+2;

    %% -----Rescaling to save as an image
    maxA=max(Abis(:));
    maxB=max(Bbis(:));
    Aratio=64/maxA;
    Bratio=64/maxB;
    [nA,mA] =size(Abis);
    Amean=(Abis)*Aratio;
    [nB,mB] =size(Bbis);
    Bmean=(Bbis)*Bratio;

    %% -----Save the matrices as an PNG images
    imwrite(Amean,jet,['SecAB_A.png'],'BitDepth',8);
    imwrite(Bmean,jet,['SecAB_B.png'],'BitDepth',8);

    %% -----Loading of the images for processing
    [I1]=imread(['SecAB_A.png']);
    [I2]=imread(['SecAB_B.png']);

    %% -----Features color enhancement
    kernelw=24;
    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    I1 = adapthisteq(I1,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');
    I2 = adapthisteq(I2,'NumTiles',[kernelw
kernelw],'ClipLimit',0.02,'Distribution','rayleigh');

    %% -----Matching features
    % Get the Key Points
    Ipts1=OpenSurf(I1,Options);
    Ipts2=OpenSurf(I2,Options);

    % Put the landmark descriptors in a matrix
    D1 = reshape([Ipts1.descriptor],64,[]);
    D2 = reshape([Ipts2.descriptor],64,[]);

    % Find the best matches
    err=zeros(1,length(Ipts1));
    cor1=1:length(Ipts1);
    cor2=zeros(1,length(Ipts1));
    for k=1:length(Ipts1)
        distance=sum((D2-repmat(D1(:,k),[1 length(Ipts2)])).^2,1);
        [err(k),cor2(k)]=min(distance);
    end
    % Sort matches on vector distance
    [err, ind]=sort(err);
    cor1=cor1(ind);
    cor2=cor2(ind);
    minA=min(min(Abis(Abis>0)));
    minB=min(min(Bbis(Bbis>0)));
    zmin=min(minA,minB);
    maxA=max(max(Abis(Abis>0)));
    maxB=max(max(Bbis(Bbis>0)));
    zmax=max(maxA,maxB);

    %% -----Make vectors with the coordinates of the best matches
    Pos1=[[Ipts1(cor1).y]',[Ipts1(cor1).x]'];
    Pos2=[[Ipts2(cor2).y]',[Ipts2(cor2).x]'];
    Pos1=Pos1(1:min(length(cor1),15),:);
    Pos2=Pos2(1:min(length(cor1),15),:);
    POS=[Pos1(:,1)-Pos2(:,1)];
    [TF,lower,upper,center] = isoutlier(POS,'quartiles');
    k=0;
    Pos1(find(TF~=0),:)=[];
    Pos2(find(TF~=0),:)=[];

    %% -----Compute the delay time
    Shifty=[Pos1(:,1)-Pos2(:,1)];
    YShift=[round(mean(Shifty))];

    %% -----Geomtric correction of the images
    tform = fitgeotrans(fliplr(Pos2),fliplr(Pos1),'similarity');
    outputView = imref2d(size(Bbis));
    Abis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=zeros(Nend+Shift-Nstart+1,mB);
    Bbis=Backsec(Nstart:Nend+Shift,:);
    Abis(1:Nend-Nstart+1,:)=Frontsec(Nstart:Nend,:);
    recovered =
imwarp(Bbis,tform,'SmoothEdges',false,'OutputView',outputView);

    Bout((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=recovered(round(Section
/2)-round(Subsection/2)+1:(round(Section/2)+round(Subsection/2)),:);
    AoutB((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2),:)=Abis(round(Section/2)-
round(Subsection/2)+1:round(Section/2)+round(Subsection/2),:);
    KAB((i-1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2))=YShift*ones(length((i-
1)*Subsection+1+round(Section/2)-round(Subsection/2):(i-
1)*Subsection+round(Section/2)+round(Subsection/2)),1);
end

%% ########### PLOTS after matching ##########################
minA=min(min(Asec(Asec>0)));
minB=min(min(Bsec(Bsec>0)));
minC=min(min(Csec(Csec>0)));
minD=min(min(Dsec(Dsec>0)));
```

```
zmin=min([minA,minB,minC,minD]);

maxA=max(max(Asec(Asec>0)));
maxB=max(max(Bsec(Bsec>0)));
maxC=max(max(Csec(Csec>0)));
maxD=max(max(Dsec(Dsec>0)));

zmax=max([maxA,maxB,maxC,minD]);

figure(1)
colormap(jet)
subplot(1,4,1),imagesc(Asec);title('Asec');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,2), imagesc(Cout);title('Cout');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,3), imagesc(Dout);title('Dout');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);

subplot(1,4,4), imagesc(Bout);title('Bout');colorbar;caxis([zmin zmax]);
ylabel('Time (ms)')
xlabel('Points (Px)')
h = colorbar;
hYLabel = ylabel(h, 'Distance (mm)');
set(hYLabel,'Rotation',90);
set(hYLabel, 'Units', 'Normalized', 'Position', [0, 0.5, 0]);
```

Distance Estimation: Distance_estimation_ALL.m

```
Pavement deflection measurement System
Data processing
  Author : Shahrazed Elmetennani
  Email  : shahrazed.elmetennani@gmail.com
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

#############################################################################

% Copyright (c) 2018. Shahrazed Elmetennani All rights reserved.
% Redistribution with or without modification is not allowed without consent
of the PIs

%% ###################ESTIMATED DISTANCE AFTER MATCHING #####################

%% Including subfunctions paths
addpath('SubFunctions\');
addpath('tdmsSubfunctions\');

%% =======================================================
nS=31000;     % Start good matching
nE=54000;    % End good matching

global w
w=1          % Distance estimation window

%% *********** Lasers angles
% Angle difference between the lasers and the beam plane
% GA = 0.988; % corrected
% GC = 0.080; % corrected
% GD = 0.769; % corrected
% GB = 0.071; % corrected
GA = 0.598;
GC = 0.080;
GD = 0.019;
GB = 0.071;

%% *********** Lasers centres
% Distance difference between the laser edge and the beam top edge
HA=212.78;
HC=210.65;
HD=210.65;
HB=212.01;

%% *********** Head A & C Eliminate dead areas
HeadA_D=AoutC(nS:nE-1,1:end);
HeadC_D=Cout(nS:nE-1,1:end);
[nC,mC]=size(HeadC_D);
KC=KAC(nS:nE,:);
```

```
k=0;
while k<mC
    k=k+1;
    if(min(abs(HeadC_D(:,k)))==0)
        HeadA_D(:,k)=[];
        HeadC_D(:,k)=[];
        [nC,mC]=size(HeadC_D);
        k=k-1;
    end
end

DisAC=Distance_estimation(HeadA_D); % C matched with A
DisC=Distance_estimation(HeadC_D);  % C geometric correction

% Compare with mean of same data
DisACmean=mean(HeadA_D');
DisCmean=mean(HeadC_D');

%% *********** Head A & D
HeadA_D=AoutD(nS:nE-1,1:end);
HeadD_D=Dout(nS:nE-1,1:end);
[nD,mD]=size(HeadD_D);
KD=KAD(nS:nE,:);

k=0;
while k<mD
    k=k+1;
    if(min(abs(HeadD_D(:,k)))==0)
        HeadA_D(:,k)=[];
        HeadD_D(:,k)=[];
        [nD,mD]=size(HeadD_D);
        k=k-1;
    end
end

DisAD=Distance_estimation(HeadA_D);
DisD=Distance_estimation(HeadD_D);


%% *********** Head A & B
HeadA_D=AoutB(nS:nE-1,1:end);
HeadB_D=Bout(nS:nE-1,1:end);
[nB,mB]=size(HeadB_D);
KB=KAB(nS:nE,:);

k=0;
while k<mB
    k=k+1;
    if(min(abs(HeadB_D(:,k)))==0)
        HeadA_D(:,k)=[];
        HeadB_D(:,k)=[];
        [nB,mB]=size(HeadB_D);
        k=k-1;
    end
```

```
end

DisAB=Distance_estimation(HeadA_D);
DisB=Distance_estimation(HeadB_D);


%% *********** Misalignement correction
HA=HA-OL*sin((GA)/180*pi);
HC=HC+OL*sin((GC)/180*pi);
HD=HD+OL*sin((GD)/180*pi);
HB=HB-OL*sin((GB)/180*pi);

HH=min([HA HC HB HD]);
HA=HA-HH;
HD=HD-HH;
HB=HB-HH;
HC=HC-HH;

DisAC=DisAC-HA;
DisAD=DisAD-HA;
DisAB=DisAB-HA;

% DisAC +.5
% DisAD +.8

DisC=DisC-HC;
DisD=DisD-HD;
DisB=DisB-HB;

%% *********** Plots
figure(2)
hold all
title('Estimated distances before vibration correction');
plot(DisAC,'k')
hold on
plot(DisC)
plot(DisD)
plot(DisB)
legend('Head A','Head C','Head D','Head B')
xlabel('Time (ms)')
ylabel('Distance (mm)')
hold off

%% ============== Delay times
figure(3)
title('Delay times')
hold all
plot(abs(KC))
plot(abs(KD))
plot(abs(KB))
xlabel('Time (ms)')
ylabel('Time (ms)')
legend('Head C','Head D','Head B')
hold off
```

Deflection Estimation:
PtoP_AC.m

```
Pavement deflection measurement System
Data processing
  Author : Suyash Vishnoi
  Email  : suyashvishnoi@gmail.com
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

#############################################################################

% Redistribution with or without modification is not allowed without consent
of the PIs

#############################################################################
%% Crop Data from Matched Images
%Set boundaries to crop out dead pixels from matched images based on visual
%examination

CropRowS = 10010; % Starting Profile (y-axis)
CropRowE = 532000; % Ending Profile (y-axis)
CropColS = 160; % Starting Profile (x-axis)
CropColE = 280; % Ending Profile (x-axis)

A1500w1500 = AoutC(CropRowS:CropRowE,CropColS:CropColE);
B1500w1500 = Bout(CropRowS:CropRowE,CropColS:CropColE);
C1500w1500 = Cout(CropRowS:CropRowE,CropColS:CropColE);
D1500w1500 = Dout(CropRowS:CropRowE,CropColS:CropColE);

KAC = KAC(CropRowS:CropRowE,1);
KAB = KAB(CropRowS:CropRowE,1);
KAD = KAD(CropRowS:CropRowE,1);

%% Detrend images
%Use detrend() to bring the mean level of each row to 0 mm

[rows,cols] = size(A1500w1500);
A1500w1500d = zeros(rows,cols);
C1500w1500d = zeros(rows,cols);
B1500w1500d = zeros(rows,cols);
D1500w1500d = zeros(rows,cols);

 dt=1; %function setting to subtract mean value from each entry
 for row=1:rows
     A1500w1500d(row,:) = detrend(A1500w1500(row,:),dt);
     C1500w1500d(row,:) = detrend(C1500w1500(row,:),dt);
     B1500w1500d(row,:) = detrend(B1500w1500(row,:),dt);
     D1500w1500d(row,:) = detrend(D1500w1500(row,:),dt);
 end
```

```
%% Profile-Profile based adjustment for Laser C
% The below code can be used for Lasers B and D by replacing the 'C' in
% variable names with 'B' or 'D' as required

%y-adjustment limits
ymax = +20;
ymin = -20;

%x-adjustment limits
xmax = +10;
xmin = -10;

%Select boundaries for profile-profile adjustment
Ps = 0 - ymin; %offset of ymax/ymin to avoid exceeding index
Pe = rows - ymax;

%initializing arrays to save adjustment and error values
merror4kC = zeros(Pe-Ps,1);
yshiftarrC = zeros(Pe-Ps,1);
xshiftarrC = zeros(Pe-Ps,1);

%initializing images to make adjustments
Aimg = A1500w1500;
Cimg = C1500w1500;

for A = Ps+1 : Pe

    mine = 10000; %initialize with a large value
    yshift = 0; xshift = 0;

    % Finding the best x y adjustment
    for y = ymin : ymax

        for x = xmin : xmax

            if(x >= 0)
                error = abs(A1500w1500d(A,1+x:cols) - C1500w1500d(A+y,1:cols-
x));
            else
                error = abs(A1500w1500d(A,1:cols-abs(x)) -
C1500w1500d(A+y,1+abs(x):cols));
            end

            se = sum(error);
            me = se/(cols-abs(x));

            if(me <= mine)
                mine = me;
                yshift = y;
                xshift = x;
            end

        end
    end

    % Making the x y adjustment
    if(xshift >= 0)
        Aimg(A,1:cols-xshift) = A1500w1500(A,1+xshift:cols);
        Cimg(A,1:cols-xshift) = C1500w1500(A+yshift,1:cols-xshift);
    else
        Aimg(A,1:cols-abs(xshift)) = A1500w1500(A,1:cols-abs(xshift));
        Cimg(A,1:cols-abs(xshift)) = C1500w1500(A+yshift,1+abs(xshift):cols);
    end

    % Saving the best matching x y adjustments and corresponding error for
    % later reference
    merror4kC(A-Ps,1) = mine;
    yshiftarrC(A-Ps,1) = yshift;
    xshiftarrC(A-Ps,1) = xshift;

end
```

Deflection Estimation: CVX_Deflection.m

```
Pavement deflection measurement System
Data processing
  Author : Joaquin Hernandez
  Email  : joaquinh@utexas.edu
  PIs:
       Jorge Prozzi (prozzi@mail.utexas.edu)
       Christian Claudel (christian.claudel@utexas.edu)

##############################################################################

% Redistribution with or without modification is not allowed without consent
of the PIs

%% ################### DEFLECTION ESTIMATION ###############################

%% Including subfunctions paths
addpath('SubFunctions\');
addpath('tdmsSubfunctions\');



%% Load Profile-to-Profile Matched images

A1500w1500 = Aimg;
B1500w1500 = Bimg;
C1500w1500 = Cimg;
D1500w1500 = Dimg;

%% Select section to optimize

nS = 20000; %Starting point
nE = nS + 15000; %Ending point

A1500w1500=A1500w1500(nS:nE,:);
B1500w1500=B1500w1500(nS:nE,:);
C1500w1500=C1500w1500(nS:nE,:);
D1500w1500=D1500w1500(nS:nE,:);


********* Run Distance Estimation at this step *********


%% Use variables from MatchingABCD.m

% KC, KD, KB are the values of how much the matching window shifted

timeshiftAC = abs(KC);
timeshiftAD = abs(KD);
```

```
timeshiftAB = abs(KB);

% DisAC, DisC, DisD, DisB are height values of each laser after averaging
%   each profile (Resulting from Distance_Estimation)

A_corrected = DisAC; % Distance of A
C_corrected = DisC;  % Distance of C
D_corrected = DisD;  % Distance of D
B_corrected = DisB;  % Distance of B

clear displacement_AD
clear displacement_AC
clear displacement_AB

%% Difference of C and A at same profile

timehorizon = 15001; % Number of profiles to optimize

for i=1:timehorizon
displacement_AC(i)=C_corrected((i))-A_corrected(i);
end

% Difference of D and A at same profile

for i=1:timehorizon
displacement_AD(i)=D_corrected((i))-A_corrected(i);
end

% Difference of B and A at same profile

for i=1:timehorizon
displacement_AB(i)=B_corrected((i))-A_corrected(i);
end

%% AC: Build A matrix and b vector
% A_matrix is the time it took C to get to A
% b_vector is the displacement vector

clear A_matrixAC b_vectorAC
b_vectorAC = zeros(timehorizon,1);
A_matrixAC =
zeros(size(A1500w1500,1),timehorizon+timeshiftAC(length(timeshiftAC)));
index=1;

for i=1:timehorizon
    if ~isnan(displacement_AC(i))
        for j=i:i+timeshiftAC(i)
            A_matrixAC(index,j)=1;
        end

        b_vectorAC(index,1)=displacement_AC(i);
        index=index+1;
    end
end
```

```
%% AD: Build A matrix and b vector
% A_matrix is the time it took D to get to A
% b_vector is the displacement vector

clear A_matrixAD b_vectorAD
b_vectorAD = zeros(timehorizon,1);
A_matrixAD =
zeros(size(A1500w1500,1),timehorizon+timeshiftAD(length(timeshiftAD)));
index=1;

for i=1:timehorizon
    if ~isnan(displacement_AD(i))

    for j=i:i+timeshiftAD(i)
        A_matrixAD(index,j)=1;
    end

    b_vectorAD(index,1)=displacement_AD(i);
    index=index+1;

    end
end

%% AB: Build A matrix and b vector
% A_matrix is the time it took B to get to A
% b_vector is the displacement vector

clear A_matrixAB b_vectorAB
b_vectorAB = zeros(timehorizon,1);
A_matrixAB =
zeros(size(A1500w1500,1),timehorizon+timeshiftAB(length(timeshiftAB)));
index=1;

for i=1:timehorizon
    if ~isnan(displacement_AB(i))

    for j=i:i+timeshiftAB(i)
        A_matrixAB(index,j)=1;
    end

    b_vectorAB(index,1)=displacement_AB(i);
    index=index+1;

    end
end

%% Equalizing the sizes of A matrices

A_matrixAC = [A_matrixAC zeros(size(A_matrixAC,1),size(A_matrixAB,2)-
size(A_matrixAC,2))];
A_matrixAD = [A_matrixAD zeros(size(A_matrixAD,1),size(A_matrixAB,2)-
size(A_matrixAD,2))];
```

```
%% Adding Misalignment columns (-1)

A_matrixAB = [A_matrixAB ones(size(A_matrixAB,1),3).*[-1 0 0]];
A_matrixAC = [A_matrixAC ones(size(A_matrixAC,1),3).*[0 -1 0]];
A_matrixAD = [A_matrixAD ones(size(A_matrixAD,1),3).*[0 0 -1]];

%% CVX L2

n = size(A_matrixAB,2); % length of timeseries
lambda = 0; % larger values flatten the result; smaller values have more
noise

cvx_begin
cvx_solver mosek % Mosek runs faster than standard CVX solver
cvx_solver_settings('MSK_IPAR_PRESOLVE_USE','MSK_PRESOLVE_MODE_OFF')

   variable x(n)

minimize norm(A_matrixAC*x-b_vectorAC)+norm(A_matrixAD*x-
b_vectorAD)+norm(A_matrixAB*x-b_vectorAB)
  subject to

cvx_end

%% Obtain displacement from velocity

Cumul=cumtrapz(x);


%% A & C correction
ss = 1;
N=length(DisC);
DO=zeros(1,N);
K=[];
K=round(abs(KC));
sC=max(K);

for j=sC:N-sC

    Delay=K(j);
    DO(j)=-Cumul(j+Delay)+ Cumul(j);

end

CF=DisC+ss*DO;
ACF=DisAC;

%% A & D correction
N=length(DisD);
DO=zeros(1,N);
K=[];
K=round(abs(KD));
```

```
sD=max(K);

for j=sD:N-sD
    Delay=K(j);
    DO(j)=-Cumul(j+Delay) + Cumul(j);
end

DF=DisD+ss*DO;
ADF=DisAD;

%% A & B correction
N=length(DisB);
DO=zeros(1,N);
K=[];
K=round(abs(KB));
sB=max(K);

for j=sB:N-sB
    Delay=K(j);
    DO(j)=-Cumul(j+Delay) + Cumul(j);
end

BF=DisB+ss*DO;
ABF=DisAB;


%% Matching error
figure(4)
hold all

%% *** AC Matching
ErrCc=ACF(:)-CF(:)-x(end-1);

plot(ErrCc);

%% *** AD Matching
ErrDc=ADF(:)-DF(:)-x(end);
plot(ErrDc);

%% *** AB Matching
ErrBc=ABF(:)-BF(:)-x(end-2);
plot(ErrBc);

ylabel('Distance(mm)')
xlabel('Profiles')
title('Estimated difference between Heads after matching and correction')
legend('A&C Matching corrected','A&D Matching corrected','A&B Matching
corrected');
```

Appendix C. User Manual for Deflection Estimation Code

The following are the steps to run the MATLAB codes from Appendix B. The figures mentioned are the result of running the code. An example of the results are given in Appendix D.

Processing steps

Run: Lasers_Loading_Filtering.m

* No tuning is needed. The filters are already working properly
--> Figure (1) : The scans in two dimensions
--> Figure (2) : The scans averaged over profiles
--> Figure (3) : The profiles normalized

* Check on Figure (1) that the flipping of the images was done properly by zooming and comparing a crack in all subplots
(This step is unnecessary if the direction of the laser-sensors is checked in the field)

* Assess the quality of the filtering from Figure (2) and the labeling of the laser-sensors
(A, C, D, and B) from the front to the back by analyzing the plots.

* Figure (3) to be used for acquisition delay estimation

Run: Gyro_Loading_Correcting.m

* if you receive an error about dimension mismatch: change w=1 (Default value is 3)
--> Figure (4) : Earth rotation correction
--> Figure (5) : Interpolation

* On figure (4), evaluate the earth correction, the corrected plot (in red) should get closer to the zero line.
* If it is farther from the zero line compared to the blue line, set the G_sign=-1 (Default value is 1)

* On figure (4), if the corrected gyro data is parallel to the zero line, it means the earth correction was done properly.
* Else G_slope needs to be estimated from the stationary data while accounting for time in ms
* For PRC: - G_slope = 3.2 for the Innovation Blvd
                - G_slope = 3.5 for the Creativity Blvd

* Figure (5), to be used for acquisition delay estimation

Save intermediate filtered data and keep final filtered data only to empty memory.

Evaluation of the acquisition delay

The objective is to detect the peaks of the oscillations in all data
and evaluate the delay between them. The delays are all evaluated with respect to Head B

Figure (3) is used for laser-sensors data and Figure (5) for the gyro data to fill the following table

- Set the datatip on the graph either at the beginning of the oscillation or on the first peak.

if the time is more than 10000 ms
- Export the cursor data value to the workspace
- Read its x position from workspace to be sure to account for the ms

Table C-1 Record of time of the peaks

	1st peak	2nd peak	3rd peak
Head A	9307	10141	10952
Head B	9318	10151	19062
Head C	9319	10152	10963
Head D	9305	10139	10950
Gyroscope	8484	9318	10125

Table C-2 Difference in delay with respect to Head B

	Delays			
Head A delay	-11	-10	-10	-10
Head C delay	1	1	1	1
Head D delay	-13	-12	-12	-12
Gyroscope delay	-834	-833	-837	-835

* These values are for test 3 of experiments on 08/07/2018

Timeline of the experiment from the Figure (1)

* Experiment started around 22000ms
* Experiment finished around 192000ms
- Values set by zooming on the subplots

* By analyzing the features in the four subplots, the time difference between heads with respect to head A scans are approximately:
Head C shift = 1000
Head D shift = 6000
Head B shift = 7000

Get the measurements of misalignment measured in the field

Lasers angles with respect to the beam (zero is set at the edge center of the beam)

```
GA = 0.598;
GC = 0.080;
GD = 0.019;
GB = 0.071;
```

Lasers distance from the edge

```
HA = 212.78;
HC = 210.65;
HD = 210.65;
HB = 212.01;
```

Lasers distances from the center of the beam

```
L = 3050;           % Length of the beam
%"O" being the center of the beam
OA=1200+114;        %Distance between O and A
OC=1200+114-120;    %Distance between O and C
OD=1200-178;        %Distance between O and D
OB=1200+263;        %Distance between O and B
```

Lasers distances from the center of the laser itself

```
OL=38-12+50;
```

Open: MatchingABCD_tuning.m

Set:

* nS anbd nE for start and end of the experiment evaluated in 5.

* Section and Subsection
In general 2000 is good. Change to 3000 for slow speeds and to 1000 for fast speeds
Subsection is set equal to Section but can be reduced down to 500 for recurrent matching

* ShiftAB, ShiftAD, ShiftAC, from point 5.

* Gyro_Delay; A_Delay; C_Delay; and D_Delay from point 4.

* L; OA; OC; OD; OB; OL from point 6.

Run: MatchingABCD_tuning.m

-->30 Figures of matching, 10 each (C to A, D to A, B to A)

if most of the matching are ok. Lines are going in the smae direction
Sometimes the first figure gives random results (up to 2 max)

Starting the third figure if the random behavior is still there, the matching parameters need to be retuned

- Start by retuning the Section
- If not enough check the Shifts.

Redo until getting good matching

Open: MatchingABCD.m

Apply same steps as point 7. with last tuning from point 8.

Run: MatchingABCD.m

--> Figure (1): Matches

- Evaluate the matching:

* If bad everywhere --> Retune and run again

* If bad locally --> Select the beginning time and the end time
of the portion that was matched correctly.

Run: PtoP_AC.m

- Set CropRowS, CropRowE, CropColS, CropColE based on the information from AoutC, Bout, Cout and Dout

- Code is set for A and C. Repeat for images of Head D and B

Run: Distance_estimation_All.m

- Set nS and nE by evaluating the good matching portion from the results of Figure (1)

- Set GA,GC,GD,GB and HA,HC,HD, and HB based on point 6.

--> Figure (2) : Estimated distance of different profiles after matching
--> Figure (3) : Time delay between Heads and Head A

13. Run: CVX_Deflection.m

- Set nS and nE based on computer's hardware capabilities: 15,000 profile optimization will work on most computers.

--> Figure(4) : Estimated difference between Heads after matching and correction

Appendix D. Deflection Estimation Example Results

Results from: Laser_loading_Filtering

What it does: Converts laser-sensor data from LabVIEW (TDMS extension) to individual matrices in MATLAB. The code then trims the data of each laser-sensor to exclude dead zones and applies filtering.

Significance and output: Three graphs are plotted for obtaining delay times between laser-sensors and gyroscope as well as seeing the general quality of the measurements.

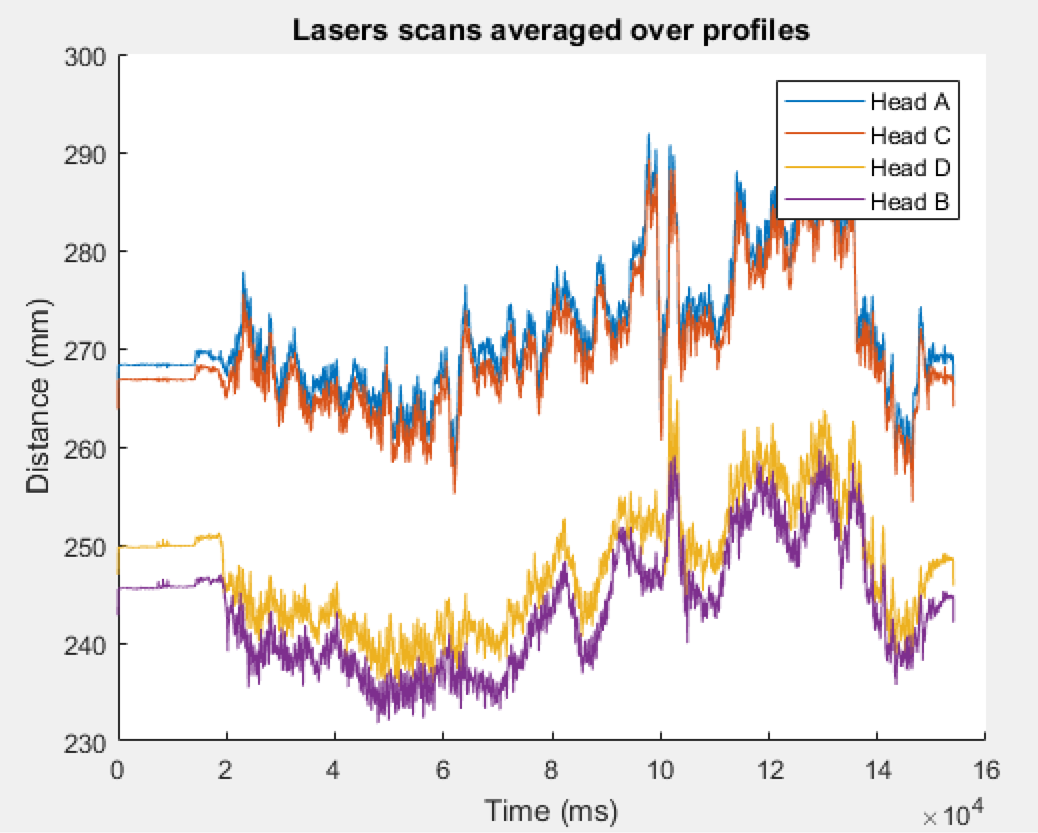


Figure D-1 Average of each scan line

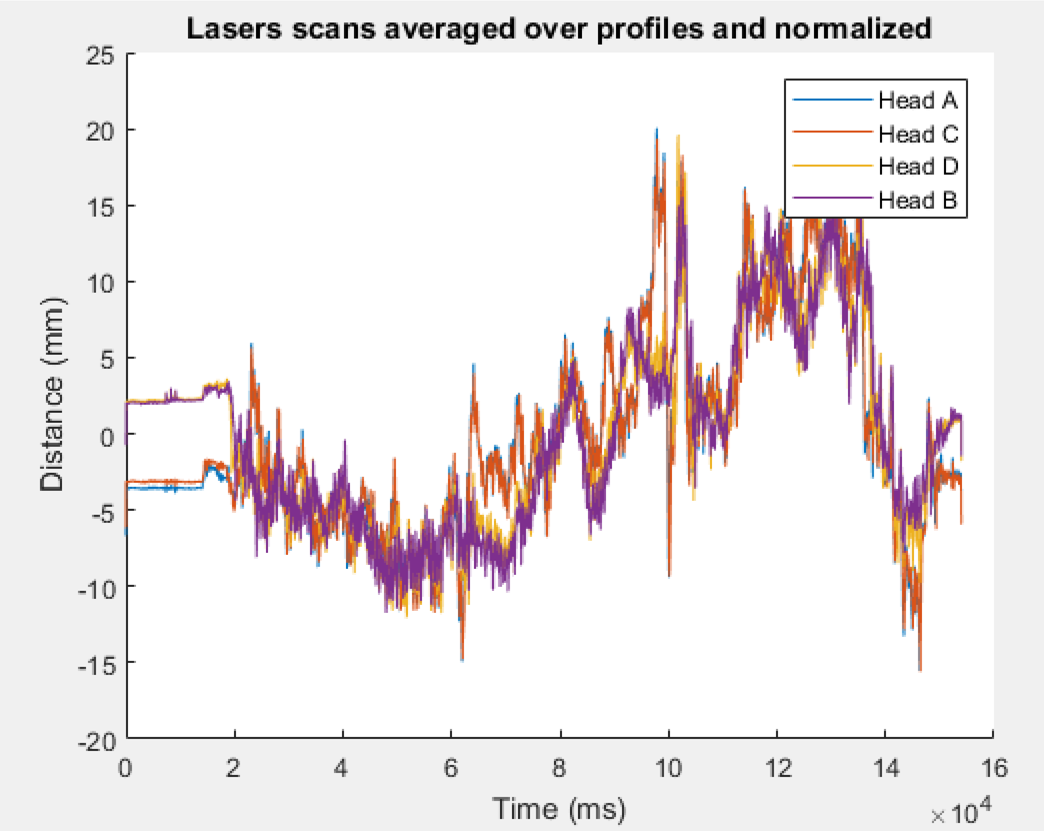


Figure D-2 Average of scan lines normalized to observe the delays between data

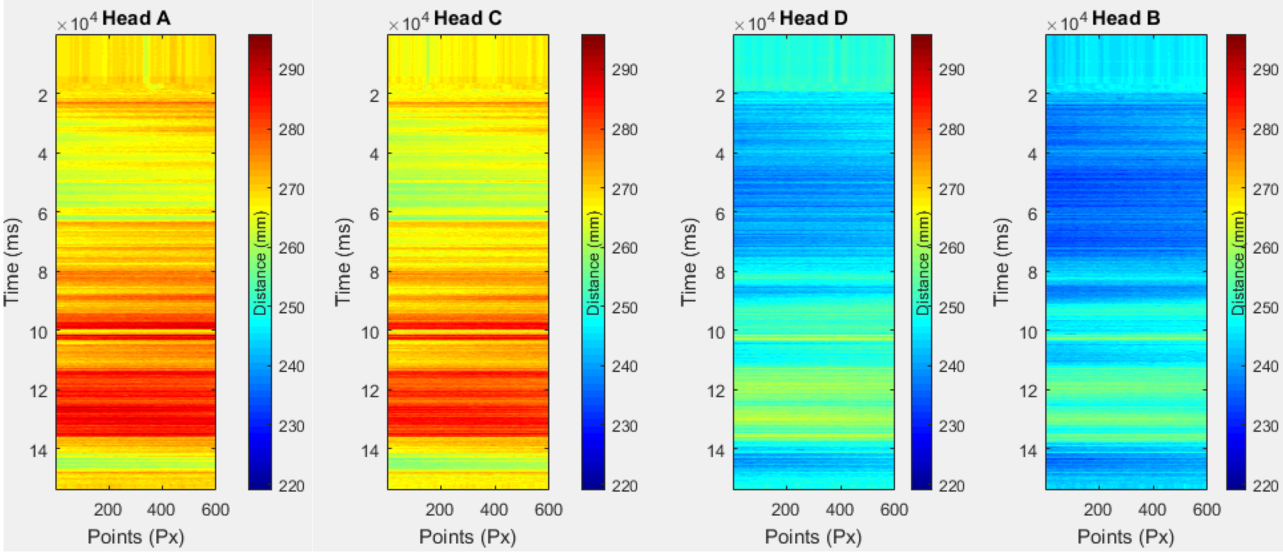


Figure D-3 Height measurements from each laser-sensor before matching

Results from Gyro_Loading

What it does: Converts measurement file from LabVIEW (TDMS extension) to a single MATLAB vector. The code corrects for the wrap-up, converts values to degrees, accounts for earth's rotation, and sets initial angle to zero.

Significance and output: Single plot of the before and after correction.

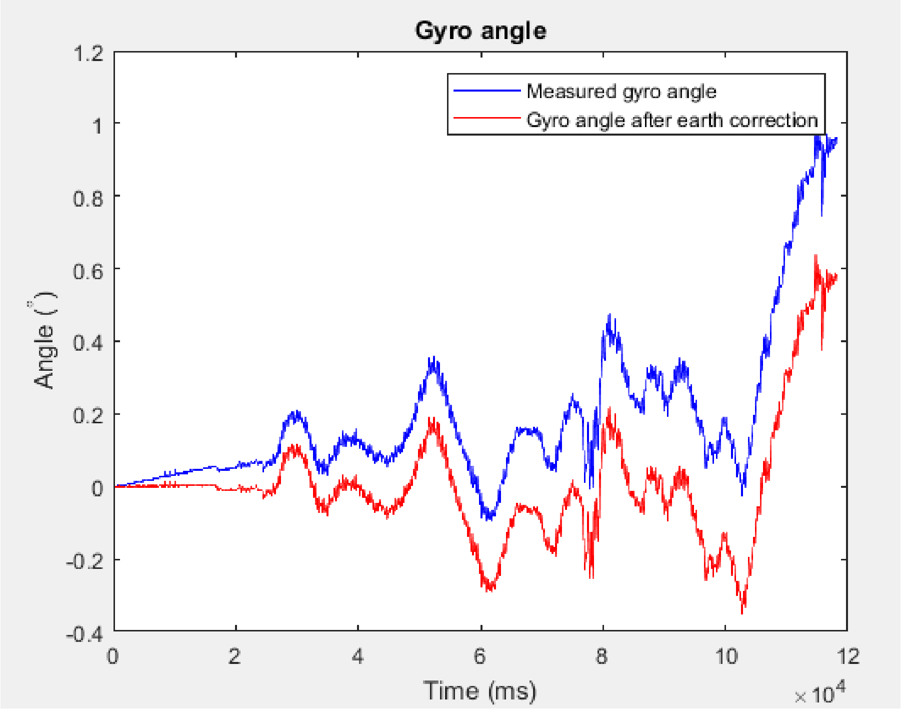


Figure D-4 Gyroscope comparison of the adjusted data based on earth's rotation

Results from MatchingABCD

What it does: Corrects laser-sensor data with the results from the gyroscope; the correction for the beam's rotation. Matches Heads C, D, and B to A. Converts the matrices to images, performs color enhancement, locates and matches key features, computes delay time, vector distances, and performs geometric corrections.

Significance and output: Corrects for the shifts between laser-sensor with respect to Head A as well as geometric corrections. Plots for the matching

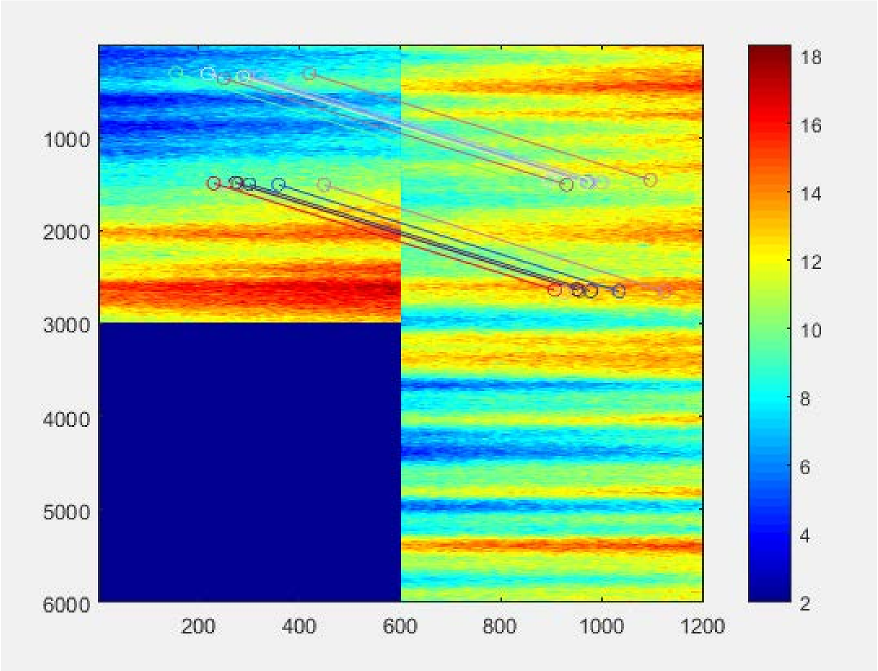


Figure D-5 Example of a match based on key features found in both laser-sensors

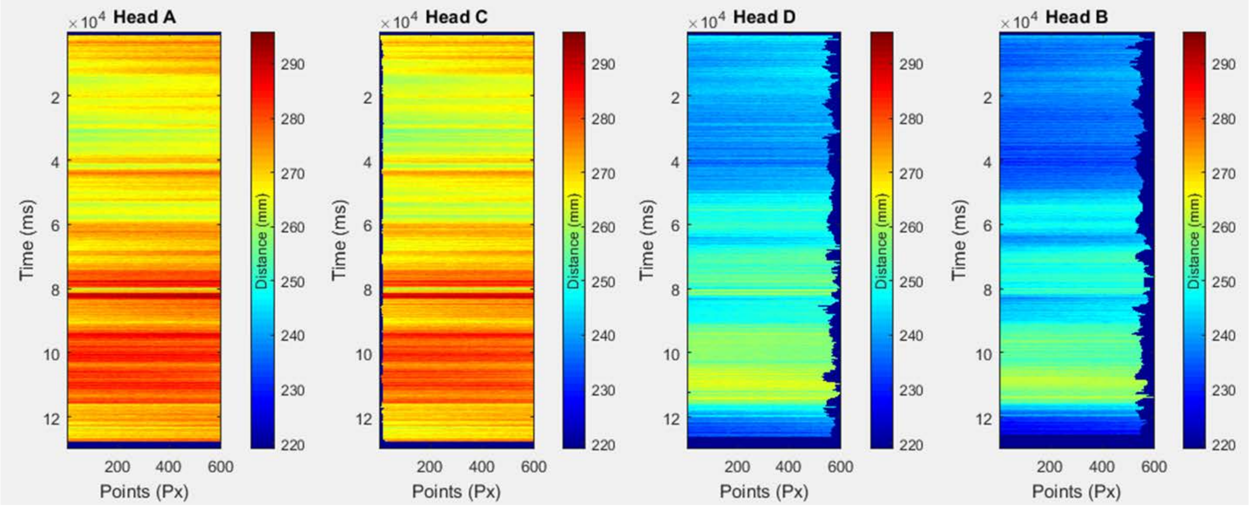


Figure D-6 Matching of all four laser-sensors

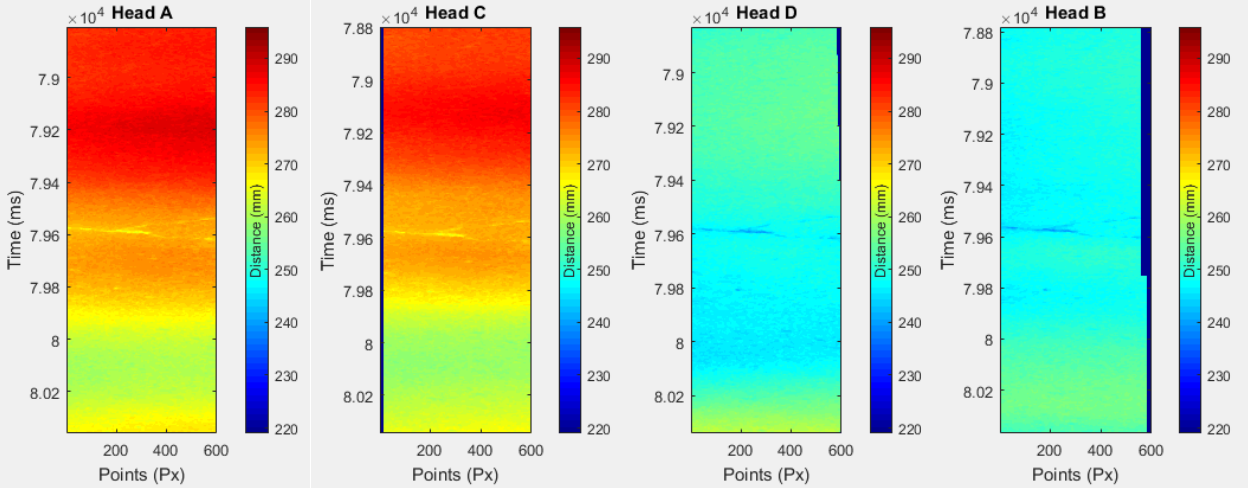


Figure D-7 Abnormality in the pavement, observed in all four sensors after matching

Results from PtoP_AC

What it does: Uses the results from the matching to line search the appropriate profile from image of Head A and corrects image of Head C, D, and B.

Significance and output: Outputs a matrix from each head that is matched profile by profile to the reference image, Head A.

Results from DistanceEstimation_all

What it does: Uses the results from the matching to compute the delay times and corrects for the misalignments.

Significance and output: Outputs the result of the delay time (velocity) as well as the distances from the sensor to the pavement before vibrational corrections.

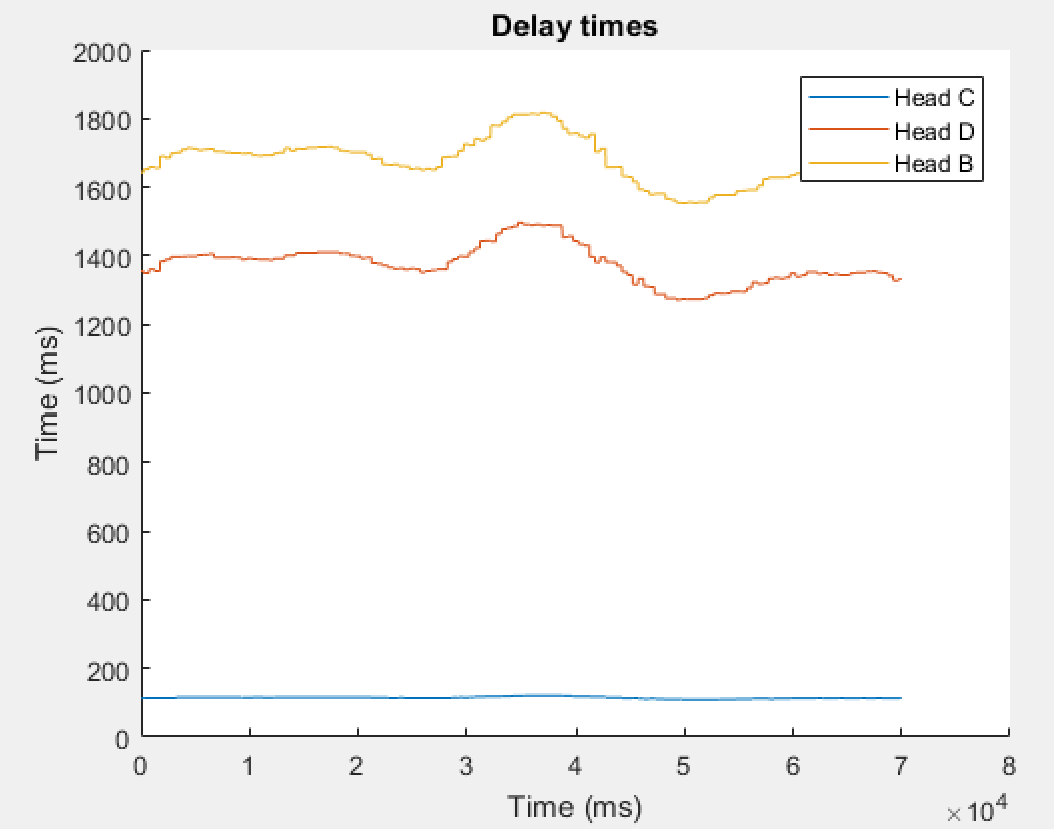


Figure D-8 Time for each laser-sensor to get to Head A

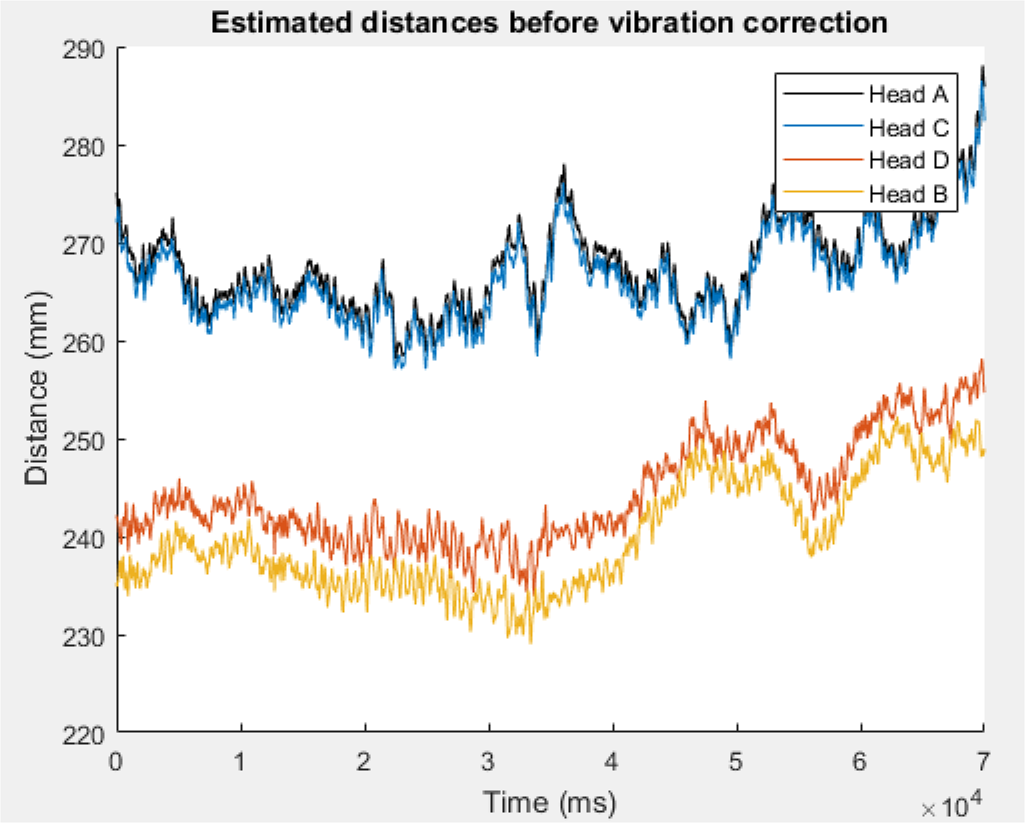


Figure D-9 Distances without accumulated vertical displacement

Results from Deflection_estimation

What it does: Performs a linear programming optimization for the road profile measured, then uses the output to correct for the vertical displacement of each laser-sensor with respect to A.

Significance and output: Plots the estimated deflection and vertical displacement of the center of the beam of the area tested by the DMD.

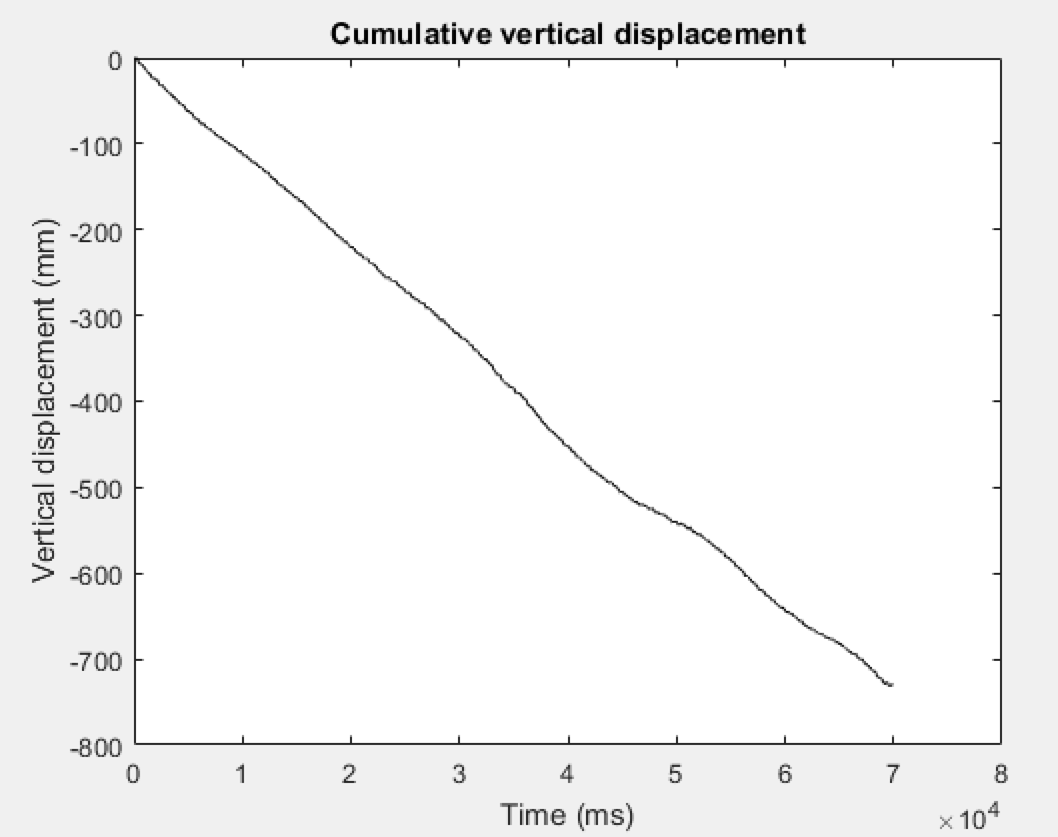


Figure D-10 Displacement from start to finish of the evaluation

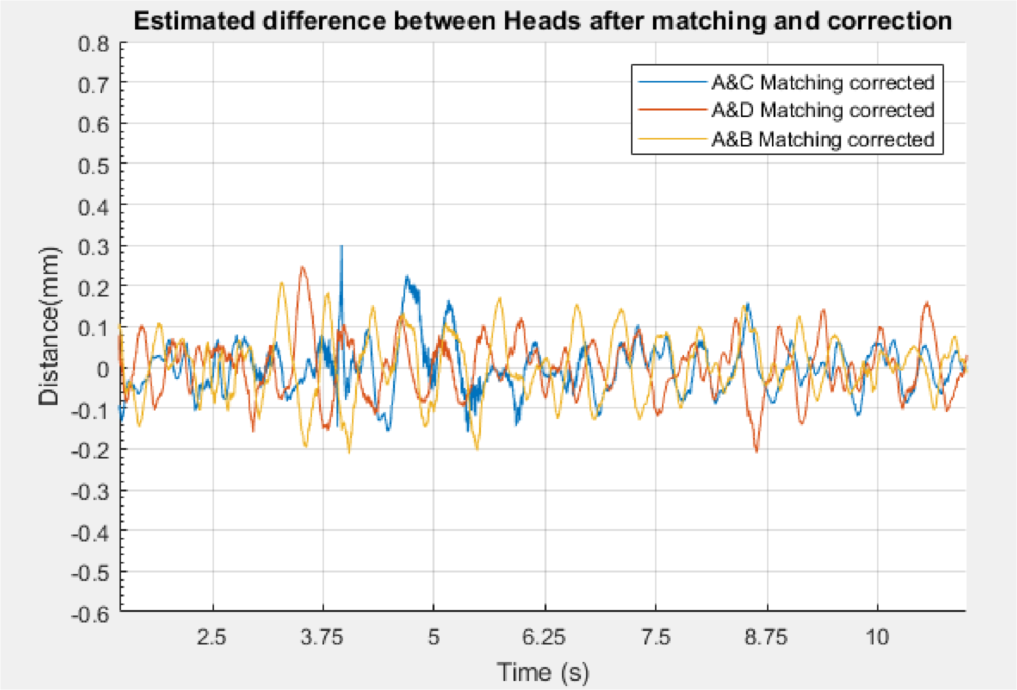


Figure D-10 Differences in laser-sensors